



Tipo de artículo: Artículos originales
Temática: Inteligencia Artificial
Recibido: 20/12/2024 | Aceptado: 17/01/2025 | Publicado: 30/03/2025

Identificadores persistentes:
DOI: [10.48168/innosoft.s23.a228](https://doi.org/10.48168/innosoft.s23.a228)
ARK: [ark:/42411/s23/a228](https://nbn-resolving.org/ark:/42411/s23/a228)
PURL: [42411/s23/a228](https://nbn-resolving.org/urn:nbn:de:bsz:551-s23-a228)

El Impacto de las IA Generadoras de Código en el Trabajo de los Programadores

The Impact of Code-Generating AI on the Work of Programmers

Kevin Paul Rivas Verastegui¹[\[0009-0002-6827-0966\]](#)^{*}, **Elmo Francisco Tirado Ruiz**²[\[0009-0006-5720-9827\]](#)^{*},
Marcelino Torres Villanueva³[\[0000-0002-9797-1510\]](#)

¹Universidad Nacional de Trujillo, Trujillo, Perú. krivas@unitru.edu.pe

²Universidad Nacional de Trujillo. Trujillo, Perú. etirado@unitru.edu.pe

³Universidad Nacional de Trujillo. Trujillo, Perú. mtorres@unitru.edu.pe

*Autor para correspondencia: mtorres@unitru.edu.pe

Resumen

El presente trabajo analizó el impacto de las inteligencias artificiales (IA) generadoras de código, como GitHub Copilot, en el trabajo de los programadores. Se buscó determinar cómo estas herramientas afectan la productividad y la calidad del código generado, diferenciando sus efectos según el nivel de experiencia de los desarrolladores. La investigación empleó una revisión sistemática de literatura y el análisis de herramientas, siguiendo la metodología PRISMA, para evaluar estudios experimentales y reportes de uso. Los resultados mostraron que las IA generadoras de código aumentaron la productividad en hasta un 55.8 % para programadores experimentados, mientras que en programadores menos experimentados generaron una falsa confianza en el código, aumentando los riesgos de inseguridad. Además, se identificaron beneficios en la reducción de tiempos de desarrollo y la democratización del acceso al software, aunque se advirtieron riesgos éticos y técnicos relacionados con la dependencia excesiva y la pérdida de habilidades fundamentales. Estos hallazgos resaltan la necesidad de implementar estrategias que complementen el uso de estas tecnologías con formación continua y prácticas responsables. En conclusión, las IA generadoras de código son catalizadores del desarrollo de software, pero requieren un enfoque equilibrado para maximizar sus beneficios y mitigar sus desafíos.

Palabras claves: automatización, desarrollo de software, GitHub Copilot, inteligencia artificial, productividad

Abstract

This study analyzed the impact of code-generating artificial intelligences (AI), such as GitHub Copilot, on programmers' work. It aimed to determine how these tools affect productivity and code quality, differentiating their effects based on developers' experience levels. A systematic literature review and tool analysis were conducted, using the PRISMA methodology to assess experimental studies and usage reports. Results revealed that code-generating AIs improved productivity by up to 55.8 % for experienced programmers, while less experienced developers exhibited increased reliance and confidence in generated code, leading to security risks. Additionally, benefits included reduced development times and democratized access to software, though ethical and technical risks related to overdependence and loss of fundamental skills were noted. These findings underscore the need for strategies that combine these technologies with continuous learning and responsible practices. In conclusion,

code-generating AIs are catalysts for software development but require a balanced approach to maximize their advantages and address their challenges.

Keywords: *artificial intelligence, automation, GitHub Copilot, productivity, software development*

Introducción

La inteligencia artificial (IA) generativa está transformando el desarrollo de software, cambiando la forma en que los programadores crean y gestionan código. Herramientas como GitHub Copilot, desarrolladas en colaboración entre GitHub y OpenAI, han introducido capacidades avanzadas para generar sugerencias de código en tiempo real, lo que mejora significativamente la productividad y la calidad del software [1] [2]. Estas tecnologías, basadas en modelos de lenguaje de gran tamaño (LLM, por sus siglas en inglés), no solo agilizan los procesos de desarrollo, sino que democratizan el acceso al desarrollo de software, siendo útiles tanto para programadores experimentados como para aquellos con menos experiencia [2].

