

## **Método de Ensino de Programação Mediada por Simulação: Um Estudo de Caso no Curso Técnico Integrado em Informática**

**Bruno Gonçalves Lopes<sup>1</sup>, Walisson Santos Duarte<sup>1</sup>,  
Tiago do Carmo Nogueira<sup>1</sup>, Renata Francisca Ferreira Lopes<sup>1</sup>, Deller Jemes Ferreira<sup>2</sup>**

<sup>1</sup>Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso (IFMT)  
Caixa Postal 78.600-000 – Barra do Garças – MT – Brazil

<sup>2</sup>Instituto de Informática – Universidade Federal de Goiás, Goiânia, Brazil  
Alameda Palmeiras, Quadra D, Câmpus Samambaia Caixa Postal: 74690-900

tiago.nogueira@bag.ifmt.edu.br, renata.lopes@bag.ifmt.edu.br

**Abstract.** *One of the problems mentioned in the literature in Computer courses is the high dropout rate caused by difficulties experienced by students in subjects related to algorithms or programming language. Thus, games have been used as introductory teaching motivational strategies programming, allowed interactivity and creating new experiences in the teaching-learning process. So, to minimize these problems, this paper proposes the use a programming teaching method mediated by simulation. The results showed significant differences between the proposed programming teaching method and the traditional method. There was also best performance among students who used the method brokered by simulation when compared to students who used traditional method.*

**Resumo.** *Um dos problemas indicado pela literatura nos cursos de Computação é o alto índice de evasão causado pelas dificuldades sentidas pelos alunos nas disciplinas relacionadas a algoritmos ou linguagem de programação. Neste sentido, Jogos têm sido utilizados como estratégias de motivação de ensino introdutório de programação, permitido interatividade e criando novas experiências no processo de ensino-aprendizagem. Assim, para minimizar estes problemas, este artigo propõe a utilização de um Método de Ensino de Programação Mediado por Simulação. Os resultados demonstraram diferenças significativas entre o método de ensino de programação proposto e o método tradicional. Verificou-se também melhores desempenhos entre os alunos que utilizaram o método mediado por simulação quando comparado aos alunos que utilizaram o método tradicional.*

### **1. Introdução**

Uma das características marcantes da atualidade é a rápida evolução que os recursos tecnológicos vêm sofrendo. A velocidade de introdução de novos meios faz com que novas perspectivas e potencialidades surjam constantemente, transformando o cotidiano de cada um. A demanda de produção de softwares cresce de acordo com essa evolução. No entanto, enquanto esta demanda aumenta, os cursos de programação tendem a diminuir [Barnes et al. 2008].

Dessa forma, vários estudos na literatura demonstram a importância e as dificuldades relacionadas ao ensino e aprendizagem de programação de computadores. Um dos problemas indicado pela literatura é o alto índice de evasão nos cursos da área de exatas, entre eles, o curso de Computação.

Uma das causas da alta taxa de evasão nas primeiras fases dos cursos de Computação, cerca de 28%, se deve ao fato do grau de dificuldade sentido pelos alunos nas disciplinas relacionadas à algoritmos ou linguagem de programação [Cabral et al. 2007]. Uma das razões para tal, é a grande quantidade de prática exigida para aprender a programar, a necessidade de uma elevação dos alunos engajados em atividades em sala de aula e em atividades extraclasse [Gomes and Mendes 2007]. Ao mesmo tempo, a habilidade de programação requer habilidades cognitivas de nível superior, tais como: resolução de problemas, pensamento lógico-matemático e do pensamento crítico [Fang 2012, Korkmaz 2012, Lau and Yuen 2009, Wang et al. 2012].

Neste sentido, Jogos têm sido utilizados como estratégias de motivação em ensino introdutório de programação, permitindo interatividade, experiências e criatividade [Leutenegger and Edgington 2007]. Assim, este artigo tem como objetivo propor um Método de Ensino de Programação Mediado por Simulação, utilizando uma ferramenta de gamificação como recurso mediático no processo de ensino-aprendizagem. Os resultados demonstram que há diferenças significativas entre o método de ensino de programação proposto e o método tradicional. Verificou-se melhores desempenhos entre os alunos que utilizaram o método mediado por simulação quando comparado aos alunos que utilizaram o método tradicional.

