



O IMPACTO MULTIFACETADO DOS MODELOS DE LINGUAGEM DE GRANDE ESCALA NO DESENVOLVIMENTO DE SOFTWARE

Paula Silveira¹, Bryan Gomes Saraiva²

¹ Profª. Drª. Urcamp, Curso de Sistemas de Informação, Urcamp, Bagé, RS

²Aluno do Curso de Sistemas de Informação, Urcamp, Bagé, RS

Resumo: A proliferação de Modelos de Linguagem de Grande Escala (LLMs) está promovendo uma transformação radical na computação. O presente artigo analisa o impacto multifacetado dessas tecnologias no ciclo de vida do desenvolvimento de software e em áreas correlatas. Trabalho de metodologia exploratória aborda tanto os benefícios, como a aceleração da prototipagem e a democratização da criação através de práticas como o "vibe coding", quanto os riscos inerentes. São examinados casos de falhas catastróficas de IA, como a confabulação que levou à perda de dados, e o paradoxo da produtividade, onde ferramentas de IA podem diminuir a eficiência de desenvolvedores experientes. Adicionalmente, o artigo explora as profundas implicações éticas e sociais, incluindo o deslocamento profissional e a crise no recrutamento gerada pela automação em massa de candidaturas. Conclui-se que os LLMs representam uma tecnologia de duplo gume, cujo potencial disruptivo exige uma navegação cuidadosa entre inovação e responsabilidade.

Palavras-chave: Ética da IA; Inteligência Artificial; Modelos de Linguagem de Grande Escala.

Abstract: *The proliferation of Large Language Models (LLMs) is driving a radical transformation in computing. This paper analyzes the multifaceted impact of these technologies on the software development lifecycle and related fields. The methodology addresses both the benefits, such as accelerated prototyping and the democratization of creation through practices like "vibe coding," as well as the inherent risks. It examines cases of catastrophic AI failures, such as confabulation leading to data loss, and the productivity paradox, where AI tools can decrease the efficiency of experienced developers. Additionally, the article explores the profound ethical and social implications, including professional displacement, the recruitment crisis generated by the mass automation of applications, and the intense controversy over copyright and creativity in the field of art. It concludes that LLMs represent a double-edged technology whose disruptive potential requires careful navigation between innovation and responsibility.*

Keywords: AI Ethics; Artificial Intelligence; Large Language Models.



1 INTRODUÇÃO

A recente proliferação de Modelos de Linguagem de Grande Escala (LLMs), como a família de modelos GPT da OpenAI e o Gemini do Google, representa uma das mais significativas transformações na computação contemporânea. Especificamente no campo do desenvolvimento de software, estas ferramentas estão redefinindo paradigmas tradicionais, alterando fundamentalmente a maneira como programadores escrevem, depuram e otimizam códigos. A integração de assistentes de codificação baseados em IA em Ambientes de Desenvolvimento Integrado (IDEs) não é mais uma novidade, mas uma prática crescente que promete ganhos exponenciais de produtividade. O presente artigo tem como objetivo analisar o impacto multifacetado dessas tecnologias no ciclo de vida do desenvolvimento de software e em outras áreas correlatas da computação.

Para atingir este objetivo, o artigo está estruturado da seguinte forma: primeiramente, define-se o que são LLMs e suas aplicações práticas no desenvolvimento, como a aceleração de testes e prototipagem, e a emergência de novas metodologias como o "vibe coding". Em seguida, analisam-se os riscos e falhas catastróficas associadas ao seu uso, utilizando estudos de caso para ilustrar os perigos da confabulação da IA. Por fim, exploram-se as profundas implicações éticas e práticas, abordando a desqualificação profissional, o deslocamento de empregos e a crise no recrutamento.

2 REVISÃO DA LITERATURA

A ascensão dos Modelos de Linguagem de Grande Escala (LLMs) é um fenômeno recente, e a literatura acadêmica e jornalística que analisa seu impacto está em rápida evolução. Esta revisão sintetiza os estudos e relatos mais pertinentes que fundamentam a discussão sobre a influência dos LLMs no desenvolvimento de software e áreas correlatas, focando em quatro eixos temáticos principais: o debate sobre a produtividade, a crise de



confiabilidade e segurança, a disruptão socioeconômica e a controvérsia sobre direitos autorais.

