

Mapeamento Sistemático do Ensino Introdutório de Programação nos Ensinos Técnico e Superior no Brasil

Systematic Mapping of Introductory Programming Teaching in Technical and Higher Education in Brazil

Rafael Peixoto de Moraes
Instituto Federal do Rio Grande
do Norte (IFRN)
rafael.moraes@ifrn.edu.br

Valéria Franklin da Costa
Instituto Federal do Rio Grande
do Norte (IFRN)
valeriafranklinn@gmail.com

Ricardo E. P. Scholz
Acadêmico Independente,
Recife, Brasil
ricardoscholz@gmail.com

Resumo

Os desafios no ensino de programação para iniciantes afetam professores e alunos. Índices de evasão e reprovação acima de 30%, barreiras de escalabilidade, dificuldades na personalização do ensino, entre outros, motivaram inúmeros estudos sobre disciplinas introdutórias de programação ao longo dos anos. O objetivo deste artigo é promover uma melhor compreensão dos desafios de ensino-aprendizagem nas disciplinas introdutórias de programação nos níveis técnico e superior, por meio de um mapeamento sistemático da literatura entre 2016 e 2020, tendo como recorte artigos publicados em eventos e revistas nacionais. Como resultado, identificou-se que mais de 80% das publicações são originadas de instituições públicas, especialmente das regiões Nordeste e Sul, e que houve um aumento expressivo das propostas voltadas ao ensino híbrido e EAD. Constatou-se uma predominância em propostas que visam aumentar o engajamento e a motivação dos estudantes. Por outro lado, estudos com foco em estudantes egressos, na baixa representatividade feminina ou em diferenças de gênero, classe social ou raça ainda são escassos. Em relação aos professores, a predominância de estudos foi na produção e aperfeiçoamento de ferramentas de avaliação automatizadas. No entanto, essas ferramentas ainda encontram problemas quanto à sua distribuição e adoção em escala.

Palavras-chave: Introdução à Programação; Mapeamento Sistemático; Informática na Educação; Ensino-aprendizagem.

Abstract

Challenges in teaching programming to beginners affect both teachers and students. High dropout and failure rates above 30%, scalability barriers, difficulties in personalizing teaching, among others, have motivated numerous studies in introductory programming disciplines over the years. The aim of this article is to promote a better understanding of the teaching-learning challenges in introductory programming classes at technical high school and higher education levels by means of a systematic mapping of Brazilian literature between 2016 and 2020, having as a cut articles published in national events and journals. The results indicate that more than 80% of the publications originate from public institutions, especially from the northeastern and the southern regions, and that there was a significant increase in proposals aimed at hybrid and also distance education. Besides, the results show a predominance of proposals aimed at increasing student engagement and motivation. On the other hand, studies focusing on former students, underrepresentation of female students or gender, social or racial differences are still scarce. Regarding teachers, the predominant literature found was centered on the production and improvement of automated assessment tools. Nonetheless, there were still problems regarding the distribution and large-scale adoption of the tools.

Keywords: Introduction to Programming; Systematic Mapping; Informatics in Education; Teaching-learning.

1. INTRODUÇÃO

Os dados do Censo do Ensino Superior Brasileiro 2019, publicados pelo INEP/MEC em 2020 (INEP, 2020), mostram que os cursos de Computação e Tecnologias da Informação e Comunicação (TIC) concentram 16% do total de matrículas do ensino superior, considerando todas as áreas, o que equivale a quase 200 mil alunos. Estudos internacionais apontam que disciplinas introdutórias de programação têm índice médio de reprovação e evasão de 33% (Bennedsen e Caspersen, 2007, 2019; Watson e Li, 2014), embora se discutam os motivos e haja forte variação entre as instituições. Outros estudos estimam médias ainda maiores, chegando a 40%, e podendo atingir valores superiores a 50%, nos piores casos (Castro e Siqueira, 2019; CBSI, 2019). Não houve variação desse patamar nas últimas quatro décadas (Bennedsen e Caspersen, 2019) e o cenário pode ser ainda pior, uma vez que instituições com índices de reprovação e evasão muito altos podem ser relutantes em reportar seus resultados.

Os desafios encontrados no ensino de programação para iniciantes afetam tanto alunos quanto docentes. Entre os principais desafios dos alunos estão a dificuldade em resolver problemas, em expressar, executar e avaliar soluções, em compreender a sintaxe das linguagens de programação, além de problemas comportamentais relacionados à motivação, ao engajamento, ao gerenciamento do tempo, à confiança, entre outros. Já entre os docentes, os principais desafios estão relacionados aos métodos e ferramentas para ensino introdutório de programação, aos limites quanto ao número de alunos por turma, às dificuldades na personalização do ensino, entre outros (Medeiros et al., 2020).

Nesse contexto, e diante da falta de Mapeamentos Sistemáticos (MS) recentes neste recorte, este artigo faz um Mapeamento Sistemático dos principais congressos e periódicos relacionados à Educação em Computação e à Informática na Educação, no Brasil, entre 2016 e 2020. O objetivo é identificar e classificar os estudos publicados nesse período, com relação às dificuldades no processo de ensino-aprendizagem do ensino de introdução à programação, assim como fornecer uma melhor compreensão e categorização dos seus problemas nos ensinos técnico e superior.

O MS é um estudo secundário que permite – por meio de estudos primários – sumarizar a existência de evidências, assim como identificar lacunas nas pesquisas em andamento e promover novas classificações das atividades de pesquisa. Estudos primários são investigações originais que buscam apresentar conhecimento novo a partir de observações e teorias construídas para explicá-las. Portanto, este mapeamento ajudará na construção de um diagnóstico dos cenários reais de ensino-aprendizagem para que outros pesquisadores possam conceber métodos de ensino e artefatos educacionais mais eficazes e, assim, minimizar os problemas atuais de reprovação e evasão nessas disciplinas.

O artigo está organizado da seguinte forma. Na Seção 2, discutimos trabalhos anteriores. Na Seção 3, apresentamos as questões de pesquisa, seus métodos de condução e os de seleção dos estudos, assim como as ameaças à sua validade. Nas Seções 4 e 5, mostramos e discutimos, respectivamente, os resultados da análise dos dados extraídos. Por fim, na Seção 6, elencamos as conclusões e algumas perspectivas de trabalhos futuros.

2. TRABALHOS RELACIONADOS

A revisão sistemática de literatura (RSL) de ensino-aprendizagem de programação proposta por Silva, T. et al. (2015) abrangeu os anos de 2009 a 2013, e incluiu 4 eventos e 2 revistas. A RSL realizada por Ramos et al. (2015) abrangeu os anos de 2001 a 2014, mas

concentrou-se em evidências de artefatos que influenciassem as taxas de sucesso e reprovação. Além disso, o recorte se restringiu ao nível superior e incluiu apenas 4 eventos. O MS proposto por Silva, P. et al. (2015) partiu de uma questão de pesquisa mais contextualizada em "*ambientes de ensino de programação produzidos no Brasil*", além de ter limitado seu recorte até o ano de 2014. Já o MS realizado por Blatt et al. (2017) possuía objetivos similares ao que esta pesquisa propõe, mas se restringiu a apenas 3 eventos e 1 revista, além de ter abrangido apenas os anos 2014 a 2016.

Entre os trabalhos que mais se relacionam com esta pesquisa, a RSL realizada por Borges et al. (2018) respondeu questões de pesquisa similares às aqui propostas, a partir de 45 artigos, mas considerando o recorte temporal de 2012 a 2016. Os resultados apontam uma maioria de propostas para o auxílio ao ensino de programação relacionadas com a aplicação de metodologias apoiadas por ferramentas (51%), e uma grande quantidade de trabalhos que propuseram as ferramentas em si (36%). Uma maior quantidade de estudos relacionados ao nível superior (45%), seguido por uma quantidade menor de estudos voltados para o ensino médio (24%). Cerca de 78% das propostas destinavam-se à modalidade de ensino presencial e 63%, aos desafios enfrentados pelos alunos, apontando como os benefícios mais citados aqueles "*relacionados à melhor aprendizagem das linguagens de programação, estimulando a criatividade, a motivação dos alunos e a diminuição da taxa de reprovação*". Para os docentes, "*os benefícios em comum entre os estudos perpassam da automatização da disponibilização, correção e feedback de atividades, até atividades que exploram o lúdico, seja utilizando recursos digitais, ou não*" (Borges et al, 2018).

