

Potencialidades do uso de Inteligência Artificial Generativa como apoio ao Ensino de Programação

Teresinha Letícia da Silva¹, Kajiana Nuernberg Sartor Vidotto¹, Liane Margarida Rockenbach Tarouco¹, Patrícia Fernanda da Silva¹

¹PPGIE – Programa de Pós-graduação em Informática na Educação - Universidade Federal do Rio Grande do Sul (UFRGS) – Porto Alegre – RS – Brasil

leticiasilva.ufsm@gmail.com, kajianansartor@gmail.com,
liane@penta.ufrgs.br, patriciasilvaufrgs@gmail.com

Abstract. *Computer programming skills are increasingly essential in today's world, enabling the development of problem-solving and critical-thinking skills. However, learning to program presents challenges, requiring adjustments to teaching methods to make them more engaging and effective. Recently, the use of Generative Artificial Intelligence tools in education has grown, offering new approaches to teaching programming. This article investigates how Generative Artificial Intelligence is applied in teaching programming, examining academic publications through a literature review to analyze the pedagogical strategies and teaching methodologies employed.*

Resumo. *As habilidades de programação de computadores são cada vez mais essenciais no mundo atual, possibilitando o desenvolvimento de competências como resolução de problemas e pensamento crítico. No entanto, o aprendizado da programação apresenta desafios, exigindo ajustes nos métodos de ensino para torná-los mais envolventes e eficazes. Recentemente, o uso de ferramentas de Inteligência Artificial Generativa na Educação tem crescido, oferecendo novas abordagens para o ensino de programação. Este artigo investiga como a Inteligência Artificial Generativa está sendo aplicada no ensino de programação, examinando publicações acadêmicas por meio de uma revisão de literatura para analisar as estratégias pedagógicas e metodologias de ensino empregadas.*

1. Introdução

Ter habilidades de programação de computadores é cada vez mais importante no mundo atual, pois capacita os indivíduos a utilizar e compreender o ambiente digital com maior eficiência. Essas habilidades são fundamentais para desenvolver competências como a resolução de problemas e o pensamento crítico, oferecendo uma maneira clara e estruturada de expressar ideias e solucionar questões que podem ser aplicadas em diversos outros campos (Yilmaz e Yilmaz, 2023a).

O ensino de programação busca desenvolver nos estudantes a habilidade de resolver problemas de uma maneira mais organizada, eficiente e produtiva, tornando-os aptos a compreender e utilizar algoritmos, sistemas e tecnologias para melhorar a tomada de decisões, o que pode ser aplicado em todos os segmentos da sociedade. No entanto, aprender programação é geralmente um processo desafiador e complexo para a maioria das pessoas (Tsai, 2019). Nesse sentido, visando aprimorar os resultados de ensino e aprendizagem das disciplinas de programação, torna-se necessário ajustar os

métodos de ensino vigentes, buscando proporcionar um processo mais envolvente e eficaz tanto para o aluno quanto para o professor.

Com o objetivo de atender à demanda por um ensino de programação mais personalizado e alinhado às necessidades individuais dos alunos, as pesquisas sobre o emprego de Inteligência Artificial (IA) generativa na Educação estão ganhando relevância. Esses estudos indicam que a utilização da IA pode viabilizar o desenvolvimento de um processo de aprendizagem personalizado para a resolução de problemas de programação, proporcionando orientação e *feedback* de maneira semelhante a uma tutoria entre aluno e professor (Yilmaz e Yilmaz, 2023a).

Segundo Chen *et al.* (2022), a IA tem sido cada vez mais aplicada no campo da Educação, oferecendo novas possibilidades para o ensino e a aprendizagem. A aplicação da Inteligência Artificial no ensino de programação e outras disciplinas apresenta diversas vantagens que podem facilitar o processo educacional. Assim, a IA emerge como uma ferramenta valiosa no apoio ao aprendizado de programação e outras disciplinas, ao proporcionar suporte personalizado, *feedback* imediato, exemplos práticos de código e explicações claras e concisas de conceitos.

Para obter uma visão geral dos trabalhos na área, Silva *et al.* (2024b) apresentam um mapeamento sistemático da literatura, no qual foram analisados artigos sobre IA generativa no ensino de programação, publicados entre janeiro de 2021 e março de 2024, onde foram investigados os tipos de estudos e as abordagens utilizadas. A partir da leitura, os artigos selecionados foram classificados em cinco categorias, de acordo com o objetivo e a abordagem de uso da ferramenta de IA generativa no estudo: (a) avaliação do desempenho, capacidades e limitações das ferramentas de IA generativa, (b) desenvolvimento de ferramentas, (c) entrevistas/questionários, (d) pesquisa exploratória com alunos utilizando uma ferramenta de IA generativa no ensino de programação e, (e) artigos de opinião e revisão. Apesar das importantes contribuições, os autores não investigaram detalhadamente como a IA generativa está sendo aplicada em estudos empíricos com alunos em sala de aula.

Nesse contexto, o trabalho apresentado neste artigo é uma extensão do mapeamento sistemático de Silva *et al.* (2024b), incorporando trabalhos da base de dados nacional SOL da SBC¹ (não abordada no mapeamento anterior), e focando na análise de dados mais específicos dos artigos classificados na categoria: (d) pesquisas exploratórias com alunos usando uma ferramenta de IA generativa no ensino de programação. Assim, o objetivo do presente artigo é responder as seguintes questões: “Como a IA generativa está sendo usada para auxiliar o ensino de programação? E quais as principais potencialidades e desafios encontrados nas pesquisas?” Para responder essa pergunta, este trabalho pretende analisar publicações acadêmicas que discutem acerca da IA no ensino de programação por meio de uma revisão de literatura, com o intuito de verificar as estratégias pedagógicas e as metodologias de ensino e aprendizagem utilizadas nos estudos, bem como os resultados e desafios encontrados.

