COMBATE À EVASÃO E À REPROVAÇÃO
EM CURSOS DE GRADUAÇÃO EM CIÊNCIA
DA COMPUTAÇÃO POR MEIO DE
TECNOLOGIA EDUCACIONAL PARA
INICIANTES NA ÁREA DE PROGRAMAÇÃO.

Nathann Zini dos Reis

## Hipótese

O uso de uma aplicação mobile para o ensino de programação básica para novos alunos de universidade federal diminuirá a ocorrência de evasão e de retenção nos períodos iniciais dos cursos de computação.

### Motivação

O ensino superior brasileiro vem passando por mudanças consideráveis. O que antigamente era restrito a poucos, se tornou abrangente e vem fazendo parte da história de milhões de brasileiros: a oportunidade de realização de um curso superior e a consequente e tão sonhada graduação.

Segundo dados do Instituto Nacional de Estudos e Pesquisas, INEP (2015), o número de matrículas na educação superior passou de 6.4007.733, em 2010, para 8.033.574 em 2015. Naturalmente, como um reflexo da educação brasileira, muitos alunos ingressantes dos cursos de graduação em Ciência da Computação, por sua vez, não têm conhecimento suficiente para saber, de fato, sobre o que realmente é o curso ou sem sequer ter tido, em muitos casos, contato prévio com programação.

## Problema de Pesquisa

Para analisar o contexto descrito, de forma a medir e avaliar o desempenho dos alunos que cursam programação, serão avaliados os histórico de cada um dos alunos, as características dos grupos dos alunos que desistem do curso e que são reprovados nas matérias introdutórias e os métodos atuais de ensino de programação básica na graduação no desempenho final desses alunos.

Aumento de alunos na educação superior de 6.4007.733 alunos, em 2010, para 8.033.574 em 2015. Como reflexo, a evasão de alunos é um problema recorrente. O caso é ainda mais grave nos cursos da área de Computação. No Brasil, setenta e três por cento (73%) dos alunos de bacharelado em Ciência da Computação desistem da faculdade precocemente.

Relação verba/alunos ativos

Analisar as principais características entre os grupos de alunos que concluíram o curso e os que evadiram. Há, entre outras, uma tendência de alunos que não tinham experiências prévias na área (Computação) ao ingressar no curso (cerca de 70%) dentre os alunos que evadiram.

Então, uma maneira de tentar combater essa taxa de retenção é analisando a eficácia dos métodos atuais de ensino de programação, entre outros fatores. Em geral, os alunos têm dificuldades em entender determinados conceitos de programação, tais como ponteiros, recursão, declaração de variáveis, dentre outros. No entanto, alguns autores ressaltam que muitos alunos entendem os conceitos de programação, mas têm dificuldades em aplicá-los durante a construção de programas. Ainda, a atividade de programação não está limitada apenas à construção de programas, sendo necessário o desenvolvimento de habilidades de compreensão e análise de programas, as quais são pouco exploradas nas disciplinas de programação.

Os principais problemas no ensino e na aprendizagem de programação investigados atualmente são: (1) as dificuldades dos alunos em aprender os conceitos de programação, (2) a dificuldade dos alunos na aplicação desses conceitos durante a construção de programas e (3) a falta de motivação entre os alunos na realização da atividade de programação. Para amenizar tais problemas e dificuldades, é interessante a utilização de visualização de programas e algoritmos, de serious games e de desenvolvimento de ambientes pedagógicos para o ensino e aprendizagem de programação.

Os principais problemas no ensino e na aprendizagem de programação investigados atualmente são: (1) as dificuldades dos alunos em aprender os conceitos de programação, (2) a dificuldade dos alunos na aplicação desses conceitos durante a construção de programas e (3) a falta de motivação entre os alunos na realização da atividade de programação. Para amenizar tais problemas e dificuldades, é interessante a utilização de visualização de programas e algoritmos, de serious games e de desenvolvimento de ambientes pedagógicos para o ensino e aprendizagem de programação.

Entre as principais tecnologias analisadas, destacam-se utilização de gamificação/ambiente lúdico, o uso de robótica (linguagens específicas e equipamentos), a mediação online, a interdisciplinaridade e abordagens tradicionais de ensino.

Destacaram-se as estratégias de aprendizagem motivacionais baseadas em robótica e baseadas em PBL; bem como, o uso de linguagens de codificação específicas para aprendizagem de lógica de programação, como: Scratch.

Ambientes de aprendizagem online destacaram-se como um dos principais meios utilizados nas IES. Jogos digitais, também tem sido uma prática bastante comum nas IES para o ensino de programação introdutória. Destacaram-se os jogos digitais: GameLogic, Defense of the Ancients 2, EGameFlow e Teddy Race.

Os Ambientes de Desenvolvimento Integrado (IDEs) também foram empregados, destacando-se: Portugol Studio e o Game Maker, ambas contendo configurações adicionais voltadas para auxiliar os alunos iniciantes em suas primeiras experiências com lógica de programação.

