

Inteligência Artificial Generativa no Ensino de Programação: Um Mapeamento Sistemático da Literatura

Teresinha Letícia da Silva, Universidade Federal do Rio Grande do Sul,
leticiasilva.ufsm@gmail.com, <https://orcid.org/0000-0001-5725-5838>

Kajiana Nuernberg Sartor Vidotto, Universidade Federal do Rio Grande do Sul,
kajianansartor@gmail.com, <https://orcid.org/0000-0003-3211-1381>

Liane Margarida Rockenbach Tarouco, Universidade Federal do Rio Grande do Sul,
liane@penta.ufrgs.br, <https://orcid.org/0000-0002-5669-588X>

Patrícia Fernanda da Silva, Universidade Federal do Rio Grande do Sul,
patriciasilvaufrgs@gmail.com, <https://orcid.org/0000-0001-9408-0387>

Resumo: Este artigo apresenta os resultados de um Mapeamento Sistemático da Literatura (MSL) cujo objetivo é identificar estudos que tratam do uso de Inteligência Artificial Generativa (GenAI) como auxílio ao ensino de programação. A pesquisa foi realizada nas bases de dados internacionais: *ScienceDirect*, *Scopus*, *ACM Digital Library*, *IEEE* e *Web of Science*, buscando publicações no período de janeiro de 2021 a março de 2024, retornando 160 estudos. Após a aplicação dos critérios de inclusão e exclusão, analisou-se 51 artigos, que foram classificados em cinco categorias com relação ao objetivo e o uso da GenAI no ensino de programação: (a) avaliação do desempenho e limitações das ferramentas de GenAI, (b) desenvolvimento de ferramenta, (c) entrevistas/questionários (d) pesquisa exploratória com alunos e, (e) artigos de opinião/revisão.

Palavras-chave: Inteligência artificial generativa, ensino de programação, mapeamento sistemático.

Generative Artificial Intelligence in Programming Teaching: A Systematic Mapping of Literature

Abstract: This article presents the results of a Systematic Literature Mapping (MSL) whose objective is to identify studies that deal with the use of Generative Artificial Intelligence (GenAI) as an aid to teaching programming. The research was carried out in international databases: *ScienceDirect*, *Scopus*, *ACM Digital Library*, *IEEE*, and *Web of Science*, searching for publications from January 2021 to March 2024, returning 160 studies. After applying the inclusion and exclusion criteria, 51 articles were analyzed, which were classified into five categories in relation to the objective and use of GenAI in teaching programming: (a) evaluation of the performance and limitations of GenAI tools, (b) tool development, (c) interviews/questionnaires (d) exploratory research with students and, (e) opinion/review articles.

Keywords: Generative artificial intelligence, teaching programming, systematic mapping.

1. Introdução

Nos últimos anos, o surgimento de tecnologias avançadas de Inteligência Artificial (IA) vem impactando significativamente organizações, sociedades e indivíduos, tornando-se um campo de estudo cada vez mais importante. Com o desenvolvimento crescente e a diversificação das aplicações de IA Generativa, o interesse na sua utilização em diversos segmentos da sociedade tem aumentado rapidamente. Uma das áreas de destaque para o uso de ferramentas de IA é a educação, pois essa tecnologia oferece muitos benefícios potenciais para estudantes e professores.

Segundo Chen *et. al* (2022), a IA está sendo cada vez mais aplicada no campo da educação, oferecendo novas possibilidades para o ensino e a aprendizagem. Pesquisas apontam que abordagens baseadas em IA podem aumentar o envolvimento, a motivação e as habilidades de pensamento crítico dos alunos. Possibilitando reconhecer as diversas preferências e necessidades dos estudantes, as ferramentas de IA Generativa oferecem oportunidades para promover experiências de aprendizagem personalizadas. Utilizando o processamento de linguagem natural e algoritmos de aprendizado de máquina, essas ferramentas podem se adaptar aos estilos de aprendizagem distintos de cada aluno, proporcionando *feedback* personalizado que pode melhorar os resultados educacionais.

IA Generativa é um ramo da inteligência artificial que se concentra na criação de novos conteúdos ou informações. Envolve treinar modelos do tipo *Large Language Models* (LLMs) para gerar resultados originais e realistas, como imagens, música, texto ou vídeos. Os LLMs usam algoritmos complexos e grandes conjuntos de dados para aprender padrões e gerar conteúdo que se assemelha a criados por humanos (Dwivedi *et al.*, 2023).

A IA Generativa pode ser utilizada na educação para criar materiais didáticos interativos e envolventes, pois tem a capacidade de gerar conteúdo de aprendizagem personalizado adaptado às necessidades e preferências individuais dos alunos, melhorando a experiência educacional. Além disso, com o uso de tecnologias de IA Generativa pode-se desenvolver tutores virtuais ou assistentes digitais que fornecem orientação e suporte personalizados aos alunos, assim, as pode-se utilizar os recursos deste tipo de tecnologia para criar ambientes de aprendizagem adaptativos e dinâmicos, que promovam a criatividade e a participação ativa (Lim *et al.*, 2023).

