

# Um Modelo Lúdico para o Ensino de Conceitos de Programação de Computadores

Almir D. V. Santiago, Artur H. Kronbauer

Universidade Salvador – UNIFACS  
Salvador – BA – Brasil

{david.valente.santiago, arturhk}@gmail.com

***Abstract.** This paper describes the process of construction and validation of a ludic model for teaching computer programming concepts. The proposal is based on scientific studies that propose the replacement of textual approach by ludic metaphors. The model was specified in order to provide an alternative mechanism of algorithmic execution visualization. The proposed model validation process occurred through an experimental study in the classroom with 38 students. The results showed that students who participated in the study achieved a significant improvement in the understanding of the algorithms.*

***Resumo.** Este artigo descreve o processo de construção e validação de um modelo lúdico para o ensino de conceitos de programação de computadores. A proposta está fundamentada em trabalhos científicos que propõem a substituição da abordagem textual por metáforas lúdicas. O modelo foi especificado com o objetivo de fornecer um mecanismo alternativo de visualização de execução algorítmica. O processo de validação do modelo proposto ocorreu por meio de um estudo experimental, realizado em sala de aula, com 38 alunos. Os resultados obtidos demonstraram uma significativa melhora no entendimento dos algoritmos por parte dos alunos participantes do estudo.*

## 1. Introdução

A problemática do ensino de programação de computadores nas universidades é um assunto que tem sido alvo de discussões em diversos trabalhos científicos recentes, tais como, Galdino et al. (2015), Ramos et al. (2015), Amaral et al. (2015), Ambrosio et al. (2011), Hernandez et al. (2010) e Eagle e Barnes (2008). Tais trabalhos reconhecem que a programação é um assunto complexo e envolve uma grande quantidade de conhecimentos. Além disso, eles relatam grande preocupação com os elevados índices de desistência existentes nos cursos de computação.

A discussão na área do ensino/aprendizagem de programação de computadores reconhece que as matérias introdutórias de programação são responsáveis pelas altas taxas de desistência, já nos primeiros semestres. A quantidade de alunos que desistem dos cursos de computação após o contato com as primeiras matérias de programação chega a 40% [Eagle e Barnes, 2008]. Galdino et al. (2015) apontam a preocupação existente com estes elevados índices de evasão e os atribuem às dificuldades enfrentadas pelos iniciantes com as matérias de lógica de programação. Além disso, existem evidências de que grande parte dos alunos não conseguem escrever programas razoáveis após dois semestres com matérias de programação e que os livros didáticos abordam apenas processos de

programação sem a devida contextualização do problema [Caspersen and Bennedsen, 2007].

De maneira resumida, o estudo da bibliografia científica sobre o ensino/aprendizagem de programação de computadores apresentou evidências de que:

- Uma metodologia de ensino de conceitos de programação de computadores baseada puramente em sintaxe é prejudicial para o processo de aprendizado dos alunos iniciantes.
- Os alunos são desestimulados devido às dificuldades com as matérias introdutórias de programação, pois não conseguem criar os corretos modelos mentais acerca do conhecimento que está sendo aprendido.
- Existe a necessidade de novas propostas de metodologias instrucionais, as quais não sejam baseadas puramente em código textual.

A partir da identificação da problemática em questão, percebeu-se a oportunidade de trabalhar em propostas direcionadas a mitigar os elevados índices de desistência dos alunos iniciantes em computação. Nesse contexto, este artigo traz como contribuição a proposta de uma metodologia lúdica para o ensino de conceitos de programação de computadores, direcionada a alunos dos cursos de computação e engenharias.

Para alcançar este objetivo, foram executados os seguintes passos: (i) definir um modelo lúdico aplicável a um conceito introdutório de programação; (ii) implementar, a partir do modelo, três provas de conceito, utilizando algoritmos recursivos; e (iii) realizar um estudo experimental com alunos iniciantes na área de programação, com o intuito de validar a metodologia proposta.

## 2. Trabalhos relacionados

O uso de abordagens lúdicas para o ensino de conceitos de programação de computadores é uma iniciativa que tem se apresentado promissora na bibliografia científica. Na revisão sistemática, realizada por Ramos et al. (2015), é apontada a redução do índice de evasão dos alunos iniciantes em computação devido ao uso de jogos. Os autores afirmam que a abordagem lúdica funcionou como fator motivacional para o ensino.

