

ARTIGO DE PESQUISA/RESEARCH PAPER

Ferramentas educacionais para o ensino de programação no nível médio e em cursos técnicos: uma revisão integrativa

Educational Tools for Teaching Programming in High School and Technical Courses: An Integrative Review

Fransoar Araújo Silva [Instituto Federal do Ceará | fransoar.araujo.silva07@aluno.ifce.edu.br]
Maria Auxiliadora de Oliveira Chaves [Instituto Federal do Ceará | auxiliadora.chaves09@aluno.ifce.edu.br]
Adonias Caetano de Oliveira [Instituto Federal do Ceará | adonias.oliveira@ifce.edu.br]
Daniel Lima Sousa [Universidade Federal do Delta do Parnaíba | danielsousa@ufdpar.edu.br]
David de Miranda Rodrigues [Instituto Federal do Ceará | davidmr@ifce.edu.br]
Paulo César de Almeida Júnior [Instituto Federal do Ceará | paulo.almeida@ifce.edu.br]

Resumo. Ensinar programação em contextos educacionais ainda representa um desafio, marcado pela abstração conceitual, pela dificuldade de aplicar teoria na prática e pelo desinteresse dos estudantes. Habilidades como raciocínio lógico, resolução de problemas e pensamento computacional, muitas vezes pouco desenvolvidas nos anos iniciais, intensificam essas dificuldades. Esta revisão integrativa buscou identificar ferramentas e metodologias que favoreçam o ensino e a aprendizagem de algoritmos e programação, oferecendo subsídios para práticas pedagógicas mais eficazes. Foram analisados 1.024 estudos publicados entre 2019 e 2024, dos quais 20 atenderam aos critérios de seleção e apresentaram evidências relevantes. Os resultados indicam que estratégias como gamificação, plataformas interativas e jogos educacionais aumentam o engajamento e facilitam a aprendizagem. A robótica educacional também se destaca por associar programação a atividades práticas, promovendo desenvolvimento técnico, cognitivo e socioemocional. Além disso, linguagens em blocos, como Scratch e Blockly, favorecem a compreensão inicial dos conceitos ao reduzir a complexidade sintática e apoiar o foco na lógica algorítmica. Apesar dos avanços, limitações estruturais persistem, sobretudo em escolas com baixo acesso à tecnologia. Assim, políticas públicas, formação docente contínua e investimentos em infraestrutura são essenciais para ampliar a adoção dessas práticas e garantir um ensino inclusivo e de qualidade em algoritmos e programação.

Abstract. Teaching programming in educational contexts remains a challenge, characterised by conceptual abstraction, difficulties in applying theory to practice, and students' lack of interest. Skills such as logical reasoning, problem-solving, and computational thinking—often underdeveloped in the early years—intensify these obstacles. This integrative review aimed to identify tools and methodologies that support the teaching and learning of algorithms and programming, providing guidance for more effective pedagogical practices. A total of 1,024 studies published between 2019 and 2024 were analysed, of which 20 met the selection criteria and presented relevant evidence. The findings indicate that strategies such as gamification, interactive platforms, and educational games enhance engagement and facilitate learning. Educational robotics also stands out for linking programming to practical activities, fostering technical, cognitive, and socio-emotional development. Furthermore, block-based languages such as Scratch and Blockly ease the initial understanding of concepts by reducing syntactic complexity and enabling students to focus on algorithmic logic. Despite these advances, structural limitations persist, particularly in schools with limited access to technology. Therefore, public policies, continuous teacher training, and investment in infrastructure are essential to expand the adoption of these practices and ensure inclusive, high-quality teaching in algorithms and programming.

Palavras-chave: Ensino de programação, Gamificação, Robótica educacional, Programação em blocos

Keywords: Programming education, Gamification, Educational robotics, Block-based programming

Recebido/Received: 26 December 2025 • **Aceito/Accepted:** 23 February 2026 • **Publicado/Published:** 27 February 2026

1 Introdução

Os avanços e tendências tecnológicas constantes na área de computação, especialmente nas décadas de 2010 e 2020, tornaram o ensino de programação uma parte essencial no aprendizado dos alunos, desempenhando um papel fundamental na preparação de indivíduos para uma participação ativa e crítica na sociedade digital [dos Anjos *et al.*, 2024]. Garlet *et al.* [2018] ressaltam que nem todos os estudantes se tornarão programadores, mas possuirão vantagens ao aprender sobre lógica de programação. O ensino de programação e informática permite aos estudantes aprimorar suas competências e habilidades que permitirão identificar e resolver problemas

encontrados no cotidiano das pessoas de maneira mais criativa e eficiente [Agbo *et al.*, 2019; Oro *et al.*, 2015].

Apesar do fato de que o ensino de programação está se tornando cada vez mais importante, muitos discentes ainda possuem dificuldades significativas em aprender, especificamente sobre conceitos abstratos e lógicas complexas. Além das dificuldades para aplicar estratégias nos problemas que nem sempre são intuitivos para todos, o que pode resultar em frustração e desmotivação [Paiva *et al.*, 2020]. Por estes motivos, o índice de desistência nessas disciplinas é consideravelmente elevado comparado às demais.

Conforme dados de 2021 da plataforma Nilo Peçanha¹, a taxa média de evasão nos cursos técnicos é de aproximadamente 13%. Número que tende a aumentar nos cursos técnicos relacionados à área de informática, que podem ultrapassar a taxa de 20% em algumas instituições da região nordeste [SETEC/MEC, 2022; Sousa *et al.*, 2022]. O mapa do ensino superior de 2023 demonstrou que os cursos da área de Tecnologia da Informação (TI) no Brasil apresentaram uma maior taxa de evasão comparada às demais áreas de ensino. A taxa de evasão foi de 38,5% no ano de 2021, sendo 7,8% maior que outras áreas de ensino [Instituto Semesp, 2023].

No ensino técnico, o aprendizado de programação enfrenta diversos obstáculos pelo fato de o currículo ser frequentemente centrado em habilidades práticas e aplicadas. Portanto, o aprendizado sobre conceitos abstratos e lógicas complexas acaba se tornando o principal desafio para esses estudantes [Vieira *et al.*, 2015; Silva *et al.*, 2018]. A falta de conhecimento prévio em áreas do ensino básico e da própria utilização do computador, além de dificuldades em relacionar teoria com prática e desenvolver raciocínio lógico, são algumas das dificuldades nas atividades de lógica de programação [Viana and dos Santos Souza, 2024]. Outros motivos também podem ser, base precária de matemática e suas lógicas, falta de dedicação dos estudantes e falta de materiais didáticos de apoio [de Farias *et al.*, 2018].