Na educação em programação, pesquisadores demonstraram que LLMs têm capacidade crescente de gerar e interpretar código-fonte por meio de uma interface de linguagem natural (Denny; Prather; Becker, 2024). Assim, a aplicação da IA no ensino de programação apresenta diversas vantagens que podem facilitar o processo educacional. Uma das principais contribuições é a capacidade de oferecer conteúdo personalizado e adaptativo aos alunos, permitindo que eles progridam de acordo com seu próprio ritmo e atendendo às suas necessidades individuais (Holmes; Bialik; Fadel, 2023). Além disso, a IA pode desempenhar um papel importante ao fornecer *feedback* imediato sobre erros e problemas no código, possibilitando que os estudantes corrijam suas falhas de forma rápida e aprendam com suas próprias experiências. Outro benefício significativo é a habilidade dos LLMs de apresentar exemplos de código e explicar de maneira clara e concisa os conceitos de programação, podendo tornar a aprendizagem mais acessível e compreensível.

Sob essa perspectiva, a evolução das ferramentas de programação baseadas em IA está transformando o cenário do ensino de programação, facilitando o acesso ao conhecimento e promovendo um ambiente onde programadores e IA trabalham juntos para aprimorar suas habilidades. Competências como pensamento crítico, resolução de problemas e tomada de decisões são necessárias para maximizar os benefícios dessas ferramentas, pois é importante entender as limitações das ferramentas de IA Generativa e avaliar criticamente seus resultados, pois elas podem gerar código com várias opções adequadas, mas cabe ao programador decidir quais trechos usar e como aplicá-los. Portanto, as disciplinas de programação devem incorporar essas tecnologias, enquanto desenvolvem as competências interpessoais necessárias para futuros profissionais.

Muitas pesquisas estão sendo desenvolvidas no sentido de utilizar a IA Generativa no ensino de programação. Desta forma, este artigo tem por objetivo identificar na literatura as pesquisas que tratam do uso de IA Generativa como auxílio ao ensino de programação. Para tanto, foi realizado um Mapeamento Sistemático da Literatura (MSL), analisando e

classificando estudos existentes, visando analisar o estado atual desse tema, identificar desafios e orientar futuros trabalhos.

O artigo está estruturado da seguinte forma: a Seção 2 apresenta o método utilizado no MSL, seleção e análise dos trabalhos; a Seção 3 apresenta os resultados e discussões; e a Seção 4 discute as considerações finais e as lacunas das pesquisas.

2. Planejamento e condução do mapeamento sistemático

O presente estudo utiliza a técnica de Mapeamento Sistemático da Literatura (MSL) proposto por Petersen *et al.* (2008), que define que o mapeamento sistemático proporciona uma visão geral dos estudos e seus resultados. Assim, um MSL oportuniza uma ampla visão sobre determinada área de pesquisa e por meio dele pode-se identificar evidências de pesquisas que existam e a sua quantidade. Este método é caracterizado por etapas, tais como: (1) definição das questões de pesquisa; (2) definição da estratégia de busca por meio da escolha das palavras-chave e montagem da *string* de busca; (3) seleção dos critérios de inclusão e exclusão dos estudos, selecionando os relevantes; (4) avaliação dos estudos selecionados e extração dos dados; (5) síntese e análise dos resultados.

Neste MSL foi utilizada a plataforma Parsifal¹, uma ferramenta *on-line* projetada para que os pesquisadores possam construir colaborativamente revisões ou mapeamentos sistemáticos. Parsifal foi desenvolvida com base no trabalho de Kitchenham e Charters (2007) permitindo identificar, planejar, conduzir, extrair dados, avaliar e interpretar estudos respondendo questões de pesquisa sobre um determinado tópico ou área.

A pesquisa foi conduzida em 2024, analisando trabalhos publicados no período de janeiro de 2021 até março de 2024. Realizou-se uma busca por artigos no Portal de Periódicos CAPES, em quatro (4) bases internacionais *IEEE Digital Library*, *ScienceDirect*, *Scopus* e *Web of Science*, e acrescentada uma quinta base de dados, a *ACM Digital Library* (atualmente não mais disponibilizada no Portal de Periódicos da CAPES).

2.1 Questões de pesquisa

O principal objetivo deste MSL é identificar os estudos mais relevantes que discutem o uso de Inteligência Artificial Generativa como auxílio no ensino de programação. Para direcionar a pesquisa, foi elaborada a seguinte questão principal (QP): Como a Inteligência Artificial Generativa (GenAI) está sendo usada no auxílio ao ensino de programação? Com base na questão principal, foram elaboradas questões secundárias que visam apoiar a investigação e proporcionar uma compreensão mais aprofundada dos resultados, conforme descritas a seguir: QP1 - Quais são os tipos de estudos e as abordagens utilizadas?; QP2 - Quais as ferramentas de GenAI (LLMs) que estão sendo usadas nas pesquisas para o ensino de programação? QP3 - Quais são os níveis de ensino explorados nos estudos?; QP4 - Quais as linguagens de programação abordadas nas pesquisas?; QP5 - De quais etapas do ensino de programação (correção de código, *feedback*, geração de código, etc) as pesquisas tratam?.

