

Agentes inteligentes no ensino de programação: uma revisão sistemática da literatura

 <https://doi.org/10.47236/2594-7036.2026.v10.2114>

Beatriz Borges de Oliveira¹
Rogério Alves dos Santos Antoniassi²
Alex Fernando de Araujo³

Data de submissão concluída: 31/3/2026. Data de aprovação: 2/6/2026. Data de publicação: 24/8/2026.

Resumo – O ensino de programação, especialmente em disciplinas introdutórias, tem sido historicamente associado a elevadas taxas de evasão e dificuldades de aprendizagem, decorrentes da alta carga cognitiva e da limitação de suporte individualizado. Nesse contexto, o uso de agentes inteligentes tem sido proposto como estratégia para apoiar o processo de aprendizagem. Este estudo tem como objetivo analisar o uso desses agentes no ensino de programação no período de 2020 a 2025. A metodologia consistiu em uma revisão sistemática da literatura, conduzida nas bases ACM Digital Library, IEEE Xplore, Scopus e Web of Science, resultando em 573 registros iniciais. Após a remoção de duplicatas e aplicação dos critérios de elegibilidade, 81 estudos compuseram o *corpus* final. Os resultados evidenciam uma transição de sistemas tutores baseados em regras para agentes fundamentados em modelos de linguagem de grande escala, acompanhada pela ampliação de suas funções pedagógicas, com destaque para a geração de dicas, explicação de erros e suporte adaptativo. Observam-se ganhos em desempenho e engajamento dos estudantes, embora persistam limitações relacionadas à dependência excessiva e à confiabilidade das respostas geradas. Conclui-se que o uso de agentes inteligentes não se restringe a um avanço tecnológico, mas implica uma reconfiguração das práticas pedagógicas, marcada pela tensão entre a assistência automatizada e o desenvolvimento da autonomia discente.

Palavras-chave: Agentes inteligentes. Aprendizagem de programação. Inteligência artificial. Revisão sistemática. Tecnologias educacionais.

Intelligent agents in programming education: a systematic literature review

Abstract – Programming education, particularly in introductory courses, has historically been associated with high dropout rates and learning difficulties, resulting from high cognitive load and limited individualized support. In this context, the use of intelligent agents has been proposed as a strategy to support the learning process. This study aims to analyze the use of such agents in programming education between

¹ Graduanda em Engenharia de Computação pelo *Campus* Três Lagoas do Instituto Federal de Mato Grosso do Sul. Três Lagoas, Mato Grosso do Sul, Brasil.  beatriz.oliveira5@estudante.ifms.edu.br 
 <https://orcid.org/0009-0008-6824-8383>  <http://lattes.cnpq.br/8083451321671636>.

² Mestre em Ciência da Computação pela Universidade Federal de Mato Grosso do Sul. Professor da área de Ciência da Computação do Ensino Básico, Técnico e Tecnológico do *Campus* Três Lagoas do Instituto Federal de Mato Grosso do Sul. Três Lagoas, Mato Grosso do Sul, Brasil. 
 <https://orcid.org/0009-0004-5242-3678> 
 <http://lattes.cnpq.br/9378799635475932>.

³ Mestre em Ciência da Computação pela Universidade Estadual Paulista “Júlio de Mesquita Filho”. Professor da área de Ciência da Computação do Ensino Básico, Técnico e Tecnológico do *Campus* Três Lagoas do Instituto Federal de Mato Grosso do Sul. Três Lagoas, Mato Grosso do Sul, Brasil. 
 <https://orcid.org/0000-0003-1464-3229> 
 <http://lattes.cnpq.br/1438908573766889>.

2020 and 2025. The methodology consisted of a systematic literature review conducted in the ACM Digital Library, IEEE Xplore, Scopus, and Web of Science databases, resulting in 573 initial records. After duplicate removal and the application of eligibility criteria, 81 studies composed the final corpus. The results reveal a transition from rule-based tutoring systems to agents based on large language models, accompanied by an expansion of their pedagogical functions, particularly in hint generation, error explanation, and adaptive support. Improvements in student performance and engagement are observed, although limitations related to over-reliance and the reliability of generated responses persist. It is concluded that the use of intelligent agents does not merely represent a technological advancement but entails a reconfiguration of pedagogical practices, characterized by a tension between automated assistance and the development of student autonomy.

Keywords: Artificial intelligence. Educational technologies. Intelligent agents. Programming education. Systematic review.

Agentes inteligentes en la enseñanza de la programación: una revisión sistemática de la literatura

Resumen – La enseñanza de la programación, especialmente en los cursos introductorios, se ha asociado históricamente con altas tasas de deserción y dificultades de aprendizaje, derivadas de la elevada carga cognitiva y de la limitación de soporte individualizado. En este contexto, se ha propuesto el uso de agentes inteligentes como estrategia para apoyar el proceso de aprendizaje. Este estudio tiene como objetivo analizar el uso de estos agentes en la enseñanza de la programación entre 2020 y 2025. La metodología consistió en una revisión sistemática de la literatura, realizada en las bases de datos ACM Digital Library, IEEE Xplore, Scopus y Web of Science, lo que resultó en 573 registros iniciales. Tras la eliminación de duplicados y a aplicación de los criterios de elegibilidad, 81 estudios compusieron el corpus final. Los resultados evidencian una transición de los sistemas tutores basados en reglas a agentes fundamentados en modelos de lenguaje a gran escala, acompañada por la ampliación de sus funciones pedagógicas, con destaque para la generación de pistas, explicación de errores y soporte adaptativo. Se observan ganancias en desempeño y compromiso de los estudiantes, aunque persisten limitaciones relacionadas con la dependencia excesiva y la confiabilidad de las respuestas generadas. Se concluye que el uso de agentes inteligentes no se limita a un avance tecnológico, sino que implica una reconfiguración de las prácticas pedagógicas, marcada por la tensión entre la asistencia automatizada y el desarrollo de la autonomía del estudiante.

Palabras clave: Agentes inteligentes. Aprendizaje de programación. Inteligencia artificial. Revisión sistemática. Tecnologías educativas.

Introdução

A programação de computadores consolidou-se como uma competência estratégica no mercado de trabalho contemporâneo, sendo fundamental para a inovação tecnológica, a alfabetização digital e a transformação dos processos organizacionais. A expansão da digitalização, intensificada durante a pandemia de COVID-19 (Sousa; Lage, 2024), ampliou essa demanda para além do setor de tecnologia da informação, tornando habilidades computacionais relevantes em áreas como saúde, indústria e finanças. Evidências empíricas indicam que profissionais com

tais competências tendem a ampliar oportunidades de inserção profissional e alcançar salários de 20% a 30% maiores em relação àqueles sem essas habilidades (Katona; Gyonyoru, 2025a).

Apesar dessa relevância crescente, o ensino de programação permanece associado a elevadas taxas de reprovação e evasão, especialmente em disciplinas introdutórias (Alshaikh; Tamang; Rus, 2020; Shum; Rosunally; Munir, 2025). A natureza abstrata da área exige dos estudantes a articulação simultânea de conceitos como algoritmos, estruturas de controle e sintaxe formal na resolução de problemas computacionais (Masetty; Vallamulla, 2025; Zhao *et al.*, 2025). Na perspectiva da psicologia cognitiva, a limitação de suporte individualizado em turmas numerosas pode elevar significativamente a carga cognitiva dos estudantes iniciantes, dificultando a consolidação do raciocínio lógico e favorecendo estratégias ineficazes baseadas em tentativa e erro (Silva; Souza, 2025).

Nesse contexto, a Inteligência Artificial passou a ser incorporada aos ambientes educacionais como mecanismos de suporte adaptativo à aprendizagem. Inicialmente, destacaram-se os Sistemas Tutores Inteligentes, capazes de modelar o estado de conhecimento dos estudantes e fornecer *feedback* personalizado (Chrysafiadi; Virvou, 2024; Eilermann *et al.*, 2023). Mais recentemente, a ascensão dos Grandes Modelos de Linguagem ampliou as possibilidades de personalização, geração de código e interação em linguagem natural no ensino de programação (Frankford *et al.*, 2024; Kazemitabaar *et al.*, 2024; Vemula, 2024).

Embora os agentes inteligentes tenham ampliado as possibilidades de suporte à aprendizagem, a literatura ainda apresenta resultados divergentes quanto aos efeitos pedagógicos em longo prazo (Akçapinar; Sidan, 2024; Masetty; Vallamulla, 2025; Ouyang *et al.*, 2024). Estudos recentes alertam que a mediação intensiva por esses sistemas pode induzir à dependência tecnológica, à compreensão superficial dos conceitos e à aceitação acrítica de respostas potencialmente incorretas geradas pelos modelos (Brown *et al.*, 2025; Feng; Liu; Ghosal, 2024; Jiang *et al.*, 2025; Kurniawan *et al.*, 2025). Além disso, emergem preocupações éticas relacionadas à integridade acadêmica e ao uso inadequado de ferramentas automatizadas em contextos avaliativos (Akçapinar; Sidan, 2024).

Diante desse cenário, ratifica-se a necessidade de sistematizar as evidências recentes e compreender de forma integrada os impactos pedagógicos associados ao uso de agentes inteligentes no ensino de programação. A predominância de estudos de curta duração e de avaliações centradas principalmente em aspectos técnicos reforça a necessidade de investigações mais abrangentes, capazes de articular fatores cognitivos, pedagógicos e socioculturais relacionados à adoção dessas tecnologias (Kazemitabaar *et al.*, 2024; Lyu *et al.*, 2024; H. Wang *et al.*, 2025). Nesse sentido, este estudo tem como objetivo consolidar e analisar criticamente as evidências empíricas relacionadas ao uso de agentes inteligentes no ensino de programação entre 2020 e 2025, buscando identificar tendências tecnológicas, funções pedagógicas, impactos educacionais e limitações reportadas na literatura.

Este artigo está estruturado em quatro seções. Após esta introdução, a seção 2 descreve os materiais e métodos adotados. A seção 3 apresenta os resultados e as respectivas discussões críticas. Por fim, a seção 4 reúne as considerações finais do estudo.

