

ANÁLISE COMPARATIVA ENTRE REUSO TRADICIONAL DE CÓDIGO E DESENVOLVIMENTO ASSISTIDO POR INTELIGÊNCIA ARTIFICIAL GENERATIVA: UM EXPERIMENTO CONTROLADO SOBRE PRODUTIVIDADE

Rafael de Miranda Florindo¹
Luciano Junior Bifano Braga²
Túlio Vieira Gomes Santana³

prof.tuliovgomes@gmail.com

ÁREA DO CONHECIMENTO: Ciências Exatas e da Terra

PALAVRAS-CHAVE: inteligência artificial generativa; reuso de código; produtividade; experimento controlado.

1 INTRODUÇÃO

A engenharia de *software* contemporânea exige entregas rápidas e manutenção da qualidade. O reuso de código tradicional é uma estratégia consolidada para produtividade e confiabilidade (Krueger, 1992). Com o avanço dos Modelos de Linguagem de Grande Escala (LLMs), ferramentas de IA generativa, como o *GitHub Copilot*, passaram a auxiliar desenvolvedores, gerando código sinteticamente (Chen *et al.*, 2021). Embora estudos indiquem que assistentes de IA podem reduzir o tempo em tarefas repetitivas (Benetti e Filannino, 2024), há riscos como inconsistências lógicas, vulnerabilidades e aumento da dívida técnica. Benetti e Filannino (2024) observaram que, apesar da percepção de redução de esforço, não há evidências estatisticamente significativas de melhora na corretude do código produzido com IA. A literatura carece de experimentos controlados que comparem diretamente o desempenho de desenvolvedores usando reuso tradicional versus assistência de IA generativa com métricas objetivas. A maioria dos estudos baseia-se em pesquisas de percepção ou análises observacionais sem controle de confundidores (Forsgren *et al.*, 2021). Diante disso, este trabalho busca responder: em tarefas de programação de complexidade controlada, existem diferenças estatisticamente significativas no tempo de conclusão, na proporção de código externo incorporado e na taxa de aprovação em testes entre desenvolvedores que utilizam ferramentas tradicionais de reuso e os que utilizam assistência de IA generativa (*GitHub Copilot*)? A pesquisa justifica-se pela geração de evidências empíricas em ambiente experimental, informando decisões sobre a adoção de ferramentas de IA generativa. O objetivo é analisar comparativamente o desempenho de desenvolvedores em tarefas de programação de complexidade controlada, considerando indicadores objetivos de produtividade.

¹ Acadêmico do 7º período de Ciência da Computação – Centro Universitário Vértice – Univértix – Matipó.

² Acadêmico do 7º período de Ciência da Computação – Centro Universitário Vértice – Univértix – Matipó.

³ Professor do curso de Ciência da Computação - Centro Universitário Vértice – Univértix – Matipó.

2 METODOLOGIA

A pesquisa é descritiva com abordagem quantitativa (Lozada e Nunes, 2019), classificação que visa determinar a natureza das relações entre as variáveis avaliadas por meio de métodos estatísticos, com dados apresentados numericamente. O delineamento adota análise baseada em dados secundários (Wohlin et al., 2012), utilizando o conjunto de dados público de Peng *et al.* (2023). Essa escolha elimina os custos e riscos associados ao recrutamento de participantes e viabiliza a replicação imediata das análises (Lozada; Nunes, 2019). O conjunto de dados registra análises de desempenho de noventa e cinco desenvolvedores profissionais distribuídos em dois grupos (com e sem assistente de IA) em tarefas padronizadas de programação. A variável independente é a abordagem de apoio ao desenvolvimento: Grupo A (reuso tradicional, com acesso irrestrito à internet e documentação) e Grupo B (desenvolvimento assistido por IA com GitHub Copilot ativado). As variáveis dependentes que operacionalizam a produtividade são: tempo de conclusão (TC), em minutos; proporção de código externo interno (PCE), definida como a razão entre linhas de origem não autoral e o total de linhas dos artefatos; e taxa de aprovação em testes (TAT), expressa como a razão entre casos de testes aprovados e o total da suíte, com valores entre 0 e 1. O experimento original de Peng et al. (2023) foi programado entre 15 de maio e 20 de junho de 2022, com desenvolvedores recrutados via Upwork e divididos aleatoriamente nos grupos A e B. Os procedimentos de coleta envolveram o acesso ao repositório público de Peng *et al.* (2023) (<https://arxiv.org/abs/2302.06590>), proteção das variáveis TC e TAT, e obtenção da PCE por análise estática dos artefatos finais de código-fonte, com registro dos dados de acesso em razão da possível atualização de repositórios públicos. Os dados foram organizados em planilhas eletrônicas e examinados à análise estatística descritiva (cálculo de médias, variações percentuais e comparações entre grupos), com resultados apresentados em tabelas comparativas e gráficos, utilizando Microsoft Excel. Quanto aos aspectos éticos, a pesquisa fundamenta-se na utilização exclusiva de dados secundários de natureza pública, e o estudo original foi aprovado pelo Microsoft Research Ethics Review Board.

