

# Aplicações de LLMs no ensino de Programação Orientada a Objetos: Um Mapeamento Sistemático da Literatura

Hudson Teles Camilo<sup>1</sup>, Jean Carlo Rossa Hauck<sup>1</sup>

<sup>1</sup>Programa de Pos-Graduação em Ciência da Computação - INE/CTC/UFSC  
Térreo - Sala 109 - Trindade - Florianópolis, SC - Brasil

hudsontc9@gmail.com, jean.hauck@ufsc.br

**Abstract.** *The use of Large Language Models (LLMs) in programming education has grown rapidly, yet secondary studies that specifically include Object-Oriented Programming (OOP) remain scarce. This study presents a Systematic Mapping of the Literature to analyze how LLMs have been employed in OOP education, which models are used, and how their applications are evaluated. Based on 18 primary studies, results indicate a predominance of LLMs as tools to support exercise resolution, with particular emphasis on ChatGPT and integration into development environments. Evaluation strategies primarily rely on code metrics and case studies, with limited use of controlled experiments. The findings suggest that the field is still undergoing methodological consolidation and calls for investigations that integrate technical performance, conceptual learning and instructional design.*

**Resumo.** *O uso de Grandes Modelos de Linguagem (LLMs) no ensino de programação têm crescido rapidamente, mas ainda são raros estudos secundários que contemplem a Programação Orientada a Objetos (POO). Este trabalho apresenta um Mapeamento Sistemático da Literatura para analisar como LLMs têm sido empregados no ensino de POO, quais modelos são utilizados e como suas aplicações são avaliadas. A partir de 18 estudos primários, observa-se predominância do uso de LLMs como ferramenta de apoio à resolução de exercícios, com destaque para o ChatGPT e integração a ambientes de desenvolvimento. As avaliações concentram-se em métricas de código e estudos de caso, com baixa incidência de experimentos controlados. Os achados indicam que o campo ainda se encontra em consolidação metodológica e demanda investigações que articulem desempenho técnico, aprendizagem conceitual e design instrucional.*

## 1. Introdução

A evolução recente dos Grandes Modelos de Linguagem (LLMs) têm transformado profundamente a forma como os estudantes interagem com sistemas computacionais, especialmente em contextos educacionais relacionados ao ensino de programação. Ferramentas como ChatGPT, Gemini, Copilot e outras, passaram a integrar rotinas de estudo e práticas pedagógicas ao oferecerem explicações, correções e sugestões de código (Torek et al. 2024; Yilmaz and Yilmaz 2023). Estudos recentes destacam que LLMs têm se consolidado como tecnologias de apoio capazes de ampliar o engajamento discente, automatizar práticas pedagógicas e fornecer suporte personalizado a estudantes iniciantes em programação (Alves et al. 2024; Cipriano 2024; Haindl and Weinberger 2024).

No entanto, aprender Programação Orientada a Objetos (POO) permanece uma tarefa cognitivamente exigente (Menolli and Strik 2025). Evidências indicam que estudantes iniciantes enfrentam mais dificuldades em POO do que em paradigmas procedurais, especialmente diante de conceitos como abstração, encapsulamento, herança e polimorfismo (Batiha et al. 2023; Lian et al. 2022). Essa complexidade conceitual, aliada à limitada experiência prévia dos alunos, tem sido associada a baixo desempenho, evasão e dificuldades persistentes de compreensão ao longo da formação (Boudjehem et al. 2023; Sim and Lau 2022; Xinogalos 2015). Sob o ponto de vista da carga cognitiva (Sweller et al. 2011; Sweller 1988), a necessidade de articular múltiplas abstrações, relações hierárquicas e regras de interação na POO pode elevar significativamente a carga intrínseca, exigindo estratégias pedagógicas que favoreçam a construção gradual de modelos mentais (Berssanette and de Francisco 2022; Uysal 2016; Alexandron et al. 2014).

Nesse contexto, estudos recentes sugerem que LLMs podem mitigar parte dessas dificuldades ao fornecer feedback instantâneo, diagnósticos de erros e explicações adaptadas ao nível de cada aprendiz, favorecendo ciclos de iteração mais curtos e potencialmente eficazes no desenvolvimento de competências em POO (Cipriano 2024; Cipriano et al. 2024; Kosar et al. 2023; Haindl and Weinberger 2024). Apesar desses benefícios, persistem preocupações relacionadas à dependência, superficialidade conceitual, alucinações e necessidade de supervisão docente (Mailach et al. 2025; Kaleem et al. 2024), além dos impactos ambientais (Guest et al. 2025).

Embora o uso de LLMs na educação esteja sendo amplamente explorado em estudos sobre programação introdutória e ambientes generalistas, revisões recentes da literatura concentram-se predominantemente em análises amplas do ensino de programação, sem delimitação específica por paradigma (Raihan et al. 2025; Marques and Morandini 2024; Cambaz and Zhang 2024). Essas revisões discutem aplicações de LLMs em contextos gerais, mas não organizam evidências de forma direcionada aos desafios cognitivos e particularidades conceituais e pedagógicas da Programação Orientada a Objetos. Isso reforça o estágio inicial de amadurecimento deste campo de pesquisa, carecendo de sínteses sistemáticas que consolidem práticas, ferramentas, limitações e oportunidades relativas ao uso de LLMs no ensino de POO.

