



## O IMPACTO DA INTELIGÊNCIA ARTIFICIAL NO DESENVOLVIMENTO DE SOFTWARE

**Acadêmico(as):**<sup>1</sup>.CAVALCANTE URIAS, Juliano Augusto; GAFFO DA ROCHA, Carlos;  
**Professor(a) Orientador(a):** SILVA, Marcelo Leandro<sup>2</sup>.  
**Professor(a) Coorientador(a):** NUNES DA SILVA, Dieimes<sup>3</sup>.

**RESUMO:** A Inteligência Artificial (IA) está desempenhando um papel crucial na otimização do trabalho de desenvolvedores de software. Ela automatiza tarefas repetitivas, acelera a criação de código com assistência inteligente, identifica anomalias e ameaças de segurança, aprimora a qualidade do software através de testes e depuração mais eficazes, além de melhorar o gerenciamento de projetos. A IA também apoia o desenvolvimento de modelos de aprendizado de máquina. Essas aplicações da IA economizam tempo, reduzem erros e aumentam a eficiência geral, permitindo que os desenvolvedores se concentrem em tarefas criativas e complexas, resultando em um desenvolvimento de software mais eficaz e de maior qualidade.

**Palavras-chave:** Inteligência Artificial, Desenvolvimento de Software, Copilot, IA

### INTRODUÇÃO

No entanto, foi apenas nas últimas décadas que os avanços tecnológicos tornaram viável a aplicação prática da inteligência artificial ( Sigla IA ) no desenvolvimento de software. Ao explorar a trajetória da IA desde o seu início até ao presente, será possível identificar os marcos históricos que moldaram a forma como os desenvolvedores utilizam esta tecnologia e como ela continuará a moldar o futuro

1 Acadêmicos(as) do 4º período, do Curso de Análise e Desenvolvimento de Sistemas, das Faculdades Integradas do Vale do Ivaí – UNIVALE Carlos Eduardo Gaffo da Rocha, Juliano Augusto Cavalcante Urias.

2Professor(a) do Curso de Análise e Desenvolvimento de Sistemas das Faculdades Integradas do Vale do Ivaí – UNIVALE Marcelo Leandro da Silva.

3 Professor(a) do Curso de Análise e Desenvolvimento de Sistemas das Faculdades Integradas do Vale do Ivaí – UNIVALE- Dieimes Nunes da Silva.

--	--	--

do desenvolvimento de software. O estudo visa compreender como métodos de análise de dados podem ser aplicados no contexto da IA, que se tornou uma ferramenta indispensável para muitos desenvolvedores de software. O estudo adota uma abordagem analítica que considera as tendências atuais e marcos históricos que moldaram a forma como os desenvolvedores usam a IA. Ao explorar a história da IA e o seu impacto no desenvolvimento de software, o estudo identifica como esta tecnologia continuará a moldar o futuro do desenvolvimento de software.

### **COMO A TECNOLOGIA CHEGOU ATÉ AQUI**

A história da Inteligência Artificial (IA) é uma jornada longa, desde tentativas antigas de replicar a inteligência humana em máquinas até os avanços modernos em aprendizado de máquina. Embora o termo "IA" seja recente, os esforços relacionados a ela remontam à antiguidade. A evolução da IA reflete a busca persistente da humanidade por criar sistemas capazes de raciocínio, aprendizado e decisões autônomas, desde autômatos mecânicos até tecnologias de redes neurais (RUSSELL; NORVIG, 2004, p. 34). Segundo Russell e Norvig (2004), o marco inicial da inteligência artificial foi estabelecido por Warren McCulloch e Walter Pitts, que desenvolveram uma rede neural com neurônios artificiais baseados na fisiologia cerebral e na teoria de Turing. Apesar disso, Alan Turing é frequentemente considerado o "pai" da inteligência artificial devido ao seu abrangente artigo de 1950, "Computing Machinery and Intelligence", que também introduziu o "Teste de Turing" para distinguir entre operações digital e biológica. Matérias teóricas, foram pouco investidos nesse intervalo de tempo até o primeiro sistema baseados em lógica própria e conhecimento que se trata do DENDRAL. Na década de 1980, a Inteligência Artificial fez progressos significativos com o surgimento de sistemas especialistas comerciais, como o R1 da Digital Equipment Corporation (DEC), e a adoção de sistemas especialistas pela DuPont, gerando economias de milhões de dólares. Os EUA (Sigla para Estados Unidos da América) permeiam o mercado, com a criação da Microelectronics and Computer Technology Corporation (Sigla MCC) para manter sua competitividade na área da IA (RODRIGUES, et al, p. 4). Nos últimos tempos, houve uma revolução no trabalho em Inteligência Artificial, tanto no conteúdo quanto na metodologia. Agora, é mais comum usar as teorias existentes

--	--	--

como bases, em vez de propor teorias inteiramente novas(RUSSELL; NORVIG, 2004, p. 35).