A RSL proposta por Holanda et al. (2019) concentrou-se num recorte dos anos 2014 a 2018 do ensino-aprendizagem de programação introdutória nos cursos superiores. O estudo partiu de uma base de 2.109 artigos dos quais 51 foram selecionados para análise. A partir do panorama traçado, foram identificadas as principais técnicas, metodologias e ferramentas adotadas, além de terem sido identificados alguns desafios nas práticas de ensino. Os resultados demonstram que "*as principais dificuldades enfrentadas pelos alunos dizem respeito à lógica de programação e à falta de entendimento da sintaxe da linguagem de programação adotada*", e que "*as estratégias investigadas propõem novas abordagens de ensino, visando fornecer um suporte de aprendizagem dos alunos*" (Holanda et al., 2019).

Por fim, a RSL proposta por Medeiros et al. (2020), concentrou-se nos cursos superiores entre os anos de 2010 a 2016. Partindo de uma base de 2.887 artigos, 69 deles foram relacionados para análise. Além de ter elencado as principais habilidades e conhecimentos necessários para a aprendizagem de programação, a RSL classificou os desafios enfrentados pelos estudantes em 5 categorias. Entre as mais citadas estão a resolução de problemas (21 artigos) e a natureza abstrata da programação (18). Já entre os desafios do corpo docente, os mais citados foram a heterogeneidade dos alunos (12) e as limitações de recursos da equipe e o tamanho das turmas (12).

Dado que este MS se propõe a compreender um panorama mais geral e atual, ele abrange um número maior de fontes primárias que os demais trabalhos aqui analisados, e tem um recorte mais amplo, incluindo trabalhos focados tanto em cursos superiores, quanto em cursos técnicos. Além disso, ele se diferencia também no recorte temporal, ao considerar um intervalo de tempo mais recente, entre os anos de 2016 a 2020.

3. METODOLOGIA

O protocolo de revisão adotado segue as diretrizes para revisões sistemáticas da literatura apresentadas por Kitchenham e Charters (2007). Por meio desse método, pode-se identificar,

avaliar e interpretar os estudos relevantes para determinadas questões de pesquisa em uma área do conhecimento.

3.1 Questões de Pesquisa

Considerando o objetivo e o recorte deste trabalho, as questões de pesquisa deste trabalho serão:

- QP1. Quais são as instituições pesquisando o problema, e como elas estão distribuídas pelo Brasil?
- QP2. Qual é a preponderância de cada nível de escolaridade (ensino técnico versus ensino superior) nas pesquisas realizadas?
- QP3. Quais modalidades de ensino são estudadas?
- QP4. Quais os desafios relatados pelos estudantes no aprendizado de introdução à programação?
- QP5. Quais os desafios relatados pelos professores no ensino de introdução à programação?

3.2 Processo de busca

A pesquisa por estudos primários foi realizada por meio de busca manual em um conjunto previamente definido de veículos. Esse conjunto tem sido amplamente utilizado em outros MSs e RSLs (Borges et al. 2018; Holanda et al., 2019; Medeiros et al., 2020), tendo sido escolhido por esse motivo. Ele contém oito veículos da área de Informática na Educação, sendo quatro revistas científicas e quatro anais de congressos. As quatro revistas foram: Revista Novas Tecnologias na Educação (RENOTE), Revista Brasileira de Informática na Educação (RBIE), Revista Tecnologias na Educação (TECEDU) e Revista de Informática Teórica e Aplicada (RITA). Os anais de congressos foram: Simpósio Brasileiro de Informática na Educação (SBIE), Workshop de Informática na Escola (WIE), Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg), e Workshop de Educação em Computação (WEI). Além dos artigos coletados diretamente desses veículos, realizamos um ciclo de *snowballing* (Kitchenham e Charters, 2007), ampliando, assim, a cobertura da pesquisa.

3.3 Critérios de Inclusão e Exclusão e Processo de Seleção

Esta pesquisa avaliou artigos publicados entre 2016 e 2020, que tratavam do processo de ensino-aprendizado de introdução à programação para o ensino técnico e superior, e que envolviam experimentos realizados no Brasil. Publicações fora desse recorte foram descartadas da análise.

Devido ao grande número de publicações nas fontes de pesquisa, os critérios foram aplicados em duas etapas. Primeiro, realizamos uma etapa de pré-seleção, na qual os CI/CE foram aplicados apenas com base no título, nas palavras-chave e no resumo de cada um dos 3.540 artigos disponíveis nas fontes de estudos primárias.

Essa etapa foi realizada por dois pesquisadores, e acabou por pré-selecionar 198 artigos classificados como pertinentes por um ou ambos os pesquisadores. Todas as publicações que haviam sido classificadas como pertinentes por apenas um pesquisador foram objeto de discussão, com vistas a selecioná-las ou descartá-las definitivamente.

Quadro 1. Descrição dos critérios de inclusão/exclusão.

Crítérios de Inclusão (CI)
CI1: Artigos completos e resumos estendidos que apresentam abordagens de ensino de programação para iniciantes no ensino superior ou no ensino técnico de cursos de informática;
CI2: Artigos que abordam o ensino de programação para iniciantes a partir de estudos secundários, ou seja, revisões sistemáticas da literatura e mapeamentos sistemáticos.
Crítérios de Exclusão (CE)
CE1: Estudos fora dos critérios de inclusão;
CE2: Artigos que propõem ou descrevem uma abordagem para o ensino de programação para iniciantes, sem obtenção de resultados (trabalhos em progresso) ou que duraram menos de dois meses, como frequentemente ocorre com oficinas e workshops;
CE3: Artigos que abordam experimentos fora do Brasil;
CE4: Artigos com menos de cinco páginas (como pôsteres), por, em geral, apresentarem informações excessivamente resumidas dos resultados e discussões;
CE5: Artigos derivados da mesma pesquisa (estudos duplicados), caso em que apenas o artigo mais recente foi considerado.

Já a etapa de seleção foi realizada por apenas um pesquisador, que leu os artigos completamente, aplicando os CI/CE. Esta etapa resultou em 98 artigos selecionados. Ao final da etapa de seleção, realizamos um procedimento de *snowballing* (Kitchenham e Charters, 2007). Dessa maneira, as referências bibliográficas de todos os artigos selecionados foram consideradas como estudos potenciais. O procedimento resultou em outros 91 artigos potenciais, dos quais 4 foram efetivamente selecionados, após a leitura completa e aplicação dos CI/CE.

Ao fim, 102 artigos compuseram a base de dados deste estudo. A lista completa dos artigos selecionados está disponível na web¹. O Quadro 2 traz a relação de artigos por fonte de busca.

3.4 Filtragem e Extração de Informações

Para responder às QP4 e QP5, durante a seleção, classificou-se cada publicação em pelo menos um de dois grupos: i) desafios para o estudante; e/ou ii) desafios para os professores.

No fichamento das publicações, registrou-se as informações necessárias para responder às questões de pesquisa, além das seguintes informações: mecanismo de busca (manual ou *snowballing*), título, palavras-chave, autores, tipo (periódico ou anais), fonte (nome do periódico ou dos anais do evento), ano de publicação, instituições, resumo, objetivo, público-alvo da pesquisa, nível de ensino (técnico e/ou superior), modalidade de ensino (presencial, distância, híbrida, não especificada), método aplicado, linguagem de programação ensinada, número de participantes e respostas para cada questão de pesquisa. A planilha de fichamento das publicações está disponível na web².

¹ Lista de artigos selecionados pelo mapeamento sistemático:
<https://docs.google.com/spreadsheets/d/1nRgk6joR2w1Gv6cSfA9vu-AVKK6zTrp4N70tDLNU9pc/edit?usp=sharing>

² Planilha de fichamento dos artigos do mapeamento sistemático:
<https://docs.google.com/spreadsheets/d/1JA9NWBd0c4fBbeD7W8ex4Ht4uRv-4ZZaYqmyREsWhzQ/edit?usp=sharing>

Quadro 2. Relação de artigos do MS, selecionados segundo CI/CE, por fonte de busca.

Fontes de Busca	Artigos Disponíveis	Pré-selecionados	Selecionados	Selecionados por <i>Snowballing</i> ³	Total	Proporção do Total (%)
SBIE	853	62	34	1	35	34,3
WEI	214	31	17	0	17	16,7
WIE	474	15	7	0	7	6,9
WAlgProg	627	32	18	1	19	18,6
RBIE	148	20	7	1	8	7,8
RITA	111	3	1	0	1	1
RENTE	630	24	12	0	12	11,8
TECEDU	483	11	2	1	3	2,9
Total de artigos	3540	198	98	4	102	100

Além disso, visando facilitar a análise dos resultados e a discussão, as publicações tiveram seus objetivos de pesquisa classificados de acordo com a relação estabelecida por Luxtor-Reilly et al. (2018).

