

POTENCIALIDADES E DESAFIOS DO GITHUB COPILOT COMO FERRAMENTA DA INTELIGÊNCIA ARTIFICIAL

Jefferson Sena¹

Universidade Federal do Pará
jeffesena@gmail.com

Adriano Barreto²

Universidade Federal do Pará
adrianosb@ufpa.br

John Barbosa³

Universidade Federal do Pará
jctai@gmail.com

Keyla Alves⁴

Universidade Federal do Pará
keylaalves18@gmail.com

Resumo

O presente artigo investiga as principais potencialidades e desafios do uso do GitHub Copilot como ferramenta da Inteligência Artificial, evidenciando as áreas em que seus benefícios vêm sendo aplicados. Essa proposta foi alcançada, a partir da exploração e análise de 12 estudos empíricos tendo por objeto o GitHub Copilot como ferramenta usada pelos Programadores/Desenvolvedores, cuja fonte foi o repositório do Centro de Ciência da Computação da Universidade de Cornell. Os resultados evidenciaram que o GitHub Copilot é uma ferramenta promissora no aumento da produtividade, na perspectiva de tornar mais fáceis e ágeis as tarefas cotidianas de seus usuários, de modo que sua plataforma hospeda uma significativa massa de usuários, composta por Programadores, Desenvolvedores, Organizações e Repositórios de códigos. O estudo conclui que o GitHub Copilot apresenta potencialidades para: aumentar a Produtividade, Programação de códigos, Processamento de Linguagem Natural; tornar mais eficiente a codificação por meio de novos recursos, a solução de problemas de algoritmo e buggy, sendo que esses benefícios ocorrem na área de Engenharia de Software, sem exceção. Os desafios do uso dessa ferramenta, ficou por conta da Segurança do código gerado e a necessidade de aprimoramento de Processos das atinentes as tarefas dos Programadores/Desenvolvedores. Esses obstáculos diagnosticados podem ser minimizados por meio de ferramentas-estratégicas e disponíveis para corrigir principalmente os erros de codificação, e outros, pelos próprios usuários, em prol do crescimento do GitHub Copilot.

Palavras-chave: Programadores; Desenvolvedores; Ferramenta; Códigos; Produtividade.

¹ Doutorando em Ciência da Computação pelo PPGCC/UFPA. Atualmente é Professor Universitário (Graduação e Pós-Graduação) nas áreas de Gestão, Administração e Tecnologia da Informação; Servidor Público do Estado do Pará.

² Doutorando em Ciência da Computação – PPGCC/UFPA.

³ Mestrado em Ciência da Informação – PPGCI/UFPA.

⁴ Doutoranda em Ciência da Computação – PPGCC/UFPA.



Esta obra está licenciada sob uma licença

Creative Commons Attribution 4.0 International (CC BY-NC-SA 4.0).

P2P & INOVAÇÃO, Rio de Janeiro, v. 10, n. 2, p. 1-21, e- e-7031, jan./jun. 2024.

POTENTIALS AND CHALLENGES OF GITHUB COPILOT AS AN ARTIFICIAL INTELLIGENCE TOOL

Abstract

This article investigates the main potentialities and challenges of using GitHub Copilot as an Artificial Intelligence tool, highlighting the areas in which its benefits have been applied. This proposal was achieved based on the exploration and analysis of 12 empirical studies using GitHub Copilot as a tool used by Programmers/Developers, whose source was the repository of the Computer Science Center at Cornell University. The results showed that GitHub Copilot is a promising tool for increasing productivity, with a view to making the daily tasks of its users easier and more agile, so that its platform hosts a significant mass of users, made up of Programmers, Developers, Organizations and Code repositories. The study concludes that GitHub Copilot has potential to: increase Productivity, Code Programming, Natural Language Processing; make coding more efficient through new resources, the solution of algorithm and buggy problems, and these benefits occur in the area of Software Engineering, without exception. The challenges of using this tool are due to the Security of the generated code and the need to improve processes related to the tasks of Programmers/Developers. These diagnosed obstacles can be minimized through strategic tools available to mainly correct coding errors, and others, by the users themselves, in favor of the growth of GitHub Copilot.

Keywords: Programmers; Developers; Tool; Codes; Productivity.

POTENCIALES Y DESAFÍOS DE GITHUB COPILOT COMO HERRAMIENTA DE INTELIGENCIA ARTIFICIAL

Resumen

Este artículo investiga las principales potencialidades y desafíos del uso de GitHub Copilot como herramienta de Inteligencia Artificial, destacando las áreas en las que se han aplicado sus beneficios. Esta propuesta se logró con base en la exploración y análisis de 12 estudios empíricos utilizando GitHub Copilot como herramienta utilizada por Programadores/Desarrolladores, cuya fuente fue el repositorio del Computer Science Center de la Universidad de Cornell. Los resultados mostraron que GitHub Copilot es una herramienta prometedora para aumentar la productividad, con miras a hacer más fáciles y ágiles las tareas diarias de sus usuarios, de modo que su plataforma albergue una masa importante de usuarios, compuesta por Programadores, Desarrolladores, Organizaciones y Repositorios de código. El estudio concluye que GitHub Copilot tiene potencial para: aumentar la productividad, la programación de códigos y el procesamiento del lenguaje natural; hacer la codificación más eficiente a través de nuevos recursos, la solución de problemas algorítmicos y con errores, y estos beneficios se dan en el área de la Ingeniería de Software, sin excepción. Los desafíos del uso de esta herramienta se deben a la Seguridad del código generado y la necesidad de mejorar los Procesos relacionados con las tareas de los Programadores/Desarrolladores. Estos obstáculos diagnosticados pueden minimizarse a través de herramientas estratégicas disponibles para corregir principalmente errores de codificación, y otras, por parte de los propios usuarios, a favor del crecimiento de GitHub Copilot.

Palabras clave: Programadores; Desarrolladores; Herramienta; Códigos; Productividad.

1 INTRODUÇÃO

No atual contexto global, permeado por avanços tecnológicos, que permite a interligação mundial e aumento da competitividade, as organizações buscam estratégias para agilizar e otimizar processos, melhorar a produtividade e oferecer um atendimento e/ou serviço de excelência e, de preferência, personalizado, visando a maior satisfação dos usuários/clientes do século XXI.

Essa dinâmica planetária, com conexão nos quatro cantos do mundo, é propiciada pela maior abrangência geográfica da rede mundial de computadores, a *internet*, por ser a responsável em configurar a rede global, tendo como plataforma as tecnologias, os recursos computacionais, as máquinas, os dispositivos digitais e similares. Disso, resulta na significativa massa de dados e informações e que, a cada momento, só faz aumentar, requerendo suportes para manipular, gerenciar, transformar e produzir, com base nesses fluxos.

