



**VIBE CODING NA EDUCAÇÃO: UMA REVISÃO DE ESCOPO SOBRE A
INCORPORAÇÃO DO PARADIGMA EM CONTEXTOS DE ENSINO E
APRENDIZAGEM**

**VIBE CODING IN EDUCATION: A SCOPING REVIEW ON THE
INCORPORATION OF THE PARADIGM IN TEACHING AND LEARNING
CONTEXTS**

Gabriel Felipe Martins Magalhães¹

¹CEFET-MG, Belo Horizonte, Minas Gerais, magalhaesgfm@yahoo.com; ORCID
0009-0002-8609-2577

RESUMO

A formalização do termo vibe coding por Andrej Karpathy, em 2025, deu visibilidade a uma prática de produção de artefatos digitais por meio de diálogo em linguagem natural com sistemas de Inteligência Artificial Generativa, sem escrita direta de código. A rápida disseminação da prática em contextos educacionais instaurou tensão entre seu potencial de democratização do desenvolvimento de software e os riscos epistêmicos do uso sem formação técnica prévia. Este estudo mapeia como a literatura científica recente caracteriza a incorporação do vibe coding em práticas de ensino e aprendizagem, identificando convergências, tensões e lacunas no debate. Adotou-se revisão de escopo segundo Mak e Thomas (2022), com busca conduzida em março de 2026. Foram identificados 14 estudos, dos quais 11 compuseram a amostra final após exclusão de preprints sem revisão por pares. Os resultados indicam que o vibe coding reduz barreiras técnicas de entrada no desenvolvimento de software e viabiliza a produção de aplicações funcionais por estudantes e educadores sem formação em engenharia, com relatos de estados de engajamento associados à teoria do flow. Ao mesmo tempo, os estudos apontam que a fluência conversacional com a IA não equivale à compreensão técnica do código gerado e que a confiança excessiva nas saídas dos modelos pode mascarar falhas de segurança e inconsistências lógicas. Conclui-se que o potencial formativo do paradigma depende do letramento prévio em Inteligência Artificial, da revisão humana qualificada e da integração de dimensões éticas à concepção das atividades pedagógicas.

Palavras-chave: vibe coding, revisão de escopo, letramento em inteligência artificial, educação em programação, inteligência artificial generativa.

1 INTRODUÇÃO

A formalização do termo *vibe coding* por Andrej Karpathy, em fevereiro de 2025, deu visibilidade pública a uma prática que já se consolidava nos ambientes profissionais de desenvolvimento de *software*: a produção de artefatos digitais por meio de diálogo conversacional com sistemas de Inteligência Artificial Generativa (IAG), sem que o desenvolvedor escreva código diretamente. Na formulação inicial, a prática foi descrita como um modo de programar em que se "vê, fala e executa" em fluxo contínuo com a IA deslocando o desenvolvedor da escrita sintática para a expressão de intenções em linguagem natural. Estudos posteriores sistematizaram o conceito como paradigma de desenvolvimento de *software* no qual humanos e IAG se engajam em fluxo colaborativo para co-criar artefatos por meio de diálogo em linguagem natural, com a mediação da intenção do desenvolvedor migrando da instrução determinística para a inferência probabilística. Esse deslocamento redistribui o trabalho epistêmico entre humanos e máquinas, com a expertise migrando da execução técnica para a orquestração colaborativa, da maestria sintática para a decomposição de problemas e a avaliação crítica das saídas geradas.

A repercussão do termo extrapolou rapidamente o domínio profissional e alcançou os contextos educacionais. As buscas pela expressão *vibe coding* aumentaram 6700% em meados de 2025, e o termo passou a integrar a lista de expressões emergentes do dicionário Merriam-Webster no mesmo ano, evidência da rapidez com que a prática se incorporou ao vocabulário cotidiano de desenvolvedores, educadores e estudantes. No campo educacional, a discussão se articula com debates sobre o lugar da IAG nos processos de ensino e aprendizagem, sobre a redistribuição de competências docentes e discentes e sobre as condições nas quais a mediação algorítmica favorece ou compromete a formação técnica e ética dos estudantes. Estudos iniciais indicam que a prática permite a usuários sem formação em engenharia de *software* produzir aplicações funcionais em períodos curtos, com baixo custo operacional e a partir de poucos prompts em linguagem natural, o que sugere potencial pedagógico em áreas nas quais a programação atua como ferramenta de domínio aplicado, e não como objeto disciplinar central.

Diante desse cenário, instala-se uma tensão analítica que orienta a presente pesquisa. De um lado, há indícios convergentes de que o *vibe coding* reduz barreiras técnicas de entrada na produção de *software*, amplia a participação de estudantes historicamente afastados do desenvolvimento computacional e favorece estados de

engajamento associados à teoria do *flow*. De outro, evidências empíricas apontam que estudantes tendem a contornar a leitura e a compreensão do código gerado, a especificar intenções de forma imprecisa e a desenvolver percepção equivocada de competência técnica. Acrescentem-se a essas constatações as limitações estruturais dos próprios modelos de linguagem, que otimizam para a plausibilidade local em detrimento da consistência global, com produção de código frequentemente inseguro e sujeito a falhas de manutenção em sessões prolongadas.

A questão que orienta esta pesquisa pode ser formulada nos seguintes termos: como o *vibe coding* tem sido incorporado em contextos educacionais e quais são as condições pedagógicas, cognitivas e éticas que sustentam ou limitam seu uso formativo? O objetivo geral consiste em mapear, por meio de revisão de escopo, como a literatura científica recente caracteriza a apropriação do *vibe coding* em práticas de ensino e aprendizagem, identificando convergências, tensões e lacunas no debate. Como objetivos específicos, propõem-se: (I) sistematizar as definições e atributos do paradigma a partir de fontes primárias; (II) examinar os contextos educacionais nos quais a prática tem sido implementada; (III) discutir os achados à luz dos referenciais de letramento em IA e da redistribuição cognitiva entre humanos e modelos de linguagem.