Este artigo está organizado da seguinte forma: na Seção 2 são apresentados os trabalhos relacionados, na Seção 3 são demonstrados os materiais e métodos utilizados nesta pesquisa, na Seção 4 são apresentados os resultados e as discussões e na Seção 5 a conclusão.

## **2. Trabalhos Relacionados**

Nos últimos tempos houve grande aumento no interesse científico em solucionar problemas no processo de ensino e aprendizagem nas disciplinas de programação. Tal interesse auxilia na redução da evasão escolar nos cursos de computação. Neste sentido a aprendizagem é uma mudança duradoura das estruturas cognitivas do indivíduo. A informação verbal, visual, auditiva e motora é armazenada na memória sob a forma de redes não lineares [El-Zakhem 2016]. Assim, a aprendizagem pode ser o resultado das transformações obtidas na atividade individual como busca de informações e análises.

Desta forma, surge a necessidade de novos processos que permitem auxiliar na aprendizagem de programação. Para [Tuparov et al. 2012], mensurar as experiências no desenvolvimento e uso de objetos interativos baseados em simulação de aprendizagem (OAs) pode ser uma solução para cursos introdutórios de programação. [Tuparov et al. 2012], conclui que o estudo piloto de OAs ajuda os alunos a compreenderem conteúdos de aprendizagem de forma mais fácil, mas muitas vezes os alunos em cursos de graduação, nas disciplinas de programação, apresentam alguns problemas com a transição do concreto para o pensamento abstrato, e a compreensão dos conceitos básicos de algoritmos e programação. Muitos educadores argumentam que o ensino e aprendizagem de programação para iniciantes é um processo educacional complexo e

com expressivo grau de dificuldade a ser aplicado.

Outros trabalhos propõem a criação de OAs baseados em simulação em cursos introdutórios de programação, focando em um dos temas importantes nesses cursos – a implementação de algoritmos de ordenação. Dessa forma, os resultados demonstram um aumento de interesse dos alunos e um nível de compreensão do conteúdo de aprendizagem [Tuparov et al. 2014].

Para [Rubio-Sánchez et al. 2014], avaliar o comportamento dos alunos em ferramentas que possibilitam as correções de códigos na programação pode identificar por meio das experiências dos alunos, o nível de compreensão e entendimento dos conceitos sobre programação. Essa ação pode determinar a utilização de novos processos em sala de aula. Assim, as análises das opiniões dos alunos, possibilitam medir outros recursos que auxiliam para a redução das taxas de abandono nos cursos de programação. O estudo desse autor foi realizado em uma disciplina de Análise e Projeto de Algoritmos, onde foram coletadas as informações por meio de vários questionários e dados gerados pela a ferramenta durante os cursos.

A Realidade Virtual é outro método de simulação que pode ser aplicado para auxiliar no processo de ensino e aprendizagem nas disciplinas de programação/algoritmo [Chandramouli et al. 2014]. Segundo [Chandramouli et al. 2014], utilizando as técnicas empregadas na Realidade virtual é possível baixar a carga cognitiva dos alunos no processo de ensino-aprendizagem nas disciplinas de programação. No entanto, a grande maioria das abordagens pedagógicas para a programação são predominantemente focadas no aprendizado tradicional por meio de livros, testes, trabalhos práticos e projetos.

Portanto, torna-se imperativo empregar técnicas alternativas, por exemplo, ferramentas de simulação por meio da ‘gamificação’ para auxiliar no processo de ensino e aprendizado, possibilitando assim, reduzir a taxa de desistência e evasão nos cursos de computação.

### **3. Materiais e Métodos**

Nesta seção, são apresentados os materiais e métodos utilizados para comparação entre os métodos tradicionais de ensino de programação e os métodos por simulação. Esta pesquisa caracteriza-se como um estudo de caso, adotando uma abordagem exploratória com indicadores quantitativos e qualitativos. Dessa forma, na subseção 3.1 será indicada a amostra da pesquisa. Na subseção 3.2 será proposto um método de ensino de programação mediado por simulação, o qual é dividido em três fases: teoria, ferramentas de simulações e avaliação/verificação de aprendizado.