2.2. Definição da estratégia de busca

A expressão de busca permite analisar os resultados obtidos a cada execução da *string* nas bases pesquisadas. As buscas foram construídas com operadores *booleanos OR* e *AND* entre os termos pesquisados e seus sinônimos, com busca avançada e filtro *TITLE-ABS-KEY*, e considerados os estudos e soluções que possuísem os termos “*Generative Artificial Intelligence*”, LLM e “*Programming Education*” e seus respectivos sinônimos (*GenAI*, *Generative AI* / *AI Code Generators*, *Codex*, *ChatGPT*, *Copilot*, *GPT*, *Gemini*,

¹ <https://parsif.al/>

OpenAI / Introductory Programming, Programming Instruction, Programming Learning, Programming Teaching). Desta forma, realizou-se a união das palavras, criando-se assim a *string* de busca apresentada no Quadro 1.

Quadro 1 - String de busca

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

2.3 Critérios de exclusão e inclusão

Na sequência, foram definidos os critérios de inclusão e exclusão, para que, após os resultados das buscas, fosse possível selecionar os artigos que se enquadram no contexto do estudo. Os critérios estão descritos na Quadro 2.

Quadro 2 - Critérios de inclusão e exclusão

Critérios de inclusão	Critérios de exclusão
Deve mencionar IA Generativa, grandes modelos de linguagem ou uma ferramenta específica que utilize essas tecnologias.	Não utiliza IA Generativa
Deve ter aplicabilidade direta ao ensino de programação.	Não se aplica ao ensino de programação
Pelo menos 4 páginas (inclusive)	Menos de 4 páginas
Publicado em ou após 2021	Publicado antes de 2021
Disponível para acesso	Acesso fechado

2.4 Seleção de Base de Dados

A seleção das bases de dados a serem utilizadas na busca dos artigos foi realizada levando-se em consideração a relevância para a área da Computação e Informática na Educação, uma vez que estas estão diretamente relacionadas com a temática do estudo. As seguintes bases foram selecionadas: *ACM Digital Library*: <http://portal.acm.org>; *IEEE Digital Library*: <http://ieeexplore.ieee.org>; *Science Direct*: <http://www.sciencedirect.com>; *Scopus*: <http://www.scopus.com> e *Springer*: <http://www.link.springer.com>. Após a seleção das bases de dados e com a *string* de busca elaborada, o processo de busca foi iniciado, sendo finalizado em abril de 2024. A busca resultou em um total de 160 estudos potenciais, como mostra a Tabela 1.

Tabela 1 - Resultados da Busca

Base	ACM	IEEE	Web of Science	ScienceDirect	Scopus
Quantidade de Estudos	46	12	30	12	60

Os resultados das pesquisas nas bases de dados foram importados para a ferramenta Parsifal, e, com auxílio da mesma, foram excluídos os artigos duplicados. Foram encontrados um total de 70 artigos duplicados, resultando 90 artigos para avaliação. Em seguida, foi realizada a leitura do título e *abstract* de cada artigo visando atender aos critérios de inclusão e exclusão, resultando na exclusão de 39 estudos e restando 51 para a leitura completa. A definição dos critérios de inclusão e exclusão se mostrou válida, pois permitiu filtrar os artigos condizentes com o objetivo desejado, bem como fazer o descarte daqueles que não se encaixavam no foco do estudo.

2.5 Fase Extração de Dados (Data Extraction)

A Fase de Extração de Dados do MSL também foi apoiada pelo Parsifal que auxiliou a definição de questões do Formulário de Extração de Dados (*Data Extraction Form*) da ferramenta. As questões incluídas no formulário permitiram extrair as

informações necessárias para responder às questões de pesquisa. Para essa fase, foi realizada a leitura completa dos 51 artigos selecionados.

Os estudos incluídos neste MSL podem ser acessados pelo [link: https://docs.google.com/document/d/1ulJBu63GfbCdRAHm8cm-6dZ_YV4gnxdWVO0tVdaTL9Y/edit?usp=sharing](https://docs.google.com/document/d/1ulJBu63GfbCdRAHm8cm-6dZ_YV4gnxdWVO0tVdaTL9Y/edit?usp=sharing), onde foi especificado um identificador (ID) para distinguir o artigo durante a escrita das discussões, o nome do artigo e sua respectiva citação e referência. O ID foi iniciado com a letra E de estudo seguido do número arábico para sequenciá-los respectivamente, por exemplo, E1 para o estudo número 1 dos achados da pesquisa.