A justificativa do estudo decorre do descompasso entre a rápida disseminação da prática e a ainda incipiente produção acadêmica que a examine sob perspectivas pedagógicas. Embora a literatura sobre programação assistida por IA tenha se expandido, o *vibe coding* constitui categoria distinta, com implicações próprias para o desenho curricular, para a formação docente e para a definição das competências a serem desenvolvidas pelos estudantes. A consolidação de evidências sobre essa prática contribui para subsidiar decisões institucionais e didáticas em um campo no qual a velocidade de incorporação tecnológica tem se sobreposto à reflexão pedagógica.

O artigo está organizado em cinco seções, além desta introdução. A segunda apresenta o referencial teórico, com a caracterização do *vibe coding*, dos Modelos de Linguagem de Grande Escala (LLMs), do letramento em Inteligência Artificial e da programação assistida por IA. A terceira descreve os procedimentos metodológicos da revisão de escopo. A quarta seção apresenta os resultados, sistematizados em tabela comparativa, e desenvolve a discussão crítica dos achados, organizada em torno dos eixos da reconfiguração cognitiva e do letramento em IA, da democratização do acesso e dos limites epistêmicos do paradigma. A quinta seção registra as considerações finais, retoma as contribuições do estudo e indica direções para pesquisas futuras.

2 REFERENCIAL TEÓRICO

2.1 *Vibe coding*: definição e caracterização

O termo *vibe coding* foi formalizado por Andrej Karpathy em 2025 para nomear uma prática que já se consolidava nos ambientes de desenvolvimento de *software*: a criação de artefatos digitais por meio de diálogo conversacional com sistemas de Inteligência Artificial Generativa (IAG), sem que o desenvolvedor necessite escrever código diretamente. Na formulação original, a prática é descrita como um modo de "ver, dizer e executar" em fluxo contínuo com a IA estabelecendo um ciclo iterativo no qual intenções são expressas em linguagem natural e traduzidas em código executável pelo modelo.

Meske *et al.* (2025) propõem a definição mais sistematizada do conceito na literatura: *vibe coding* é um paradigma de desenvolvimento de *software* no qual humanos e IAG se engajam em fluxo colaborativo para co-criar artefatos de *software* por meio de diálogo em linguagem natural, deslocando a mediação da intenção do desenvolvedor da instrução determinística para a inferência probabilística. Essa definição desloca o eixo analítico da produtividade técnica individual para o processo de mediação de intenção, entendido como o mecanismo pelo qual objetivos conceituais são traduzidos em representações executáveis por sistemas computacionais. Os autores identificam cinco atributos que caracterizam a prática: expressão de intenção orientada a objetivos, interação dialógica acelerada, abstração de implementação, refinamento semântico dinâmico e estados de fluxo co-criativo.

O paradigma se distingue de outras formas de programação assistida por IA por alterar a posição do desenvolvedor no processo de produção. Enquanto ferramentas convencionais de auxílio ao código, como o GitHub Copilot, pressupõem um desenvolvedor imerso na base de código que aceita ou rejeita sugestões geradas pela IA, o *vibe coding* inverte essa relação: o desenvolvedor atua no nível conversacional, orientando por intenções de alto nível, enquanto a IA assume a responsabilidade pela implementação. Essa inversão implica redistribuição do trabalho epistêmico entre humanos e máquinas, com deslocamento da expertise da execução técnica para a orquestração colaborativa.

Do ponto de vista da qualidade do código gerado, Mitchell e Shaaban (2025) argumentam que a natureza iterativa e prompt-a-prompt do *vibe coding* é intrinsecamente

frágil, pois os modelos otimizam a plausibilidade local em vez da consistência global, o que resulta em um fenômeno de *constraint-reconciliation decay*: à medida que as sessões se prolongam e as dependências se acumulam, as taxas de sucesso na manutenção da coerência do código diminuem. Evidências empíricas reforçam essa preocupação: análise de repositórios em larga escala revelou que apenas 28% dos programas gerados pelo modelo mais avançado produz código seguro e correto simultaneamente. Apesar desses riscos, Meyer (2026) demonstrou que aplicações funcionais de análise de dados ônicos podem ser geradas em menos de dez minutos com apenas quatro prompts em linguagem natural, a um custo operacional inferior a dois dólares, o que evidencia a viabilidade prática do paradigma para usuários sem formação em engenharia de *software*.

2.2 Modelos de Linguagem de Grande Escala (LLMs)

Os Modelos de Linguagem de Grande Escala (*Large Language Models* - LLMs) constituem a infraestrutura tecnológica que torna possível o *vibe coding*. Trata-se de sistemas de aprendizado de máquina treinados em vastos corpora textuais que desenvolvem capacidade de inferência probabilística sobre sequências linguísticas, incluindo código-fonte. A integração de LLMs nos fluxos de desenvolvimento de *software* ocorreu de forma progressiva: iniciou-se com modelos estatísticos de completção de código nos anos 2010 e culminou, na década de 2020, com ferramentas como GitHub Copilot, Amazon CodeWhisperer e Tabnine, capazes de propor blocos funcionais completos de código a partir de comentários ou assinaturas de função.

A principal distinção operacional dos LLMs em relação a abordagens predecessoras reside no mecanismo de inferência. Modelos baseados em redes neurais de grande escala aprenderam a identificar regularidades estatísticas no código de forma análoga à linguagem natural, o que lhes permite não apenas completar sintaxe, mas também inferir semântica e propor implementações contextualmente coerentes. Meske *et al.* (2025) descrevem esse processo como mediação probabilística-interpretativa: o modelo interpreta a especificação em linguagem natural do desenvolvedor e produz saídas de código executável, enquanto o processamento final permanece determinístico, seja via compilação ou interpretação. Essa distinção é analiticamente relevante porque preserva a previsibilidade computacional do código gerado, independentemente da imprecisão da instrução que o originou.

A adoção de LLMs em ambientes de desenvolvimento industrial apresenta dados que indicam transformação estrutural nos processos de produção de *software*. Análises de bases de código corporativas mostram aumento de 70% nas *pull requests* desde o final de 2022, sem crescimento equivalente no número de desenvolvedores, o que sugere que os modelos funcionam como multiplicadores de capacidade produtiva. Ao mesmo tempo, avaliações de segurança em larga escala de mais de cem modelos de linguagem revelaram que apenas 55% do código gerado por IA é seguro, com desempenho de segurança estagnado ao longo do tempo, mesmo à medida que os modelos melhoraram na geração de código sintaticamente correto. Esse dado indica que correção funcional e qualidade de segurança são dimensões independentes no código produzido por LLMs.

