

Inteligência Artificial nos Cursos de Programação: Transformando a Aprendizagem e Desafiando os Paradigmas do Ensino

Ycaro Yan Lourenço Santos ¹

Samuel Ryan da Fonseca Anunciação ²

José Wilker Pereira Luz ³

RESUMO

A integração da Inteligência Artificial (IA) nos cursos de programação está revolucionando o ensino e a aprendizagem, oferecendo ferramentas como assistentes de código (GitHub Copilot) e plataformas adaptativas (Coursera), que democratizam o acesso ao conhecimento através de feedback instantâneo, correção de erros em tempo real e trilhas personalizadas. Essas tecnologias permitem que estudantes de diferentes níveis superem barreiras iniciais, focando em desafios complexos, como design de algoritmos, enquanto sistemas adaptativos ajustam o ritmo de acordo com o desempenho individual. No entanto, a IA também desafia paradigmas tradicionais: educadores precisam migrar de aulas expositivas para enfoques em pensamento crítico, ética e criatividade, já que tarefas rotineiras (como sintaxe) são automatizadas. Avaliações estão sendo reinventadas com projetos práticos e discussões sobre código gerado por IA para combater, por exemplo, a dependência excessiva. O objetivo desse estudo é analisar e refletir sobre o impacto da integração da inteligência artificial na educação de programação. Olhando para o futuro, a educação de programação deverá equilibrar eficiência da IA com mentoria humana, adotando ambientes imersivos e currículos dinâmicos, preparando profissionais capazes de colaborar com a IA, não apenas depender dela. Embora a IA otimize a aprendizagem e personalize trajetórias, seu sucesso depende de preservar a essência pedagógica — formar desenvolvedores críticos, éticos e inovadores, aptos a resolver problemas complexos em um mundo cada vez mais moldado pela inteligência artificial. Assim, a revolução tecnológica exige não apenas adaptação, mas uma redefinição consciente dos objetivos educacionais, garantindo que o avanço técnico ande lado a lado com o desenvolvimento humano e social.

Palavras-chave: Inteligência Artificial, Ensino de programação, Pedagogia adaptativa, Ética tecnológica, Inovação educacional.

¹ Graduando do Curso de Bacharelado em Ciência da Computação pelo Instituto Federal de Educação, Ciência e Tecnologia do Maranhão - IFMA Campus Caxias, ycarolourenco@acad.ifma.edu.br;

² Graduando do Curso de Bacharelado em Ciência da Computação pelo Instituto Federal de Educação, Ciência e Tecnologia do Maranhão - IFMA Campus Caxias, samuel.ryan@acad.ifma.edu.br;

³ Professor Orientador: Mestre em Engenharia da Computação e Sistemas. Professor do Instituto Federal de Educação, Ciência e Tecnologia do Maranhão - IFMA Campus Caxias, josewilkerluz@ifma.edu.br.

INTRODUÇÃO

Nas últimas décadas, a Inteligência Artificial emergiu de um campo de pesquisa especializado para se tornar uma força transformadora em praticamente todos os setores da sociedade, e a educação em programação não é exceção. O advento de ferramentas de IA generativa, notadamente assistentes de codificação como o GitHub Copilot, está redefinindo o processo de desenvolvimento de software, automatizando tarefas rotineiras e aumentando significativamente a produtividade [3]. Essa mudança tecnológica impõe uma reflexão profunda sobre como a próxima geração de desenvolvedores deve ser formada.

Tradicionalmente, os cursos de programação dedicavam tempo substancial ao domínio da sintaxe, à depuração manual e à resolução de problemas elementares. Contudo, a IA assume progressivamente essas responsabilidades, permitindo que os estudantes superem a barreira inicial da sintaxe e se concentrem mais rapidamente em desafios de ordem superior, como o design de algoritmos, a arquitetura de sistemas e o pensamento computacional [5]. Além disso, plataformas de aprendizagem adaptativa, impulsionadas por IA, oferecem trilhas de conhecimento personalizadas, feedback instantâneo e correção de erros em tempo real, democratizando o acesso ao conhecimento e ajustando o ritmo de acordo com o desempenho individual do aluno.

No entanto, a integração da IA no ensino de programação não é isenta de desafios. A facilidade com que o código pode ser gerado levanta questões cruciais sobre a dependência excessiva dos alunos em relação à ferramenta, a integridade acadêmica e, fundamentalmente, a necessidade de reinventar os métodos de avaliação [4]. O paradigma tradicional, focado na memorização de comandos e na entrega de código funcional, torna-se obsoleto. O novo foco pedagógico deve migrar para o pensamento crítico, a ética no uso de código gerado por terceiros (ou IA) e a criatividade na solução de problemas complexos que a IA ainda não consegue abordar de forma autônoma [2].