Amaral et al. (2015) realizaram um estudo experimental sobre o uso de uma plataforma denominada *Robocode* como ferramenta de apoio ao ensino de algoritmos e programação para alunos do ensino médio. Eles concluem que a ferramenta se mostrou como facilitadora para o processo de aprendizagem. Os autores salientaram que o uso da ferramenta lúdica aumentou o interesse dos alunos pelo assunto.

Um outro trabalho científico apresenta um jogo denominado *Wu's Castle* [Eagle e Barnes, 2008], que propõe o uso de metáforas lúdicas para o ensino de conceitos de programação (Figura 1). Os autores do trabalho propuseram metáforas para o ensino dos conceitos computacionais de *arrays* (Figura 1a) e *estruturas de repetição* (Figura 1b). Os alunos, por meio da manipulação de um personagem, executam as ações referentes às execuções algorítmicas impostas pelos respectivos conceitos computacionais, conseguindo, desta forma, melhor compreendê-los. Os autores concluem que os alunos que aprenderam por meio das metáforas lúdicas obtiveram ganhos significativos de aprendizado quando comparados ao grupo de controle.



Figura 1. Metodologia lúdica [Eagle e Barnes 2008].

Covington e Benegas (2005) propuseram uma mudança da abordagem do ensino de programação de computadores baseada em sintaxe para uma abordagem baseada no desenvolvimento de modelos mentais sobre padrões de programação. Os autores criticam a abordagem vigente para o ensino de programação, denominada por eles de *syntax-driven*, e sugerem uma nova abordagem, denominada *schema-driven*. A principal prerrogativa para a proposta é que a abordagem *syntax-driven* foca apenas em modelos textuais de programação, quando deveriam desenvolver nos alunos a habilidade de reconhecer padrões de resolução de problemas do mundo real. Os autores concluem que o ensino da sintaxe só deve ser aplicado quando o aluno já tiver desenvolvido a capacidade de reconhecer e resolver problemas, dessa forma, a sintaxe poderia ser contextualizada em um domínio específico e bem conhecido.

### 3. O modelo proposto

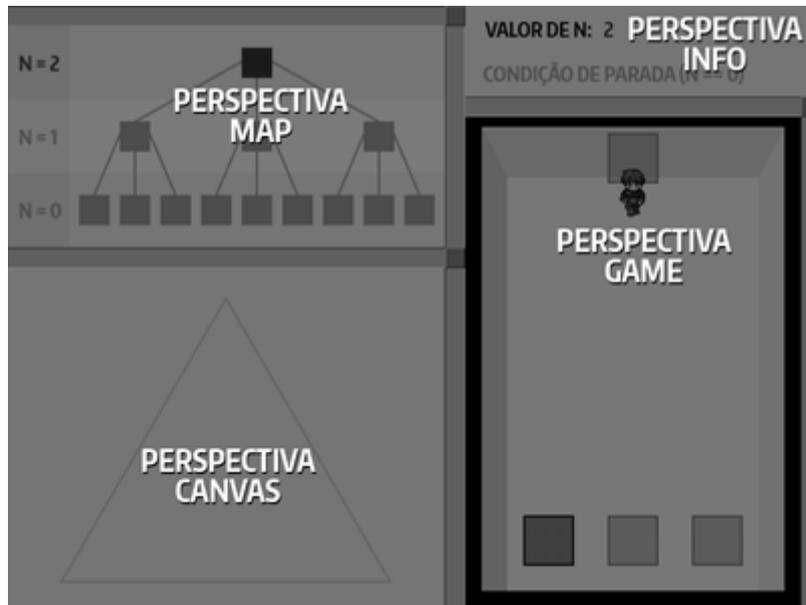
O modelo proposto por este trabalho teve o objetivo de especificar a construção de uma ferramenta a ser aplicada como metodologia lúdica para o ensino de conceitos de programação de computadores. Para validar o modelo, o assunto referente a Recursividade foi escolhido como prova de conceito, devido à sua complexidade de aprendizado por parte dos alunos iniciantes em programação. Wilcocks e Sanders (1994) afirmam que a recursividade é um conceito dinâmico e que os métodos instrucionais utilizados para o seu ensino são inadequados. Eles ainda salientam sobre a necessidade de desenvolver no aluno um senso abstrato para que se possa visualizar mentalmente o fluxo de execução e controle recursivo.

