



As dificuldades enfrentadas pelos estudantes na aprendizagem de programação de computadores: Uma Revisão Sistemática da Literatura

Jefferson Ribeiro de Lima - PPGIE - UFRGS - Brasil - jlima@ifes.edu.br

<https://orcid.org/0000-0001-6677-0970>

Crediné Silva de Menezes - PPGIE - UFRGS - Brasil - credine@gmail.com

<https://orcid.org/0000-0002-2709-7135>

Resumo. Apresenta-se uma Revisão Sistemática da Literatura (RSL) com o objetivo de levantar as principais dificuldades na aprendizagem de programação introdutória e possíveis estratégias para mitigação. Foram analisados 85 estudos primários relevantes. Neles identificamos os seguintes obstáculos: a resolução de problemas, a elaboração de estruturas lógicas e o entendimento da sintaxe das linguagens de programação. Além disso, os estudantes enfrentam barreiras emocionais, tais como, falta de motivação, ansiedade e frustração. Entre as estratégias para amenização, destacamos, o uso de ferramentas visuais, a gamificação e metodologias ativas.

Palavras-Chaves: programação de computadores, dificuldades, ensino, aprendizagem

The Difficulties Faced by Students in Learning Computer Programming: A Systematic Literature Review

Abstract. Here is a Systematic Literature Review (SLR) aimed at identifying the main difficulties in learning introductory programming and possible mitigation strategies. Eighty-five relevant primary studies were analyzed. We identified the following obstacles: problem-solving, the development of logical structures, and understanding the syntax of programming languages. Additionally, students face emotional barriers such as lack of motivation, anxiety, and frustration. Among the strategies for alleviation, we highlight the use of visual tools, gamification, and active methodologies.

Keywords: computer programming, difficulties, teaching, learning

1. Introdução

Ao longo dos anos, o pensamento computacional vem se destacando como uma estratégia fundamental para apoiar o desenvolvimento de habilidades dos estudantes em diversos níveis educacionais, principalmente nos cursos superiores de graduação, que demandam habilidades de raciocínio lógico e matemático [Carvalho et al. 2017]. Esse cenário, ainda ganha mais destaque com a decisão de alguns países de incluírem a disciplina de programação como parte obrigatória dos currículos, a partir da educação primária [Valente et al. 2016]. Vislumbra-se que essas mudanças educacionais podem ser benéficas ao longo dos anos na formação do estudante.

Contudo, aprender programação de computadores, ainda continua sendo um dos desafios enfrentados por muitas pessoas. Segundo [Chang 2016; Da Conceição et al. 2022], a resolução de problemas é uma abordagem fundamental para que o estudante compreenda, abstraia e modele uma ideia com a intenção de fornecer uma solução. Na programação de computadores, ocorre a transcrição da ideia para uma linguagem compreensível pela máquina, que envolve as etapas da escrita, testes, depuração e execução de um conjunto de códigos, usando uma linguagem de programação. Esses



estágios podem ser trabalhosos e complexos, especialmente para os iniciantes desta aprendizagem.

Para muitos estudantes, a programação é considerada difícil, pois sua compreensão abrange desenvolver habilidades que podem levar algum tempo e comprometimento para que haja um domínio do conteúdo [Baist et al. 2017]. No entanto, alguns esforços têm sido aplicados no intuito de facilitar e apoiar a resolução de problemas de programação. Alguns pesquisadores, vem testando abordagens de ensino diferenciadas para simplificar esse entendimento, como: virtualização, jogos, quebra-cabeças e outros. [Ferri et al. 2016].

Até o presente momento, foram diagnósticas diversas origens das dificuldades em relação ao aprendizado de programação de computadores. Segundo [Zanin et al. 2019] estas vão desde os conhecimentos prévios até as questões relacionadas com as atitudes dos estudantes, incluindo também a ausência de métodos de ensino mais dinâmicos e interativos. Além disso, propõem que as atividades envolvidas no ensino de programação estejam sempre alinhadas com assuntos do interesse dos alunos. Portanto, este artigo apresenta um panorama atualizado sobre os estudos que abordam o ensino e aprendizagem de programação de estudantes iniciantes, principalmente dos cursos superiores. Além disso, analisa e fornece os resultados das experiências realizadas dos artigos selecionados.