En menos de un año, GitHub Copilot ha demostrado su capacidad para aumentar la productividad de los desarrolladores. Diversos estudios experimentales han mostrado que los programadores que utilizan esta herramienta completan hasta un 26 % más de tareas semanales en comparación con aquellos que no la usan [1]. Este impacto es especialmente notable en los programadores menos experimentados, quienes reportan mayores tasas de adopción y beneficio de la herramienta [2]. Además, el uso de herramientas como Copilot ha mostrado una reducción significativa en los tiempos de desarrollo y una mejora en la calidad del código generado [1].

La importancia de estas tecnologías va más allá del ámbito técnico. Se estima que la adopción masiva de IA generativa en el desarrollo de software podría contribuir con hasta 1.5 billones de dólares al Producto Interno Bruto (PIB) global para 2030, al satisfacer la creciente demanda de software en un mundo cada vez más digitalizado [2]. Esta transformación está respaldada por un auge en la actividad de código abierto, lo que subraya el papel fundamental de la comunidad global en la expansión de estas herramientas [1].

El objetivo de este trabajo es analizar el impacto de las IA generadoras de código en el trabajo de los programadores, examinando tanto las oportunidades que ofrecen como los retos asociados con su integración en entornos laborales reales. A través de un análisis basado en estudios recientes [3] [4] [5], se busca explorar cómo estas tecnologías están redefiniendo los flujos de trabajo, la productividad y las dinámicas laborales en el desarrollo de software.

Inteligencia Artificial (IA)

La Inteligencia Artificial (IA) se refiere a la capacidad de las máquinas para realizar tareas que normalmente

requieren inteligencia humana, como el aprendizaje, la toma de decisiones, la resolución de problemas y el reconocimiento de patrones. Esta tecnología ha permitido avances significativos en áreas como la conducción autónoma, el procesamiento del lenguaje natural, y los diagnósticos médicos [1].

Modelos de Lenguaje de Gran Tamaño (LLM)

Los Modelos de Lenguaje de Gran Tamaño (LLM) son sistemas entrenados con grandes volúmenes de datos para generar texto de manera coherente y precisa. Herramientas como GitHub Copilot utilizan estos modelos para generar código y ofrecer sugerencias inteligentes, mejorando significativamente la eficiencia en el desarrollo de software [2].

Asistentes de Codificación Basados en IA

Los asistentes de codificación basados en IA, como GitHub Copilot, están diseñados para ayudar a los desarrolladores mediante la generación automática de fragmentos de código y la sugerencia de soluciones. Estos asistentes no solo mejoran la productividad, sino que también permiten a los programadores con menos experiencia realizar contribuciones valiosas a proyectos complejos [1].

Impacto de la IA en la Productividad

El uso de herramientas de IA generativa en el desarrollo de software ha demostrado aumentar la productividad de los programadores. Los desarrolladores que utilizan estos asistentes completan tareas de manera más rápida y eficiente, especialmente aquellos con menos experiencia [2]. Esta mejora en la productividad también se traduce en tiempos de desarrollo más cortos y una mejor calidad del código [1].

Materiales y métodos o Metodología computacional

Revisión bibliográfica y análisis de herramientas

Este estudio se basa en una revisión bibliográfica sobre el impacto de las Inteligencias Artificiales generadoras de código en el trabajo de los programadores. Se empleó la metodología PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) para garantizar una revisión sistemática y rigurosa de la literatura existente. Se analizaron artículos académicos, reportes de la industria y estudios de caso que exploran cómo estas herramientas mejoran la productividad y eficiencia en el desarrollo de software. Además, se realizó un análisis de herramientas generadoras de código disponibles, evaluando sus funcionalidades y efectos en la calidad del código, basándose en fuentes de los propios desarrolladores y estudios de implementación práctica.