2 Materiais e métodos

2.1 Caracterização da revisão

Este estudo caracteriza-se como uma revisão sistemática da literatura (RSL), modalidade de pesquisa que sugere protocolos específicos e busca garantir rigor e reprodutibilidade à investigação científica (Galvão; Ricarte, 2019). Conforme os autores, a RSL diferencia-se de revisões narrativas por explicitar as estratégias de busca, os critérios de seleção dos estudos e os procedimentos empregados na análise do *corpus* documental. Para assegurar transparência nas etapas de identificação, triagem e elegibilidade dos estudos, a pesquisa foi conduzida com base nas diretrizes da declaração PRISMA 2020 (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*) (Page *et al.*, 2021).

O recorte temporal adotado (2020-2025) abrange o contexto da pandemia de COVID-19 e a intensificação do uso de tecnologias digitais em ambientes educacionais (Sousa; Lage, 2024), cenário que impulsionou o desenvolvimento de soluções baseadas em Inteligência Artificial aplicadas ao ensino de programação. Dessa forma, o período selecionado mostra-se relevante para a análise da evolução recente dos agentes inteligentes na educação.

A pesquisa foi norteadada pela seguinte questão central: “Como os agentes inteligentes têm sido utilizados para apoiar o processo de aprendizagem de programação?”. Para operacionalizar a análise, a questão foi desdobrada em cinco questões de pesquisa (QPs): (QP1) arquiteturas e abordagens tecnológicas; (QP2) funções pedagógicas e estratégias de mediação; (QP3) benefícios, desafios e limitações; (QP4) níveis de ensino, linguagens de programação e contextos socioculturais; e (QP5) metodologias e métricas utilizadas para avaliar a efetividade dessas soluções.

2.2 Estratégia de busca e seleção dos estudos

A estratégia de busca foi elaborada usando combinações de termos relacionados a agentes inteligentes e à aprendizagem de programação, com ajustes sintáticos feitos de acordo com as exigências de cada base de dados. Foram selecionadas quatro bases: ACM Digital Library e IEEE Xplore, devido à sua importância técnica na computação; e Scopus e Web of Science, pela ampla cobertura multidisciplinar e indexação de periódicos de alto impacto. As *strings* de busca, os filtros aplicados e a quantidade de estudos recuperados em cada fonte estão sintetizados no Quadro 1.

Quadro 1 – Estratégia de busca e resultados por base de dados

Base de dados	String de busca	Filtros aplicados	Retornos
ACM Digital Library	("Intelligent Agent*" OR "Educational Agent*" OR "Intelligent Tutoring System*" OR "AI Agent*" OR "Personalized Learning" OR "AI in Education" OR "Adaptive Learning") AND ("Programming Education" OR "Coding Education" OR "Programming Learning" OR "Learning to Code" OR "Introductory Programming" OR "Computer Science Education" OR "CS Education")	Conteúdo: Research Article; Publicação: Journals; Período: 2020 – 2025	60
IEEE Xplore	("Intelligent Agent*" OR "Educational Agent*" OR "Intelligent Tutoring System*" OR "AI Agent*" OR "Personalized Learning" OR "AI in Education" OR "Adaptive Learning") AND ("Programming Education" OR "Coding Education" OR "Programming Learning" OR "Learning to Code" OR "Introductory Programming" OR "Computer Science Education" OR "CS Education")	Publicação: Conference e Journals; Período: 2020 – 2025	131

Scopus	(TITLE-ABS-KEY ("Intelligent Agent*" OR "Educational Agent*" OR "Intelligent Tutoring System*" OR "AI Agent*" OR "Personalized Learning" OR "AI in Education" OR "Adaptive Learning")) AND (TITLE-ABS-KEY ("Programming Education" OR "Coding Education" OR "Programming Learning" OR "Learning to Code" OR "Introductory Programming" OR "Computer Science Education" OR "CS Education"))	Tipo: Conference paper e Article; Área: Computer Science e Engineering; Período: 2020 – 2025	245
Web of Science	TS=("Intelligent Agent*" OR "Educational Agent*" OR "Intelligent Tutoring System*" OR "AI Agent*" OR "Personalized Learning" OR "AI in Education" OR "Adaptive Learning") AND TS=("Programming Education" OR "Coding Education" OR "Programming Learning" OR "Learning to Code" OR "Introductory Programming" OR "Computer Science Education" OR "CS Education") PD = 2020-01-01 até 2025-12-31	Tipo: Article e Proceeding Paper; Web of Science Categories: Ciência da Computação, Engenharia e Educação; Período: 2020 – 2025	137

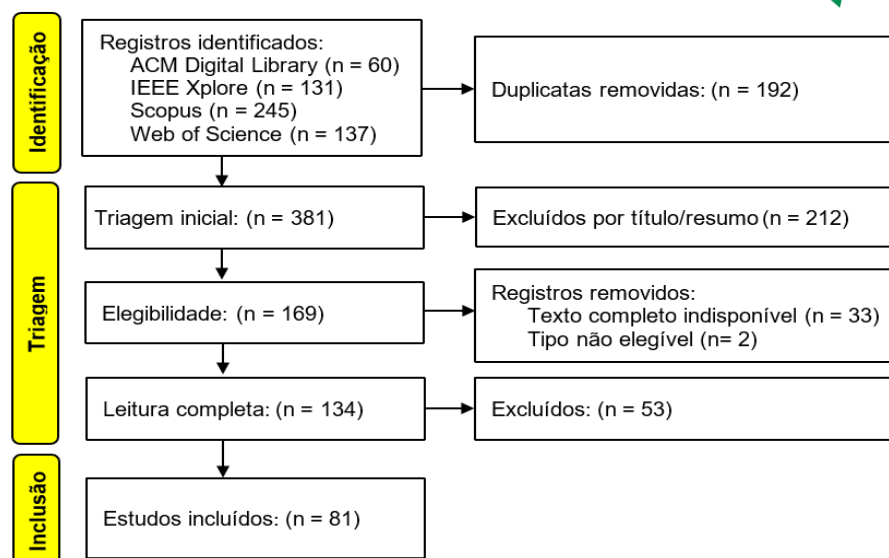
Fonte: Elaborado pelos autores (2026).

Após a implementação das estratégias de busca nas bases de dados, foram encontrados 573 registros. Em seguida, realizou-se a remoção de 192 duplicatas com o auxílio do *software* Zotero (ferramenta de gerenciamento bibliográfico empregada na organização e filtragem dos estudos recuperados), resultando em 381 estudos para a triagem inicial. Essa etapa ocorreu por meio da leitura de títulos e resumos, na qual 212 publicações foram excluídas por se tratar de estudos secundários, não apresentarem contribuição técnica relevante ou não possuírem relação direta com o uso de agentes inteligentes no ensino de programação.

Os 169 trabalhos restantes avançaram para a análise de elegibilidade. Nessa fase, 35 estudos foram removidos, dos quais 33 devido à indisponibilidade do texto completo e dois por se enquadrarem em tipos de publicação não elegíveis. Com isso, restaram 134 artigos para leitura na íntegra. Durante essa avaliação detalhada, 53 foram excluídos por não atenderem integralmente aos critérios estabelecidos. Assim, 81 artigos compuseram a amostra final desta revisão sistemática. Todo esse processo de identificação, triagem, elegibilidade e inclusão está ilustrado na Figura 1.

Figura 1 – Fluxograma do processo de seleção dos estudos

Artigo Científico



Fonte: Elaborado pelos autores (2026).

2.3 Critérios de inclusão e exclusão

Para assegurar a relevância e a consistência do *corpus* analisado, foram considerados os seguintes critérios de inclusão: a) artigos publicados entre 2020 e 2025; b) redigidos em inglês ou português; c) pesquisas primárias; d) foco no ensino de programação; e) uso de agentes inteligentes; e f) suporte direto ao estudante no processo de aprendizagem.

Como critérios de exclusão, foram adotados: a) duplicatas entre as bases; b) estudos secundários (revisões e mapeamentos); c) formatos não definitivos (*short papers*, pôsteres ou trabalhos em andamento); d) trabalhos sem profundidade técnica ou fora do escopo; e e) indisponibilidade do texto completo.

2.4 Extração e análise dos dados

Para garantir a robustez desta revisão, procedeu-se inicialmente à avaliação da qualidade dos estudos selecionados. Esse processo considerou aspectos como a clareza dos objetivos, a adequação metodológica, a consistência dos resultados e o nível de detalhamento técnico das soluções propostas. Adicionalmente, examinaram-se a descrição da arquitetura dos agentes inteligentes, os métodos de avaliação e as evidências empíricas apresentadas. Estudos com informações insuficientes ou baixa consistência metodológica foram excluídos.

A partir desse filtro de qualidade, consolidou-se a amostra final em 81 estudos. Cabe destacar que o *corpus* analisado foi composto exclusivamente por pesquisas primárias. Embora revisões sistemáticas e mapeamentos da literatura tenham sido recuperados nas etapas iniciais da busca, esses trabalhos não integraram a síntese final, de modo a evitar sobreposição de evidências e priorizar análises empíricas diretas sobre o uso de agentes inteligentes no ensino de programação. Dessa amostra, realizou-se a extração de variáveis relacionadas aos aspectos centrais da pesquisa, incluindo o tipo de agente inteligente, sua arquitetura tecnológica predominante, a função pedagógica exercida, o nível de ensino contemplado e a linguagem de programação abordada. Também foram levantadas as metodologias e métricas de avaliação, bem como as vantagens e limitações reportadas nas soluções.

Em alguns estudos, determinadas informações contextuais ou demográficas não eram explicitadas integralmente pelos autores. Nesses casos, algumas classificações analíticas foram inferidas com base em elementos presentes nos próprios trabalhos,

como afiliações institucionais, contexto educacional descrito e objetivos declarados das intervenções. Buscou-se, contudo, preservar a coerência interpretativa e evitar extrapolações não sustentadas pelos textos originais.

Com base nesses dados, os estudos foram organizados de forma iterativa a partir da recorrência de temas e da proximidade entre os elementos identificados no *corpus*. Esse processo possibilitou o agrupamento dos trabalhos segundo abordagens tecnológicas, funções pedagógicas, formas de interação e contextos educacionais, contribuindo para uma análise mais coerente e sistematizada da literatura.