Mitchell e Shaaban (2025) identificam limitações estruturais dos LLMs que se tornam particularmente evidentes no contexto do *vibe coding*: os modelos otimizam para completude de padrões locais, não para planejamento combinatorial necessário à consistência global do código. Isso produz quatro classes recorrentes de falhas: inconsistência de restrições, propagação parcial de alterações, divergência de máquinas de estado e acumulação de código duplicado. Cada uma dessas falhas decorre da incapacidade dos LLMs de manter rastreamento de dependências entre módulos ao longo de sessões de desenvolvimento extensas, problema agravado pela tendência dos modelos de priorizar comandos imediatos do usuário sobre a consistência da base de código. Gao *et al.* (2025), referenciados por Meske *et al.* (2025), identificam desafios adicionais associados à adoção de LLMs em engenharia de *software*, incluindo geração de código inseguro, alucinações, herança de vulnerabilidades e ausência de protocolos padronizados de validação.

2.3 Letramento em Inteligência Artificial

O letramento em Inteligência Artificial (*AI literacy*) designa o conjunto de competências que permitem a indivíduos compreender, utilizar, avaliar e interagir com sistemas de IA de forma reflexiva e eticamente orientada. Ng *et al.* (2021) identificaram, a partir de revisão sistemática de 18 artigos, quatro dimensões que compõem esse constructo: conhecer e compreender a IA, aplicar a IA, avaliar e criar com IA, e considerar questões éticas. Essa estrutura quadridimensional opera em progressão: o domínio das camadas anteriores condiciona o desenvolvimento das subsequentes, de modo que a

aplicação responsável pressupõe compreensão dos mecanismos subjacentes, e a avaliação crítica pressupõe experiência de uso.

A demarcação do conceito em relação às literacias digitais predecessoras é feita por Kandlhofer *et al.* (2016) a partir de uma analogia com o letramento clássico em leitura e escrita. Assim como a literacia clássica não se limita à capacidade de decodificar texto, mas abrange a compreensão de estruturas linguísticas e seu uso produtivo, o letramento em IA não se reduz ao uso de aplicações baseadas em IA, mas requer entendimento das técnicas e conceitos que as sustentam. Os autores definem AI literacy como a capacidade de compreender as técnicas e os conceitos subjacentes a produtos e serviços baseados em IA distinguindo essa compreensão do mero aprendizado sobre como utilizar tecnologias ou aplicações específicas. Nessa perspectiva, o letramento em IA constitui uma forma de meta-conhecimento que habilita o indivíduo a compreender sistemas IA novos, em vez de apenas memorizar procedimentos de uso de sistemas já conhecidos.

A formação em letramento em IA apresenta gradações que acompanham o desenvolvimento cognitivo e os níveis de escolarização. Kandlhofer e Steinbauer (2016) propõem um modelo de desenvolvimento em quatro estágios, em analogia com os estágios de aquisição do letramento clássico: construção de consciência e exploração lúdica (educação infantil e anos iniciais), experimentação e familiarização com teoria (ensino fundamental II), aprofundamento nos tópicos centrais e introdução a temas avançados (ensino médio) e fluência em IA com aplicação em nível de abstração elevado (universidade). A correspondência entre esses estágios e os da literacia tradicional não é meramente metafórica: ela estrutura a sequência curricular e determina os objetivos de aprendizagem em cada nível.

Long e Magerko (2020), referenciados por Ng *et al.* (2021), ampliam o conceito ao incluir competências de colaboração e avaliação crítica com sistemas de IA argumentando que o letramento em IA vai além do conhecimento declarativo sobre o funcionamento dos modelos para abarcar a capacidade de comunicar e colaborar efetivamente com eles. Essa extensão do conceito é particularmente pertinente no contexto de ferramentas generativas, onde a interação com o sistema deixa de ser instrumental e passa a ser colaborativa. A dimensão ética, presente em metade dos estudos revisados por Ng *et al.* (2021), aponta para a necessidade de que o letramento em IA incorpore princípios de equidade, responsabilização, transparência e consideração das implicações sociais das tecnologias.

2.4 Programação Assistida por Inteligência Artificial

A programação assistida por IA designa práticas de desenvolvimento de *software* nas quais sistemas de aprendizado de máquina, particularmente LLMs, participam ativamente da geração, completção, revisão ou depuração de código. O conceito se distingue do *vibe coding* por pressupor que o desenvolvedor mantém responsabilidade direta sobre a base de código e integra as contribuições do modelo de forma seletiva, em vez de delegar a implementação ao sistema de IA. Meske *et al.* (2025) localizam essa prática em um estágio intermediário na trajetória histórica da mediação de intenção em desenvolvimento de *software*: mais avançado que a completção estática por regras lexicais dos ambientes de desenvolvimento integrado tradicionais, mas distinto do fluxo conversacional holístico que caracteriza o *vibe coding*.

Ferramentas como o GitHub Copilot representam a configuração mais difundida de programação assistida por IA. Ziegler *et al.* (2024), referenciados por Meske *et al.* (2025), mediram o impacto do Copilot sobre a produtividade e verificaram ganhos em velocidade de produção de código, com o sistema estendendo o conceito de autocompletar de palavras individuais para blocos funcionais com múltiplas linhas. Sergeyuk *et al.* (2025), também referenciados por Meske *et al.* (2025), documentaram o estado de adoção de assistentes de codificação baseados em IA na prática profissional, identificando que desenvolvedores os utilizam principalmente para geração de código repetitivo, exploração de APIs desconhecidas e aceleração de prototipação. Em ambos os estudos, a interação permanece ancorada na produção de trechos de código sintaticamente delimitados, exigindo que o desenvolvedor atue como curador das sugestões geradas.

A redistribuição do trabalho cognitivo entre humano e máquina na programação assistida por IA altera as demandas de expertise, mas não as elimina. O foco cognitivo do desenvolvedor se desloca da autoria minuciosa da lógica de cada componente para a articulação precisa de requisitos, a supervisão especializada das saídas do modelo e a integração coerente do código gerado ao sistema maior. Esse deslocamento pressupõe capacidade de avaliar se o código produzido pelo modelo é correto, seguro e eficiente, o que exige compreensão dos princípios subjacentes à implementação, ainda que o desenvolvedor não a realize manualmente. Pesquisas apontam que usuários que utilizam assistentes de IA frequentemente atribuem um nível elevado de confiança à segurança do código produzido, o que pode resultar em um risco de competência superficial: a

percepção de funcionalidade pode ocultar vulnerabilidades que somente uma análise técnica especializada é capaz de detectar.