Nessas demandas, tem destaque a Inteligência Artificial, uma tecnologia que permite construir “máquinas inteligentes” capazes de realizar tarefas que normalmente requerem inteligência humana. Essas máquinas aprendem através da experiência, percebem as mudanças e ajustam-se a elas, semelhantes aos humanos, aumentando a automação e agilizando processos nas organizações (Andrade, 2023). A inteligência artificial sistematiza e automatiza tarefas intelectuais e, portanto, é potencialmente relevante para qualquer esfera da atividade intelectual humana, considerada um campo universal (Russell; Norvig, 2013).

A busca por métodos ou dispositivos capazes de simular o raciocínio humano vem sendo o objetivo central da área da Inteligência Artificial, a qual vem transformando diversas áreas do conhecimento humano (Zhang *et al.*, 2022). Nesse sentido, a inteligência artificial se utiliza de vários suportes virtuais para realizar seus objetivos, tendo espaço privilegiado para isso o GitHub Copilot, por ser uma ferramenta de programação inteligente, que utiliza algoritmos de aprendizado de máquina para gerar códigos (símbolos, palavras e instruções para compor as informações ou transmitir mensagens), com base nos comentários e sugestões fornecidos pelo usuário.

Assim, o GitHub Copilot passou ser associado ao aumento da produtividade de desenvolvedores e programadores, na perspectiva de tornar mais fáceis e ágeis as tarefas do cotidiano. Para se ter noção do crescimento dessa ferramenta, como plataforma de hospedagem para projetos de software de código aberto e privado, o GitHub Copilot representa uma significativa base de usuários de mais de 100 milhões de desenvolvedores, mais de 4 milhões de organizações e hospeda mais de 330 milhões de repositórios de códigos (Wermelinger, 2023).

Ainda que os aplicativos de inteligência artificial sejam promissores para aumentar a

produtividade humana, no domínio digital, tanto desenvolvedores, como programadores, enfrentam desafios para aplicar o GitHub Copilot e alcançar maior abrangência, como ferramenta da inteligência artificial.

Embora exista uma literatura crescente, principalmente na língua inglesa, que estuda as percepções das ferramentas de inteligência artificial e que discute o aumento da produtividade de desenvolvedores e programadores, seu uso e implicações para a segurança do código, destaca-se que há carência de pesquisas que tomem como escopo as potencialidades, áreas beneficiadas e desafios do uso do GitHub Copilot, como ferramenta de IA. Acredita-se que investigar essas dimensões sejam de fundamental importância, por impactar na produtividade das organizações, as quais dependem dessa tecnologia para se manter, competir e crescer no mercado em que atuam.

Diante dessa vertente alcançada, inquirir-se em duas reflexões para serem investigadas: Quais as principais potencialidades e desafios do uso do GitHub Copilot como ferramenta da Inteligência Artificial? Em quais áreas, ou situações, essa ferramenta vem sendo mais aplicada?

Face a essa explanação, o presente artigo tem objetivo de investigar as principais potencialidades e desafios do uso do GitHub Copilot como ferramenta da Inteligência Artificial, com recorte nas áreas em que seus benefícios vêm sendo aplicados.

O artigo está estruturado, de modo a abordar e fundamentar sobre dois eixos temáticos: a Inteligência Artificial e o GitHub Copilot e suas perspectivas, porém, antes, precede a metodologia, com a descrição dos procedimentos para alcançar a proposta desse estudo. Por fim, temos os Resultados e Discussões, mais as conclusões e referências.

2 MATERIAL E MÉTODO

A investigação transcorreu por meio da Pesquisa Bibliográfica, tendo como estratégia a exploração de um conjunto de publicações, cujos corpo de autores contribuíram para fundamentar e definir as categorias mais representativas que compõem esse estudo, sobretudo a Inteligência Artificial, GitHub Copilot e categorias similares.

Por definição usual, a Pesquisa Bibliográfica “tem a finalidade de aprimorar e atualizar o conhecimento, através da investigação científica de obras já publicadas e consolidadas” (Prodanov; Freitas, 2013, p. 54). Adotou-se essa metodologia, porque a Pesquisa Bibliográfica permite ao pesquisador o contato direto com o material já escrito sobre o objeto de estudo, a partir do levantamento e exploração de um conjunto de publicações de autores que discutem o GitHub

Copilot.

O arcabouço teórico foi composto por autores (Chatterjee *et al.*, 2023; Res *et al.*, 2024; Koyanagi *et al.*, 2024; Zhou *et al.*, 2024; Majdinasab *et al.*, 2023; Fu *et al.*, 2024; Zhang *et al.*, 2023a; Jaworski *et al.*, 2023; Dakhel *et al.*, 2023; Zhang *et al.*, 2023b; Erhabor *et al.*, 2023; Yetiştirren *et al.*, 2023) somente de artigos científicos, cuja fonte foi o *Site* (<https://arxiv.org/abs/2206.15000v3>) do repositório do Centro de Ciência da Computação da Universidade de Cornell, o qual disponibiliza trabalhos na área de GitHub Copilot.

De um total de 48 artigos, foram obtidos 12 (Doze) publicações, todos escritos no idioma inglês, e do ano de 2023, sendo eles traduzidos para a língua materna para se fazer a devida leitura e análise para responder o propósito da pesquisa. Outros autores também foram chamados para dissertar sobre a inteligência artificial.

O critério de seleção das bibliografias foi que os autores discutissem o GitHub Copilot centrado nas categorias Programação, Desenvolvedor e Códigos, por serem termos que mais representam essa ferramenta de inteligência artificial. As publicações avulsas na *Web* e/ou que não atenderam o rigor científico adotado, foram excluídos da coleta, visando à qualidade e a credibilidade da pesquisa.

O procedimento para a construção textual do corpo do artigo passou pela síntese conceitual das publicações selecionadas, com recorte em partes pontuais, com definição do tema, objetivos, metodologia, principais resultados e conclusões dos estudos requeridos, os quais foram sistematizados qualitativa e quantitativamente, para fins de análises.

Os resultados foram sintetizados em forma de quadro, composto pelas categorias Áreas, Potencialidades/benefícios e Desafios do uso do GitHub Copilot. A exploração desse produto teve a orientação de Minayo (2001), que prescreve três tipos de aferições no objeto bibliográfico: a) descrição do conteúdo do material tabulado; b) leitura para entender a causa e efeitos do uso do Copilot; e, c) análise, que exige *expertise* e ponto de vista crítico do pesquisador atinente ao tema, subsidiando os resultados da pesquisa, com apoio dos autores.

Destarte, as discussões dos resultados refletiram puramente os referenciais teóricos supracitados, cuja fundamentação se manifestou no diálogo entre os autores, visando colher argumentos consistentes para alcançar uma síntese conclusiva para dar respostas ao questionamento levantado, de modo a respaldar o objetivo desse artigo. E, ainda: adverte-se sobre a carência de estudos sobre o GitHub Copilot na língua materna portuguesa disponível na *Web*, o que significou uma das limitações dessa pesquisa.