Existem muitas ferramentas de apoio disponíveis para o ensino e aprendizado de programação, principalmente para auxiliar os estudantes a compreender os conceitos abstratos e lógicos. Scratch Scratch [2024] e MIT App Inventor Inventor [2012] utilizam o conceito de programação em bloco para ensinar principalmente os estudantes iniciantes sobre lógica de programação de uma forma visual e intuitiva. Plataformas de robótica educacional como Arduino e LEGO Mindstorms que combinam a programação com a criação de projetos interativos ao combinar *software* e *hardware*, permitindo que os discentes possam aprender de forma prática e possam praticar sobre resolução de problemas complexos [Pan and Zhu, 2018; Afari and Khine, 2017]. O uso de ferramentas gamificadas como CodeCombat e Robocode tem se tornado abordagens cada vez mais utilizadas, por incorporarem elementos de jogos para tornar o aprendizado mais atraente e motivador, por ser possível aprender enquanto se diverte [CodeCombat, 2024; Larsen, 2021].

A revisão das ferramentas utilizadas para apoiar o ensino de programação é fundamental para a comunidade acadêmica. Possibilitam um entendimento mais aprofundado de quais ferramentas são mais eficientes nos diferentes níveis de ensino, auxiliando os professores a adaptarem suas estratégias às demandas de seus estudantes [Rovshenov and Sarsar, 2023]. Pesquisas envolvendo diversos grupos de usuários podem gerar resultados comparativos que permitem identificar nuances entre os diversos estudantes [Asgari *et al.*, 2024]. Ao realizar esse tipo de pesquisa, é possível coletar e identificar diferentes métodos e tecnologias, tornando o ensino mais flexível, se adaptando melhor para cada tipo de grupo. Assim, os estudantes não apenas memorizam conceitos, mas também os aplicam de maneira prática, facilitando a compreensão e

reduzindo o índice de desistências nos cursos de computação e áreas afins [José Richter *et al.*, 2019].

Mediante uma revisão integrativa acerca das ferramentas educacionais voltadas ao ensino de programação nesses níveis de ensino, este estudo parte da seguinte questão central: quais ferramentas e abordagens pedagógicas têm se mostrado eficazes no ensino de programação para estudantes do ensino médio e técnico? A partir dessa pergunta, busca-se identificar e sistematizar os principais métodos, tecnologias e ferramentas empregadas em sala de aula para apoiar estudantes e docentes no processo de ensino-aprendizagem, bem como discutir seus benefícios e limitações na prática educacional.

Este artigo está organizado em seis seções. Na seção 2, são descritos os trabalhos relacionados, oferecendo uma visão geral sobre cada um. Na seção 3, detalham-se os procedimentos metodológicos adotados. Na seção 4, são apresentados os resultados da revisão juntamente com as discussões dos achados, destacando as ferramentas encontradas nos estudos. Por fim, na seção 5, são expostas as conclusões desta revisão integrativa.

2 Trabalhos Relacionados

Nesta seção, realizou-se a seleção de revisões bibliográficas relacionadas ao tema, com o objetivo de contextualizar e fundamentar a pesquisa. Essa busca é independente da metodologia principal desta revisão, a qual será apresentada na seção 3. A seleção dos estudos secundários foi realizada por meio do repositório *Google Acadêmico* utilizando a string de busca em português: (“*programação*” OR “*computação*”) AND (“*educação*”) AND (“*ferramentas*” OR “*metodologia*”), baseada na leitura dos títulos e resumos, priorizando aqueles que apresentavam maior relevância ao tema investigado. Foram consideradas apenas estudos secundários no idioma português Brasil e publicados entre 2019 à 2024, garantindo a inclusão de pesquisas recentes, alinhadas às novas tendências, ferramentas e metodologias, especialmente em áreas como tecnologia, onde as práticas e teorias podem mudar significativamente em curtos espaços de tempo [Snyder, 2019].

Eugenio F. Pereira *et al.* [2021] conduziram um Mapeamento Sistemático da Literatura (MSL) sobre ferramentas de Apoio ao Ensino Introdutório de Programação. O objetivo dessa MSL foi identificar ferramentas de apoio ao ensino introdutório de programação. Para isso, foram utilizadas seis diferentes bibliotecas digitais para o estudo, resultando em 22 artigos relevantes. Em seguida, as ferramentas educacionais catalogadas foram categorizadas em nove categorias distintas, são elas: (*Ambiente de programação visual e drag&drop*, *robôs para programação*, *ambiente de programação móvel*, *ambiente móvel de programação*, *correção e avaliação de atividades/feedback*, *ambiente de programação com uso de Realidade Aumentada*, *ambiente de programação baseado em jogos*, *ambiente de programação com simulação 3D*, e *ambiente de programação baseado em nuvem*), visando facilitar o processo de ensino e aprendizagem. Chegando à conclusão de que a variedade de ferramentas disponíveis pode contribuir significativamente para o ensino introdutório de programação. Essas ferramentas oferecem suporte aos educadores e promovem maior engajamento dos alunos, especialmente em cursos de nível superior e técnico.

¹A Plataforma Nilo Peçanha é um ambiente virtual de coleta, validação e disseminação das estatísticas oficiais da Rede Federal de Educação Profissional. Disponível em <http://plataformanilopecanha.mec.gov.br/>.

José Richter *et al.* [2019] realizaram uma Revisão Sistemática da Literatura (RSL) para investigar as estratégias e tecnologias educacionais utilizadas no ensino de programação, enfatizando a Programação Orientada a Objetos. Essa revisão abrangeu diferentes níveis de ensino, indo do fundamental até o superior. Para esse trabalho, foram analisados 136 estudos selecionados pelos critérios de seleção final entre 2010 e 2018. Os *softwares* mais citados foram Alice e Scratch. As abordagens foram oficinas, aprendizagem baseada em problemas e em projetos. Concluindo, a importância de estratégias que aproximem os alunos da realidade. Essas estratégias promovem a autonomia e abordam a complexidade da programação de maneira lúdica e acessível. Alcançando assim o objetivo de facilitar a compreensão e, ao mesmo tempo, de reduzir o índice de desistências nos cursos de computação.

Na revisão sistemática da literatura conduzida por Moraes *et al.* [2022] foram revisadas as estratégias de ensino-aprendizagem de programação introdutória em cursos de ensino superior no Brasil. Os autores utilizaram como fontes de busca eventos nacionais na área de Informática na Educação para a seleção dos trabalhos no período entre 2014 e 2018. Ao final da seleção, foram considerados 51 artigos, com os quais foi possível traçar e identificar as principais técnicas, metodologias e ferramentas adotadas por esses autores. Os achados da revisão sistemática da literatura revelaram uma diversidade de ferramentas e metodologias que auxiliam o ensino de programação, utilizando abordagens baseadas em metodologias ativas, como a gamificação e a aprendizagem baseada em problemas. Além disso, foram identificadas ferramentas de programação visual, como o Scratch, App Inventor e Helpblock, além de linguagens adaptadas, como o PortuCol [Barbosa *et al.*, 2016]. No entanto, desafios foram encontrados, como a desmotivação dos estudantes, a ausência de mulheres nessa área e a necessidade de avaliações mais abrangentes.