Para uma melhor apresentação da problemática, o trabalho está dividido em cinco seções, a partir da introdução. A Seção 2 aborda alguns trabalhos relacionados com o tema desta pesquisa. A Seção 3 descreve o método empregado na construção desta Revisão Sistemática da Literatura (RSL). Na Seção 4, são expostos os resultados e discussões alcançados. E, finalmente, na Seção 5 são apresentadas as considerações finais.

2. Trabalhos Relacionados

Vários estudos de RSL têm sido realizados ao longo dos anos como os de [Silva et al., 2015; Borges et al., 2018 e Holanda, 2019] com a finalidade de obter informações a respeito das dificuldades, metodologias, experimentos e ferramentas que possam contribuir com a aprendizagem de programação introdutória.

A RSL dos autores [Silva et al. 2015] destaca as abordagens para o ensino-aprendizagem de programação, publicadas entre os anos de 2009 a 2013, em periódicos nacionais como fonte de busca. Ao fim do estudo, 73 trabalhos foram incluídos para a coleta de dados. Os resultados mostraram uma tendência no desenvolvimento de novas ferramentas para apoiar o ensino de algoritmos nos cursos do ensino médio. A proposta destaca a utilização de jogos digitais. Em sua maioria, esses jogos digitais estão voltados para melhorar o rendimento dos alunos em disciplinas introdutórias de programação.

No ano de 2018, uma RSL conduzida por [Borges et al. 2018], analisou cenários relativos às abordagens de ensino-aprendizagem de programação no Brasil, entre os anos de 2012 e 2016. Os pesquisadores utilizaram fontes de busca nacionais para a seleção dos trabalhos. Ao todo, foram selecionados 45 artigos. Como resultado, os autores sinalizam a adoção de abordagens pedagógicas mais ativas que estimulam uma aprendizagem mais centrada no aluno, valorizando a participação, a colaboração, a autoria, a construção do conhecimento, e assim, consequentemente, incentivando e motivando os alunos neste processo de aprendizagem. O trabalho também destaca a inclusão de novas ferramentas que foram elaboradas para serem utilizadas em diferentes níveis de ensino.



[Holanda et al. 2019] aplicou uma Revisão Sistemática de Literatura para analisar as principais contribuições sobre o ensino-aprendizagem de programação introdutória no ensino superior, disponibilizando os resultados das experiências realizadas entre 2015 a 2019, destacando as principais técnicas, metodologias e aplicações utilizadas para apoiar os alunos. A busca resultou em um total 51 artigos incluídos para coleta e síntese dos dados. Como resultado deste estudo, a pesquisa destaca as altas taxas de reprovação, dificuldades para compreensão de lógica, além de citar que diversos métodos de ensino e aprendizagem têm sido desenvolvidos para apoiar os estudantes.

3. Metodologia

Para efetuarmos a realização da busca, seguimos a abordagem de revisão da literatura de [Kitchenham 2007], que visa identificar, avaliar e interpretar todas as evidências científicas disponíveis e relevantes para uma questão de pesquisa, o uso deste protocolo sugere as seguintes etapas de trabalho: 1) formulação das questões de pesquisa; 2) definição de termos de busca; 3) estabelecimento de critérios para inclusão e exclusão; 4) revisão dos artigos selecionados e 5) apresentação e análise dos resultados. Vamos detalhar melhor essas etapas a partir das próximas seções, onde serão apresentados o modo operante deste protocolo. Através do uso desta metodologia, analisamos as técnicas e ferramentas relacionadas ao ensino e aprendizagem de programação que estão sendo utilizadas para apoiar os estudantes em disciplinas introdutórias de programação.

3.1 Questões de Pesquisa

A seleção dos estudos primários foi precedida pela definição das questões de pesquisa norteadora, importante marco para a extração de dados e fundamentação do estudo principal do problema pesquisa, que é “Quais são as dificuldades de aprendizagem encontradas por estudantes ingressantes em disciplinas introdutórias de programação de computadores?”. Para tentar compreender essa dificuldade, elencamos algumas questões de pesquisa (QP) que podem auxiliar neste entendimento:

QP1: Quais dificuldades dos estudantes foram identificadas nas pesquisas de ensino e aprendizagem de programação de computadores?

QP2: Quais soluções são oferecidas na literatura para atenuar as dificuldades no ensino e aprendizagem de programação?

QP3: Como as tecnologias digitais são utilizadas para apoiar aprendizagem de programação?