3 A INTELIGÊNCIA ARTIFICIAL E SEUS FUNDAMENTOS

A inteligência artificial (IA) é um termo criado pelo norte-americano John McCarthy em 1956, professor de matemática da Dartmouth College durante uma conferência realizada nessa instituição de ensino superior renomada mundialmente. Nesse evento, McCarthy mostrou que os computadores, no futuro, teriam aptidão para copiar o pensamento humano e reproduzir algumas de suas habilidades (Oliveira, 2019; Russell; Norvig, 2013).

Esse comunicado científico despertou interesse de pesquisadores do mundo todo, especialmente da área da computação, com vista a adequar essa tecnologia com referência ao objetivo proposto por McCarthy. O primeiro trabalho de notoriedade na IA foi atribuído aos norte-americanos Warren Macculloch e Walter Pitts, em 1943. Esses pesquisadores desenvolveram a IA a partir de três fontes: (a) o conhecimento da fisiologia básica e da função dos neurônios do cérebro; (b) a análise formal da lógica proposicional criada por Russell e Whitehead; e (c) teoria da computação de Turing (Russell; Norvig, 2013).

Desde então, o segmento da IA foi se desenvolvendo de maneira não linear, dadas as dificuldades da época, em elucidar questões de percepção atinente à inteligência humana, o que demandou tempo e elevados investimentos para pesquisas.

De modo generalizado, a IA é vista como a capacidade de realizar qualquer tarefa cognitiva, próxima a que o ser humano faz. É dotada de características que a tornam uma tecnologia inovadora, pois Caetano (2023) afirma que a IA é constituída de habilidades de Capacidade de Aprendizado (para analisar grandes volumes de dados); Capacidade de Raciocínio (para tomar decisões eficientes); Reconhecimento de Padrões (para identificar padrões em dados, imagens, linguagem); e Interação com Seres Humanos (por meio de interfaces de voz, cujo diálogo permite respostas, de forma mais natural e intuitiva).

Essas qualificadoras são corroboradas por Caetano (2023), quando define a IA como um ramo da ciência da computação que se concentra em criar máquinas capazes de realizar tarefas que requerem inteligência humana, como reconhecimento de fala, aprendizagem, tomada de decisão e resolução de problemas. Para tanto, a AI tem como componentes algoritmos, modelos matemáticos e redes neurais, os quais permitem que os computadores analisem grandes quantidades de dados e aprendam a reconhecer padrões, visando tomar decisões com base nos dados.

Nesse sentido, a IA é “o resultado da junção de muitos algoritmos e de sua organização de modo inteligente e, nesse caso, o algoritmo é visto como um conjunto de instruções passo a passo, tão explícitas que até algo com uma ‘mente’ tão literal como um computador pode segui-las” (Polson; Scott, 2020, p. 11).

Dada a possibilidade de a IA desenvolver tarefas, semelhantes as desenvolvidas pelos humanos, é essa tecnologia vem ganhando destaque significativamente, explicado por três aspectos: a) crescimento exponencial na velocidade dos computadores e ampliação da rede da internet; b) crescimento da quantidade de dados disponíveis; e c) a existência de computação em nuvem, que é um facilitador na redução de custos para armazenamento e análise de dados em larga escala por meio de uma infraestrutura de uma organização (Polson; Scott, 2020).

A IA pode ser discutida também pelo conceito de ‘agente inteligente’, cuja acepção está associada as categorias ‘agentes autônomos e sistemas multiagentes’, associadas as “técnicas para serem integradas, de modo mais eficaz e efetivo em um único agente. Também busca observar como um conjunto destes agentes poderia interagir de forma coordenada e cooperativa [...]” (Sichman, 2021, p. 26). Na perspectiva da ciência da computação, agente significa um sistema computacional encapsulado, situado em algum ambiente, capaz de executar ação autônoma e flexível naquele ambiente, a fim de cumprir seus objetivos (Sichman, 2021).

A IA, por definição usual, pode ser considerada um campo da ciência que estuda o desenvolvimento de máquinas e sistemas inteligentes para o fim de realizar, com eficiência e autonomia própria, atividades humanas. Disso decorre que o objetivo da IA seja “criar mecanismos e/ou máquinas tecnológicas que tenham a competência de realizar atividades humanas, como a capacidade de raciocinar, perceber, compreender a linguagem, aprender, ver, ouvir e efetuar atividades das mais diversas efetuadas pelos humanos” (Polson; Scott, 2020, p. 9).

Oliveira (2019) reitera, dizendo que o objetivo central da IA é fazer com que as máquinas solucionem dificuldades ou problemas complexos. Esse autor pondera ainda que seja essa acepção que deva predominar na IA, posto que muitas situações a inteligência humana não conseguem resolver. Ou seja, a ideia é aquilo em que o ser humano não consegue fazer e/ou elucidar, a máquina do computador tenta realizar e, *destarte*, sustenta-se nessa lacuna.

De modo tal, que Domingos (2017) considera que o objetivo da IA é que o computador auxilie os humanos em fazer o melhor, a partir da aprendizagem com a máquina. Por isso, que a literatura até aqui pesquisada, expressa em Andrade (2023) e Oliveira (2019), dissemina o mesmo informe, de que a IA estar estritamente relacionada as categorias *Machine Learning* e *Deep*

Learning.

O *Machine Learning* (aprendizado de máquina) é uma subcategoria da IA que permite mais precisão nas aplicações de software, possibilitando que as máquinas realizem atividades como um ser humano. Disso, resulta que as máquinas de computadores são aprimoradas de maneira a aprender, paulatinamente, conforme as informações que lhes são apresentadas, por meio de diálogo entre máquina-humano (Souza, 2023).

Esse processo é possível, visto que o *Machine Learning* tem como base a criação de algoritmos inteligentes, capazes de ler, compreender e interpretar dados, extraíndo respostas diferentes por meio da realização de análises estatísticas. Para tanto, o *Machine Learning* ocorre sob a supervisão de humanos, que alimentam as máquinas com as devidas informações para alcançar resultados requeridos (Souza, 2023).

O *Deep Learning* (aprendizado profundo), por sua vez, representa um subtipo de *Machine Learning* e tem a finalidade de automatizar a análise preditiva, simulando a forma de aprendizado dos seres humanos por meio de mecanismos não lineares de compreensão de informações e, com isso, possibilita análises mais complexas e subjetivas. À medida que o sistema vai reunindo os dados em uma série de modelos estatísticos hierarquicamente encadeados, a compreensão dos mais diferentes contextos vai se tornando cada vez mais precisa (Andrade, 2023).