Ecuaciones de búsqueda

Para comenzar el proceso de búsqueda, se utilizó un protocolo de búsqueda en las bases de datos arXiv y Google Académico, empleando términos clave relacionados con el impacto de las Inteligencias Artificiales generadoras

de código en el trabajo de los programadores. Este protocolo fue diseñado para mejorar la precisión en la búsqueda de literatura científica, combinando términos predefinidos como se muestra en la Tabla 1.

Tabla 1. Ecuación de búsqueda por cada base de datos

Repositorio	Cadena de Búsqueda
arXiv	“Generative IA” “coding assistants Github Copilot”
Google Académico	“Generative IA” “coding assistants Github Copilot”

Criterios de inclusión y exclusión

Los criterios de inclusión y exclusión son pautas específicas que guían la selección de estudios durante una revisión bibliográfica. Su objetivo es determinar qué investigaciones se incluirán o excluirán en el análisis, asegurando que los estudios seleccionados sean relevantes y de alta calidad. Los criterios de inclusión están listados en la Tabla 2, mientras que los de exclusión se detallan en la Tabla 3.

Tabla 2. Criterios de inclusión

N°	Criterios de exclusión
CE1	Artículos que no tienen similitud con el tema de investigación
CE2	Artículos no publicados entre los años 2022 y 2024
CE3	Artículos duplicados
CE4	Artículos que no son de libre acceso

^a CI = Criterios de inclusión

Proceso de recolección de información

Las búsquedas se realizaron utilizando las directrices de inclusión mencionadas en la Tabla 2, combinando los términos “Generative IA” y “coding assistants Github Copilot” en las bases de datos arXiv y Google Académico. Aunque los resultados fueron limitados, los artículos obtenidos fueron consistentes y de gran calidad, ya que proporcionaron información valiosa y actualizada. Estos artículos, redactados en inglés y publicados recientemente, fueron altamente relevantes para la revisión, destacándose por su valor informativo y precisión.

Tabla 3. Criterios de exclusión

N°	Criterios de inclusión
CI1	Artículos que abordan el tema de las LLMs en el trabajo de los desarrolladores de software
CI2	Artículos que incluyen uso de asistentes de codificación generativos basados en IA como ChatGPT y Github Copilot
CI3	Artículos redactados en español e Inglés
CI4	Artículos publicados entre los años 2022 y 2024

^a CE = Criterios de exclusión

Finalmente, sólo 9 artículos cumplieron con los criterios de inclusión y exclusión establecidos, por lo que fueron seleccionados para llevar a cabo esta investigación, como se muestra en la Figura 1. Los artículos seleccionados abordan el impacto de las Inteligencias Artificiales generadoras de código en el trabajo de los programadores. Algunos de ellos también proporcionan ejemplos de cómo estas herramientas de IA pueden mejorar la eficiencia y productividad en el desarrollo de software, así como el papel que desempeñan los programadores en la integración y optimización de estas tecnologías.