O objetivo desse estudo é analisar e refletir sobre o impacto da integração da inteligência artificial na educação de programação, examinando tanto as potencialidades transformadoras (personalização, eficiência) quanto os desafios pedagógicos e éticos (mudança curricular, avaliação).

METODOLOGIA

O presente estudo adota uma abordagem qualitativa e exploratória, fundamentada em revisão bibliográfica e análise documental, com o propósito de compreender o impacto da Inteligência Artificial (IA) na educação de programação. A metodologia foi delineada para articular produções científicas nacionais e internacionais recentes (publicadas entre 2020 e 2025), permitindo identificar tendências contemporâneas, potencialidades de ferramentas específicas (como o GitHub Copilot) e os desafios pedagógicos emergentes.

A pesquisa caracteriza-se como um estudo de natureza descritiva e analítica, que busca interpretar e discutir as evidências teóricas acerca da transformação dos currículos e das práticas de ensino de programação frente à ascensão da IA generativa. A abordagem qualitativa é apropriada para investigar as percepções, os paradigmas e as implicações éticas da tecnologia no contexto educacional, alinhando-se ao objetivo de refletir sobre a redefinição dos objetivos de formação de desenvolvedores.

A coleta de dados foi realizada por meio de uma análise bibliográfica sistematizada, com seleção de publicações indexadas em bases de dados científicas (como Scielo, UFRGS Lume, SBC Sol) e artigos de referência que abordam:

- 1 O uso e o impacto do GitHub Copilot e ferramentas similares no aprendizado de programação [4][5].
- 2 A aplicação de sistemas adaptativos e feedback automatizado por IA em plataformas de ensino.
- 3 A necessidade de atualização curricular e a migração do foco da sintaxe para o pensamento crítico e a ética [2].

A análise dos materiais seguiu uma interpretação crítica, focada em três eixos temáticos: Otimização da Aprendizagem (personalização, feedback), Desafios Pedagógicos (avaliação, dependência) e Reorientação Curricular (ética, pensamento crítico). Essa sistematização permitiu compreender de que forma a IA está sendo incorporada e quais estratégias se mostram essenciais para formar profissionais que saibam colaborar com a IA, e não apenas depender dela, garantindo que o avanço técnico ande lado a lado com o desenvolvimento humano e social.

REFERENCIAL TEÓRICO

O pensamento computacional, historicamente, tem sido a base do ensino de programação, focando na decomposição de problemas, no reconhecimento de padrões, na abstração e na formulação de algoritmos [6]. A IA, particularmente a Inteligência Artificial Generativa, introduz um novo elemento nesse ecossistema, automatizando a fase de formulação do algoritmo e, em muitos casos, a própria codificação.

A IA como Ferramenta de Otimização da Aprendizagem

A aplicação da IA na educação de programação pode ser categorizada em dois eixos principais: sistemas de tutoria inteligente e assistentes de codificação.

Sistemas de Tutoria Inteligente e Aprendizagem Adaptativa: Plataformas como *Coursera*, *edX* e sistemas internos de universidades utilizam IA para analisar o desempenho do aluno, identificar lacunas de conhecimento e ajustar dinamicamente o currículo e o ritmo de estudo. Essa personalização, baseada em modelos preditivos, otimiza o tempo de aprendizado e garante que o estudante receba o conteúdo mais relevante no momento certo. O feedback instantâneo sobre a correção de código e a sugestão de recursos adicionais são exemplos de como a IA atua como um tutor incansável, permitindo que o professor se dedique a questões conceituais mais complexas [1].

Assistentes de Codificação (GitHub Copilot e Similares): O GitHub Copilot, baseado no modelo Codex da OpenAI, é o exemplo mais proeminente de IAG no ensino de programação. Sua capacidade de gerar blocos de código, funções inteiras e até mesmo documentação a partir de comentários em linguagem natural ou de código incompleto tem um impacto direto na produtividade e na experiência do aluno. Estudos como os de Lira e Santos Neto (2024) [3] e Velho (2025) [4] indicam que alunos que utilizam o Copilot resolvem problemas mais rapidamente.

Desafios e a Transformação do Paradigma Educacional

Apesar dos benefícios de otimização, a IA impõe desafios que exigem uma redefinição dos objetivos educacionais. O principal deles reside na necessidade de migrar o foco do "saber fazer" (codificar) para o "saber pensar" (criticar e aplicar).