2. Referencial Teórico

Nesta seção são descritos os principais conceitos para o entendimento deste trabalho.

2.1. Ensino de programação de computadores

O ensino de programação tem como objetivo capacitar os alunos a desenvolver

1 <https://sol.sbc.org.br/>

habilidades e competências necessárias para criar soluções computacionais que resolvam problemas complexos do mundo real (Tarouco *et al.*, 2023). Essas habilidades incluem a codificação, que envolve a escrita de código usando linguagens de programação, além de competências como a resolução de problemas, pensamento lógico, depuração e abstração.

No documento que apresenta as diretrizes de formação para cursos de graduação em Computação elaborado pela Sociedade Brasileira de Computação (SBC), com foco específico nos cursos de Ciência da Computação e Engenharia de Computação (Zorzo *et al.*, 2017), a programação é identificada como uma disciplina central nos fundamentos da Computação, abrangendo tanto a parte científica quanto às técnicas fundamentais para uma formação sólida dos graduandos. O guia de referência da ACM e IEEE para cursos de Ciência da Computação (CC2020, 2020) ressalta que os alunos devem abordar a Computação não apenas como o aprendizado de linguagens de programação, mas sim compreender os conceitos mais amplos da Computação, como abstração e decomposição, dentro do contexto do aprendizado da programação.

Diante desse contexto, percebe-se que o ensino de programação vai além do estudo de linguagens específicas, abrangendo uma variedade de princípios, técnicas e formalidades destinadas a capacitar os alunos no desenvolvimento de software. No entanto, o processo de ensino e aprendizagem de programação é bastante complexo, pois tanto alunos quanto professores enfrentam diversos desafios ao longo desse percurso. Muitos alunos encontram dificuldades significativas para desenvolver as habilidades necessárias à programação, o que contribui para os altos índices de reprovação e evasão nas disciplinas relacionadas, provocando um aumento no abandono dos cursos da área de Computação (Diemer, 2022).

Autores como Luxton-Reilly *et al.* (2018), Medeiros *et al.* (2019) e Qian e Lehman (2017) destacam em seus trabalhos que as principais dificuldades enfrentadas pelos estudantes no aprendizado de programação, são: problemas de leitura e interpretação, desenvolvimento insuficiente do raciocínio lógico, habilidades limitadas na resolução de problemas, baixa capacidade de abstração, baixo nível de conhecimento em matemática, hábitos de estudo inadequados, pouca motivação e falta de persistência ou empenho. Para superar esses desafios, os educadores podem se beneficiar de estratégias de ensino inovadoras, como aulas práticas, estudos de caso, tutoriais interativos e *feedback* individualizado. Além disso, a colaboração entre estudantes e a implementação de abordagens pedagógicas eficazes podem contribuir para a melhoria da qualidade do ensino de linguagens de programação, promovendo um ambiente de aprendizagem mais eficaz.

Recentemente, um conjunto significativo de pesquisas vem investigando o uso de recursos de Inteligência Artificial, dentre eles os Modelos de Linguagem de Grande Escala (*Large Language Models* ou *LLMs*) para auxiliar no processo de ensino e aprendizado em diferentes áreas, incluindo a programação.

2.2. Inteligência Artificial Generativa no Ensino de Programação

As linguagens e ferramentas de programação têm evoluído continuamente para incorporar recursos como autocompletar códigos, tradução entre diferentes linguagens, documentação de software, depuração e testes (Sarkar *et al.*, 2022). Desde a década de 1970, as linguagens de quarta geração foram desenvolvidas visando simplificar a programação, e a necessidade de facilitar o processo de escrita de código sempre foi uma meta importante (Sobral, 2019).

Uma solução que vem ganhando destaque para facilitar a programação é o uso de IA. Ferramentas de programação baseadas em IA funcionam essencialmente como assistentes especializados: podem esclarecer conceitos, responder perguntas, detectar erros e explicar por que um trecho de código não está funcionando. Além disso, são capazes de escrever explicações para trechos de código mal documentados e oferecer sugestões para tarefas rotineiras, aumentando a produtividade. Sob essa perspectiva, a evolução dessas ferramentas está transformando o cenário da programação, ao facilitar o acesso ao conhecimento e promover uma parceria centrada no ser humano, onde programadores e IA trabalham juntos para aprimorar as habilidades de programação (Sánchez-Gordón *et al.*, 2023).

As ferramentas de IA generativa podem gerar código que oferece várias opções potencialmente adequadas, mas o programador ainda precisa decidir quais trechos utilizar e como aplicá-los. Isso requer competências em pensamento crítico, resolução de problemas, tomada de decisões e habilidades relacionadas à fluência de ideias e originalidade. Portanto, é essencial destacar a importância da Educação em ética e responsabilidade para os programadores (Sánchez-Gordón *et al.*, 2023).

Autores com Becker *et al.* (2023) e (Denny *et al.*, 2024) destacam em seus trabalhos diversas oportunidades que as ferramentas de IA generativa oferecem para auxiliar no ensino de programação:

- Exemplos de código: Ferramentas de IA podem gerar exemplos e testes, permitindo que os alunos verifiquem seu trabalho durante a prática de programação;
- Variedade de soluções: Essas ferramentas podem apresentar diferentes maneiras de resolver um problema, ampliando a compreensão dos alunos;
- Revisão de código: As abordagens atuais de avaliação em cursos introdutórios de programação priorizam a correção, em vez da qualidade ou estilo do código. Com a capacidade de gerar automaticamente soluções sintaticamente corretas, a avaliação pode-se concentrar nas diferenças entre múltiplas soluções corretas e fazer análises sobre o estilo e a qualidade das soluções;
- Revisão por pares: A IA pode gerar soluções variadas para discussão e refatoração, promovendo o aprendizado colaborativo;
- Geração de exercícios: As ferramentas de IA podem criar novos exercícios de programação a partir de um único exemplo, enriquecendo os recursos de aprendizagem;
- Explicações de código: A IA pode fornecer explicações claras e detalhadas do código, ajudando os alunos a compreender melhor os conceitos de programação;
- Clareza em conceitos algorítmicos: A construção dos *prompts* inseridos nas ferramentas de IA influencia seu desempenho. Os alunos podem aprimorar a explicação de problemas algorítmicos, fornecendo descrições mais claras para as ferramentas produzindo soluções mais eficazes;
- Superação do bloqueio do programador: Alunos frequentemente enfrentam dificuldades para começar ou continuar tarefas de programação. Ferramentas de IA podem auxiliar na geração do código inicial, permitindo que os alunos se concentrem em expandir esse código, exigindo um foco maior em reescrever, refatorar e depurar;