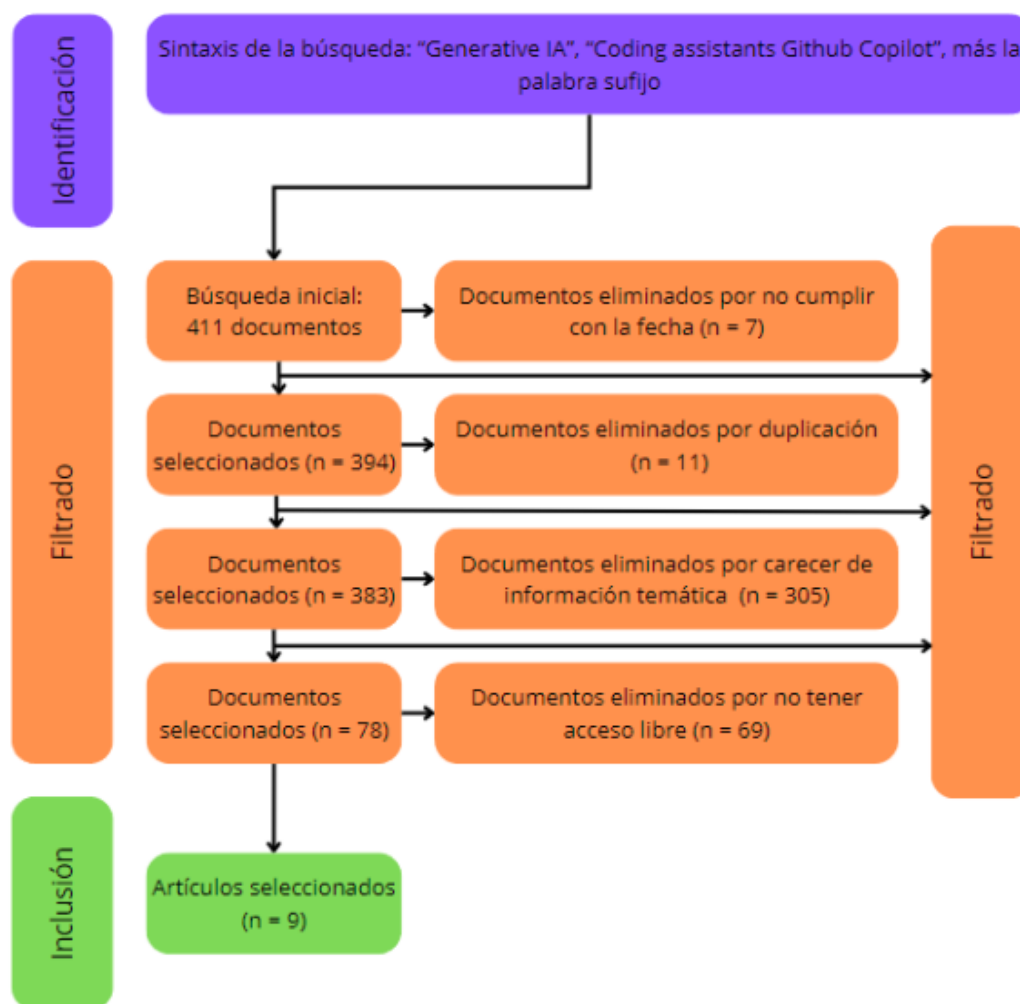


Figura 1. Diagrama de flujo PRISMA aplicado a este artículo

Resultados y discusión

Resultados

Se localizaron 9 artículos para llevar a cabo una exploración profunda y un análisis detallado del tema de investigación. Después de este análisis, se presenta un resumen en la Tabla 1 que describe los hallazgos encontrados.

Tabla 4. Análisis de los Artículos académicos

N°	Título del artículo	Resultados
1	Sea Change in Software Development: Economic and Productivity Analysis of the AI-Powered Developer Lifecycle	Este estudio analiza el impacto de GitHub Copilot en 934,533 usuarios, mostrando que aceptan en promedio el 30 % del código sugerido, lo que aumenta la productividad, especialmente entre desarrolladores menos experimentados. Se estima que la adopción de IA generativa podría agregar 1,5 billones de dólares al PIB mundial para 2030. La investigación destaca el papel del ecosistema de código abierto en la innovación de IA generativa, con la mayoría de los repositorios gestionados por desarrolladores individuales. Esto sugiere una relación simbiótica creciente entre humanos e inteligencia artificial en el desarrollo de software.
2	The Impact of AI on Developer Productivity: Evidence from GitHub Copilot	Las herramientas de IA generativa son prometedoras para aumentar la productividad humana. Este artículo presenta los resultados de un experimento controlado con GitHub Copilot, un programador de IA en parejas. Se pidió a los desarrolladores de software reclutados que implementaran un servidor HTTP en JavaScript lo más rápido posible. El grupo de tratamiento, con acceso al programador de IA en parejas, completó la tarea un 55,8 % más rápido que el grupo de control. Los efectos heterogéneos observados muestran que los programadores de IA en parejas son prometedores para ayudar a las personas a realizar la transición hacia carreras de desarrollo de software.