O Desafio da Dependência e da Integridade Acadêmica

A facilidade de geração de código levanta a preocupação com a dependência cognitiva. Se o aluno não precisa mais lutar contra a sintaxe ou depurar erros simples, ele pode não desenvolver a memória muscular e a capacidade de raciocínio lógico necessárias para resolver problemas sem o auxílio da IA. Velho (2025) [4] investiga esse impacto, sugerindo que, embora a produtividade aumente, a compreensão conceitual pode ser comprometida se o uso não for mediado.

Além disso, a integridade acadêmica é posta em questão. A avaliação tradicional, baseada em exercícios de codificação, perde a validade, pois o código pode ser facilmente gerado por IA. Isso exige que os educadores reinventem as avaliações, focando em:

- Projetos práticos complexos que exijam criatividade e integração de sistemas.
- Discussões e defesas orais sobre o código gerado, exigindo que o aluno demonstre compreensão do porquê e como a IA chegou àquela solução.
- Ênfase na arquitetura e no design em detrimento da mera implementação.

A Reorientação Curricular: Ética, Crítica e Colaboração

O novo paradigma educacional, conforme defendido por Azambuja (2024) [2], exige que o currículo de programação incorpore o pensamento crítico e a ética como pilares centrais. O desenvolvedor do futuro não apenas codifica, mas também avalia a qualidade, a segurança e as implicações éticas do código gerado pela IA.

A formação deve preparar o aluno para ser um colaborador da IA, e não um mero usuário. Isso implica em:

Validação Crítica: Ensinar o aluno a questionar e testar o código da IA, reconhecendo que ele pode conter erros, vulnerabilidades ou vieses.

Ética do Código: Debater a propriedade intelectual, a segurança de dados e os vieses algorítmicos presentes nos modelos de Inteligência Artificial Generativa.

Foco em Problemas Complexos: Redirecionar o tempo de aula, antes gasto com sintaxe, para o design de soluções inovadoras e a discussão de arquiteturas complexas.

Essa transformação exige que o professor migre de transmissor de conteúdo para mentor e mediador, orientando o aluno no uso ético e crítico das ferramentas de IA.

RESULTADOS E DISCUSSÃO

A análise bibliográfica sistematizada permitiu identificar tendências, convergências e lacunas nas discussões acadêmicas sobre o impacto da Inteligência Artificial na educação de programação. As obras analisadas foram agrupadas em três eixos temáticos, que sintetizam as principais contribuições e desafios observados na literatura contemporânea.

Eixo 1: Otimização e Personalização da Aprendizagem

Os estudos mais recentes apontam que a IA tem desempenhado um papel central na personalização do aprendizado e na otimização do processo de ensino de programação. Pesquisas como as de Lira e Santos Neto (2024) e Silva (2023) [1][3] destacam que sistemas de tutoria inteligente e plataformas adaptativas conseguem ajustar o conteúdo e o ritmo conforme o desempenho individual do aluno, promovendo uma aprendizagem mais eficiente e contínua.

A convergência dos resultados sugere que a IA atua como um agente democratizador, reduzindo as barreiras iniciais da codificação e ampliando o acesso a recursos educacionais personalizados. No entanto, a literatura também indica divergências quanto à profundidade do aprendizado alcançado. Enquanto alguns autores ressaltam ganhos significativos na velocidade e precisão das resoluções, outros alertam que a automatização excessiva pode limitar o desenvolvimento da autonomia cognitiva e da criatividade do aluno.

Os assistentes de codificação, como o GitHub Copilot, emergem como o exemplo mais emblemático dessa transformação. Estudos [3][4] indicam aumento da produtividade e do engajamento dos estudantes, mas também revelam uma dependência crescente das sugestões automatizadas. Assim, o desafio não reside em adotar ou rejeitar essas ferramentas, mas em compreender como integrá-las pedagogicamente sem comprometer a formação do raciocínio computacional.

Eixo 2: Desafios Pedagógicos – Dependência Cognitiva e Crise da Avaliação

A literatura converge em apontar a dependência cognitiva como um dos principais riscos da integração da IA na educação de programação. Pesquisas como as de Velho (2025) [4]

demonstram que, ao reduzir a necessidade de depuração e de compreensão sintática, a IA pode enfraquecer o desenvolvimento de habilidades de raciocínio lógico e de resolução autônoma de problemas. Essa constatação é reforçada por estudos que observam uma lacuna crescente entre execução técnica e entendimento conceitual.