Dentre as ferramentas encontradas, a maior parte foi desenvolvida para plataforma desktop ou web e, embora o acesso à dispositivos móveis seja abundante atualmente, poucas ferramentas foram desenvolvidas para ambiente mobile.

# Referências Bibliográficas

[1] HOED, R. M. (2016). Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de computação. Brasília, DF: Universidade de Brasília.

[2] MANHÃES, L. M. B., DA CRUZ, S. M. S., COSTA, R. J. M., ZAVALETA, J., and ZIMBRÃO,G. (2011). **Previsão de estudantes com risco de evasão utilizando técnicas de mineração de dados.** In Brazilian symposium on computers in education (simpósio brasileiro de informática na educação-sbie), volume 1

[3] MARQUES, L. T.; MARQUES, B. T.; SILVA, C. A. M.; ROCHA, R. S.; SILVA, J. C. P.; SILVA, L. C. e; QUEIROZ, P. G. G.; CASTRO, A. F. de. **A Evasão Escolar no Ensino Superior: Um Estudo de Caso do Curso de Ciência da Computação da UFERSA** / School Evasion in Higher Education: A Case Study of the Computer Science Course at UFERSA. Brazilian Journal of Development, [S. I.], v. 6, n. 12, p. 103334–103350, 2020. DOI: 10.34117/bjdv6n12-728. Disponível em: https://brazilianjournals.com/ojs/index.php/BRJD/article/view/22330. Acesso em: 15 sep. 2022.

[4] J. Helminen and L. Malmi. Jype - a program visualization and programming exercise tool for python. In Proceedings of the ACM Conference on Computer and Communications Security, pages 153–162, 2010

[5] O. Meerbaum-Salant, M. Armoni, and M. BenAri. **Learning computer science concepts with scratch**. In ICER'10 - Proceedings of the International Computing Education Research Workshop, pages 69–76, 2010.

## Referências Bibliográficas

[6] M Piteira and C Costa. **Learning computer programming: Study of difficulties in learning programming.** In ACM International Conference Proceeding Series, pages 75–80, 2013.

[7] L. L Hu, S. S Tseng, and T. J Lee. Towards scaffolding problem-solving implementation process in undergraduate programming course. In Proceedings - 2013 IEEE 13th International Conference on Advanced Learning Tec

[8] S. R b Jantan and S. A Aljunid. **An experimental evaluation of scaffolded educational games design for programming**. In 2012 IEEE Conference on Open Systems, ICOS 2012, 2012.

[9] P. Panwong and K. Kemavuthanon. **Problembased learning framework for junior software developer: Empirical study for computer programming students**. Wireless Personal Communications, 76(3):603–613, 2014.

[10] S. Kollmansberger. **Helping students build a mental model of computation**. In ITiCSE'10 - Proceedings of the 2010 ACM SIGCSE Annual Conference on Innovation and Technology in Comp

[11] F. Mei. The application of trail-and-error learning in c language curriculum for non-majors in computer. In Proceedings of the 8th International Conference on Computer Science and Education, ICCSE 2013, pages 1239–1242, 2013.

## Referências Bibliográficas

[12] E. Vrachnos and A. Jimoyiannis. **Design and evaluation of a web-based dynamic algorithm visualization environment for novices**. In Procedia Computer Science, volume 27, pages 229–239, 2013.

[13] Anthony Robins, Janet Rountree, and Nathan Rountree. **Learning and teaching programming: A review and discussion**. Computer Science Education, 13(2):137–172, 2003.

[14] A. Vihavainen, J. Airaksinen and C. Watson. A Systematic Review of Approaches for Teaching Introductory Programming and Their Influence on Success. In Proceeding of The International Computing Education Research Conference, ICER, pages 19-26, 2014.

[15] SOUZA, Draylson Micael; BATISTA, Marisa Helena da Silva; BARBOSA, Ellen Francine. **Problemas e Dificuldades no Ensino de Programação: Um Mapeamento Sistemático**. Revista Brasileira de Informática na Educação, [S.I.], p. 39, abr. 2016. ISSN 2317-6121. Disponível em: <a href="http://ojs.sector3.com.br/index.php/rbie/article/view/3317">http://ojs.sector3.com.br/index.php/rbie/article/view/3317</a>>. Acesso em: 15 set. 2022. doi:http://dx.doi.org/10.5753/rbie.2016.24.1.39.

[16] DUARTE DE HOLANDA, W.; DE PAIVA FREIRE, L.; CÁSSIA DA SILVA COUTINHO, J. **Estratégias de ensino-aprendizagem de programação introdutória no ensino superior: uma Revisão Sistemática da Literatura**. RENOTE, Porto Alegre, v. 17, n. 1, p. 527–536, 2019. DOI: 10.22456/1679-1916.95905. Disponível em: https://seer.ufrgs.br/index.php/renote/article/view/95905. Acesso em: 29 set. 2022.