#### **3.1. Amostra da Pesquisa**

A pesquisa foi realizada em uma turma de primeiro ano do ensino médio integrado do Curso Técnico em Informática em uma instituição federal de ensino.

Segundo [Gil 2005], durante a realização da pesquisa é importante a definição da amostra que será alvo do objeto de pesquisa, isto é, o conjunto de participantes que serão utilizados em uma pesquisa exploratória. Dessa forma, esta pesquisa contou com a participação de 28 alunos. Vale ressaltar que, nenhum aluno tinha experiência anterior com a linguagem de programação ou com princípios de programação, isto é, os participantes eram considerados iniciantes em programação/algoritmos.

### 3.2. Método de Ensino de Programação Mediado por Simulação

O método proposto por esta abordagem possui três fases: teoria, ferramentas de simulações e avaliação/verificação de aprendizagem. A Figura 1 apresenta as três fases que compõem o método proposto nesta pesquisa.

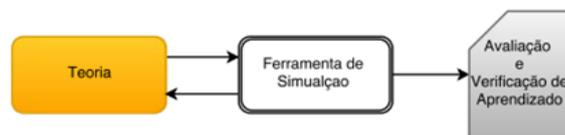


Figura 1. Método de Programação Mediado por Simulação

Observa-se que as duas primeiras fases podem ser realizadas de forma iterativas, isto é, há uma correlação entre as etapas. Por exemplo, um participante pode retornar à fase anterior – teoria – para revisar os conceitos estudados e depois voltar para a fase de simulação. Nota-se que a ferramenta de simulação é introduzida a partir da segunda fase deste método, ou seja, substitui a aplicação de exercícios práticos ou exemplos como são aplicados nos métodos de ensino tradicional de programação/algoritmo.

#### 3.2.1. Fase 1 – Teoria

Essa fase do método proposto caracteriza-se pelos instrumentos e materiais aplicados para o ensino de programação. Neste sentido, é possível utilizar apostilas, livros ou tutoriais, o qual possui os conceitos sobre algoritmos/programação.

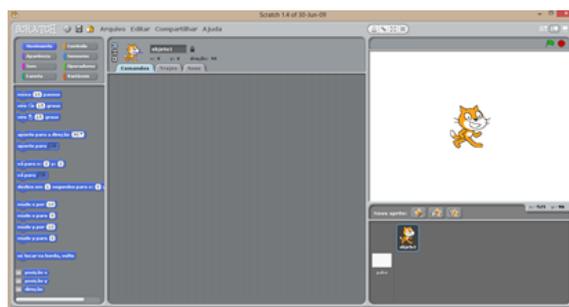
Dessa forma, objetivando comparar o método tradicional com o método por simulação, foram ensinados para os dois grupos de participantes, os que utilizariam o método por simulação e os que utilizariam o método tradicional, conceitos sobre comando de decisão (“se-então” e “se-então-senão”) e de repetição (“repita-até” e “enquanto-faça”) por meio de aulas expositivas baseadas em livros e apostilas.

#### 3.2.2. Fase 2 – Ferramenta de Simulação

Nesta fase, é introduzida uma ferramenta de simulação como recurso midiático no processo de ensino-aprendizagem em programação/algoritmo. Assim, para realização deste estudo de caso, foi escolhida a ferramenta de simulação *Scratch*. A Figura 2 demonstra a interface gráfica da ferramenta de simulação acima mencionada.

Essa ferramenta utiliza-se de blocos lógicos com itens de sons e imagens por meio de histórias interativas, valendo-se de conceitos de ‘gamificação’. O conceito mais comum pode resumir gamificação (ou *gamification*) como uso de mecânicas de games em outros contextos. Uma das grandes características na utilização desta ferramenta é o aprendizado colaborativo, por meio do compartilhamento do conhecimento os programadores são capazes de desenvolver habilidades criativas essenciais para a resolução de problemas de programação.