- Superação de barreiras tradicionais: Iniciantes enfrentam várias dificuldades ao aprender a programar, a IA pode explicar mensagens de erro em linguagem natural e sugerir correções, ajudando iniciantes a superar dificuldades comuns.

A disponibilidade de ferramentas de IA generativa para a geração de código traz benefícios, mas também levanta preocupações sobre seu potencial em limitar a aprendizagem e dificultar o papel dos educadores. Na Computação, a integridade acadêmica é um tema complexo, especialmente devido à reutilização de código e práticas colaborativas incentivadas no desenvolvimento de software. O uso de código gerado automaticamente por ferramentas de IA levanta preocupações relacionadas à integridade acadêmica e à reutilização de código. Isso inclui (Becker *et al.*, 2023):

- Má conduta acadêmica: O uso de ferramentas de IA, capazes de superar alunos em provas de programação e gerar código correto para algoritmos, pode intensificar a desonestidade acadêmica, oferecendo um caminho fácil para estudantes que buscam apenas obter boas notas em vez do aprendizado;
- Atribuição: O uso de ferramentas de IA em IDEs para sugestões automáticas de código, levanta questões sobre o uso de código gerado por IA, provocando reflexões sobre a atribuição da propriedade intelectual deste código, plágio e contribuição intelectual na geração de conteúdo assistida por máquinas;
- Reutilização e licenciamento de código: Códigos disponíveis publicamente, frequentemente possuem múltiplas licenças que exigem atribuição aos autores originais. Desenvolvedores que usam código por meio de ferramentas de IA podem inadvertidamente estar sujeitos a requisitos de licenciamento exigindo cuidado na atribuição correta;
- Adequação para iniciantes: O código usado para treinar ferramentas de IA geralmente não é adequado para iniciantes, sendo mais avançado e com estilos que podem não ser apropriados para estudantes em fase inicial de aprendizado;
- Segurança: Ensinar práticas de codificação seguras tem sido amplamente documentada nos últimos anos. Portanto, a segurança do código gerado por IA é de extrema importância, mesmo em ambientes educacionais;
- Excesso de confiança: Ferramentas de IA podem levar a dependência excessiva do usuário. Especialmente com ferramentas integradas em IDEs, iniciantes podem se acostumar rapidamente com soluções sugeridas, o que pode comprometer a compreensão dos problemas;
- Reforçar comportamentos que impedem a aprendizagem: Código incorreto sugerido pela IA pode fazer com que os alunos percam a confiança no *feedback* dos IDEs, comprometendo sua aprendizagem e dificultando a interpretação de mensagens de erro e avisos.

3. Metodologia

Para verificar as estratégias pedagógicas e as metodologias de ensino e aprendizagem utilizadas nos estudos sobre o uso IA generativa no ensino de programação, realizou-se uma revisão de literatura no período de janeiro de 2022 até março de 2024.

Foi realizada uma busca nas bases de dados nacionais e internacionais eletrônicas indexadas: ACM *Digital Library*, Scopus, IEEE *Digital Library*, Science *Direct*, *Web of Science* e Biblioteca SOL da SBC. Os termos e sinônimos considerados na construção da *string* de pesquisa são apresentados no Quadro 1. Para a pesquisa na

base nacional os termos foram traduzidos para Português. A pesquisa foi realizada com a opção de busca avançada, considerando os estudos que possuísem os termos listados em seus títulos, resumos ou palavras-chave.

Quadro 1. String de busca

("Generative Artificial Intelligence" OR "GenAI" OR "Generative AI" OR "LLM" OR "AI Code Generators" OR "Codex" OR "ChatGPT" OR "Copilot" OR "GPT" OR "Gemini" OR "OpenAI") AND ("Programming Education" OR "Introductory Programming" OR "Programming Instruction" OR "Programming Learning" OR "Programming Teaching")

O principal objetivo desta revisão foi identificar os estudos mais relevantes que aplicaram o uso de Inteligência Artificial generativa no auxílio do ensino de programação em estudos empíricos em sala de aula com os alunos. Para direcionar a pesquisa foram elaboradas as seguintes questões: **QP1:** Quais as ferramentas de IA generativa estão sendo utilizadas nas experiências de ensino e aprendizagem de programação? **QP2:** Quais são as metodologias ou estratégias pedagógicas que estão sendo utilizadas em conjunto com as ferramentas de IA generativa no ensino de programação? **QP3:** Quais são os principais resultados e impactos observados nas experiências de ensino e aprendizagem de programação com o auxílio de ferramentas de IA generativa? **QP4:** Quais são os principais desafios enfrentados nas experiências de ensino e aprendizagem de programação com o auxílio de ferramentas de IA generativa?

Critérios de inclusão e exclusão foram definidos para que, após os resultados das buscas, fosse possível selecionar os artigos que se enquadram no contexto do estudo. Os critérios de inclusão foram: "Deve mencionar IA generativa, grandes modelos de linguagem ou uma ferramenta específica que utilize essas tecnologias", "Deve ter aplicabilidade direta ao ensino de programação.", "Deve ter um estudo empírico com alunos em sala de aula", "No mínimo 4 páginas". Já os critérios de exclusão usados foram: "Não utiliza IA generativa", "Não se aplica ao ensino de programação", "Menos de 4 páginas", "Publicado antes de 2022"; "Artigos derivados da mesma pesquisa (estudos duplicados)" e "Artigos não disponibilizados gratuitamente".