## INTELIGÊNCIA ARTIFICIAL COMO ASSISTENTE DE PROGRAMAÇÃO

O sonho de possuir uma IA assistente de programação permeia o sonho dos programadores por décadas, e graças aos avanços recentes nos modelos largos de linguagem (MLL) (em inglês: Large language model ou LLM) agora este sonho parece ter se tornado realidade (BARKE, Shraddha; 2023). As IA's assistentes de programação, como GitHub Copilot, surgiram como ferramentas com potencial de diminuir a barreira de entrada na programação e aumentar a produtividade dos desenvolvedores (TABACHNYK, Maxim; NIKOLOV, Stoyan; 2022). Um destes modelos, o OpenAI Codex, contém 12 bilhões de parâmetros e foi treinado com 54 milhões de repositórios do GitHub, sendo capaz de resolver entre 30-70% dos problemas de Python a qual foi submetido (CHEN, Mark et al; 2021). Com este número expressivo de assertividade, já era de se espera que não levaria muito tempo para que este modelo saísse dos laboratórios de pesquisa e passasse a ser usado em ferramentas assistentes de programação, e assim o GitHub Copilot o fez (FRIEDMAN, N; 2021).

### GITHUB

### COPILOT

O GitHub Copilot foi lançado em Junho de 2021, e desde então foi integrado em diversos ambientes de desenvolvimento, incluindo Visual Studio Code, JetBrains e NeoVim (FRIEDMAN, N; 2021). O assistente teve rápida aderência, e nos 12 primeiros meses após seu lançamento mais de 1,2 milhão de desenvolvedores fizeram uso da ferramenta (DOHMKE, T; 2022). Além de sua notoriedade entre os desenvolvedores e empresas de software, diversos estudos têm surgido utilizando o GitHub Copilot (PENG, Sida et al; 2023)(CHEN, Mark et al; 2021)(BARK, Shraddha; 2023), e por esses motivos iremos utilizá-lo neste artigo para exemplificar o impacto da inteligência artificial no processo de codificação.

### GANHO DE PRODUTIVIDADE NA CODIFICAÇÃO

Os estudos recentes vêm mostrando que as IA's assistentes de programação,

--	--	--

especificamente o GitHub Copilot, têm um impacto notável no aumento da produtividade no campo da programação. A colaboração entre programadores e assistentes de IA tem demonstrado resultados promissores nesse aspecto (PENG; Sida et al, 2023) (BARKE; Shraddha, 2023). Um exemplo concreto disso é um experimento controlado que envolveu o desenvolvimento de um servidor HTTP em JavaScript. Nesse experimento, os desenvolvedores de software que tiveram acesso ao GitHub Copilot concluíram a tarefa 55,8% mais rapidamente do que o grupo de controle, que não utilizou a assistência da IA.

Essa diferença substancial no tempo necessário para concluir a tarefa ilustra de forma clara o potencial de aumento de produtividade proporcionado por essas ferramentas (PENG; Sida et al, 2023). Além disso, é importante destacar que os efeitos heterogêneos observados neste experimento indicam que a IA pode se adaptar às necessidades individuais dos programadores. Isso significa que a assistência de IA não apenas acelera o processo de desenvolvimento, mas também pode ser personalizada para atender às preferências e estilos de trabalho específicos de cada programador. Isso tem o potencial de otimizar ainda mais a produtividade, tornando o desenvolvimento de software mais eficiente e efetivo (PENG; Sida et al, 2023).