Portanto, a IA, como máquina inteligente, vem sendo aplicada em várias áreas, indicando que seus benefícios têm aceitação social para atender diversos fins. Ainda assim, McCarthy (2007) adverte que a associação da mente humana à máquina (IA) do computador é apenas uma situação figurada, pois a mente humana é dotada de riquíssimas e complexas especificidades. Bostrom (2018) diz que a IA pode não expressar a mente humana, pois o objetivo da IA é distinto do objetivo da mente humana, até porque os algoritmos são inerentes a máquina, e não ao ser humano. Argumenta ainda que seja a junção de muitos algoritmos, sua estruturação e organização que definem, de forma inteligente, a IA.

3.1 O GITHUB COPILOT E SUAS PERSPECTIVAS

O GitHub Copilot é uma ferramenta revolucionária desenvolvida em conjunto pelo GitHub e pela OpenAI. De modo geral, ela é considerada como uma renomada plataforma de hospedagem para projetos de software privados e de código aberto, contando com uma considerável base de usuários.

Em junho de 2022, o GitHub lançou o Copilot para clientes individuais e o incluiu no Student Pack gratuito. O Codex, disponível publicamente desde novembro de 2021, é um serviço pago, acessado por meio da criação de um programa que chama a API da Open AI. Esse vasto ecossistema fornece ao GitHub Copilot acesso a uma coleção extensa e diversificada de dados de código-fonte, ou seja, a OpenAI colabora com o GitHub e aproveita a riqueza de dados de código-fonte disponíveis na plataforma. Modelos de aprendizagem profunda, como redes neurais recorrentes (RNNs) ou transformadores, servem como base para o GitHub Copilot (Xu, 2023).

Esses modelos têm a capacidade de aprender as complexidades do código, incluindo sua estrutura sintática e semântica, bem como as práticas comuns de codificação empregadas pelos desenvolvedores. Ao combinar os recursos e a experiência do GitHub e da OpenAI, o GitHub Copilot representa um avanço significativo no campo.

Assim, o GitHub Copilot se torna uma ferramenta inovadora, pelo seu poder de colaboração entre uma plataforma líder de desenvolvimento de software e uma potência de IA oferecendo aos desenvolvedores um nível sem precedentes de geração de código e assistência. Arquitetura codificador-decodificador para um modelo típico de transformador (Xu, 2023). Depois de uma camada de incorporação, a entrada é alimentada em vários blocos de atenção do codificador. A saída do codificador, bem como a saída atual de todo o modelo, é então processada por vários blocos de atenção no decodificador para produzir a saída final

Conforme ainda Xu (2023), esse processo se configura em aspectos significativos, como: na construção do conjunto de dados, que usa grande número de bibliotecas de código-fonte aberto e códigos contribuídos por outros desenvolvedores para treinamento; modelo de linguagem, baseado em aprendizagem profunda, como rede neural recorrente (RNN); compreensão do contexto, por entender os objetivos do desenvolvedor analisando o contexto do código desenvolvido; na geração de código, quando é reinado para a compreensão do contexto, gerando sugestões de código que correspondam ao trecho de código atual; etc.

Essas características contribuem para o funcionamento do GitHub Copilot, como base de tecnologias computacionais e similares, mediante aprendizado de máquina e processamento de linguagem natural para sua aplicação. Enfim, o GitHub Copilot, como ferramenta de IA, é capaz de gerar trechos de código de forma inteligente e automatizar tarefas de codificação repetitivas. Esse recurso avançado demonstra o potencial do aprendizado de máquina para aprimorar os processos de desenvolvimento de código e simplificar os fluxos de trabalho de desenvolvimento de software, com impactos na produtividade.

Aqui reforçamos que o GitHub Copilot significa um importante suporte para os desenvolvedores, por meio da escrita de código, sugerindo conclusões para nomes de variáveis e funções, bem como descrições e exemplos de uso de determinadas funções. Ele também pode fornecer sugestões para refatoração de código e pode ajudar na correção de erros e na identificação de possíveis bugs. Outro benefício é que essa ferramenta é fonte de aumento de produtividade, na medida em que os desenvolvedores gastam menos tempo digitando e mais tempo concentrando-se na lógica e na estrutura de seu código (Matos, 2023).

Dentre as limitações do GitHub Copilot, como ferramenta de IA, é que ela pode se tornar muito dependentes de suas sugestões, o que pode levar à falta de compreensão do código subjacente. Além disso, pode levar a erros no código difíceis de identificar e corrigir se a ferramenta não estiver configurada corretamente ou se as sugestões fornecidas por ela não forem precisas. Outro risco é que a ferramenta pode perpetuar práticas de codificação ruins, caso sejam confiadas demais, porém sem entender o código subjacente e correspondente (Chatterjee *et al.*, 2023).

O GitHub Copilot é uma ferramenta que enfrenta algumas situações indesejadas socialmente. Ela pode levar à substituição de empregos pelas máquinas, liberando assim trabalhadores; o seu uso pode influenciar a privacidade e a segurança de seus usuários; a massiva armazenada traz preocupações sobre o uso indevido dessas informações e a possibilidade de violações de privacidade dos indivíduos, assim como a possibilidade de disseminação de informações falsas, com impacto na credibilidade da mídia e das instituições. Outra questão, é a discriminação presentes nos algoritmos, pois os sistemas são alimentados com dados históricos que reproduzem preconceitos e estereótipos da sociedade, afetando grupos minoritários.

No demais, sintetiza-se que o GitHub Copilot é uma ferramenta de IA dotada de potencial para alterar a forma como os desenvolvedores escrevem códigos. Essa inovação se manifesta no aumento da produtividade, com aplicação em várias áreas do conhecimento. Significa uma ferramenta versátil onde os programadores podem aplicar em várias etapas do desenvolvimento de software, desde a escrita inicial do código até a manutenção e refatoração, cuja produtividade e eficiência podem beneficiar as organizações e usuários.

4 RESULTADOS E DISCUSSÃO

4.1 POSSIBILIDADES DO USO DO GITHUB COPILOT

Nesse bloco se incursiona nas doze publicações de Estudos empíricos, na perspectiva de conceber as principais potencialidades e desafios do uso do GitHub Copilot como ferramenta da Inteligência Artificial – IA, com foco nas áreas em que seus benefícios vêm sendo mais aplicados.

Chatterjee *et al.* (2023) investigaram o aprimoramento da segurança da síntese de código baseada na IA com o GitHub Copilot, por meio de engenharia de prompts barata e eficiente. O objetivo desse estudo foi detalhar um experimento com o uso do GitHub Copilot dentro de um ambiente controlado para avaliar sua eficácia em tarefas de engenharia do mundo real, com observância na produtividade após a adoção dessa ferramenta de IA.

A partir do experimento realizado no ANZ Bank australiano com mil engenheiros, os resultados da pesquisa de Chatterjee *et al.* (2023) mostraram aumento notável na produtividade e qualidade do código com o uso do GitHub Copilot, embora seu impacto na segurança do código tenha permanecido inconclusivo. As respostas dos participantes foram globalmente positivas, confirmando a eficácia do GitHub Copilot em ambientes de engenharia de software de grande escala. As respostas dos 1000 engenheiros acenaram também para um aumento significativo na produtividade e satisfação no trabalho.

