v7.n5.2026

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

O IMPACTO DA INTELIGÊNCIA ARTIFICIAL GENERATIVA NO PAPEL DO DESENVOLVEDOR DE SOFTWARE

THE IMPACT OF GENERATIVE ARTIFICIAL INTELLIGENCE ON THE ROLE OF THE SOFTWARE DEVELOPERS

EL IMPACTO DE LA INTELIGENCIA ARTIFICIAL GENERATIVA EN EL ROL DEL DESARROLLADOR DE SOFTWARE

Jonas Victor Crescencio Ferreira[1], Reudismam Rolim de Sousa[2]

e757843
https://doi.org/10.47820/recima21.v7i5.7843

PUBLICADO: 05/2026

RESUMO

O avanço da Inteligência Artificial Generativa (IAG) vem transformando diversas áreas da sociedade, especialmente o desenvolvimento de software, no qual tem impactado diretamente a atuação dos desenvolvedores. Nesse contexto, ferramentas de IAG passaram a automatizar tarefas e a apoiar atividades como escrita, depuração, refatoração e documentação de código. A pesquisa caracteriza-se como qualitativa, de natureza exploratório-descritiva, sendo conduzida por meio da comparação entre diferentes ferramentas de IAG, utilizando tarefas padronizadas e *prompts* equivalentes para avaliação de desempenho. Como resultados, observa-se que a IAG atua como uma ferramenta de apoio, ampliando a eficiência e a qualidade do software, embora as ferramentas ainda exijam validação humana constante, demandando do desenvolvedor maior responsabilidade, conhecimento técnico, visão estratégica e capacidade crítica na validação das soluções geradas. Conclui-se que a IAG não substitui o desenvolvedor, mas redefine suas funções no processo de desenvolvimento de software.

PALAVRAS-CHAVE: Inteligência Artificial Generativa. Desenvolvimento de Software. Ferramentas de IA. Produtividade. Papel do Desenvolvedor.

ABSTRACT

The advancement of Generative Artificial Intelligence (GAI) has been transforming various areas of society, especially software development, where it has directly impacted the role of software developers. In this context, GAI tools have begun to automate tasks and support activities such as code writing, debugging, refactoring, and documentation. This research is characterized as qualitative, with an exploratory-descriptive approach, and was conducted through the comparison of different GAI tools, using standardized tasks and equivalent prompts for performance evaluation. The results indicate that GAI acts as a support tool, enhancing software efficiency and code quality, although these tools still require constant human validation, demanding greater responsibility, technical knowledge, strategic thinking, and critical evaluation from developers. It is concluded that GAI does not replace developers, but redefines their roles in the software development process.

KEYWORDS: *Generative Artificial Intelligence. Software Development. Generative AI Tools. Productivity. Software Engineering.*

---

[1] Bacharelando do curso Interdisciplinar em Tecnologia da Informação na Universidade Federal Rural do Semi-Árido (UFERSA), Pau dos Ferros-RN Brasil.

[2] Doutor em Ciência da Computação pela Universidade Federal de Campina Grande (UFCG). Professor na Universidade Federal Rural do Semi-Árido (UFERSA), Pau dos Ferros-RN.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

RESUMEN
El avance de la Inteligencia Artificial Generativa (IAG) ha transformado diversas áreas de la sociedad, especialmente el desarrollo de software, donde ha impactado directamente la actuación de los desarrolladores. En este contexto, las herramientas de IAG han comenzado a automatizar tareas y apoyar actividades como la escritura, depuración, refactorización y documentación de código. La investigación se caracteriza como cualitativa, de naturaleza exploratorio-descriptiva, y fue realizada mediante la comparación de diferentes herramientas de IAG, utilizando tareas estandarizadas y prompts equivalentes para la evaluación del desempeño. Los resultados indican que la IAG actúa como una herramienta de apoyo, mejorando la eficiencia y la calidad del software, aunque estas herramientas aún requieren validación humana constante, exigiendo del desarrollador mayor responsabilidad, conocimiento técnico, visión estratégica y capacidad crítica en la validación de las soluciones generadas. Se concluye que la IAG no sustituye al desarrollador, sino que redefine sus funciones en el proceso de desarrollo de software.

PALABRAS CLAVE: *Inteligencia Artificial Generativa. Desarrollo de Software. Herramientas de IAG. Productividad. Rol de los Desarrolladores.*

1. INTRODUÇÃO

A Inteligência Artificial (IA) tornou-se uma tecnologia disruptiva, alterando profundamente a sociedade e suas estruturas (Viana, 2021). Essa tecnologia, em especial a Inteligência Artificial Generativa (IAG), apresenta crescimento acelerado e está cada vez mais presente no cotidiano das pessoas (Gama; Sousa; Gonçalves, 2025; Souza; Sousa; Gonçalves, 2025), sendo utilizada, de forma intencional ou não, em diversas situações do dia a dia (Barros, 2024).

No contexto do desenvolvimento de software, a IA auxilia em diversas tarefas, como geração de código, resolução de problemas lógicos, arquitetura de sistemas e realização de testes (Gonçalves *et al.*, 2026; Pereira, 2001; Pinheiro, 2025a; Bona; Leal, 2024; Ribeiro; Santos, 2025). Além disso, a tecnologia também é aplicada em outras áreas, como saúde, educação e contextos organizacionais, ampliando suas possibilidades de uso e impacto (Lima *et al.*, 2025; Challco *et al*., 2024; Yu *et al*., 2024). Como se observa, a IA vem apoiando diversas atividades e continua em expansão, ampliando significativamente suas possibilidades de aplicação.

Uma área que vem sendo fortemente impactada pela IA, em especial a IAG, é o desenvolvimento de software. Nesse contexto, torna-se pertinente refletir sobre o impacto dessa tecnologia no papel do desenvolvedor de software. Diante desse cenário, este trabalho realiza um estudo comparativo entre diferentes ferramentas de IAG (e.g., ChatGPT[3], Gemini[4], Monica[5]

[3] https://chatgpt.com/
[4] https://gemini.google.com/
[5] https://monica.im/

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

e Stitch[6]) em tarefas comuns associadas ao desenvolvimento de software, tais como geração de interfaces, correção de código, geração de testes e refatoração. Busca-se compreender como essas tecnologias influenciam a produtividade, a qualidade do software e a atuação dos desenvolvedores.