Mitchell e Shaaban (2025) argumentam que os sistemas de verificação formal constituem um complemento necessário à programação assistida por IA, especialmente no contexto de bases de código que evoluem iterativamente por meio de prompts. Os autores propõem uma arquitetura do tipo *sidecar* que autoforma especificações de verificação, valida o código gerado contra essas especificações e entrega retroalimentação acionável ao LLM, mantendo o desenvolvedor no controle sem que a verificação formal se torne sobrecarga. Essa arquitetura é classificada como sistema Tipo III, diferenciada de abordagens que ou utilizam métodos formais para filtrar saídas do modelo (Tipo I) ou pós-processam as saídas para satisfazer especificações por construção (Tipo II), por preservar a colaboração humana e a adaptabilidade do fluxo de desenvolvimento. A proposta aponta para uma direção de pesquisa na qual a confiabilidade da programação assistida por IA depende não apenas da qualidade dos modelos, mas da integração de mecanismos de verificação que acompanhem o ritmo de desenvolvimento conversacional.

3 MATERIAL E MÉTODOS

Este estudo adota a abordagem de revisão de escopo proposta por Mak e Thomas (2022), orientada pelas seis etapas: formulação da questão de pesquisa, identificação dos estudos, seleção dos trabalhos, extração dos dados, síntese dos resultados e consulta a partes interessadas. A questão norteadora investigou como o *vibe coding* tem sido incorporado em contextos educacionais.

A busca bibliográfica foi conduzida em março de 2026 nos seguintes repositórios e bases de dados: Google Acadêmico, ACM Digital Library, IEEE Xplore e Arxiv. Os descritores foram combinados por operadores booleanos AND e OR, seguindo a estratégia principal: ("*vibe coding*" OR "*vibe-coding*") AND ("*education*" OR "*teaching*" OR "*learning*" OR "*higher education*" OR "*programming education*" OR "*AI-assisted learning*" OR "*ensino*" OR "*educação*" OR "*aprendizagem*"). A busca abrangeu publicações a partir de 2025, considerando o surgimento do termo *vibe coding* em fevereiro daquele ano.

Foram identificados inicialmente 14 estudos para leitura completa, com base na análise de títulos e resumos. Após a triagem, 3 trabalhos foram excluídos por se tratar de

preprints sem registro de submissão a processo formal de revisão por pares, resultando em uma amostra final de 11 publicações. Os critérios de inclusão adotados foram: (a) estudos que abordassem o *vibe coding* em contextos de ensino, aprendizagem ou formação docente; (b) publicações nos idiomas inglês e português; (c) textos submetidos a avaliação por pares aceitos/publicados em periódicos ou anais de eventos científicos ou realizados em centros de pesquisa relevantes. O critério de exclusão consistiu na disponibilização do trabalho exclusivamente em repositórios de *preprints*, sem evidência de revisão editorial.

4 RESULTADOS E DISCUSSÃO

A seguir, apresentam-se os principais achados obtidos a partir da revisão de escopo, os quais foram organizados na tabela abaixo para facilitar a visualização comparativa entre autores, metodologias, objetivos e resultados. Em sequência, realiza-se a análise crítica e discussão dos dados, buscando evidenciar tendências, desafios e contribuições relacionadas à incorporação do *vibe coding* em contextos educacionais.

Tabela 1 – Sintetização dos resultados

Autores	Metodologia/Tipo de estudo	Objetivo(s)	Resultado(s)
CHEN <i>et al.</i> (2025) ¹	Estudo longitudinal com 16 participantes (tutorial, tarefa prática e entrevistas após duas semanas)	Investigar colaboração de programadores com leitores de tela com assistentes de código avançados e identificar empoderamento e desafios	Assistentes de código aumentam eficiência e reduzem lacunas de acessibilidade; desafios persistem na comunicação de intenções, interpretação de saídas, gerenciamento de múltiplas visões e equilíbrio entre automação e controle
ZHAO; TU (2026)	Estudo de caso qualitativo com workshops estruturados em Replit, avaliações cruzadas e reflexões em disciplina de Análise e Design de Sistemas de Informação	Examinar como técnicas de <i>vibe coding</i> (estilos ágeis e waterfall) melhoram prototipagem e vinculam análise de requisitos à construção de protótipos	Reduz barreiras técnicas; direciona atenção para necessidades de negócio e experiência do usuário; melhora comunicação com clientes; surgem desafios de dependência de código gerado por IA e garantia de qualidade

¹ Pesquisa conduzida no Microsoft Research Lab – Asia. O estudo encontra-se disponível em: <https://www.microsoft.com/en-us/research/publication/screen-reader-users-in-the-vibe-coding-era-adaptation-empowerment-and-new-accessibility-landscape/>. Acesso em: 14 abr. 2026.