Diante dessa lacuna, emerge a pergunta de pesquisa deste estudo: **“Como os LLMs têm sido utilizados no ensino de Programação Orientada a Objetos?”**. Para respondê-la, este artigo apresenta um Mapeamento Sistemático da Literatura (MSL), estruturado segundo as diretrizes de Kitchenham and Charters 2007 e Petersen et al. 2015, com o objetivo de identificar, organizar e analisar as evidências disponíveis sobre usos, ferramentas, tecnologias de suporte e estratégias de avaliação envolvendo LLMs no ensino de POO. Ao selecionar e sintetizar dezoito estudos primários recentes, este trabalho busca oferecer um panorama que subsidie decisões pedagógicas e direcione futuras investigações sobre o impacto da IA generativa no ensino de POO.

## 2. Trabalhos Relacionados

O uso de tecnologias e ferramentas baseadas em inteligência artificial tem crescido significativamente como apoio ao ensino de programação, chamando a atenção da comunidade de pesquisa em educação em computação. Nos últimos anos, estudos secundários vêm

explorando de forma abrangente o emprego de LLMs no ensino de programação, sobretudo em contextos gerais de linguagens e níveis de ensino. No entanto, observa-se que, apesar do aumento no número de estudos secundários voltados ao tema, tais estudos ainda tendem a dar um enfoque mais geral, sem delimitação específica por paradigma ou aprofundamento nas particularidades conceituais do ensino de POO.

Marques and Morandini 2024 realizaram uma revisão sistemática da literatura sobre práticas, diretrizes e estratégias para a integração de LLMs, especialmente o ChatGPT, em ambientes educacionais diversos. Analisando 20 estudos primários, os autores evidenciam que o uso de LLMs contribui para estimular engajamento e produtividade de alunos, assim como apoiar diferentes metodologias de ensino. Entretanto, apontam desafios significativos, como riscos de plágio, vieses algorítmicos, dilemas éticos e a necessidade de fortalecer habilidades de pensamento crítico em contextos mediados por IA. Embora relevante, o estudo concentra-se em ambientes educacionais amplos, não diferenciando impactos segundo paradigmas específicos de programação.

Alves et al. 2024 conduziram uma revisão de escopo sobre o uso de chatbots baseados em LLMs no ensino de programação, analisando 17 estudos primários. Os autores destacam impactos positivos no engajamento e na personalização das atividades de aprendizagem. Contudo, apontam limitações relacionadas ao papel dos docentes, acessibilidade e dependência tecnológica. A análise, entretanto, não segmenta os resultados por paradigma de programação, tratando o ensino de programação de forma generalista.

Cambaz and Zhang 2024 realizaram uma revisão sistemática centrada no uso de modelos de geração automática de código baseados em LLMs para apoio ao ensino de programação, abrangendo 21 estudos primários. Os autores identificaram benefícios relevantes, como tutoria automatizada e suporte à resolução de tarefas práticas, mas também destacaram riscos de baixa precisão, dependência excessiva e desafios éticos. Ainda assim, o estudo não delimita suas análises a contextos específicos como a POO, tratando tarefas de programação de forma geral.

Raihan et al. 2025 apresentaram uma revisão sistemática sobre o uso de LLMs no ensino de Ciência da Computação, abrangendo 125 estudos primários publicados entre 2020 e 2024. O estudo indica predominância de aplicações em cursos introdutórios, sobretudo com Python e Java, destacando ganhos em acessibilidade e automação, mas ressaltam que os LLMs não substituem o ensino formal e demandam acompanhamento docente para evitar dependência e superficialidade. Ainda assim, a análise é conduzida em um escopo amplo, sem foco no paradigma orientado a objetos.

Em síntese, as revisões existentes convergem ao destacar benefícios e riscos do uso de LLMs no ensino de programação, especialmente em níveis introdutórios. Contudo, não foi possível encontrar estudos secundários voltados especificamente ao ensino de POO, considerando suas demandas cognitivas e estruturais próprias. Assim, a presente pesquisa busca preencher essa lacuna ao sintetizar evidências, tecnologias empregadas, estratégias de avaliação e limitações sob a perspectiva específica da POO.

### 3. Metodologia

Este estudo adota a abordagem de Mapeamento Sistemático da Literatura (MSL), com base nas diretrizes metodológicas propostas por Kitchenham and Charters 2007 e

Petersen et al. 2015, que definem MSL como uma técnica estruturada voltada a identificar, classificar e sintetizar a produção científica referente a temas emergentes em computação. Diferentemente das Revisões Sistemáticas da Literatura, MSLs buscam fornecer uma visão abrangente do espaço de pesquisa, destacando tendências e lacunas metodológicas, priorizando amplitude de cobertura em vez de avaliação aprofundada da qualidade individual dos estudos (Kitchenham and Charters 2007; Biolchini et al. 2005).

### 3.1. Protocolo de Pesquisa

O protocolo deste MSL contempla: definição das perguntas de pesquisa, especificação da string de busca, critérios de seleção, processo de triagem e categorização dos estudos.