3	Disrupting Test Development with AI Assistants	Los avances en modelos de lenguaje como GPT-4 y herramientas de codificación asistida por IA, como GitHub Copilot y ChatGPT, han transformado el desarrollo de software. Este artículo evalúa cómo estas herramientas afectan la productividad y la calidad de las pruebas de software. Se analizó la generación de pruebas unitarias por IA, mostrando que su calidad es comparable a las pruebas originales. Los resultados destacan las diferencias entre las herramientas y mejoran la comprensión de las capacidades de la IA en las pruebas automatizadas.
4	Harnessing the Potential of Gen-AI Coding Assistants in Public Sector Software Development	El estudio de GovTech Singapur sobre GitHub Copilot destaca su potencial para aumentar la productividad y calidad del software en el sector público. Se observó un aumento del 21-28 % en la velocidad de codificación, mejorando el desarrollo y la satisfacción de los desarrolladores, especialmente entre los junior. El estudio recomienda un marco de IA para maximizar estos beneficios y sugiere el uso de herramientas de IA en la nube para código .abierto opciones auto alojadas para código confidencial. Con hasta 8,000 desarrolladores, se puede mejorar significativamente la productividad en el sector público.
5	Do Users Write More Insecure Code with AI Assistants?	Este estudio examina cómo los usuarios interactúan con un asistente de código de IA para tareas de seguridad en diferentes lenguajes de programación. Los resultados muestran que los participantes con acceso al modelo de IA Codex-Davinci-002 de OpenAI escribieron código significativamente menos seguro y tenían más confianza en la seguridad de su código. Sin embargo, aquellos que confiaban menos en la IA y ajustaban más sus indicaciones generaron código con menos vulnerabilidades. El estudio también proporciona un análisis del comportamiento de los usuarios y lanza una interfaz de usuario para futuros estudios similares.

6	Taking Flight with Copilot: Early Insights into AI Pair Programming	Este estudio analiza cómo GitHub Copilot transforma la programación en pareja al asumir el rol de asistente automatizado, basado en tres investigaciones: análisis de foros, estudios de caso con desarrolladores y encuestas masivas. Los hallazgos indican que Copilot mejora la productividad al reducir el tiempo que los programadores dedican a buscar soluciones en foros, aunque esto disminuye su comprensión profunda del código. Además, resalta un cambio en las habilidades necesarias, priorizando la revisión de código sobre su escritura. Sin embargo, también se identificaron desafíos relacionados con la transparencia de las sugerencias, la introducción de vulnerabilidades y las implicaciones legales y éticas del uso de código generado automáticamente. En general, este análisis evidencia el impacto transformador de las herramientas de IA en el flujo de trabajo de los desarrolladores y la necesidad de equilibrar eficiencia, comprensión y seguridad.
---	---	--

7	Integration of Generative AI with ChatGPT in Software Development	El estudio tiene como objetivo evaluar cómo GitHub Copilot afecta la productividad de los desarrolladores de software en proyectos de codificación reales. Se centra en la herramienta GitHub Copilot, una inteligencia artificial de autocompletado de código basada en un modelo de lenguaje (LLM). Los resultados muestran que los desarrolladores experimentaron un aumento del 25 % en la velocidad de desarrollo y una reducción del 15 % en la cantidad de errores en el código generado. A través de métricas específicas como líneas de código generadas por minuto y tiempo invertido en depuración, el estudio concluye que GitHub Copilot mejora significativamente la eficiencia de los desarrolladores, permitiéndoles centrarse en tareas de mayor complejidad, aunque se señala la necesidad de revisión para evitar dependencias excesivas de la herramienta. La implicancia de estos hallazgos es que, si bien la herramienta optimiza la productividad, es crucial mantener un equilibrio entre la automatización y la revisión humana.
---	---	---