#### 3.1 Caracterização do modelo

A interface do modelo (Figura 2) fornece quatro perspectivas distintas:

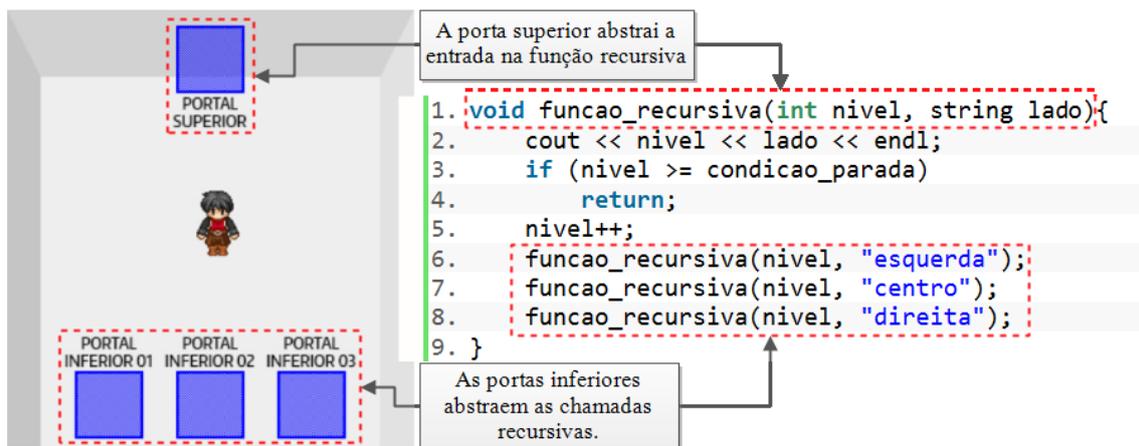
1. *Game* – fornece a abstração lúdica para substituir o código fonte textual do algoritmo.
2. *Canvas* – fornece a representação gráfica modelada pelo algoritmo aplicado ao modelo. Tem o objetivo de ilustrar como as operações modelam o problema a cada etapa da execução recursiva.

3. *Map* – fornece o mapa conceitual das chamadas recursivas, permitindo que o aluno consiga identificar o estágio atual da resolução algorítmica recursiva.
4. *Info* – fornece informações textuais sobre as variáveis que controlam o fluxo recursivo e outras informações relevantes.



**Figura 2. Modelo lúdico proposto.**

O modelo especifica um ambiente de simulação algorítmica. O fluxo de execução consiste em adentrar por portais na parte inferior da perspectiva *Game* e, pelo portal superior, retornar ao mesmo cenário, simulando o fluxo de repetições impostas pelas funções recursivas (Figura 3). Ao passo que o aluno executa as etapas, são apresentadas alterações nas outras perspectivas, com o objetivo de demonstrar para o aluno outros pontos de vista sobre a resolução do problema modelado pelo algoritmo recursivo.



**Figura 3. Metáfora das portas.**

### 3.2 Implementação do Modelo

Os algoritmos recursivos para a resolução do *Triângulo de Sierpinski*, da *Torre de Hanoi* e do *Cálculo Fatorial* foram implementados com base no modelo especificado, no âmbito

de tornar possível a sua avaliação e validação através de um estudo experimental. A implementação foi realizada por meio da ferramenta de desenvolvimento *Construct 2*.

#### 4. Metodologia

Nesta seção estão contidos os procedimentos metodológicos empregados durante todo o processo de pesquisa realizado durante este estudo.

##### 4.1 Quadro metodológico

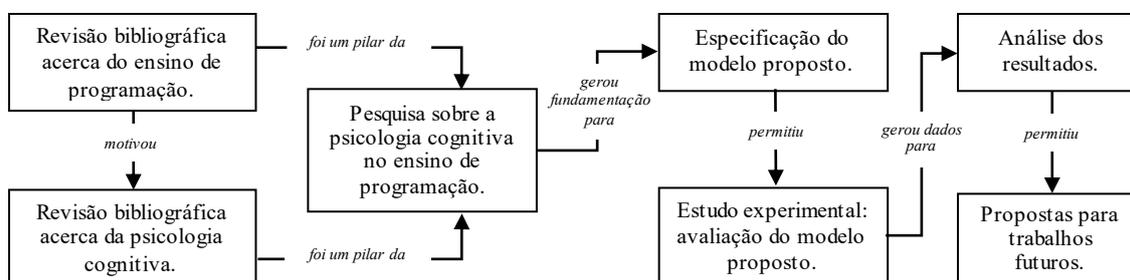
Segundo Gil (2000) as pesquisas científicas podem ser classificadas segundo dois critérios: quanto aos **objetivos** e aos **procedimentos técnicos**. Tendo em vista os objetivos deste trabalho, optou-se por utilizar os tipos descritos na Tabela 1.