3. Resultados e discussões

Analisando os artigos selecionados, identificou-se inicialmente um panorama dos estudos quanto ao ano de publicação, o local de realização dos estudos e a base de dados com maior porcentagem de publicação. O ano que mais teve publicações referentes à temática abordada foi o ano de 2023, com 68,6% do quantitativo verificado, seguido pelo ano de 2024 com 25,5% e 2022 com 5,9% das publicações. No ano de 2021 não resultou em nenhuma publicação encontrada, isso se deve ao período de lançamento das ferramentas de IA Generativa que foi a partir de 2022. O ano de 2024 provavelmente ultrapassará 2023 em número de publicações futuramente, visto que só entraram neste MSL publicações realizadas até o mês de março, e esta área de pesquisa está em franco crescimento. Os países que mais tiveram estudos desenvolvidos nesta área foram os Estados Unidos com 19%, seguido da Nova Zelândia com 13,8% dos artigos pesquisados. Quanto às bases que mais publicaram artigos sobre a temática, identificou-se a base *Scopus* com 37,5% e a *ACM Digital Library* com 28,7% das publicações analisadas.

Na sequência, com base na análise dos estudos, buscou-se responder às questões de pesquisa. A seguir, apresentam-se os resultados encontrados para cada questão.

3.1 QP1 - Quais são os tipos de estudos e as abordagens utilizadas?

A partir da leitura, os artigos revisados foram classificados em cinco categorias, com relação ao objetivo e a abordagem de uso da ferramenta de GenAI no estudo: (a) avaliação do desempenho, capacidades e limitações das ferramentas de GenAI, (b) desenvolvimento de ferramenta, (c) entrevistas/questionários (d) pesquisa exploratória com alunos usando uma ferramenta de GenAI no ensino de programação e, (e) artigos de opinião / revisão de literatura. Alguns artigos se enquadram em mais de uma categoria, e nesses casos, foram classificados na categoria mais adequada (Figura 1).

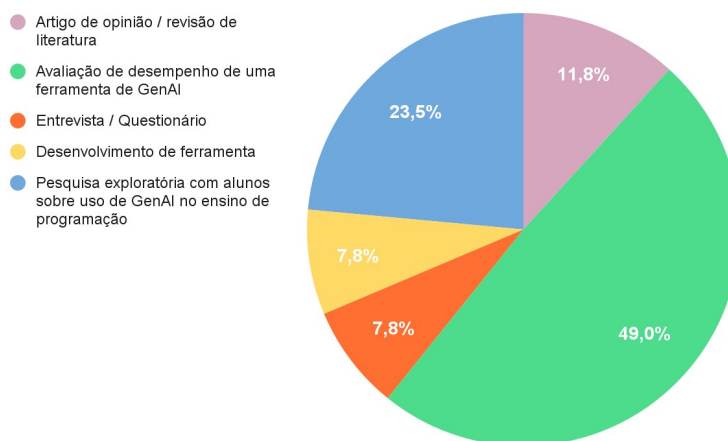


Figura 1 - Classificação dos artigos com relação ao objetivo de uso da ferramenta de GenAI no estudo

A categoria “Avaliação do desempenho, capacidades e limitações das ferramentas de GenAI”, é o maior grupo com 25 trabalhos (49% do total), indicando uma tendência inicial das pesquisas em avaliar as capacidades e limitações atuais das ferramentas de GenAI em contextos de educação em programação. Nesta categoria, grande parte das pesquisas analisou o desempenho dos LLMs na geração de código, em sua maioria para resolver exercícios de programação [E5, E26, E28, E29, E34, E47 e E49]. Outros trabalhos analisaram como as ferramentas respondem a diferentes tipos de exercícios de programação, tais como questões de múltipla escolha [E35] e problemas de *Parsons*²[E20], ou ainda, serem aprovados em questões de provas [E9, E17, E37 e E45], visando demonstrar o desempenho acadêmico da IA em uma disciplina de programação. Em geral, segundo os estudos, os LLMs apresentaram um desempenho equivalente, ou melhor, que o dos alunos iniciantes em programação, pelo menos para tarefas de geração de código de exercícios introdutórios. Já em situações de exercícios mais complexos, tais como questões de múltipla escolha ou problemas de *parson*, os resultados sugerem que estes são mais desafiadores para os LLMs resolverem em comparação com problemas de escrita de código tradicionais. Neste sentido, alguns estudos destacam a importância de considerar que pequenas modificações nas instruções (*prompts*) podem afetar o desempenho dos LLMs.

