

Avaliação da Influência dos Tamanhos das Turmas de Programação no Desempenho dos Estudantes de Computação

Leonardo Silva

¹Instituto Federal de Pernambuco
Garanhuns – PE – Brazil

leonardo.silva@garanhuns.ifpe.edu.br

Abstract. *The teaching and learning process of programming education is complex and impacted by numerous variables. Teachers and researchers argue that the high number of students in programming classes negatively influences learning. This topic was investigated in different international contexts, and a lack of data regarding the Brazilian context was found, which is expected to be mitigated with the present study. Therefore, this work aims to provide statistical evidence on this topic by analyzing the academic data of 206 students enrolled in a technical computing course. No statistically significant difference between students' performance was found, regardless of the number of students in the analyzed classes. The implications for the programming teaching and learning process are discussed.*

Resumo. *Sabe-se que o processo de ensino e aprendizagem de programação é complexo e impactado por inúmeras variáveis. Dentre elas, professores e pesquisadores argumentam que o elevado número de alunos em turmas de programação influencia negativamente o aprendizado. Esta temática foi objeto de investigação em diferentes contextos internacionais, e foi observada uma carência de dados no contexto brasileiro, o qual espera-se mitigar com o presente estudo. Portanto, este trabalho tem por objetivo fornecer evidências estatísticas sobre essa temática, o que pode subsidiar políticas educacionais na área de educação em computação. Os dados acadêmicos de 206 estudantes matriculados em um curso técnico de informática integrado ao ensino médio foram analisados. Para a amostra realizada, não foi observada diferença estatisticamente significativa entre os desempenhos dos estudantes de turmas com mais ou menos alunos. As implicações ao processo de ensino e aprendizagem de programação são discutidas.*

1. Introdução

O processo de ensino e aprendizagem de programação é conhecido por ser complexo e resultar em um número elevado de evasões e retenções [Silva 2018]. Diversos fatores podem impactar esse processo, sendo o quantitativo de estudantes nas turmas um dos fatores investigados [Sim and Lau 2018, Watson and Li 2014, Gomes and Mendes 2007, Slavin 1989]. Professores argumentam que em turmas com muitos alunos há uma maior dificuldade em realizar o acompanhamento individual do processo de aprendizagem, o que prejudica o feedback que podem oferecer, aspectos importantes ao sucesso educacional [Qian and Lehman 2017].

Há na literatura um corpo de evidências sobre como o quantitativo de alunos em turmas de programação influencia no desempenho acadêmico. O "senso comum" e os dados de estudos internacionais apontam que os estudantes em turmas menores alcançam melhores resultados acadêmicos [Watson and Li 2014]. No entanto, há a necessidade de prover mais evidências estatísticas sobre essa temática, em especial pela carência de dados sobre o contexto educacional brasileiro, pois sabe-se que fatores sócio-econômicos e culturais influenciam o aprendizado [Balan and Nunes 2014]. Esse corpo de conhecimento é importante para subsidiar a decisão dos gestores das instituições educacionais em definir políticas educacionais, como exemplo a redução do quantitativo de alunos por turma [Camargo 2012].

Propõe-se neste estudo prover evidências estatísticas sobre como o tamanho das turmas impacta no desempenho acadêmico de alunos de programação. A investigação proposta neste estudo orientou-se nas seguintes questões de pesquisa: A) *Qual a relação entre o tamanho das turmas de programação e o desempenho acadêmico dos estudantes?* e B) *Qual a relação entre o tamanho das turmas de programação e a desistência dos estudantes?* Para respondê-las, uma investigação quantitativa observacional ex-post-facto foi realizada. Os dados acadêmicos de 206 estudantes matriculados no componente curricular de introdução à programação da instituição omitida para fins de avaliação às cegas foram analisados. O componente faz parte da matriz curricular de um curso técnico integrado ao ensino médio.

Este trabalho está organizado da seguinte forma: na Seção 2 apresenta-se o estado da arte sobre a influência que o tamanho das turmas pode causar no aprendizado de programação, e compara-se os dados de estudos em diferentes contextos educacionais ao encontrado por nós. Na Seção 3 apresenta-se a metodologia de investigação utilizada, descrição do contexto educacional e amostragem e planejamento estatístico. Os resultados são apresentados na Seção 4 e sua análise na Seção 5. Por fim, as limitações são discutidas na Seção 6.