Em contraste, este trabalho visa traçar um panorama atual sobre diferentes tipos de ferramentas e metodologias educacionais que podem auxiliar os estudantes em seu aprendizado de programação, enfatizando os ensinamentos médio e técnico. Para este estudo, foram escolhidos os repositórios digitais *ACM Digital Library*, *IEEE Xplore* e *SBC-OpenLib*, sendo realizada a busca e análise em dois idiomas, sendo eles português e inglês. Além disso, estão presentes no recorte temporal apenas estudos mais recentes entre 2019 e 2024 para esta revisão integrativa.

3 Procedimentos Metodológicos

A escolha de conduzir uma revisão integrativa se deu por se tratar de um método capaz de sintetizar conhecimentos já construídos por estudos anteriores sobre determinados temas investigados. Possibilitando integrar conhecimentos de diversos estudos já publicados, permitindo que novos conhecimentos surjam a partir da análise dos dados obtidos, potencializando a construção da ciência [Botelho *et al.*, 2011]. Tanto a análise quanto a síntese dos dados estudos foram realizadas de forma descritiva, permitindo observar, quantificar, detalhar e categorizar as informações, visando consolidar o conhecimento existente sobre o tema investigado na revisão.

Para a elaboração desta revisão integrativa, visando quan-

tificar e analisar os estudos relacionados com o tema abordado neste trabalho, foram consideradas as seis etapas definidas pelos Botelho *et al.* [2011], que, segundo os próprios autores, o processo de criação de uma revisão integrativa deve seguir uma sucessão de etapas bem definidas, como mostrado na Figura 1. Inicialmente, na etapa 1 é necessário identificar qual será o tema e a pergunta de pesquisa. Posteriormente, são definidos os critérios de inclusão e exclusão necessários para a seleção dos estudos. Após selecionar os estudos escolhidos por meio dos critérios, é possível classificá-los conforme suas abordagens metodológicas, objetivos e contribuições para a área. Logo em seguida, os estudos selecionados anteriormente passarão pelo processo de interpretação que determinará as tendências científicas da área. Por fim, os dados obtidos são sintetizados e apresentados.

Em relação à Figura 1, inicialmente na etapa 1 é necessário identificar qual será o tema e a pergunta de pesquisa. Posteriormente, são definidos os critérios de inclusão e exclusão necessários para a seleção dos estudos. Após selecionar os estudos escolhidos por meio dos critérios, é possível classificá-los conforme suas abordagens metodológicas, objetivos e contribuições para a área. Logo em seguida, os estudos selecionados anteriormente passarão pelo processo de interpretação que determinará as tendências científicas da área. Por fim, os dados obtidos são sintetizados e apresentados.

3.1 Definição do protocolo de revisão

Nesta seção, são apresentados os procedimentos adotados para a condução da revisão integrativa. As subseções a seguir detalham os aspectos metodológicos da pesquisa, como as questões de pesquisa que orientam o estudo 3.1.1, os critérios de inclusão e exclusão para a análise dos estudos 3.1.2, o processo de seleção dos estudos 3.1.3 e a categorização desses estudos revisados 3.1.4. Vale ressaltar que esta revisão integrativa foi fruto de trabalho de graduação, sendo conduzida durante os meses de abril até dezembro de 2024, antes da compilação dos resultados em manuscrito. Por isso, os estudos de 2025 não foram incluídos no filtro desta pesquisa.

3.1.1 Questões de Pesquisa

Esta revisão integrativa tem como principal objetivo identificar e analisar as principais ferramentas e metodologias utilizadas para o propósito de apoiar o ensino e aprendizagem de disciplinas de programação no ensino médio e técnico. A partir dessa análise, foi definida a Questão de Pesquisa Principal (QPP): Quais as principais ferramentas e metodologias utilizadas no ensino e aprendizagem de disciplinas de programação no ensino médio e técnico?

A partir da QPP, foram derivadas três subquestões de pesquisa (QP), que buscam aprofundar aspectos específicos do tema abordado:

- a) QP1: Quais ferramentas ou tecnologias são utilizadas no ensino de programação?
- b) QP2: quais são seus benefícios e limitações?
- c) QP3: Quais as abordagens pedagógicas mais utilizadas no ensino de programação utilizando as ferramentas?

3.1.2 Critérios de inclusão e exclusão

Seguindo o processo apresentado na Figura 1, a próxima etapa a ser seguida é o estabelecimento dos critérios de inclusão



Figura 1. Processo de revisão integrativa. Adaptado de Botelho *et al.* [2011]

e exclusão. Esses critérios possuem o propósito de incluir ou eliminar estudos que não atendem aos requisitos ou que podem introduzir viés na revisão. Com o decorrer do processo, a seleção de estudos sofre afunilamentos à medida que o pesquisador avança em sua pesquisa. Dessa maneira, na Tabela 1 são apresentados todos os critérios de inclusão e exclusão utilizados para a seleção dos estudos presentes nesta revisão integrativa.

Tabela 1. Critérios de inclusão e exclusão

Critérios de Inclusão
1. Ano de publicação entre 2019 e 2024
2. Estudos primários
3. Escritos em inglês ou português
4. Estudos que propõem ou utilizam ferramentas / metodologias pedagógicas
Critérios de exclusão
1. Artigos sem resumo
2. Estudos que não mencionam ferramentas educacionais
3. Estudos secundários ou terciários
4. Literatura cinzenta
5. Não aplicados ou avaliados em turmas do ensino médio ou técnico
6. Artigos inacessíveis na íntegra

3.1.3 Seleção dos Estudos

Na terceira etapa desta revisão, foi realizada a seleção dos estudos identificados a partir da busca utilizando a string de

pesquisa nos repositórios previamente definidos na primeira etapa deste estudo. Para isso, foram utilizadas as ferramentas Parsifal¹ e Rayyan² para apoiar o processo desta revisão integrativa. A seleção foi feita por meio da leitura dos títulos, resumos e palavras-chave de todas as publicações encontradas, a fim de confirmar a conformidade com os critérios de inclusão definidos para a pesquisa [Botelho *et al.*, 2011].