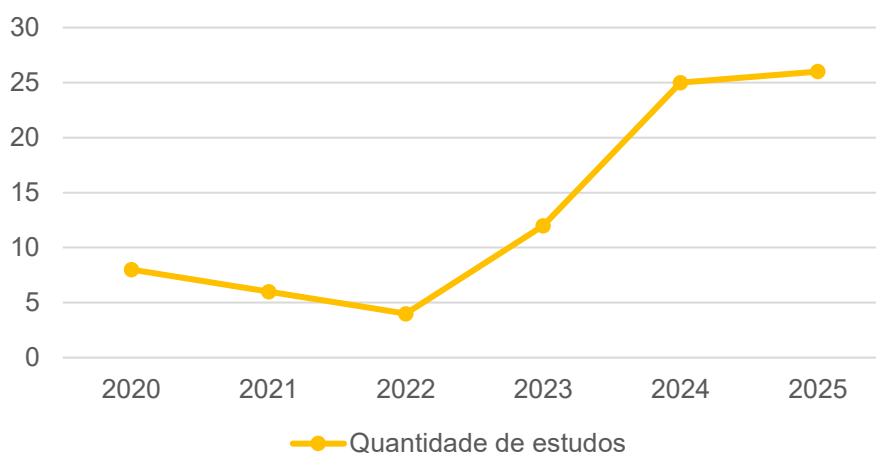
A análise foi conduzida por meio de uma abordagem mista, combinando procedimentos quantitativos e qualitativos. Inicialmente, realizou-se uma caracterização descritiva da amostra. Em seguida, procedeu-se à interpretação dos estudos com base nos agrupamentos definidos, em alinhamento com a pergunta de pesquisa e os objetivos deste trabalho. Essa estratégia metodológica permitiu a identificação de padrões, tendências e lacunas na literatura.

3 Resultados e discussões

3.1 Panorama geral da produção científica

A distribuição temporal dos 81 estudos selecionados evidencia a expansão progressiva das pesquisas sobre agentes inteligentes aplicados ao ensino de programação ao longo do período analisado. O volume de publicações sugere um interesse crescente da comunidade científica pela temática, especialmente nos anos mais recentes. A evolução anual das publicações está apresentada no Gráfico 1.

Gráfico 1 – Evolução anual das publicações (2020 – 2025)



Fonte: Elaborado pelos autores (2026).

A análise cronológica revela um volume de produção mais modesto e relativamente estável entre os anos de 2020 e 2022. Nesse período, o desenvolvimento de agentes inteligentes dependia fortemente de processos extensivos de modelagem de domínio e engenharia de conhecimento, fatores que demandavam elevado esforço técnico para implementação e adaptação em ambientes educacionais reais. Essas limitações contribuíam para restringir a escalabilidade das soluções propostas e, conseqüentemente, o volume de estudos empíricos publicados.

Em contrapartida, o período subsequente apresenta um crescimento expressivo da produção científica. Esse avanço coincide com a popularização dos Grandes Modelos de Linguagem (*LLMs*) e das *APIs* de Inteligência Artificial generativa. Estudos

conduzidos por Frankford *et al.* (2024), Kazemitabaar *et al.* (2024) e Ouyang *et al.* (2024) demonstram que a disponibilidade de ferramentas baseadas em modelos como ChatGPT e OpenAI Codex reduziu significativamente as barreiras técnicas para o desenvolvimento de agentes inteligentes aplicados ao ensino de programação. Como consequência, a integração de sistemas generativos em ambientes reais de aprendizagem tornou-se mais acessível, favorecendo a ampliação das investigações empíricas e diversificando as possibilidades de aplicação pedagógica desses agentes.

Além da evolução temporal da produção científica, buscou-se caracterizar individualmente os estudos selecionados quanto às tecnologias empregadas, aos delineamentos metodológicos e às estratégias de avaliação adotadas. Essa sistematização encontra-se apresentada no Quadro 2.

Quadro 2 – Caracterização dos estudos selecionados

Estudo	Feedback	Tecnologia	Contexto	Método	Estatística	Controle
Ahmed <i>et al.</i> , 2020	Sim	RNN	ES	Est. Campo	Sim	Não
Akçapinar e Sidan, 2024	Sim	GPT / LLM	ES	Quase-Exp.	Sim	Sim
Alzoubi <i>et al.</i> , 2020	Sim	Grafo / ITS	ES	Quase-Exp.	Sim	Não
Alanazi <i>et al.</i> , 2025	Não exp.	Não exp.	ES	Survey	Sim	Não
Alencar <i>et al.</i> , 2025	Sim	NLP / AST	ES	Est. Retros.	Sim	Não
Alshaikh; Tamang; Rus, 2020	Sim	STS	ES	Ensaio Cont.	Sim	Sim
Alshaikh; Tamang; Rus, 2021	Sim	NLP	Não exp.	Ensaio Cont.	Sim	Sim
Anikin, Sychev e Denisov, 2021	Sim	Ontologias	ES	Validação	Não exp.	Não
Araújo <i>et al.</i> , 2020	Sim	App Script	ES	Est. Caso	Sim	Não
Bahrevar e Moshirpour, 2023	Sim	LLM / AST	ES	Validação	Sim	Não
Brown <i>et al.</i> , 2025	Sim	Multi-LLM	ES	Ensaio Cont.	Sim	Sim
Błażejowska <i>et al.</i> , 2023	Sim	Robótica	EB	Est. Caso	Não exp.	Não
Chang e Chen, 2022	Sim	YouTube API	EB	Quase-Exp.	Sim	Sim
Chandrasekara <i>et al.</i> , 2025	Sim	GPT / RAG	Não exp.	Validação	Não exp.	Não
Chapagain e Rus, 2025	Não exp.	BERT / LLM	Não exp.	Validação	Sim	Não
Chiang, Lin e Chen, 2024	Sim	OpenAI API	ES	Validação	Não exp.	Não
Chrysafiadi e Virvou, 2024	Sim	Lógica Fuzzy	ES	Ensaio Cont.	Sim	Sim
Cárdenas-Cobo <i>et al.</i> , 2024	Sim	Scratch	EB	Quase-Exp.	Sim	Não
Dannath, Deriyeva e Paaßen, 2024	Sim	Code Llama	Não exp.	Validação	Não exp.	Não
Efremov, Ghosh e Singla, 2020	Sim	Ap. Reforço	EB	Validação	Sim	Não

Eilermann <i>et al.</i> , 2023	Sim	ChatGPT	EP	Validação	Não exp.	Não
Fan <i>et al.</i> , 2025	Sim	LLaMA	ES	Est. Campo	Sim	Não
Feng, Liu e Ghosal, 2024	Sim	LLM / RAG	ES	Validação	Não exp.	Não
Fischer <i>et al.</i> , 2023	Sim	Banco Erros	ES	Quase-Exp.	Sim	Não
Frankford <i>et al.</i> , 2024	Sim	GPT-3.5 API	ES	Est. Campo	Sim	Não
Gutierrez, Denny e Luxton-Reilly, 2024	Sim	OpenAI API	ES	Est. Campo	Sim	Não
Huang <i>et al.</i> , 2023	Sim	CTAT / ITS	ES	Ensaio Cont.	Sim	Sim
Jebara, Nawahdah e Sawalha, 2025	Não exp.	Não exp.	Não exp.	Validação	Não exp.	Não
Jiang <i>et al.</i> , 2025	Sim	RAG / FAISS	ES	Validação	Não exp.	Não
Katona e Gyonyoru, 2025a	Sim	Algoritmos	ES	Ensaio Cont.	Sim	Sim
Katona e Gyonyoru, 2025b	Sim	Adaptativos	ES	Quase-Exp.	Sim	Sim
Kazemitabaar <i>et al.</i> , 2024	Sim	LLM	ES	Est. Campo	Sim	Não
Koch <i>et al.</i> , 2024	Sim	Trace Tool	ES	Validação	Sim	Não
Koike <i>et al.</i> , 2020	Sim	Ontologias	ES	Validação	Não exp.	Não
Krouska <i>et al.</i> , 2023	Sim	Lógica Fuzzy	Não exp.	Validação	Sim	Não
Kurniawan <i>et al.</i> , 2025	Sim	Copilot/LLM	ES	Est. Piloto	Não exp.	Não
Lai e Lin, 2025	Sim	GPT API	ES	Quase-Exp.	Sim	Não
Lee, 2020	Sim	Alg. Recur.	Não exp.	Est. Observ.	Não exp.	Não
Ying Li <i>et al.</i> , 2023	Sim	GAT / Bi-LSTM	ES	Validação	Sim	Não
Yiwei Li <i>et al.</i> , 2025	Sim	LLM (Agentes)	ES	Validação	Sim	Não
W. Li <i>et al.</i> , 2025	Sim	RAG / LLM	ES	Validação	Sim	Não
C. Li, Lee e Wu, 2025	Sim	Llama 3 / RAG	ES	Ensaio Cont.	Sim	Sim
Linghu e Xiang, 2024	Não exp.	Não exp.	Não exp.	Não exp.	Não exp.	Não
Lohr <i>et al.</i> , 2024	Sim	LLM	ES	Validação	Não exp.	Não
Lyu <i>et al.</i> , 2024	Sim	OpenAI API	ES	Quase-Exp.	Sim	Sim
Mentari <i>et al.</i> , 2024	Sim	Análise Log	ES	Est. Retros.	Sim	Não
Ma <i>et al.</i> , 2024	Sim	ChatGPT API	ES	Ensaio Cont.	Sim	Não
Marougkas <i>et al.</i> , 2025	Sim	Lógica Fuzzy	EB	Ensaio Cont.	Sim	Sim
Masetty e Vallamulla, 2025	Sim	IA Generativa	ES	Ensaio Cont.	Sim	Sim
Maskeliūnas <i>et al.</i> , 2023	Sim	Otimização	ES	Ensaio Cont.	Sim	Não