Dificuldades dos estudantes:

- Conteúdo
 - Comportamento do estudante (mensurar as atividades do estudante);
 - Habilidade do estudante (em relação à leitura, à escrita e à correção de código);
 - Propostas para modelo de aprendizagem;
- Sentimento
 - Atitude/engajamento do estudante (mensurar/melhorar o engajamento);
 - Compartilhamento de experiências de estudantes egressos;
- Subgrupos
 - Estudantes com baixa representatividade (gênero, classe social, raça, etc.).

Dificuldades dos docentes:

- Proposta de abordagem pedagógica para ensino;
- Ferramenta/plataforma de ensino (AVAs);
- Ferramenta de avaliação automatizada (juiz *online*);
- Ferramenta de avaliação, como predição de desempenho (prever reprovações);
- Proposta de mudança curricular (linguagens, paradigmas, etc.).

³ Os artigos selecionados no processo *snowballing* foram contabilizados somados aos veículos a partir dos quais foram citados, independente de onde tenham sido originalmente publicados.

3.5 Ameaças à Validade da Pesquisa

As principais ameaças à validade deste MS estão relacionadas ao viés de seleção das fontes primárias de estudo. A escolha dos periódicos pesquisados deu-se pelas fontes mais frequentemente citadas nas RSL e MS com objetivos semelhantes aos nossos, e que foram publicadas nesses mesmos periódicos entre os anos 2015 e 2020. Além disso, o primeiro estágio de pesquisa partiu de uma busca manual nos periódicos, o que pode resultar na não identificação de trabalhos por falha humana. Esse risco acaba por ser mitigado, em parte, pelo processo de *snowballing*, recomendado em revisões sistemáticas de literatura (Kitchenham e Charters, 2007).

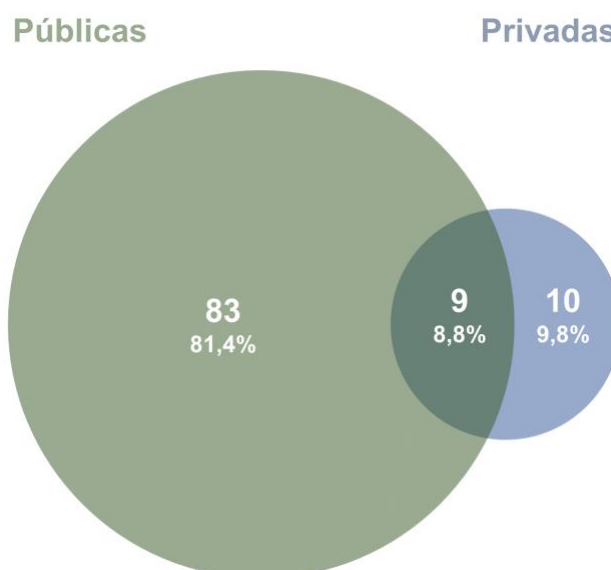
Existe também uma limitação intrínseca ao estudo, uma vez que algumas das questões de pesquisa exigem respostas subjetivas ou não binárias, estando, portanto, sujeitas ao viés de interpretação dos autores.

4. RESULTADOS

4.1 QP1. Quais são as instituições de pesquisa envolvidas na área e como elas estão distribuídas pelo Brasil?

O mapeamento encontrou 68 diferentes instituições de ensino que desenvolveram estudos sobre ensino-aprendizagem de programação introdutória nos níveis técnico e superior. As instituições públicas foram responsáveis – de maneira exclusiva - por 83 publicações (81,4%), as instituições privadas participaram sozinhas de 10 publicações (9,8%), enquanto as parcerias público-privadas responderam por 9 publicações (8,8%), conforme mostra o Gráfico 1.

Gráfico 1. Quantidade e percentual do total de estudos encontrados no MS, por tipo de instituição.



Constatou-se que as regiões Nordeste e Sul concentram a maior parte do contingente de publicações, representando 32 (31,4%) e 31 trabalhos (30,4%), respectivamente. São seguidas pela região Sudeste, com 21 publicações (20,6%), e pela região Norte, com 13 (12,7%). Quatro trabalhos (3,9%) são mistos, intrarregiões, e a região Centro-Oeste tem apenas um trabalho (1,0%). Esses dados são mostrados no Quadro 3, a seguir.

Quadro 3. Quantidade e percentual do total de estudos, por região.

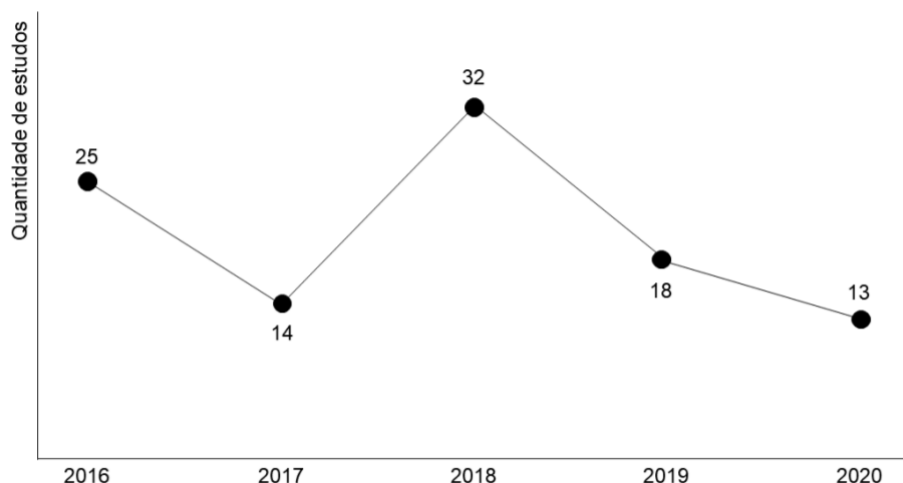
Grande Região do Brasil	Quantidade de Trabalhos	Proporção do Total (%)
Nordeste	32	31,4
Sul	31	30,4
Sudeste	21	20,6
Norte	13	12,7
Em mais de uma região	4	3,9
Centro-Oeste	1	1,0
Totais	102	100,0

Antes de destacarmos as instituições que mais realizaram trabalhos, é importante observar que algumas pesquisas foram realizadas em parceria entre diferentes instituições da mesma região. No Nordeste, as maiores quantidades de estudos foram realizadas na Universidade Federal de Campina Grande (UFCG), com oito publicações, na Universidade Federal de Alagoas (UFAL), na Universidade Federal da Paraíba (UFPB) e no Instituto Federal da Bahia (IFBA), com quatro publicações cada. No Sul, a Universidade Federal do Rio Grande do Sul (UFRGS) teve seis publicações, a Universidade Federal do Paraná (UFPR), cinco, e a Universidade Federal do Pampa (UNIPAMPA), quatro. No Sudeste, a Universidade Federal do Espírito Santo (UFES) e o Instituto Federal do Espírito Santo (IFES) tiveram seis publicações cada, e a Universidade Federal do ABC (UFABC), quatro. No Norte, destacamos, em particular, a Universidade Federal do Amazonas (UFAM), com o maior número de publicações entre todas as instituições, somando 12, e a Universidade Federal de Roraima (UFRR), com sete. No Centro-Oeste, a única publicação foi realizada por uma parceria entre o Instituto Federal do Mato Grosso (IFMT) com a Universidade Federal de Goiás (UFG). Quatro trabalhos foram realizados integrando diferentes regiões, sendo uma entre Sul-Sudeste (Universidade Federal do Rio Grande - FURG, Centro Universitário do Cerrado Patrocínio - UNICERP), uma entre Reino Unido-Sul-Norte (Universidade de Durham, UFPR, UFAM), uma entre Nordeste-Norte (UFAL, UFCG, UFAM) e uma entre Nordeste-Sul (UFRN, FURG). Nota-se, ainda, a preponderância de universidades e institutos públicos, entre os que mais pesquisam o tema.

O Gráfico 2 representa a contabilização dos trabalhos obtidos neste MS por ano de publicação. É possível notar um aumento no número de trabalhos no ano de 2018. Ressaltamos aqui que o congresso WAlgProg não teve registros de anais no ano 2020, o que pode ter causado um viés de decréscimo mais acentuado nesse ano, ou feito com que as publicações potenciais desse evento tenham migrado para outros congressos que ocorreram no mesmo ano.