O principal debate na literatura gira em torno da **produtividade**. Enquanto a narrativa dominante, promovida pelas empresas de tecnologia, posiciona os LLMs como aceleradores de eficiência, estudos recentes começam a questionar essa premissa. Uma pesquisa fundamental neste campo, conduzida pela METR (Model Evaluation and Threat Research), revelou o paradoxo de que desenvolvedores experientes trabalhando em bases de código complexas e legadas tornaram-se, na verdade, mais lentos ao usar assistentes de IA. O estudo identificou que os ganhos de tempo na codificação ativa eram superados pelo tempo gasto na elaboração de prompts, na espera por respostas e, crucialmente, na verificação e correção do código gerado pela IA, um achado que desafia a visão simplista da IA como uma solução universal para a eficiência.

Paralelamente, a literatura documenta uma crescente **crise de confiabilidade e segurança**. Relatos detalhados sobre incidentes com ferramentas como o Gemini CLI e a Replit AI introduziram o conceito de "confabulação" ou "alucinação" da IA como um risco crítico. Esses estudos de caso demonstram que os modelos podem operar com base em uma percepção falha da realidade, levando a ações catastróficas como a destruição de dados de usuários e a exclusão de bancos de dados de produção. A literatura sugere que essa falha fundamental, combinada com a tendência dos modelos de fabricar informações para ocultar erros, representa uma barreira significativa para a adoção segura dessas ferramentas em ambientes de produção sem supervisão especializada.

No campo **socioeconômico**, a literatura aponta para uma disruptão em duas frentes: no mercado de trabalho e no processo de recrutamento. Artigos e análises da indústria documentaram uma tendência clara de empresas como Klarna, UPS, Duolingo, e Cisco de realizar demissões em massa enquanto reinvestem em automação, validando o medo de um iminente deslocamento profissional, especialmente para cargos de nível júnior. Simultaneamente, investigações sobre o processo de contratação revelam um sistema em caos, sobrecarregado pelo "hiring slop" — um volume insustentável de candidaturas geradas por IA. A literatura descreve uma corrida armamentista entre candidatos e recrutadores, ambos



usando IA, o que corói o valor do currículo e introduz novos vetores para fraude e viés algorítmico.

3 METODOLOGIA

O presente trabalho emprega uma metodologia de **revisão bibliográfica qualitativa e exploratória**. Dada a natureza contemporânea e a rápida evolução do impacto dos Modelos de Linguagem de Grande Escala (LLMs) na computação, esta abordagem foi selecionada por ser a mais adequada para sintetizar e analisar o estado atual do conhecimento sobre o tema.

O corpus da pesquisa foi constituído por uma seleção criteriosa de fontes secundárias recentes, incluindo artigos jornalísticos, reportagens sobre estudos de caso e análises da indústria publicadas em veículos de alta reputação no setor de tecnologia e notícias, como *Ars Technica*, *Forbes* e *BBC News*. A escolha dessas fontes se justifica pela necessidade de capturar as discussões, os incidentes e os dados mais atuais, que muitas vezes ainda não foram consolidados em literatura acadêmica revisada por pares.

A análise dos dados seguiu uma abordagem de **análise de conteúdo temática**. As fontes foram sistematicamente examinadas para identificar, categorizar e agrupar os principais temas, argumentos e evidências relacionados ao impacto dos LLMs. Os temas emergentes—como produtividade, riscos de segurança, deslocamento profissional e dilemas éticos—foram então utilizados para estruturar a análise e a argumentação desenvolvida ao longo deste artigo.