Muepu, Watanobe e Amin, 2025	Sim	CodeT5 / AST	ES	Validação	Sim	Não
Nutalapati, Velmurugan e Tiglao, 2024	Sim	KT Contínuo	Não exp.	Validação	Não exp.	Não
Wijaya e Purwarianti, 2024	Sim	Llama2 / RAG	Não exp.	Validação	Sim	Não
Santos e Cury, 2023	Sim	ChatGPT API	ES	Validação	Não exp.	Não
Oli <i>et al.</i> , 2023	Sim	NLP / Regras	EB	Ensaio Cont.	Sim	Sim
Ouyang <i>et al.</i> , 2024	Sim	ChatGPT API	ES	Quase-Exp.	Sim	Sim
Prokudin, Sychev e Denisov, 2023	Sim	AST / RDF	ES	Validação	Sim	Não
Rahman <i>et al.</i> , 2022	Sim	<i>K-Means</i>	ES	<i>Data Mining</i>	Sim	Não
Risha <i>et al.</i> , 2021	Sim	Python Tutor	ES	Quase-Exp.	Sim	Sim
Rump, Fehnker e Mader, 2021	Sim	Regras C	ES	Est. Comp.	Sim	Não
Savitha <i>et al.</i> , 2025	Sim	GPT-3	ES	Validação	Não exp.	Não
Schez-Sobrinho <i>et al.</i> , 2020	Sim	AST	ES	Validação	Não exp.	Não
Shum; Rosunally; Munir, 2025	Sim	Lógica Fuzzy	ES	Validação	Sim	Não
Sychev e Shashkov, 2025	Sim	PNL / AST	ES	Est. Cego	Sim	Não
Sychev, Penskov e Prokudin, 2022	Sim	AST / RDF	ES	Est. Cego	Sim	Sim
Sychev, Prokudin e Denisov, 2023	Sim	AST / Grafos	ES	Validação	Sim	Não
Taylor <i>et al.</i> , 2024	Sim	LLM	ES	Validação	Sim	Não
Tho, 2024	Sim	Filt. Colab.	ES	Quase-Exp.	Sim	Sim
Troussas <i>et al.</i> , 2024	Sim	OpenAI API	ES	Ensaio Cont.	Sim	Sim
Troussas <i>et al.</i> , 2025	Sim	GPT API / Fuzzy	ES	Validação	Sim	Sim
Vemula, 2024	Sim	Llama-3	ES	Validação	Sim	Sim
Vesin <i>et al.</i> , 2022	Sim	<i>Elo-Rating</i>	ES	Est. Campo	Sim	Não
Vieira, Menezes e Netto, 2024	Sim	Algoritmos	ES	Validação	Sim	Não
F. Wang, 2025	Sim	LLM	ES	Ensaio Cont.	Sim	Sim
H. Wang <i>et al.</i> , 2025	Sim	IA Convers.	ES	Ensaio Cont.	Sim	Sim
Wiggins <i>et al.</i> , 2021	Sim	ITS Blocos	EB	Est. Retros.	Sim	Não
Wu e Wan, 2025	Sim	Alg. Genéticos	ES	Validação	Sim	Sim
Syaifudin <i>et al.</i> , 2021	Sim	APIs Android	ES	Est. Observ.	Sim	Não
Yang <i>et al.</i> , 2024	Sim	ChatGPT API	ES	Quase-Exp.	Sim	Sim

Zabala e Narman, 2024	Sim	OpenAI API	ES	Est. Piloto	Sim	Não
Zhao <i>et al.</i> , 2025	Sim	LLM	ES	Quase-Exp.	Sim	Sim

Legenda: **Feedback:** Sim (o agente fornece *feedback* pedagógico ao estudante); Não exp. (informação não explicitada no estudo). **Contexto:** ES (ensino superior); EB (educação básica); EP (educação profissionalizante); Não exp. (não explicitado no estudo). **Método:** Quase-Exp. (quase-experimento); Ensaio Cont. (ensaio controlado); Est. Campo (estudo de campo); Est. Caso (estudo de caso); Est. Retros. (estudo retrospectivo); Est. Observ. (estudo observacional); Est. Piloto (estudo piloto); Est. Comp. (estudo comparativo); Est. Cego (estudo com revisão cega); *Data Mining* (mineração de dados); Validação (validação técnica ou prova de conceito). **Estatística:** Sim (uso de análise estatística); Não exp. (informação não explicitada no estudo). **Controle:** Sim (presença de grupo de controle); Não (ausência de grupo de controle).

Fonte: Elaborado pelos autores (2026).

3.2 Arquiteturas e abordagens tecnológicas (QP1)

A evolução das arquiteturas computacionais evidencia uma transição progressiva de modelos baseados em regras estritas para arquiteturas híbridas mais complexas. Essa trajetória demonstra o aumento da sofisticação dos mecanismos de inferência e processamento semântico utilizados para apoiar a personalização e a mediação pedagógica.

Nos anos iniciais do recorte (2020-2022), predominam abordagens fundamentadas em modelos simbólicos e representações formais. Estudos como os de Anikin, Sychev e Denisov (2021) e Alshaikh, Tamang e Rus (2021) estruturam seus Sistemas Tutores Inteligentes priorizando a avaliação estrutural do código por meio de Árvores de Sintaxe Abstrata (ASTs) e mecanismos determinísticos. Nessas abordagens, o controle explícito do domínio é utilizado como estratégia para garantir maior precisão na inferência e no acompanhamento do processo de aprendizagem.

Nesse mesmo período, alternativas voltadas à automação da geração de dicas começaram a ser exploradas, como o uso de Aprendizado por Reforço apresentado por Efremov, Ghosh e Singla (2020). Posteriormente, estudos como os de Krouska *et al.* (2023) e Chrysafiadi e Virvou (2024) ampliaram a modelagem adaptativa por meio da Lógica Difusa e de Sistemas de Recomendação. Apesar dos avanços, essas arquiteturas frequentemente apresentavam limitações relacionadas à escalabilidade e à dependência de engenharia de conhecimento previamente estruturada.

Estudos de Frankford *et al.* (2024) e Zhao *et al.* (2025) demonstram que arquiteturas baseadas em redes neurais ampliaram a capacidade de interação em linguagem natural, permitindo que os sistemas deixassem de atuar apenas na validação estática do código para oferecer explicações contextualizadas, geração de exemplos e suporte à resolução de problemas.

Entretanto, limitações relacionadas à confiabilidade das respostas e à possibilidade de geração de informações imprecisas impulsionaram a adoção de arquiteturas híbridas baseadas em mecanismos de recuperação e validação da informação. Estratégias fundamentadas em Geração Aumentada por Recuperação e integração com grafos de conhecimento são apresentadas por Jiang *et al.* (2025) e Wijaya e Purwianti (2024) como alternativas relevantes para reduzir inconsistências e alinhar as respostas aos objetivos pedagógicos da disciplina.

Paralelamente, observa-se o crescimento de arquiteturas multiagentes, nas quais diferentes componentes especializados executam tarefas complementares de análise sintática, recomendação e geração de *feedback* personalizado (Yiwei Li *et al.*, 2025; H. Wang *et al.*, 2025). As principais relações entre arquiteturas computacionais

e funções pedagógicas identificadas na literatura encontram-se sintetizadas no Quadro 3.

Quadro 3 – Relação entre arquiteturas tecnológicas e funções pedagógicas dos agentes inteligentes

Categoria tecnológica	Características computacionais	Função pedagógica predominante
Sistemas Tutores Inteligentes	Modelagem do domínio, aluno e tutor para acompanhamento adaptativo	Tutoria adaptativa, <i>scaffolding</i> e avaliação automática
Grandes Modelos de Linguagem	Processamento semântico e geração contextualizada de código	Explicação de erros, geração de dicas e suporte interativo
<i>Chatbots</i> conversacionais	Interfaces baseadas em processamento de linguagem natural	<i>Feedback</i> automático e suporte metacognitivo
Lógica <i>Fuzzy</i> e Aprendizado por Reforço	Modelagem adaptativa baseada em incerteza e recompensas	Ajuste dinâmico de dificuldade e detecção de dificuldades
Sistemas de recomendação e grafos	Estruturas semânticas e personalização baseada em histórico	Recomendação personalizada e organização de trajetórias
Arquiteturas híbridas e multiagentes	Integração entre LLMs, <i>RAG</i> e agentes especializados	Colaboração assistida por IA e validação complexa

Fonte: Elaborado pelos autores (2026).

3.3 Funções pedagógicas e estratégias de mediação (QP2)

A análise do *corpus* evidencia que os agentes inteligentes passaram a desempenhar funções pedagógicas mais amplas do que a simples correção automática de código, ampliando as possibilidades de acompanhamento individualizado da aprendizagem.

O nível mais fundamental de mediação pedagógica identificado na literatura concentra-se na avaliação automática e no fornecimento de *feedback* corretivo. Em vez de apenas rejeitar submissões incorretas, sistemas contemporâneos passaram a traduzir mensagens de compilação em explicações contextualizadas e mais compreensíveis ao estudante. Como demonstram os estudos de Ahmed *et al.* (2020) e Taylor *et al.* (2024), essa estratégia busca reduzir a carga cognitiva associada à interpretação de erros sintáticos e lógicos, permitindo que o aluno compreenda a natureza da falha antes de reformular sua solução.

Além da correção reativa, a literatura também evidencia um avanço das estratégias de *scaffolding* e geração progressiva de dicas. Trabalhos como os de Risha *et al.* (2021) e Jiang *et al.* (2025) ilustram como os agentes inteligentes passaram a decompor problemas complexos em etapas menores e mais gerenciáveis. Em vez de fornecer respostas prontas, essas ferramentas oferecem suporte gradual por meio de *next-step hints*, orientando o raciocínio do estudante ao longo da resolução do problema. De forma complementar, estudos de Prokudin, Sychev e Denisov (2023) e Sychev e Shashkov (2025) demonstram que a geração automatizada de exercícios e desafios personalizados amplia as oportunidades de prática e reforço em diferentes níveis de dificuldade.

Essas interações tornam-se ainda mais sofisticadas com a integração de mecanismos de tutoria adaptativa e recomendação personalizada. O objetivo dessas estratégias é alinhar o nível de dificuldade das atividades ao perfil e ao progresso do estudante em tempo real. Autores como Vesin *et al.* (2022) e Muepu, Watanobe e Amin (2025) propõem arquiteturas capazes de mapear o histórico de resoluções dos alunos para recomendar novos conteúdos e trajetórias de aprendizagem mais adequadas ao seu desempenho. Esse direcionamento frequentemente é associado a mecanismos de detecção preventiva de impasses (*struggle detection*), como investigado por Dannath, Deriyeva e Paaßen (2024), permitindo que intervenções

pedagógicas ocorram em momentos nos quais o sistema identifica sinais de dificuldade ou estagnação.