2. REFERENCIAL TEÓRICO

A Inteligência Artificial pode ser vista como uma tecnologia capaz de operar por meio de sistemas que realizam tarefas que exigem inteligência humana (Barbosa; Portes, 2023), podendo aprender a partir da experiência, utilizando dados para identificar padrões e aprimorar seu desempenho ao longo do tempo. Ela busca simular capacidades cognitivas humanas, como aprendizado, raciocínio e resolução de problemas (Gonçalves *et al.*, 2026; Ludermir, 2021). Nesse contexto, destacam-se três principais tipos de aprendizado: o supervisionado, o não supervisionado e o por reforço (Bianchi, 2004). No aprendizado supervisionado, o modelo é treinado com dados rotulados, nos quais as respostas corretas são previamente conhecidas e consideradas corretas (Bianchi, 2004). Já o aprendizado não supervisionado caracteriza-se pelo uso de dados que não possuem rótulos ou respostas previamente definidas (Bianchi, 2004). Por sua vez, o aprendizado por reforço baseia-se em sinais de recompensa ou penalidade, permitindo que o sistema aprenda a partir da interação com o ambiente, por meio de sinais de reforço positivos ou negativos (Bianchi, 2004).

Embora a IA tenha ganhado maior destaque recentemente, especialmente com o avanço da IAG (Monteiro, 2025), o termo Inteligência Artificial foi cunhado em 1956 para designar esse novo campo científico (Nakamiti, 2009). Entre os marcos históricos, destaca-se o software Eliza, que simulava conversas entre humanos, capacidade atualmente ampliada por ferramentas como o ChatGPT (Silva, 2024).

A IAG apresenta potencial para revolucionar a economia e a sociedade em escala global, de modo semelhante a outras tecnologias transformadoras do passado, como a internet e os smartphones. Nesse sentido, torna-se cada vez mais visível, popular e integrada ao cotidiano, além de ser percebida como uma tecnologia amplamente disponível e de fácil utilização (Matsuda 2024).

No contexto da engenharia de software, a utilização da IAG tem sido associada ao conceito de copilotagem de código, no qual ferramentas atuam como assistentes no processo de desenvolvimento, auxiliando na geração, revisão e refatoração de código. Essa abordagem

[6] https://stitch.withgoogle.com/

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

caracteriza-se como uma forma de automação assistida, na qual o desenvolvedor permanece responsável pela validação das soluções produzidas.

Entretanto, o uso dessas ferramentas levanta questões relevantes quanto à confiabilidade dos resultados, especialmente em função da possibilidade de alucinação, fenômeno no qual modelos de IA geram respostas plausíveis, porém incorretas ou inconsistentes. Além disso, a utilização inadequada pode impactar a qualidade do software e contribuir para o aumento da dívida técnica, caso o código gerado não seja devidamente analisado.

Diante dessa expansão, o desenvolvimento de software tornou-se uma das áreas diretamente influenciadas pela IA, especialmente pela automatização de tarefas e pelo apoio à produção de código (Tavares, 2025). Esse cenário impacta significativamente a atuação dos desenvolvedores e exige o uso adequado e criterioso dessa tecnologia. Nesse contexto, torna-se relevante investigar o impacto da IA no universo dos desenvolvedores de software, considerando os efeitos já observados dessa tecnologia na sociedade, com ênfase no processo de desenvolvimento de software.

3. METODOLOGIA

Diante desse cenário de ampla adoção da IAG no desenvolvimento de software, torna-se importante compreender o impacto dessa tecnologia na atuação dos profissionais da área. Para isso, este estudo adota uma abordagem qualitativa, de caráter exploratório-descritivo, aplicada à comparação do uso de diferentes ferramentas que empregam essa tecnologia em atividades de desenvolvimento de software, seguindo os procedimentos adotados por Gonçalves *et al.* (2026).

O mercado de desenvolvimento de software conta com diversas ferramentas baseadas em IAG, que oferecem suporte às diferentes etapas do ciclo de desenvolvimento (Tavares, 2025). Essas soluções visam auxiliar os desenvolvedores na escrita, manutenção, testes e compreensão do código, atuando como ferramentas de apoio que contribuem para o aumento da produtividade e da qualidade do software (Tavares, 2025). Nesse contexto, tais ferramentas auxiliam o desenvolvedor durante a escrita do código ao sugerirem trechos em tempo real, gerarem estruturas completas de classes e automatizarem a implementação de funcionalidades repetitivas. Dessa forma, tarefas que demandam maior esforço manual passam a ser realizadas de maneira mais produtiva (Tavares, 2025).

As ferramentas foram selecionadas com base em sua relevância e utilização no contexto do desenvolvimento de software e comparadas a partir das mesmas atividades e comandos

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

(prompts), buscando garantir igualdade de condições para a análise, conforme realizado por Gonçalves *et al.* (2026), contemplando tarefas como geração de interfaces, correção de código, criação de testes automatizados e refatoração.

Os experimentos foram realizados em ambiente web, considerando as versões disponíveis ao público no período da pesquisa. Os resultados produzidos por cada ferramenta foram analisados individualmente e comparados entre si, com base em critérios como acurácia das respostas, completude das soluções, legibilidade do código gerado, aderência ao prompt proposto e necessidade de intervenção humana. Esses parâmetros permitiram uma avaliação qualitativa das capacidades e limitações das ferramentas analisadas.

Como limitação do estudo, destaca-se que a análise foi realizada com base em um conjunto específico de tarefas e ferramentas, podendo não representar todos os cenários do desenvolvimento de software. Além disso, os resultados dependem das versões das ferramentas utilizadas no momento da pesquisa, podendo variar ao longo do tempo.

4. RESULTADOS E DISCUSSÃO

Nesta seção, as ferramentas de IAG são analisadas e comparadas na execução de diferentes tarefas associadas ao desenvolvimento de *software*.