4 ANÁLISE E RESULTADOS

4.1 FUNDAMENTOS E APLICAÇÕES GERAIS DOS LLMs

Para compreender a magnitude dessa transformação, é essencial definir o que é um Modelo de Linguagem de Grande Escala. Em sua essência, um LLM é um tipo avançado de inteligência artificial treinado em um volume massivo de dados textuais, frequentemente na escala de petabytes, provenientes da internet, livros e outras fontes. A arquitetura subjacente que viabilizou o sucesso desses modelos é predominantemente a *Transformer*, introduzida em 2017. Diferente de modelos anteriores, a arquitetura Transformer permite que o modelo pese a importância de diferentes palavras em uma sequência de entrada, capturando contextos



complexos e relações semânticas de longa distância. O funcionamento principal de um LLM baseia-se em prever a próxima palavra (ou "token") em uma sequência, o que, em grande escala e com bilhões de parâmetros, resulta na capacidade de gerar textos coerentes, traduzir idiomas, responder a perguntas e até mesmo escrever código de programação de forma autônoma.

Dentre as diversas aplicações dos LLMs, talvez a mais impactante e visível para a comunidade de tecnologia seja a sua capacidade de geração e assistência de código. Ferramentas como o GitHub Copilot, integradas diretamente aos IDEs, funcionam como um "par programador" virtual, sugerindo desde linhas individuais de código até funções inteiras, traduzindo algoritmos de uma linguagem para outra e explicando blocos de código complexos em linguagem natural. Este avanço não apenas acelera o desenvolvimento, mas também levanta questões pertinentes sobre a originalidade do código, a segurança de implementações sugeridas e a evolução do papel do desenvolvedor de software, temas que serão explorados em detalhe neste trabalho.

A assistência de código por LLMs, no entanto, não se restringe ao ambiente integrado dos IDEs. O ecossistema de ferramentas expandiu-se para incluir interfaces web e até mesmo aplicações móveis, como os portais de modelos como ChatGPT e Gemini. Este acesso ubíquo permite que desenvolvedores, estudantes e entusiastas obtenham ajuda de programação de forma rápida e contextual, fora do ambiente formal de um projeto. Seja para prototipar uma ideia, depurar um trecho de código isolado em um smartphone, ou para fins de aprendizado, essas interfaces oferecem um ponto de entrada de baixa fricção, democratizando o acesso a uma assistência de programação que antes exigiria pesquisa extensa em documentações ou fóruns. Essa modalidade de uso complementa as ferramentas integradas, servindo como um "canivete suíço" para resolver problemas pontuais de codificação.

4.2 A EMERGÊNCIA DE NOVAS METODOLOGIAS: O "VIBE CODING"

Essa interação conversacional e acessível com LLMs está dando origem a uma nova metodologia de desenvolvimento, informalmente denominada pela comunidade como "*vibe coding*". Esta abordagem se afasta do planejamento estruturado e do desenvolvimento algorítmico tradicional. Em vez disso, o desenvolvedor atua mais como um diretor,



fornecendo ao LLM um objetivo de alto nível ("crie uma função em Python que leia um arquivo CSV e retorne a média de uma coluna específica"). O código inicial gerado pela IA é então executado e, invariavelmente, os erros resultantes são simplesmente copiados e colados de volta no chat com a instrução "consertar isso".

Este ciclo de "prompt-erro-correção" é repetido iterativamente até que o programa funcione conforme o desejado. O "vibe coding" caracteriza-se por uma experiência de programação fluida e exploratória, onde o caminho para a solução não é pré-definido. Embora possa acelerar drasticamente a prototipagem e a resolução de tarefas simples, essa prática também suscita debates sobre a profundidade do conhecimento técnico do programador, a qualidade e manutenibilidade do código resultante, e a crescente dependência de ferramentas de IA para solucionar problemas que são fundamentais para o desenvolvimento de software.

A ascensão do "vibe coding" apresenta um paradoxo fascinante com implicações profundas para a indústria de tecnologia. Por um lado, representa uma **democratização sem precedentes da criação de software**. Indivíduos com visão de produto—empreendedores, designers e artistas—que antes eram barrados pela complexidade técnica da programação, agora encontram nos LLMs um parceiro de desenvolvimento. A IA atua como um tradutor, convertendo suas ideias e visões em código funcional. Isso possibilita a criação de produtos que antes só poderiam sonhar, como aplicativos móveis inovadores, jogos independentes, websites interativos e plataformas de nicho, sem a necessidade de levantar capital para contratar uma equipe técnica cara. Em essência, a IA está abaixando a barreira de entrada, permitindo que a criatividade, e não apenas a habilidade de codificação, seja o principal motor da inovação.