Por fim, os estudos mais recentes ampliam a atuação dos agentes inteligentes para dimensões metacognitivas, afetivas e sociais da aprendizagem. Estratégias de suporte motivacional, incluindo gamificação e utilização de avatares guiados por emoção (Jebara; Nawahdah; Sawalha, 2025; Masetty; Vallamulla, 2025), passaram a ser empregadas para estimular o engajamento e a persistência dos estudantes diante de tarefas complexas. Paralelamente, a Inteligência Artificial também assume funções de mediação em ambientes colaborativos. Pesquisas como as de Ouyang *et al.* (2024) e H. Wang *et al.* (2025) demonstram como agentes inteligentes participam de dinâmicas de *pair programming*, favorecendo processos de reflexão conjunta e aproximando a interação homem-máquina de práticas colaborativas de aprendizagem.

3.4 Benefícios, desafios e limitações (QP3)

A literatura analisada evidencia que a inserção de agentes inteligentes no ensino de programação produz impactos ambivalentes. Se, por um lado, essas ferramentas ampliam significativamente as possibilidades de tutoria individualizada, por outro, também introduzem desafios pedagógicos, técnicos e éticos relacionados ao uso de Inteligência Artificial em ambientes educacionais.

Os estudos reportam ganhos consistentes em desempenho e engajamento associados à personalização da aprendizagem. Sistemas adaptativos permitem ajustar dinamicamente o nível de dificuldade e o ritmo das atividades (Chrysafiadi; Virvou, 2024; Gutierrez; Denny; Luxton-Reilly, 2024), enquanto mecanismos de *feedback* imediato e estratégias afetivo-motivacionais contribuem para reduzir a frustração inicial frequentemente observada em disciplinas introdutórias de programação (Jebara; Nawahdah; Sawalha, 2025; Shum; Rosunally; Munir, 2025). Essa mediação também está associada à redução do tempo de depuração e ao aumento das taxas de conclusão de exercícios (Ahmed *et al.*, 2020; Lohr *et al.*, 2024). Contudo, a efetividade dessas estratégias frequentemente encontra limitações relacionadas à implementação dos sistemas. A literatura aponta que a alta sensibilidade à calibração dos algoritmos e o problema de *cold-start* (situação em que o sistema ainda possui poucos dados sobre estudantes iniciantes) podem restringir a precisão das recomendações durante as etapas iniciais das disciplinas (Vesin *et al.*, 2022).

Apesar dos ganhos reportados em desempenho e engajamento, a literatura também aponta desafios pedagógicos relevantes relacionados ao uso intensivo de agentes inteligentes. Estudos de Jiang *et al.* (2025) e H. Wang *et al.* (2025) indicam que a oferta excessiva de explicações e soluções automatizadas pode favorecer formas superficiais de resolução de problemas, reduzindo o esforço cognitivo necessário à consolidação do raciocínio computacional.

Essas limitações tornam-se ainda mais relevantes diante da possibilidade de geração de respostas plausíveis, porém incorretas, por Grandes Modelos de Linguagem. Pesquisas como a de Akçapinar e Sidan (2024) demonstram que parte dos estudantes tende a aceitar automaticamente respostas produzidas pela IA sem realizar verificações críticas adequadas. Esse cenário evidencia a importância de estratégias pedagógicas voltadas ao desenvolvimento do letramento em Inteligência Artificial e da autonomia intelectual dos estudantes.

Por fim, a consolidação dessas evidências ainda é limitada por fragilidades metodológicas presentes nos próprios estudos. Predominam investigações

conduzidas em ambientes laboratoriais ou em disciplinas introdutórias de programação (CS1), frequentemente baseadas em amostras reduzidas, intervenções de curta duração e escalas de percepção autorreportadas, o que amplia o risco de vies de desejabilidade social (Jebara; Nawahdah; Sawalha, 2025; Lai; Lin, 2025). Observa-se também limitação quanto à generalização dos resultados, uma vez que as validações empíricas permanecem concentradas em determinados contextos socioculturais.

3.5 Contextos educacionais, linguagens e distribuição sociocultural (QP4)

A integração da IA no ensino de programação ocorre de maneira assimétrica quando analisada na perspectiva dos níveis educacionais, das linguagens de programação adotadas e dos contextos socioculturais investigados. Essa heterogeneidade reflete tanto as restrições técnicas associadas à modelagem de domínios computacionais específicos quanto às desigualdades estruturais presentes nos diferentes ambientes educacionais.

A predominância das pesquisas no ensino superior, especialmente em disciplinas introdutórias de Computação e Engenharia, está associada aos índices de reprovação e evasão observados nesses cursos (Ahmed *et al.*, 2020; Lyu *et al.*, 2024). A análise quantitativa dos 81 estudos revela a consolidação de Python e Java como as principais linguagens utilizadas para a validação de agentes inteligentes. O ensino de Python lidera o *corpus* analisado, estando presente em 24 investigações (29,6%), seguido pelo uso de Java em 18 estudos (22,2%). A elevada legibilidade sintática, a ampla adoção acadêmica e a disponibilidade de grandes repositórios de código nessas linguagens favorecem o treinamento e a aplicação de Grandes Modelos de Linguagem em contextos educacionais (Alanazi *et al.*, 2025; Yang *et al.*, 2024). Em contraste, linguagens estruturadas e orientadas a sistemas, como C e C++, respondem por 14 estudos (17,2%) e aparecem predominantemente associadas a tarefas específicas de depuração de baixo nível, gerenciamento de memória e interpretação de mensagens de compilação (Sychev; Prokudin; Denisov, 2023; Taylor *et al.*, 2024). Os demais estudos distribuem-se entre linguagens voltadas ao desenvolvimento web, abordagens agnósticas e intervenções direcionadas à educação básica.

Na educação básica, observa-se uma adaptação do desenho pedagógico das intervenções mediadas por IA. Com o objetivo de reduzir a sobrecarga cognitiva associada à sintaxe formal, os agentes inteligentes passam a atuar em ambientes de programação visual baseados em blocos, como *Scratch* e plataformas robóticas educacionais. Nesse contexto, o foco das intervenções desloca-se da correção sintática para o desenvolvimento do pensamento computacional e da lógica algorítmica em estudantes iniciantes (Błażejowska *et al.*, 2023; Cárdenas-Cobo *et al.*, 2024; Wiggins *et al.*, 2021). Em contrapartida, a educação técnica e profissionalizante ainda representa um campo menos explorado na literatura, possivelmente em razão da elevada especificidade dos domínios industriais e das dificuldades de adaptação dessas tecnologias a ambientes produtivos complexos (Eilermann *et al.*, 2023).

Além das diferenças relacionadas às linguagens e aos níveis de ensino, a distribuição geográfica das pesquisas evidencia desigualdades socioeconômicas relevantes. Historicamente, o desenvolvimento e a validação de tutores inteligentes concentraram-se em contextos classificados como *WEIRD* (*Western, Educated, Industrialized, Rich, and Democratic*), nos quais os estudantes geralmente possuem maior acesso prévio à infraestrutura tecnológica e ao letramento digital (Brown *et al.*, 2025; Lyu *et al.*, 2024). Entretanto, a literatura mais recente demonstra um movimento

de descentralização das investigações, com o crescimento de estudos conduzidos em contextos não *WEIRD*, incluindo países asiáticos, latino-americanos e do Oriente Médio (Cárdenas-Cobo *et al.*, 2024; Ouyang *et al.*, 2024). A implementação dessas tecnologias em diferentes realidades educacionais evidencia que a eficácia dos agentes inteligentes depende de adaptações socioculturais específicas, como a oferta de interfaces bilíngues e a adequação a normas culturais e educacionais locais (Alanazi *et al.*, 2025).

Por fim, a expansão global dos agentes inteligentes introduz uma discussão relevante sobre inclusão e equidade digital. Embora plataformas adaptativas apresentem potencial para ampliar o acesso à tutoria personalizada em populações socialmente desfavorecidas (Katona; Gyonyoru, 2025a), a efetividade dessas soluções permanece condicionada à disponibilidade de conectividade, infraestrutura computacional e acesso adequado a dispositivos tecnológicos (Cárdenas-Cobo *et al.*, 2024; Syaifudin *et al.*, 2021). A escassez de estratégias pedagógicas desenvolvidas especificamente para contextos com restrições técnicas mais severas sugere que essas tecnologias podem contribuir para ampliar desigualdades já existentes quando implementadas sem políticas adequadas de democratização do acesso digital.

3.6 Metodologias e métricas de avaliação (QP5)

A avaliação da eficácia de agentes inteligentes no ensino de programação mobiliza um conjunto diversificado de abordagens metodológicas. A complexidade envolvida na mensuração da aprendizagem mediada por Inteligência Artificial exige que os estudos combinem indicadores de desempenho acadêmico, métricas computacionais e análises comportamentais. O Quadro 4 sintetiza as principais estratégias avaliativas identificadas no *corpus*, seus focos metodológicos e as limitações recorrentes observadas nos estudos.

Quadro 4 – Abordagens metodológicas e métricas de avaliação

Abordagem avaliativa	Foco metodológico	Limitações
Experimentos empíricos e ensaios controlados	Comparação de ganho de aprendizagem via pré e pós-testes em grupos experimentais e de controle	Intervenções curtas e amostras frequentemente homogêneas ou reduzidas
Quase-experimentos e estudos longitudinais	Testes em salas de aula reais ao longo de semanas ou semestres letivos	Dificuldade em isolar o efeito da tecnologia de variáveis externas
Mineração de Dados Educacionais (EDM)	Extração de grafos, modelagem de conhecimento e validação algorítmica sobre <i>datasets</i> históricos	Validação retrospectiva sem aferição contínua da aprendizagem em tempo real
Análise de <i>logs</i> e métricas comportamentais	Monitoramento de interações em <i>IDEs</i> , uso de dicas e padrões de submissão	Ambiguidade interpretativa de métricas comportamentais isoladas
<i>Surveys</i> e escalas psicométricas	Avaliação de aceitação tecnológica, usabilidade e carga cognitiva	Dependência de dados autorreportados suscetíveis a vieses
Precisão de sistema e qualidade de código	Avaliação computacional de respostas geradas e qualidade sintática do código	Métricas sintáticas não garantem efetividade pedagógica

Fonte: Elaborado pelos autores (2026).