GENG <i>et al.</i> (2026)	Estudo observacional com think-aloud; gravações de tela e análise qualitativa de prompts, alterações de código e logs de erro em Replit	Comparar interações de estudantes de programação introdutória e avançada com plataforma de <i>vibe coding</i> durante construção de aplicação web	Interações majoritárias concentram-se em testes de protótipos e depuração via prompts; edição manual de código é rara; estudantes avançados incluem mais contexto de funcionalidades e codebase nos prompts
NG; CHOW (2025)	Monografia com guias práticos, dicas passo a passo e análise de armadilhas em plataformas como Claude e Replit	Oferecer estratégias para educadores de profissões da saúde desenvolverem ferramentas educacionais via <i>vibe coding</i>	Viabiliza criação de aplicações interativas sem formação técnica; exige alinhamento explícito de cada funcionalidade a objetivos de aprendizagem; destaca riscos de complexidade desnecessária e necessidade de supervisão clínica
PRESTES (2025)	Artigo teórico-propositivo ancorado em construcionismo e design centrado no humano	Propor <i>vibe coding</i> como abordagem pedagógica inclusiva e humanizada para formação em engenharia	Posiciona IA generativa como mediadora dialógica que favorece criatividade, autoria e inclusão de estudantes com deficiências, neurodivergentes ou altas habilidades; alinha-se às Diretrizes Curriculares Nacionais da Engenharia
ROYAL (2026)	Estudo de caso com feedback de estudantes em disciplina de desenvolvimento de aplicações móveis; aplicação do modelo Vibe and Flow	Integrar <i>vibe coding</i> à teoria do <i>flow</i> para propor modelo centrado no estudante em educação de comunicação	Facilita aquisição de competências de programação por meio de amostras de código, explicações e troubleshooting; promove concentração, controle e experiência autotelica
SANTOS <i>et al.</i> (2025)	Estudo de caso em disciplina de Metodologia da Pesquisa de mestrado profissional em Vigilância em Saúde	Avaliar convergência entre metodologias ágeis e <i>vibe coding</i> na prototipagem de soluções digitais	Permite construção de cinco MVPs funcionais em uma única aula; supera barreiras técnicas tradicionais; confirma que domínio da lógica sanitária é o principal vetor de inovação
KUSPER; SZABÓ (2025)	Análise preliminar de dados de estudo anterior com questionário específico para <i>vibe coding</i> (versão futura incluirá experimento dedicado)	Testar hipóteses de que <i>vibe coding</i> aumenta confiança nas soluções e produtividade	Confiança melhora de forma fraca, mas associa-se fortemente a fruição e experiências de <i>flow</i> ; estudantes percebem e demonstram aumento de produtividade com suporte de IA
PIMENOVA <i>et al.</i> (2025)	Estudo qualitativo com análise de 190 mil palavras de entrevistas, posts no Reddit e LinkedIn	Caracterizar percepções e práticas de <i>vibe coding</i> , co-criação, <i>flow</i> e confiança	Define <i>vibe coding</i> como paradigma de co-criação regulado pela confiança na IA que sustenta <i>flow</i> ; identifica dores em especificação, confiabilidade,

			depuração e colaboração; propõe práticas mitigadoras
ELIZONDO-GARCÍA (2026)	Revisão de escopo (PRISMA-ScR) de quatro publicações de 2025; análise temática dedutiva segundo modelo TPACK ²	Mapear reconfiguração das competências docentes no uso de <i>vibe coding</i> em educação médica	Reconfigura Conhecimento Tecnológico para prompts e diálogo iterativo; surge papel de clínico-desenvolvedor que utiliza Conhecimento de Conteúdo para auditoria clínica; Conhecimento Tecnológico-Pedagógico permite decomposição de lógicas complexas
SOUZA (2025)	Trabalho de conclusão de curso com desenvolvimento de protótipo de aplicativo	Aplicar Kanban, design centrado no usuário e psicologia do trabalho na prototipagem de ferramenta de gestão de vigilância sanitária	Demonstra viabilidade de protótipo que integra princípios de <i>vibe coding</i> para apoiar coordenação de processos sanitários em portos, aeroportos e fronteiras

Fonte: Elaborado pelo autor, 2026

4.1 Discussão dos resultados

Os estudos reunidos na Tabela 1 permitem identificar um padrão convergente: o *vibe coding* reconfigura, no plano educacional, tanto a natureza da interação entre estudante e ferramenta quanto o perfil de competências necessárias para o desenvolvimento de *software*. Meske *et al.* (2025) definem esse paradigma como aquele em que humanos e IA generativa co-criam artefatos por meio de diálogo em linguagem natural, deslocando a mediação da intenção do desenvolvedor da instrução determinística para a inferência probabilística. Esse deslocamento redistribui o trabalho epistêmico: a expertise migra da maestria sintática para a decomposição de problemas, a articulação de objetivos e a avaliação crítica das saídas geradas pela IA. Interpretar os resultados da tabela à luz dessa proposição teórica revela tanto as condições em que o *vibe coding* funciona como via de acesso ao letramento em programação quanto os limites que esse uso impõe à formação do estudante.

² Consiste em uma estrutura teórica que articula, de forma integrada, os conhecimentos tecnológico, pedagógico e de conteúdo, com vistas à qualificação das práticas de ensino. Foi proposto por Punya Mishra e Matthew J. Koehler no trabalho *Introducing Technological Pedagogical Content Knowledge*, apresentado na Annual Meeting of the American Educational Research Association 2008, em New York, 2008.

4.1.1 Reconfiguração cognitiva, letramento em IA e competência de domínio

A comparação entre estudantes iniciantes e avançados realizada por Geng *et al.* (2026) constitui um dos achados mais elucidativos do conjunto analisado. Por meio de protocolo think-aloud com gravação de tela, os autores constataram que as interações predominantes se concentravam em testes de protótipos e depuração via prompts, ao passo que a edição manual de código era pouco frequente. Estudantes avançados incluíam consistentemente mais contexto de funcionalidades e da codebase em seus prompts, enquanto os iniciantes operavam com requisições menos estruturadas. Esse dado empírico evidencia que a qualidade da mediação linguística com a IA não é independente do conhecimento técnico prévio, mas o pressupõe. Ng *et al.* (2021) fornecem o enquadramento conceitual para esse fenômeno ao propor que o letramento em IA se estrutura em quatro dimensões articuladas: conhecer e compreender a IA, aplicá-la, avaliá-la e criar com ela, e considerar suas implicações éticas. Para que um estudante decomponha requisitos com precisão e julgue a adequação do código gerado, é necessário que já possua ao menos as duas primeiras dimensões desse letramento. Nessa perspectiva, o *vibe coding* não elimina a necessidade do letramento em IA, mas exige que ele seja desenvolvido como condição prévia para que o uso das ferramentas generativas seja pedagogicamente produtivo.