Res *et al.* (2024) pesquisaram o impacto da Ferramenta de IA do Copilot do GitHub na Engenharia do Banco ANZ australiano, um ambiente Corporativo. Esses autores procederam, primeiro, de uma revisão do atual estado da arte para identificar áreas para melhoria nesta questão. Em seguida, desenvolveram uma abordagem sistemática baseada em métodos de alteração imediata para alcançar uma melhor segurança de código de geradores de código baseados em IA, com enfoque no GitHub Copilot, com vista a minimizar a complexidade desse aplicativo, do ponto de vista do usuário e dos custos operacionais.

Os resultados da pesquisa de Res *et al.* (2024) indicaram que o GitHub Copilot, com suporte do projeto Open VPN, que os métodos propostos reduzem o número de amostras de código geradas inseguras em até 16% e aumentam o número de código seguro em até 8%, sugerindo que a IA, de modo geral, pode ser aplicada a qualquer sintetizador de código baseado em IA, e não apenas ao GitHub Copilot. Ainda que os assistentes de IA para codificação estejam em ascensão, no entanto os desenvolvedores e empresas não aproveitam todo o potencial do GitHub Copilot, dada a

segurança questionável do código gerado.

Koyanagi *et al.* (2024) buscaram explorar o efeito de várias linguagens naturais na sugestão de código usando o copilot do GitHub. Desde sua instituição, em 2021, estudos vêm sendo realizados, no sentido de avaliar as capacidades do Copilot em várias tarefas de programação, bem como seus problemas de segurança. No entanto, pouco se sabe sobre o efeito de diferentes linguagens naturais na sugestão de código. A linguagem natural é considerada um viés social no campo do Processamento da Linguagem Natural - PNL, com impactos na diversidade da engenharia de software.

A partir da aplicação de questionário, composto de 756 questões de diferentes níveis de dificuldade dos concursos AtCoder para fins de avaliação, os resultados da pesquisa de Koyanagi *et al.* (2024) sugerem que a capacidade da IA do Copilot varia entre as línguas naturais, sendo que o chinês alcançou o pior desempenho. Além disso, independentemente do tipo de linguagem natural, o desempenho diminui significativamente à medida que a dificuldade das questões aumenta, o que significa um desafio no PNL e um espaço de oportunidades futuras para investigar essa questão.

Majdinasab *et al.* (2023) avaliaram a segurança do código gerado pelo Copilot do GitHub, configurado como um estudo de replicação direcionado. Por definição geral, os modelos de geração de código alimentados por IA têm se desenvolvido rapidamente, permitindo que os desenvolvedores acelerem a geração de código e, assim, melhorem sua produtividade. Esses modelos são treinados em grandes corporações de código (principalmente provenientes de repositórios públicos), que podem conter bugs e vulnerabilidades.

Colocando em um contexto de discussão a segurança do código gerado pelo Copilot do GitHub, os resultados do estudo de Majdinasab *et al.* (2023) revelaram que, mesmo com as melhorias nas novas versões do Copilot, o percentual de sugestões de código vulnerável reduziu de 36,54% para 27,25%. No entanto, permanece evidente que o modelo ainda sugere código inseguro, sendo que o grande desafio é que os sistemas precisam melhorar seus protocolos de segurança para evitar código inseguro aos desenvolvedores.

Zhou *et al.* (2024) realizaram um estudo sobre as principais preocupações dos desenvolvedores ao usarem o GitHub Copilot, com argumento de que, apesar de sua popularidade, há pouca evidência empírica sobre as experiências reais dos desenvolvedores de software que trabalham com o Copilot. Ou seja, esse estudo tomou como escopo os problemas e desafios que os desenvolvedores enfrentam ao usar o Copilot na sua prática, bem como suas causas subjacentes e possíveis soluções.

Os resultados do estudo de Zhou *et al.* (2024) identificaram que: (1) o Problema de Uso e Problema de Compatibilidade são os problemas mais comuns enfrentados pelos usuários do Copilot, (2) o Problema Interno do Copilot, Problema de Conexão de Rede e Problema de Compatibilidade Editor/IDE são identificados como as causas mais frequentes, e (3) o Bug Corrigido pelo Copilot, Modificar Configuração/Configuração e Versão Adequada de Uso são as soluções predominantes. Os desafios que os usuários encontram, ao implementar o Copilot, são os impactos dessa IA no processo de codificação, requerendo assim o aprimoramento configurado em novos recursos desejados pelos usuários do Copilot.

Fu *et al.* (2024) analisaram os Pontos fracos de segurança do código gerado pelo GitHub Copilot. O argumento é que, embora essa ferramenta de IA esteja se tornando populares e amplamente disponíveis para os desenvolvedores, o uso desses suportes geralmente é acompanhado por desafios de segurança, tomando como objeto de investigação os tipos de problemas de segurança e sua escala em cenários do mundo real.

Os resultados do estudo de Fu *et al.* (2024) mostram que (a) 35,8% dos trechos de código gerados pelo Copilot contêm CWEs, e esses problemas estão espalhados por vários idiomas, (b) as fraquezas de segurança são diversas e relacionadas a 42 CWEs diferentes, nos quais CWE-78: OS Command Injection, CWE-330: Use of Insuficientemente Random Values e CWE-703: Improper Check or Handling of Exceptional Conditions ocorreram com mais frequência, e (c) entre as 42 CWEs identificadas, 11 delas pertencem ao reconhecido CWE 2022 Top-25. A observância é que os desenvolvedores devam ter cuidado ao adicionar código gerado pelo Copilot, assim como devem executar verificações de segurança apropriadas.

Zhang *et al.* (2023a) objetivaram desmistificar e explicar as práticas, desafios e recursos esperados do uso do GitHub Copilot. Esses autores questionam que, embora o GitHub Copilot seja uma ferramenta de IA que tem sido cada vez mais usada desde seu lançamento em junho de 2021, pouco esforço tem sido dedicado para entender as práticas, desafios e características esperadas do uso do Copilot na programação de código-fonte autopreenchido do ponto de vista dos praticantes.

Os resultados da investigação de Zhang *et al.* (2023a) identificaram que, quando os praticantes usam o Copilot: (1) As principais linguagens de programação usadas com o Copilot são JavaScript e Python, (2) o IDE principal usado com o Copilot é o Visual Studio Code, (3) a tecnologia mais comumente usada com o Copilot é Node.js, (4) a principal função implementada é o processamento de dados, (5) o principal objetivo dos usuários é ajudar a gerar código, (6) o benefício central do Copilot é gerar código útil, (7) a principal limitação dos profissionais ao usar o

Copilot é a dificuldade de integração, e (8) a característica mais comum é que o Copilot possa ser integrado a mais IDEs. Enfim, o Copilot requer que os desenvolvedores considerem seus aspectos ao decidirem usar essa ferramenta de IA.