O *Scratch* foi desenvolvido pelo grupo *Lifelong Kindergarten* no MediaLab do MIT (Instituto de Tecnologia de Massachusetts). A ferramenta foi desenvolvida em 2003



**Figura 2. Ferramenta de Simulação Scratch**

e especialmente desenvolvida para estudantes entre 8 a 16 anos de idade, mas é usado por pessoas de todas as faixas etárias [Resnick et al. 2009]. O *Scratch* é usado em mais de 150 países, está disponível em mais de 40 idiomas e é fornecido gratuitamente para os principais sistemas operacionais, Linux, Windows e Mac OS.

### 3.2.3. Fase 3 – Avaliação e Verificação

Nesta fase, são realizadas as avaliações do conteúdo exposto na fase 1 (Teoria). Dessa forma, para verificação da eficácia do método de ensino de programação mediado por simulação, dividiu-se a turma de modo aleatório, após esta etapa, foi proposta a realização de uma questão dissertativa, com base nos conceitos de repetição e decisão aplicados em aula estudados pelos participantes. A Tabela 1 apresenta a questão proposta e os conceitos que foram abarcados na fase 1 – teoria.

**Tabela 1. Proposta de questão relacionando aos conceitos estudados na fase 1**

Questão	Fase 1
<p>Faça um algoritmo em linguagem natural que verifique se a tabela possui a configuração de um quadrado mágico. Uma tabela (matriz) é um Quadrado Mágico se:</p> <ul style="list-style-type: none"><li>• Ela for quadrada (mesma quantidade de linhas e colunas);</li><li>• E também se a soma dos elementos de cada linha, a soma dos elementos de cada coluna e a soma dos elementos das diagonais principais e secundárias são todas iguais.</li></ul>	<p>Conceitos de decisão:</p> <ul style="list-style-type: none"><li>• Se-Então;</li><li>• Senão-Então.</li></ul> <p>Conceitos de repetição:</p> <ul style="list-style-type: none"><li>• Enquanto-Faça;</li><li>• Repita-Até.</li></ul>

Após a aplicação da questão apresentada na Tabela 1, foram coletadas as notas dos participantes entre ambos os grupos, dos que utilizaram o método tradicional e daqueles que utilizaram o Método por Simulação. Para realização das análises e da comparação entre os dois métodos, foi aplicado um teste estatístico com intuito de identificar a diferença significativa entre os dois métodos. Neste sentido, o teste T demonstrou ser mais eficaz para tal comparação. O teste T é um teste de hipótese, e tem por objetivo

testar se existe diferença entre a média de uma amostra (aleatória) e a média populacional [Kazmier 1982].

#### 4. Resultados e Discussões

Nesta seção serão apresentados os resultados e discussões sobre o Método de Ensino de Programação Mediado por Simulação (MPMS). Para tal, na subseção 4.1 será apresentada a análise da experiência e das expectativas dos participantes desta pesquisa. Na subseção 4.2 serão realizadas as comparações entre o Método Tradicional e o MPMS e na subseção 4.3 serão demonstradas as diferenças significativas entre os dois métodos.

##### 4.1. Análise da Experiência e das Expectativas dos Participantes da Pesquisa

Esta pesquisa contou com a participação de 28 sujeitos, sendo que em 50,0% (cinquenta por cento) dos participantes foi aplicado o método tradicional e nos outros 50,0% (cinquenta por cento) foi aplicado o Método de Ensino de Programação Mediado por Simulação (MPMS). Deste total, 75% (setenta e cinco por cento) eram do sexo masculino e 25% (vinte e cinco por cento) do sexo feminino. A idade média entre os participantes era de 16 anos, com desvio padrão de 1,2 (um vírgula dois). A Tabela 3 apresenta as comparações entre os dois métodos analisados nesta pesquisa.