2. Revisão da Literatura

O impacto do tamanho das turmas no desempenho acadêmico dos estudantes já foi objeto de estudo por inúmeros trabalhos na área de educação, incluindo o ensino em programação. Bennedsen e Caspersen fizeram um levantamento sobre os índices de reprovação em cursos de introdução à programação de diferentes instituições de ensino em todo o mundo [Bennedsen and Caspersen 2007]. O estudo destaca-se como um dos primeiros a buscar compreender, dentre outros fatores, o impacto que o tamanho das turmas pode significar no aprendizado de programação. Foi observado que em turmas com menos de 30 alunos a aprovação dos estudantes foi consideravelmente maior (82% contra 69% em turmas com 30 ou mais estudantes). Em uma reedição do estudo anterior, Watson e Li coletaram e analisaram novos dados. Os resultados encontrados foram similares, reforçando que para o contexto analisado por eles as turmas menores (também com menos que 30 alunos) alcançaram melhores resultados no aprendizado de programação (80,1% contra 65,4%) [Watson and Li 2014].

Vihavainen et al. analisaram o impacto de diferentes intervenções no aprendizado de programação, e dentre as ações institucionais, a que melhor resultou em ganhos ao aprendizado foi a redução do tamanho das turmas [Vihavainen et al. 2014]. Os dados re-

latados neste estudo exemplificam uma ação de política educacional que pode ser adotada neste contexto de discussão acerca do impacto que o tamanho das turmas pode causar no aprendizado. No estudo realizado por Robins, que também destina-se à análise de diferentes fatores que impactam o aprendizado de programação, reporta um resultado similar ao de [Vihavainen et al. 2014], no qual turmas menores alcançaram menores índices de retenção [Robins 2010].

O conjunto de evidências apresentado reforça o argumento que o tamanho das turmas pode impactar no aprendizado de estudantes de programação. No entanto, o contexto dos estudos supradescritos envolveu estudantes da Europa e Estados Unidos, que possuem condições sociais significativamente diferentes que os estudantes brasileiros (e.g., maior acesso a computador em suas residências, conectividade com Internet, maior capacidade de se dedicar aos estudos, entre outros) [Balan and Nunes 2014], que podem impactar a generalização dos resultados para a nossa realidade. Por exemplo, Bosse e Gerosa analisaram diferentes fatores que influenciam a reprovação de estudantes de cursos superiores de computação de São Paulo. Seus resultados apontam que não há correlação entre o desempenho final (mensurado pela média final dos estudantes e índices de reprovação) com os tamanhos de turmas nos quais os estudantes assistiram às aulas [Bosse and Gerosa 2015]. Apesar do resultado similar ao deste trabalho, o formato de cálculo é diferente e utiliza correlação estatística para chegar à esta conclusão. Ainda sim, somados, os estudos proveem uma análise inicial que aponta para possíveis diferenças no contexto educacional brasileiro de ensino de computação, quando comparado aos resultados anteriores. Por fim, destaca-se que os trabalhos analisados nesta seção são de ensino de computação no nível superior, enquanto que trazemos para discussão os dados de estudantes matriculados em um curso técnico de computação integrado ao ensino médio, ampliando a discussão para esse nicho.

3. Metodologia

Este estudo caracteriza-se como uma investigação de natureza quantitativa e *ex-post-facto*. Portanto, foram utilizados os dados acadêmicos passados de estudantes que cursaram o componente curricular de introdução à programação, após a sua realização.

3.1. Descrição da Amostra e Contexto Pedagógico

A amostra deste estudo é composta por 206 estudantes matriculados em um curso técnico de computação integrado ao ensino médio de uma instituição omitida para fins de avaliação às cegas. Os estudantes apresentavam faixa etária entre 13 e 16 anos, e há uma distribuição equivalente entre os gêneros. Aproximadamente 60% dos estudantes recebem bolsas de estudo em razão de sua vulnerabilidade social e 30% deles não possui computador em casa, sendo o campus a única oportunidade para prática da programação. Em geral, a grande maioria destes alunos (aproximadamente 90%) não tiveram contato com programação previamente ao curso. Os dados coletados representam um período de análise de três anos de estudantes matriculados em seis turmas diferentes.