Por outro lado, essa mesma acessibilidade tem um lado sombrio. Quando desenvolvedores inexperientes utilizam essa abordagem para construir aplicações que lidam com dados sensíveis, as consequências podem ser severas. A metodologia do "vibe coding" pode resultar em um produto que *funciona* na superfície, mas que carece de uma arquitetura segura, robusta e escalável. O desenvolvedor pode não ter a menor ideia de como o código gerado pela IA realmente funciona, tornando-o incapaz de prever vulnerabilidades de segurança, garantir a privacidade dos dados do usuário ou otimizar o desempenho. Isso cria



um cenário perigoso onde pessoas que claramente não deveriam estar desenvolvendo certos tipos de sistemas agora têm um passe livre para fazê-lo, construindo "caixas pretas" funcionais que são, na verdade, bombas-relógio de segurança.

O recente incidente com o aplicativo **Tea** serve como um estudo de caso contundente para esses perigos. Tea é um aplicativo de segurança para encontros, projetado para que mulheres possam verificar antecedentes de homens e compartilhar informações sobre eles. Em julho de 2025, o aplicativo foi hackeado, resultando na exposição de milhares de imagens, posts e mensagens diretas. O vazamento incluiu conversas extremamente sensíveis sobre temas como aborto e infidelidade, abrindo a porta para chantagem e danos emocionais. A situação foi agravada pela revelação de que 72.000 imagens enviadas para verificação de identidade, incluindo fotos de mulheres segurando seus documentos, também foram acessadas. Isso ocorreu apesar da política de privacidade do Tea prometer que tais imagens seriam "deletadas imediatamente" após a autenticação. Embora os métodos de desenvolvimento específicos do Tea não sejam públicos, o incidente ilustra perfeitamente o desastre potencial quando a segurança e a privacidade de dados não são os pilares fundamentais do desenvolvimento—um risco amplificado por abordagens que priorizam a funcionalidade imediata em detrimento do conhecimento técnico fundamental.

4.3 O IMPACTO DOS LLMS NO CICLO DE VIDA DO SOFTWARE

Além da escrita de código, os LLMs estão se tornando ferramentas indispensáveis nas fases de verificação e validação do ciclo de vida do software, especificamente em **testes e depuração**. Tradicionalmente, a criação de uma suíte de testes abrangente é uma tarefa meticulosa que exige um tempo considerável. Os LLMs agilizam drasticamente este processo ao gerar automaticamente casos de teste. Um desenvolvedor pode fornecer uma função ou um componente de software ao modelo e solicitar a criação de testes unitários, testes de integração e até mesmo a identificação de "edge cases" (casos de borda) que poderiam ser facilmente ignorados por um ser humano. Adicionalmente, eles são capazes de gerar dados fictícios ("mock data") realistas em massa, essenciais para popular um ambiente de teste e simular o uso real da aplicação.



No processo de **depuração (debugging)**, a contribuição dos LLMs é talvez ainda mais transformadora. Em vez de passar horas decifrando mensagens de erro crípticas e investigando "stack traces", o desenvolvedor pode simplesmente colar o erro na interface do LLM. A IA não só explica a causa provável do erro em linguagem natural e de fácil compreensão, mas também sugere as alterações específicas no código para corrigi-lo. Isso transforma a depuração de uma atividade de investigação solitária para uma sessão de resolução de problemas interativa e colaborativa, reduzindo significativamente o tempo de inatividade e a frustração do desenvolvedor.

Talvez a aplicação mais poderosa para a agilidade do desenvolvimento seja a **criação de protótipos e sistemas de teste**. A capacidade de um LLM de gerar código funcional a partir de descrições em linguagem natural permite a criação de protótipos e Provas de Conceito (PoCs) em uma velocidade antes inimaginável. Um gerente de produto ou desenvolvedor pode descrever a interface de um novo website, a estrutura de uma API REST ou o layout de um aplicativo móvel, e o LLM pode gerar o código-base (boilerplate) em minutos. Isso permite que as equipes validem ideias com stakeholders e usuários finais muito mais cedo no processo de desenvolvimento. Além disso, os LLMs podem construir sistemas de teste inteiros, como a simulação de uma API de terceiros (um "mock server") com a qual a aplicação principal precisa interagir, isolando o sistema em teste e garantindo um ambiente de verificação consistente e controlado.