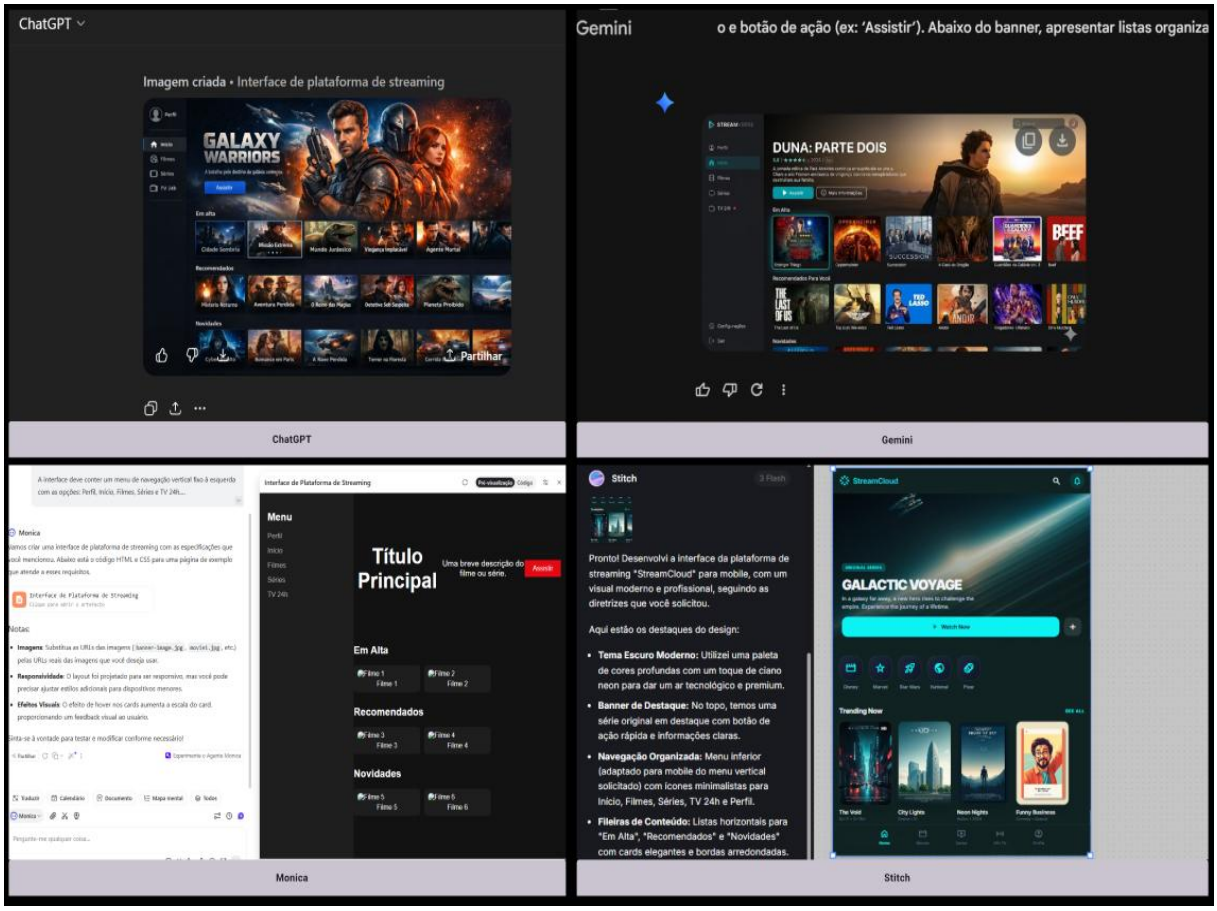
4.1. Geração de Interfaces

O primeiro aspecto analisado é a geração de interfaces. Nesse contexto, algumas ferramentas permitem a criação de interfaces completas a partir de comandos em linguagem natural, o que contribui significativamente para a redução do tempo de desenvolvimento, especialmente em aplicações front-end, nas quais o desenvolvedor pode partir de uma base pronta e personalizá-la conforme os requisitos do cliente.

A Figura 1 apresenta os resultados produzidos por diferentes ferramentas: ChatGPT, Gemini, Monica e Stitch. Para isso, foi utilizado um comando em linguagem natural solicitando a criação de uma interface para uma aplicação de streaming, especificando que “a interface deve conter um menu de navegação vertical fixo à esquerda, com as opções: Perfil, Início, Filmes, Séries e TV 24h...”.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 1. Resultado para diferentes ferramentas de IA



Fonte: Autoria Própria (2026).

Conforme pode ser visto na Figura 1, todas as ferramentas foram capazes de atender ao comando proposto, gerando automaticamente uma tela inicial com um menu de navegação vertical com opções como perfil, início, filmes, séries e TV 24 horas. Além disso, a interface gerada inclui banners em destaque, modais informativos e descrições organizadas para cada categoria, fornecendo ao desenvolvedor uma base visual completa que pode ser posteriormente customizada conforme os requisitos do projeto. No entanto, foram identificadas diferenças qualitativas entre os resultados obtidos. O ChatGPT e o Gemini apresentaram interfaces mais estruturadas e organizadas, com melhor distribuição dos elementos e maior aderência ao padrão solicitado. O Stitch, por sua vez, destacou-se pela geração de interfaces visualmente mais modernas, incorporando recursos estéticos mais atuais e elementos gráficos mais elaborados, o que contribui para uma melhor experiência visual.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Por outro lado, a ferramenta Monica apresentou limitações em relação às demais, gerando um *layout* mais simples, com menor utilização de elementos visuais, ausência de imagens e, em alguns casos, problemas de formatação. Embora tenha atendido ao comando proposto, o resultado exigiria maior intervenção do desenvolvedor para adequação estética e estrutural. Ressalta-se que o Stitch foi utilizado apenas nesta tarefa específica de geração de interfaces, enquanto as demais ferramentas foram empregadas nas etapas subsequentes do estudo.