Grande parte das validações identificadas na literatura concentra-se em delineamentos quase-experimentais e experimentos controlados. Nos estágios iniciais de desenvolvimento ou validação de mecanismos específicos (como sistemas de diálogo socrático ou geração automatizada de dicas) predominam ensaios

laboratoriais com intervenções de curta duração (Alshaikh; Tamang; Rus, 2020). Embora experimentos controlados ampliem o rigor metodológico das avaliações, estudos ecológicos e longitudinais passaram a ganhar espaço por permitirem observar o uso contínuo dos agentes inteligentes em contextos reais de aprendizagem (Kazemitabaar *et al.*, 2024; Lyu *et al.*, 2024). Entretanto, esses delineamentos introduzem dificuldades relacionadas ao isolamento do impacto exclusivo da tecnologia diante de variáveis externas presentes no ambiente educacional.

Além da tradicional comparação entre pré e pós-testes, os estudos analisados demonstram um esforço crescente para compreender os processos de aprendizagem dos estudantes durante a interação com os agentes inteligentes. Inicialmente dependentes de instrumentos perceptivos e escalas psicométricas (Katona; Gyonyoru, 2025a), as pesquisas passaram a incorporar técnicas de Mineração de Dados Educacionais (*EDM*) e análise de *logs* de interação. Abordagens baseadas no monitoramento contínuo de submissões em Ambientes de Desenvolvimento Integrado (*IDEs*) permitem identificar padrões de aprendizagem dificilmente observáveis em avaliações tradicionais. Estudos como os de Vesin *et al.* (2022) e Huang *et al.* (2023) utilizam esses dados para inferir estratégias de resolução de problemas, enquanto Dannath, Deriyeva e Paaßen (2024) empregam métricas temporais entre submissões para identificar sinais preditivos de frustração e dificuldade (*struggle detection*). Apesar do potencial analítico dessas abordagens, a literatura também destaca limitações interpretativas importantes, uma vez que métricas comportamentais isoladas nem sempre refletem de forma precisa a intenção ou o estado cognitivo do estudante.

Por fim, os estudos analisados estabelecem uma distinção relevante entre validação algorítmica e validação pedagógica. A aferição da precisão computacional dos sistemas – seja pela extração automatizada de componentes de conhecimento (Alencar *et al.*, 2025) ou pela avaliação sintática das respostas geradas – constitui um requisito importante para o funcionamento dos agentes inteligentes, mas não garante, por si só, efetividade educacional. A literatura indica que métricas puramente técnicas podem ocultar impactos instrucionais mais amplos. Um sistema eficiente na geração automática de respostas para erros de programação, por exemplo, pode reduzir a necessidade de elaboração cognitiva do estudante durante a resolução de problemas. Nesse sentido, os resultados sugerem que a maturidade metodológica da área depende da integração entre métricas robustas de engenharia de software e indicadores capazes de avaliar aprendizagem significativa e retenção de conhecimento ao longo do tempo.

Considerações finais

Este estudo analisou o uso de agentes inteligentes no ensino de programação entre 2020 e 2025, evidenciando um campo em rápida transformação impulsionado pela expansão da Inteligência Artificial generativa e pela digitalização educacional. Os achados indicam que essas tecnologias não representam apenas avanços técnicos, mas uma reconfiguração das formas de mediação pedagógica.

Observou-se a transição de sistemas tutores baseados em regras para arquiteturas fundamentadas em Grandes Modelos de Linguagem e abordagens híbridas, ampliando possibilidades de personalização, geração de *feedback* e suporte adaptativo. Entretanto, os achados também evidenciam tensões relacionadas à preservação da autonomia discente, especialmente diante da dependência excessiva de sistemas automatizados e da confiabilidade variável das respostas geradas.

A análise revelou assimetrias importantes na produção científica, concentrada predominantemente no ensino superior, em linguagens como Python e Java e em contextos com maior disponibilidade de infraestrutura tecnológica. Além disso, verificou-se a predominância de delineamentos quase-experimentais e avaliações de curta duração, limitando a compreensão dos efeitos dessas tecnologias em longo prazo.

Cumprе reconhecer, igualmente, as limitações desta revisão sistemática. A estratégia de busca concentrou-se em estudos publicados em inglês ou português, podendo restringir a recuperação de pesquisas relevantes em literatura cinzenta ou em outros idiomas. Adicionalmente, a rápida evolução dos modelos de linguagem impõe desafios relacionados à atualização contínua do estado da arte.

Como agenda futura, destacam-se a necessidade de investigações longitudinais, a ampliação de estudos em contextos socioculturais diversos e o desenvolvimento do letramento em Inteligência Artificial como competência educacional.

Em síntese, a efetividade dos agentes inteligentes no ensino de programação depende menos da automação de tarefas e mais de sua integração a estratégias pedagógicas capazes de apoiar o raciocínio e preservar a autonomia intelectual dos estudantes.

Referências

AHMED, Umair Z. *et al.* Characterizing the pedagogical benefits of adaptive feedback for compilation errors by novice programmers. *In: Proceedings of the 42nd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '20)*, Nova York, EUA, p. 139-150, 2020. DOI: 10.1145/3377814.3381703.

AKÇAPINAR, Gökhan; SIDAN, Elif. AI chatbots in programming education: guiding success or encouraging plagiarism. **Discover Artificial Intelligence**, v. 4, n. 1, p. 87, 2024. DOI: 10.1007/s44163-024-00203-7.

ALANAZI, Manal *et al.* Examining the influence of ai on python programming education: an empirical study and analysis of student acceptance through TAM3. **Computers**, v. 14, n. 10, p. 411, 2025. DOI: 10.3390/computers14100411.

ALENCAR, Rafaella Sampaio de *et al.* Integrating expert knowledge with automated knowledge component extraction for student modeling. *In: Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization (UMAP '25)*, Nova York, EUA, p. 307-312, 2025. DOI: 10.1145/3699682.3728348.

ALSHAIKH, Zeyad; TAMANG, Lasang Jimba; RUS, Vasile. A Socratic tutor for source code comprehension. *In: International Conference on Artificial Intelligence In Education*, Ifrane, Marrocos, p. 15-19, 2020. DOI: 10.1007/978-3-030-52240-7_3.

ALSHAIKH, Zeyad; TAMANG, Lasang Jimba; RUS, Vasile. Experiments with auto-generated socratic dialogue for source code understanding. *In: International Conference on Computer Supported Education (CSEDU)*, p. 35-44, 2021. DOI: 10.5220/0010398100350044.

ALZOUBI, Omar *et al.* Learning recursion: insights from the ChiQat intelligent tutoring system. *In: International Conference on Computer Supported Education (CSEDU)*, v. 2, p. 336-343, 2020. DOI: 10.5220/0009413903360343.

ANIKIN, Anton; SYCHEV, Oleg; DENISOV, Mikhail. Ontology reasoning for explanatory feedback generation to teach how algorithms work. *In: Novelties in Intelligent Digital Systems: Proceedings of the 1st International Conference (NIDS 2021)*, Atenas, Grécia, p. 239, 2021. DOI: 10.3233/FAIA210100.

ARAÚJO, Pedro *et al.* Automatic personalisation of study guides in flipped classroom: a case study in a distributed systems course. *In: 2020 IEEE Frontiers in Education Conference (FIE)*, Uppsala, Suécia, p. 1-9, 2020. DOI: 10.1109/FIE44824.2020.9274186.

BAHREHVAR, Majid; MOSHIRPOUR, Mohammad. Agile teaching: Automated student support and feedback generation. *In: 2023 IEEE Frontiers in Education Conference (FIE)*, College Station, TX, EUA, p. 1-9, 2023. DOI: 10.1109/FIE58773.2023.10343212.

BŁAŻEJOWSKA, Gabriela *et al.* A study on the role of affective feedback in robot-assisted learning. *Sensors*, v. 23, n. 3, p. 1181, 2023. DOI: 10.3390/s23031181.

BROWN, Neil C. C. *et al.* Howzat? Appealing to expert judgement for evaluating human and AI next-step hints for novice programmers. *ACM Transactions on Computing Education*, v. 25, n. 3, art. 31, p. 1-43, 2025. DOI: 10.1145/3737885.

CÁRDENAS-COBO, Jesennia *et al.* Applying recommendation system for developing programming competencies in children from a non-weird context. *Education and Information Technologies*, v. 29, n. 8, p. 9355-9386, 2024. DOI: 10.1007/s10639-023-12156-y.

CHANDRASEKARA, Santhush *et al.* Gamifying coding education for beginners: empowering learners with HTML, CSS and JavaScript. *In: 2025 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, Colombo, Sri Lanka, p. 1-7, 2025. DOI: 10.1109/SCSE65633.2025.11030977.

CHANG, Chih-Kai; CHEN, Pin-Chen. Development and assessment of recommendation system based on behavioral intervention analysis. *In: 2022 12th International Congress on Advanced Applied Informatics (IIAI-AAI)*, Kanazawa, Japão, p. 208-213, 2022. DOI: 10.1109/IIAIAAI55812.2022.00049.

CHAPAGAIN, Jeevan; RUS, Vasile. Automated assessment of student self-explanation in code comprehension using pre-trained language models. *In: Proceedings of the 39th Annual AAAI Conference on Artificial Intelligence*, v. 39, n. 28, p. 28996-29003, 2025. DOI: 10.1609/aaai.v39i28.35169.

CHIANG, Yueh-Hui Vanessa; LIN, Yu-Chen; CHEN, Nian-Shing. Developing a course-specific chatbot powered by generative ai for assisting students' learning in a programming course. *In: 2024 IEEE International Conference on Advanced*

Learning Technologies (ICALT), Nicosia, Chipre do Norte, Chipre, p. 182-184, 2024. DOI: 10.1109/ICALT61570.2024.00059.

CHRYSAFIADI, Konstantina; VIRVOU, Maria. PerFuSIT: personalized fuzzy logic strategies for intelligent tutoring of programming. **Electronics**, v. 13, n. 23, 4827, 2024. DOI: 10.3390/electronics13234827.

DANNATH, Jesper; DERIYEVA, Alina; PAAßEN, Benjamin. Evaluating task-level struggle detection methods in intelligent tutoring systems for programming. *In: Proceedings of DELFI*, 2024. DOI: 10.18420/delfi2024_07.