QP4: Quais abordagens pedagógicas foram mais utilizadas para apoiar o ensino e aprendizagem de programação?

3.2 Busca de artigos

A pesquisa por estudos primários foi realizada através de três fontes de busca de relevância acadêmica internacional: ACM Digital Library, IEEE Xplore e Scopus. A decisão de selecionar essas bases de dados para nossa pesquisa ocorreu pela disponibilidade de acesso aos artigos publicados de áreas múltiplas, incluindo informática na educação. Os dados foram obtidos utilizando um filtro aplicado correspondente aos últimos 5 anos, de 2019 até 2023.

A partir da definição das questões de pesquisa, criamos uma string de busca em inglês, conforme apresentamos na Tabela 1. Utilizamos uma expressão lógica para retornar os estudos que se enquadrassem em nosso objetivo, focando nas dificuldades do ensino e aprendizagem de programação.

**Tabela 1. String de busca utilizada**

Base de Dados	String de busca
ACM, IEEE e Scopus	("programming" and "computer") and ("teaching" or "learning") and ("difficulty" or "problem" or "obstacle" or "barrier") and ("approach" or "strategy")

3.3 Critérios de inclusão e exclusão

A seleção dos trabalhos ocorreu em quatro etapas: 1) avaliação do título, resumo e palavras chaves; 2) revisão e leitura da introdução e conclusão; 3) leitura completa dos trabalhos, e por fim, 4) a inclusão e extração dos dados selecionados. As fontes de pesquisa selecionadas na primeira etapa foram avaliadas de periódicos completos, onde os títulos, resumos e palavras-chaves foram verificados na primeira fase da seleção. Os artigos pré-selecionados, foram catalogados para posterior reavaliação, com base nos critérios de inclusão e exclusão, conforme o Quadro 1.

Quadro 1. Critérios de inclusão e exclusão

Critérios para inclusão (CI)	Critérios para exclusão (CE)
- Estudos que podem ser relevantes sobre o tema de pesquisa (após análise de título, resumo e palavras-chave relacionadas). - Artigos publicados de modo completo em inglês. - Que tratem as dificuldades, métodos e/ou técnicas e/ou ferramentas no ensino/aprendizagem de programação de computadores, de acordo com o público alvo.	- Trabalhos com acesso negado ou incompletos, tais como pôsteres e apresentações. - Estudos duplicados (Selecionar o mais completo) - Quando não identificadas as contribuições relacionadas às questões de investigação.

Na etapa 2, foram realizadas as leituras dos títulos, introdução e conclusão dos artigos pré-selecionados, sendo reclassificados conforme os critérios de inclusão ou exclusão. Na etapa 3, os artigos escolhidos do estágio anterior foram lidos na íntegra, aplicando-se os critérios de inclusão e exclusão. Os estudos aceitos, foram incluídos na etapa 4.

3.4. Resultados da busca

A busca inicial resultou em 1026 trabalhos primários coletados. Na segunda fase, foram selecionados 412 artigos. Na terceira fase, foram classificados 154 documentos, e por fim, na última fase, foram incluídos 85 artigos, arranjados em uma planilha, incluindo informações como título do artigo, origem da publicação, ano da contribuição e algumas informações complementares primordiais para a condução desta pesquisa. Os dados incluídos neste documento podem ser acessados através do link [\[https://bit.ly/rs1pesquisa\]](https://bit.ly/rs1pesquisa). A Tabela 2, apresenta o resultado geral da busca realizada, considerando os trabalhos disponíveis em cada fonte de busca e seleção desta RSL.

Tabela 2: Resultado geral das buscas entre 2019 a 2023.

Base de dados	Fase 1	Fase 2	Fase 3	Incluídos
ACM	217	106	29	21
IEE	312	128	31	19
Scopus	497	178	94	45
Total	1026	412	154	85



Como é possível visualizar no Gráfico 1, nota-se um acréscimo de publicações relacionadas com o ensino e aprendizagem de programação ao longo dos anos. O ano de 2023 concentrou o maior índice das publicações incluídas na pesquisa, com 29,41% do total de trabalhos avaliados com base nos critérios definidos. Em 2022, incluímos 21,17% dos trabalhos. Em 2021, obtivemos uma concentração de 20% de publicações para inclusão. Em 2020, obtivemos o menor índice de publicações incluídas, com aproximadamente 14,11%. Por fim, em 2019, foram incluídos aproximadamente 15,29% das publicações selecionadas neste período.