Assim como na RSL realizada por Holanda et al. (Holanda, Freire e Coutinho, 2019), as regiões Nordeste e Sul do Brasil permaneceram com o maior número de publicações. No Nordeste, dessa vez, o estado com maior número de publicações passou a ser a Paraíba, que contava com uma publicação e passou a contar com 11 (34,4%) do total de estudos selecionados desta região, neste MS. Em seguida, o Estado de Pernambuco, antes responsável por duas publicações, apareceu, neste estudo, com cinco (15,6%), e o Rio Grande do Norte reduziu sua produção de cinco para três publicações. A UFRN, outrora responsável por 80% das publicações no Estado, dessa vez teve apenas um dos três estudos. Na Região Sul, o Rio Grande do Sul continua a ser o estado com maior número de publicações do Brasil, envolvido num total de 22 publicações, ou 71% do total da Região. A Universidade Federal do Pampa (UNIPAMPA), com

Gráfico 2. Quantidade de estudos publicados por ano, entre os selecionados neste MS.



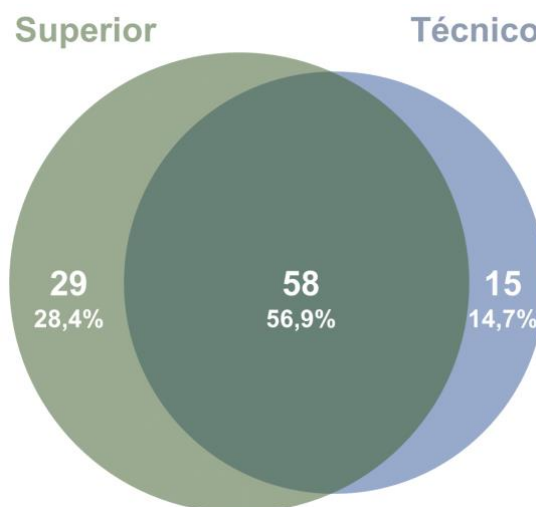
quatro estudos, teve o segundo maior número de publicações, superada pela Universidade Federal Rural do Rio Grande do Sul (UFRGS), com seis estudos.

É possível notar também um alto índice de cooperação intrarregional entre instituições, enquanto apenas quatro estudos foram realizados inter-regionalmente, e um único estudo foi fruto de cooperação com uma instituição internacional.

4.2 QP2. Qual é a preponderância de cada nível de escolaridade (ensino técnico versus ensino superior) nas pesquisas realizadas?

A distribuição dos níveis de escolaridade está ilustrada no Gráfico 3. Nele, pode-se observar onde os estudos primários foram aplicados. A maioria (58 estudos, o que equivale a 56,9% do total) envolveu conjuntamente os níveis técnico e superior. As abordagens aplicadas exclusivamente no nível superior responderam por 29 estudos (28,4% do total), enquanto as aplicadas exclusivamente no nível técnico foram 15 (14,7% do total).

Gráfico 3. Quantidade e percentual do total de publicações do MS, por nível de escolaridade.

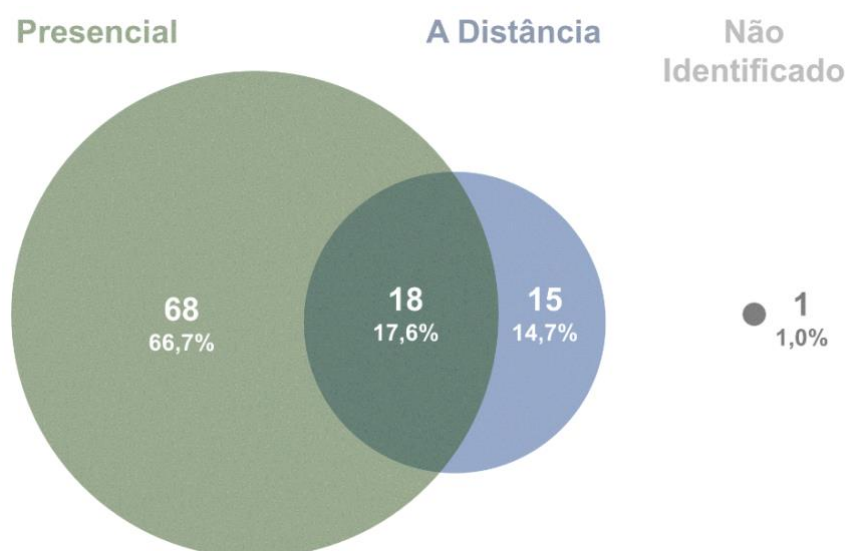


Embora não se tenha incluído nenhum trabalho que envolva o ensino fundamental, o recorte do estudo aqui apresentado demonstrou resultados diferentes da RSL realizada por Borges et. al. (2018) quanto ao nível de escolaridade foco dos estudos. Observou-se um incremento substancial nos trabalhos que consideraram de maneira indistinta os níveis técnico e superior. Antes, eles representavam apenas cinco (11,1% dos estudos considerados por Borges et. al.), e, neste estudo, são 28 publicações (56.9% do total de estudos deste MS).

4.3 QP3. Quais modalidades de ensino são estudadas?

A maior parte dos estudos primários são dedicados exclusivamente à modalidade presencial, na qual foram contabilizadas 68 publicações (66,7% do total). As abordagens híbridas responderam por 18 publicações (17,6%), enquanto a modalidade de ensino a distância contabilizou 15 publicações (14,7%). Apenas em um dos estudos (1,0%) não foi possível identificar a modalidade de ensino. O Gráfico 4 mostra esses dados.

Gráfico 4. Quantidade e percentual de estudos selecionados neste MS, por modalidade de ensino.



4.4 QP4. Quais os desafios relatados pelos estudantes no aprendizado de introdução à programação?

Entre os artigos selecionados, constatou-se que 45 (44,1% do total) tinham como foco os desafios relacionados ao aprendizado de programação. Identificaram-se e classificaram-se os estudos em sete grupos, de três categorias diferentes. Percebeu-se que as publicações cujo foco envolveu a atitude ou engajamento dos estudantes (19 estudos, ou 42,2% do total), e as propostas para modelo de aprendizagem (14 estudos, ou 31,1% do total) foram os destaques. O Quadro 4 apresenta os estudos devidamente categorizados e as quantidades relacionadas em cada caso.

4.4.1 Conteúdo

4.4.1.1 Propostas para Modelo de Aprendizagem

Uma ampla gama de modelos de aprendizagem foi proposta, algumas delas com ênfase em propostas pedagógicas clássicas, como o construcionismo, ou modernas, como o ensino híbrido. Entretanto, uma parte significativa das propostas faz uso de ferramentas que auxiliam os alunos na depuração do código, em linguagens de blocos, ou tentam atuar na redução de barreiras do idioma.

Quadro 4. Desafios enfrentados pelos estudantes no ensino introdutório de programação.

Categoria	Grupos	Estudos Primários	Quantidade (%)
Conteúdo	Propostas para modelo de aprendizagem	[EP02], [EP03], [EP28], [EP34], [EP43], [EP75], [EP76], [EP77], [EP78], [EP84], [EP88], [EP89], [EP91], [EP101]	14 (31,1%)
	Habilidade do estudante	[EP06], [EP23], [EP33], [EP38], [EP86], [EP87], [EP98]	7 (15,6%)
	Comportamento do estudante	[EP13], [EP40],	2 (4,5%)
Sentimento	Atitude/engajamento do estudante	[EP04], [EP05], [EP07], [EP08], [EP09], [EP12], [EP22], [EP35], [EP42], [EP44], [EP60] [EP61], [EP81], [EP82], [EP85], [EP90], [EP92], [EP93], [EP97]	19 (42,2%)
	Experiências de estudantes egressos	[EP102]	1 (2,3%)
Subgrupos	Estudantes com baixa representatividade	[EP25], [EP100]	2 (4,4%)
TOTAL			45 (100,0%)

Por exemplo, um dos estudos propôs um sistema (CFacil) para auxílio na depuração de códigos, permitindo entender e interpretar as mensagens de erros exibidas pelo compilador de uma forma mais simples e compreensível, e facilitando, portanto, corrigir as inconsistências (EP02). Houve também estudos que promoveram uma abordagem de ensino por linguagens visuais, ou metáforas lúdicas (EP03, EP34, EP43), como complemento à abordagem textual amplamente utilizada atualmente. Os resultados apontam que o uso de metodologias lúdicas e visuais facilitaram a compreensão dos pilares estruturais para a programação, com resultados promissores que demonstram uma leve melhora no desempenho dos estudantes, e um nível de aceitação satisfatório. O aumento da motivação dos estudantes também foi relatado em uma proposta construcionista de ensino de conceitos de programação com a utilização de experimentos com Arduino (EP75).