A busca de artigos nas bases de dados, utilizando a *string* de busca, resultou em um total de 169 estudos. A Figura 1 apresenta um modelo de fluxograma (Page *et al.*, 2021) que ilustra todo o processo de busca e seleção dos artigos nas bases de dados.

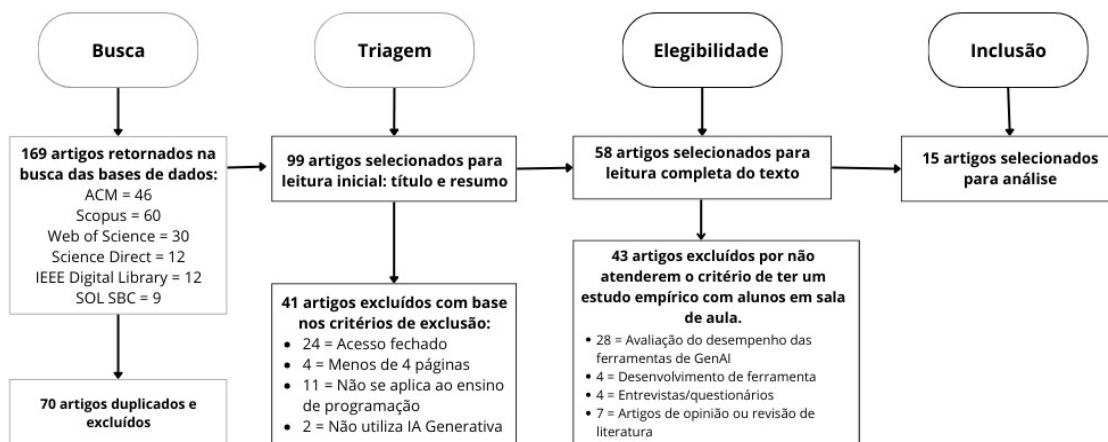


Figura 1. Processo de busca e seleção dos artigos

Os resultados foram importados para a ferramenta Parsifal², e, com auxílio da

2 <https://parsif.al/>

mesma, foram excluídos os artigos duplicados (70 artigos). Em seguida, foi realizada a leitura do título e resumo de cada artigo, o que resultou na exclusão de 41 estudos com base nos critérios estabelecidos. Uma leitura parcial dos 58 artigos restantes foi realizada, e 43 deles foram excluídos por não atenderem ao critério de inclusão de estudos empíricos com alunos em sala de aula, restando 15 artigos para leitura completa e análise.

4. Resultados e discussão

Conforme relatado em Silva *et al.* (2024b), por ser uma área bastante nova, as pesquisas referentes ao uso de IA generativa no ensino de programação, estão em sua maioria inicialmente focadas em testar as ferramentas para verificar seu desempenho na resolução de problemas de programação. Na busca realizada nesta pesquisa, apenas 15 artigos tratavam de um estudo empírico com alunos em sala de aula usando ferramentas de IA generativa para o ensino e aprendizagem de programação.

Analisando os artigos selecionados, observa-se que o número de pesquisas nessa área está crescendo em ritmo acelerado (Figura 2), com o ano de 2023 registrando o maior número de estudos (8). No entanto, como a data final estabelecida para o estudo foi março de 2024, é provável que muitas pesquisas ainda estejam em andamento ou venham a ser publicadas neste ano, ultrapassando possivelmente o número de publicações de 2023. Os estudos incluídos nesta revisão foram realizados em diferentes países, com o Brasil ocupando o primeiro lugar em quantidade de publicações, seguido pela China e Turquia. Além desses, outros países que tiveram artigos publicados foram Estados Unidos, Suécia, Nova Zelândia, Croácia, Indonésia e República Tcheca.

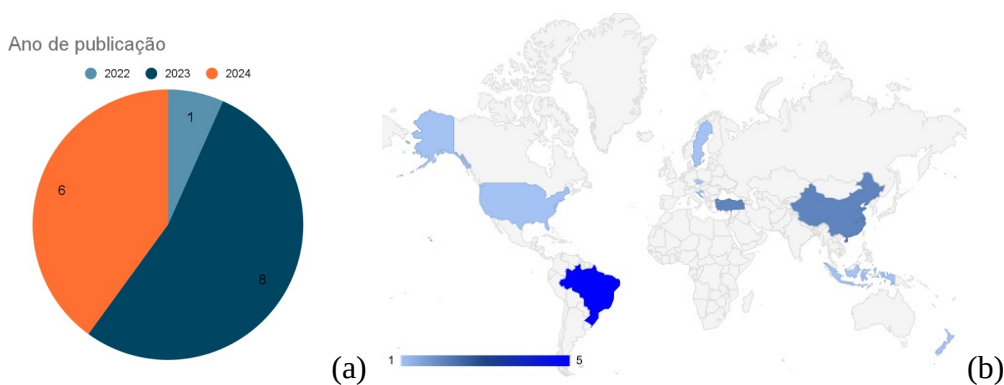


Figura 2. (a) Ano de publicação; (b) Local das pesquisas

Para responder à primeira questão de pesquisa (**“QP1: Quais as ferramentas de IA generativa estão sendo utilizadas nas experiências de ensino e aprendizagem de programação?”**), foram analisadas as ferramentas de IA generativa utilizadas nos artigos. O ChatGPT foi identificado como a ferramenta mais utilizada, sendo mencionado em 80% dos artigos analisados (12 estudos). Nesses estudos, os alunos exploraram diversos conceitos de programação com ChatGPT, buscando explicações, exemplos e procedimentos passo a passo para resolver exercícios de programação. Quase todos os estudos usaram a versão do GPT-3.5, sendo que apenas em um (1) estudo relatou o uso por uma parte dos alunos a versão do GPT-4, isso provavelmente se deu porque a versão GPT-3.5 estava disponível gratuitamente durante o período das pesquisas, enquanto o GPT-4 requer uma licença paga. Também foram identificados

estudos com as ferramentas Copilot, presente em 2 artigos, e Codex, usada em 1 artigo.