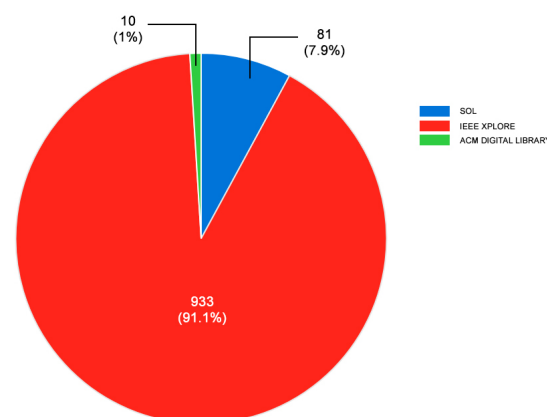


Figura 2. Documentos encontrados por repositórios

Conforme ilustrado pela Figura 2, foram encontradas 1024 publicações em três repositórios distintos, são eles: IEEE Xplore, SBC-OpenLib e ACM Digital Library. Utilizou-

¹Disponível em <https://parsif.al/>

²Disponível em <https://www.rayyan.ai/>

Tabela 2. Documentos excluídos na segunda etapa de seleção.

Critério de Exclusão	Motivo	Qtd
2	Não citavam ferramentas ou métodos de ensino	2
3	Eram estudos secundários ou terciários	2
4	Tratavam-se de literatura cinzenta	6
5	Não aplicados ou avaliados no ensino médio/técnico	4
6	Não foi possível acessar o conteúdo completo	17

se um filtro de data, abrangendo o período de 2019 a 2024, o recorte temporal prioriza estudos recentes e tecnologicamente atualizados sobre o ensino de programação. Após a pré-seleção (leitura dos títulos e resumos), para garantir maior rigor na seleção, aplicando os critérios de inclusão e exclusão, esse número foi reduzido para 52 estudos, que seguiram para a próxima etapa de seleção, para a leitura na íntegra de seus conteúdos. A Tabela 2 a seguir apresenta o resumo dos motivos de exclusão dos 32 estudos na segunda etapa de seleção, ocorrendo quando pelo menos um dos critérios fosse atendido, garantindo que apenas os estudos alinhados ao escopo da pesquisa fossem considerados.

Após a conclusão desta etapa, foi elaborada uma tabela com os artigos selecionados, que serviu como base para as próximas fases da revisão. O diagrama PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*) apresentado na Figura 3 ilustra o processo de inclusão e exclusão dos artigos. Ele descreve o fluxo de informações obtidas por meio de diferentes fases de uma revisão sistemática, mapeando o número de estudos identificados, incluídos e excluídos, além dos motivos das exclusões [MJ, 2020].

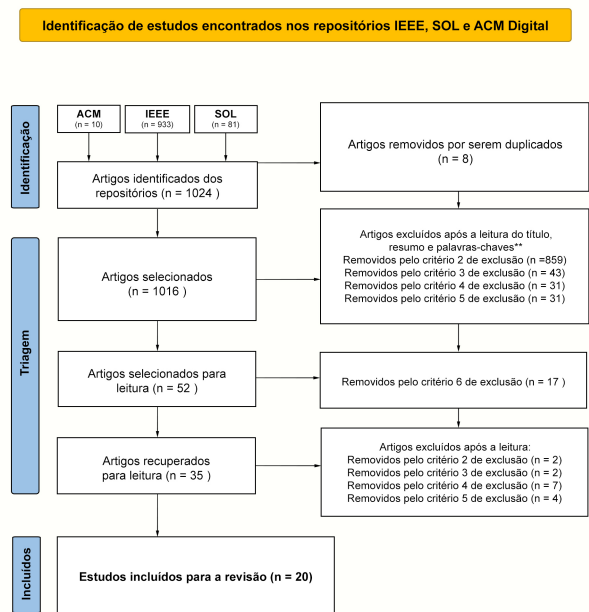


Figura 3. Diagrama de fluxo PRISMA

3.1.4 Categorização dos Estudos

Esta etapa visa sintetizar as informações, que foram obtidas a partir dos estudos analisados nas fases anteriores. Logo, para a categorização das informações desta revisão integrativa, foram consideradas relevantes as informações extraídas das seguintes questões: título, autores, data de publicação, ferramentas educacionais utilizadas, metodologias utilizadas, quais benefícios essas abordagens obtiveram, as limitações da abordagem e quais foram os resultados obtidos na aprendizagem.

4 Resultados e Discussão

A Tabela 3 apresenta os 20 estudos que passaram por todos os métodos de inclusão e exclusão estabelecidos, com seus respectivos títulos e identificadores (ID, do inglês *Identify*).

4.1 Síntese de dados

Na Tabela 4 é possível visualizar detalhadamente o resumo dos dados de cada trabalho analisado neste estudo. Os trabalhos foram cuidadosamente categorizados com base em diversos aspectos, incluindo o público-alvo, o número de participantes envolvidos, a metodologia de avaliação empregada e os principais resultados observados em cada caso.

Verificou-se que 59,09% dos trabalhos foram aplicados ao ensino médio. O ensino médio integrado ao ensino técnico foi abordado em 27,27% dos trabalhos, enquanto os que abordavam apenas o ensino técnico, representam cerca de 13,64%. Na Tabela 4 estão detalhados os níveis de escolaridade que cada estudo aborda, conforme mencionado pelos próprios autores dos estudos.

4.2 Respostas às questões de pesquisa

Nas próximas subseções, são respondidas as questões de pesquisa deste estudo descritas na subseção 3.1.1. Cada questão foi analisada com base nos dados coletados e sintetizados a partir dos estudos selecionados. Apresentando as ferramentas, metodologias, estratégias e abordagens pedagógicas identificadas, bem como seus benefícios e limitações, destacando os desafios encontrados pelos educadores e alunos.

4.2.1 Resposta à QP1

Para responder à QP1, foram identificadas e sistematizadas todas as ferramentas educacionais cujo propósito é apoiar e facilitar o ensino e a aprendizagem de programação de computadores no contexto do ensino médio e técnico, conforme os estudos selecionados nesta revisão integrativa.

A Tabela 5 apresenta o conjunto de ferramentas encontradas, bem como as publicações nas quais foram reportadas. Observa-se uma ampla diversidade de soluções, incluindo ambientes de programação visual, plataformas educacionais online, ferramentas para robótica educacional e dispositivos físicos programáveis, evidenciando a multiplicidade de abordagens adotadas para o ensino de programação nesses níveis de ensino.

De modo geral, verificou-se que 19 dos 20 estudos analisados utilizaram uma ou mais ferramentas tecnológicas como suporte às práticas pedagógicas. Apenas o estudo A1 não fez uso direto de ferramentas computacionais, concentrando-se exclusivamente em metodologias e estratégias didáticas. Esse resultado reforça o papel central das ferramentas educacionais

Tabela 3. Estudos selecionados.