EFREMOV, Aleksandr; GHOSH, Ahana; SINGLA, Adish. Zero-Shot learning of hint policy via reinforcement learning and program synthesis. *In: International Conference on Educational Data Mining (EDM 2020)*, p. 388-394, 2020.

EILERMANN, Sebastian *et al.* KIAAA: An AI assistant for teaching programming in the field of automation. *In: 2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*, Lemgo, Alemanha, p. 1-7, 2023. DOI: 10.1109/INDIN51400.2023.10218157.

FAN, Zhiyu *et al.* Software engineering educational experience in building an intelligent tutoring system. *In: 2025 IEEE/ACM 37th International Conference on Software Engineering Education and Training (CSEET)*, Ottawa, ON, Canadá, p. 75-86, 2025. DOI: 10.1109/CSEET66350.2025.00015.

FENG, Ty; LIU, Sa; GHOSAL, Dipak. Courseassist: pedagogically appropriate ai tutor for computer science education. *In: Proceedings of the 2024 on ACM Virtual Global Computing Education Conference V. 2 (SIGCSE Virtual 2024)*, Nova York, EUA, p. 310-311, 2024. DOI: 10.1145/3649409.3691094.

FISCHER, Björn *et al.* Addressing misconceptions in introductory programming: Automated feedback in integrated development environments. *In: International Conference on Education Technology and Computers (ICETC '23)*, Nova York, EUA, p. 1-8, 2023. DOI: 10.1145/3629296.3629297.

FRANKFORD, Eduard *et al.* AI-tutoring in software engineering education. *In: Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '24)*, Nova York, EUA, p. 309-319, 2024. DOI: 10.1145/3639474.3640061.

GALVÃO, Maria Cristiane Barbosa; RICARTE, Ivan Luiz Marques. Revisão sistemática da literatura: conceituação, produção e publicação. **Logeion: Filosofia da Informação**, Rio de Janeiro, v. 6, n. 1, p. 57-73, 2019. DOI: 10.21728/logeion.2019v6n1.p57-73.

GUTIERREZ, Andre del Carpio; DENNY, Paul; LUXTON-REILLY, Andrew. Automating personalized parsons problems with customized contexts and concepts. *In: Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024)*, Nova York, EUA, p. 688-694, 2024. DOI: 10.1145/3649217.3653568.

HUANG, Yun *et al.* Supporting skill integration in an intelligent tutoring system for code tracing. **Journal of Computer Assisted Learning**, v. 39, n. 2, p. 477-500, 2023. DOI: 10.1111/jcal.12757.

JEBARA, Hadeel; NAWAHDAH, Mamoun; SAWALHA, Hadeel. An empirical evaluation of emotion-aware AI chatbots effect on engagement and programming skills. *In*: **2025 International Conference on Smart Learning Courses (SCME)**, Hebron, Palestina, p. 184-189, 2025. DOI: 10.1109/SCME62582.2025.11104881.

JIANG, Bo *et al.* E2A2: An encouraging and engaging assignment assistant for learning beyond answers. *In*: **2025 IEEE World AI IoT Congress (AlloT)**, Seattle, WA, EUA, p. 507-513, 2025. DOI: 10.1109/AlloT65859.2025.11105276.

KATONA, Jozsef; GYONYORU, Klara Ida Katonane. AI-based adaptive programming education for socially disadvantaged students: Bridging the digital divide. **TechTrends**, v. 69, n. 5, p. 925-942, 2025a. DOI: 10.1007/s11528-025-01088-8.

KATONA, Jozsef; GYONYORU, Klara Ida Katonane. Integrating AI-based adaptive learning into the flipped classroom model to enhance engagement and learning outcomes. **Computers and Education: Artificial Intelligence**, v. 8, 100392, 2025b. DOI: 10.1016/j.caeai.2025.100392.

KAZEMITABAAR, Majeed *et al.* Codeaid: evaluating a classroom deployment of an llm-based programming assistant that balances student and educator needs. *In*: **Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)**, Nova York, EUA, art. 650, p. 1-20. DOI: 10.1145/3613904.3642773.

KOCH, Nadine Nicole *et al.* Enhanced code comprehension: individualized learning of code tracing with the feedback buddy. *In*: **Proceedings of DELFI**, 2024. DOI: 10.18420/delfi2024_16.

KOIKE, Kento *et al.* Compogram: development and evaluation of ITS for organizing programming-knowledge by visualizing behavior. *In*: **International Conference on Human-Computer Interaction**, p. 151-162, 2020. DOI: 10.1007/978-3-030-60152-2_12.

KROUSKA, Akrivi *et al.* Optimizing Player Engagement in an Educational Virtual Game through Fuzzy Logic-based Challenge Adaptation. *In*: **2023 8th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)**, Piraeus, Grécia, p. 1-6, 2023. DOI: 10.1109/SEEDA-CECNSM61561.2023.10470784.

KURNIAWAN, Oka *et al.* Designing for Novice Debuggers: A Pilot Study on an AI-Assisted Debugging Tool. *In*: **Proceedings of the 25th Koli Calling International Conference on Computing Education Research (Koli Calling '25)**, Nova York, EUA, art. 41, p. 1-7, 2025. DOI: 10.1145/3769994.3769997.

LAI, Chien-Hung; LIN, Cheng-Yueh. Analysis of learning behaviors and outcomes for students with different knowledge levels: A case study of intelligent tutoring system for coding and learning (ITS-CAL). **Applied Sciences**, v. 15, n. 4, p. 1922, 2025. DOI: 10.3390/app15041922.

LEE, Michael J. Auto-generated game levels increase novice programmers' engagement. **Journal of Computing Sciences in Colleges**, v. 36, n. 3, p. 70-79, 2020.

LI, Chong; LEE, Xin; WU, Xiaojin. Provide personalized programming learning for individuals based on large language models. **Alexandria Engineering Journal**, v. 132, p. 396-406, 2025. DOI: 10.1016/j.aej.2025.10.026.

LI, Wengxi *et al.* CogGen: A Learner-Centered Generative AI Architecture for Intelligent Tutoring with Programming Videos. *In: International Conference on Artificial Intelligence in Education*, Palermo, Itália, p. 11-18, 2025. DOI: 10.1007/978-3-031-98462-4_2.

LI, Ying *et al.* Intelligent tutoring for large-scale personalized programming learning based on knowledge graph. *In: 2023 IEEE Frontiers in Education Conference (FIE)*, College Station, TX, EUA, p. 1-5, 2023. DOI: 10.1109/FIE58773.2023.10342641.

LI, Yiwei *et al.* Beyond Test Cases: Multi-Agent Collaboration for Detecting Defects in Full-Score Code Implementations. *In: International Conference on Software Engineering and Knowledge Engineering (SEKE)*, Pompeia, Itália, p. 124-129, 2025. DOI: 10.18293/SEKE2025-053.

LINGHU, Hongying; XIANG, Chengguan. Research on Human-Machine Hybrid Enhanced Programming Teaching Model. *In: 2024 14th International Conference on Information Technology in Medicine and Education (ITME)*, Guiyang, China, p. 1161-1166, 2024. DOI: 10.1109/ITME63426.2024.00231.

LOHR, Dominic *et al.* Adaptive learning systems in programming education: A prototype for enhanced formative feedback. *In: Proceedings of DELFI*, 2024. DOI: 10.18420/delfi2024_57.

LYU, Wenhan *et al.* Evaluating the effectiveness of LLMs in introductory computer science education: a semester-long field study. *In: Proceedings of the Eleventh ACM Conference on Learning @ Scale*, Nova York, EUA, p. 63-74, 2024. DOI: 10.1145/3657604.3662036.

MA, Qianou *et al.* How to teach programming in the AI era? Using LLMs as a teachable agent for debugging. *In: International Conference on Artificial Intelligence in Education*, Recife, PE, Brasil, p. 265-279, 2024. DOI: 10.1007/978-3-031-64302-6_19.

MAROUGKAS, Andreas *et al.* An adaptive virtual reality game for programming education using fuzzy cognitive maps and pedagogical models. **Smart Learning Environments**, v. 12, n. 1, p. 62, 2025. DOI: 10.1186/s40561-025-00392-3.

MASETTY, Sai Veda Prakash; VALLAMULLA, Sreeja. Enhancing novice programming education through AI-Driven mentorship and project-based learning. *In: 2025 IEEE Integrated STEM Education Conference (ISEC)*, Princeton, NJ, EUA, p. 1-7, 2025. DOI: 10.1109/ISEC64801.2025.11147276.

MASKELIŪNAS, Rytis *et al.* FGPE+: The mobile FGPE environment and the Pareto-optimized gamified programming exercise selection model – An empirical evaluation. **Computers**, v. 12, n. 7, p. 144, 2023. DOI: 10.3390/computers12070144.

MENTARI, Mustika *et al.* A study of code typing problems as start-up programming practices in Java programming learning assistant system. *In: 2024 5th International Conference on Information Technology and Education Technology (ITET)*, Tottori, Japão, p. 45-50, 2024. DOI: 10.1109/ITET64267.2024.00017.

MUEPU, Daniel M.; WATANOBÉ, Yutaka; AMIN, Md Faizul Ibne. A comprehensive content-based recommendation system for programming problems through multi-faceted code analysis. **IEEE Access**, v. 13, p. 93712-93734, 2025. DOI: 10.1109/ACCESS.2025.3574246.

NUTALAPATI, Harsha; VELMURUGAN, Sachin; TIGLAO, Nestor Michael. Coding Buddy: an adaptive AI-powered platform for personalized learning. *In: 2024 International Symposium on Networks, Computers and Communications (ISNCC)*, Washington DC, DC, EUA, p. 1-6, 2024. DOI: 10.1109/ISNCC62547.2024.10759044.

OLI, Priti *et al.* Improving code comprehension through scaffolded self-explanations. *In: International Conference on Artificial Intelligence In Education*, Tóquio, Japão, p. 478-483, 2023. DOI: 10.1007/978-3-031-36336-8_74.

OUYANG, Fan *et al.* Comparing the effects of instructor manual feedback and ChatGPT intelligent feedback on collaborative programming in China's higher education. **IEEE Transactions on Learning Technologies**, v. 17, p. 2173-2185, 2024. DOI: 10.1109/TLT.2024.3486749.

PAGE, Matthew J. *et al.* The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. **BMJ**, [S. l.], v. 372, n. 71, p. 1-9, 2021. DOI: 10.1136/bmj.n71.