Neste curso é ofertada qualificação profissional para que os egressos atuem no segmento de desenvolvimento de software. A formação ocorre concomitantemente ao período em que cursam o ensino médio, de forma integrada ao seu currículo. Esta modalidade de ensino profissionalizante possui duração de 4 anos, na qual os estudantes

possuem componentes curriculares tradicionais do ensino médio, como matemática, português, física, dentre outros, mas também componentes de computação, como introdução à programação, análise orientada a objetos, programação para a Internet, testes de software, entre outros.

O componente curricular de introdução à programação está presente no primeiro ano de curso e dispõe de uma carga horária semanal de 2 horas, totalizando 80 horas ao término do ano letivo. O conteúdo curricular contempla: pensamento algorítmico, variáveis, tipos de dados, estruturas de controle de fluxo e repetição, vetores e funções, ministrados com a linguagem de programação Java. As aulas ocorreram em um laboratório de informática, no qual em alguns casos havia a disponibilidade de um computador por estudante e em outros era necessário que dois estudantes o compartilhassem. O professor deste componente curricular, em geral, utilizou uma metodologia de ensino expositiva, intercalada com atividades práticas de laboratório. Quatro avaliações foram realizadas ao longo do ano, sendo uma por bimestre e com direito a recuperação. As atividades avaliativas foram realizadas com papel e caneta e os alunos deveriam construir algoritmos para problemas que eram passados. Nos anos analisados, os índices de reprovação no componente curricular foi de aproximadamente 50%.

3.2. Variáveis Confundidoras

Sabe-se que o processo de ensino e aprendizagem de programação é influenciado por inúmeros fatores, como a metodologia do professor, a linguagem de programação utilizada, característica dos alunos (e.g., conhecimento prévio em programação, base matemática, conhecimento no idioma Inglês, entre outros) [Robins 2019]. Assim, pesquisas na área de educação em computação devem buscar minimizar a influência dos fatores não relevantes à investigação, o que pode ser feito por meio do controle de determinadas variáveis.

No contexto deste estudo foram definidos que apenas seriam considerados os dados acadêmicos dos estudantes lecionados pelo mesmo professor (que não participou deste estudo). Ainda que sua metodologia possa ter apresentado variações em razão das turmas, avalia-se que essa decisão mitiga um viés maior que seria analisar os dados de turmas de diferentes professores. Também foi definido que apenas seriam utilizados os dados das turmas que tenham sido lecionadas com a mesma linguagem de programação e possuíssem a mesma carga horária. Além disso, foram considerados apenas os dados dos estudantes que cursaram o componente curricular pela primeira vez, pois alunos repetentes trazem um histórico de aprendizagem que poderia influenciar a análise.

3.3. Planejamento estatístico

Dois cálculos estatísticos foram realizados para responder as questões de pesquisa definidas. O primeiro teve por objetivo verificar a diferença entre as médias finais dos estudantes nos dois tamanhos de turmas estabelecidos. Como primeira etapa de investigação foi analisado se os dados coletados seguiam uma distribuição normal, para assim determinar o tipo de teste estatístico que seria utilizado (paramétrico ou não paramétrico). Este procedimento foi realizado por meio do teste de Shapiro-Wilk, reconhecido para esta finalidade [Razali et al. 2011] e que reportou a inexistência de uma distribuição normal. Portanto, a considerar a informação acima, e o tipo de dado coletado, não pareado e com dois grupos,

utilizou-se o teste não paramétrico de Mann-Whitney [McKnight and Najab 2010]. Neste teste, realiza-se um procedimento de ranqueamento dos dados e compara-se as medianas.

O segundo cálculo verificou a associação entre o resultado final no componente curricular (aprovado ou reprovado) com os tamanhos de turmas. Neste, optou-se pelo uso do teste de Qui-Quadrado de Pearson, pois o resultado final e o tamanho das turmas são variáveis categóricas [McHugh 2013]. A base para o cálculo do Qui-Quadrado consiste na construção de uma tabela de contingência, em que as variáveis analisadas são dispostas em linhas e colunas, construindo assim associações, e após isto contabiliza-se a frequência das ocorrências de cada associação. A tabela de contingência deste estudo é apresentada na Tabela 1. Todos os cálculos estatísticos foram realizados utilizando a ferramenta R ¹ e os dados estatísticos estão disponíveis em: [ii](#) omitido para fins de revisão às cegas *ii*.