ID	Título	Ano	Local
A1	The Brazilian School Computing Standard - Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1	2023	ACM
A2	Meninas.comp Project: Programming for Girls in High School in Brazil	2020	IEEE
A3	Teaching Computational Thinking to a Student with Attention Deficit Through Programming	2020	IEEE
A4	Robotics Teaching and Tools to Promote Better Didactic in Public High-Schools	2019	IEEE
A5	Teaching CT through Internet of Things in High School: Possibilities and Reflections	2020	IEEE
A6	Arduino Baby Project: Robotics as a Tool to Support Permanence and Success of Technical Course Students in Informatics	2021	IEEE
A7	Codify: Uma plataforma gamificada para auxiliar e motivar estudantes no ensino de programação	2022	SOL
A8	Ensino Remoto de Programação em Java para Estudantes do Ensino Técnico Utilizando Tablets: Um Relato de Experiência	2023	SOL
A9	Pensamento Computacional e Scratch: Um relato de Experiências com Estudantes do Ensino Médio Público no Distrito Federal	2023	SOL
A10	Ensinando Pensamento Computacional para Alunas de Disciplinas Introdutórias de Programação no Ensino Técnico através de um MOOC	2023	SOL
A11	Ensino de Programação para Estudantes do Ensino Médio de Escolas Públicas com Igualdade de Gênero	2023	SOL
A12	Desenvolvimento de um Método Para Impulsionar os Estudos de Programação Competitiva no Ensino Médio e Fundamental	2023	SOL
A13	Analyzing the Effectiveness of an Educational Process for Teaching Programming Through Educational Robotics in a Brazilian Technical and Vocational High School	2023	SOL
A14	Poké Prog: Jogo educacional para o ensino dos conceitos básicos de programação	2023	SOL
A15	Fostering Programming Logic Skills in High School Students Through Project-Based Learning: An Educational "*****"	2024	SOL
A16	Abordagem Participativa de Desenvolvimento de Jogos Digitais Educacionais no Contexto Escolar	2019	SOL
A17	Coletânea de uma Década de Ensino de Programação para Estudantes da Rede Pública no Projeto Introcomp	2021	SOL
A18	Uma experiência em ensino de programação de computadores com alunos da rede pública usando a linguagem Scratch	2019	SOL
A19	Um Estudo Exploratório Sobre o Uso Da Robótica Educacional Como Ferramenta de Apoio ao Ensino-Aprendizagem de Lógica de Programação para Alunos da Rede Pública do Ensino Médio	2021	SOL
A20	Investigating the Teaching of Block Programming in High School	2022	SOL

como mediadoras do processo de ensino e aprendizagem de programação no ensino médio e técnico.

No que se refere às contribuições desta revisão integrativa, os achados indicam que, mais do que a proposição de ferramentas totalmente inéditas, a literatura recente tende a consolidar e reaproveitar soluções já estabelecidas, como *Scratch*, *Arduino*, *MIT App Inventor* e *Micro:bit*, frequentemente adaptadas a diferentes contextos educacionais. Além disso, a análise evidenciou o uso recorrente de ambientes baseados em programação visual e robótica educacional, sugerindo uma preferência por abordagens que reduzam a complexidade inicial da aprendizagem e favoreçam o engajamento dos estudantes.

Assim, esta revisão contribui ao oferecer uma visão consolidada das principais ferramentas utilizadas no ensino de programação no ensino médio e técnico, auxiliando pesquisadores e educadores na identificação de soluções amplamente adotadas, bem como de tendências tecnológicas predominantes nesse domínio.

É importante destacar que a metodologia mais utilizada se trata da aprendizagem baseada em projetos, que propõe que os estudantes desenvolvam projetos, permitindo aplicar o que aprenderam na sala de aula em um contexto prático e real. Assim, possibilitando que desenvolvam as habilidades e competências necessárias para o século XXI, como trabalho em equipe, resolução de problemas, pensamento crítico e criativo [Oliveira et al., 2020]. Na Tabela 6 é possível vi-

sualizar as metodologias e estratégias encontradas durante a análise dos estudos que visam facilitar o processo de ensino-aprendizagem.

A Tabela 7 apresenta a classificação dos diferentes cursos encontrados neste estudo, juntamente com a identificação dos documentos associados. Os cursos analisados abrangem diferentes áreas de formação, incluindo agricultura, desenvolvimento de jogos digitais, informática e programação para jogos e aplicativos, além do ensino médio regular, contribuindo para uma visão abrangente das iniciativas mapeadas. Essa categorização é fundamental para compreender como as diferentes áreas do ensino médio e técnico estão sendo exploradas.

A partir dessa classificação, foi possível avançar na análise das tecnologias empregadas em cada contexto educacional. Em particular, identificaram-se as linguagens de programação mais recorrentes nos estudos revisados, cuja distribuição é apresentada na Tabela 8.

4.2.2 Resposta à QP2

Com a análise qualitativa dos estudos selecionados, foi possível observar diversos benefícios e limitações no ensino e aprendizagem de programação em diferentes contextos. Entre as principais vantagens, nos estudos [A1], [A3], [A4], [A5], [A7], [A8], [A9], [A10], [A12], [A13], [A14], [A16], [A17], [A18], [A19] e [A20] podemos destacar o aumento significativo do engajamento e motivação dos estudantes, ocasionado

Tabela 4. Resumo dos dados de cada trabalho analisado.