Além de testar a capacidade de gerar código, muitos estudos se concentraram em analisar os LLMs em atividades de auxílio aos alunos no aprendizado de programação. Estudos como os apresentados em [E4, E8 e E24], trataram de analisar o *feedback* gerado pelo LLM para correção de exercícios de programação. Os estudos [E4 e E24] apresentam resultados que destacam tanto as capacidades quanto as limitações dos LLMs no contexto educacional de programação. Ambos os estudos indicam que esses modelos de IA conseguem detectar e corrigir certos tipos de erros no código, fornecendo explicações textuais e sugestões para melhorias. No entanto, os dois estudos também apontam que os modelos têm dificuldades com erros lógicos e semânticos, além de fornecerem *feedbacks* enganosos ou inadequados em algumas situações. Em [E24], o ChatGPT teve um desempenho razoável na correção de erros de compilação, mas mostrou limitações com múltiplos erros e erros lógicos, além de fornecer informações enganosas em muitas respostas. Em [E4], o GPT-3.5 teve uma taxa de precisão de 73% na identificação correta das submissões de estudantes e forneceu *feedback* adequado em 47% dos casos, mas cometeu erros e sugeriu mudanças desalinhadas com as instruções. Ambos os estudos destacam a necessidade de orientar os alunos sobre as capacidades e limitações desses modelos de IA e desenvolver métodos pedagógicos apropriados para o seu uso eficaz. O estudo apresentado em [E8], teve por objetivo comparar diferentes modelos de LLM para correção e *feedback*, e os resultados do estudo indicam que a utilização de um modelo mais fraco, GPT-3.5, para validar dicas de programação geradas por um modelo mais avançado, GPT-4, resultou em uma precisão de cerca de 95%, equiparando-se à qualidade de tutores humanos. A pesquisa demonstrou a eficácia da geração de *feedback* automatizado para educação em programação, destacando a possibilidade do uso de modelos de IA Generativa para avaliações automáticas e análises de aprendizado. No entanto, em ambos os estudos, não foi realizado um estudo empírico em sala de aula, e uma sugestão é conduzir estudos com alunos reais para avaliar as técnicas propostas a partir da perspectiva dos estudantes, a fim de compreender melhor a eficácia e a usabilidade do *feedback* gerado.

Os estudos apresentados em [E13, E25 e E30], buscaram avaliar a capacidade das ferramentas de GenAI em processos de correção e explicação de código, visando identificar quão eficazes os LLMs são na identificação de problemas nos códigos dos estudantes. Em

² Problemas de *Parsons* são uma forma de atividade de programação na qual os alunos são solicitados a organizar fragmentos de código, os quais compreendem a solução do problema.

[E13] os estudantes avaliaram as explicações de código geradas pelo modelo de linguagem GPT-3 como sendo mais fáceis de entender e com resumos mais precisos do código em comparação com as explicações criadas por outros estudantes.

Há ainda estudos que avaliam os LLMs para a geração de exercícios de programação [E2, E7, E23, E38 e E39], cujo objetivo é auxiliar os professores e instrutores, permitindo a geração de uma ampla variedade de exercícios, evitando a repetição de questões e oferecendo aos alunos uma gama diversificada de desafios, podendo personalizar os mesmos conforme as necessidades e níveis de habilidade dos estudantes, e possibilitando a criação rápida e escalável de exercícios para turmas grandes, sem comprometer a qualidade ou a individualização do ensino.

Na categoria “Pesquisa exploratória com alunos usando uma ferramenta de GenAI no ensino de programação” foram classificados 12 estudos que avaliam o uso de ferramentas de IA Generativa como apoio aos alunos na aprendizagem de programação. Nesta categoria foram realizados estudos empíricos em sala de aula com grupos de alunos que foram incentivados a utilizar as ferramentas de IA Generativa para resolver atividades e exercícios de programação. Nos estudos [E1, E6, E12, E41, E44, E46, E50 e E51], não foram especificadas a abordagem pedagógica para a intervenção. Os alunos experimentaram uma abordagem direta e interativa de aprendizagem e trabalharam na resolução de problemas de programação, sendo observado como ocorreu a interação desses alunos com o LLM. Já nos estudos [E10, E14, E23 e E40], o LLM foi tratado como um colega virtual e foi usada a metodologia de programação em pares.

Os resultados dos estudos com os alunos indicaram, de forma geral, que houve um aumento na motivação e no interesse em aprender programação após utilizar a ferramenta de IA Generativa. Também houve um aumento na autoeficácia na programação, especialmente em tarefas que consistiam em criar partes do código e soluções completas. A experiência contribuiu para o aumento das habilidades de pensamento computacional e autoeficácia em programação. Porém, os estudantes perceberam que, apesar de útil, a IA não substitui o papel de um profissional de computação, mas complementa suas habilidades. Alguns alunos acharam as sugestões da IA úteis, mas enfrentaram desafios na compreensão de partes complexas do código gerado, pois o mesmo nem sempre era facilmente compreensível, especialmente para aqueles com dificuldades. Houve dúvidas sobre a correção do código gerado e a necessidade de instruções claras para obter respostas precisas. O LLM em algumas situações forneceu explicações textuais e versões melhoradas do código, mas teve limitações com erros lógicos e semânticos, oferecendo, por vezes, informações enganosas. Assim, os alunos consideraram o uso de IA Generativa importante para fornecer respostas rápidas, auxílio na depuração de códigos, e estímulo ao pensamento crítico. No entanto, houve preocupações sobre dependência excessiva, dificuldade em compreender respostas complexas e ansiedade profissional.