Outro ponto recorrente é a obsolescência dos métodos avaliativos tradicionais. Exercícios de codificação e tarefas padronizadas tornam-se insuficientes para mensurar o aprendizado real, uma vez que podem ser facilmente resolvidos com o auxílio de ferramentas generativas. Nesse cenário, autores como Azambuja (2024) [2] defendem a necessidade de avaliar o processo, e não apenas o produto final. As principais alternativas sugeridas incluem: Defesas orais e apresentações reflexivas sobre o código gerado; avaliações centradas na arquitetura e na lógica de projeto e análise crítica de códigos produzidos por IA. Essas abordagens buscam garantir que o estudante demonstre compreensão conceitual e possa justificar as decisões técnicas adotadas, promovendo assim uma aprendizagem mais reflexiva e ética.

Reorientação Curricular – Ética, Crítica e Colaboração com a IA

Os resultados indicam um consenso sobre a necessidade de reconfigurar os currículos de programação para incluir competências éticas e críticas relacionadas ao uso da IA. Conforme argumenta Azambuja (2024) [2], o objetivo não deve ser formar apenas codificadores eficientes, mas profissionais capazes de avaliar, interpretar e colaborar criticamente com sistemas inteligentes. Nesse contexto, a literatura destaca três dimensões fundamentais dessa reorientação:

Validação crítica do código, desenvolvendo a capacidade de questionar e testar as soluções propostas pela IA;

Ética do código, abordando temas como propriedade intelectual, segurança de dados e vieses algorítmicos;

Foco em problemas complexos, estimulando o uso da IA como meio de exploração criativa e não como substituto do raciocínio humano.

Além disso, observa-se uma transformação no papel docente: o professor deixa de ser o transmissor exclusivo do conhecimento técnico para se tornar um mentor e mediador do pensamento crítico. Essa mudança exige formação continuada e um reposicionamento

pedagógico que valorize a autonomia intelectual e o diálogo sobre o impacto social e ético da IA.

CONSIDERAÇÕES FINAIS

A integração da Inteligência Artificial nos cursos de programação representa um ponto de inflexão que transcende a mera adoção de novas tecnologias. Ferramentas como o GitHub Copilot e os sistemas de aprendizagem adaptativa otimizam o processo de aquisição de conhecimento, reduzindo a curva de aprendizado inicial e permitindo que os estudantes se dediquem mais rapidamente a problemas de ordem superior, como o design de software e a arquitetura de sistemas. A IA atua, portanto, como um poderoso agente de democratização e personalização do ensino.

Entretanto, essa otimização impõe a redefinição dos paradigmas educacionais. O foco do ensino deve migrar da sintaxe para o pensamento crítico, a validação de código e a ética profissional. O desafio central para as instituições de ensino reside em combater a dependência cognitiva, garantindo que o aluno compreenda o porquê por trás do código gerado pela IA, e não apenas o como.

Para o futuro, a educação em programação deve buscar um equilíbrio entre a eficiência da IA e a insubstituível mentoria humana. Os currículos precisam ser dinâmicos, incorporando discussões sobre vieses algorítmicos, segurança e propriedade intelectual. As avaliações devem ser reinventadas, priorizando projetos complexos e defesas orais que exijam do estudante a demonstração de compreensão conceitual e a capacidade de colaborar de forma crítica com a IA.

Em última análise, a revolução tecnológica exige não apenas adaptação, mas uma redefinição consciente dos objetivos educacionais. O sucesso da IA no ensino de programação não será medido pela velocidade com que o código é escrito, mas sim pela capacidade da academia de formar desenvolvedores críticos, éticos e inovadores, aptos a resolver os problemas complexos de um mundo cada vez mais moldado pela inteligência artificial.

REFERÊNCIAS

- [2] Azambuja, C. C. (2024). Novos desafios para a educação na Era da Inteligência Artificial. Scielo
- [3] Lira, W. A. L., & Santos Neto, P. A. (2024). Uma análise do uso de ferramentas de geração de código por alunos de computação. Anais do Congresso Brasileiro de Educação em Computação (EDUCOMP).
- [4] Velho, M. A. B. (2025). Investigando o impacto da inteligência artificial generativa no aprendizado em programação. Lume UFRGS.
- [5] Silva, E. L. R. (2025). O IMPACTO DA INTELIGÊNCIA ARTIFICIAL NA VIDA DOS DESENVOLVEDORES DE SOFTWARE. Ciências em Sintonia: Explorando Conexões entr
- [6] WING, J. M. Computational thinking. Communications of the ACM, v. 49, n. 3, 2006.