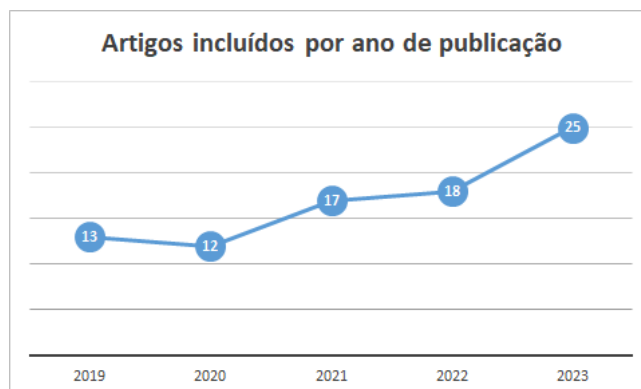


Gráfico 1. Estudos incluídos por ano de publicação

4. Resultados e discussões

Nesta seção, são apresentados os resultados obtidos da RSL. Após a extração e caracterização das informações gerais extraídas dos trabalhos completos (incluídos), visando responder as questões norteadoras propostas na seção na 3.1, os conteúdos foram organizados de acordo com sua importância para responder as questões de pesquisa, tais como: dificuldades no ensino/aprendizagem da programação de computadores e a capacidade de resolução de problemas. Além disso, também levamos em consideração fatores comportamentais e pedagógicos como: à falta de motivação e ausência da implantação de práticas de aprendizagem mais ativas [Souza et al. 2016]. As próximas seções respondem individualmente as questões de investigação elaboradas para esse trabalho.

4.1 - QP1: Quais dificuldades dos estudantes foram identificadas nas pesquisas de ensino e aprendizagem de programação de computadores?

Conforme sinalizam as pesquisas de [7, 11, 12, 13, 15, 17, 20, 25, 26, 28, 30, 46, 53, 54, 60, 65, 71, 74 e 76] as dificuldades de aprendizagem em programação de computadores enfrentadas pelos alunos de disciplinas introdutórias não são incomuns, questões como interpretação e resolução de problemas, dificuldades para especificar as soluções, além dos relatos das dificuldades de conversão de abstrações conceituais em algoritmos funcionais, são frequentemente sinalizadas nos estudos investigados. Além disso, o modo de abordagem aplicado no processo de ensino/aprendizagem do conteúdo de programação é constantemente questionado por especialistas e/ou estudantes. Uma outra dificuldade frequentemente mapeada, diz respeito ao uso de linguagem de programação, que envolve a falta de compreensão da sintaxe da linguagem, na dificuldade para depuração e detecção de erros, além da falta de um acompanhamento adequado para o esclarecimento de dúvidas. Além dos pontos já destacados, outros trabalhos [11, 18, 22, 30, 40, 53, 54, 65 e 83] também identificam problemas que incluem os métodos aplicados na aprendizagem e a falta de motivação dos alunos (às vezes relacionada ao desempenho desta



aprendizagem) combinado com um processo avaliativo que normalmente é o causador de um alto índice de reprovação, e consequentemente, de evasão nesta disciplina.

4.2 - QP2: Quais soluções são oferecidas na literatura para atenuar as dificuldades no ensino e aprendizagem de programação?

A fim de organizar as informações sobre essas estratégias, o resultado desta análise foi categorizado com base nos indicadores de [Silva et al. 2015] que sugere o uso de uma lista, entretanto, não descarta a existência de outras classificações. Porém, nesta pesquisa, identificamos e aplicamos apenas parte das categorias sugeridas, conforme exibimos no Gráfico 2.