#### Perguntas de pesquisa

A pergunta principal de pesquisa é: **“Como os LLMs têm sido utilizados no ensino de Programação Orientada a Objetos?”**. A partir desta pergunta principal, são derivadas as seguintes perguntas de análise: **PA<sub>1</sub>** – *Quais os usos de LLMs no ensino de POO?*; **PA<sub>2</sub>** – *Quais os principais modelos e ferramentas baseadas em LLMs aplicados no ensino de POO?*; **PA<sub>3</sub>** – *Quais tecnologias dão suporte ao uso de LLMs no ensino de POO?*; **PA<sub>4</sub>** – *Como a aplicação de LLMs no ensino de POO é avaliada?*

#### String de busca

A string de busca apresentada na Tabela 1 foi construída considerando termos diretamente relacionados aos LLMs e às suas aplicações educacionais, incluindo nomes de modelos amplamente utilizados, combinados com termos específicos do ensino de POO. A string foi refinada por meio de testes exploratórios, utilizando artigos previamente conhecidos, com o objetivo de avaliar sua capacidade de recuperar estudos pertinentes.

**Tabela 1. String de busca**

(llm OR chatgpt OR gemini OR claude OR copilot OR deepseek OR “meta ai” OR “machine learning”) AND (“computer science” OR “computing education” OR undergraduate OR cs1) AND (“object oriented programming” OR oop OR polymorphism OR inheritance OR encapsulation)
---

#### Bases de busca

As buscas foram realizadas nas bases IEEE Xplore, ACM Digital Library, SpringerLink e Google Scholar, sem restrição de idioma. O Google Scholar foi utilizado com a intenção de ampliar a abrangência da busca (Gehanno et al. 2013).

#### Crítérios de Inclusão e Exclusão

Com base nas questões de análise, são definidos os seguintes **Crítérios de Inclusão** (CI): **CI<sub>1</sub>** – *Artigos publicados entre 2020 e 2025, intervalo posterior à popularização dos LLMs e o subsequente uso em contextos educacionais, impulsionado pelo lançamento e adoção do GPT-3 (Brown et al. 2020; Vaswani et al. 2017)*; **CI<sub>2</sub>** – *Estudos que incluem o uso de LLMs no ensino de POO*; **CI<sub>3</sub>** – *Estudos primários completos e revisados por pares*. Como **Crítérios de Exclusão** (CE), excluir estudos focados: **CE<sub>1</sub>** – *em ensino de machine learning*; **CE<sub>2</sub>** – *em avaliações de machine learning* e **CE<sub>3</sub>** – *em learning analytics*.

### 3.2. Seleção dos Estudos

A seleção dos estudos foi conduzida em dois ciclos sucessivos (Tabela 2). Na **Busca Inicial**, realizou-se a execução das buscas automáticas utilizando a string previamente definida e adaptada à sintaxe das bases selecionadas. Essa etapa ocorreu em novembro de 2025, retornando um total de 26.657 registros.

**Tabela 2. Seleção e filtragem por ciclo**

Base de Dados	Busca Inicial	Ciclo 1	Ciclo 2
IEEEExplore	389	17	8
ACM Digital Library	5.571	25	5
Google Scholar	19.100	33	3
Springer Link	1.597	17	2
<b>Total</b>	<b>26.657</b>	<b>92</b>	<b>18</b>

Considerando o elevado número de resultados recuperados, a triagem foi conduzida em dois ciclos, com base nos 100 primeiros registros ranqueados por relevância em cada uma das bases, aplicando-se sistematicamente os critérios de inclusão e exclusão previamente definidos. No **Ciclo 1**, os estudos foram filtrados com base na leitura dos títulos, eliminando-se duplicidades e registros sem acesso ao texto completo, resultando em 11 estudos classificados como relevantes e 81 como potencialmente relevantes, conforme o critério CI2. No **Ciclo 2**, procedeu-se à leitura dos resumos dos 81 estudos potencialmente relevantes, aplicando-se novamente os critérios estabelecidos. Nessa etapa, 74 estudos foram excluídos por não atenderem ao critério CI2 ou por se enquadrarem nos critérios CE1, CE2 ou CE3. Como exemplo, os estudos de (Mailach et al. 2025) e (Grévisse 2024) foram excluídos por não apresentarem foco em POO (CI2), enquanto (Kaleem et al. 2024) e (P et al. 2024) foram excluídos por tratarem predominantemente de *machine learning* sem uso de LLMs no processo de ensino-aprendizagem (CE2 e CE3). A aplicação de *backward snowballing* (Wohlin 2014) não identificou estudos adicionais relevantes. O processo resultou na seleção de 18 estudos primários, conforme apresentado na Tabela 3.

**Tabela 3. Estudos primários selecionados**

ID	Autor(es)	Contexto Educacional
S01	Cipriano et al. 2024	Graduação
S02	Torek et al. 2024	Graduação
S03	Yilmaz and Yilmaz 2023	Graduação
S04	Kosar et al. 2023	Graduação
S05	Haindl and Weinberger 2024	Graduação
S06	y Fernandez et al. 2024	Graduação
S07	Cipriano and Alves 2023	Testes em laboratório
S08	Speth et al. 2023	Graduação
S09	Menolli et al. 2024	Graduação
S10	Cipriano 2024	Graduação
S11	Andersen-Kiel and Linos 2024	Graduação
S12	Puigcercos i Vilar et al. 2025	Testes em laboratório
S13	Gumonán et al. 2025	Graduação
S14	Alturki et al. 2025	Testes em laboratório
S15	Kumar et al. 2024	Graduação
S16	Jain et al. 2024	Testes em laboratório

ID	Autor(es)	Contexto Educacional
S17	Phung et al. 2024	Testes em laboratório
S18	Dosaru et al. 2025	Graduação

Os estudos primários selecionados foram publicados nos anos de 2023 (3 estudos; 16,67%), 2024 (9 estudos; 50%) e 2025 (6 estudos; 33,3%), refletindo o interesse recente e crescente da comunidade acadêmica no uso de LLMs em contextos de ensino de POO. Quanto ao **escopo do uso de POO**, 13 dos 18 estudos (72%) situaram explicitamente suas aplicações em disciplinas ou atividades relacionadas à Programação Orientada a Objetos, enquanto 5 (28%) abordaram POO de forma indireta, inserida em contextos mais amplos de ensino de programação.