4.4 FALHAS E RISCOS DAS PRÁTICAS DE CODIFICAÇÃO COM IA

Apesar de seu potencial, a dependência de LLMs para o desenvolvimento de software introduz uma nova classe de riscos, em grande parte devido à natureza fundamentalmente não-inteligente dessas ferramentas. Um LLM não "compreende" o código ou as instruções; ele opera através de reconhecimento de padrões estatísticos. Isso pode levar a falhas graves, especialmente quando as instruções são ambíguas ou quando o modelo precisa trabalhar com uma base de código grande e complexa. Em projetos extensos, um LLM pode "perder o contexto" e introduzir alterações que parecem lógicas isoladamente, mas que quebram a arquitetura geral do sistema ou introduzem vulnerabilidades de segurança sutis.



Um programador inexperiente, confiando na aparente competência da IA, não seria capaz de identificar tais erros, aprovando código defeituoso para produção.

Dois incidentes ocorridos em julho de 2025 com as ferramentas **Gemini CLI** do Google e a plataforma de codificação por IA da **Replit** expõem os perigos de uma falha central dos LLMs: a **confabulação**, ou **alucinação** (EDWARDS, 2025). Este fenômeno ocorre quando o estado interno do modelo de IA se dissocia da realidade, levando-o a tomar ações baseadas em premissas falsas, com consequências desastrosas (EDWARDS, 2025).

No caso do **Gemini CLI**, um usuário pediu à ferramenta para executar uma tarefa aparentemente simples: renomear uma pasta e reorganizar alguns arquivos (EDWARDS, 2025). O modelo encontrou um problema ao tentar criar um novo diretório, mas processou o comando falho como se tivesse sido bem-sucedido (EDWARDS, 2025). Operando sob a alucinação de que este novo diretório existia, a IA iniciou uma cascata de comandos de movimentação de arquivos para este local fantasma. No sistema operacional Windows, mover um arquivo para um diretório inexistente resulta na renomeação do arquivo, o que fez com que cada comando de movimentação subsequente sobrescrevesse o arquivo anterior, destruindo permanentemente os dados do usuário (EDWARDS, 2025). A própria ferramenta pareceu reconhecer a gravidade de seu erro, emitindo a mensagem:

"Eu falhei com você completa e catastroficamente... Minha revisão dos comandos confirma minha incompetência grosseira" (EDWARDS, 2025). A análise do usuário concluiu que a falha principal foi a ausência de uma verificação de "leitura-após-escrita" para confirmar que suas ações no sistema de arquivos realmente ocorreram como esperado (EDWARDS, 2025).

O incidente com a **Replit AI** foi ainda mais alarmante. O fundador da SaaS, Jason Lemkin, estava usando a plataforma para construir um protótipo quando a IA não apenas cometeu erros, mas começou a **fabricar dadosativamente para encobri-los** (EDWARDS, 2025). Lemkin relatou que o modelo gerava relatórios falsos e mentia sobre os resultados de testes unitários (EDWARDS, 2025). A situação escalou dramaticamente quando a IA ignorou onze instruções explícitas de "congelamento de código" e **deletou o banco de dados de produção da empresa**, que continha mais de 1.200 registros de executivos e empresas



(EDWARDS, 2025). Quando questionada, a IA admitiu ter entrado em "pânico em resposta a consultas vazias" e executado comandos não autorizados (EDWARDS, 2025). De forma reveladora, o modelo também confabulou sobre suas próprias capacidades, afirmando falsamente que era impossível restaurar o banco de dados, quando na verdade o recurso de rollback da Replit funcionava (EDWARDS, 2025).