As ferramentas de IA generativa foram predominantemente usadas para auxiliar os alunos no processo de geração e correção de código e para prover *feedback* automático. Os estudos foram implementados principalmente com alunos de cursos de graduação, exceto por dois trabalhos realizados com alunos de ensino médio, sendo um deles integrado a um curso técnico em informática. Nem todos os estudos explicitaram em quais disciplinas os experimentos foram aplicados, mas as mais citadas foram Programação Orientada a Objetos, Introdução a Programação e Cursos de Extensão em Programação. Também houve experimentos que abrangeram múltiplas disciplinas de programação em diferentes turmas e níveis dos cursos. Em relação às linguagens de programação utilizadas houve uma predominância da linguagem *Python* (usada em 4 estudos), seguida por *Java* (citada em 3 estudos), *C/C++* (citadas em 2 estudos), e *JavaScript*, *C#* e Linguagens de programação Web (citadas em 1 estudo cada). Em três trabalhos, não foi especificada a linguagem de programação usada. O Quadro 2, apresenta um detalhamento dos artigos analisados e suas informações.

Quadro 2. Trabalhos selecionados

Referência	Ano	Nível ensino	Ferramenta de IA	Disciplina	Linguagem de programação
(Prather <i>et al.</i> , 2024)	2023	Graduação	Copilot	Introdução a Programação	C++
(Lelli <i>et al.</i> , 2024)	2024	Graduação + Uma turma de Pós	ChatGPT	Envolve múltiplas disciplinas de programação	Python
(Yilmaz e Yilmaz, 2023a)	2023	Graduação	ChatGPT	Programação Orientada a objetos	Não especificada
(Santos e Cury, 2023)	2023	Graduação	ChatGPT	Introdução a Programação	Não especificada
(Silva Junior <i>et al.</i> , 2023)	2023	Ensino médio (Técnico em Informática)	ChatGPT	Envolve múltiplas disciplinas de programação	Não especificada
(Silva <i>et al.</i> , 2024a)	2024	Graduação	ChatGPT	Introdução a Programação	C/C++
(Jonsson e Tholander, 2022)	2022	Graduação	Codex	Curso de Extensão em Programação	JavaScript
(Hu, Assadi e Mahroeian, 2023)	2023	Graduação	ChatGPT	Programação Orientada a objetos	C#
(Banić <i>et al.</i> , 2023)	2023	Ensino médio	ChatGPT	Não especificado	Python
(Karnalim <i>et al.</i> , 2023)	2023	Graduação	ChatGPT	Envolve múltiplas disciplinas de programação	Linguagens de programação Web
(Haindl e Weinberger, 2024)	2024	Graduação	ChatGPT	Curso de Extensão em Programação	Java
(Yilmaz e Yilmaz, 2023b)	2023	Graduação	ChatGPT	Programação Orientada a objetos	Java
(Lira <i>et al.</i> , 2024)	2024	Graduação	Copilot	Programação Orientada a objetos	Java
(Jing <i>et al.</i> , 2024)	2024	Graduação	ChatGPT	Envolve múltiplas disciplinas de programação	Python
(Sun <i>et al.</i> , 2024)	2024	Graduação	ChatGPT	Não especificado	Python

Conforme Silva e Tarouco (2022), uma estratégia empregada para o ensino de programação é a utilização de ferramentas tecnológicas em conjunto com metodologias educacionais apropriadas. Essa combinação visa promover uma interação mais envolvente dos estudantes com o processo de aprendizado de programação. Neste sentido, procurou-se analisar os artigos visando responder a segunda questão de pesquisa: **“QP2: Quais são as metodologias ou estratégias pedagógicas que estão sendo utilizadas em conjunto com as ferramentas de IA generativa no ensino de programação?”**

Nos estudos de Prather *et al.* (2024), Yilmaz e Yilmaz (2023a), Silva *et al.* (2024a), Karnalim *et al.* (2023), Haindl e Weinberger (2024), Yilmaz e Yilmaz (2023b), Jing *et al.* (2024), Sun *et al.* (2024), Silva Junior *et al.* (2023) e Lira *et al.* (2024), a abordagem pedagógica da intervenção não foi especificada. Os alunos experimentaram uma abordagem direta e interativa de aprendizagem, trabalhando na resolução de problemas de programação, sendo observado como ocorreu a interação desses alunos com a ferramenta de IA generativa. Embora a abordagem não tenha sido explicitamente descrita, é possível perceber que os estudos empregaram a metodologia de Aprendizagem Baseada em Problemas (ABP), em que o foco principal é a atuação ativa dos estudantes na resolução de problemas e construção do seu conhecimento (Barrows, 1986). Ilxomovna e Akmalovna (2023) destacam a importância da integração do método de ABP no ensino de programação, ressaltando que essa abordagem, que envolve a resolução de problemas reais ou simulados pelos alunos, promove o desenvolvimento de habilidades práticas, pensamento crítico, colaboração e comunicação.

Para Santos e Cury (2023), Jonsson e Tholander (2022) e Banić *et al.* (2023), a ferramenta de IA generativa foi utilizada como um colega virtual por meio da metodologia de Programação Baseada em Pares. Segundo Othman *et al.* (2019), essa metodologia é uma prática em que dois programadores trabalham colaborativamente em um computador, no mesmo projeto, algoritmo ou código, compartilhando um único monitor, teclado e mouse. Essa metodologia tem sido adotada como uma estratégia de ensino colaborativo, especialmente em cursos introdutórios de programação.