Em relação ao **contexto educacional**, a maioria dos estudos (12 de 18) foi conduzida em disciplinas de graduação, integrando o uso de LLMs a atividades regulares de sala de aula. Os demais (6 estudos) foram classificados como testes em laboratório, caracterizados por avaliações controladas com tarefas estruturadas ou *datasets* específicos de POO. Essa distribuição indica predominância de investigações em contextos pedagógicos reais, embora parte relevante das evidências derive de aplicações fora de sala de aula.

#### 4. Extração e Análise

Esta seção apresenta os resultados para as Perguntas de Análise (PAs) definidas no protocolo de pesquisa. A Tabela 4 resume as quatro perguntas de análise, a categorização dos dados extraídos e os estudos primários. As informações extraídas dos estudos primários foram agrupadas em categorias utilizando *thematic analysis* (Cruzes and Dyba 2011). Os dados brutos extraídos dos estudos primários estão disponíveis na plataforma Zenodo<sup>1</sup>.

**Tabela 4. Mapa de análise: perguntas, categorias e estudos correspondentes**

Pergunta	Categoria de Análise	Estudos Selecionados
PA1 – Quais os usos de LLMs no ensino de POO?	Ferramenta de apoio ao aluno	S01, S02, S03, S04, S05, S06, S07, S09, S10, S11, S12, S13, S15, S16
	Interação educacional	S03, S04, S06, S10, S11, S13, S15, S16
	Gerador de conteúdo	S08, S10, S12, S14, S15, S16
PA2 - Quais os principais modelos e ferramentas baseadas em LLMs aplicados no ensino de POO?	GPT	S01, S03, S04, S05, S06, S07, S08, S09, S10, S11, S13, S14, S15, S16, S17, S18
	Gemini / Bard	S01, S06, S10
	Open Source	S02, S12, S18
	Copilot	S02
	Claude	S14, S18
PA3 - Quais tecnologias dão suporte ao uso de LLMs no ensino de POO?	Acesso via Navegador	S01, S03, S04, S05, S06, S08, S09, S10, S11, S13, S14, S15, S16
	Ferramentas de Análise de Código	S02, S05, S07, S09, S10, S12, S17, S18
	Ambientes de Desenvolvimento	S02, S04, S09, S11, S15
	Plataformas de Ensino e Distribuição de Conteúdo	S04, S11, S15, S16
	Infraestrutura de Armazenamento e Deploy	S02, S08, S12, S17, S18
PA4 - Como a aplicação de LLMs no ensino de POO é avaliada?	Análise Estática do Código	S02, S05, S07, S09, S10, S11, S12, S17, S18
	Estudo de Caso com Estudantes	S03, S04, S06, S09, S10, S11, S13, S15, S16
	Experimento com Grupo de Controle	S02, S04, S05
	Desempenho dos LLMs em Tarefas reais	S02, S10, S12, S14, S17, S18

<sup>1</sup><https://doi.org/10.5281/zenodo.18705921>

#### 4.1. PA<sub>1</sub> – Quais os usos de LLMs no ensino de POO?

Os estudos que utilizam LLMs como **ferramenta de apoio ao aluno** (14 estudos) os empregam majoritariamente como suporte direto à resolução de exercícios, esclarecimento de dúvidas conceituais e fornecimento de feedback imediato sobre código orientado a objetos. Nessa modalidade, o modelo atua como tutor sob demanda, explicando conceitos fundamentais da orientação a objetos, tais como modelagem de classes, relacionamentos, herança, encapsulamento, polimorfismo e padrões de projeto, além de sugerir correções estruturais, auxiliar na depuração e propor melhorias na organização arquitetural do código. A predominância dessa categoria indica que os estudos se concentram em intervenções reativas e operacionais, centradas na resolução imediata de problemas técnicos enfrentados pelos estudantes.

Quando utilizados na **interação educacional** (8 estudos), os LLMs são mediadores em atividades mais estruturadas, como experimentos controlados, projetos colaborativos ou dinâmicas orientadas em laboratório. Nessas situações, a ferramenta influencia o fluxo da atividade pedagógica, podendo impactar estratégias de estudo, autonomia e organização das tarefas. Ainda assim, na maior parte dos casos o modelo atua como apoio complementar sem assumir papel central no redesenho curricular ou reorganização metodológica da disciplina.

Os estudos que utilizam LLMs como **geradores de conteúdo** (6 estudos) os empregam na produção de exercícios, quizzes, rubricas de avaliação ou análises automáticas de código, ampliando seu papel para além da interação direta com o aluno. Essa modalidade aponta para um movimento de automação parcial de tarefas docentes, incluindo geração de materiais e apoio à avaliação formativa. Contudo, sua menor incidência em comparação ao uso como ferramenta de apoio, sugere que são privilegiadas aplicações voltadas ao suporte discente, em detrimento de intervenções pedagógicas mais amplas.