3.3.1. Ameaças à validade

Existem fatores que podem influenciar o resultado estatístico deste trabalho e que serão discutidos nesta seção. Sabe-se que o professor pode modificar a sua metodologia e/ou avaliações entre turmas, elementos que podem influenciar o aprendizado dos alunos. Considera-se também que os estudantes entre turmas diferentes podem apresentar conhecimentos significativamente diferentes, por exemplo, a concorrência para o curso técnico de informática tem aumentado nos últimos anos, e assim, pode resultar em estudantes que ingressam com uma base educacional melhor estabelecida. Como reportado por Silva, a nota do vestibular é uma variável que explica parte do desempenho do estudante neste componente curricular [Silva et al. 2021]. Elementos como este podem afetar o resultado final e são difíceis de serem controlados em estudos desta natureza.

4. Resultados

A amostra é composta por 206 estudantes matriculados em seis turmas diferentes de introdução à programação. Foram criados dois grupos para agrupar os estudantes, de acordo com os tamanhos das turmas, seguindo a proposição feita por estudos anteriores. No estudo conduzido por Bennedsen e Caspersen [Bennedsen and Caspersen 2007] e Vihavainen et al. [Vihavainen et al. 2014] os autores agruparam os estudantes em turmas com mais de 30 alunos e outras com 30 ou menos. No entanto, em razão dos tamanhos das turmas analisadas neste trabalho, não foi possível replicar essa característica, pois resultaria em grupos com quantitativos de estudantes desequilibrados. Assim, optou-se por agrupar os estudantes em turmas com 33 alunos ou menos (grupo 1) e com mais que 33 alunos (grupo 2), resultando três turmas para cada grupo. As turmas com 28, 25 e 33 alunos, foram classificadas no grupo 1 e apresentaram nota média 4,38 e desvio padrão de 2,75. As turmas com 34, 38 e 48 alunos foram agrupadas no grupo 2 e apresentaram média de 4,51 e desvio padrão de 2,59 (Figura 1). No cálculo foram considerados os dados de estudantes que desistiram no último semestre do curso, ou seja, poderiam realizar o exame final, mas não o fizeram, sendo estes classificados como reprovados.

O teste de Mann-Whitney foi realizado para responder a questão de pesquisa A, verificando a existência de diferenças estatisticamente significantes nas médias finais dos

¹<https://www.r-project.org/>

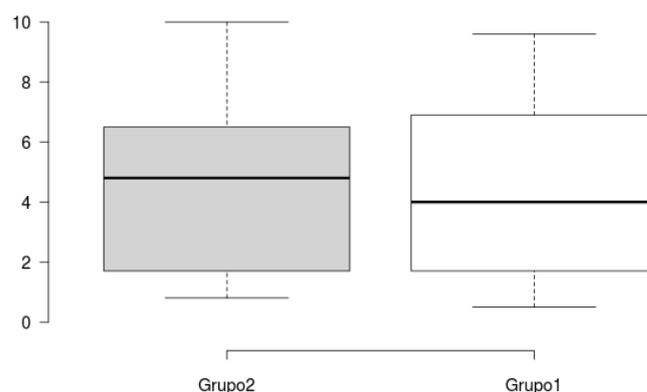


Figura 1. Boxplot das médias finais dos grupos.

estudantes. Com o resultado de $W = 5076,5$ e um p-value de 0,76, não é possível rejeitar a hipótese nula para um nível de confiança de 95%. Assim, não é possível afirmar que há diferença estatisticamente significativa entre as médias finais dos estudantes de turmas com tamanhos diferentes.

Para responder a questão de pesquisa B construiu-se uma tabela de contingência (Tabela 1) com a frequência de desistentes e não desistentes por grupo. Foram considerados desistentes os estudantes que poderiam realizar o exame final, mas não o fizeram. Com o cálculo do teste de Qui-Quadrado de Pearson obteve-se o resultado de $\chi^2 = 3,84$ com um p-value de 0,6. Assim, para um nível de confiança de 95% não é possível afirmar que o número de desistentes está associado à turma ao qual o estudante frequentou.