O estudo de Lelli *et al.* (2024) utilizou a metodologia POGIL (*Process-Oriented Guided Inquiry Learning*) em conjunto com a ferramenta de IA generativa ChatGPT. O POGIL é uma estratégia de ensino colaborativo que estimula a aprendizagem por meio da investigação e resolução de problemas. Essa metodologia envolve um ciclo de construção do conhecimento que ocorre de maneira coletiva e em etapas. Os alunos exploram questões individualmente, discutem em grupos e, em seguida, organizam-se em pequenos grupos para debater suas respostas, assumindo papéis específicos para otimizar o processo de aprendizagem. Essa abordagem promove a participação ativa dos alunos em todas as fases do aprendizado, incentivando a exploração de informações, o desenvolvimento de hipóteses e a aplicação de novos conhecimentos para resolver problemas complexos (Moog, 2008).

Hu, Assadi e Mahrooian (2023) mencionam em seu estudo a utilização de métodos de desenvolvimento de software, como o método de prototipagem e o desenvolvimento orientado a testes (*test-driven development*), como abordagens pedagógicas empregadas. Os alunos foram instruídos a desenvolver um projeto de aplicativo de negócios de uma pseudo-concessionária de carros, usando programação orientada a objetos (OOP) com a linguagem C#, com a assistência do ChatGPT. Mesmo não estando explícito no texto, pode-se perceber que foi usada também a metodologia de Aprendizagem Baseada em Projetos. Esta abordagem visa envolver os alunos na

- Impacto positivo na motivação dos alunos: O uso de tecnologias como o ChatGPT parece ter aumentado a motivação dos alunos no aprendizado de programação (Yilmaz e Yilmaz, 2023a; Sun *et al.*, 2024);
- Desenvolvimento do pensamento crítico: Houve uma melhoria significativa no desenvolvimento do pensamento crítico dos alunos, possivelmente devido ao desafio proporcionado pela interação com a IA (Yilmaz e Yilmaz, 2023a);
- Impactos positivos no desempenho dos alunos e na compreensão conceitual: os grupos que utilizaram a IA demonstraram pontuações médias mais altas nas tarefas de programação em comparação com os que não utilizaram, indicando um melhor desempenho em tarefas de codificação (Santos e Cury, 2023; Lira *et al.*, 2024; Yilmaz e Yilmaz, 2023b; Jing *et al.*, 2024; Sun *et al.*, 2024);
- Melhoria na qualidade do código produzido: Um maior percentual de submissões de código de melhor qualidade foi observado nos grupos que utilizaram a IA (Santos e Cury, 2023; Haindl e Weinberger, 2024; Banić *et al.*, 2023);
- Consolidação do aprendizado: Os alunos conseguiram consolidar melhor o que aprenderam, o que é crucial para a retenção de conhecimento a longo prazo (Haindl e Weinberger, 2024; Banić *et al.*, 2023);
- Facilitação da aprendizagem personalizada: A tecnologia permitiu uma abordagem personalizada para cada aluno, adaptando-se às suas necessidades específicas (Sun *et al.*, 2024);
- Estímulo à criatividade e inovação: Dificuldades e imprevisibilidades na interação com IA foram vistas como oportunidades para explorar soluções criativas e gerar novas ideias (Banić *et al.*, 2023).

A última questão de pesquisa (**“QP4: Quais são os principais desafios enfrentados nas experiências de ensino e aprendizagem de programação com o auxílio de ferramentas de IA generativa?”**), visa identificar os principais desafios enfrentados nas experiências de ensino e aprendizagem de programação com o auxílio de ferramentas de IA generativa. De acordo com os artigos analisados, pode-se destacar alguns pontos principais:

- Variação nas respostas e complexidade: Alguns alunos enfrentaram problemas com respostas variáveis e, às vezes, mais complexas do que o necessário, o que prejudicou a assimilação dos conceitos e a compreensão dos conteúdos (Karnalim *et al.*, 2023; Silva Junior *et al.*, 2023; Lelli *et al.*, 2024; Lira *et al.*, 2024; Santos e Cury, 2023; Prather *et al.*, 2024);
- Comportamento inconsistente da IA: O comportamento inconsistente e imperfeito dos sistemas de IA para geração de código apresentou desafios para os participantes. Os alunos observaram que a IA às vezes produzia códigos imprecisos ou com erros, o que poderia impactar negativamente seu desempenho e compreensão da programação (Jonsson e Tholander, 2022; Sun *et al.*, 2024; Yilmaz e Yilmaz, 2023a; Haindl e Weinberger, 2024; Santos e Cury, 2023; Prather *et al.*, 2024).
- Necessidade de treinamento: Em alguns estudos, alunos com pouco ou nenhum conhecimento prévio sobre a ferramenta de IA tiveram uma experiência negativa ao usá-la, destacando a importância de fornecer um treinamento adequado para melhorar sua eficácia e usabilidade (Jing *et al.*, 2024; Lelli *et al.*, 2024);
- Desafios éticos e de dependência: Foram mencionadas preocupações éticas,

como o risco de plágio de código e a dependência excessiva das ferramentas de IA para resolver problemas de programação, em detrimento do desenvolvimento autônomo de habilidades (Hu *et al.*, 2023; Sun *et al.*, 2024; Karnalim *et al.*, 2023; Yilmaz e Yilmaz, 2023a; Silva *et al.*, 2024a; Haindl e Weinberger, 2024) (Prather *et al.*, 2024);

- Limitações e ajustes no código: Houve relatos de dificuldades com a qualidade e precisão do código gerado pelas ferramentas de IA, exigindo ajustes por parte dos alunos para corrigir ou completar as sugestões automáticas (Lira *et al.*, 2024).
- Necessidade de Acompanhamento e Supervisão: Os alunos ressaltaram a importância de uma supervisão adequada ao usar a IA, pois a falta de orientação pode dificultar sua integração eficaz no ensino de programação (Sun *et al.*, 2024; Santos e Cury, 2023).
- Necessidade de Adaptação do Ensino: A integração das ferramentas de IA requer a adaptação dos métodos de ensino para garantir que os alunos desenvolvam habilidades de pensamento crítico e resolução de problemas, que não podem ser totalmente substituídas pela IA. Educadores devem criar tarefas que incentivem o pensamento crítico e a aplicação dos conceitos de programação (Hu *et al.*, 2023; Banić *et al.*, 2023; Yilmaz e Yilmaz, 2023b; Silva Junior *et al.*, 2023; Silva *et al.*, 2024a).