**Tabela 1. Quadro metodológico.**

QUANTO AOS OBJETIVOS	QUANTO AOS PROCEDIMENTOS TÉCNICOS
<b>Exploratória:</b> Compreender a psicologia da programação.	<b>Levantamento bibliográfico:</b> Identificar os trabalhos sobre o uso da psicologia cognitiva na programação.
<b>Explicativa:</b> Explicar as dificuldades no aprendizado de programação.	<b>Estudo experimental:</b> Avaliar o modelo criado para o ensino de conceitos de programação de computadores.

##### 4.2 Etapas da pesquisa

O processo de pesquisa consistiu nas etapas ilustradas na Figura 4. A primeira etapa refere-se a uma revisão bibliográfica preliminar sobre o ensino de programação, onde foi percebida a associação de algumas iniciativas com teorias relacionadas à psicologia cognitiva. Essa percepção motivou a pesquisa sobre o referido tema. Tais estudos formaram o alicerce para a proposta de um mapeamento sistemático acerca da psicologia cognitiva no ensino de conceitos de programação de computadores. Os resultados deste mapeamento serviram como fundamentação teórica durante a especificação do modelo lúdico. A etapa final do trabalho consistiu em avaliar e validar o modelo por meio da realização de um estudo experimental.



**Figura 4. Etapas da pesquisa.**

##### 4.3 Planejamento do Estudo Experimental

O estudo experimental teve como objetivo avaliar o modelo lúdico proposto. Para tanto, seu planejamento foi dividido em seis fases distintas, tomando como base as diretrizes

propostas pelo *framework DECIDE* [Sharp et al., 2007], que norteou a especificação dos passos realizados durante todas as fases do experimento.

#### **4.3.1 Determinar o objetivo da análise**

Obter informações sobre a eficácia de uma metodologia lúdica fundamentada na psicologia cognitiva para o ensino de conceitos de programação de computadores.

#### **4.3.2 Explorar perguntas a serem respondidas**

- Os alunos enfrentaram dificuldades para entender o funcionamento da ferramenta?
- A ferramenta conseguiu construir os modelos mentais acerca do fluxo de execução dos algoritmos recursivos?
- Ao final do experimento foi possível observar uma melhora geral de desempenho nos testes que pudesse ser atribuída à metodologia proposta?
- Os alunos aprovaram a metodologia como ferramenta de apoio ao ensino de conceitos de programação de computadores?

#### **4.3.3 Escolher método de avaliação**

A execução do estudo experimental fez uso de dois tipos de avaliações: (i) avaliação quantitativa, realizada por meio do método *pretest-posttest* [Dimitrov e Rumrill 2003]; e (ii) avaliação qualitativa, realizada por meio de questionário formatado em função da escala de Likert (1932).

#### **4.3.4 Identificar e administrar as questões práticas**

O estudo experimental foi realizado no dia 20 de abril de 2016, no laboratório de informática, com a participação de três turmas da disciplina de Estrutura de Dados, dos cursos de Engenharia da Computação, Elétrica e Mecatrônica, contemplando 38 participantes. Os alunos foram divididos em três turmas, de acordo com o turno em que a disciplina foi ministrada. O laboratório foi previamente preparado com as ferramentas lúdicas desenvolvidas, com o objetivo de validar a abordagem.

#### **4.3.5 Decidir como lidar com as questões éticas**

O projeto desta pesquisa foi submetido ao Comitê de Ética, identificado na Plataforma Brasil por meio do identificador CAAE: 54856416.3.0000.5033, sendo devidamente aprovado. O experimento foi conduzido preservando o anonimato dos participantes. Foi especificado no *Termo de Consentimento Livre e Esclarecido* que as informações pessoais dos participantes não seriam divulgadas. Além disso, todos os voluntários possuíam mais de dezoito anos de idade e gozavam de plena capacidade física e mental.

#### **4.3.6 Estabelecer forma de avaliar, interpretar e apresentar os resultados**

As informações geradas com o estudo experimental abrangeram dados quantitativos e subjetivos que, após avaliados e interpretados, geraram gráficos e resultados que serão apresentados e discutidos na próxima seção.