ID	Público	Participantes	Total	Método de Avaliação	Resultados
A1	Ensino Médio	-	-	Ciclo de feedback.	Melhor compreensão de conceitos; Desenvolvimento de habilidades.
A2	Ensino Médio	Alunas 1º- 3º Ano	-	Feedback das Participantes.	Aumento da participação feminina em engenharia da computação.
A3	Ensino Médio	Aluno com TDA	1	Acompanhamento estruturado.	Aprimoramento da Autonomia; Melhora no aprendizado.
A4	Ensino Médio	15 a 19 Anos	22 alunos	Questionários; Entrevistas; Observações.	Desenvolvimento de habilidades: resolução de problemas, pensamento criativo e colaboração.
A5	Ensino Técnico	14 a 17 anos	40 alunos.	Questionários; observação.	Desenvolvimento de habilidade: comunicação, colaboração, pensamento crítico e criatividade.
A6	Ensino Técnico	1º ano	22 alunos.	Entrevistas; Questionários.	Desenvolvimento de habilidades: Melhora do trabalho em grupo.
A7	Ensino Técnico Integrado	1º ano	51 alunos.	Feedback; Questionário.	Engajamento e motivação; Evolução de habilidades em programação.
A8	Ensino Técnico Integrado	1º e 2º Ano	82 alunos.	Feedback; Questionário.	Melhora na produtividade; Melhora na comunicação, colaboração e motivação.
A9	Ensino Médio	1º e 2º Ano	38 alunos.	Questionários; Observações.	Melhora no engajamento Melhora no desempenho e em suas Habilidades.
A10	Ensino Técnico	Alunas 18 a 30 Anos	18 alunos.	Modelo ARCS; Questionário; Observação.	Aumento de motivação das alunas; Alcançou o objetivo do curso.
A11	Ensino Médio	4 Alunas 4 Alunos	8 alunos.	Avaliação; Questionário Lund.	Melhora no Raciocínio lógico e resolução de problemas; Igualdade de gênero e social.
A12	Ensino Médio	4 Alunas 4 Alunos	8 alunos.	Design-Based Research.	Melhora de desempenho dos estudantes.
A13	Ensino Técnico Integrado	-	-	Aplicação da Intervenção; Análise de impacto.	Desenvolvimento de habilidade: Pensamento computacional.
A14	Ensino Médio	76 Alunos	76 alunos.	Teste de usabilidade; Observação; Questionário.	Melhora na aprendizagem em conteúdos básicos de programação.
A15	Ensino Médio	3 Alunas 7 Alunos.	13 Alunos.	Questionário; Entrevistas.	Melhora nas habilidades; Interesse em continuar os estudos.
A16	Ensino Técnico Integrado	20 alunos.	20 alunos.	Entrevistas; Observações.	Melhora no engajamento; Desenvolvimento de habilidades práticas.
A17	Ensino Médio	1º ao 3º ano	4 mil alunos.	Avaliações; Feedback contínuo.	54% dos estudantes indicaram interesse em seguir na área.
A18	Ensino Médio	8 Alunas 21 Alunos.	30 alunos.	Questionários estruturados.	Melhora no pensamento lógico; Melhora no engajamento.
A19	Ensino Técnico Integrado	1º ano	44 alunos.	Sistema de notas; Questionários.	Desenvolvimento de habilidades fundamentais e essenciais.
A20	Ensino Técnico Integrado	Educadores	36 educadores.	Questionários.	Caracterização dos Docentes; Ferramentas catalogadas.

por metodologias ativas, gamificadas e abordagens práticas. Projetos que utilizam robótica educacional, jogos educativos e plataformas interativas [A2], [A4], [A5], [A6], [A7], [A13], [A14], [A16], [A18] e [A20] demonstraram ser eficazes para o desenvolvimento de habilidades técnicas, resolução de problemas, pensamento criativo e trabalho em equipe.

Além disso, iniciativas que buscam a inclusão, como as direcionadas para mulheres em [A2], [A10] e [A11] ou para pessoas com déficit de atenção em [A3], demonstraram-se promissoras para ampliar a participação e diversidade na área de computação. Também o uso de ferramentas de programação visual em [A18] e [A20] demonstrou ser eficiente para o ensino de iniciantes, que possuem pouco ou nenhum conhecimento sobre programação, enquanto o ensino remoto e

aprendizagem autônoma em [7], [A8], [A12], [A15] e [A17] mostrou ser eficaz por sua flexibilidade e efetividade. Além disso, estudos de cursos e projetos, como os de [A2], [A6], [A9], [A10], [A11], [A12], [A13], [A15] e [A17] demonstraram impactos positivos em relação ao aumento do interesse e até mesmo de continuar estudando e seguir carreira nesta área de computação.

No entanto, limitações significativas também foram identificadas, sendo a mais relatada a falta de infraestrutura e recursos tecnológicos adequados mencionados nos estudos [A1], [A2], [A5], [A6], [A8], [A9], [A12], [A15], [A16], [A19] e [A20] e barreiras socioeconômicas dos estudantes em [A6], [A7], [A8] e [A9], dificultam ou impedem que certas metodologias sejam utilizadas. A falta de formação adequada para

Tabela 5. Ferramentas educacionais.

Ferramentas	Publicações	Ferramentas	Publicações
AgentsCubes	A20	Micro: bit	A17; A20
Alice	A20	MIT App Inventor	A2; A20
Arduino	A2; A4; A6; A13	Modle	A10; A15
Blockuino	A20	Neps Academy	A12
Blockly	A20	NetsBlox	A20
ChoiCo	A20	NetTango	A20
CodCad	A12	OzoBlockly	A20
Code.org	A18; A20	PencilCode	A20
Codify	A7	Poké Prog	A14
Csnap	A20	Powtoon	A10
DeviceMakAR	A20	Programaê	A18
Dr. Scratch	A9	Replit	A8
DuinoBlocksII	A19	Roblock	A20
Engage	A20	Robocore	A19
H5P	A10	Run.codes	A17
Inkscape	A16	S4A	A20
IntelliBox	A20	Scratch	A9; A18; A20
Jupyter	A17	Snap!	A20
Kodetu	A20	SparkiDuino	A4
Kodu	A2	Stencyl	A16
Lego	A20	Tickle	A20
mBlock	A20		

Tabela 6. Metodologias pedagógicas encontradas.

Metodologias	Publicações
Aprendizagem Baseada em Projetos	A1; A2; A3; A4; A5; A6; A8; A13; A15; A16; A17; A18; A19; A20
Aprendizagem Baseada em Problemas	A1; A3; A4; A8; A9; A14; A17; A18; A19; A20
Oficinas	A2; A5; A6; A9; A11; A13; A16; A17; A20
Metodologias Ativas	A3; A4; A8; A9; A10; A14; A18; A19
Gamificação	A2; A7; A13; A14; A16; A17; A20
Competições	A11; A12; A16; A20
Aprendizagem Colaborativa	A5; A8; A18
Atividades Desplgadas:	A1; A18
Design-Based Research (DBR)	A11; A12
Interdisciplinaridade	A6
Sala de Aula Invertida	A15

Tabela 7. Tipos de cursos.

Cursos	Publicações
Ensino Médio (Normal)	A2, A3, A4, A9, A11, A12, A14, A15, A17, A18
Informática	A5, A6, A7, A8, A10, A20
Não Especificado	A1, A13
Agricultura	A5
Desenvolvimento de Jogos Digitais	A16
Programação para Jogos e Aplicativos	A19

Tabela 8. Linguagens de programação empregadas.

Linguagens	Publicações
Linguagem de blocos	A2, A7, A9, A13, A17, A18, A20
C++	A4, A6, A7, A11, A12
C	A2, A4, A6, A17
Linguagem Visual	A2, A16, A19
Python	A13, A15, A17
Pseudocódigo	A3, A9
Java	A8
JavaScript	A19
Portugol	A7

os professores foi outro desafio mencionado em [A1], [A3] e [A20] impactando, a qualidade de ensino, principalmente em novas abordagens.