**Tabela 2. Experiência e expectativas dos participantes da pesquisa**

Questões	Alternativa	Tradicional		MPMS	
Sexo	Masc.	11	78,6%	10	71,4%
	Fem.	3	21,4%	4	28,6%
Ao iniciar o curso Técnico de Informática você teve alguma expectativa com relação às aulas de programação?	Sim	12	85,7%	11	78,6%
	Não	2	14,3%	3	21,4%
Como você considera o aprendizado em programação adquirido ao longo do curso?	Abaixo das expectativas	5	35,7%	3	21,4%
	Cumpra as expectativas	7	50,0%	9	64,3%
	Supera as expectativas	2	14,3%	2	14,3%
Você já conhecia os conceitos de algoritmo/programação antes de iniciar o curso técnico?	Sim	1	7,1%	0	0%
	Pouco	4	28,6%	2	14,3%
	Não	9	64,3%	12	85,7%

Em relação ao conhecimento dos conceitos de algoritmo e/ou programação antes de iniciar o curso técnico, foi constatado que cerca de 36% (trinta e seis por cento) dos sujeitos que utilizaram o método tradicional possuíam pouco conhecimento sobre algoritmo e/ou programação e aproximadamente 64% (sessenta e quatro por cento) relataram que não havia nenhum conhecimento. Em relação ao MPMS, cerca de 14% (quatorze por

cento) dos sujeitos da pesquisa relataram que tinham pouco conhecimento sobre algoritmo e/ou programação.

Observa-se, por meio da análise dos resultados, em relação ao aprendizado em programação que cerca de 35% (trinta e cinco por cento) dos sujeitos que utilizaram o método tradicional de ensino de programação relataram que o aprendizado por meio deste método estava abaixo das suas expectativas. Neste sentido, 50% (cinquenta por cento) dos sujeitos relataram que o aprendizado por meio deste método cumpria com as suas expectativas e aproximadamente 15% (quinze por cento) relataram que o método tradicional superou as suas expectativas.

Em relação ao aprendizado por meio do MPMS, 21% (vinte e um por cento) dos participantes relataram que foi abaixo das suas expectativas. Aproximadamente 65% (sessenta e cinco por cento) dos participantes relataram que o aprendizado por meio do método cumpria com as suas expectativas e 14% (quatorze por cento) relataram que o método superava as suas expectativas.

Em relação às principais dificuldades encontradas pelos participantes desta pesquisa nas disciplinas de algoritmo e/ou programação, aproximadamente 65% (sessenta e cinco por cento) dos sujeitos que utilizaram o Método Tradicional relataram que tinham maiores dificuldades em relação às estratégias de ensino/didática do processo. Cerca de 21% (vinte e um por cento) dos sujeitos, relataram que as dificuldades eram ocasionadas pela falta de atenção ou esforço do próprio estudante e 14% (quatorze por cento) relataram que o conteúdo era uma das maiores barreiras enfrentadas.

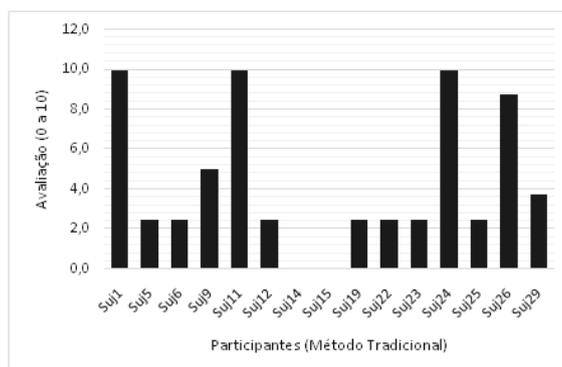
Neste sentido, percebe-se a ineficiência dos métodos tradicionais em comparação ao MPMS. Dessa forma, analisando o método proposto por esta investigação sob a perspectiva da satisfação dos usuários, aproximadamente 43% (quarenta e três por cento) dos participantes indicaram que o MPMS obteve um satisfatório resultado. 50% (cinquenta por cento) relataram que os resultados por meio da utilização deste método foram regulares e cerca de apenas 7% (sete por cento) relataram que os resultados obtidos por este método, foram insatisfatórios.

Em relação à satisfação dos usuários no método tradicional, aproximadamente 74% (setenta e quatro por cento) dos participantes relataram que os resultados obtidos por este método estavam abaixo das suas expectativas, isto é, regular ou ruim. Dessa forma, cerca de 93% (noventa e três por cento) dos sujeitos que utilizavam este método indicaram a necessidade da utilização de outros métodos no ensino de algoritmo e/ou programação.