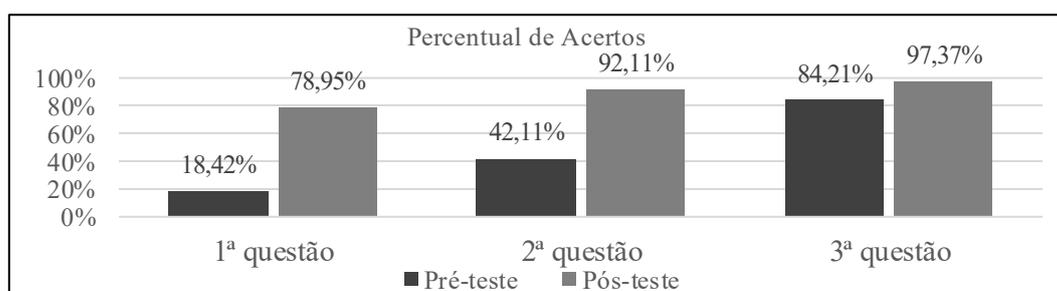
## 5. Resultados

A análise quantitativa teve o objetivo de medir numericamente a eficácia do modelo proposto. Para tanto, foi submetido aos alunos um questionário contendo três perguntas de múltipla escolha, referentes aos três algoritmos utilizados na validação do modelo. O questionário foi submetido tanto no pré-teste, como no pós-teste, com o intuito de avaliar o contraste de desempenho antes e depois da metodologia lúdica ser aplicada.

A análise qualitativa teve por objetivo verificar o grau de concordância ou discordância dos alunos com relação aos diversos aspectos da metodologia lúdica empregada. A análise foi aplicada por meio de um questionário contendo sete perguntas formatadas em função da escala de Likert (1932) e uma questão discursiva, na qual os alunos opinaram de maneira livre sobre a experiência com a metodologia.

### 5.1 Análise quantitativa

O gráfico ilustrado na Figura 5, mostra as médias dos resultados para todos os 38 alunos participantes do experimento. O contraste entre os índices de acerto entre o pré-teste e o pós-teste sugere o real ganho de desempenho provido pela metodologia.



**Figura 5. Resultados da análise quantitativa.**

A primeira questão (referente ao *Triângulo de Sierpinski*) apresentou o menor índice de acertos, tendo 7 alunos acertado o pré-teste e 30 o pós-teste. A segunda questão (referente à *Torre de Hanoi*) apresentou o índice mediano de acertos, tendo 16 alunos acertado a questão no pré-teste e 35 no pós-teste. A terceira questão (referente ao Cálculo do Fatorial) apresentou o maior índice de acertos, 32 alunos acertaram a questão no pré-teste e 37 no pós-teste.

Os resultados expostos na Tabela 2 mostram que houve ganho de desempenho promovido pela intervenção proposta pelo estudo experimental. É possível observar que, para todas as questões, houveram variações positivas, totalizando, entre as três questões, uma média de 41,23% de ganho de desempenho por parte dos alunos.

**Tabela 2. Dados sobre o ganho de desempenho.**

Questão	Desempenho Pré-teste	Desempenho Pós-teste	Ganho de desempenho ( <i>pós-teste - pré-teste</i> )
1ª Questão	18,42%	78,95%	<b>60,53%</b>
2ª Questão	42,11%	92,11%	<b>50,00%</b>
3ª Questão	84,21%	97,37%	<b>13,16%</b>
Média	48,24%	89,47%	<b>41,23%</b>

## 5.2 Análise qualitativa

Na análise qualitativa foram submetidas sete perguntas baseadas na escala de Likert (1932), no âmbito de obter a opinião dos alunos com relação aos aspectos da metodologia, e uma pergunta discursiva:

1. Dificuldade enfrentada para responder ao pré-teste.
2. Dificuldade enfrentada para responder ao pós-teste.
3. Dificuldade enfrentada para entender o funcionamento da metodologia proposta.
4. Eficácia da metodologia para a compreensão do algoritmo de *Sierpinski*.
5. Eficácia da metodologia para a compreensão do algoritmo de *Hanoi*.
6. Eficácia da metodologia para a compreensão do algoritmo do cálculo do *Fatorial*.
7. Relevância de uma abordagem que utiliza metáforas lúdicas, no lugar da textual.
8. *Escreva com suas palavras sobre a proposta lúdica para o ensino de conceitos de programação e o que você mudaria ou adicionaria para melhorá-la.*

A Tabela 3 apresenta os resultados obtidos para cada questão submetida na análise qualitativa. De maneira geral, os resultados demonstram que houve real aceitação da metodologia por parte dos alunos.