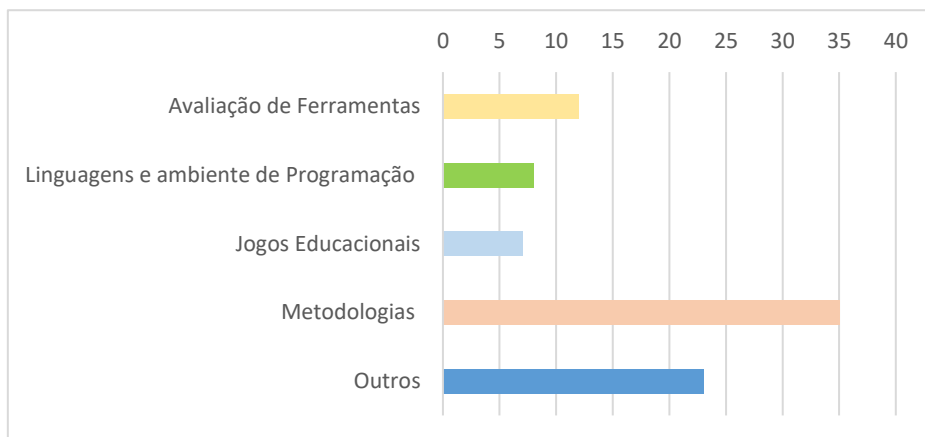


Gráfico 2. Tipos de categorias

Jogos Educacionais: A utilização de jogos digitais no ensino de programação para [69, 72, 74, 77 e 84] na maioria das vezes está relacionada com as estratégias de aprendizagem que visam melhorar a motivação, o desempenho e o raciocínio lógico dos alunos nas disciplinas introdutórias de programação. Além disso, essa abordagem lúdica quando aplicada em um cenário adequado, na visão de [14] pode favorecer o processo de aprendizagem, através de exemplos, analogias e atividades práticas que podem simplificar a percepção do aluno.

Metodologias: De acordo com os artigos de [1, 6, 7, 13, 17, 21, 27, 31, 32, 49, 55, 56, 65, 66, 69 e 70], diversas metodologias foram aplicadas objetivando minimizar as dificuldades enfrentadas pelos alunos na aprendizagem de programação. Essas abordagens descritas nestes trabalhos incluem desde a elaboração e utilização de técnicas que exploram uma combinação de práticas pedagógicas experimentais até implantação de métodos consolidados que agregam instrumentos como programação visual ou elementos tangíveis para motivar e melhorar o ensino de programação, além de manter o foco do aluno por mais tempo. Bem como, outros estudos [80, 82 e 83] destacam a importância da implementação de estratégias práticas e colaborativas que visam aprimorar o aprendizado dos alunos em disciplinas introdutórias de programação.

Linguagens e ambientes de programação: Nesta categoria, foram identificados os estudos [12, 15, 42, 44, 45, 51, 59 e 60] que empregam o uso de linguagem de programação para apoiar o ensino introdutório de programação. Já os artigos [14 e 61] combinaram a linguagem visual com o ambiente de programação formal. Entre os ambientes de programações visuais mais utilizadas para o ensino/aprendizagem de programação destaca-se a utilização da linguagem Scratch.

Avaliação da ferramenta: Essa categoria engloba uma variedade de estudos [10, 16, 30, 36, 41, 43, 56, 62, 76 e 78] que comparam diferentes suportes computacionais



utilizados no ensino de programação, nos quais, são medidos e analisados o desempenho dos alunos após experimentação de alguns instrumentos, muitas vezes experimentais. Em alguns estudos [46 e 64] são adotadas abordagens quantitativas ou qualitativas destes recursos para compreender melhor a percepção e a experiência dos estudantes dentro deste processo de aprendizagem.

Outros: Essa categoria abrange uma variedade de temas [4, 19, 26, 27, 34, 37, 39, 40, 66 e 85] que não apresentam uma classificação clara nesta proposta de estudo, mas podem conter indicadores importantes que não foram categorizados. Dentro desta perspectiva, esse estudo pode incluir tendências que incluem: robótica educacional, técnicas de avaliação, computação desplugada e outros. Entre os conteúdos sugeridos, destacamos a utilização de oficinas como um modo atrativo no processo de ensino-aprendizagem de algoritmos, assim como, a inserção de novas ferramentas e técnicas.

4.3 - QP3: Como as tecnologias digitais são utilizadas para apoiar aprendizagem de programação?

Diversas tecnologias digitais têm sido empregadas para apoiar a aprendizagem de programação, conforme apresentamos na Tabela 3. Essas tecnologias têm sido usadas de forma integrada e complementar na educação, proporcionando aos alunos uma variedade de recursos e ferramentas para aprender programação de maneira eficaz.

Tabela 3: Ambientes utilizados para apoiar a aprendizagem de programação.