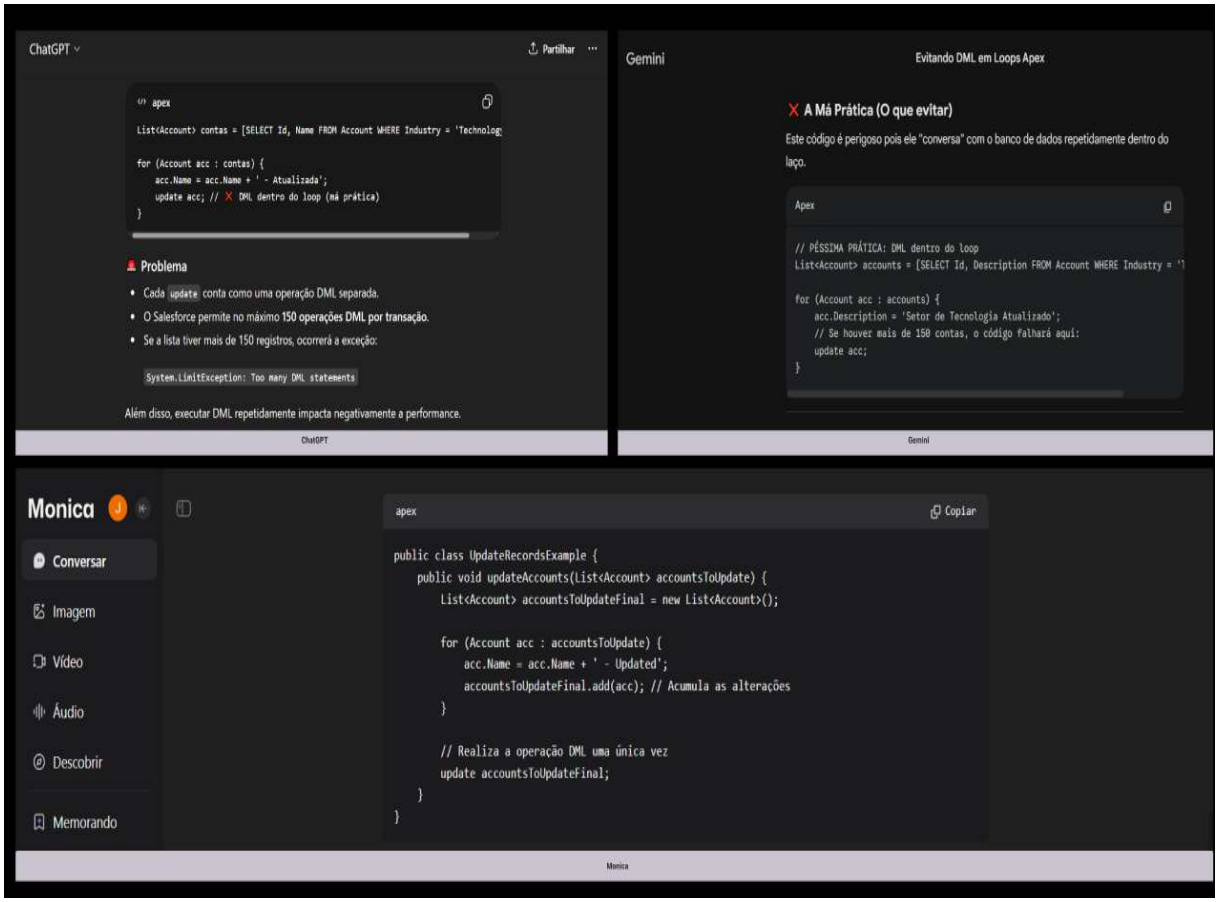
4.2. Correção de Código

As ferramentas de IA também foram analisadas quanto à sua capacidade de corrigir código, auxiliando na identificação de falhas. Para essa tarefa, foi fornecido um trecho de código Apex — linguagem de programação semelhante ao Java, utilizada no desenvolvimento de funcionalidades de acordo com as necessidades na plataforma Salesforce, um sistema de CRM voltado à gestão de clientes, vendas e processos de negócio — que ilustrava a execução de uma operação DML (*update*) dentro de um laço *for*. Essa prática pode ocasionar exceções devido à violação de limites da plataforma Salesforce, além de provocar degradação de desempenho.

Buscou-se avaliar a capacidade das diferentes ferramentas de identificar esse problema no trecho de código apresentado. Os resultados produzidos por cada ferramenta podem ser observados na Figura 2.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 2. Resultado para diferentes ferramentas de IA



Fonte: Autoria Própria (2026).

Conforme apresentado na Figura 2, todas as ferramentas foram capazes de identificar o problema no código e sugerir soluções para sua correção, incluindo trechos ajustados. No entanto, foram observadas diferenças na forma como cada ferramenta apresentou os resultados.

O ChatGPT e o Gemini forneceram explicações mais detalhadas sobre o problema identificado, descrevendo não apenas a falha, mas também suas possíveis causas e impactos na execução do código. Esse nível de detalhamento contribui para uma melhor compreensão por parte do desenvolvedor, especialmente em cenários de aprendizado ou depuração mais complexa.

Por outro lado, a ferramenta Monica apresentou uma abordagem mais direta, focando na correção do código com menor nível de explicação. Embora tenha resolvido o problema de

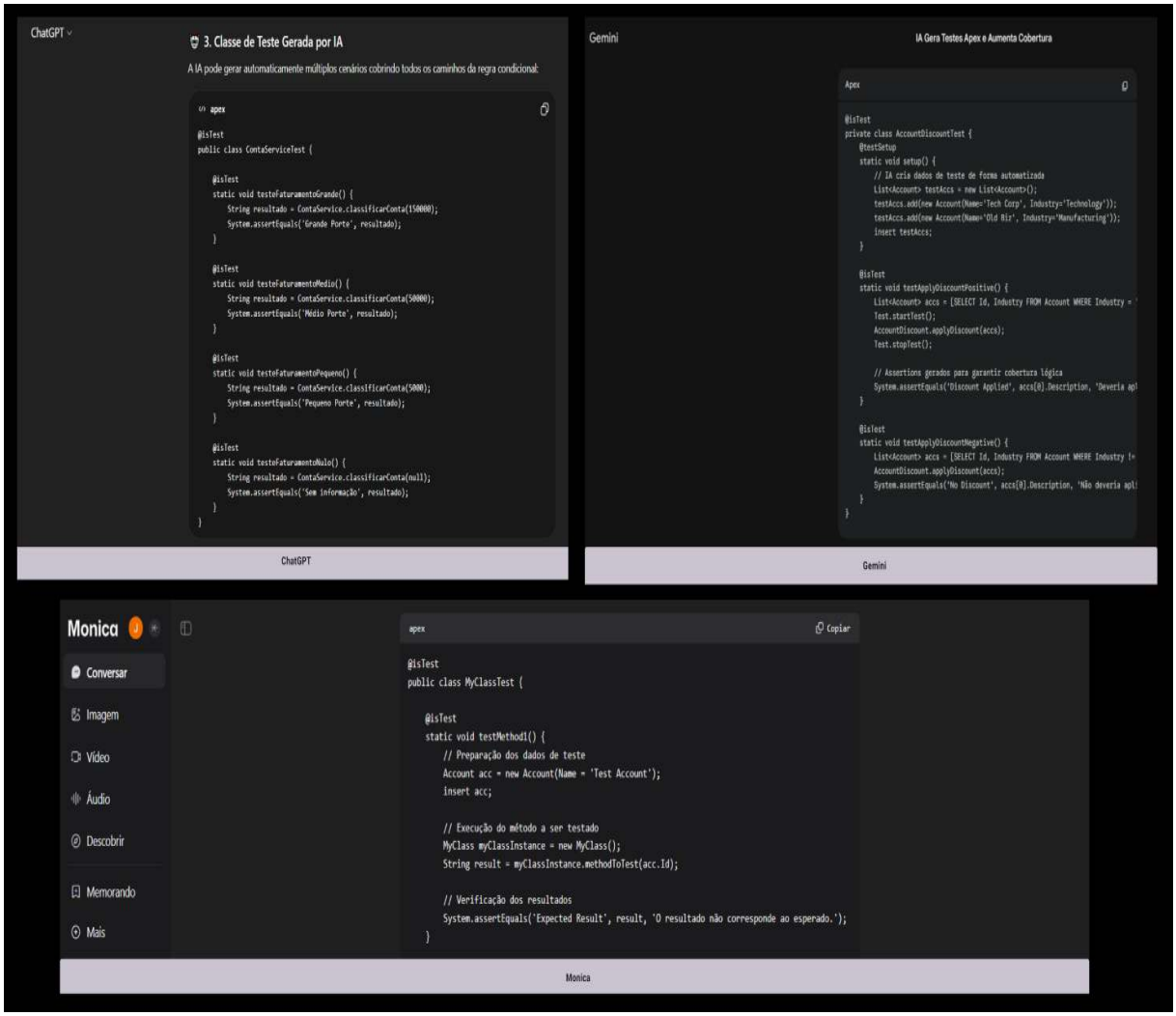
Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