Erhabor *et al.* (2023) empreenderam em mensurar o desempenho de tempo de execução do código produzido com o GitHub Copilot, considerando também o não uso dessa ferramenta de IA. Tomaram como vertente a linha de que muitos estudos foram realizados para avaliar os riscos de segurança do uso do Copilot, mas nenhum deles mostrou se ele ajuda os desenvolvedores a produzir código com melhor desempenho de tempo de execução, justificando a discussão do referido tema.

A partir de um teste com 32 participantes para resolver problemas de programação em C++, com Copilot e com ferramenta para medir o desempenho de tempo, os resultados da pesquisa de Erhabor *et al.* (2023) revelaram que o uso do Copilot pode produzir código com um desempenho de tempo de execução significativamente mais lento, quando comparado sem o uso e aplicação dessa ferramenta.

Yetiştirren *et al.* (2023) avaliaram a qualidade do código de ferramentas de geração de código assistidas por IA tomando como ferramenta para esse estudo empírico sobre o GitHub Copilot, o Amazon CodeWhisperer e o ChatGPT. O fundo dessa pesquisa foi comparar o desempenho dessas proeminentes ferramentas de geração de código em termos de métricas de qualidade de código, tais como Validade de Código, Correção de Código, Segurança de Código, Confiabilidade de Código e Manutenibilidade de Código, para identificar seus pontos fortes e deficiências.

Resultados da investigação de Yetiştirren *et al.* (2023) revelaram que as versões mais recentes do ChatGPT, GitHub Copilot e Amazon CodeWhisperer geram código correto 65,2%, 46,3% e 31,1% das vezes, respectivamente. Em comparação, as versões mais recentes do GitHub Copilot e do Amazon CodeWhisperer mostraram taxas de melhoria de 18% para o GitHub Copilot e 7% para o Amazon CodeWhisperer. A dívida técnica média, considerando cheiros de código, foi de 8,9 minutos para o ChatGPT, 9,1 minutos para o GitHub Copilot e 5,6 minutos para o Amazon CodeWhisperer. Enfim, essa comparação pode auxiliar os profissionais na seleção da ferramenta de IA ideal para tarefas específicas, aprimorando seu processo de tomada de decisão.

Zhang *et al.* (2023b) tomaram como objeto de pesquisa o impacto da IA na produtividade do desenvolvedor, com evidência no uso do copiloto do GitHub. O ponto de partida desse estudo é que as ferramentas de IA generativa têm perspectiva de aumentar a produtividade humana.

A partir de um teste em que os desenvolvedores de software recrutados para implementar

um servidor HTTP em JavaScript, os resultados do estudo de Zhang *et al.* (2023b) mostraram que o grupo de tratamento, com acesso ao programador de pares de IA, completou a tarefa 55,8% mais rápido do que o grupo controle. Os efeitos heterogêneos observados são promissores para programadores de pares de IA para ajudar as pessoas na transição para carreiras de desenvolvimento de software.

Jaworski *et al.* (2023) pesquisaram a experiência de desenvolvedores de software utilizando a ferramenta Github Copilot. O foco desta pesquisa foi investigar a abordagem de desenvolvedores de software para esta ferramenta. Esses autores se apoiaram na premissa de no desenvolvimento de software há uma pressão constante para produzir código cada vez mais rápido sem comprometer a qualidade. Em 29 de junho de 2021, o Github Copilot foi anunciado para atender esse propósito, modelo o qual é treinado para gerar código baseado em linguagem humana compreensível.

Os resultados da pesquisa de Jaworski *et al.* (2023) indicaram que as opiniões dos desenvolvedores estão divididas, pois a maioria deles conheceu o Github Copilot antes de participar da pesquisa. A atitude em relação ao instrumento foi majoritariamente positiva, mas poucos participantes estavam dispostos a usá-lo. As preocupações são com problemas de segurança associados ao uso do Github Copilot, tornando-se obstáculo para o crescimento dessa ferramenta.

Por último, Dakhel *et al.* (2023) investigaram se o Programador de pares de IA Copilot do GitHub é Ativo ou Passivo? Essa proposta tem contundência, na medida em que a síntese automática de programas significa uma meta para a engenharia de software, podendo isso ser feito com o Deep Learning (DL), componente do Copilot. Ainda que muitos estudos avaliem a correção de soluções do Copilot, no entanto esse tema carece de avaliações empíricas para entender como os desenvolvedores podem se beneficiar disso de forma eficaz.

Os resultados da pesquisa de Dakhel *et al.* (2023) mostraram que o Copilot é capaz de fornecer soluções para quase todos os problemas algorítmicos fundamentais, no entanto, algumas soluções são buggy e não reproduzíveis. Além disso, o Copilot apresentou algumas dificuldades em combinar vários métodos para gerar uma solução. Comparando o Copilot com os humanos, indica que a proporção correta de soluções humanas é maior do que as sugestões do Copilot, enquanto as soluções buggy geradas pelo Copilot, requerendo assim menos esforço para serem reparadas.

Diante dos autores supracitados e analisados, apurou-se as seguintes sínteses sobre as Potencialidades, Desafios e as áreas onde os Benefícios do Github Copilot vêm sendo aplicados.

Quadro 1: Áreas, Potencialidades/Benefícios e os Desafios do uso do Github Copilot

Autores	Áreas	Potencialidades/Benefícios	Desafios
Chatterjee <i>et al.</i> (2023)	Engenharia de Software	Aumento da Produtividade e na Qualidade do código	Segurança do código gerado
Res <i>et al.</i> (2023)	Engenharia de Software	Aumento na Produtividade de códigos	Segurança do código gerado
Koyanagi <i>et al.</i> (2023)	Engenharia de Software	Processamento de Linguagem Natural-PNL, menos no idioma chinês	Aprimorar o Processamento PNL
Majdinasab <i>et al.</i> (2023)	Engenharia de Software	Os Github Copilot mais recentes reduzem os códigos vulneráveis	Segurança do código gerado
Zhou <i>et al.</i> (2023)	Engenharia de Software	Programação de Códigos	Aprimoramento do recurso que processa códigos
Fu <i>et al.</i> (2023)	Engenharia de Software	Programação de Códigos	Segurança do código gerado
Zhang <i>et al.</i> (2023a)	Engenharia de Software	Programação de códigos	Segurança do código gerado
Erhabor <i>et al.</i> (2023)	Engenharia de Software	Programação de códigos	Aprimorar o tempo de produção do código
Yetiştiren <i>et al.</i> (2023)	Engenharia de Software	As recentes versões do Github Copilot são mais eficientes na codificação	Segurança do código gerado
Zhang <i>et al.</i> (2023b)	Engenharia de Software	Aumento da produtividade dos Programadores	Segurança do código gerado
Jaworski <i>et al.</i> (2023)	Engenharia de Software	Aumento da produtividade dos Programadores	Segurança do código gerado
Dakhel <i>et al.</i> (2023)	Engenharia de Software	Solução de problemas de algorítmicos e de Buggy	Segurança do código gerado
Resultados			
Potencialidades	%	Desafios	%
Aumento de Produtividade	33,00%	Segurança do código gerado	75%
Programação de códigos	33,00%	Aprimoramento de Processos	25%
PNL	8,34%	---	---
Eficiência dos recentes Copilots	17,32%	---	---
Soluciona de problemas de algoritmo e buggy	8,34%	---	---
Publicações: 12	100%	----	100%

Fonte: Dos Autores.