Kandlhofer *et al.* (2016) aprofundam essa argumentação ao propor que o letramento em IA deve ser construído progressivamente, em analogia ao letramento em leitura e escrita, desde a exploração lúdica na educação infantil até a fluência em abstração no ensino superior. A ausência dessa progressão ajuda a compreender os desafios documentados por Zhao e Tu (2026), que identificaram, em disciplina de Análise e Design de Sistemas de Informação, problemas relacionados à dependência do código gerado pela IA e à dificuldade de garantia de qualidade por parte dos estudantes. Quando o repertório conceitual do estudante não é suficiente para avaliar as saídas do modelo, a fluência conversacional com a IA não se converte em compreensão técnica, o que fragiliza a formação. Elizondo-García (2026), em revisão de escopo orientada pelo modelo TPACK, amplifica essa questão para o plano docente: o *vibe coding* reconfigura o Conhecimento Tecnológico para prompts e diálogo iterativo, ao mesmo tempo que faz emergir o papel do clínico-desenvolvedor, que mobiliza Conhecimento de Conteúdo para auditar clinicamente as saídas da IA. Tanto o estudante quanto o professor precisam,

portanto, de uma base de letramento em IA que sustente o exercício do julgamento crítico sobre os artefatos co-produzidos.

4.1.2 Democratização, engajamento e os limites epistêmicos do paradigma

A redução das barreiras técnicas de entrada no desenvolvimento de *software* é o resultado mais recorrente nos estudos analisados. Santos *et al.* (2025) demonstraram que cinco produtos mínimos viáveis funcionais foram desenvolvidos em uma única aula de mestrado profissional em Vigilância em Saúde, confirmando que o domínio da lógica sanitária, e não da programação, tornou-se o principal vetor de inovação. Ng e Chow (2025) corroboram esse padrão ao demonstrar que educadores de profissões da saúde sem formação técnica foram capazes de criar aplicações interativas funcionais desde que cada funcionalidade fosse explicitamente alinhada a objetivos de aprendizagem.

Meske *et al.* (2025) enquadram esses achados ao identificar a democratização como uma das oportunidades centrais do *vibe coding*: especialistas de domínio passam a traduzir sua expertise diretamente em *software* funcional, sem dependência de habilidades formais de codificação. Essa perspectiva dialoga com a proposta de Kandlhofer *et al.* (2016) de que o letramento em IA deve alcançar o mesmo grau de universalidade que o letramento em leitura e escrita, constituindo base para a participação cidadã em economias orientadas pela IA.

A dimensão afetiva desse processo aparece como variável mediadora nos estudos de Royal (2026) e Kusper e Szabó (2025): a integração do *vibe coding* à teoria do *flow* resultou em estados de concentração, controle e experiência autotélica, com aumento percebido de produtividade associado não à confiança técnica isolada, mas à fruição do processo de co-criação. Meske *et al.* (2025) sustentam teoricamente esse achado ao descrever os estados de *flow* co-criativo como um dos atributos centrais do paradigma, em que a cadência conversacional substitui os ciclos discretos de escrita, compilação e depuração, reduzindo a fragmentação cognitiva e liberando recursos atencionais para a resolução de problemas de domínio.

Prestes (2025) avança nessa direção ao propor o *vibe coding* como abordagem pedagógica inclusiva para a formação em engenharia, posicionando a IA generativa como mediadora dialógica que favorece criatividade, autoria e inclusão de estudantes com deficiências, neurodivergentes ou com altas habilidades. Essa proposição encontra suporte na análise de Ng *et al.* (2021), que destacam a necessidade de ampliar a

participação na IA para grupos sub-representados, assegurando que o design e o uso das tecnologias sejam inclusivos e não reproduzam assimetrias sociais preexistentes. No entanto, os resultados da tabela também sinalizam que a democratização do acesso ao desenvolvimento de *software* não equivale, automaticamente, à formação de competências técnicas consolidadas.

Chen *et al.* (2025), em estudo longitudinal com programadores usuários de leitores de tela, constataram que, mesmo quando os assistentes de código aumentam a eficiência e reduzem lacunas de acessibilidade, persistem desafios na comunicação de intenções, na interpretação de saídas e no equilíbrio entre automação e controle. Pimenova *et al.* (2025), por sua vez, identificaram como principais dificuldades dos praticantes a especificação imprecisa, a confiabilidade variável e a depuração de comportamentos inesperados. Meske *et al.* (2025) sistematizam esses riscos ao apontar para a erosão da expertise de programação, as lacunas de responsabilidade epistêmica e a possibilidade de que estudantes desenvolvam uma percepção equivocada de competência ao gerar código funcional sem compreender sua lógica interna.

Ng *et al.* (2021) reforçam essa preocupação no plano da ética: estudantes tendem a dedicar pouca atenção a questões como viés algorítmico, responsabilidade legal e propriedade intelectual, o que exige que os educadores não se restrinjam ao desenvolvimento de habilidades operacionais, mas orientem os estudantes na compreensão dos impactos sociais das tecnologias de IA.

O percurso de Souza (2025), que integrou princípios de *vibe coding* ao desenvolvimento de um protótipo de gestão sanitária com metodologia centrada no usuário e fundamentos da psicologia do trabalho, indica que esse equilíbrio é viável: o uso de ferramentas generativas orientado pelo domínio técnico do problema e pela responsabilidade epistêmica do desenvolvedor produz resultados mais consistentes do que a adoção irrestrita do paradigma. Esse modelo confirma que o ensino de programação mediado por IA alcança maior profundidade formativa quando ancorado em um letramento em IA que integre, de forma articulada, as quatro dimensões propostas por Ng *et al.* (2021): conhecimento, aplicação, avaliação crítica e consciência ética.

5 CONSIDERAÇÕES FINAIS

A Inteligência Artificial Generativa tem aberto caminhos formativos que, até recentemente, permaneciam restritos a quem dominava as linguagens formais de

programação. O *vibe coding* se insere nesse movimento ao permitir que estudantes e educadores sem trajetória técnica em engenharia de *software* produzam artefatos digitais funcionais por meio de diálogo em linguagem natural com modelos de linguagem. A revisão de escopo realizada neste estudo evidenciou que, em contextos tão distintos quanto o ensino de engenharia, a formação em vigilância sanitária, a educação médica e o ensino de comunicação, a prática tem sido incorporada como via de acesso à produção computacional, com redução das barreiras sintáticas e com deslocamento do foco pedagógico da escrita de código para a articulação de problemas do domínio. Esse deslocamento, quando conduzido com intencionalidade formativa, amplia o escopo de quem pode participar ativamente da construção de soluções digitais e abre possibilidades de inclusão de estudantes historicamente afastados do desenvolvimento de *software*.