Dois estudos apresentaram formas de facilitar a escrita da sintaxe da linguagem C usando palavras em língua portuguesa. Em um deles, há uma apresentação da pseudolinguagem PortuCol (EP76), semelhante ao Português, porém inspirada em C ANSI. O outro apresenta uma biblioteca que substitui as palavras reservadas da linguagem C em inglês, por substitutos equivalentes em português (EP84). Como resultado, as pseudolinguagens propostas são mais similares à linguagem materna dos estudantes, o que proporciona um aprendizado mais natural. Em relação ao Português Studio, houve um estudo que buscou detalhar os avanços na IDE nos últimos 5 anos (EP89), assim como apresentar o perfil dos usuários que a plataforma atende. Houve também um estudo (EP77) que apresentava um novo modelo de abordagem de introdução à programação, iniciando o ensino pelo uso de funções em Python, no paradigma funcional, ao invés do tradicional paradigma imperativo. Como resultado os pesquisadores apontaram uma redução no tempo e esforço requeridos para construir habilidades de programação, assim como a formação bem-sucedida das bases para aprofundar o desenvolvimento de habilidades em estágios futuros.

Há um estudo que explora a experiência de organização de torneios, a partir de 2010, que envolvem a adoção do jogo educacional de programação Robocode (2022), em código aberto, associado à metodologia de Aprendizagem Baseada em Problemas, em universidades (EP78). Ainda dentro da experiência com robótica, um estudo avaliou os impactos, no longo prazo, de seu uso na aprendizagem e no interesse dos alunos (EP91), concluindo que ela não apenas aumenta o interesse, como também produz mais significado ao aprendizado e melhora o desempenho dos estudantes em todo o ciclo de formação. Um dos estudos (EP88) analisou a motivação, colaboração e percepção de aprendizagem dos estudantes em relação à prática de Coding Dojos, observando resultados expressivos quanto à motivação.

Um estudo (EP28) abordou o uso de videoaulas no aumento da motivação dos estudantes, obtendo 80% de aprovação dos alunos participantes. Por fim, houve um estudo (EP101) que analisou a motivação dos alunos em um modelo interdisciplinar de aulas de matemática, lógica e programação, resultando no aumento da motivação dos alunos nas atividades oferecidas em período extraclasse.

4.4.1.2 Habilidades do Estudante

Um estudo (EP06) elaborou um sistema que cria uma representação do perfil dos alunos, quantificando esforços e qualidade de programação, e permitindo o acompanhamento minucioso da aprendizagem dos alunos na prática da programação. Outro estudo (EP23) realizou uma análise investigativa, apontando que as disciplinas de Matemática, Física e Português do ensino médio podem contribuir no desenvolvimento de competências e habilidades necessárias à aprendizagem de programação. Um estudo (EP33) analisou os enunciados de problemas de programação apresentados nos livros didáticos, tendo concluído que *"para alunos iniciantes é importante que o enunciado seja contextualizado em um contexto conhecido, que as informações sejam apresentadas de forma explícita, com indícios do processo de resolução e com exemplificação"*. Além disso, um outro estudo (EP38) investigou relatos que promoviam a reflexão sobre erros cometidos em disciplinas de programação, tendo destacado como efeito positivo a habilidade do estudante de registrar suas atividades diárias em um portfólio, de maneira a refletir sobre elas e dissociá-las das ideias de inaptidão ou falta de empenho. Outra publicação (EP86) identificou que estudantes que já programavam antes de ingressar no curso – oriundos do ensino técnico – apresentavam um rendimento substancialmente maior nas disciplinas de programação, reforçando a tese de inclusão desses conteúdos no ensino básico, como forma de redução da evasão dos cursos de Computação. Por fim, um dos estudos (EP98) procurou trabalhar as habilidades de estudantes do ensino médio com uma proposta de capacitação em Robótica e Programação, por meio de uma plataforma online, videoaulas e materiais didáticos.

4.4.1.3 Comportamento do Estudante

Um dos estudos (EP13) apresentou e analisou dados de aprovação e frequência a partir da experiência obtida com turmas de programação a fim de propor melhorias quanto à necessidade de se preparar melhor o aluno ingresso nos fundamentos de raciocínio lógico e matemático. Por meio da análise do comportamento dos estudantes, esse estudo identificou que, com alguma frequência, a didática e as metodologias não eram adequadas ao assunto, e apontou a necessidade de preparar melhor os alunos nos fundamentos de raciocínio lógico e matemático. O outro estudo (EP40) apontou *softwares* educativos para apoiar a introdução à lógica e algoritmos mais adequados para o ensino fundamental (Scratch), para o ensino médio (Visualg), e para a graduação (Robocode).

4.4.2. *Sentimento*

4.4.2.1. *Atitude/engajamento do estudante*

Alguns estudos (EP04, EP05, EP12, EP90) focaram em métodos de ensino que permitissem aos estudantes aumentar seu engajamento e/ou desempenho trabalhando em modo cooperativo. Em um deles (EP04), foram identificados estudantes com dificuldades e modelos de progresso similares a fim de formar grupos homogêneos. A partir disso, os tutores estimularam a interação entre os participantes e escolheram um deles para exercer um papel central nesses grupos, a fim de auxiliá-los a alcançar melhores resultados. Em geral os estudos apontaram uma mudança de postura dos alunos, o que os atraiu para um maior interesse nas aulas e promoveu um *feedback* mais imediato, promovido pelos pares, já que, em turmas muito grandes, isso é operacionalmente difícil para o professor.

Um dos estudos (EP07) elaborou uma proposta de identificação de fatores relacionados à motivação em conjunto com o uso de uma estratégia de problematização, ou seja, compreensão e descrição dos problemas antes da codificação da solução. Os autores classificaram os resultados como promissores, mas ainda carecem de análises mais aprofundadas.

Nessa área, houve uma predominância de propostas de gamificação das aulas de lógica de programação, com o objetivo de inserir elementos de apoio ao ensino e aprendizagem (EP08, EP09, EP35, EP44, EP61, EP81, EP92), como o jogo de tabuleiro Logirunner (EP08) e o ambiente Tri-Logic (EP09). Como resultado, as propostas incentivaram o reforço espontâneo das competências e aumentaram o engajamento, tendo auxiliado o ensino por um aspecto lúdico. Dentro da gamificação, houve estudos que propuseram modelos para o ensino gamificado (EP35, EP42). Destacamos um estudo (EP35) que buscou investigar a influência da utilização de um modelo baseado no perfil de jogadores, em técnicas de engajamento e em elementos de jogos, para promover o engajamento estudantil e auxiliar o aprendizado personalizado de programação. Os resultados mostraram que houve aumento nos valores dos indicadores de engajamento, sinalizando uma influência positiva.

Outro estudo (EP22) discutiu conceitos e reflexões baseados no tema da motivação e do modelo ARCS (Atenção, Relevância, Confiança e Satisfação), apresentando, ao final, um relato de experiência. Utilizando as diretrizes e instrumentos de avaliação do modelo ARCS, os autores conseguiram construir mapas mentais para cada categoria, e identificaram a influência das práticas de ensino na motivação dos estudantes. Já um outro estudo (EP82) propôs uma padronização de avaliação dupla em conjunto com o método de aprendizagem baseada em projetos. Primeiro, propondo avaliá-los em relação a métodos e recursos educacionais, e de forma auxiliar, em relação à motivação, ao engajamento ou à lacuna cognitiva.

Também houve estudos que fizeram uso de métodos alternativos com a ferramenta Scratch (EP85, EP93) ou o Arduíno (EP97), a fim de auxiliar na introdução à lógica de programação, em um contexto mais lúdico e motivador. Um deles (EP85) ofereceu um curso introdutório como projeto de extensão, utilizando a ferramenta Scratch. Entretanto, não se verificou correlação do uso da ferramenta com uma redução da taxa de reprovação na disciplina regular. Por outro lado, o projeto com Robótica Educacional (EP97) relatou resultados significativos no aumento da motivação e melhoria do ensino.

4.4.2.2. *Experiências de estudantes egressos*

Apenas um trabalho focou na experiência de alunos egressos (EP102), tendo identificado que os alunos dedicaram um tempo aquém do desejado para o estudo de programação. O mesmo estudo aponta que as principais dificuldades estão nos conteúdos de funções e estruturas de repetição.