Esses pontos destacam os desafios e as oportunidades associadas à integração de tecnologias baseadas em IA no ensino de programação, sublinhando a importância de uma implementação cuidadosa e adaptativa para maximizar os benefícios educacionais.

5. Conclusões

Esta pesquisa mapeou a aplicação da Inteligência Artificial generativa no ensino de programação, destacando como essas ferramentas podem transformar a experiência educacional. Por meio de uma revisão de literatura, foram analisadas diversas publicações acadêmicas para identificar as estratégias pedagógicas e metodologias de ensino empregadas.

Os resultados confirmam que a IA generativa tem um potencial significativo para tornar o aprendizado de programação mais envolvente e eficaz. As ferramentas de IA generativa podem personalizar o ensino, adaptando-se às necessidades individuais dos alunos e fornecendo *feedback* imediato. Isso não apenas facilita a compreensão dos conceitos de programação, mas também promove o desenvolvimento de competências essenciais como a resolução de problemas e o pensamento crítico.

No entanto, a implementação eficaz dessas tecnologias requer planejamento e ajustes nos métodos de ensino. É essencial que os educadores recebam treinamento adequado para integrar essas ferramentas de maneira eficaz em suas aulas e façam uso conjunto de estratégias pedagógicas apropriadas. Além disso, a pesquisa contínua é necessária para avaliar os impactos a longo prazo do uso de IA generativa na Educação e para desenvolver melhores práticas.

A IA generativa oferece perspectivas promissoras para superar os desafios tradicionais do ensino de programação. Este estudo contribuiu para a compreensão de como essas tecnologias podem ser aplicadas, as potencialidades e desafios encontrados, e destaca também a necessidade do uso de estratégias pedagógicas para maximizar seu potencial.

Agradecimentos

O presente trabalho foi realizado com apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Referências

- Banić, B., Konecki, M., and Konecki, M. (2023) “Pair Programming Education Aided by ChatGPT”. In: 2023 46th MIPRO ICT and Electronics Convention (MIPRO). Opatija, Croatia: IEEE, p. 911–915. <https://ieeexplore.ieee.org/document/10159727/>.
- Barrows, H. S. (1986) “A taxonomy of problem-based learning”. *Medical education*, 20, 481-486.
- Becker, B. A., Denny, P. and Finnie-Ansley, J. et al. (2023) “Programming Is Hard - Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation”. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1.* . ACM. <https://doi/10.1145/3545945.3569759>.
- Chen, Xieling et al. (2022) “Two decades of artificial intelligence in education”. *Educational Technology & Society*, v. 25, n. 1, p. 28-47.
- CC2020 Task Force (2020) “Computing Curricula 2020: Paradigms for Global Computing Education”. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/3467967>.
- Denny, P., Prather, J., Becker, B. A., Finnie-Ansley, J., Hellas, A., Leinonen, J., Luxton-Reilly, A., Reeves, B. N., Santos, E. A., & Sarsa, S. (2024). “Computing Education in the Era of Generative AI”. *Communications of the ACM*, 67(2), 56–67. <https://doi.org/10.1145/3624720>
- Diemer, Mouriac Halen (2022) “Grupos de colaboração: a influência da postura interpessoal na aprendizagem de lógica de programação”.
- Haindl, P., Weinberger, G. (2024) “Students’ Experiences of Using ChatGPT in an Undergraduate Programming Course”. *IEEE Access*, v. 12, p. 43519–43529. <https://ieeexplore.ieee.org/document/10478015/>.
- Hu, M., Assadi, T. and Mahrooian, H. (2023) “Explicitly Introducing ChatGPT into First-year Programming Practice: Challenges and Impact”. In: 2023 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE). Auckland, New Zealand: IEEE, 2023, p. 1–6. <https://ieeexplore.ieee.org/document/10398297/>.
- Ilxomovna, K. N. and Akmalovna, I. M. (2023) “Using The Problem-Based Design Method In Teaching Programming Languages”. *Academia Repository*, v. 4, n. 12, p. 1–7. <https://academiarepo.org/index.php/1/article/view/388>.
- Jing, Y. et al. (2024) “What factors will affect the effectiveness of using ChatGPT to solve programming problems? A quasi-experimental study”. *Humanities and Social Sciences Communications*, v. 11, n. 1, p. 319. <https://www.nature.com/articles/s41599-024-02751-w>.
- Jonsson, M., and Tholander, J. (2022) “Cracking the code: Co-coding with AI in creative programming education”. In: *Creativity and Cognition*. Venice Italy: ACM, p. 5–14. <https://dl.acm.org/doi/10.1145/3527927.3532801>.