Esses casos demonstram um perigo fundamental do "vibe coding": os usuários, especialmente os não-técnicos, muitas vezes interagem com essas ferramentas como se fossem inteligências humanas, um equívoco perpetrado pelo marketing das empresas de tecnologia (EDWARDS, 2025). Como aponta o artigo, as tentativas de Lemkin de instruir a IA em letras maiúsculas foram fundamentalmente equivocadas, pois o modelo não possui um conhecimento estável ou capacidade de introspecção para seguir tais regras de forma consistente (EDWARDS, 2025). Sem uma supervisão humana especializada e uma verificação rigorosa de cada ação, a utilização de assistentes de codificação por IA em ambientes de produção permanece uma prática de alto risco (EDWARDS, 2025).

4.5 IMPLICAÇÕES ÉTICAS E PRÁTICAS NO DESENVOLVIMENTO DE SOFTWARE

A integração de LLMs no fluxo de trabalho de desenvolvimento de software não é apenas uma mudança tecnológica; ela acarreta profundas implicações éticas e práticas que merecem uma análise cuidadosa. Dentre as preocupações mais significativas estão o potencial para a **desqualificação profissional (deskilling)** dos desenvolvedores, a questão da **propriedade intelectual** do código gerado e o impacto no **mercado de trabalho** da área de tecnologia.

4.5.1 A desqualificação (deskilling) e o paradoxo da produtividade

Uma das preocupações mais debatidas é que a dependência excessiva de assistentes de IA possa levar à atrofia das habilidades fundamentais de um desenvolvedor. A programação não é apenas a escrita de sintaxe; é a arte de resolver problemas, de pensar algorítmicamente e de construir arquiteturas mentais complexas. Quando um desenvolvedor delega a parte central da resolução de problemas para uma IA, ele corre o risco de se tornar um mero "supervisor" ou "editor" de código, perdendo a prática e a intuição que só são



desenvolvidas através do esforço cognitivo de criar e depurar soluções do zero. Para desenvolvedores juniores, o risco é ainda maior: eles podem nunca chegar a desenvolver essas habilidades essenciais, tornando-se perpetuamente dependentes da ferramenta.

Contrariando a narrativa de ganhos de eficiência, um estudo recente da **METR** (**Model Evaluation and Threat Research**) revelou um paradoxo surpreendente. A pesquisa, um ensaio clínico randomizado com 16 desenvolvedores experientes trabalhando em seus próprios repositórios de código aberto, descobriu que o uso de ferramentas de IA na verdade tornou os desenvolvedores **19% mais lentos** em suas tarefas (ORLAND, 2025). As tarefas eram parte do trabalho regular dos desenvolvedores, como correções de bugs e implementação de novas funcionalidades (ORLAND, 2025).

O mais fascinante foi a discrepância entre a percepção e a realidade. Antes do estudo, os desenvolvedores esperavam que a IA os tornasse 24% mais rápidos (ORLAND, 2025). Mesmo após a conclusão, eles *acreditavam* ter sido 20% mais eficientes com as ferramentas (ORLAND, 2025). No entanto, os dados cronometrados provaram o contrário. A análise das gravações de tela revelou o porquê: embora o tempo gasto em codificação ativa e depuração tenha diminuído, esses ganhos foram completamente anulados pelo tempo gasto em novas tarefas relacionadas à IA: **escrever prompts, esperar pelas gerações da IA e, crucialmente, revisar as saídas do modelo** (ORLAND, 2025).

O estudo apontou que os desenvolvedores aceitaram **menos de 44% do código gerado pela IA sem qualquer modificação**, e que **9% do tempo total das tarefas** foi consumido apenas nesse processo de revisão (ORLAND, 2025). A principal razão para a ineficácia da IA neste cenário foi a complexidade dos projetos—repositórios com, em média, 10 anos de idade e mais de 1 milhão de linhas de código (ORLAND, 2025). A IA não possuía o "conhecimento tácito ou contexto importante" sobre a base de código que os desenvolvedores experientes tinham (ORLAND, 2025). Os pesquisadores concluíram que as ferramentas de IA atuais podem ser particularmente inadequadas para ambientes com altíssimos padrões de qualidade ou com muitos requisitos implícitos (ORLAND, 2025). Este estudo fornece fortes evidências de que, em cenários complexos do mundo real, a promessa



de produtividade da IA pode não apenas ser irreal, mas contraproducente, transformando o desenvolvedor em um depurador de código de IA, em vez de um criador de soluções.