A leitura desses estudos científicos mostra que o GitHub Copilot vem sendo usado pelos programadores/desenvolvedores para: o aumento de Produtividade (33,00%); a Programação de códigos (33,00%); o Processamento de Linguagem Natural-PNL (8,34%); ser mais eficiente na codificação, por meio do Copilot mais recente (17,32%); e para solucionar problemas de algoritmo e buggy (8,34%).

Esses indicadores sugerem que a IA do GitHub Copilot, quando aplicado, cumpre a sua finalidade de programar, fazendo mais com menos, aumentando a produtividade e eficiência de processos, como efeito da automatização de tarefas repetitivas e trabalhosas. Enfim, acelera processos, melhora a precisão e a qualidade de códigos, contribuindo, assim, para aumentar a capacidade de análise de grande quantidade de dados.

Quanto aos principais desafios enfrentados pelos programadores/desenvolvedores para gerar códigos ou aprimorar Software ou outras tarefas afins, esses são de diversas ordens. Porém, o estudo colocou em evidência os problemas de Segurança do código gerado (75%) e necessidade de aprimoramento de Processos (25%).

Esses dois fatores diagnosticados certamente se traduzem em obstáculos para que programadores/desenvolvedores sejam mais produtivos na codificação, ou em qualquer outra tarefa, assim como também impacta no crescimento e na consolidação do próprio GitHub Copilot, como importante ferramenta de Inteligência artificial, carecendo tratamento especial principalmente a Segurança do código gerado, por ser uma questão mais expressiva.

CONCLUSÃO

O presente artigo investigou as principais potencialidades e desafios do uso do GitHub Copilot como ferramenta da Inteligência Artificial, evidenciando as áreas em que seus benefícios vêm sendo aplicados, a partir da análise de 12 estudos empíricos.

De modo generalizado, a pesquisa mostra que o GitHub Copilot é uma inteligência artificial promissora no aumento da produtividade de desenvolvedores e programadores, na perspectiva de tornar mais fáceis e ágeis as tarefas cotidianas no contexto da ciência da computação. Tão importante é a importância do GitHub Copilot, que sua plataforma de hospedagem representa uma significativa base de usuários, composta de mais de 100 milhões de desenvolvedores, mais de 4 milhões de organizações e hospeda mais de 330 milhões de repositórios de códigos.

Diante da proposta colocada, esse estudo alcançou duas conclusões para o objeto

pesquisado:

a) O estudo revelou que as Potencialidades do uso GitHub Copilot se manifestam, pela ordem de importância, no aumento de Produtividade; na Programação de códigos; no Processamento de Linguagem Natural; no uso mais eficiente na codificação, por meio do Copilot mais recente; e na solução de problemas de algoritmo e buggu. Esses benefícios ocorrem na área de Engenharia de *Software*, sem exceção.

b) A pesquisa revelou que os principais desafios dos programadores/desenvolvedores nas suas tarefas no dia a dia são, pela ordem de importância, a Segurança do código gerado e necessidade de aprimoramento de Processos das atinentes tarefas ao GitHub Copilot.

Para aumentar as potencialidades do GitHub Copilot, sugerimos que os desafios enfrentados pelos programadores/desenvolvedores diagnosticados nesse estudo sejam mitigados, por meio de ferramentas estratégicas e disponíveis para corrigir principalmente os erros de codificação, e outros, pelos próprios usuários.

Nessa observância, a Segurança do código gerado pode aumentar, contribuindo para o crescimento e consolidação do GitHub Copilot, desde que isso seja acompanhado do aperfeiçoamento de seu processo, pois se trata de uma ferramenta recente e que requer melhoramento. Enfim, espera-se que esse estudo tenha contribuído com significativos fundamentos (empíricos) para subsidiar políticas e programas para aprimorar do GitHub Copilot nas tarefas praticadas pelos programadores/desenvolvedores, e sirva também de base para futuras pesquisas.

REFERÊNCIAS

- ANDRADE, R. O. ChatGPT reacende debate sobre o potencial criativo de sistemas de linguagem natural e as implicações éticas relacionadas ao seu uso. *Universo Expandido da Inteligência Artificial. Revista Pesquisa FAPESP*, São Paulo, Edição 325, p. 17-22, mar. 2023. Disponível em: https://revistapesquisa.fapesp.br/wp-content/uploads/2023/03/016-022_capa-chatgpt_325-Parte-1.pdf. Acesso em: 20 abr. 2024.
- BOSTROM, N. **Superinteligência: Caminhos, perigos, estratégias**. Rio de Janeiro: DarkSide Books, 2018. Disponível em: <https://livrariapublica.com.br/livros/superinteligencia-caminhos-perigos-estrategias-nick-bostrom/>. Acesso em: 20 abr. 2024.
- CAETANO, L. M. D. **Inteligência Artificial e ChatGPT: ameaça ou ferramenta na educação superior?** 2023. Disponível em: https://www.researchgate.net/publication/370491136_Inteligencia_Artificial_e_ChatGPT_ameaca_ou_ferramenta_na_educacao_superior. Acesso em: 10 mar. 2024.
- CHATTERJEE, S. *et al.* **The impact of ai tool on engineering at anz bank an emperical study on github copilot within coporate environment**. 2023. Disponível em: <https://arxiv.org/ftp/arxiv/papers/2402/2402.05636.pdf>. Acesso em: 10 mar. 2024.
- DAKHEL, A. M. *et al.* **GitHub Copilot AI pair programmer: Asset or Liability?** arXiv:2206.15331v2 [cs.SE], p. 1-27, 14 Apr 2023. Disponível em: <https://arxiv.org/pdf/2206.15331.pdf>. Acesso em: 10 mar. 2024.
- DOMINGOS, P. **O algoritmo mestre**. São Paulo: Novatec, 2017. Disponível em: <https://pdfshared.com/download/B01N39DSJO?type=pdf&font=livros>. Acesso em: 20 abr. 2024.
- ERHABOR, D. *et al.* **Measuring the Runtime Performance of Code Produced with GitHub Copilot**. ArXiv:2305.06439v1 [cs.SE], p. 1-11, 10 May 2023. Disponível em: <https://arxiv.org/pdf/2305.06439.pdf>. Acesso em: 10 mar. 2024.
- FU, Y. *et al.* Security Weaknesses of Copilot Generated Code in GitHub. *ACM Trans. Softw. Eng. Methodol*, v. 1, n. 1, p. 1-23, Apr. 2024. Disponível em: <https://arxiv.org/pdf/2310.02059>. Acesso em: 27 abr. 2024.
- JAWORSKI, M.; PIOTRKOWSKI, D. **Study of software developers' experience using the Github Copilot Tool in the software development process**. ArXiv:2301.04991v1 [cs.SE], p. 1-5, 12 jan. 2023. Disponível em: <https://arxiv.org/pdf/2301.04991.pdf>. Acesso em: 10 mar. 2024.
- KOYANAGI, K. *et al.* **Exploring the Effect of Multiple Natural Languages on Code Suggestion Using GitHub Copilot**. Lisbon, v. 1, 2 fev. 2024. Disponível em: <https://arxiv.org/pdf/2402.01438.pdf>. Acesso em: 10 mar. 2024.
- MATOS, C. **GitHub Copilot: Como a ferramenta baseada em IA pode melhorar a produtividade do desenvolvedor**. 2023. Disponível em: <https://medium.com/the-code-career/github-copilot-como-a-ferramenta-baseada-em-ia-pode-melhorar-a-produtividade-do-desenvolvedor-a2a762ef37c6>. Acesso em: 10 mar. 2024.