A categoria “Desenvolvimento de ferramenta” apresenta 4 estudos que tratam do desenvolvimento de aplicações criadas com as APIs de IA Generativa. Em [E3 e E19] descreve-se o desenvolvimento de tutores de programação capaz de gerar *feedback* automático; em [E16] apresenta-se o desenvolvimento de um detector de plágio para código assistido por IA baseado em anomalias de código; e em [E21] relata-se uma ferramenta para geração de exercícios de programação.

Na categoria “Entrevista/questionário” foram identificados 3 estudos [E27, E31 e E42] que realizam entrevistas via questionários com instrutores de programação visando investigar as práticas atuais, preocupações e adaptações planejadas em relação ao uso de ferramentas de geração de código de IA na educação em ciência da computação. Houve divergências sobre a integração dessas ferramentas nos cursos, com alguns instrutores

defendendo sua proibição e outros, propondo incorporá-las para preparar os alunos para o mercado de trabalho. A maioria dos instrutores reconheceu que as ferramentas de IA podem ajudar os estudantes a gerar exemplos de código, fornecer explicações personalizadas e oferecer um ponto de partida quando estiverem em dúvida. No entanto, expressaram preocupações sobre a dependência excessiva dessas ferramentas, destacando a importância do pensamento crítico no aprendizado de programação. Discutiram a necessidade dos alunos entenderem como programar para aproveitar ao máximo as ferramentas de IA, enfatizando a importância do conhecimento prévio em programação. Nesta categoria, o estudo [E11] relata que os alunos foram entrevistados, em que, a maioria demonstrou satisfação com o suporte de programação oferecido pelo LLM, com uma porcentagem significativa de estudantes respondendo positivamente em relação à qualidade das sugestões e assistência recebidas. Em uma das perguntas foi questionado se as sugestões do ChatGPT os ajudavam a resolver problemas de programação, e cerca de 86,7% dos alunos responderam "Sim". Isso indica que a maioria dos estudantes participantes da pesquisa estavam recorrendo ao ChatGPT para obter suporte em suas atividades de programação.

3.2 QP2 - Quais são as ferramentas de GenAI estão sendo usadas nas pesquisas para o ensino de programação?

Os artigos analisados utilizaram uma ampla gama de ferramentas de GenAI, conforme apresentado na Figura 2. Alguns estudos utilizaram mais de uma ferramenta, principalmente os que tratavam de comparações entre elas. O ChatGPT foi a ferramenta mais utilizada, sendo objeto de pesquisa em 35,5% dos estudos, e este número pode ser maior, pois em alguns trabalhos os modelos do LLM (GPT-3, GPT-3.5 ou GPT-4) são citadas ao invés do ChatGPT. Porém, a atualidade desta área de pesquisa e o ritmo acelerado do campo é uma ameaça potencial aos resultados aqui encontrados, pois, por exemplo, o Codex foi utilizado por 9,7% das pesquisas, e a versão mais recente do GPT (GPT-4) por sua vez, lançada em 14 de março de 2023, aparece apenas em 8,1% das publicações. Já o Gemini, LLM da Google, lançado em dezembro de 2023, não é utilizado ainda em nenhum artigo, devido ao recente lançamento.

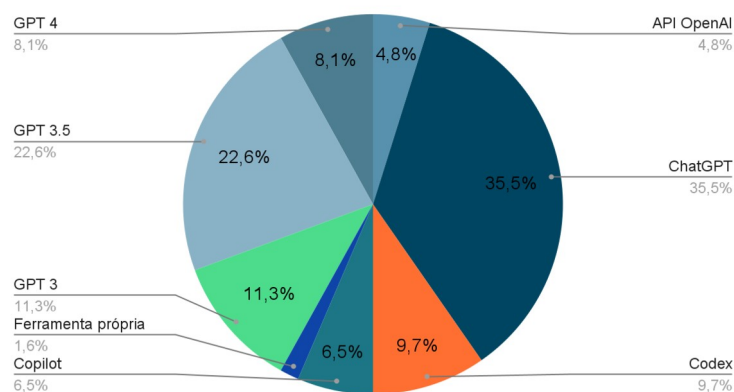


Figura 2 - Ferramentas GenAI

Com relação às ferramentas de GenAI utilizadas nas pesquisas pode-se observar que a nomenclatura ChatGPT apareceu em 22 estudos, seguido de ChatGPT-3.5 em 14 estudos, ChatGPT-3 em 7 pesquisas, ChatGPT-4 em 5 artigos e OpenAI em 3 estudos. Vale ressaltar que ChatGPT é o nome genérico para o *chat*, ChatGPT-3 é a terceira versão do modelo, ChatGPT-3.5 é uma atualização aprimorada do GPT-3, ChatGPT-4 é a quarta e mais avançada versão do modelo e OpenAI é a organização responsável pelo desenvolvimento desses modelos. Diante disso, as diferentes nomenclaturas usadas por alguns autores podem gerar interpretações duvidosas, não ficando claro se foram utilizados nos estudos os *chats*

e/ou a API OpenAI. A ferramenta Codex foi citada em 6 estudos, o Copilot em 4, e em 2 estudos foram utilizadas ferramentas próprias desenvolvidas nas respectivas pesquisas.