A análise também indicou que o potencial formativo do *vibe coding* não se realiza de forma automática pela adoção das ferramentas. Os estudos examinados apontam que a qualidade da mediação conversacional com a IA depende do repertório conceitual prévio do estudante, que a fluência na produção de prompts não equivale à compreensão técnica do código gerado e que a confiança excessiva nas saídas dos modelos pode mascarar falhas de segurança, inconsistências lógicas e vulnerabilidades estruturais. Por essa razão, a incorporação pedagógica do paradigma exige atenção às observações críticas formuladas pelos autores examinados: a necessidade de desenvolvimento do letramento em IA como condição prévia ao uso formativo das ferramentas; a importância da revisão humana qualificada como contraponto à inferência probabilística dos modelos; a exigência de protocolos que integrem considerações éticas desde a concepção das atividades, incluindo viés algorítmico, responsabilização e propriedade intelectual. O uso ético e a capacitação docente e discente emergem, portanto, como condições constitutivas da prática, e não como requisitos acessórios.

O estudo apresenta limitações que devem ser consideradas na interpretação de seus achados. A revisão de escopo foi conduzida a partir de um corpus de publicações recentes, em razão do surgimento do termo *vibe coding* em fevereiro de 2025, o que restringe a amostra a uma janela temporal curta e compromete a identificação de padrões longitudinais. A predominância de estudos de caso, relatos de experiência e análises qualitativas com amostras reduzidas limita a generalização dos resultados. A exclusão de *preprints* sem registro de submissão a revisão por pares, ainda que justificada pelo critério de qualidade editorial, pode ter deixado de incorporar contribuições pertinentes em um campo cuja produção científica se encontra em acelerado processo de consolidação. A

diversidade metodológica e contextual dos trabalhos analisados, embora tenha favorecido a identificação de convergências, dificultou a comparação sistemática entre intervenções pedagógicas e a mensuração de seus efeitos sobre a aprendizagem.

Para pesquisas futuras, recomenda-se o desenvolvimento de estudos longitudinais que acompanhem trajetórias de estudantes expostos a currículos estruturados em torno do *vibe coding*, com avaliação do impacto da prática sobre o letramento em IA, sobre a autonomia técnica e sobre a capacidade de avaliação crítica do código gerado. Investigações comparativas entre diferentes níveis de escolaridade e áreas disciplinares podem esclarecer em que condições a prática produz ganhos formativos mais consistentes e em que contextos compromete a construção de competências técnicas.

Estudos experimentais que contrastem grupos com e sem formação prévia em programação contribuiriam para examinar a hipótese de que o letramento em IA constitui requisito, e não consequência, do uso pedagógico do paradigma. A elaboração de instrumentos de avaliação específicos, capazes de capturar dimensões como a qualidade da mediação conversacional, a profundidade da compreensão do código gerado e a consciência ética sobre o uso de sistemas de IA, representa agenda necessária para a consolidação do campo. Investigações que articulem o *vibe coding* a estruturas de verificação formal e a práticas de auditoria pedagógica do código produzido podem oferecer respostas aos desafios de confiabilidade apontados pela literatura. A ampliação das pesquisas para contextos culturais e linguísticos diversos, em especial aqueles nos quais as ferramentas de IA apresentam desempenho desigual, permitirá examinar as condições de equidade na apropriação do paradigma.

O *vibe coding* se apresenta como direção promissora para o ensino mediado por IA, desde que compreendido como prática pedagógica que demanda fundamentação teórica, orientação ética e capacitação continuada de docentes e estudantes. A contribuição deste estudo reside em organizar, a partir da literatura disponível, um mapa inicial das possibilidades e dos limites dessa incorporação, oferecendo subsídios para decisões institucionais, para o desenho curricular e para a formulação de novas questões de pesquisa em um campo cuja maturação acadêmica está apenas em seu início.

ABSTRACT

The formalization of the term vibe coding by Andrej Karpathy in 2025 brought public visibility to a practice of producing digital artifacts through natural language dialogue with Generative Artificial Intelligence systems, without direct code writing. The rapid dissemination of this practice in educational contexts established a tension between its potential for democratizing software development and the epistemic risks of its adoption without prior technical training. This study maps how recent scientific literature characterizes the incorporation of vibe coding into teaching and learning practices, identifying convergences, tensions, and gaps in the debate. A scoping review methodology was adopted following Mak and Thomas (2022). Fourteen studies were identified, of which 11 composed the final sample after the exclusion of preprints without peer review. Results indicate that vibe coding reduces technical barriers to software development and enables the production of functional applications by students and educators without engineering backgrounds, with reports of engagement states associated with flow theory. At the same time, the studies show that conversational fluency with AI does not equate to technical understanding of the generated code, and that excessive trust in model outputs may conceal security flaws and logical inconsistencies. It is concluded that the formative potential of the paradigm depends on prior Artificial Intelligence literacy, qualified human review, and the integration of ethical dimensions into the design of pedagogical activities.

Keywords: vibe coding, scoping review, artificial intelligence literacy, programming education, generative artificial intelligence.

Agradecimentos

Agradeço aos professores Cindy Royal, Ph.D., da School of Journalism and Mass Communication da Texas State University; Gábor Kusper, Ph.D., do Departamento de Ciência da Computação da Eszterházy Károly Catholic University; e Martin Kandlhofer, Ph.D., pesquisador da Austrian Computer Society, por disponibilizarem acesso às suas pesquisas.

REFERÊNCIAS

CHEN, Nan *et al.* **Screen Reader Programmers in the Vibe Coding Era: Adaptation, Empowerment, and New Accessibility Landscape.** [S. l.], 2025. Disponível em: <https://arxiv.org/abs/2506.13270>. Acesso em: 14 abr. 2026.

ELIZONDO-GARCIA, Josemaria. Vibe-coding as a Teaching Competency in the Use of Artificial Intelligence for Medical Education: A Scoping Review. **Revista Española de Educación Médica**, Murcia, v. 7, n. 2, 705121, 2026. DOI: 10.6018/edumed.705121. Acesso em: 14 abr. 2026.

GENG, Francis *et al.* Exploring Student-AI Interactions in Vibe Coding. In: AUSTRALASIAN COMPUTING EDUCATION CONFERENCE (ACE), 28., 2026, Melbourne. **Proceedings** [...]. New York: ACM, 2026. DOI: 10.1145/3786228.3786236. Acesso em: 14 abr. 2026.