forma eficiente, a ausência de detalhamento pode limitar a compreensão do desenvolvedor sobre a origem da falha e as implicações da solução proposta.

4.3. Geração de Testes Automatizados

Outra tarefa analisada no ciclo de desenvolvimento foi a capacidade das ferramentas de gerar testes automatizados. As ferramentas avaliadas foram as mesmas da seção anterior. A tarefa proposta consistiu na geração automática de uma classe de teste em Apex, com o objetivo de criar casos de teste para diferentes cenários, visando ao aumento da cobertura de código. A Figura 3 apresenta um exemplo dessa geração automática de testes.

Figura 3 – Resultado para diferentes ferramentas de IA



Fonte: Autoria Própria (2026).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Os resultados demonstram que as ferramentas de IA podem auxiliar na criação e execução de casos de teste com maior variedade de cenários, contribuindo para a identificação antecipada de falhas, ampliando a cobertura dos testes e aumentando a confiabilidade do sistema. Esse suporte favorece o processo de homologação e reduz os riscos no momento da implantação em ambiente de produção. Na tarefa proposta, ao analisar um método específico, a ferramenta pode gerar automaticamente classes de teste capazes de cobrir diferentes cenários de execução.

Observou-se, entretanto, diferenças na qualidade das respostas entre as ferramentas analisadas. O Gemini destacou-se por apresentar uma solução mais estruturada e detalhada, incluindo a utilização de *testSetup*, recurso que permite a criação de dados de teste independentes do ambiente produtivo, evitando a dependência de dados da organização. Esse aspecto contribui para maior robustez e boas práticas no desenvolvimento de testes automatizados. As demais ferramentas também foram capazes de gerar os testes, porém com menor nível de detalhamento e organização, exigindo maior intervenção do desenvolvedor para adequação às boas práticas recomendadas.

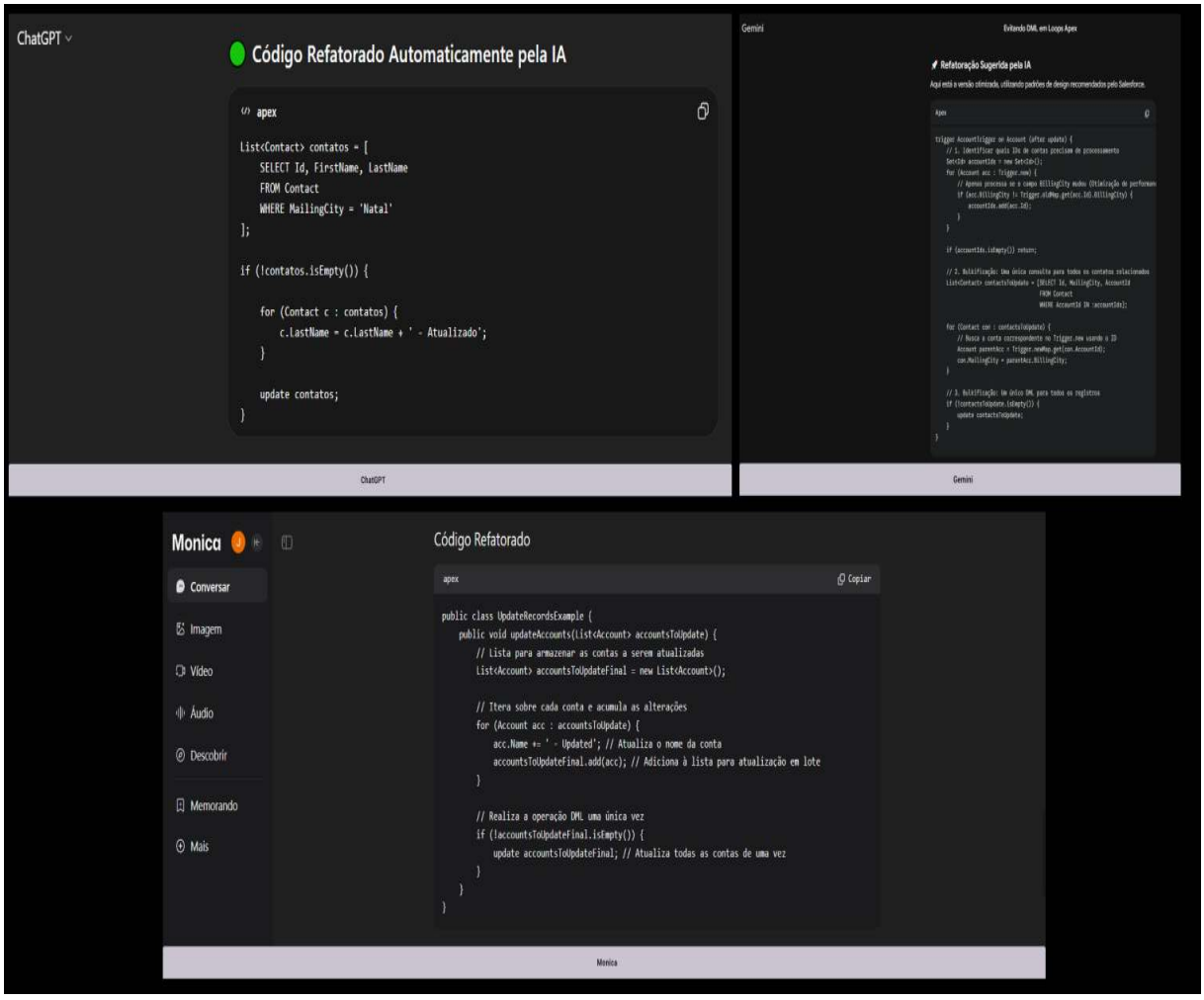
4.4. Refatoração

A última tarefa analisada envolve o uso das ferramentas de IAG para refatoração. Essa atividade consiste na modificação de um código com o objetivo de melhorar sua estrutura interna, sem alterar seu comportamento externo (Fowler, 2020). Nessa tarefa, solicitou-se à ferramenta de IA a refatoração de um código Apex previamente apresentado na Figura 2, com a correção de violações de limites da plataforma, a melhoria da legibilidade e a explicação das alterações aplicadas.