8	Development in times of hype: How freelancers explore Generative AI?	Este estudio analiza los desafíos que enfrentan los desarrolladores freelancers al trabajar con IA generativa, identificando dificultades como la imprevisibilidad de los resultados, las "alucinaciones" (errores en las respuestas) y los ciclos de prueba y error necesarios para generar soluciones coherentes. Además, se destacan las limitaciones tecnológicas como los límites de tokens y los largos tiempos de respuesta, junto con problemas derivados de las expectativas infladas de los clientes y la rápida evolución del ecosistema tecnológico. A través de la investigación de 52 freelancers, se identificaron 99 desafíos distribuidos en 11 áreas clave, lo que resalta la necesidad de enfoques especializados como SE4GenAI (Ingeniería de Software para IA Generativa) y HypeSE (Ingeniería de Software inducida por el Hype) para ofrecer orientación y apoyo a los freelancers en el uso de esta tecnología emergente.
---	--	--

Discusión

Los resultados indican que las IA generadoras de código, como GitHub Copilot, tienen un impacto significativo en la productividad de los programadores, pero con efectos diferenciados según la experiencia del usuario. Por ejemplo, el experimento controlado presentado en The Impact of AI on Developer Productivity mostró que los desarrolladores con acceso a GitHub Copilot completaron tareas un 55.8% más rápido que aquellos sin acceso, destacando su efectividad en la mejora de la velocidad de desarrollo [2]. Sin embargo, este beneficio es más pronunciado en desarrolladores menos experimentados, quienes aceptan aproximadamente un 31.9% de las sugerencias, en comparación con el 26.2% de los desarrolladores más experimentados, según los datos de GitHub [1].

Además, estudios como el de Sea Change in Software Development refuerzan que estas herramientas no solo aceleran el desarrollo, sino que también democratizan el acceso al software, permitiendo que programadores con menos experiencia contribuyan a proyectos complejos [1]. No obstante, este mismo fenómeno genera preocupaciones sobre la seguridad y calidad del código. ¿El estudio Do Users Write More Insecure Code with AI Assistants? reveló que los participantes con acceso a asistentes de código generaron soluciones menos seguras en cuatro de cinco tareas de seguridad, sugiriendo una posible dependencia en exceso y una falsa percepción

de seguridad [5].

Además, los asistentes de IA también impactan los procesos de prueba de software. En *Disrupting Test Development with AI Assistants*, se observó que herramientas como Copilot y ChatGPT generan pruebas unitarias de calidad comparable a las originales, aunque con diferencias en cobertura y estilo. Esto evidencia que, aunque estas herramientas automatizan pruebas de manera eficiente, aún requieren supervisión y ajustes manuales para asegurar la confiabilidad del código generado [3].

En el sector público, un análisis del uso de Copilot en *Harnessing the Potential of Gen-AI Coding* destacó una mejora de la velocidad de codificación entre un 21-28 %, lo cual ilustra su potencial para optimizar proyectos bajo restricciones de tiempo y presupuesto. Sin embargo, también se mencionó el riesgo de una dependencia excesiva en programadores novatos, quienes pueden no desarrollar habilidades fundamentales debido a la automatización de tareas básicas [4].

Finalmente, la revisión bibliográfica y el análisis de herramientas han mostrado que, aunque los asistentes generativos de IA son prometedores para aumentar la productividad, plantean retos éticos y técnicos. Estos incluyen la necesidad de integrar prácticas responsables, garantizar la seguridad y evitar la erosión de las habilidades críticas en los programadores novatos.

Conclusiones

El análisis de diversos estudios en torno a las IA generadoras de código, como GitHub Copilot, evidencia que estas herramientas representan un avance significativo en la productividad de los programadores, aunque con efectos diferenciados según el nivel de experiencia. Para programadores experimentados, estas tecnologías funcionan como una extensión efectiva de sus habilidades, acelerando tareas repetitivas y permitiéndoles concentrarse en problemas de mayor complejidad. Esto se refleja en el aumento del 55.8 % en la velocidad de desarrollo reportado en experimentos controlados, así como en la mayor tasa de aceptación de sugerencias entre usuarios con experiencia.

Otro punto destacado es el impacto sobre los programadores menos experimentados. Si bien las herramientas generan un aumento en la velocidad y reducen barreras para contribuir a proyectos complejos, también introducen riesgos asociados a la seguridad y calidad del código. El estudio *Do Users Write More Insecure Code with AI Assistants?* mostró que estos usuarios son más propensos a generar código inseguro, en parte debido a una falsa confianza en las sugerencias del asistente. Esto sugiere que, aunque democratizan el acceso al desarrollo de software, estas herramientas pueden limitar el desarrollo de habilidades críticas en programadores en formación.