KANDLHOFER, Martin *et al.* Artificial Intelligence and Computer Science in Education: From Kindergarten to University. In: FRONTIERS IN EDUCATION CONFERENCE (FIE), 2016, Erie. **Proceedings** [...]. [S. l.]: IEEE, 2016. DOI: 10.1109/FIE.2016.7757570. Acesso em: 14 abr. 2026.

KUSPER, Gábor; SZABÓ, Csaba. Vibe Coding in Education. In: INTERNATIONAL CONFERENCE ON EMERGING ELEARNING TECHNOLOGIES AND APPLICATIONS (ICETA), 23., 2025, Starý Smokovec. **Proceedings** [...]. [S. l.]: IEEE, 2025. DOI: 10.1109/ICETA67772.2025.11280285. Acesso em: 14 abr. 2026.

LONG, D.; MAGERKO, B. What is AI literacy? Competencies and design considerations. In: Proceedings of the 2020 CHI **Conference on Human Factors in Computing Systems**. New York: Association for Computing Machinery, 2020. p. 1–16. DOI: <https://doi.org/10.1145/3313831.3376727>

MAK, Susanne; THOMAS, Alik. Steps for Conducting a Scoping Review. **Journal of Graduate Medical Education**, [S. l.], v. 14, n. 5, p. 565-567, out. 2022. DOI: 10.4300/JGME-D-22-00621.1. Acesso em: 14 abr. 2026.

MESKE, Christian *et al.* Vibe Coding as a Reconfiguration of Intent Mediation in Software Development: Definition, Implications, and Research Agenda. **IEEE Access**, [S. l.], v. 13, p. 213242-213259, 2025. DOI: 10.1109/ACCESS.2025.3645466. Acesso em: 14 abr. 2026.

MEYER, Jesse G. Vibe Coding Omics Data Analysis Applications. **Journal of Proteome Research**, [S. l.], v. 25, p. 1191-1197, 2026. DOI: 10.1021/acs.jproteome.5c00984. Acesso em: 14 abr. 2026.

MITCHELL, Jacqueline; SHAABAN, Yasser. Position: Vibe Coding Needs Vibe Reasoning: Improving Vibe Coding with Formal Verification. In: ACM SIGPLAN INTERNATIONAL WORKSHOP ON LANGUAGE MODELS AND PROGRAMMING LANGUAGES (LMPL), 1., 2025, Singapore. **Proceedings** [...]. New York: ACM, 2025. p. 84-90. DOI: 10.1145/3759425.3763390. Acesso em: 14 abr. 2026.

MISHRA, Punya; KOEHLER, Matthew J. **Introducing Technological Pedagogical Content Knowledge**. In: ANNUAL MEETING OF THE American Educational Research Association, 2008, New York. Anais [...]. New York: AERA, 2008. Disponível em: https://www.researchgate.net/publication/242385653_Introducing_Technological_Pedagogical_Content_Knowledge. Acesso em: 14 abr. 2026.

NG, Davy Tsz Kit *et al.* Conceptualizing AI literacy: An exploratory review. **Computers and Education: Artificial Intelligence**, [S. l.], v. 2, 100041, 2021. DOI: 10.1016/j.caeai.2021.100041. Acesso em: 14 abr. 2026.

NG, Olivia; CHOW, Minyang. Empowering Health Professions Educators: Developing Educational Tools with AI-Assisted Vibe Coding. **Medical Science Educator**, [S. l.], 2025. DOI: 10.1007/s40670-025-02596-1. Acesso em: 14 abr. 2026.

PIMENOVA, Veronica *et al.* **Good Vibrations? A Qualitative Study of Co-Creation, Communication, Flow, and Trust in Vibe Coding**. [S. l.], 2025. Disponível em: <https://arxiv.org/abs/2509.12491>. Acesso em: 14 abr. 2026.

PRESTES, José Augusto de Lima. Vibe coding e ensino humanizado com IA: uma proposta inclusiva para a formação em engenharia. In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA (COBENGE), 2025, Campinas. **Anais [...]**. Campinas: Associação Brasileira de Educação em Engenharia (ABENGE), 2025. DOI: 10.37702/2175-957X.COBENGE.2025.6030. Acesso em: 14 abr. 2026.

ROYAL, Cindy. Integrating Vibe Coding and *Flow* Theory: A Student-Centered Model for AI-Augmented Coding Education. **Journalism & Mass Communication Educator**, [S. l.], p. 1-22, 2026. DOI: 10.1177/10776958251407389. Acesso em: 14 abr. 2026.

SANTOS, Ricardo Marciano dos *et al.* Metodologias ágeis e vibe coding na Universidade Iguazu (UNIG): aceleração da inovação tecnológica no mestrado profissional em vigilância em saúde: um estudo de caso na disciplina metodologia da pesquisa sobre a prototipagem de soluções digitais mediada por inteligência artificial. In: **Avanços na saúde: pesquisas e práticas transformadoras**. [S. l.]: Editora Omnis Scientia, 2025. v. 2, cap. 59, p. 589-601. Disponível em: <https://editora.editoraomnisscientia.com.br/artigoPDF/24202015917.pdf>. Acesso em: 14 abr. 2026.

SOUZA, Francisco Canindé Gerlândio de. **Desenvolvimento de protótipo de aplicativo baseado em kanban, princípio de design centrado no usuário e abordagens da psicologia do trabalho para apoiar a gestão da coordenação de vigilância sanitária de portos, aeroportos, fronteiras e recintos alfandegados do Rio Grande do Norte**. 2025. 35 f. Trabalho de Conclusão de Curso (Bacharelado em Administração) – Centro de Ciências Sociais Aplicadas, Universidade Federal do Rio Grande do Norte, Natal, 2025. Disponível em: <https://repositorio.ufrn.br/bitstreams/a0f9ebbd-7671-4491-991f-2355abe582c8/download>. Acesso em: 14 abr. 2026.

ZHAO, Gary Yu; TU, Cindy Zhiling. Utilizing Vibe Coding Techniques to Improve Prototyping in the Information Systems Analysis and Design Class. **The Journal of Computing Sciences in Colleges**, [S. l.], v. 19, 2026.