3.3 QP3 - Quais são os níveis de ensino explorados nos estudos?

Em relação aos níveis de ensino presentes nos artigos, verificou-se que 33 estudos se referem a pesquisas realizadas com estudantes da Graduação [E1, E2, E4, E6, E10, E11, E12, E13, E14, E15, E17, E19, E21, E22, E23, E24, E25, E26, E29, E30, E32, E34, E36, E37, E41, E44, E45, E46, E47, E48, E49, E50 e E51], 2 pesquisas foram realizadas com estudantes de Pós-graduação [E11 e E35], e 1 artigo menciona que o estudo foi realizado com alunos do Ensino Médio [E40]. Os demais artigos (16 estudos) não especificaram o nível de ensino [E3, E5, E7, E8, E9, E16, E18, E20, E27, E28, E31, E33, E38, E39, E42 e E43]. Pode-se constatar que, no contexto educacional, esse tipo de pesquisa foi amplamente adotado na Graduação, proporcionando aos estudantes uma experiência no uso de estratégias de ensino de programação relevantes para seu desenvolvimento acadêmico, porém percebe-se a carência de estudos em outros níveis de ensino.

3.4 QP4 - Quais as linguagens de programação abordadas nas pesquisas?

A maioria das pesquisas concentra-se nas linguagens de programação *Python* [E2, E3, E8, E9, E16, E17, E21, E28, E30, E34, E35, E37, E39, E40, E47, E49, E50 e E51] e alguns trabalhos envolveram Java [E4, E19, E29, E32, E44 e E46] e C/C++ [E1, E11, E13 e E45]. O fato de Python emergir como a linguagem mais popular nestes estudos não é surpreendente, já que LLMs como *Codex* foram relatados como sendo mais proficientes em *Python*. Outro fator reside no fato de que grande parcela dos estudos tratava de disciplinas introdutórias de programação, sendo que *Python* é amplamente adotada nestas disciplinas para ensinar conceitos de programação a iniciantes.

Em vários artigos [E5, E6, E7, E11, E15, E18, E20, E23, E24, E25, E26, E27, E31, E33, E36, E38, E42, E43 e E48] não há menção específica à linguagem de programação utilizada no estudo. Estes artigos apenas citam que os estudos foram realizados em disciplinas introdutórias de programação, cujo foco principal é geralmente a avaliação do desempenho da ferramenta ou o auxílio aos alunos na resolução de problemas.

3.5 QP5 - De quais etapas do ensino de programação (correção de código, *feedback*, geração de código, etc) as pesquisas tratam?

As ferramentas de IA Generativa podem ser usadas em diferentes etapas no ensino de programação. Os artigos analisados utilizaram o LLM para auxiliar em uma ou mais destas etapas, conforme apresentado na Figura 3.

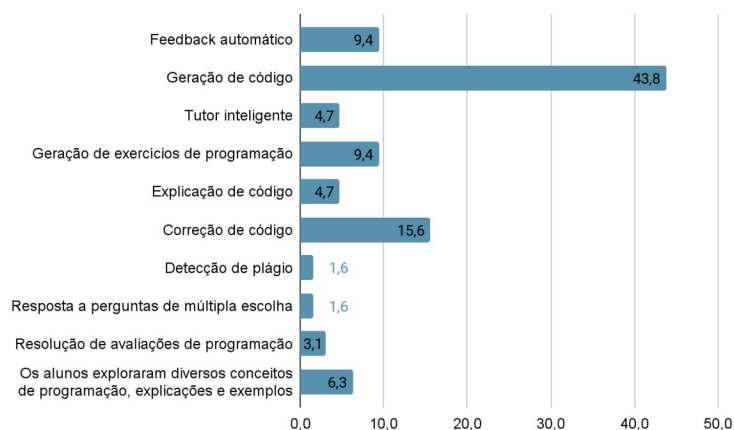


Figura 3 - Resultados (%) das etapas do ensino de programação

Em sua grande maioria, 43,8% dos estudos, o foco foi a geração de código, tanto para estudos que avaliaram as ferramentas quanto para estudos empíricos e de desenvolvimento. A etapa de Correção de código também teve destaque em 15,6% dos artigos analisados. E as etapas de *Feedback* automático e Geração de exercícios de programação aparecem ambas com 9,4%. Esse resultado indica que, em um panorama inicial, a preocupação dos pesquisadores está em avaliar se as ferramentas de IA Generativa são capazes de gerar códigos que respondam os exercícios e atividades de programação de forma eficiente e se tais ferramentas podem ser usadas para criar exercícios de programação para serem usados pelos professores em suas aulas. Já em se tratando dos estudos empíricos com os alunos, as etapas mais exploradas nos estudos foram os processos de correção de código e *feedback* automático, visando validar o uso das ferramentas de IA generativa como apoio do processo de aprendizagem dos alunos.