#### 4.2. PA<sub>2</sub> – Quais os principais modelos e ferramentas baseadas em LLMs aplicados no ensino de POO?

Em relação aos modelos e ferramentas baseadas em LLMs aplicados no ensino de POO, observa-se predominância do ChatGPT (15 estudos), seguido por Gemini (4 estudos), Claude (3 estudos) e CodeLlama (2 estudos), além de ocorrências pontuais de modelos open source e versões customizadas (3 estudos). Essa distribuição revela que a maior parte das investigações se apoia em modelos proprietários amplamente disponíveis via interface web ou API. Também se observa que as implementações locais, fine-tuning ou arquiteturas abertas aparecem de forma mais limitada.

#### 4.3. PA<sub>3</sub> – Quais tecnologias dão suporte ao uso de LLMs no ensino de POO?

A maior parte das interações com LLMs no ensino de POO ocorre por meio de **acesso via navegador** (13 estudos), evidenciando uso predominantemente externo aos ambientes tradicionais de desenvolvimento (IDEs) e integração simplificada por interfaces web ou APIs.

Em 8 estudos, **ferramentas de análise de código** aparecem associadas a verificações automáticas de qualidade, testes unitários e análise de conformidade com princípios de orientação a objetos, por meio de ferramentas como Checkstyle, SonarQube e PMD. Esse padrão revela uma adoção centrada na acessibilidade, combinada a tentativas

de incorporar mecanismos técnicos já consolidados de avaliação da qualidade de código ao uso educacional dos LLMs no ensino de POO.

Os **ambientes de desenvolvimento** e as **infraestruturas de armazenamento e deploy**, identificados em 5 estudos cada, envolvem integrações com IDEs, execução local, gestão de dados e fluxos automatizados de submissão e avaliação. As **plataformas de ensino e distribuição de conteúdo**, presentes em 4 estudos, mediam a gestão de atividades, coleta de respostas e integração institucional.

Em conjunto, os resultados indicam que, no ensino de POO, os LLMs têm sido majoritariamente acoplados a ferramentas externas, como analisadores estáticos, testes automatizados e *pipelines* de submissão, enquanto integrações nativas a IDEs, sistemas de avaliação contínua ou arquiteturas pedagógicas completas ainda são pouco exploradas.

#### 4.4. PA<sub>4</sub> – Como a aplicação de LLMs no ensino de POO é avaliada?

Em relação às abordagens metodológicas e resultados observados na avaliação do uso de LLMs no ensino de POO, em nove (9) estudos que aplicam **análise estática de código**, a avaliação concentra-se na verificação de aderência a princípios de orientação a objetos. Esses estudos examinam métricas como complexidade ciclomática, coesão e acoplamento entre classes, visibilidade de atributos, conformidade com convenções de nomenclatura, modularidade e detecção de code smells, mensurando propriedades relacionadas à organização arquitetural, estrutura hierárquica e consistência interna do código OO.

Os **estudos de caso com estudantes** (9 estudos) aplicam LLMs em sala de aula, examinando desempenho discente, interação com o modelo e percepção de utilidade. Nesses casos, a avaliação envolve notas em atividades práticas, análise qualitativa das soluções submetidas e questionários que investigam clareza das explicações, apoio à resolução de problemas, contribuição para compreensão de conceitos de orientação a objetos, confiança nas respostas e percepção de autonomia ou dependência. Esses estudos capturam dimensões comportamentais e pedagógicas do uso da IA, ainda que com menor controle sobre variáveis intervenientes.

Já a **avaliação do desempenho dos LLMs em tarefas reais de POO** (6 estudos) submete os modelos a problemas envolvendo múltiplas classes, sobrescrita de métodos, padrões de projeto e hierarquias complexas, geralmente com comparação a gabaritos ou avaliação humana. Esses trabalhos apontam limitações recorrentes, como geração incompleta de métodos abstratos, hierarquias inconsistentes, uso inadequado de polimorfismo e divergências entre avaliação automática e julgamento humano, permitindo analisar a robustez dos modelos frente às exigências estruturais do paradigma orientado a objetos. Diferentemente dos estudos de caso, essa categoria focaliza a capacidade técnica do modelo, e não a aprendizagem discente, reforçando a distinção entre avaliação do artefato gerado e ganho conceitual.

Somente três (3) estudos aplicam **experimentos com grupo de controle** para comparar o desempenho entre estudantes com e sem apoio de LLMs no aprendizado de POO. Embora a incidência desse delineamento experimental seja pequena nos estudos selecionados (3 estudos; 17%), os resultados apontam que, no contexto do ensino de POO, o uso de LLMs não produziu ganhos estatisticamente significativos de desempenho para os estudantes, levando até mesmo à maior produção de erros estruturais de codificação.

## 5. Discussão

Os resultados deste MSL indicam que a integração de LLMs no ensino de POO ocorre predominantemente como extensão operacional de práticas já consolidadas, e não como transformação estrutural do desenho pedagógico. A maioria dos estudos concentra-se no uso como apoio à resolução de exercícios em estudos de caso, enquanto experimentos controlados permanecem pouco frequentes. Esse cenário, também apontado em revisões mais amplas sobre LLMs na educação (Raihan et al. 2025; Marques and Morandini 2024; Cambaz and Zhang 2024), sugere que o campo ainda está em fase de consolidação empírica, com predominância de abordagens exploratórias e avaliativas.