Ambientes	ID [Artigos]
Ambiente Virtual de Aprendizagem	18, 28, 46, 58, 64, 73, 80, 83
Interface de Desenvolvimento	7, 15, 21, 30, 48, 52, 62, 71, 78, 79
Inteligência Artificial	11, 12, 36, 63, 77
Tutores Inteligentes	2, 40, 47, 49, 56
Sistemas de Recomendação	26, 44, 50, 59
Robótica Educacional	63, 85
Outros	[Todos os não incluídos acima]

Ambientes Virtuais de Aprendizagem têm sido empregados, como evidenciado nos estudos [18, 28 e 46], para proporcionar um ambiente interativo e colaborativo dentro da proposta metodológica. Os recursos oferecidos possibilitam que haja uma interação ativa e mediada entre os participantes. Já as Interfaces de Desenvolvimento são ferramentas fundamentais, conforme apresentado nos estudos [7, 15, 21 e 30] para oferecer aos alunos um espaço para escrever, compilar e depurar os seus códigos produzidos.

A Inteligência Artificial (IA) vem desempenhando um papel significativo como instrumento de apoio na aprendizagem, conforme demonstrado nos estudos [11, 12 e 36] onde são implementados e utilizados os tutores inteligentes que permitem personalizar o ensino com base no desempenho e nas necessidades individuais dos alunos. Além disso, a IA tem sido utilizada para analisar o comportamento dos alunos e prever seu desempenho futuro. Os Sistemas de Recomendação, presentes nos estudos [2, 26 e 44], têm sido utilizados para sugerir atividades e recursos de aprendizagem com base no progresso e nas preferências individuais. Essas recomendações ajudam os alunos a se manterem engajados e motivados durante o processo de aprendizagem. Além dos recursos citados, a Robótica Educacional, abordada nos estudos [63 e 85], tem sido empregada como uma ferramenta prática para ensinar programação, permitindo programar e controlar robôs para realizar tarefas específicas. Isso proporciona uma experiência tangível, que ajuda na melhor compreensão dos conceitos abstratos de programação.



4.4 - QP4: Quais abordagens pedagógicas foram mais utilizadas para apoiar o ensino/aprendizagem de programação?

Identificamos um panorama diversificado de abordagens pedagógicas utilizadas no ensino de programação. A classificação dos artigos nas diferentes categorias pedagógicas reflete a variedade de métodos empregados para essa aprendizagem, conforme exibido no Gráfico 3.

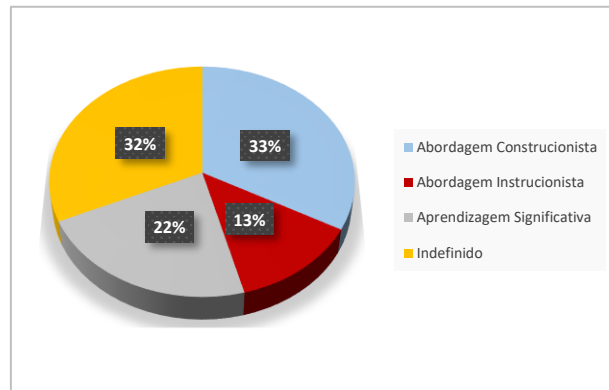


Gráfico 3. Abordagens pedagógicas

A partir do resultado obtido, nota-se que a abordagem Construcionista tem sido a mais utilizada pelos pesquisadores, representando um total de 33% do montante. Na sequência, a abordagem Significativa, com um percentual de 22%, sendo identificada com frequência na literatura pesquisada. Para nossa surpresa (autores), a utilização de abordagem Instrucionista apresenta um percentual elevado, com 13%. E, com 32% do total de artigos, apresentamos os artigos indefinidos, no qual, não identificamos o estilo de abordagem aplicado. É importante salientar que o enquadramento dos artigos foi realizado com base no entendimento dos parâmetros estabelecidos na Tabela 4.