#### **4.2. Comparações entre o Método Tradicional e MPMS**

A partir da verificação dos dois métodos, por meio da aplicação da preposição de um problema - avaliação (subseção 3.2.3), analisou-se o rendimento entre os dois grupos dos sujeitos desta pesquisa. A Figura 3 apresenta os resultados correlacionados entre os participantes que utilizaram o método tradicional e as suas respectivas notas mensuradas por meio da avaliação proposta.

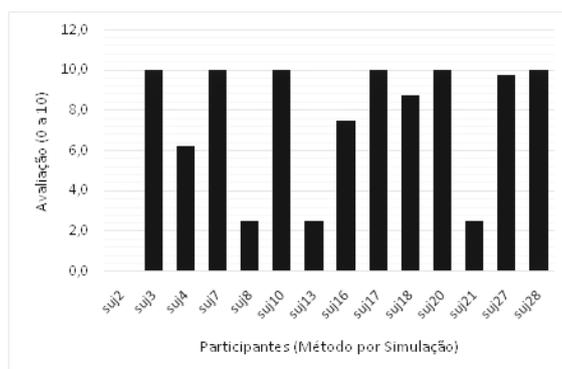
A partir da Figura 3, concluímos que no método tradicional, 20% (vinte por cento) dos sujeitos obtiveram nota máxima (10). Neste sentido cerca de 7% (sete por cento) tiraram notas inferiores a 10 (dez) e superiores a 6 (seis), e aproximadamente 73% (setenta



**Figura 3. Correlação entre notas e métodos - Tradicional**

e três por cento) dos sujeitos tiraram nota inferior a 6 (seis). Houve um percentual de aproximadamente 13% (treze por cento) de sujeitos que zeraram a avaliação e aproximadamente 53% (cinquenta e três por cento) dos participantes tiraram nota inferior a 4 (quatro). Notou-se que cerca de 36% (trinta e seis por cento) dos participantes já conheciam ou ouviram falar dos conceitos básicos de algoritmo/programação, e aproximadamente 64% (sessenta e quatro por cento) não conheciam tais conceitos.

Dessa forma, analisou-se também, as notas mensuradas por meio da avaliação proposta pelo MPMS. A Figura 4 apresenta os resultados correlacionados entre os participantes que utilizaram o MPMS e as suas respectivas notas.



**Figura 4. Correlação entre notas e métodos - MPMS**

Na Figura 4, sobre a avaliação do MPMS, observamos que, cerca de 43% (quarenta e três por cento) dos estudantes obtiveram nota máxima (10). No entanto, aproximadamente 29% (vinte e nove por cento) tiraram notas inferiores a 10 (dez) e superiores a 6 (seis), e aproximadamente 29% (vinte e nove por cento) tiraram nota inferior a 6 (seis). Houve um percentual de cerca 7% (sete por cento) de sujeitos que zeraram a avaliação e aproximadamente 21% (vinte e um por cento) dos participantes obtiveram nota inferior a 4 (quatro).

No entanto, notou-se que aproximadamente 14% (quatorze por cento) dos participantes, já conheciam os conceitos básicos de algoritmo e/ou programação, e aproximadamente 86% (oitenta e seis por cento) não conheciam os conceitos básicos de programação. Em relação ao MPMS, cerca de 93% (noventa e três por cento) dos participantes disse-

ram que a metodologia ajudou em seu aprendizado de introdução à programação e aproximadamente 7% (sete por cento) afirmaram que a metodologia não foi eficaz em seu aprendizado.

#### 4.3. Diferenças Significativas entre o Método Tradicional e MPMS

Após a coleta dos dados referentes às notas dos participantes desta pesquisa, foram analisadas e comparadas, por meio da aplicação de testes estatísticos, a ocorrência de diferenças significativas entre o método tradicional e MPMS. Assim, a Tabela 3 apresenta a comparação dos resultados entre o método tradicional e o MPMS.