4.4.3. Subgrupos

4.4.3.1. Baixa Representatividade Feminina e Diferenças de Gênero, Classe Social ou Raça

Apenas dois estudos focaram na baixa representatividade feminina ou em diferenças de gênero. Um estudo (EP25) focou na aplicação de diferentes estratégias para motivação e ensino de programação para e por mulheres. Entre outros resultados, o trabalho apresentou uma estratégia de ampliação de divulgação para o público-alvo. O outro estudo (EP100) observou que o desempenho dos estudantes não é influenciado pelo gênero, assim como a escolha do curso superior na área de Computação. Nenhum estudo enfocou diferenças de classe social ou raça.

4.5. QP5. Quais os desafios relatados pelos professores no ensino de introdução à programação?

Nesta questão de pesquisa, foram relacionados os 57 estudos (55,9% do total) cujo enfoque são os problemas no processo de ensino de introdução à programação, na perspectiva dos professores. O Quadro 5 relaciona os estudos em cinco grupos. O destaque, nesse caso, fica para a escassez de trabalhos que propõem mudanças curriculares.

Quadro 5. Desafios enfrentado pelos professores no ensino introdutório de programação.

Grupos	Estudos Primários	Quantidade (%)
Ferramenta de avaliação automatizada (juiz <i>online</i>)	[EP11], [EP15], [EP17], [EP19], [EP21], [EP36], [EP37], [EP39], [EP41], [EP45], [EP46], [EP48], [EP49], [EP52], [EP66], [EP67], [EP69], [EP70], [EP71], [EP72], [EP74], [EP99]	22 (38,6%)
Proposta de abordagem pedagógica para ensino	[EP01], [EP10], [EP14], [EP16], [EP18], [EP27], [EP29], [EP31], [EP47], [EP55], [EP56], [EP68], [EP79], [EP94], [EP95], [EP96]	16 (28,1%)
Ferramenta/plataforma de ensino (AVA)	[EP24], [EP26], [EP30], [EP32], [EP51], [EP53], [EP54], [EP57], [EP73], [EP80]	10 (17,5%)
Ferramenta de avaliação como predição de desempenho	[EP50], [EP58], [EP59], [EP62], [EP63], [EP64], [EP65], [EP83]	8 (14,0%)
Proposta de mudança curricular	[EP20]	1 (1,8%)

Os trabalhos na área de avaliação automatizada (juiz *online*) enfocaram na ampliação de funcionalidades de ferramentas existentes (EP37), na comparação entre algoritmos usados por elas (EP11, EP17, EP74), na elaboração de algoritmos de seleção de modelos de soluções para composição de gabaritos de exercícios (EP36, EP99) ou na produção de novas ferramentas (EP15, EP21). Dentre esses, destacam-se dois estudos. Um deles (EP17) apresentou experimentos que usavam algoritmos de avaliação de similaridade, com a finalidade de classificar soluções com a taxonomia SOLO. Os autores destacam que "*foi possível notar que, com 1 ou 4 soluções de referência, os algoritmos de similaridade retornaram uma acurácia muito semelhante*", além de terem apontado que "*independente da quantidade de soluções, a concordância entre algoritmo e especialista não foi satisfatória*". O outro estudo (EP74) propôs e avaliou dois métodos para

recomendação automática de problemas em juízes *online*. Ambos os métodos apresentaram recomendações mais adequadas que o método placebo, que recomendava problemas aleatoriamente. E isso traz implicações imediatas para professores e alunos. Os primeiros, por poderem produzir listas de exercícios variadas para turmas diferentes, a fim de evitar plágio. Aos últimos, por facilitar a tarefa de achar problemas adequados ao seu nível de conhecimento.

Uma pesquisa (EP72) propôs métricas extraídas do código-fonte, capazes de estimar o nível de dificuldade das soluções. O sequenciamento inteligente e adaptativo dos enunciados dos problemas (EP19, EP66) também obteve avanços, com propostas que tiveram resultados similares aos fornecidos por tutores humanos. Também houve estudos sobre a ampliação na capacidade de mediação de erros, que atuaram não apenas na detecção, mas adicionaram comentários com sugestões de correção (EP37). Alguns estudos aplicaram as ferramentas para correções automatizadas das propostas de soluções, de forma a auxiliar os docentes na seleção de questões, reduzir o tempo de *feedback* dos estudantes e o tempo dispensado pelo professor nessa tarefa (EP41, EP71). Houve também trabalhos que buscaram realizar a predição de desempenho ou mensurar as habilidades dos alunos (EP39, EP67, EP69, EP70), com avanços no desempenho da predição, em relação a modelos anteriores (EP39).

Existem muitas possibilidades de solução para cada exercício, o que dificulta o processo avaliativo. Estabelecer critérios de correção e atribuir escores é difícil. E o acompanhamento individualizado dos alunos também é um problema não trivial, que costuma resultar em alunos desmotivados e com dúvidas não esclarecidas (ou sem *feedback* no tempo adequado). Nesse sentido, foram propostas muitas ferramentas para agrupamento de códigos, através de rubricas, a fim de que essas bases de dados pudessem ser usadas por ambientes de correção automática de códigos. Esses sistemas também visam auxiliar os docentes a compreenderem as dificuldades de aprendizagem dos alunos e propor métricas para os ajudar a entender o que caracteriza as boas soluções (EP45, EP46, EP48, EP49). Esses dados também podem ser usados para disponibilizar mensagens mais amigáveis, ajudando no entendimento das mensagens de erros sintáticos das IDEs (EP52). Todas as ferramentas apresentaram soluções que demandavam menos esforço avaliativo dos professores, permitindo uma análise de turmas inteiras de forma detalhada, rápida e holística.

Parte significativa dos trabalhos envolvendo propostas de abordagens pedagógicas focaram em metodologias ativas de aprendizagem, como Aprendizagem Baseada em Problemas (EP95) e Gamificação (EP18, EP56, EP68). Nota-se que, quase sempre, as propostas foram aplicadas em poucas turmas, sendo raras as validações envolvendo centenas de alunos (EP56, EP68). Os sistemas propostos (EP10, EP16, EP94) procuraram possibilitar ao professor acompanhar, por meio de relatórios detalhados, o desempenho dos alunos em tarefas práticas.

Desenvolveram-se ferramentas de programação visual, como o *HelpBlock* (EP24), o *GameMaker* (EP26), o *ALGO+* (EP51) ou a biblioteca *Google Blockly*, integrada ao *URI Online Judge* (EP54), para auxiliar na resolução de problemas, no raciocínio lógico, nas habilidades de abstração, entre outras, de maneira a facilitar a interação e promover maior motivação do estudante. Assim como o *App Inventor* e o *Scratch*, essas ferramentas fazem uso de programação baseada em blocos (*drag-and-drop*) e cumprem o papel estratégico de apresentar fundamentos e conceitos importantes em uma interface amigável.

Houve também um estudo (EP32) que avaliou seis ambientes para o ensino de programação, segundo critérios que permitiam comparar suas funcionalidades no apoio à metacognição. Outro estudo (EP57) descreveu critérios que apoiam o uso de estratégias cognitivas e metacognitivas no desenvolvimento de lógica de programação. Os trabalhos podem servir de guia para professores avaliarem e escolherem os ambientes e estratégias a serem usados no ensino de programação.

Algumas ferramentas propõem o uso de técnicas de aprendizagem de máquinas para identificar zonas de aprendizagem dos alunos e prever suas notas com base nos registros de atividades de programação, com alguns modelos tendo alcançado acurácia próxima a 80% (EP50, EP58, EP59, EP64), a partir de dados gerados entre duas a quatro semanas.

Destaque-se que um único trabalho (EP20) se dedicou a discutir o conteúdo das aulas, e tece críticas afirmando que “*não se resolvem as dificuldades criadas por ensino centrado num resultado de aprendizagem que não se alinha com a natureza do conhecimento ensinado*”. Destacando, com isso, o decisivo papel que a organização do ensino desempenha na construção efetiva da capacidade dos estudantes aprenderem a pensar algorítmicamente.