Além disso, a quantidade de pesquisas insuficientes sobre certas metodologias em [A3] e [A8] e a dificuldade de continuidade dos projetos, dada por falta de financiamento em [A2] e [A19], pela evasão dos estudantes em [A4], [A9] e [A17] ou pelo conteúdo complexo e abstrato em [A5], [A7], [A10], [A11], [A14], [A17], [A18], [A19] e [A20], afetando a continuidade de alguns projetos, foram outros desafios encontrados durante a análise dos estudos. Por fim, a quantidade de dados é insuficiente para a generalização dos resultados em [A13].

4.2.3 Resposta à QP3

A QP3 pretende sistematizar e analisar as principais abordagens pedagógicas presentes nos estudos selecionados. Para isso, é preciso organizar as informações sobre as estratégias e abordagens identificadas nos estudos. Dessa forma, a classificação dessas abordagens é detalhada na Tabela 9.

Tabela 9. Classificação de Atividades por Tipo de Tecnologia Educacional

Abordagens	Publicações
Software	A1; A2; A3; A8; A9; A10; A11; A12; A15; A16; A17; A18; A20
Gamificação	A2; A5; A7; A14; A16; A17; A20
Robótica	A2; A4; A5; A6; A13; A17; A19
Programação em Bloco	A9; A16; A18; A19; A20
Computação Desplugada	A1; A3; A5; A17
IoT	A5; A17; A20

Como visto na Tabela 9, o uso da abordagem de ferramentas de *software* demonstrou ser o tipo de abordagem mais utilizada, estando presente em 34,21% do total de estudos analisados. O uso da gamificação e da robótica corresponde, cada uma, a cerca de 18,42%, sendo ambas as segundas abordagens mais utilizadas nas salas de aula, segundo os dados sintetizados. Em seguida, a programação em bloco está pre-

sente em 13,16% dos estudos. E os menos utilizados foram a computação desplugada e Internet das Coisas (do inglês, *Internet of Things*), ambos abordados em 7,89% dos estudos. Sendo que alguns dos estudos apresentam mais de uma abordagem por serem desenvolvidos para uma metodologia complementar à outra, caso dos estudos [A2], [A3], [A5], [A9], [A16], [A17], [A18], [A19], [A20].

5 Considerações finais

Este trabalho conduziu uma revisão integrativa sobre as diferentes ferramentas educacionais voltadas para o ensino de programação no ensino médio e técnico. Neste estudo, visamos identificar e catalogar as principais metodologias e tecnologias utilizadas pelos estudos, destacando suas vantagens e limitações, a fim de oferecer um panorama recente sobre as práticas de ensino mais eficazes. Para isso, esta revisão analisou somente estudos entre os anos de 2019 a 2024.

A fim de manter a qualidade dos resultados encontrados, foram estabelecidos critérios de inclusão e exclusão. No entanto, vieses podem ter impactado os resultados, como o viés de seleção, dado pela exclusão inadvertida de estudos que poderiam ser relevantes. Além disso, o viés de generalização pode ter sido outro risco, pois alguns dos estudos analisados podem não representar diferentes realidades educacionais e econômicas dos estudantes. Para reduzir esses riscos, foi utilizado o diagrama PRISMA, quadros e tabelas para documentar e visualizar o processo da seleção.

Por meio dos estudos revisados, conclui-se que as ferramentas gamificadas, como plataformas interativas e jogos educacionais, e as abordagens como a aprendizagem baseada em projetos e em problemas, metodologias ativas e oficinas, potencializaram o envolvimento dos alunos, tornando o processo de aprendizagem mais envolvente, lúdico e divertido. Estimulando, assim, o pensamento crítico, a resolução de problemas e o trabalho em equipe, elementos essenciais para uma aprendizagem aplicável ao contexto real do dia a dia. A programação em blocos foi reconhecida como uma estratégia eficiente para ensinar conceitos fundamentais de programação, minimizando os desafios ligados a sintaxes complexas das linguagens de programação. Também, a robótica educacional mostrou ser um método eficaz para estimular o aprendizado prático, unindo conceitos de diversas disciplinas, como física, matemática e eletrônica. No entanto, foram detectados desafios consideráveis, sendo os principais a ausência de infraestrutura apropriada, incluindo salas de aula mal equipadas, recursos tecnológicos restritos, pela falta de hardware e softwares para a utilização nas atividades e pela falta de formação docente para utilizar dessas ferramentas.

Para trabalhos futuros, a intenção é investigar a aplicação de tecnologias emergentes, como a inteligência artificial e a realidade virtual, visando aprimorar o aprendizado de programação. Para isso, devem ser realizadas novas buscas, incluindo novos periódicos e repositórios, a fim de fornecer uma visão mais abrangente das ferramentas educacionais em uso.

Declarações complementares

Agradecimentos

Este trabalho foi apoiado pelo Programa de Iniciação Científica e Tecnológica Voluntária (PICTV) do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE).

Contribuições dos autores

Fransoar AS: Redação, revisão e edição; coleta de dados; metodologia; análise e discussão dos resultados. Maria AOC: Formatação, redação, revisão e edição; discussão dos resultados. Adonias CO: Coordenação e orientação do projeto; formatação; revisão, edição e submissão; metodologia; análise e discussão dos resultados. Daniel LS: revisão e edição; discussão dos resultados. David MR: revisão e edição; discussão dos resultados. Paulo CAJ: Orientação; revisão. Todos os autores revisaram e aprovaram o manuscrito final.

Outras informações relevantes

Durante a preparação deste trabalho, os autores usaram o ChatGPT-5.2 para aprimorar a redação e estrutura do texto, além de corrigir erros ortográficos e gramaticais. Após utilizar esta ferramenta/serviço, os autores revisaram e editaram o conteúdo conforme necessário e assumem total responsabilidade pelo conteúdo do artigo publicado.