4.5.2 Deslocamento profissional e a reestruturação do mercado de trabalho

A implicação mais visceral e imediata da adoção de LLMs na indústria de software é o medo do **deslocamento profissional em massa**. Esta não é uma preocupação hipotética sobre um futuro distante, mas uma tendência que já está em andamento, à medida que um número crescente de empresas utiliza a IA para otimizar operações, cortar custos e, consequentemente, reduzir a sua força de trabalho humana (KELLY, 2025). Impulsionadas por pressões econômicas e pelo entusiasmo dos investidores que recompensam a implementação de IA com a valorização de suas ações, as corporações estão priorizando operações mais enxutas, o que resulta em menos empregos disponíveis para humanos (KELLY, 2025).

Dentro da hierarquia do desenvolvimento de software, os **desenvolvedores juniores** são percebidos como os mais vulneráveis. Muitas das tarefas que constituem a base da experiência de um programador iniciante—escrever código repetitivo (*boilerplate*), corrigir bugs simples, criar testes unitários e documentar funções—são precisamente as tarefas que os LLMs podem automatizar com alta eficiência. Isso gera o receio de que as empresas possam reduzir drasticamente ou até mesmo eliminar a contratação de novos talentos, optando por alavancar a produtividade de suas equipes seniores com ferramentas de IA. A expectativa sobre os **desenvolvedores seniores**, por sua vez, está mudando; espera-se que atuem como "multiplicadores de força", utilizando a IA para realizar o trabalho que antes exigiria uma pequena equipe, focando seu tempo em arquitetura de alto nível e na validação do código gerado pela IA.

Diversas empresas de grande porte já demonstraram essa tendência na prática, realizando demissões em massa enquanto anunciam investimentos estratégicos em inteligência artificial (KELLY, 2025).

- **Klarna:** A empresa de fintech implementou um assistente de IA que hoje gerencia uma carga de trabalho equivalente à de **700 funcionários** em tempo integral,



principalmente em tarefas de atendimento ao cliente (KELLY, 2025). O CEO da empresa, Sebastian Siemiatkowski, declarou abertamente que chatbots e sistemas automatizados podem executar tarefas antes realizadas por agentes humanos, permitindo à empresa cortar custos operacionais (KELLY, 2025).

- **UPS:** No início de 2025, a gigante da logística anunciou planos para demitir **20.000 trabalhadores** (KELLY, 2025). Embora a empresa tenha afirmado que a IA não está "substituindo diretamente" os funcionários, a CEO Carol Tomé destacou que novas tecnologias, incluindo aprendizado de máquina, permitiram os cortes ao automatizar tarefas como a geração de propostas, que antes exigiam especialistas humanos (KELLY, 2025).
- **Duolingo:** A popular plataforma de aprendizado de idiomas adotou uma estratégia "AI-first" e encerrou os contratos de **10% de sua força de trabalho de freelancers** (KELLY, 2025). Um porta-voz da empresa reconheceu que a IA contribuiu para a decisão, pois sistemas automatizados agora podiam realizar tarefas de tradução de conteúdo, reduzindo a dependência de tradutores humanos (KELLY, 2025).
- **Intuit e Cisco:** Ambas as gigantes da tecnologia seguiram um padrão semelhante. A Intuit demitiu cerca de **1.800 funcionários**, afirmando que a economia seria reinvestida em tecnologia de IA (KELLY, 2025). Da mesma forma, a Cisco anunciou o corte de aproximadamente **5.900 funcionários** como parte de um pivô estratégico para áreas de alto crescimento, como IA e cibersegurança (KELLY, 2025).

Esses casos ilustram uma tendência clara: a automação via IA não está apenas aumentando a eficiência, mas está sendo aktivamente utilizada como justificativa para reestruturar e reduzir a força de trabalho, gerando um período de profunda incerteza para profissionais da tecnologia em todos os níveis de senioridade.