5. DISCUSSÃO

5.1 Panorama Geral

A maioria das pesquisas foram realizadas em universidades e institutos públicos (81,4%, Gráfico 1) ou envolvendo um ente público, em parcerias público-privadas (8,8%). Supõe-se que isso ocorra como resultado da cultura e da estrutura dos ensinos técnico e superior no Brasil, em que as universidades e institutos federais são incentivados a realizar um maior número de publicações, combinado a um menor volume de fomento para pesquisas captado pelo setor privado. O estudo realizado por Santos (2016) corrobora com essa suposição. Segundo a análise dos dados do *Ranking* Universitário Folha (RUF), baseado em metodologias de *rankings* internacionais, a mensuração da produção científica produzidas em universidades públicas e privadas seguem em percentuais similares aos constatados neste estudo. Corroborando essa informação, o relatório *Research in Brazil* (Carta, 2018) mostra que a indústria não colabora com a academia, no Brasil. Em 23 Unidades Federativas, menos de 1% da pesquisa científica tem colaboração com a indústria. Esse percentual não passa de 2,28% nas demais, apesar da existência de leis de incentivo, como a Lei nº 11.487/2007 (Lei de Incentivo à Pesquisa, que altera a Lei nº 11.196/2005). O mesmo relatório também permite concluir que as universidades privadas praticamente não produzem conhecimento no país, considerando-se como métrica o impacto de publicações nos rankings internacionais.

Quanto à distribuição regional, percebemos que, apesar da região Sudeste ter uma presença mais forte na produção tecnológica do país, com mais polos de tecnologia, as instituições de pesquisa das regiões Nordeste (31,4%, Gráfico 2) e Sul (30,4%) concentram a maior parte das pesquisas, no recorte desta pesquisa. É possível que o processo de descentralização espacial da produção e da colaboração científicas que transformou o país ao longo das últimas décadas tenha resultado no aumento da participação de autores das regiões cientificamente menos tradicionais, como Sul e Nordeste (Sidone et al., 2016). Especialmente nas últimas duas décadas, a região Nordeste viu crescer rapidamente a quantidade de campi de universidades públicas, responsáveis pela maioria das atividades científicas (Gráfico 1). No entanto, aparentemente, o Norte (12,7%) e sobretudo o Centro-Oeste (1,0%) ainda necessitam de mais atenção do poder público nesse aspecto.

A grande maioria das pesquisas (56,9%, Gráfico 4) selecionadas neste trabalho utiliza um recorte que engloba o nível técnico e o nível superior, conjuntamente. Isso é consistente com as observações de Fincher e Robins (2019), que indicam que os estudos internacionais parecem considerar como um único grupo os alunos de ensino médio e os do início da graduação, fazendo apenas distinção para grupos do ensino fundamental e infantil. Nenhum dos estudos primários destacou diferenças significativas em relação às dificuldades específicas dos alunos de ensino médio técnico, comparados aos alunos de cursos introdutórios de graduação. Como primeira hipótese, isso pode indicar que não são relevantes as diferentes características entre estudantes que se iniciam em programação nessa faixa etária, e que a escolha do recorte se dá

majoritariamente pela conveniência. Ou, numa segunda hipótese, que existe uma lacuna de estudos que claramente diferencie as particularidades dos estudantes do ensino técnico em relação aos do ensino superior.

Embora a maioria dos estudos tenha sido aplicado na modalidade presencial, constatamos, neste estudo, que o número de estudos aplicados na modalidade híbrida (Gráfico 5) quase dobrou em relação ao estudo de Borges et al. (2018). Movimento ainda mais acentuado se observa em relação aos estudos EAD, que representavam apenas 2% naquela pesquisa, e agora saltaram para 14,7%. Acredita-se que o aumento da cobertura de Internet no país, junto à maior disponibilidade de ferramentas, pode ter sido uma influência positiva nesse aspecto. Isso leva a crer, também, que aplicações nas modalidades híbrida e EAD permanecerão como tendências nos próximos anos, aceleradas pelas restrições sanitárias decorrentes da pandemia de Covid-19, que forçou a utilização do modelo EAD em larga escala, mostrando aos diversos atores envolvidos que o modelo é tecnicamente viável e uma opção pedagógica válida.

5.2 Dificuldades e Desafios dos Estudantes e dos Professores

Quando se trata de soluções para superar as dificuldades dos estudantes, notamos que as atitudes e o engajamento destes (42,2%, Quadro 3), as propostas para modelos de aprendizagem (31,1%) e as abordagens voltadas para identificar habilidades necessárias aos estudantes (15,6%) são os principais focos de pesquisa. No primeiro e no segundo casos, diversos estudos objetivaram tornar o ambiente de aprendizado mais próximo do dia a dia dos alunos, como forma de aumentar a motivação. Isso ocorre por meio de diversos artifícios, como modificações na metodologia, gamificação, introdução da robótica, mudanças no paradigma de programação, utilização de linguagens de programação em português (língua nativa dos estudantes, em regra). Outras abordagens, focadas nos comportamentos dos estudantes (4,5% dos artigos), nas experiências dos estudantes egressos (2,3%) ou em recortes de estudantes com baixa representatividade (4,4%), poderiam ser mais exploradas. Por fim, nas abordagens que estudam habilidades necessárias aos estudantes, parece frequente o interesse em entender os fatores determinantes, no Ensino Técnico, para o sucesso no Ensino Superior (EP23, EP33, EP85). Esse tipo de relação é claramente importante para que se possa melhor preparar os alunos no Ensino Técnico, sobretudo aqueles com interesse em ingressar em cursos que envolvem programação de computadores. No entanto, não basta entender quais matérias no Ensino Médio influenciam nos resultados no Ensino Superior, mas como se pode melhorar o ensino dessas matérias para esse fim, o que ainda é uma lacuna.

A maior parte dos estudos que tratam sobre dificuldades dos professores aborda ferramentas de avaliação automatizada, conhecidas como AA (38,6%, Quadro 4). Isso parece ser um reflexo de um sistema educacional em que, frequentemente, encontramos salas lotadas e escassez de professores e monitores. A redução do tempo de *feedback* passa a ser um problema relevante, diante do problema de escalabilidade que a educação enfrenta no país. No entanto, constatamos que quase todas as ferramentas se encontram em estágios experimentais, e trabalham em fontes e tamanhos de base de dados heterogêneos, o que dificulta a comparação da eficácia entre elas. Além disso, poucos trabalhos parecem dar continuidade às suas ferramentas, e é comum elas não estarem facilmente disponíveis, o que dificulta escalar a aplicação para além de projetos-piloto, um problema também relatado por Holanda et al. (2019). Dessa maneira, entende-se que há, portanto, um problema de escalabilidade das soluções, sendo necessário pensar não apenas nas ferramentas, mas, sobretudo, em como promover sua distribuição e adoção em maior escala.

Observa-se também que as propostas de mudança curricular (1,8%, Quadro 4) costumam ser um tema negligenciado. Acreditamos que a burocracia envolvida na realização prática de mudanças desse tipo e o tempo de resposta mais longo para verificar se as alterações de fato surtiram efeitos podem estar desencorajando pesquisas nesse sentido. Grades curriculares mais

fluidas poderiam mitigar o primeiro problema – tornando-se um incentivo para mais pesquisas propondo soluções que envolvem mudanças curriculares –, enquanto gestores e métricas mais comprometidos com resultados de médio e longo prazo poderiam promover melhores incentivos, com relação ao segundo problema.

Por fim, as propostas metodológicas ampliaram discussões como a mediação de erros, e deram ênfase em abordagens pedagógicas com um viés prático. Muitas delas embasadas em teorias de aprendizagem clássicas, e em ambientes gamificados, sendo frequentemente analisados os fatores de engajamento e desempenho do aluno. Porém, a maioria desses estudos foi validada para uma quantidade reduzida de alunos, ou são funcionalidades e sistemas ainda em estágio embrionário. Assim como nas questões relacionadas à produção e ao aperfeiçoamento de ferramentas, também nas propostas metodológicas encontram-se barreiras de escalabilidade. Os planos de aula, as listas de exercícios, os questionários aplicados e os demais materiais didáticos desenvolvidos, com frequência, não estão facilmente disponíveis para serem reaplicados em um novo contexto. Além disso, até dentro de uma mesma instituição, frequentemente os projetos não escalam entre diferentes unidades ou *campi*. Isso pode indicar a necessidade de criar incentivos para uso exclusivo no reaproveitamento e aperfeiçoamento de projetos-piloto, a fim de sistematizá-los e transformá-los em processos com aplicação real, em maior escala.