**Tabela 3. Resultados da análise qualitativa.**

Questão	Valores da escala de Likert				
	<i>1-muito difícil</i>	<i>2-difícil</i>	<i>3-razoável</i>	<i>4-fácil</i>	<i>5-muito fácil</i>
1ª	5,3%	<b>47,4%</b>	39,5%	7,9%	0,0%
2ª	0,0%	0,0%	10,5%	<b>57,9%</b>	31,6%
3ª	0,0%	2,6%	23,7%	<b>36,8%</b>	<b>36,8%</b>
Questão	<i>1-muito eficaz</i>	<i>2-pouco eficaz</i>	<i>3-razoável</i>	<i>4-eficaz</i>	<i>5-muito eficaz</i>
4ª	0,0%	2,6%	7,9%	<b>55,3%</b>	34,2%
5ª	0,0%	5,3%	7,9%	<b>47,4%</b>	39,5%
6ª	0,0%	0,0%	0,0%	39,5%	<b>60,5%</b>
Questão	<i>1-nada relevante</i>	<i>2-pouco relevante</i>	<i>3-razoável</i>	<i>4-relevante</i>	<i>5-muito relevante</i>
7ª	0%	0%	2,6%	23,7%	<b>73,7%</b>

No que diz respeito à oitava questão (questão discursiva) foi possível reafirmar a aceitação da metodologia. Os alunos observaram que a metodologia lúdica:

- Torna mais eficiente a compreensão do problema.
- É uma inovação metodológica.
- Torna concreto um conceito abstrato.
- Aumenta o interesse pela programação.
- Tem grande potencial como metodologia complementar a ser implementada nas universidades.
- Deve ser utilizada em conjunto com a abordagem baseada em código.

## 6. Conclusões e Trabalhos Futuros

No que diz respeito aos pontos positivos do trabalho, os resultados expressos nos gráficos quantitativos do estudo experimental deixam claro o fato de que houve um real ganho de desempenho para o entendimento dos algoritmos recursivos por parte dos alunos, e que este ganho de desempenho sugere a eficácia da intervenção experimental proposta por este trabalho. Os resultados expressos nos gráficos qualitativos deixam claro o fato de

que houve uma real aceitação da proposta pelos alunos. Portanto, com base nas análises pode-se responder às perguntas definidas para o estudo experimental (Seção 4.3.2):

- *Os alunos enfrentaram dificuldades para entender o funcionamento da ferramenta?* **Resposta** – Não. Conforme ilustra a terceira Questão da Tabela 3, os alunos avaliaram a ferramenta como fácil e muito fácil.
- *A ferramenta conseguiu construir os modelos mentais acerca do fluxo de execução dos algoritmos recursivos?* **Resposta:** Sim. Conforme os próprios comentários dos alunos, houve um melhor esclarecimento acerca da lógica de funcionamento dos algoritmos apresentados.
- *Ao final do experimento foi possível observar uma melhora geral de desempenho nos testes que pudesse ser atribuída à metodologia proposta?* **Resposta:** Sim. Conforme ilustra a Tabela 2, houve melhora significativa de desempenho dos alunos entre o pré-teste e o pós-teste.
- *Os alunos aprovaram a metodologia como ferramenta de apoio ao ensino de conceitos de programação de computadores?* **Resposta:** Sim. Conforme os próprios comentários dos alunos, houve grande aceitação da metodologia e expectativa para que seja utilizada como ferramenta de apoio no ensino de conceitos de programação de computadores nas universidades.

Com relação aos pontos negativos do modelo, as respostas apresentadas para a questão discursiva (Questão 8) indicaram que os alunos, apesar de terem observado vantagens no aprendizado por meio de metáforas lúdicas, sentiram a falta da visualização do código fonte. Neste sentido, chegou-se à conclusão de que a visualização simultânea do código é importante para uma melhor compreensão do algoritmo, pois permite que o aluno observe e relacione as metáforas com as etapas algorítmicas descritas no código textual.