Os resultados podem ser visualizados na Figura 4. Esses resultados ilustram o trecho de código refatorado a partir da Figura 2, no qual a atualização dos registros passa a ser realizada em lote, ao final do loop, conforme as boas práticas recomendadas.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 4. Resultado para diferentes ferramentas de IA



Fonte: Autoria Própria (2026).

Como pode ser visto na Figura 4, as ferramentas buscaram melhorar a estrutura do código, tornando-o mais limpo e de fácil compreensão. No exemplo apresentado, um código mal estruturado, que poderia exceder os limites da plataforma, foi reestruturado de modo a respeitar essas restrições internas, sendo reescrito para uma versão mais clara, com a indicação das alterações realizadas e das justificativas para a mudança.

Dessa forma, a refatoração assistida por IAG possibilita a obtenção de uma versão do código que respeita os limites da plataforma, apresentando melhor organização e maior legibilidade. No entanto, foram observadas diferenças na forma como as explicações foram apresentadas pelas ferramentas. O ChatGPT destacou-se por fornecer explicações mais diretas e objetivas, separadas do código, facilitando a compreensão das alterações realizadas. Por outro

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

lado, as demais ferramentas apresentaram as explicações incorporadas ao próprio código, por meio de comentários em linha, o que também contribui para o entendimento, mas pode tornar a leitura menos fluida em alguns casos.

Apesar dessas diferenças, todas as ferramentas demonstraram capacidade de realizar a refatoração de forma eficaz, evidenciando seu potencial como suporte ao aprimoramento da qualidade do código.

5. CONSIDERAÇÕES SOBRE O USO DE IA NO DESENVOLVIMENTO

Os resultados apontam que as ferramentas de IA podem auxiliar no aumento tanto da produtividade quanto do aprendizado, além de favorecerem o processo de onboarding, permitindo que novos desenvolvedores superem barreiras iniciais (Pinheiro, 2025), compreendam regras de negócio, aprimorem o raciocínio lógico e a qualidade do código por meio da adoção de boas práticas.

Nesse sentido, observa-se que o papel do desenvolvedor de software passa por uma transformação significativa (Pinheiro, 2025). Com o avanço das ferramentas de IA, muitas organizações tendem a adaptar seus sistemas com o objetivo de integrar essas tecnologias, considerando as possibilidades de melhoria (Pinheiro, 2025). Paralelamente, os usuários vêm se habituando à presença de soluções baseadas em IA, que facilitam o acesso a funcionalidades e informações de forma mais rápida e eficiente (Heggler; Szmoski; Miquelin, 2025).

Nesse cenário, as empresas podem passar a demandar desenvolvedores para a integração de soluções de IA ao processo de produção de software (Pinheiro, 2025). Além disso, torna-se essencial a experiência em programação para a análise, validação e adaptação do código gerado por ferramentas automatizadas, uma vez que tais soluções nem sempre atendem plenamente às necessidades do desenvolvedor (Ferreira, 2025).

Diante dessas mudanças, os desenvolvedores precisam se adaptar ao uso da IA como aliada em suas atividades profissionais (Farias; Peixoto; Fausto, 2025). Embora essas tecnologias tragam benefícios significativos, elas também elevam o nível de exigência do mercado, reforçando a necessidade de aprendizado contínuo (Braga, 2024).

Em um cenário de constante evolução tecnológica, a capacidade de adaptação, a aquisição constante de novas habilidades e o acompanhamento das transformações do mercado tornam-se diferenciais competitivos fundamentais. Nesse contexto, surge a necessidade de o desenvolvedor redefinir o foco de seu aprendizado e de sua atuação profissional. O papel

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

anteriormente centrado na resolução direta de problemas e na execução técnica passa a dar lugar a uma atuação mais estratégica, na qual o profissional é responsável por decidir como determinada solução será implementada, por que ela deve ser adotada e se realmente é a abordagem mais adequada. Essas decisões exigem uma comunicação técnica mais qualificada e uma visão orientada ao produto, considerando não apenas os impactos imediatos, mas também as consequências futuras para o sistema.

Adicionalmente, observa-se um aumento significativo na responsabilidade atribuída ao desenvolvedor (Braga, 2024). Embora ferramentas de IAG sejam capazes de gerar código de forma automatizada, a responsabilidade final sobre sua qualidade, segurança e funcionamento permanece com o profissional (Ferreira, 2025). Nesse sentido, torna-se essencial que o desenvolvedor possua conhecimento suficiente para compreender, validar e, quando necessário, ajustar o código gerado, garantindo que eventuais falhas sejam corrigidas antes da implantação em ambiente de produção. Assim, a IAG atua como um recurso para otimizar tempo e produtividade, sem eliminar a necessidade de domínio técnico e senso crítico por parte do desenvolvedor.

Observa-se também que, à medida que as ferramentas de IAG passaram a gerar códigos com qualidade, por vezes superior à de desenvolvedores iniciantes em determinadas linguagens, o mercado de trabalho elevou suas exigências (Braga, 2024). As empresas passaram a demandar profissionais com conhecimento sólido em programação e que saibam utilizar a IAG de forma estratégica. Nesse contexto, desenvolvedores iniciantes acabam sendo impactados.

Como consequência, o mercado tem se tornado mais restritivo para candidatos sem experiência profissional (Braga, 2024). Diante desse cenário, percebe-se que a IAG não elimina o papel do desenvolvedor de software, mas redefine suas atribuições e eleva o nível de senioridade exigido para ingresso nas empresas. O profissional deixa de atuar apenas como executor técnico e passa a assumir uma função mais estratégica, analítica e responsável, na qual a capacidade de interpretar, validar e decidir torna-se tão importante quanto a escrita do código. Assim, a IAG consolida-se como uma aliada no desenvolvimento de software, desde que utilizada de forma consciente, ética e integrada ao conhecimento técnico humano.

Nesse contexto, a análise das ferramentas de IAG neste estudo, incluindo ChatGPT, Gemini, Monica e Stitch, permite evidenciar, na prática, como essas tecnologias se comportam em tarefas comuns do desenvolvimento de software, reforçando os impactos discutidos ao longo desta seção.