Para docentes de POO, esse padrão indica que o uso de LLMs tende a favorecer a produtividade imediata em atividades de escrita, correção e depuração de código OO, sem substituir a mediação necessária à construção de modelos mentais orientados a objetos (Xinogalos 2015). Embora o suporte automatizado possa reduzir a carga extrínseca associada a erros sintáticos ou estruturais (Sweller 1988; Sweller et al. 2011), sua adoção sem avaliação crítica pode reforçar a dependência e reduzir o engajamento em processos de apropriação conceitual, como observado em estudos que relatam alta confiança nas respostas do modelo e uso frequente como solucionador direto de exercícios (Gumonon et al. 2025; Yilmaz and Yilmaz 2023; y Fernandez et al. 2024; Mailach et al. 2025; Kaleem et al. 2024). Assim, o impacto pedagógico dos LLMs parece depender menos da sofisticação técnica do modelo e mais de sua articulação ao planejamento instrucional.

Para pesquisadores em educação em computação, os achados revelam uma lacuna metodológica relevante. Embora análises estáticas de código mensurem coesão, acoplamento, complexidade e conformidade estrutural, e estudos de caso capturem percepções e desempenho em sala de aula, poucos trabalhos investigam ganhos de aprendizagem comparativos em delineamentos experimentais robustos (Torek et al. 2024; Kosar et al. 2023; Haindl and Weinberger 2024). Esse cenário sugere que a produção de evidências sobre a aprendizagem efetiva com LLMs em POO ainda é incipiente, abrindo espaço para novas investigações.

A distinção entre avaliar código e avaliar aprendizagem emerge como ponto crítico. Estudos que submetem LLMs a tarefas com múltiplas classes e padrões de projeto evidenciam limitações estruturais e divergências entre avaliação automática e julgamento humano (Phung et al. 2024; Dosaru et al. 2025). Contudo, o desempenho técnico não equivale automaticamente à compreensão conceitual pelo estudante. A qualidade estrutural do código pode coexistir com a internalização parcial de princípios de abstração e modelagem (Xinogalos 2015), reforçando a necessidade de instrumentos que avaliem o processo de raciocínio crítico e não apenas o produto final.

Para projetistas de ferramentas educacionais, os resultados indicam que soluções mais promissoras tendem a combinar LLMs com mecanismos complementares de contexto (Puigcercos i Vilar et al. 2025), avaliadores automáticos guiados por rubricas (Phung et al. 2024) e integração com ambientes de desenvolvimento. No entanto, percebe-se a ausência de ferramentas capazes de integrar geração automática, análise estrutural e mediação pedagógica em ambientes que não apenas produzam código funcional, mas apoiem explicitamente a construção de conhecimento em orientação a objetos.

Em síntese, este MSL levanta indícios de que o uso de LLMs no ensino de POO já apresenta consolidação em termos de aplicações práticas, mas ainda carece de evidências mais robustas sobre seu impacto na aprendizagem. A baixa incidência de delineamentos experimentais controlados sugere que a área de pesquisa se encontra em fase de amadurecimento metodológico. Os estudos primários sugerem que o avanço da área dependerá menos da ampliação de usos operacionais e mais da produção de evidências que articulem desempenho técnico, compreensão conceitual e *design* instrucional fundamentado.

### 5.1. Ameaças à validade

Este MSL está sujeito às ameaças à validade típicas de estudos secundários. A **exclusão de estudos relevantes** pode ter ocorrido, uma vez que a análise concentrou-se nos resultados ranqueados por relevância nas bases consultadas, embora a string de busca tenha sido refinada por testes exploratórios e aplicada em múltiplas bases reconhecidas. Quanto ao **viés de publicação**, seu impacto tende a ser limitado pelo caráter descritivo do mapeamento, que buscou caracterizar abordagens e não estimar efeitos. O **viés de seleção e extração** foi mitigado por critérios explícitos, protocolo e ciclos previamente definidos, com disponibilização pública dos dados. Por fim, a **classificação e interpretação** dos estudos pode envolver julgamento analítico, ainda que tenha sido conduzido de forma sistemática e orientado por categorias derivadas das perguntas de pesquisa.

## 6. Conclusão

Este mapeamento sistemático sintetiza o cenário atual do uso de LLMs no ensino de Programação Orientada a Objetos, evidenciando predominância de aplicações voltadas ao suporte operacional à resolução de exercícios e ao apoio imediato ao estudante. Embora se observe diversidade nas formas de aplicação, desde geração de conteúdo até avaliação automatizada, a literatura concentra-se majoritariamente em usos instrumentais, com incidência limitada de delineamentos experimentais controlados e avaliações comparativas de ganhos conceituais.

Os resultados também evidenciam uma distinção relevante entre avaliar desempenho técnico do código e avaliar o ensino conceitual de POO. Enquanto métricas estruturais e análises automatizadas permitem examinar propriedades formais do código produzido, ainda são escassos estudos que investiguem de forma robusta a internalização de conceitos como abstração, modelagem e relações entre objetos. A consolidação do uso pedagógico de LLMs nesse domínio exige, portanto, instrumentos avaliativos capazes de capturar raciocínio conceitual, e não apenas produção de código e conformidade estrutural.