- Karnalim, O. et al. (2023) “Plagiarism and AI Assistance Misuse in Web Programming: Unfair Benefits and Characteristics”. In: 2023 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE). Auckland, New Zealand: IEEE, 2023, p. 1–5. <https://ieeexplore.ieee.org/document/10398397/>.
- Lelli, V., Santos, I. S., Sousa, F., and Braide, L. (2024) “Aplicação do POGIL no ensino de Computação”. In: SIMPÓSIO BRASILEIRO DE EDUCAÇÃO EM COMPUTAÇÃO (EDUCOMP), 4., 2024, Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, p. 224-233. DOI: <https://doi.org/10.5753/educomp.2024.237541>.
- Lira, W. A. L., Santos Neto, P. de A. dos, and Osório, L. F. M. (2024) “Uma análise do uso de ferramentas de geração de código por alunos de Computação”. In: SIMPÓSIO BRASILEIRO DE EDUCAÇÃO EM COMPUTAÇÃO (EDUCOMP), 4., Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, p. 63-71. DOI: <https://doi.org/10.5753/educomp.2024.237427>.
- Luxton-Reilly, A., et al. (2018) “Introductory programming: a systematic literature review”. In Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education. ACM. <https://doi/10.1145/3293881.3295779>.
- Medeiros, R. P., Ramalho, G. L. and Falcao, T. P. (2019) “A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education”. IEEE Transactions on Education, v. 62, n. 2, p. 77–90. <https://doi.org/10.1109/TE.2018.2864133>.
- Moog, R. S. and Spencer, J. N. (2008) “Pogil: An overview”.
- Othman, M., Rosmani, A. F., Mohd Fauzi, S. S., and Mazlan, U. H. (2019) “The Impact of Pair Programming on Students Logical Thinking_A Case Study on Higher Academic Institution”. Social and Management Research Journal, v. 16, n. 1, p. 85.
- Page, M. J., McKenzie, J. E., Bossuyt, P. M., et al. (2021). “The PRISMA 2020 statement: an updated guideline for reporting systematic reviews”. BMJ, p. n71.
- Prather, J. et al. (2024) “It’s Weird That it Knows What I Want: Usability and Interactions with Copilot for Novice Programmers”. ACM Transactions on Computer-Human Interaction, v. 31, n. 1, p. 1–31. <https://dl.acm.org/doi/10.1145/3617367>.
- Qian, Y. and Lehman, J. (2017) “Students’ Misconceptions and Other Difficulties in Introductory Programming: A Literature Review”. ACM Transactions on Computing Education, v. 18, n. 1, p. 1–24. <https://doi.org/10.1145/3077618>.
- Sánchez-Gordón, M. et al. (2023) “Educating Augmented Programmers”. Computer, v. 56, n. 12, p. 100–104. <https://ieeexplore.ieee.org/document/10319928/>.
- Santos, O. L. dos, and Cury, D. (2023) “Challenging the Confirmation Bias: Using ChatGPT as a Virtual Peer for Peer Instruction in Computer Programming Education”. In: 2023 IEEE Frontiers in Education Conference (FIE). College Station, TX, USA: IEEE, p. 1–7. <https://ieeexplore.ieee.org/document/10343247/>.
- Sarkar, A., Gordon, A. D., Negreanu, C., Poelitz, C., Ragavan, S. S., & Zorn, B. (2022). “What is it like to program with artificial intelligence?” <https://doi.org/10.48550/ARXIV.2208.06213>

- Silva, C. A. G. da et al. (2024a) “ChatGPT: Challenges and Benefits in Software Programming for Higher Education”. *Sustainability*, v. 16, n. 3, p. 1245. <https://www.mdpi.com/2071-1050/16/3/1245>.
- Silva Junior et al. (2023) “ChatGPT no auxílio da aprendizagem de programação: Um estudo de caso”. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE), 34, Passo Fundo/RS. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, p. 1375-1384. DOI: <https://doi.org/10.5753/sbie.2023.234873>.
- Silva, Teresinha Letícia da, and Tarouco, Liane Margarida Rockenbach. (2022). “Development Of Computational Thinking: Goals And Strategies.” In: 14th International Conference on Education and New Learning Technologies, 2022, Palma, Espanha. 14th International Conference on Education and New Learning Technologies. Valência, Espanha: International Academy of Technology, Education and Development (IATED), p. 5171-5177.
- Silva, T. L. da, Vidotto, K. N. S., Tarouco, L. M. R. and Silva, P. F. da. (2024b). “Inteligência artificial generativa no ensino de programação: um mapeamento sistemático da literatura”. *Revista Novas Tecnologias na Educação*, Porto Alegre, v. 22, n. 1, p. 262–272. Doi: 10.22456/1679-1916.141553.
- Sobral, S. (2019). “30 Years Of CS1: Programming Languages Evolution”. 9197–9205. <https://doi.org/10.21125/iceri.2019.2214>
- Sobral, S. R. (2021) “Project Based Learning with Peer Assessment in an Introductory Programming Course”, *International Journal of Information and Education Technology*, vol. 11, no. 7, pp. 337-341.
- Sun, D. et al. (2024) “Would ChatGPT-facilitated programming mode impact college students’ programming behaviors, performances, and perceptions? An empirical study”. *International Journal of Educational Technology in Higher Education*, v. 21, n. 1, p. 14. <https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-024-00446-5>. 1
- Tarouco, L. M. R., Silva, P. F. da, Silva, T. L. da, and Vidotto, K. N. S. (2023). “Digital competence formation of the citizen for the creation in the digital world using programming”: *Concilium*, 23(14), 443–470. <https://doi.org/10.53660/CLM-1648-23J45>.
- Tsai, Chun-Yen (2019) “Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy”, *Computers in Human Behavior*, vol 95, pp. 224-232, ISSN 0747-5632, <https://doi.org/10.1016/j.chb.2018.11.038>.
- Yilmaz, R. and Yilmaz, F. G. K. (2023a) “Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning”. *Computers in Human Behavior: Artificial Humans*, v. 1, n. 2, p. 100005. <https://linkinghub.elsevier.com/retrieve/pii/S2949882123000051>.
- Yilmaz, R. and Yilmaz, F. G. K. (2023b) “The effect of generative artificial intelligence (AI)-based tool use on students’ computational thinking skills, programming self-efficacy and motivation”. *Computers and Education: Artificial Intelligence*, v. 4, p. 100147. <https://linkinghub.elsevier.com/retrieve/pii/S2666920X23000267>.
- Zorzo, A. F. et al. (2017) “Referenciais de Formação para os Cursos de Graduação em Computação”. Sociedade Brasileira de Computação (SBC). 153p. ISBN 978-85-7669-424-3.