3 RESULTADOS E DISCUSSÃO

O presente estudo encontra-se em fase de desenvolvimento, com a pesquisa alocada na etapa de levantamento bibliográfico. As principais referências teóricas e metodológicas que fundamentarão a análise já foram sistematizadas para a construção da fundamentação teórica. As etapas subsequentes compreendem a coleta quantitativa dos dados secundários, seguida da análise estatística descritiva, síntese dos achados e confrontamento com as evidências identificadas na literatura. Estima-se a conclusão e formatação final do trabalho em outubro de 2026.

4 CONSIDERAÇÕES FINAIS

O parecer final será apresentado após a finalização da pesquisa, com base na análise dos dados obtidos. Espera-se que este estudo contribua para a identificação das principais tendências de transformação no mercado de trabalho de TI no Brasil, evidenciando como a adoção da IA influencia a produtividade e a redefinição de perfis

profissionais. Além disso, a pesquisa pretende fornecer um diagnóstico atualizado sobre o deslocamento de tarefas e a demanda por novas competências técnicas e comportamentais, servindo como base para futuras discussões acadêmicas e estratégicas sobre o desenvolvimento da carreira em tecnologia frente ao avanço da automação inteligente no cenário nacional.

REFERÊNCIAS

BENETTI, A.; FILANNINO, M. GitHub Copilot: a systematic study. *In: ITAL-IA THEMATIC WORKSHOPS*, 4., 2024, Nápoles. **Anais [...]**. Aachen: CEUR-WS, 2024. p. 6-11.

CHEN, M. *et al.* **Evaluating large language models trained on code**. Ithaca: arXiv, 2021. Disponível em: <https://arxiv.org/abs/2107.03374>. Acesso em: 6 maio 2026.

CUI, K. Z. *et al.* **The effects of generative AI on high-skilled work**: evidence from three field experiments with software developers. *Management Science*, Catonsville, 2026. Disponível em: <https://doi.org/10.1287/mnsc.2025.00535>. Acesso em: 13 maio 2026.

FORSGREN, N. *et al.* **The SPACE of developer productivity**. Queue, New York, v. 19, n. 1, p. 20-48, mar. 2021. Disponível em: <https://queue.acm.org/detail.cfm?id=3454124>. Acesso em: 6 maio 2026.

GAO, R. *et al.* **A survey of bugs in AI-generated code**. Ithaca: arXiv, 2025. Disponível em: <https://arxiv.org/abs/2512.05239>. Acesso em: 6 maio 2026.

HE, H. *et al.* **Does AI-assisted coding deliver?** A difference-in-differences study of Cursor's impact on software projects. Ithaca: arXiv, 2025. Disponível em: <https://arxiv.org/abs/2511.04427>. Acesso em: 6 maio 2026.

KALLIAMVAKOU, E. *et al.* **An in-depth study of the promises and perils of mining GitHub**. *Empirical Software Engineering*, Dordrecht, v. 21, n. 5, p. 2035-2071, 2016.

KRUEGER, C. W. Software reuse. **ACM Computing Surveys**, New York, v. 24, n. 2, p. 131-183, jun. 1992.

LOZADA, G.; NUNES, J. S. **Metodologia científica**. Porto Alegre: SAGAH, 2019.

PENG, S. *et al.* **The impact of AI on developer productivity**: evidence from GitHub Copilot. Ithaca: arXiv, 2023. Disponível em: <https://arxiv.org/abs/2302.06590>. Acesso em: 6 maio 2026.

SOMMERVILLE, I. **Engenharia de software**. 10. ed. São Paulo: Pearson, 2019.

VASWANI, A. *et al.* Attention is all you need. *In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, 30., 2017, Long Beach. **Anais** [...]. Red Hook: Curran Associates, 2017. p. 5998-6008.

WOHLIN, C. *et al.* **Experimentation in software engineering**. Berlin: Springer, 2012.