Em termos práticos, o potencial dos LLMs no ensino de POO depende mais da forma como são articulados ao *design* instrucional, à mediação docente e a estratégias explícitas de validação crítica do que da sofisticação técnica dos modelos. A maturidade atual da área abre espaço para produção de evidências empíricas mais robustas, da integração entre geração automática e mecanismos de avaliação formativa, e da explicitação de fundamentos teóricos que orientem seu uso em contextos conceitualmente complexos.

## Uso de Inteligência Artificial

As ferramentas ChatGPT<sup>2</sup> e Perplexity<sup>3</sup> foram utilizadas exclusivamente para a revisão textual, sem prejuízo à autoria, originalidade e responsabilidade intelectual dos autores.

## Referências

- Alexandron, G., Armoni, M., Gordon, M., and Harel, D. (2014). Scenario-based programming: reducing the cognitive load, fostering abstract thinking. In *Companion Proceedings of the 36th International Conference on Software Engineering, ICSE Companion 2014*, page 311–320, New York, NY, USA. Association for Computing Machinery.
- Alturki, R., Santos, J., Alliyu, U., Gan, J., and Guo, Z. A. (2025). Evaluating chatgpt to answer multi-modal exercises in computer science education. In *Proceedings of the 2025 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, pages 58–64. ACM.
- Alves, J. V. B., Gonçalves, Y. T., and Silva, H. B. (2024). Use of chatbots in programming education: A scoping review. In *Anais do XXXV Simpósio Brasileiro de Informática na Educação (SBIE 2024)*, Fortaleza, CE, Brasil. SBC.
- Andersen-Kiel, N. and Linos, P. (2024). Using chatgpt in undergraduate computer science and software engineering courses: A students’ perspective. In *IEEE Frontiers in Education Conference (FIE)*, San Francisco, USA. IEEE.
- Batiha, Q., Majid, N. A. A., Sahari, N., and Ali, N. M. (2023). Analysis of the learning object-oriented programming factors. *International Journal of Electrical and Computer Engineering (IJECE)*.
- Berssanette, J. H. and de Francisco, A. C. (2022). Cognitive load theory in the context of teaching and learning computer programming: A systematic literature review. *IEEE Transactions on Education*, 65(3):440–449.
- Biolchini, J., Mian, P. G., Natali, A. C. C., and Travassos, G. H. (2005). Systematic review in software engineering. Technical report, Technical Report, PESC/COPPE - Universidade Federal do Rio de Janeiro. Acesso em: 13 maio 2025.
- Boudjehem, R., Benyounes, A., and Lafifi, Y. (2023). Liss’nap: A peer-advising collaborative learning approach for object-oriented programming learning. *Acta Infologica*.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33.
- Cambaz, D. and Zhang, X. (2024). Use of ai-driven code generation models in teaching and learning programming: a systematic literature review. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education (SIGCSE ’24)*, SIGCSE 2024, pages 172–178, New York, NY, USA. Association for Computing Machinery.

---

<sup>2</sup><https://openai.com/chatgpt>

<sup>3</sup><https://www.perplexity.ai/>

- Cipriano, B. P. (2024). Towards the integration of large language models in an object-oriented programming course. In *Proceedings of the 2024 International Symposium on Computers in Education (SIIE)*, Lisbon, Portugal. IEEE.
- Cipriano, B. P. and Alves, P. (2023). Gpt-3 vs object oriented programming assignments: An experience report. In *Proceedings of the 2023 International Symposium on Computers in Education (SIIE)*, Lisbon, Portugal. IEEE.
- Cipriano, B. P., Alves, P., and Denny, P. (2024). A picture is worth a thousand words: Exploring diagram and video-based oop exercises to counter llm over-reliance. In *Proceedings of the 2024 Innovation and Technology in Computer Science Education V. 2 (ITiCSE)*, pages 1–2, Milan, Italy. ACM.
- Cruzes, D. S. and Dyba, T. (2011). Recommended steps for thematic synthesis in software engineering. In *2011 international symposium on empirical software engineering and measurement*, pages 275–284. IEEE.
- Dosaru, D.-F., Simion, D.-M., Ignat, A.-H., Negreanu, L.-C., and Olteanu, A.-C. (2025). Using genai to assess design patterns in student-written code. *IEEE Transactions on Learning Technologies*.
- Gehanno, J.-F., Rollin, L., and Darmoni, S. (2013). Is the coverage of google scholar enough to be used alone for systematic reviews. *BMC medical informatics and decision making*, 13(1):7.
- Grévisse, C. (2024). Comparative quality analysis of gpt-based multiple choice question generation. In Florez, H. and Leon, M., editors, *Applied Informatics. ICAI 2023*, volume 1874 of *Communications in Computer and Information Science*, pages 435–447. Springer, Cham.
- Guest, O., Suarez, M., Müller, B., van Meerkerk, E., Beverborg, A. O. G., de Haan, R., Elizondo, A. R., Blokpoel, M., Scharfenberg, N., Kleinherenbrink, A., et al. (2025). Against the uncritical adoption of ‘ai’ technologies in academia. *Digital Culture Education*, 16(2).
- Gumonan, K. M. V. C., Bernus, J. U., Godinez, V. A. L., and Sayson, J. J. Z. (2025). Chatgpt perception and usage in object-oriented programming of students from a philippine private university. In *2025 11th International Conference on Education and Training Technologies (ICETT)*, pages 50–56. IEEE.
- Haindl, P. and Weinberger, G. (2024). Does chatgpt help novice programmers write better code? results from static code analysis. *Journal of Software Engineering and Applications*, 17(2):57–73.
- Jain, S., Kapoor, V., and Mukherjee, S. (2024). Student-perceived cognitive load of llm-generated programming exercises. *IEEE Transactions on Learning Technologies*.
- Kaleem, M., Hassan, M. A., and Khurshid, S. K. (2024). A machine learning-based adaptive feedback system to enhance programming skill using computational thinking. *IEEE Access*, 12:59431–59440.
- Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Keele University and Durham University Joint Report. Acesso em: 13 maio 2025.
- Kosar, T., Ostojić, D., Liu, Y. D., and Mernik, M. (2023). Computer science education