No que diz respeito ao objetivo geral de propor uma metodologia lúdica para o ensino de conceitos de programação de computadores, conclui-se que este trabalho cumpriu o seu papel, realizando todas as etapas necessárias para o alcance do objetivo. Quanto à problemática sobre como mitigar os elevados índices de desistência, ocasionadas pelas dificuldades com as matérias introdutórias de programação, conclui-se que este trabalho contribuiu de forma significativa, mesmo que ainda inicial. Entende-se que a implementação de metodologias lúdicas baseadas em metáforas, aliadas à abordagem textual utilizada atualmente nas universidades, apresentou-se como uma proposta promissora.

As perspectivas para a continuidade desta pesquisa foram idealizadas de acordo com as seguintes etapas: (i) incluir a perspectiva *CODE*, na qual o aluno poderá visualizar o código fonte gerado pelo algoritmo; (ii) implementar mais algoritmos recursivos, com o intuito de avaliar a visualização de outros problemas; e (iii) extensão da metodologia aos outros conceitos computacionais, tais como, estruturas de repetição, *arrays*, gerenciamento de memória, entre outros descritos na bibliografia científica.

No que se refere à metodologia e aos processos de avaliação, pretende-se realizar mais experimentos com alunos universitários, incluindo novos participantes, grupos de controles e utilizar novas técnicas de análise estatística como, por exemplo, o método *Analysis of Variance* (ANOVA).

## Referências

- Amaral, L.R.; Silva, G.B.; Pantaleão E. Plataforma Robocode como Ferramenta Lúdica de Ensino de Programação de Computadores- Extensão Universitária em Escolas Públicas de Minas Gerais. In: Simpósio brasileiro de informática na educação SBIE; 2015, out 26-30; Maceió, AL, Brasil. 2015. p.200-208.
- Ambrosio, A.P.; Costa, F.M.; Almeida, L.; Franco, A.; Macedo, J. Identifying cognitive abilities to improve CS1 outcome. In: Frontiers in education conference (FIE); 2011, out 12-15; Rapid City, SD, USA. IEEE; 2011. p.F3G-1 - F3G-7.
- Caspersen, M.E.; Bennedsen, J. Instructional design of a programming course. In: International computing education research ICER '07; 2007, set 15-16; Atlanta, GA, USA. ACM; 2007. p. 111-122.
- Covington, R.; Benegas, L. A cognitive based approach for teaching programming to computer science and engineering students. In: ASEE PEER 2005 annual conference; 2005, jun 12-15; Portland, OR, USA. ASEE; 2005. p.10.17.1-10.17.25.
- Dimitrov, D.M.; Rumrill, P.D. Pretest-posttest designs and measurement of change. *Work* (Reading, Mass.) 2003, 20, 159-65.
- Eagle, M.; Barnes, T. Wu's Castle: Teaching arrays and loops in a game. In: 13th annual conference on Innovation and technology in computer science education ITiCSE '08; 2008, jun 30 – jul 02; Madrid, Spain. ACM; 2008. p. 245-249, 2008.
- Galdino, C.B.T.; Neto, S.R.S.; Costa, E.B. KidCoder: Uma Proposta de Ensino de Programação de forma Lúdica. In: Simpósio brasileiro de informática na educação SBIE; 2015, out 26-30; Maceió, AL, Brasil. 2015. p.687-691.
- Gil, A.C. Como Elaborar Projetos de Pesquisa. 5ª ed. São Paulo: Atlas S.A., 2000.
- Hernandez, C.C.; Silva, L.; Segura, R.A.; Schimiguel, J.; Ledon, M.F.; Bezerra, L.N.; et al. Teaching Programming Principles through a Game Engine. *Sciences-New York*, p. 1-8, 2010.
- Likert, R. A Technique for the Measurement of Attitudes. *Archives of Psychology*, 1932, 140, 1-55.
- Ramos, V.; Freita, M.; Galimbert, M.; Mariani, A.C.; Wazlawick, R. A Comparação da realidade mundial do ensino de programação para iniciantes com a realidade nacional: Revisão sistemática da literatura em eventos brasileiros. In: Simpósio brasileiro de informática na educação SBIE; 2015, out 26-30; Maceió, AL, Brasil. 2015. p.318-327.
- Sharp, H.; Rogers, Y.; Preece, J. Interaction design: beyond human-computer interaction. 2ª ed. New York: John Wiley & Sons, 2007.
- Wilcocks, D.; Sanders, I. Animating recursion as an aid to instruction. *Computers & Education* 1994, 23, 221-226.