6. CONSIDERAÇÕES FINAIS

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Neste trabalho, analisou-se o impacto da Inteligência Artificial Generativa no papel do desenvolvedor de software, por meio da avaliação de diferentes ferramentas aplicadas a tarefas como geração de interfaces, correção de código, testes automatizados e refatoração. Os resultados indicam que essas ferramentas atuam como suporte ao desenvolvimento, contribuindo para a automação de tarefas, melhoria da produtividade e apoio à qualidade do software. Assim, conclui-se que a IAG não substitui o desenvolvedor, mas redefine sua atuação, exigindo maior capacidade de análise, validação crítica e tomada de decisão. Assim, o uso dessas ferramentas demanda supervisão humana constante e domínio técnico para garantir a adequação das soluções geradas.

Como trabalhos futuros, sugere-se ampliar as ferramentas e tarefas analisadas, visando aprofundar a compreensão sobre o uso da IAG no desenvolvimento de software.

DECLARAÇÃO SOBRE O USO DE IA

Os autores declaram ter utilizado ferramentas de inteligência artificial, incluindo o Gemini e o ChatGPT, exclusivamente como apoio à revisão gramatical, ao aprimoramento da coesão textual e à adequação das transições entre parágrafos deste manuscrito. Todas as sugestões geradas foram analisadas, revisadas e aprovadas pelos autores, que assumem integral responsabilidade pelo conteúdo final do trabalho.

AGRADECIMENTOS

Agradecemos aos grupos LIS — Laboratório de Inovações em *Software* — e LISA — Laboratório de Inovações em *Software* e Automação — pelo apoio a este trabalho, bem como à UFERSA — Universidade Federal Rural do Semi-Árido — pelo financiamento, por meio da Pró-Reitoria de Pesquisa e Pós-Graduação (PROPPG), através dos Editais PROPPG nº 25/2025, nº 26/2025 e nº 27/2025.

REFERÊNCIAS

BARBOSA, Lucia Martins; PORTES, Luiza Alves Ferreira. A inteligência artificial. **Revista Tecnologia Educacional**, Rio de Janeiro, n. 236, p. 16–27, jan./mar. 2023. Disponível em: https://abt-br.org.br/wp-content/uploads/2023/03/RTE_236.pdf. Acesso em: 28 mar. 2026.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

BARROS, A. Da Máquina à Emoção: Percepções do Uso da Inteligência Artificial no Desenvolvimento da Inteligência Emocional em Ambientes Educacionais. **Revista Tópicos**, Rio de Janeiro, v. 2, n. 10, p. 1-40, 2024. ISSN: 2965-6672. Disponível em: https://revistatopicos.com.br/artigos/da-maquina-a-emocao-percepcoes-do-uso-da-inteligencia-artificial-no-desenvolvimento-da-inteligencia-emocional-em-ambientes-educacionais. Acesso em: 15 fev. 2026.

BIANCHI, Reinaldo Augusto da Costa. **Uso de heurísticas para a aceleração do aprendizado por reforço**. 2004. Tese (Doutorado em Engenharia) – Universidade de São Paulo, São Paulo, 2004. Disponível em: https://www.teses.usp.br/teses/disponiveis/3/3141/tde-28062005-191041/publico/tese-bianchi.pdf. Acesso em: 28 mar. 2026.

BONA, Anderson Andrei de; LEAL, Eduardo Ribeiro Pereira. Engenharia de software cognitiva: integração de IA, DevOps e arquiteturas cloud-native. **Lumen et Virtus**, v. 15, n. 43, p. 9207–9231, 2024. DOI: 10.56238/levv15n43-115. Disponível em: https://periodicos.newsciencepubl.com/LEV/article/view/2518. Acesso em: 15 fev. 2026.

BRAGA, Victor Paz de Farias. **O impacto das ferramentas baseadas em IA generativa no mercado de desenvolvimento de software**. 2024. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade Federal de Campina Grande, 2024. Disponível em: https://dspace.sti.ufcg.edu.br/handle/riufcg/38355. Acesso em: 28 mar. 2026.

CHALLCO, Geiser Chalco et al. **Inteligência artificial generativa na educação**. 2024. E-book. Disponível em: https://iaedu.nees.ufal.br/wp-content/uploads/2025/04/NT-1-Inteligencia-Artificial-Generativa-na-Educacao.pdf. Acesso em: 22 jan. 2026.

FARIAS, Carolainy Tiemy Simões; PEIXOTO, Cleide Cielia Soares; FAUSTO, Ilma Rodrigues de Souza. **Automação e empregabilidade: os efeitos da inteligência artificial no cenário profissional**. In: AGUIAR, Denison Melo de; VERCIANO, Maralice Cunha (org.). Inovação e estratégia na gestão pública e comercial. Ponta Grossa: AYA Editora, 2025. cap. 22. Disponível em: https://ayaeditora.com.br/livros/LF010C22.pdf. Acesso em: 28 mar. 2026.

FERREIRA, Vitor Lany Freitas. **Impacto do código gerado por inteligência artificial na manutenibilidade do software**. 2025. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Software) – PUC Minas, 2025. Disponível em: https://bib.pucminas.br/pergamumweb/download/67ab4daf-b77c-47d1-9769-d1ac6f1355c5.pdf. Acesso em: 28 mar. 2026.

FOWLER, Martin. **Refatoração**: aperfeiçoando o design de códigos existentes. 2. ed. São Paulo: Novatec, 2020. Disponível em: https://www.amazon.com.br/Refatoração-Aperfeiçoando-Design-Códigos-Existentes/dp/8575227246. Acesso em: 11 abr. 2026.

GAMA, Júlio César Lima; SOUSA, Reudismam Rolim de; GONÇALVES, Samara Martins Nascimento. A inteligência artificial para o ensino-aprendizagem do cálculo diferencial e integral na educação superior. **Revista Científica Multidisciplinar**, [S. l.], v. 6, n. 12, p. e6127058, 2025. DOI: 10.47820/recima21.v6i12.7058. Disponível em: https://recima21.com.br/recima21/article/view/7058. Acesso em: 11 abr. 2026.

GONÇALVES, Leticia Vieira; SOUZA, Anderson Mateus de; GONÇALVES, Samara Martins Nascimento; SOUSA, Reudismam Rolim de; SILVA, Verônica Maria Lima. Ferramentas de inteligência artificial generativa na educação: um estudo comparativo. **Revista Científica**

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Multidisciplinar, [S. l.], v. 7, n. 2, p. e727247, 2026. DOI: 10.47820/recima21.v7i2.7247. Disponível em: https://recima21.com.br/recima21/article/view/7247. Acesso em: 11 abr. 2026.