PROKUDIN, Artem; SYCHEV, Oleg; DENISOV, Mikhail. Learning problem generator for introductory programming courses. **Software Impacts**, v. 17, p. 100519, 2023. DOI: 10.1016/j.simpa.2023.100519.

RAHMAN, Md Mostafizer *et al.* Educational data mining to support programming learning using problem-solving data. **IEEE Access**, v. 10, p. 26186-26202, 2022. DOI: 10.1109/ACCESS.2022.3157288.

RISHA, Zak *et al.* Stepwise help and scaffolding for java code tracing problems with an interactive trace table. *In: Proceedings of the 21st Koli Calling International*

Conference on Computing Education Research (Koli Calling '21), Nova York, EUA, art.27, p. 1-10, 2021. DOI: 10.1145/3488042.3490508.

RUMP, Arthur; FEHNKER, Ansgar; MADER, Angelika. Automated assessment of learning objectives in programming assignments. *In: International Conference on Intelligent Tutoring Systems*, p. 299-309, 2021. DOI: 10.1007/978-3-030-80421-3_33.

SANTOS, Otávio Lube dos; CURY, Davidson. Challenging the confirmation bias: Using ChatGPT as a virtual peer for peer instruction in computer programming education. *In: 2023 IEEE Frontiers in Education Conference (FIE)*, College Station, TX, EUA, p. 1-7, 2023. DOI: 10.1109/FIE58773.2023.10343247.

SAVITHA, N. J. *et al.* Guild Based Online Coding Platform with Gamification Features. *In: 2025 Global Conference in Emerging Technology (GINOTECH)*, Pune, Índia, p. 1-5, 2025. DOI: 10.1109/GINOTECH63460.2025.11077053.

SCHEZ-SOBRINO, Santiago *et al.* An intelligent tutoring system to facilitate the learning of programming through the usage of dynamic graphic visualizations. **Applied Sciences**, v. 10, n. 4, p. 1518, 2020. DOI: 10.3390/app10041518.

SHUM, Lok Cheung; ROSUNALLY, Yasmine; MUNIR, Kamran. Transforming Programming Education: The Effectiveness of Motivational Scenario-Based Design in Serious Games. **IEEE Transactions on Education**, v. 68, n. 4, p. 394-406, aug. 2025. DOI: 10.1109/TE.2025.3584187.

SILVA, Isac Neto da; SOUZA, Júlio Cesar de. Inteligência Artificial e Psicologia Cognitiva: contribuições no desenvolvimento de tecnologias educacionais adaptativas na educação básica. **Revista Sítio Novo**, Palmas, v. 9, p. e1781, 2025. DOI: 10.47236/2594-7036.2025.v9.1781.

SOUSA, Ricardo Ferreira de; LAGE, Eric Fellippe Ribeiro. Educação e pandemia da covid-19: a importância das tecnologias de comunicação e informação neste cenário. **Revista Sítio Novo**, Palmas, v. 8, n. 4, p. 59-76, 2025. DOI: 10.47236/2594-7036.2024.v8.i4.59-76p.

SYAIFUDIN, Yan Watequlis *et al.* Blending Android programming learning assistance system into online android programming course. *In: 2021 9th International Conference on Information and Education Technology (ICIET)*, Okayama, Japão, p. 26-33, 2021. DOI: 10.1109/ICIET51873.2021.9419650.

SYCHEV, Oleg; PENSKOY, Nikita; PROKUDIN, Artem. Generating expression evaluation learning problems from existing program code. *In: 2022 IEEE International Conference on Advanced Learning Technologies (ICALT)*, Bucareste, Romênia, p. 183-187, 2022. DOI: 10.1109/ICALT55010.2022.00061.

SYCHEV, Oleg; PROKUDIN, Artem; DENISOV, Mikhail. Generation of code tracing problems from open-source code. *In: Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023)*, Nova York, EUA, p. 875-881, 2023. DOI: 10.1145/3545945.3569774.

SYCHEV, Oleg; SHASHKOV, Dmitry. Mass Generation of Programming Learning Problems from Public Code Repositories. **Big Data and Cognitive Computing**, v. 9, n. 3, p. 57, 2025. DOI: 10.3390/bdcc9030057.

TAYLOR, Andrew *et al.* dcc--help: Transforming the Role of the Compiler by Generating Context-Aware Error Explanations with Large Language Models. *In: Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024)*, Nova York, EUA, p. 1314-1320, 2024. DOI: 10.1145/3626252.3630822.

THO, Pham-Duc. PERS: A Personalized Recommender System for Student-Generated Questions in Programming Courses. *In: 32nd International Conference on Computers in Education (ICCE 2024): Conference Proceedings V. 1*, Quezon City, Filipinas, 2024. DOI: 10.58459/icce.2024.4913.

TROUSSAS, Christos *et al.* Evaluating ChatGPT-driven Automated Test Generation for Personalized Programming Education. *In: 2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, Dubai, Emirados Árabes Unidos, p. 194-200, 2024. DOI: 10.1109/FLLM63129.2024.10852510.

TROUSSAS, Christos *et al.* Fuzzy memory networks and contextual schemas: Enhancing ChatGPT responses in a personalized educational system. **Computers**, v. 14, p. 89, 2025. DOI: 10.3390/computers14030089.

VEMULA, Srikanth. Enriching python programming education with generative ai: Leveraging large language models for personalized support and interactive learning. *In: 2024 IEEE Frontiers in Education Conference (FIE)*, Washington, DC, EUA, p. 1-8, 2024. DOI: 10.1109/FIE61694.2024.10893561.

VESIN, Boban *et al.* Adaptive assessment and content recommendation in online programming courses: On the use of Elo-rating. **ACM Transactions on Computing Education**, v. 22, n. 3, p. 1-27, 2022. DOI: 10.1145/3511886.

VIEIRA, Angela Vitoria Mota; MENEZES, Ramayana Assunção; NETTO, Jose Francisco Magalhães. A Proposal of Adaptive Learning System for Object-Oriented Programming Education. *In: 2024 IEEE Frontiers in Education Conference (FIE)*, Washington, DC, EUA, p. 1-7, 2024. DOI: 10.1109/FIE61694.2024.10893496.

WANG, Feng Hsu. Exploring the Impact of Transparent Generative AI on Learning Outcomes Through a Human-Machine Trust Model. *In: 2025 IEEE International Conference on Advanced Learning Technologies (ICALT)*, Changhua, Taiwan, p. 229-233, 2025. DOI: 10.1109/ICALT64023.2025.00072.

WANG, Haoming *et al.* Impact of AI-agent-supported collaborative learning on the learning outcomes of university programming courses. **Education and Information Technologies**, v. 30, p. 17717-17749, 2025. DOI: 10.1007/s10639-025-13487-8.

WIGGINS, Joseph B. *et al.* Exploring novice programmers' hint requests in an intelligent block-based coding environment. *In: Proceedings of the 52nd ACM*

Technical Symposium on Computer Science Education (SIGCSE 2021), Nova York, EUA, p. 52-58, 2021. DOI: 10.1145/3408877.3432538.

WIJAYA, Owen Christian; PURWARIANTI, Ayu. An interactive question-answering system using large language model and retrieval-augmented generation in an intelligent tutoring system on the programming domain. *In: 2024 11th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA)*, Cingapura, p. 1-6, 2024. DOI: 10.1109/ICAICTA63815.2024.10763263.

WU, Zhiqiang; WAN, Shuhui. A knowledge-driven approach to AI-based personalized test paper creation in programming education. **International Journal of Knowledge Management (IJKM)**, v. 21, n. 1, 2025. DOI: 10.4018/IJKM.369825.

YANG, Albert C. M. *et al.* Enhancing python learning with PyTutor: efficacy of a ChatGPT-Based intelligent tutoring system in programming education. **Computers and Education: Artificial Intelligence**, v. 7, p. 100309, 2024. DOI: 10.1016/j.caeai.2024.100309.

ZABALA, Eric; NARMAN, Husnu S. Development and evaluation of an AI-enhanced python programming education system. *In: 2024 IEEE 15th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, Yorktown Heights, NY, EUA, p. 787-792, 2024. DOI: 10.1109/UEMCON62879.2024.10754661.

ZHAO, Yanni *et al.* Exploration of Computer Programming Teaching Reform Based on Large Language Models. *In: 2025 11th International Conference on Computing and Artificial Intelligence (ICCAI)*, Kyoto, Japão, p. 359-363, 2025. DOI: 10.1109/ICCAI66501.2025.00062.

Informações complementares

Descrição		Declaração
Financiamento		Não se aplica.
Aprovação ética		Não se aplica.
Conflito de interesses		Não há.
Disponibilidade dos dados de pesquisa subjacentes		Este trabalho não é um <i>preprint</i> e os conteúdos subjacentes ao texto da pesquisa estão disponíveis neste artigo.
Uso de Inteligência Artificial		Não há.
CrediT	Beatriz Borges de Oliveira	Funções: conceitualização, curadoria de dados, análise formal, investigação, metodologia, visualização, escrita – rascunho original e escrita – revisão e edição.
	Rogério Alves dos Santos Antoniassi	Funções: conceitualização, administração do projeto, supervisão, validação e visualização.
	Alex Fernando de Araujo	Funções: administração do projeto, supervisão e validação.

Avaliadores: Dr. João Batista Rodrigues da Silva* (Instituto Federal da Bahia, Bahia, Brasil).

Os avaliadores “A” e “C” optaram pela avaliação fechada e pelo anonimato.

Revisora do texto em português: Laura Akemi Côrtes Massunari.

Revisora do texto em inglês: Patrícia Luciano de Farias Teixeira Vidal.

Revisora do texto em espanhol: Graziani França Claudino de Anicézio.

Como citar (ABNT):

OLIVEIRA, Beatriz Borges de; ANTONIASSI, Rogério Alves dos Santos; ARAUJO, Alex Fernando de. Agentes inteligentes no ensino de programação: uma revisão sistemática da literatura. **Revista Sítio Novo**, Palmas, v. 10, p. e2114, 2026. DOI: 10.47236/2594-7036.2026.v10.2114. Disponível em:
<https://sitionovo.ifto.edu.br/index.php/sitionovo/article/view/2114>.

* Optou pela avaliação aberta e autorizou somente a divulgação da identidade como avaliador no trabalho publicado.