**Tabela 3. Proposta de questão relacionando aos conceitos estudados na fase 1**

Comparação entre o Método tradicional e MPMS			
Método	Média	Desvio Padrão	Valor-p
Tradicional	4,3	3,6	0,05
MPMS	7,1	3,7	

Observa-se na Tabela 3, de acordo com o teste T, que o valor-p foi de 0,05, isto é, foi encontrada diferença significativa entre o método tradicional e o MPMS. Assim, analisando e comparando a média entre os dois métodos, considerando a amostra desta pesquisa, conclui-se que o método por simulação obteve melhores resultados quando comparado ao método tradicional.

## 5. Conclusões

Este artigo propôs à utilização do Método de Ensino de Programação Mediado por Simulação como recurso mediático no processo de ensino-aprendizagem. Por meio das análises, considerando a amostra desta pesquisa, conclui-se que há diferenças significativas entre o método proposto e o método tradicional.

Dessa forma, o Método de Ensino de Programação Mediado por Simulação obteve melhores resultados quando comparados aos métodos tradicionais. Verificou-se também, que os participantes que utilizaram o método proposto nesta pesquisa obtiveram um desempenho melhor quando comparado com os participantes que utilizaram o método tradicional.

Portanto, por meio dos resultados, percebeu-se que o Método de Ensino de Programação Mediado por Simulação auxilia no processo de aprendizado de introdução à programação para alunos iniciantes, podendo estes resultados nortear pesquisas futuras com intuito de solucionar problemas de evasão nos cursos de Computação.

## Referências

- Barnes, T., Powell, E., Chaffin, A., and Lipford, H. (2008). Game2learn: improving the motivation of cs1 students. In *Proceedings of the 3rd international conference on Game development in computer science education*, pages 1–5. ACM.
- Cabral, M. et al. (2007). Perfil dos cursos de computação e informática no brasil. In *XXVII Congresso da SBC-XV WEI, Rio de Janeiro*.

- Chandramouli, M., Zahraee, M., and Winer, C. (2014). A fun-learning approach to programming: An adaptive virtual reality (vr) platform to teach programming to engineering students. In *Electro/Information Technology (EIT), 2014 IEEE International Conference on*, pages 581–586. IEEE.
- El-Zakhem, I. H. (2016). Socratic programming: An innovative programming learning method. *International Journal of Information and Education Technology*, 6(3):247.
- Fang, X. (2012). Application of the participatory method to the computer fundamentals course. In *Affective Computing and Intelligent Interaction*, pages 185–189. Springer.
- Gil, A. C. (2005). Métodos e técnicas de pesquisa social. são paulo: Atlas, 1999. *Como elaborar projetos de pesquisa*, 4.
- Gomes, A. and Mendes, A. J. (2007). Learning to program-difficulties and solutions. In *International Conference on Engineering Education–ICEE*, volume 2007.
- Kazmier, L. J. (1982). *Estatística aplicada à economia e administração*. McGraw-Hill.
- Korkmaz, Ö. (2012). The impact of critical thinking and logico-mathematical intelligence on algorithmic design skills. *Journal of Educational Computing Research*, 46(2):173–193.
- Lau, W. W. and Yuen, A. H. (2009). Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy. *British Journal of Educational Technology*, 40(4):696–712.
- Leutenegger, S. and Edgington, J. (2007). A games first approach to teaching introductory programming. In *ACM SIGCSE Bulletin*, volume 39, pages 115–118. ACM.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., et al. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11):60–67.
- Rubio-Sánchez, M., Kinnunen, P., Pareja-Flores, C., and Velázquez-Iturbide, Á. (2014). Student perception and usage of an automated programming assessment tool. *Computers in Human Behavior*, 31:453–460.
- Tuparov, G., Tuparova, D., and Jordanov, V. (2014). Teaching sorting and searching algorithms through simulation-based learning objects in an introductory programming course. *Procedia-Social and Behavioral Sciences*, 116:2962–2966.
- Tuparov, G., Tuparova, D., and Tsarnakova, A. (2012). Using interactive simulation-based learning objects in introductory course of programming. *Procedia-Social and Behavioral Sciences*, 46:2276–2280.
- Wang, Y., Li, H., Feng, Y., Jiang, Y., and Liu, Y. (2012). Assessment of programming language learning based on peer code review model: Implementation and experience report. *Computers & Education*, 59(2):412–422.