HEGGLER, J. M.; SZMOSKI, R. M.; MIQUELIN, A. F. As dualidades entre o uso da inteligência artificial na educação e os riscos de vieses algorítmicos. **Educação & Sociedade**, v. 46, 2025. Disponível em: https://www.scielo.br/j/es/a/qrTryFvZR9Y9WsRpG5fWGHB/. Acesso em: 28 mar. 2026.

LIMA, L. A. de O.; PARENTE, A. M. B.; LEIMANN, A. H.; SCHIAVAO, L. J. V.; MAIA, L. A.; SOUSA, F. B. da S.; ARAÚJO, T. S. S.; SOUSA, G. C. Gestão na Saúde Pública: Contribuições da Inteligência Artificial para a Otimização dos Processos e Tomada de Decisões. **Revista de Gestão e Secretariado**, [S. l.], v. 16, n. 9, p. e5243, 2025. DOI: 10.7769/gesec.v16i9.5243. Disponível em: https://ojs.revistagesec.org.br/secretariado/article/view/5243. Acesso em: 11 abr. 2026.

LUDERMIR, Teresa Bernarda. Inteligência Artificial e Aprendizado de Máquina: estado atual e tendências. **Estudos Avançados**, v. 35, n. 101, 2021. Disponível em: https://www.scielo.br/j/ea/a/wXBdv8yHBV9xHz8qG5RCgZd. Acesso em: 22 jan. 2026.

MATSUDA, Keyni Borges**. Concorrência nos mercados de infraestrutura de nuvem e de ia generativa**. 2024. Dissertação (Mestrado em Economia) – Universidade Católica de Brasília, 2024. Disponível em: https://bdtd.ucb.br:8443/jspui/bitstream/tede/3655/2/KeyniMatsudaDissertacao2024.pdf. Acesso em: 15 fev. 2026.

MONTEIRO, Sílvia. Inteligência Artificial Generativa e (re)configuração do conhecimento científico: Entre o potencial tecnológico e a literacia crítica. **Revista Portuguesa de Educação**, v. 38, n. 1, 2025. DOI: 10.21814/rpe.41982. Disponível em: https://revistas.rcaap.pt/rpe/article/download/41982/28778. Acesso em: 15 fev. 2026.

NAKAMITI, Eduardo Kiochi. **Agentes inteligentes artificiais**. 2009. Dissertação (Mestrado em Comunicação) – Pontifícia Universidade Católica de São Paulo, São Paulo, 2009. Disponível em: https://repositorio.pucsp.br/handle/handle/5240. Acesso em: 15 fev. 2026.

PINHEIRO, José Henrique Salles. Inteligência artificial generativa na transformação dos processos de desenvolvimento de software: oportunidades, desafios e impactos na produtividade. **Aurum Revista Multidisciplinar**, [S. l.], v. 1, n. 5, p. 201–219, 2025. DOI: 10.63330/armv1n5-016. Disponível em: https://aurumpublicacoes.com/index.php/MA/article/view/222. Acesso em: 11 abr. 2026.

PINHEIRO, José Henrique Salles. Inteligência artificial explicável (xai) na detecção de fraudes financeiras: desafios e oportunidades para aumentar a transparência em sistemas bancários. **Aurum Revista Multidisciplinar**, [S. l.], v. 1, n. 5, p. 166–183, 2025. DOI: 10.63330/armv1n5-014. Disponível em: https://aurumpublicacoes.com/index.php/MA/article/view/220. Acesso em: 11 abr. 2026.

PEREIRA, Luís Moniz. A incidência da inteligência artificial na lógica. **Disputatio**, v. 10, n. 8, 2001. Disponível em: https://sciendo.com/2/v2/download/article/10.2478/disp-2001-0003.pdf. Acesso em: 14 fev. 2026.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

RIBEIRO, Gabriel; SANTOS, Paulo César dos. Inteligência artificial na geração automatizada de testes de software: uma revisão bibliográfica. *In*: Jornada Científica e Tecnológica do IFSULDEMINAS, 17., 2025. **Anais** [...]. IFSULDEMINAS, 2025. Disponível em: https://josif.ifsuldeminas.edu.br/ojs/index.php/anais/article/view/2748. Acesso em: 11 abr. 2026.

SILVA, Matheus Afonso Batista da. **Do Eliza ao ChatGPT: história e evolução da inteligência artificial**. 2024. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Pontifícia Universidade Católica de Goiás, Escola Politécnica, Goiânia, 2024. Disponível em: https://repositorio.pucgoias.edu.br/jspui/handle/123456789/7928. Acesso em: 11 abr. 2026.

SOUZA, Maria Eduarda Ferreira; SOUSA, Reudismam Rolim de; GONÇALVES, Samara Martins Nascimento. A Inteligência Artificial No Ensino Superior Na Visão Discente: Uma Revisão Sistemática Da Literatura. **Revista Científica Multidisciplinar**, [S. l.], v. 6, n. 7, p. e676611, 2025. DOI: 10.47820/recima21.v6i7.6611. Disponível em: https://recima21.com.br/recima21/article/view/6611. Acesso em: 11 abr. 2026.

TAVARES, Yasmim Coelho. **Impactos das ferramentas de inteligência artificial no ciclo de vida do desenvolvimento de software**: uma revisão narrativa. 2025. Trabalho de Conclusão de Curso (Graduação em Tecnologias da Informação e Comunicação) – Universidade Federal de Santa Catarina, Campus Araranguá, Araranguá, 2025. Disponível em: https://repositorio.ufsc.br/handle/123456789/267136. Acesso em: 11 abr. 2026.

VIANA, Ana Cristina Aguilar. Transformação digital na administração pública. **Revista Eurolatinoamericana de Derecho Administrativo**, v. 8, n. 1, 2021. Disponível em: https://www.redalyc.org/journal/6559/655969720005/655969720005.pdf. Acesso em: 07 jan. 2026.

YU, Abraham Sin Oih; NARDY, André; HIRANO, Heloiza Izumi; OLIVEIRA, Jefferson Freitas Amancio de; RIBEIRO, Nathan de Vasconcelos; GRANDO, Nei. Tomada de decisão nas organizações: o que muda com a inteligência artificial? **Estudos Avançados**, São Paulo, v. 38, n. 111, p. 327-348, 2024. DOI: https://doi.org/10.1590/s0103-4014.202438111.017. Disponível em: https://www.scielo.br/j/ea/a/wLBpkMX6WNpcfcCbyddJf9P/. Acesso em: 11 abr. 2026.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.