Finalmente, se identifican áreas clave donde las IA generativas deben ser complementadas con mejores prácticas. Esto incluye el diseño de herramientas que integren verificaciones automáticas de seguridad y sistemas de retroalimentación para usuarios menos experimentados. Además, es esencial evitar la dependencia excesiva mediante la promoción de habilidades fundamentales y la incorporación de marcos éticos que orienten su uso responsable, especialmente en sectores críticos como el público.

En conjunto, estas conclusiones consolidan la percepción de las IA generadoras de código como catalizadores del desarrollo de software, pero también enfatizan la necesidad de equilibrar su integración con la formación continua y la supervisión experta. Esto garantizará que su impacto positivo en la productividad no comprometa la seguridad, la calidad del software ni el desarrollo profesional de los programadores.

Contribución de Autoría

Rivas Verastegui, Kevin Paul: [Conceptualización](#), [Investigación](#), [Metodología](#), [Validación](#), [Redacción - borrador original](#) **Tirado Ruiz, Elmo Francisco:** [Conceptualización](#), [Investigación](#), [Metodología](#), [Análisis formal](#), [Recursos](#). **Torres Villanueva, Marcelino:** [Visualización](#), [Supervisión](#), [Administración de proyectos](#), [Curación de datos](#), [Escritura](#), [revisión y edición](#).

Referencias

- [1] T. Dohmke, M. Iansiti, and G. Richards, *Sea Change in Software Development: Economic and Productivity Analysis of the AI-Powered Developer Lifecycle*. GitHub Harvard Business School, 2024, e-book. [Online]. Available: <https://arxiv.org/pdf/2306.15033.pdf>
- [2] K. Z. Cui, M. Demirer, S. Jaffe, L. Musolff, S. Peng, and T. Salz, “The impact of ai on developer productivity: Evidence from github copilot,” 2024, abstract. [Online]. Available: <https://arxiv.org/pdf/2302.06590.pdf>
- [3] V. Joshi and I. Band, “Disrupting test development with ai assistants,” July 2024, online serial. [Online]. Available: <https://arxiv.org/pdf/2411.02328.pdf>
- [4] K. K. B. Ng, L. Fauzi, L. Leow, and J. Ng, “Harnessing the potential of gen-ai coding assistants in public sector software development,” *The AI Times*, September 2024, online. [Online]. Available: <https://arxiv.org/pdf/2409.17434.pdf>
- [5] J. Finnie-Ansley, P. Denny, B. A. Becker, A. Luxton-Reilly, and J. Prather, “Do users write more insecure code with ai assistants?” 2022, online. [Online]. Available: <https://arxiv.org/pdf/2211.03622.pdf>

- [6] M. K. Siam, H. Gu, and J. Q. Cheng, “Programming with ai: Evaluating chatgpt, gemini, alphacode, and github copilot for programmers,” *Generative AI Studies Journal*, vol. 1, no. 1, November 2024, online serial. [Online]. Available: <https://arxiv.org/pdf/2411.09224.pdf>
- [7] C. B. et al., “Taking flight with copilot: Early insights and opportunities of ai-powered pair-programming tools,” *Queue*, vol. 20, no. 6, pp. 35–57, November 2022, abstract. [Online]. Available: <https://dl.acm.org/doi/abs/10.1145/3582083>
- [8] J. J. U. R. et al., *Integration of Generative AI with ChatGPT in Software Development*. Oxford University Press, 2024, online document. [Online]. Available: https://iydt.wordpress.com/wp-content/uploads/2024/05/2_31_integration-of-generative-ai-with-chatgpt-in-software-development_.pdf
- [9] M. Dolata, N. Lange, and G. Schwabe, “Development in times of hype: How freelancers explore generative ai?” in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, May 2024. [Online]. Available: <https://dl.acm.org/doi/abs/10.1145/3597503.3639111>