Esse ganho de produtividade é particularmente relevante no contexto da transição de pessoas para carreiras de desenvolvimento de software. A IA pode servir como uma ferramenta poderosa para ajudar as pessoas a adquirirem habilidades de programação mais rapidamente e a se tornarem programadores mais proficientes em um período de tempo mais curto. Isso não apenas beneficia indivíduos que buscam entrar na indústria de desenvolvimento de software, mas também as empresas que podem aproveitar uma força de trabalho mais qualificada e produtiva (BARKE; Shraddha, 2023). Em suma, as ferramentas de assistência por IA, como o GitHub Copilot, estão desempenhando um papel crucial no aumento da produtividade no campo da programação. Elas aceleram o desenvolvimento de software, adaptam-se às necessidades individuais dos programadores e facilitam a transição para carreiras de desenvolvimento de software. Esse aumento de

--	--	--

produtividade representa um avanço significativo que tem o potencial de transformar a forma como a programação é realizada e impulsionar a eficiência em projetos de desenvolvimento de software (PENG; Sida et al, 2023)(BARKE; Shraddha, 2023).

## O CÓDIGO GERADO É CONFIÁVEL?

Um estudo recente conduzido por Perry et al. (2022) investigou os riscos de segurança associados ao uso de assistentes de código de IA por desenvolvedores. O estudo envolveu 47 participantes em programas de segurança relacionados a Python, JavaScript e C.

Eles concluíram que usuários com acesso a assistentes de IA tendem a produzir mais vulnerabilidades de segurança em seus códigos, especialmente em casos de string criptografada e injeção de SQL. Surpreendentemente, os usuários que usam assistentes de IA são mais propensos a acreditar que estão escrevendo código seguro, apesar do aumento das vulnerabilidades associadas ao seu uso. No geral, a maneira como os desenvolvedores interagem com os assistentes de IA, incluindo a confiança na IA e o envolvimento na linguagem e formato dos prompts, afeta a segurança do código resultante (Perry et al., 2022).

## CONSIDERAÇÕES FINAIS

O uso da Inteligência Artificial (IA) no desenvolvimento de software, exemplificado pelo GitHub Copilot, tem se mostrado uma ferramenta indispensável para aumentar o desempenho no desenvolvimento do código. Esta revolucionária abordagem de IA não apenas identifica e previne erros comuns, mas também fornece assistência em tempo real, acelerando o processo de codificação. Isso resulta em um código mais robusto e eficiente, com menos bugs e uma maior qualidade geral do software, porém deve-se considerar que pode haver falhas de segurança no código gerado, pois estudos mostraram que a falsa sensação de segurança pode fazer com que códigos mais vulneráveis sejam produzidos.

## REFERÊNCIAS

GOMES, Dennis dos Santos. Inteligência Artificial: conceitos e aplicações. **Revista Olhar Científico**, v. 1, n. 2, p. 234-246, 2010.

--	--	--

RUSSEL, Stuart; NORVIG, Peter. Inteligência Artificial. 2. Ed. Rio de Janeiro: Campos, 2004.

RODRIGUES, Zenaldo Almeida; PINHEIRO, Luiz. INTELIGÊNCIA ARTIFICIAL (IA) E SUA INFLUÊNCIA NO TRABALHO: UMA PERSPECTIVA GLOBAL.

CHEN, Mark et al. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374, 2021.

TABACHNYK, Maxim; NIKOLOV, Stoyan. ML-enhanced code completion improves developer productivity. 2022.

SARKAR, Advait et al. What is it like to program with artificial intelligence?. arXiv preprint arXiv:2208.06213, 2022.

PENG, Sida et al. The impact of ai on developer productivity: Evidence from github copilot. arXiv preprint arXiv:2302.06590, 2023.

BARKE, Shraddha; JAMES, Michael B.; POLIKARPOVA, Nadia. Grounded copilot: How programmers interact with code-generating models. Proceedings of the ACM on Programming Languages, v. 7, n. OOPSLA1, p. 85-111, 2023.

FRIEDMAN, N. Introducing GitHub Copilot: your AI pair programmer. Disponível em: <<https://github.blog/2021-06-29-introducing-github-copilot-ai-pair-programmer/>>. Acesso em: 23 set. 2023.

DOHMKE, T. GitHub Copilot is generally available to all developers. Disponível em: <<https://github.blog/2022-06-21-github-copilot-is-generally-available-to-all-developers/>>. Acesso em: 23 set. 2023.

PERRY, Neil et al. Do users write more insecure code with AI assistants?. arXiv preprint arXiv:2211.03622, 2022.

Com o uso de assistentes de código de IA, produzem mais vulnerabilidades de segurança em seus códigos, de acordo com um estudo envolvendo 47 participantes em programas de segurança relacionados a Python, JavaScript e C. No entanto, os usuários de assistentes de IA tendem a acreditar que eles estão escrevendo código seguro.

--	--	--