4. Conclusões

A IA Generativa pode auxiliar no ensino de programação por oferecer suporte personalizado aos estudantes, adaptando-se ao nível de conhecimento e ritmo de aprendizagem de cada um. Ao interagir com o modelo os alunos recebem *feedback* imediato sobre seus códigos, sugestões de melhorias e explicações detalhadas, permitindo um aprendizado mais interativo e adaptativo, podendo explorar, experimentar e aprender a programar de forma mais eficaz, atendendo às suas necessidades individuais (Lim *et al.*, 2023; Holmes; Bialik; Fadel, 2023).

O MSL apresentado neste artigo pode contribuir na identificação de tendências, lacunas e áreas emergentes, proporcionando uma base sólida para entender o panorama geral desta área e direcionar novas pesquisas mais focadas. Foram identificados que: a avaliação do desempenho, capacidades e limitações das ferramentas de GenAI, é uma tendência inicial das pesquisas (49% do total) pois busca avaliar as capacidades e limitações atuais dos LLMs em contextos de educação em programação; o ChatGPT e suas versões (GPT-3, GPT-3.5 ou GPT-4) são as ferramentas mais utilizada como objeto de pesquisa dos estudos; o nível de ensino que mais aparece nos estudos é a Graduação com 64,70%; e as linguagens de programação mais citadas são *Python*, *Java* e *C/C++*.

Ao fazer a análise dos estudos apresentados nos 51 artigos selecionados, pode-se identificar as potencialidades do uso das tecnologias de IA Generativa no ensino de programação. Os dados obtidos revelam que essas ferramentas são capazes de resolver atividades e exercícios de programação de forma satisfatória e auxiliar os alunos, promovendo habilidades relacionadas à autoeficácia de programação, motivação e engajamento. Com a evolução das ferramentas de programação baseadas em IA, é necessário considerar como essas inovações estão impactando as competências e habilidades necessárias para os estudantes de programação. A integração dessas ferramentas nos cursos de programação pode melhorar a proficiência dos alunos, mas ainda é importante garantir que eles desenvolvam habilidades como raciocínio lógico, resolução de problemas e tomada de decisões.

Em relação às lacunas identificadas que demandam investigações futuras, muitos estudos apontam para a necessidade da replicação das pesquisas em ambientes educacionais reais. É igualmente importante explorar mais a fundo as limitações das ferramentas de programação baseadas em IA, bem como desenvolver estratégias para ajudar os alunos a reconhecer quando é apropriado utilizar essas ferramentas e quando é necessário recorrer a outras abordagens. Explorar essas lacunas e direcionar futuras pesquisas para essas áreas pode contribuir significativamente para o avanço do conhecimento sobre a educação de programadores em um cenário cada vez mais influenciado pela IA.

Agradecimentos

O presente trabalho foi realizado com apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), por meio de bolsa de pós-graduação concedida à bolsista Teresinha Letícia da Silva.

Referências Bibliográficas

CHEN, X.; ZOU, D.; XIE, H. Two Decades of Artificial Intelligence in Education: Contributors, Collaborations, Research Topics, Challenges, and Future Directions. *Educational Technology & Society*, v. 25, n. 1, p. 28–47, 2022.

DENNY, P.; PRATHER, J.; BECKER, B. A. Computing Education in the Era of Generative AI. *Communications of the ACM*, v. 67, n. 2, p. 56–67, 2024. Disponível em: <https://dl.acm.org/doi/10.1145/3624720>. Acesso em: mai. 2024.

DWIVEDI, Y. K.; SHARMA, A.; RANA, N. P. Evolution of artificial intelligence research in Technological Forecasting and Social Change: Research topics, trends, and future directions. *Technological Forecasting and Social Change*, v. 192, p. 122579, 2023. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0040162523002640>. Acesso em: mai. 2024.

HOLMES, W.; BIALIK, M.; FADEL, C. Artificial intelligence in education. In: STÜCKELBERGER, C.; DUGGAL, P. (org.). *Data ethics : building trust : how digital technologies can serve humanity*. [S. l.]: Globethics Publications, 2023. p. 621–653. Disponível em: <http://hdl.handle.net/20.500.12424/4276068>. Acesso em: jun. 2024.

KITCHENHAM, Barbara; CHARTERS, Stuart. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

LIM, W. M.; GUNASEKARA, A.; PALLANT, J. L. Generative AI and the future of education: Ragnarök or reformation? A paradoxical perspective from management educators. *The International Journal of Management Education*, v. 21, n. 2, p. 100790, 2023. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1472811723000289>. Acesso em: mai. 2024.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. In: PROCEEDINGS OF THE 12th INTERNATIONAL CONFERENCE ON EVALUATION AND ASSESSMENT IN SOFTWARE ENGINEERING. *Anais...*: EASE'08, GBR: BCS Learning & Development Ltd., 2008, p. 68-77.