Tabela 4. Estilo de abordagens pedagógicas

Abordagem Pedagógica	Descrição	ID [Artigos]
Aprendizagem Construtivista	Permite que o aluno participe e construa ativamente o próprio aprendizado, por meio de suas práticas colaborativas, das interações em grupo e dos momentos de reflexões.	1, 5, 9, 10, 12, 14, 15, 17, 18, 21, 23, 25, 26, 28, 31, 35, 37, 40, 45, 50, 62, 66, 70, 71, 75, 77, 78, 81, 83
Aprendizagem Significativa	Ocorre quando novos conhecimentos se conectam de maneira substantiva e não arbitrária aos conhecimentos prévios do aluno, facilitando a retenção e aplicação. Em outras palavras, aprende através do seu conhecimento prévio.	2, 7, 17, 30, 31, 35, 44, 61, 64, 72, 79,
Aprendizagem Instrucionista	Consiste na informatização dos métodos de ensino tradicionais, ou seja, por transmissão de ideias e realização de atividades gabaritais, entre outros.	6, 9, 19, 21, 28, 33, 41, 43, 46, 49, 50, 51, 69, 73, 76, 79, 80, 82, 84
Indefinidos	Não classificados pelos autores deste trabalho.	[Todos os não incluídos acima]

Cada abordagem pedagógica identificada e classificada têm sido utilizadas de diferentes maneiras para apoiar o ensino-aprendizagem de programação. Por exemplo, na



Abordagem Construcionista referenciadas por trabalhos como: [1, 10, 14, 24, 50, 62, 78, 81] e outros, concentra-se na ideia de que os alunos constroem seu próprio conhecimento através da exploração ativa de ideias, interações e de suas experiências participando e atuando como elemento central na construção do conhecimento. Esse método é bastante eficaz porque engaja os alunos em atividades que envolvem experiências colaborativas e cooperativas na resolução de problemas. Em contraponto, a Abordagem Instrucionista, identifica nos trabalhos de [2, 7, 17, 30, 44] e outros, baseia-se na instrução direta e na transmissão de conhecimento do professor para o aluno. Esta abordagem é estruturada e sequencial, com foco na entrega de conteúdo de forma sistemática.

Já na aprendizagem significativa, enfatiza-se a importância de relacionar novos conhecimentos com o que o aluno já sabe. Esta abordagem busca fazer conexões entre conceitos e experiências prévias dos estudantes, sendo eficaz porque possibilita que integrem novos conhecimentos de maneira mais profunda e relevante, facilitando a retenção e aplicação prática dos conceitos aprendidos.

Além das abordagens pedagógicas mapeadas nesta pesquisa, não identificamos em alguns artigos o tipo de abordagem pedagógica aplicada, sendo, portanto, classificadas na categoria de não identificadas. Além desta peculiaridade, destacamos o uso do método avaliativo da Taxonomia de Bloom, que embora não seja uma abordagem pedagógica, mas sim, método avaliativo, é mencionada em alguns artigos como um processo que pode ser utilizado para mapear e apoiar a personalização dos conteúdos, de acordo com os diferentes níveis de habilidades cognitivas, conforme apresentado nos trabalhos de [3, 13, 20, 34 e 55]. Assim, através deste tipo avaliação, é possível definir as estratégias visando promover as competências dos alunos em diferentes níveis, para uma aprendizagem progressiva e bem organizada.

5. Considerações Finais

Esse artigo apresentou uma Revisão Sistemática da Literatura sobre as dificuldades enfrentadas por estudantes iniciantes na aprendizagem de programação de computadores, com foco em artigos publicados entre 2019 e 2023. Através de uma análise criteriosa, foram selecionados 85 estudos, que identificam os principais obstáculos, bem como, as estratégias propostas para superá-los.

Entre as dificuldades mais comuns na aprendizagem de programação estão a compreensão de conceitos fundamentais, como a resolução de problemas, a aplicação de raciocínio lógico e a manipulação de linguagens de programação. Muitos estudantes enfrentam desafios significativos ao tentar entender e abstrair problemas, além da dificuldade de converter essas abstrações em códigos executáveis. A falta de motivação e o uso de métodos de ensino tradicionais e pouco interativos também são apontados como barreiras substanciais neste aprendizado.

Para mitigar essas dificuldades, diversas abordagens de ensino têm sido aplicadas, incluindo o uso de jogos educativos, que melhoram a motivação e estimulam o raciocínio dos alunos. As aplicações de linguagens de programação visuais, como o Scratch, facilitam a compreensão dos conceitos básicos como, o funcionamento da sequência de um algoritmo. Além disso, o uso de tecnologias digitais, como ambientes virtuais de aprendizagem, interfaces de desenvolvimento integrado, inteligência artificial e robótica, oferecem recursos valiosos para apoiar a aprendizagem de programação. O uso conjunto ou individual destas ferramentas permite uma interação mais prática, colaborativa e personalizada.