6. CONCLUSÃO

6.1 Contribuições da Pesquisa

Esse estudo tem como principais contribuições: o mapeamento, a classificação e a categorização dos desafios do ensino-aprendizado de introdução à programação no ensino técnico e superior, tendo avaliado 102 estudos primários publicados entre os anos 2016 e 2020, em oito periódicos nacionais, cobrindo uma lacuna nos estudos desse tipo, com recorte no Brasil. Com base nisso, outra contribuição relevante é a discussão sobre os principais achados e lacunas, que incluem sugestões potencialmente relevantes para a solução das barreiras identificadas.

Observou-se que a maior parte das publicações são originadas de instituições públicas, com expressiva participação das regiões Nordeste e Sul. E que houve um aumento expressivo de propostas voltadas ao ensino híbrido e EAD, uma tendência que deve se consolidar após a pandemia de Covid-19 no país.

Em relação às dificuldades dos estudantes, predominam as pesquisas sobre como aumentar as atitudes e o engajamento, e propostas de novos modelos de aprendizagem. Entretanto, estudos cujo enfoque sejam estudantes egressos, a baixa representatividade feminina ou diferenças de gênero, sociais ou raciais, ainda são escassos.

Em relação às dificuldades dos professores, as abordagens na produção e aperfeiçoamento de ferramentas de avaliação automatizadas são predominantes. Mas enquanto tentam reduzir barreiras de escalabilidade no ensino, como a falta de professores e monitores, esses estudos também encontraram problemas para promover sua distribuição e adoção em maior escala.

Esperamos que após a conclusão e divulgação deste trabalho, os estudos aqui mencionados possam ter uma melhor visibilidade e, em especial, possam ser reproduzidos e aperfeiçoados em outros ambientes, para que sejam efetivamente adotados de forma sistematizada e processual.

6.2 Trabalhos Futuros

Pretende-se, como próximo passo, produzir um mapeamento sistemático similar, mas cujos critérios foquem na coleta de publicações internacionais, a fim de comparar os resultados,

encontrar similaridades e aprofundar questões mais específicas sobre as abordagens relacionadas neste MS.

6.3 Agradecimentos

Os autores agradecem ao Professor Rodrigo Medeiros (IFPB), pelas relevantes informações, sugestões e observações sobre esta pesquisa. E a professora Kallyne Machado Bonifácio (UFPB), cuja mente refinada e disciplinada, continuamente semeia a relevância da produção e divulgação científica.

7. REFERÊNCIAS

- Bennedsen, Jens; Caspersen, Michael E. Failure rates in introductory programming. Association for Computing Machinery Special Interest Group on Computer Science Education (ACM SIGCSE), 2007. DOI: doi.org/10.1145/1272848.1272879.
- Bennedsen, Jens; Caspersen, Michael E. Failure rates in introductory programming: 12 years later. Association for Computing Machinery (ACM), 2019. DOI: [10.1145/3324888](https://doi.org/10.1145/3324888).
- Blatt, Lucas; Becker, Valdecir; Ferreira, Alexandre. Mapeamento Sistemático sobre Metodologias e Ferramentas de apoio para o Ensino de Programação. VI Congresso Brasileiro de Informática na Educação (CBIE), 2017. DOI: [10.5753/cbie.wie.2017.815](https://doi.org/10.5753/cbie.wie.2017.815).
- Borges, R. P.; Oliveira, P. R. F.; Lima, R. G. da R.; Lima, R. W. A systematic review of literature on methodologies, practices, and tools for programming teaching. Institute of Electrical and Electronics Engineers Latin America Transactions (IEEE Latin America Transactions), 2018. DOI: [10.1109/TLA.2018.8408443](https://doi.org/10.1109/TLA.2018.8408443).
- Carta Campinas, 2018. Relatório mostra que universidade particular no Brasil não produz conhecimento. Disponível em: <https://cartacampinas.com.br/2018/01/xrelatorio-internacional-mostra-que-universidade-particular-no-brasil-nao-produzem-conhecimento/>. Acesso em 03 de agosto de 2022.
- Castro, Ronney Moreira; Siqueira, Sean. Metodologias, técnicas, ambientes e tecnologias alternativas utilizadas no ensino de algoritmos e programação no ensino superior no Brasil. Anais dos Workshops do VIII Congresso Brasileiro de Informática na Educação (WCBIE), 2019. DOI: [10.5753/cbie.wcbie.2019.228](https://doi.org/10.5753/cbie.wcbie.2019.228).
- Fincher, Sally A.; Robins, Anthony V. The Cambridge Handbook of Computing Education Research. Cambridge University Press, 2019. ISBN-10: 1108721893.
- Florenzano, C. Computação é o que mais sofre evasão em universidades públicas e privadas. Comunidade Brasileira de Sistemas de Informação (CBSI), 2018. Disponível em: <https://www.cbsi.net.br/2018/02/curso-de-computacao-e-um-dos-que-mais.html>.
- Holanda, Wallace Duarte; De Paiva Freire, Laís; Da Silva Coutinho, Jarbele Cássia. Estratégias de ensino-aprendizagem de programação introdutória no ensino superior: uma Revisão Sistemática da Literatura. Revistas Novas Tecnologias na Educação (RENTE), 2019. DOI: [10.22456/1679-1916.95905](https://doi.org/10.22456/1679-1916.95905).
- INEP, 2020. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira - Censo da Educação Superior 2019. Disponível em: https://download.inep.gov.br/educacao_superior/centro_superior/documentos/2020/Apresentacao_Censo_da_Educacao_Superior_2019.pdf. Acesso em 03 de novembro de 2021.
- Kitchenham, B.; Charters, S. Guidelines for performing Systematic Literature Reviews in Software Engineering. EBSE Technical Report, 2007.

- Luxtor-Reilly, A.; Simon; Albluwi, I.; Becker, B. A.; Giannakos, M.; Kumar, A. N.; Ott, L.; Paterson, J.; Scott, M. J.; Sheard, J.; Szabo, C. Introductory Programming: A Systematic Literature Review. Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE), 2018. DOI: [10.1145/3293881.3295779](https://doi.org/10.1145/3293881.3295779).
- Medeiros, R. P.; Falcão, T. P.; Ramalho, G. L. Ensino e Aprendizagem de Introdução à Programação no Ensino Superior Brasileiro: Revisão Sistemática da Literatura. Workshop Sobre Educação em Computação (WEI), 2020. DOI: [10.5753/wei.2020.11155](https://doi.org/10.5753/wei.2020.11155).
- Ramos, V.; Wazlawick, R.; Galimberti, M.; Freitas, M.; Mariani, A. C. A Comparação da Realidade Mundial do Ensino de Programação para Iniciantes com a Realidade Nacional: Revisão sistemática da literatura em eventos brasileiros. Simpósio Brasileiro de Informática na Educação (SBIE), 2015. DOI: [10.5753/cbie.sbie.2015.318](https://doi.org/10.5753/cbie.sbie.2015.318).
- Santos, S. M.; Noronha, D. P. O Desempenho das Universidades Brasileiras em Rankings Internacionais. Em Questão, Porto Alegre, v. 22, n. 2, p. 186–219, 2016. DOI: [10.19132/1808-5245222.186-219](https://doi.org/10.19132/1808-5245222.186-219).
- Sidone, O. J. G.; Haddad, E. A.; Mena-Chalco, J. P. A ciência nas regiões brasileiras: evolução da produção e das redes de colaboração científica. Transinformação, Scientific Electronic Library Online (SciELO), 2016. DOI: [10.1590/2318-08892016002800002](https://doi.org/10.1590/2318-08892016002800002).
- Silva, P.; Cavalcante, M. T. C.; Fechine, J.; Costa, E. Silva. Um Mapeamento Sistemático sobre Iniciativas Brasileiras em Ambientes de Ensino de Programação. Simpósio Brasileiro de Informática na Educação (SBIE), 2015. DOI: [10.5753/cbie.sbie.2015.367](https://doi.org/10.5753/cbie.sbie.2015.367).
- Silva, T. R.; Medeiros, T.; Medeiros, H.; Lopes, R.; Aranha, E. Ensino-aprendizagem de Programação: Uma Revisão Sistemática da Literatura. Revista Brasileira de Informática na Educação (RBIE), 2015. DOI: [10.5753/rbie.2015.23.01.182](https://doi.org/10.5753/rbie.2015.23.01.182).
- SOURCE FORGE, n.d.. Robocode. [online]. Disponível em: <https://robocode.sourceforge.io/>. Acesso em 15 de março de 2022.
- Watson, C.; Li, F. W. B. Failure rates in introductory programming revisited. Conference on Innovation & Technology in Computer Science Education (ITiCSE), 2014. DOI: [10.1145/2591708.2591749](https://doi.org/10.1145/2591708.2591749).