Tabela 1. Tabela de contingência com os dados de aprovações e reprovações agrupados.

	Quantitativo de desistentes	Quantitativo de não desistentes
33 alunos ou menos	28	58
Mais que 33 alunos	35	85

5. Análise dos Resultados

Respondendo as questões de pesquisa estabelecidas, para os dados analisados não é possível afirmar que o tamanho da turma teve influência nas médias finais dos estudantes ou em sua desistência do componente curricular de introdução à programação. O resultado difere de estudos internacionais [Bennedsen and Caspersen 2007, Robins 2010, Watson and Li 2014], mas corrobora os achados nacionais de [Bosse and Gerosa 2015]. Novas investigações se mostram necessárias para confirmar se o contexto educacional brasileiro possui características próprias que influenciam os resultados e que devem ser consideradas, antes de generalizar os resultados de estudos internacionais em nossa realidade.

Para analisar o resultado é preciso considerar as dinâmicas sociais presentes no

contexto do estudo realizado. Para contribuir com essa discussão o professor do componente curricular foi entrevistado, onde identificou-se que nas turmas com mais de 35 alunos fazia-se necessário agrupá-los em pares em razão do número limitado de computadores. Assim, destaca-se que essa interação social pode ter contribuído para minimizar os problemas das turmas maiores. Acredita-se que nestes casos o ensino colaborativo promoveu maior interação entre os alunos que compartilhavam dúvidas e soluções para os problemas que surgiam. Pela literatura acadêmica sabe-se que o aprendizado colaborativo é uma importante estratégia educacional que favorece o conhecimento de programação e motivação dos estudantes [Umaphy and Ritzhaupt 2017]. Apesar das nossas limitações em atribuir causalidade à este fator, ele deve ser considerado em discussões futuras, em especial pelas evidências científicas já publicadas sobre o tema [Silva et al. 2020].

Evidencia-se nessa discussão o importante papel do professor em buscar transformar suas estratégias pedagógicas para adaptar-se ao contexto educacional em que se encontra. Sugere-se também que outras ações podem ser realizadas para dar suporte aos estudantes nestas condições, na qual a instituição educacional também possui a sua contribuição, como por exemplo o uso de ferramentas automatizadas de avaliação [Dagostini et al. 2017], promover atividades de monitoria [Gonçalves and Schepke 2018], uso de tutores inteligentes [Martins et al. 2012], entre outros. Sugere-se como trabalho futuro que pesquisadores e professores avaliem o uso de diferentes estratégias para lidar com o elevado número de estudantes nas turmas, buscando assim subsidiar educadores que enfrentam este problema.

Ressalta-se ainda que a educação em programação é um assunto complexo e impactado por inúmeras variáveis para além do tamanho das turmas e trabalho colaborativo [Robins 2019]. Portanto, as conclusões apresentadas devem ser interpretadas à luz da literatura já publicada sobre o tema, considerando também outros fatores já conhecidos. Por exemplo, sabe-se que o aprendizado dos estudantes é impactado pelo seu conhecimento em componentes correlatos (e.g., matemática, resolução de problemas), suas habilidades cognitivas e metacognitivas, pelas estratégias pedagógicas dos professores, dentre outros [Souto and Tedesco 2017]; [de Holanda et al. 2019]. Investigações futuras podem investigar o impacto desses fatores, associados ao tamanho das turmas, no aprendizado.

Por fim, algumas questões permanecem em aberto e necessitam de maiores investigações em trabalhos futuros, como: i) quais são os fatores que contribuíram para um melhor desempenho dos estudantes em turmas maiores?. Para essa questão é fundamental prover uma visão holística do processo de ensino e aprendizagem, analisando as diversas variáveis presentes. O uso de entrevistas com professores e estudantes para avaliar suas percepções sobre o tema, pode ser uma estratégia a ser adotada neste caso. Uma outra pergunta apresentada é: ii) o acesso individualizado à computadores nos laboratórios é um fator determinante para o sucesso dos estudantes?. Sabe-se que a prática de programação é fundamental para o alcance de bons resultados durante o aprendizado [Scott et al. 2015], e limitar os estudantes nisto pode comprometer o aprendizado. No entanto, os resultados deste estudo sugerem que não basta apenas prover um computador por estudante, e que há outros fatores que podem influenciar ainda mais o aprendizado. Em razão da necessidade de maiores dados sobre isto, sugere-se investigar essa questão que perpassa por compreender como o processo colaborativo de aprendizagem influencia na educação em computação [Teague and Roe 2008]. Por fim, levanta-se a seguinte questão:

iii) quais os limites (mínimos e máximos) nos tamanhos das turmas podem favorecer o aprendizado de programação?. Como discutido, avalia-se que a interação entre estudantes parece contribuir com o aprendizado. Assim, reduzir o tamanho das turmas pode impactar neste processo. Por outro lado, aumentar o número de alunos em turmas de programação dificulta o papel do professor em prover *feedback* adequado [Gomes 2013]. Para avaliar essa questão mais estudos observacionais podem ser realizados, e para ampliar a capacidade de atribuir causalidade aos fatos, realizar estudos experimentais definindo o tamanho das turmas como uma variável de análise [Maxwell 2004]. Isto é em especial importante a considerar que os estudos descritos no estado da arte são todos observacionais.

6. Limitações do Estudo

Estudos observacionais são limitados quanto à capacidade de associar conclusões aos fatores observados [Gersten et al. 2000]. Neste sentido, ponderamos que os resultados apresentados não implicam necessariamente em uma relação de causalidade. Além disto, reforça-se que aprendizado de programação é influenciado por múltiplos fatores, além do tamanho das turmas [Medeiros et al. 2018], que também podem impactar no resultado. Apesar de algumas variáveis confundidoras terem sido controladas, como ter optado pela coleta de dados de turmas ministradas pelo mesmo professor, outros fatores certamente influenciam no aprendizado e devem ser ponderados.

Outro ponto a ser considerado é que o estudo foi realizado com dados dos estudantes de um curso de nível técnico integrado ao ensino médio, podendo existir diferenças para um nível educacional superior, ainda que os resultados sejam similares aos encontrados por [Bosse and Gerosa 2015] que avaliou os dados de universidades.

7. Conclusão

O elevado número de estudantes em turmas de programação é um fator que pode impactar o aprendizado de programação. O presente estudo forneceu evidências estatísticas sobre o desempenho acadêmico de estudantes em turmas de diferentes tamanhos. Os dados acadêmicos de 206 estudantes matriculados em um curso técnico de computação integrado ao ensino médio foram analisados. Observou-se que os estudantes em turmas maiores apresentaram desempenho acadêmico levemente superior. No entanto, os testes estatísticos não apontam significância estatística. O resultado aponta para uma possível influência de fatores desconhecidos, para além do tamanho da turma. Além disto, foram discutidas estratégias pedagógicas que podem ser adotadas nessas situações para mitigar o problema no elevado número de alunos. Sugere-se que estudos futuros repliquem a metodologia adotada neste trabalho em outros contextos educacionais para ampliar o corpo de conhecimento sobre o tema.

Referências

- Balan, D. S. L. and Nunes, K. D. (2014). O ensino superior tecnológico na formação profissional: caso comparativo entre brasil e itália. *Revista latinoamericana de educación comparada*, 5(6):80–95.
- Bennedsen, J. and Caspersen, M. E. (2007). Failure rates in introductory programming. *AcM SIGcSE Bulletin*, 39(2):32–36.