A partir dos dados obtidos, foi possível identificar uma diversidade de abordagens pedagógicas empregadas, como: construtivista, significativa, instrucionista e outras, que evidenciam que não há uma solução única. Cada contexto educacional pode se beneficiar de uma combinação específica dessas metodologias, ajustando-se às características e necessidades dos estudantes e dos currículos. A predominância da abordagem construtivista reflete uma tendência em promover um aprendizado mais ativo e centrado no aluno, valorizando a construção do conhecimento através da prática e da reflexão.

Por fim, os resultados desta revisão sugerem que, embora existam muitas dificuldades inerentes ao ensino e aprendizagem de programação, há também um vasto repertório de estratégias e ferramentas disponíveis para enfrentar o problema. A adoção de práticas pedagógicas mais dinâmicas e interativas, alinhadas aos interesses dos estudantes, bem como, o desenvolvimento contínuo de ferramentas de suporte, são essenciais para melhorar este cenário, consequentemente, a formação destes alunos. Esta revisão, ao fornecer uma visão abrangente das dificuldades e das soluções no ensino de programação, contribui para o avanço do conhecimento nesta área, além de abrir caminhos para futuras investigações e melhorias das práticas educacionais.

Referências

- BORGES, R.P.; OLIVEIRA, P.R.F.; LIMA, R.G.R.; LIMA, R.W. A Systematic Review of Literature on Methodologies, Practices, and Tools for Programming Teaching. *Ieee Latin America Transactions*. v. 16. n. 5. p.1468-1475, 2018.
- BAIST, Abdul; PAMUNGKAS, Aan Subhan. Analysis of student difficulties in computer programming. *VOLT: Jurnal Ilmiah Pendidikan Teknik Elektro*, v. 2, n. 2, p. 81-92, 2017.
- CARVALHO, Joethe; NETTO, José Francisco; ALMEIDA, Thais. Revisao sistemática de literatura sobre pensamento computacional por meio de objetos de aprendizagem. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. 2017.
- CHANG, Chih-Kai. Using computational thinking patterns to scaffold program design in introductory programming course. In: *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*. IEEE, 2016. p. 397-400.
- CHENG, Yih-Ping et al. Applying online content-based knowledge awareness and team learning to develop students' programming skills, reduce their anxiety, and regulate cognitive load in a cloud classroom. *Universal Access in the Information Society*, p. 1-16, 2022.
- DA CONCEIÇÃO Coelho, Marcos; GUEDES, Albertina Marilia Alves Aprendizagem baseada em problemas aplicada à programação de computadores: Um mapeamento sistemático. *Revista Novas Tecnologias na Educação (RENOTE)*, v. 18, n. 2, p. 570-580, 2020.
- FERRI, Juliana; DOS SANTOS ROSA, Selma. Como o ensino de programação de computadores pode contribuir com a construção de conhecimento na educação básica uma revisão sistemática da literatura. *Revista Novas Tecnologias na Educação (RENOTE)*, v. 14, n. 2, 2016.
- 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. *Revista Novas Tecnologias na Educação (RENOTE)*, v. 17, n. 1, p. 527-536, 2019.



- KITCHENHAM, B. Guidelines for performing Systematic Literature Reviews in Software Engineering, version 2.3. Technical Report EBSE. Software Engineering Group. School of Computer Science and Mathematics Keele University. 2007.
- NABI, Rebwar Mala. The Impact of Teaching Materials on Learning Computer Programming Languages in Kurdistan Region Universities and Institutes. vol.3, no. 1, 2018.
- SILVA, T. R.; MEDEIROS, T. J.; MEDEIROS, H.; LOPES, R.; ARANHA, E. Ensino-aprendizagem de programação: uma revisão sistemática da literatura. Revista Brasileira de Informática na Educação. v. 23, n. 1, 2015.
- SOUZA, C.M. et al. Lord of code: Uma ferramenta de apoio ao ensino da programação. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2016
- VALENTE, José Armando. Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. Revista E-curriculum, v. 14, n. 3, p. 864-897, 2016.
- ZANIN, Aline; SPARREMBERGER, Adalto Selau; BARBOSA, Jorge LV. Uma proposta de boas práticas para o ensino colaborativo de programação de computadores. Revista Novas Tecnologias na Educação, v. 17, n. 3, p. 41-50, 2019.