Referências

- Afari, E. and Khine, M. (2017). Robotics as an educational tool: Impact of lego mindstorms. *International Journal of Information and Education Technology*, 7:437–442. DOI: 10.18178/ijiet.2017.7.6.908.
- Agbo, F. J., Oyelere, S. S., Suhonen, J., and Adewumi, S. (2019). A systematic review of computational thinking approach for programming education in higher education institutions. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, Koli Calling '19, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3364510.3364521.
- Asgari, M., Tsai, F.-C., Mannila, L., Strömbäck, F., and Sadique, K. M. (2024). Students' perspectives on using digital tools in programming courses. *Discover Education*, 3. DOI: 10.1007/s44217-024-00144-4.
- Barbosa, L., Couto, C., and Terra, R. (2016). Portucol: uma pseudo linguagem inspirada em c ansi para o ensino de lógica de programação e algoritmos. In *Anais do XXIV Workshop sobre Educação em Computação*, pages 2343–2352, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/wei.2016.9678.
- Botelho, L. L. R., Cunha, C. C. d. A., and Macedo, M. (2011). O MÉTODO DA REVISÃO INTEGRATIVA NOS ESTUDOS ORGANIZACIONAIS. *Gestão e Sociedade*, 5(11):121–136. DOI: 10.21171/ges.v5i11.1220.
- CodeCombat (2024). About codecombat (codecombat documentation site). Disponível em: <https://codecombat.com/>. Acessado em 21 de fevereiro de 2026.
- de Farias, C. M., de Oliveira, A. S., and de A. Silva, E. D. (2018). Uso do scratch na introdução de conceitos de lógica de programação: relato de experiência. In *Anais do XXVI Workshop sobre Educação em Computação*, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/wei.2018.3491.
- dos Anjos, S. M., Perin, T. A., de Oliveira Meda, M. P., Andrade, H. R. I., Freire, K. C. P., and Minetto, V. A. (2024). *Tecnologia na educação : uma jornada pela*

- evolução o histórica, desafios atuais e perspectivas futuras*. Quipá Editora, 1 edition. Disponível em: <https://educapes.capes.gov.br/handle/capes/742072>.
- Eugenio F. Pereira, D., Duarte Seabra, R., and Diniz de Souza, A. (2021). Ferramentas de apoio ao ensino introdutório de programação: um mapeamento sistemático. *RENOTE*, 18(2):491–500. DOI: 10.22456/1679-1916.110289.
- Garlet, D., Bigolin, N. M., and Silveira, S. R. (2018). Ensino de programação de computadores na educação básica: um estudo de caso. *Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica*, 9:135–160. Disponível em: <http://periodicos.unifacef.com.br/resiget/article/view/1604>.
- Instituto Semesp (2023). Mapa do ensino superior no brasil: 13ª edição. Publicação baseada em dados do Censo da Educação Superior 2021 (Inep) e outras fontes como IBGE, ENEM, PROUNI, CAGED e RAIS. Disponível em: <https://www.semesp.org.br/mapa/edicao-13/>.
- Inventor, M. A. (2012). About mit app inventor (mit app inventor documentation site). Disponível em: <https://appinventor.mit.edu/about-us>. Acessado em 21 de fevereiro de 2026.
- José Richter, C., Bernardi, G., and Zanki Cordenonsi, A. (2019). O ensino de programação mediado por tecnologias educacionais: uma revisão sistemática de literatura. *RENOTE*, 17(1):517–526. DOI: 10.22456/1679-1916.95903.
- Larsen, F. N. (2021). Readme for robocode. Disponível em: <https://robocode.sourceforge.io/docs/ReadMe.html>. Acessado em 21 de fevereiro de 2026.
- MJ, P. (2020). Prisma 2020 flow diagram for new systematic reviews which included searches of databases and registers only. Disponível em: <https://www.prisma-statement.org/prisma-2020-flow-diagram>. Acessado em 21 de fevereiro de 2026.
- Moraes, R. P. d., Costa, V. F. d., and Scholz, R. E. P. (2022). Mapeamento sistemático do ensino introdutório de programação nos ensinos técnico e superior no brasil. *Revista Brasileira de Informática na Educação*, 30:628–647. DOI: 10.5753/rbie.2022.2611.
- Oliveira, S., Siqueira, A., and Romão, E. (2020). Aprendizagem baseada em projetos no ensino medio: estudo comparativo entre metodos de ensino. *Bolema: Boletim de Educação Matemática*, 34:764–785. DOI: 10.1590/1980-4415v34n67a20.
- Oro, N., Pazinato, A., Teixeira, A., and Ádler Gross (2015). A olimpíada de programação de computadores para estudantes do ensino fundamental: A interdisciplinaridade por meio do software scratch. In *Anais do XXI Workshop de Informática na Escola*, pages 102–111, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/cbie.wie.2015.102.
- Paiva, J. C., Leal, J. P., and Queirós, R. (2020). Fostering programming practice through games. *Information*, 11(11). DOI: 10.3390/info11110498.
- Pan, T. and Zhu, Y. (2018). *Getting Started with Arduino*, pages 3–16. Springer Singapore, Singapore. DOI: 10.1007/978-981-10-4418-2_1.
- Rovshenov, A. and Sarsar, F. (2023). Research trends in programming education: A systematic review of the articles published between 2012-2020. *Journal of Educational Technology and Online Learning*, 6:34. DOI: 10.31681/jetol.1201010.
- Scratch (2024). ABOUT Scratch (Scratch Documentation Site). Disponível em: <https://scratch.mit.edu/about>. Acessado em 21 de fevereiro de 2026.
- SETEC/MEC (2022). Rede federal de educação profissional, científica e tecnológica. Disponível em: <https://www.gov.br/mec/pt-br/assuntos/ept/rede-federal>. Acessado em 21 de fevereiro de 2026.
- Silva, W. d. S., Lima, M. S., Raposo, J. C. d. S., and Silva Júnior, L. C. F. d. (2018). Levantamento sobre as dificuldades dos discentes nas disciplinas de programação no curso técnico de informática. *Diversitas Journal*, 3(3):761–770. DOI: 10.17648/diversitas-journal-v3i3.616.
- Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104:333–339. DOI: 10.1016/j.jbusres.2019.07.039.
- Sousa, M., Albuquerque, D., Leal, R., Maia, M., Gomes, A., and Perkusich, M. (2022). Análise dos fatores de evasão dos alunos dos cursos técnicos da Área de informática no ifpb campus campina grande. In *Anais do XXX Workshop sobre Educação em Computação*, pages 369–380, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/wei.2022.223124.
- Viana, A. M. and dos Santos Souza, T. M. (2024). UMA ANÁLISE DAS DIFICULDADES DE APRENDIZAGEM DA PROGRAMAÇÃO DE COMPUTADORES NO CURSO TÉCNICO INTEGRADO EM INFORMÁTICA DO IFBA - CAMPUS BRUMADO. *Seminário Nacional e Seminário Internacional Políticas Públicas, Gestão e Práxis Educacional*, 1:2974–2990. Disponível em: <https://anais2.uesb.br/index.php/semgepraxis/article/view/1953>.
- Vieira, C. E. C., Lima Junior, J. A. T. d., and Vieira, P. d. P. (2015). Dificuldades no processo de aprendizagem de algoritmos: uma analise dos resultados na disciplina de al1 do curso de sistemas de informação da faeterj – campus paracambi. *Cadernos UniFOA*, 10(27):5–15. DOI: 10.47385/cadunifoa.v10.n27.293.