MAJDINASAB, V. *et al.* **Assessing the Security of GitHub Copilot's Generated Code: Targeted Replication Study.** arXiv:2311.11177v1 [cs.SE], p. 1-10, 18 nov. 2023. Disponível em: <https://arxiv.org/pdf/2311.11177v1>. Acesso em: 10 mar. 2024.

MINAYO, M. C. S. (org.). **Pesquisa social: teoria, método e criatividade.** 18 ed. Petrópolis: Vozes, 2001. Disponível em: https://www.faed.udesc.br/arquivos/id_submenu/1428/minayo__2001.pdf. Acesso em: 20 abr. 2024.

MCCARTHY, J. **What is artificial intelligence?** Stanford University, nov. 2007. Disponível em: <http://www-formal.stanford.edu/jmc/whatisai.pdf>. Acesso em: 10 mar. 2024.

OLIVEIRA, A. **Inteligência Artificial.** Lisboa: Fundação Francisco Manuel dos Santos, 2019. Disponível em: <https://ffms.pt/pt-pt/livraria/inteligencia-artificial>. Acesso em: 10 abr. 2024.

POLSON, N.; SCOTT, J. **Inteligência Artificial.** Amadora: Vogais, 2020. Disponível em: <https://www.wook.pt/livro/inteligencia-artificial-nick-polson/23894844>. Acesso em: 10 abr. 2024.

PRODANOV, C. C.; FREITAS, E. C. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico.** Novo Hamburgo, RS: Feevale, 2013. Disponível em: <https://www.feevale.br/Comum/midias/0163c988-1f5d-496f-b118-a6e009a7a2f9/E-book%20Metodologia%20do%20Trabalho%20Cientifico.pdf>. Acesso em: 10 mar. 2024.

RES, J. *et al.* **Enhancing Security of AI-Based Code Synthesis with GitHub Copilot via Cheap and Efficient Prompt-Engineering.** ArXiv:2403.12671v1 [cs.CR], 19 mar. 2024. Disponível em: <https://arxiv.org/pdf/2403.12671.pdf>. Acesso em: 10 abr. 2024.

RUSSEL, S.; NORVIG, P. **Inteligência Artificial.** 3. ed. Rio de Janeiro: Elsevier, 2013. Disponível em: <https://1library.org/document/q7lj7nry-inteligencia-artificial-ed-russell-stuart-norvig-peter-pdf.html>. Acesso em: 10 mar. 2024.

SICHMAN, J. S. Inteligência Artificial e sociedade: avanços e riscos. **Estudos Avançados**, São Paulo, v. 35, n. 101, p. 37-49, dez. 2021. Disponível em: <https://www.revistas.usp.br/eav/article/view/185024/171207>. Acesso em: 12 abr. 2024.

SOUZA, I. **Inteligência artificial nas empresas: como aplicar e qual é o impacto.** 2023. ZapSign. 28 fev. 2023. Disponível em: <https://blog.zapsign.com.br/inteligencia-artificial-empresas/>. Acesso em: 10 mar. 2024.

WERMELINGER, M. Using GitHub Copilot to Solve Simple Programming Problems. *In*: SIMPÓSIO TÉCNICO ACM DE EDUCAÇÃO EM CIÊNCIA DA COMPUTAÇÃO, 54., 2023. **Proceeding** [...]. Toronto: SIGCSE, 2023. p. 172-178. Doi: <https://doi.org/10.1145/3545945.3569830>. Disponível em: https://www.researchgate.net/publication/369248394_Using_GitHub_Copilot_to_Solve_Simple_Programming_Problems. Acesso em: 12 abr. 2024.

XU, H. **GitHub Copilot: uma ferramenta inovadora para preencher automático de códigos.** Universiti Utara Malaysia. dezembro de 2023. Disponível em: https://www.researchgate.net/publication/376406939_Github_Copilot_-_A_Groundbreaking_Code_Autocomplete_Tool. Acesso em: 10 mar. 2024.

YETIŞTIREN, B. *et al.* Evaluating the Code Quality of AI-Assisted Code Generation Tools: An Empirical Study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT. **arXiv:2304.10778v2 [cs.SE]**, p. 1-45, 22 Oct 2023. Disponível em: <https://arxiv.org/pdf/2304.10778.pdf>. Acesso em: 10 mar. 2024.

ZHANG, B. *et al.* Practices and Challenges of Using GitHub Copilot: An Empirical Study. **ArXiv:2303.08733v3 [cs.SE]**, p. 1-6, 27 Apr 2023a. Disponível em: <https://arxiv.org/pdf/2303.08733.pdf>. Acesso em: 28 abr. 2024.

ZHANG, B. *et al.* Demystifying Practices, Challenges and Expected Features of Using GitHub Copilot. **arXiv:2309.05687v1 [cs.SE]**, p. 1-19, 13 Sep 2023b. Disponível em: <https://arxiv.org/pdf/2309.05687.pdf>. Acesso em: 10 mar. 2024.

ZHANG, D. *et al.* **Artificial Intelligence Index Report 2022**. Stanford University, March 2022. 230 p. Disponível em: <https://arxiv.org/pdf/2205.03468>. Acesso em: 10 mar. 2024.

ZHOU, X. *et al.* On the Concerns of Developers When Using GitHub Copilot. **ArXiv:2311.01020v2 [cs.SE]**, p. 1-19, 28 Apr. 2024. Disponível em: <https://arxiv.org/pdf/2311.01020>. Acesso em: 29 abr. 2024.