- Bosse, Y. and Gerosa, M. A. (2015). Reprovações e trancamentos nas disciplinas de introdução à programação da universidade de são paulo: Um estudo preliminar. In *XXIII WEI–Workshop sobre Educação em Computação*.
- Camargo, J. (2012). O efeito do tamanho da turma sobre o desempenho escolar: uma avaliação do impacto da "enturmação" no ensino fundamental do rio grande do sul.
- Dagostini, J., de Moura Lima, M. V., Bucior, L., Tonin, N., and Bez, J. L. (2017). Incentivando a aprendizagem de algoritmos através do uri online judge forum 2.0. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 28, page 1781.
- de Holanda, W. D., de Paiva Freire, L., and da Silva Coutinho, J. C. (2019). Estratégias de ensino-aprendizagem de programação introdutória no ensino superior: uma revisão sistemática da literatura. *RENOTE-Revista Novas Tecnologias na Educação*, 17(1):527–536.
- Gersten, R., Baker, S., and Lloyd, J. W. (2000). Designing high-quality research in special education: Group experimental design. *The Journal of Special Education*, 34(1):2–18.
- Gomes, A. and Mendes, A. J. (2007). Learning to program-difficulties and solutions. In *International Conference on Engineering Education–ICEE*, volume 7.
- Gomes, J. M. (2013). *A importância do suporte e do feedback do professor na regulação da aprendizagem*. PhD thesis, Instituto Superior de Psicologia Aplicada.
- Gonçalves, L. B. and Schepke, C. (2018). Monitoria: Análise do impacto da monitoria no ensino de disciplinas de programação. *Anais do Salão Internacional de Ensino, Pesquisa e Extensão*, 9(2).
- Martins, L. C., Lopes, D. A., and Raabe, A. (2012). Um assistente de predição de evasão aplicado a uma disciplina introdutória do curso de ciência da computação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 23.
- Maxwell, J. A. (2004). Causal explanation, qualitative research, and scientific inquiry in education. *Educational researcher*, 33(2):3–11.
- McHugh, M. L. (2013). The chi-square test of independence. *Biochemia medica: Biochemia medica*, 23(2):143–149.
- McKnight, P. E. and Najab, J. (2010). Mann-whitney u test. *The Corsini encyclopedia of psychology*, pages 1–1.
- Medeiros, R. P., Ramalho, G. L., and Falcão, T. P. (2018). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, (99):1–14.
- Qian, Y. and Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, 18(1):1–24.
- Razali, N. M., Wah, Y. B., et al. (2011). Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of statistical modeling and analytics*, 2(1):21–33.

- Robins, A. (2010). Learning edge momentum: A new account of outcomes in cs1. *Computer Science Education*, 20(1):37–71.
- Robins, A. V. (2019). Novice programmers and introductory programming. *The Cambridge Handbook of Computing Education Research, Cambridge Handbooks in Psychology*, pages 327–376.
- Scott, M. J., Counsell, S., Lauria, S., Swift, S., Tucker, A., Shepperd, M., and Ghinea, G. (2015). Enhancing practice and achievement in introductory programming with a robot olympics. *IEEE Transactions on Education*, 58(4):249–254.
- Silva, L., Mendes, A., and Gomes, A. (2020). Computer-supported collaborative learning in programming education: A systematic literature review. In *Proceedings of the IEEE Global Engineering Education Conference*.
- Silva, L. S. (2018). Análise do aprendizado em programação de estudantes do ensino técnico integrado do instituto federal de pernambuco. In *Anais do V Encontro Nacional de Computação dos Institutos Federais*. SBC.
- Silva, L. S., dos Santos Silva, J. G., and Santos, M. S. (2021). Investigando a relação entre a nota no vestibular com o desempenho em introdução à programação. In *Anais do Simpósio Brasileiro de Educação em Computação*, pages 66–71. SBC.
- Sim, T. Y. and Lau, S. L. (2018). Online tools to support novice programming: A systematic review. In *2018 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*, pages 91–96. IEEE.
- Slavin, R. E. (1989). Class size and student achievement: Small effects of small classes. *Educational Psychologist*, 24(1):99–110.
- Souto, M. and Tedesco, P. (2017). Uma revisão sistemática da literatura sobre conhecimentos, habilidades, atitudes e competências desejáveis para auxiliar a aprendizagem de programação. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 6, page 1162.
- Teague, D. M. and Roe, P. (2008). Collaborative learning: towards a solution for novice programmers.
- Umapathy, K. and Ritzhaupt, A. D. (2017). A meta-analysis of pair-programming in computer programming courses: Implications for educational practice. *ACM Transactions on Computing Education (TOCE)*, 17(4):16.
- Vihavainen, A., Airaksinen, J., and Watson, C. (2014). A systematic review of approaches for teaching introductory programming and their influence on success. In *Proceedings of the tenth annual conference on International computing education research*, pages 19–26. ACM.
- Watson, C. and Li, F. W. (2014). Failure rates in introductory programming revisited. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pages 39–44. ACM.