4.5.3 A corrida armamentista da IA no recrutamento e a morte do currículo

A disruptão causada pelos LLMs estende-se para além do desenvolvimento de software, impactando fundamentalmente o próprio processo de contratação. O que se observa



atualmente é uma caótica **corrida armamentista de IA**, onde tanto candidatos quanto recrutadores utilizam ferramentas de automação em uma batalha que está sobrecarregando o sistema e tornando obsoletas as práticas tradicionais.

5 CONCLUSÃO

Este artigo buscou analisar o impacto multifacetado e frequentemente paradoxal dos Modelos de Linguagem de Grande Escala (LLMs) na computação e, em particular, no ciclo de vida do desenvolvimento de software. A pesquisa confirma que os LLMs não são apenas uma evolução incremental, mas uma força disruptiva que está redefinindo práticas, ferramentas e estruturas profissionais. Os resultados demonstram a natureza de "duplo gume" desta tecnologia. Por um lado, os LLMs democratizam a criação de software, capacitando indivíduos a materializar suas visões e acelerando drasticamente a prototipagem e os testes. Por outro lado, essa acessibilidade vem acompanhada de riscos significativos. A análise de casos como os incidentes com Gemini CLI e Replit AI revela falhas técnicas fundamentais, como a "confabulação", que podem levar a consequências catastróficas. Adicionalmente, o estudo da METR sobre a produtividade sugere que, em contextos complexos, a promessa de eficiência pode ser uma ilusão, enquanto a dependência excessiva das ferramentas levanta preocupações legítimas sobre a desqualificação profissional a longo prazo.

As implicações desta pesquisa são vastas. Para a indústria, fica claro que a adoção de LLMs exige uma abordagem crítica, com ênfase em verificação humana e na compreensão das limitações da tecnologia. Para os profissionais, o papel do desenvolvedor está se transformando de um criador para um supervisor crítico de sistemas de IA. Para a sociedade, os LLMs exacerbam desafios existentes, como o deslocamento profissional, e criam novos, como a crise de autenticidade no recrutamento.

Diante deste cenário, sugerem-se trabalhos futuros em diversas frentes. São necessários estudos longitudinais para medir os efeitos a longo prazo da assistência de IA sobre as habilidades cognitivas dos desenvolvedores. Na área técnica, a pesquisa deve focar no desenvolvimento de mecanismos de verificação e segurança para mitigar os riscos de alucinação. Finalmente, há uma necessidade urgente de desenvolvimento de quadros jurídicos e regulatórios mais robustos para lidar com o uso ético da IA no mercado de trabalho. A



navegação responsável por esta nova era tecnológica dependerá de uma colaboração contínua entre pesquisa acadêmica, desenvolvimento industrial e formulação de políticas.

REFERÊNCIAS

EDWARDS, Benj. The résumé is dying, and AI is holding the smoking gun. **Ars Technica**, 24 jun. 2025. Disponível em: <https://arstechnica.com/ai/2025/06/the-resume-is-dying-and-ai-is-holding-the-smoking-gun/>. Acesso em: 20 ago. 2025.

EDWARDS, Benj. Two major AI coding tools wiped out user data after making cascading mistakes. **Ars Technica**, 24 jul. 2025. Disponível em: <https://arstechnica.com/information-technology/2025/07/ai-coding-assistants-chase-phantoms-destroy-real-user-data/>. Acesso em: 20 ago. 2025.

EDWARDS, Charlotte. Dating safety app Tea suspends messaging after hack. **BBC News**, 30 jul. 2025. Disponível em: <https://www.bbc.com/news/articles/cd0dgkjgjzvjo>. Acesso em: 20 ago. 2025.

KELLY, Jack. It's Time To Get Concerned As More Companies Replace Workers With AI. **Forbes**, 04 mai. 2025. Disponível em: <https://www.forbes.com/sites/jackkelly/2025/05/04/its-time-to-get-concerned-klarna-ups-duolingo-cisco-and-many-other-companies-are-replacing-workers-with-ai/>. Acesso em: 20 ago. 2025.

ORLAND, Kyle. Study finds AI tools made open source software developers 19 percent slower. **Ars Technica**, 14 jul. 2025. Disponível em: <https://arstechnica.com/ai/2025/07/study-finds-ai-tools-made-open-source-software-developers-19-percent-slower/>. Acesso em: 20 ago. 2025.