- in chatgpt era: Experiences from an experiment in a programming course for novice programmers. *Mathematics*, 11(13).
- Kumar, Y., Manikandan, A., Li, J., and Morreale, P. (2024). Optimizing large language models for auto-generation of programming quizzes. In *2024 17th International Conference on Software Engineering and Consortium (ISEC)*, pages 1–8. IEEE.
- Lian, V., Varoy, E., and Giacaman, N. (2022). Learning object-oriented programming concepts through visual analogies. *IEEE Transactions on Learning Technologies*, 15:78–92.
- Mailach, A., Gorgosch, D., Siegmund, N., et al. (2025). ”ok pal, we have to code that now”: Interaction patterns of programming beginners with a conversational chatbot. *Empirical Software Engineering*, 30(34).
- Marques, D. and Morandini, M. (2024). Uso do chatgpt no contexto educacional: Uma revisão sistemática da literatura. In *Anais do XXXV Simpósio Brasileiro de Informática na Educação (SBIE 2024)*, Brasília, Brasil. SBC.
- Menolli, A. and Strik, B. (2025). Educational insights from code: Mapping learning challenges in object-oriented programming through code-based evidence.
- Menolli, A., Strik, B., and Rodrigues, L. (2024). Teaching refactoring to improve code quality with chatgpt: An experience report in undergraduate lessons. In *Anais do XXXVI Simpósio Brasileiro de Informática na Educação (SBIE 2024)*, Brasil. SBC.
- P, M. M., Jain, V., Khandelwal, P., Kataruka, K., and Shah, Y. (2024). Intelligent learning analytics through e-learning. In *Proceedings of the International Conference on Emerging Technologies in Computer Science for Interdisciplinary Applications (ICETCS)*, Bangalore, India. IEEE.
- Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18.
- Phung, T., Pădurean, V.-A., Cambroner, J., Gulwani, S., Kohn, T., Majumdar, R., Singla, A., and Soares, G. (2024). Rubric is all you need: Improving llm-based code evaluation with question-specific rubrics. In *Proceedings of the XXIII Brazilian Symposium on Software Quality (SBQS 2024)*, pages 1–12. ACM.
- Puigcerros i Vilar, G., Rashid, P., and Tonekaboni, N. H. (2025). Accessible and reliable ai coding tutors: Augmenting large language models with retrieval-augmented generation for java programming. In *2025 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–8. IEEE.
- Raihan, N., Siddiq, M. L., Santos, J. C., and Zampieri, M. (2025). Large language models in computer science education: A systematic literature review. In *Proceedings of the 1st ACM Conference on Computing in Computer Science Education (SIGCSETS ’25)*, SIGCSETS 2025, pages 938–944, New York, NY, USA. Association for Computing Machinery.
- Sim, T. Y. and Lau, S. L. (2022). Review on challenges and solutions in novice programming education. In *2022 IEEE International Conference on Computing (ICOCO)*, pages 55–61.
- Speth, S., Meißner, N., and Becker, S. (2023). Investigating the use of ai-generated

- exercises for beginner and intermediate programming courses: A chatgpt case study. In *Proceedings of the 2023 IEEE Global Engineering Education Conference (EDU-CON)*, Vienna, Austria. IEEE.
- Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2):257–285.
- Sweller, J., Ayres, P., and Kalyuga, S. (2011). *Cognitive Load Theory*. Springer, New York.
- Torek, A., Sorensen, E., Hahle, N., and Kennington, C. (2024). A systematic evaluation of code-generating chatbots for use in undergraduate computer science education. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education (SIGCSE)*, Portland, OR, USA. ACM.
- Uysal, M. P. (2016). Evaluation of learning environments for object-oriented programming: measuring cognitive load with a novel measurement technique. *Interactive Learning Environments*, 24(7):1590–1609.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.
- Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, pages 1–10.
- Xinogalos, S. (2015). Object-oriented design and programming: An investigation of novices' conceptions on objects and classes. *ACM Trans. Comput. Educ.*, 15(3).
- y Fernandez, C. A. F., Sánchez-Soto, E., Cisnero, J. R. A., and Juárez-Ramírez, R. (2024). Exploring the frontier of software engineering education with chatbots. *Programming and Computer Software*, 50(8):796–815.
- Yilmaz, R. and Yilmaz, F. G. K. (2023). Augmented intelligence in programming learning: Examining student views on the use of chatgpt for programming learning. *Education and Information Technologies*.