

Efecto de la IA sobre las habilidades técnicas de los desarrolladores de software

Effect of AI on the Technical Skills of Software Developers

Angie Jackeline Montes Jácome¹ <https://orcid.org/0009-0004-6026-9367>

¹Universidad TECH de México, México

montesjacome@hotmail.com



Esta obra está bajo una licencia internacional
Creative Commons Atribución-NoComercial 4.0

Enviado: 2026/01/09

Aceptado: 2026/02/19

Publicado: 2026/06/15

Resumen

Este artículo estudia el impacto de la inteligencia artificial (IA) en las habilidades técnicas de los programadores de software, con énfasis en la resolución de problemas, codificación y productividad. La investigación adoptó un enfoque metodológico mixto, combinando un análisis documental de literatura científica vigente con la aplicación de encuestas estructuradas a programadores con diversos niveles de experiencia. Los hallazgos revelaron un empleo habitual de los recursos de IA en torno al desarrollo, generalmente para apoyar el aprendizaje y efectuar tareas de rutina, además de una opinión mayoritaria de mejoras en la productividad. Sin embargo, se detectaron distinciones notables de acuerdo al grado de experiencia, demostrando una mayor confianza y dependencia de la IA en programadores con menor trayectoria, influyendo en la utilización de habilidades de análisis, creatividad y de autonomía tecnológica. Concluyendo así que la IA ejerce un efecto importante sobre las habilidades técnicas del programador, potenciando la eficacia operativa en contextos determinados, pero también transformando la forma en que se practican el razonamiento lógico y la resolución autónoma de problemas, en especial, en fases tempranas de capacitación profesional.

Palabras clave: Dependencia, productividad, programación, software.

Sumario: Introducción, Materiales y Métodos, Resultados y Discusión, Conclusiones.

Como citar: Montes, A. (2026). Efecto de la IA sobre las habilidades técnicas de los desarrolladores de software. *Revista Tecnológica - Espol*, 387(1), 87-102.

<https://rte.espol.edu.ec/index.php/tecnologica/article/view/1417>

Abstract

This article examines the impact of artificial intelligence (AI) on the technical skills of software developers, with a particular focus on problem-solving, coding, and productivity. The study adopts a mixed-methods approach, combining a review of current scientific literature with the administration of structured surveys to programmers with varying levels of experience. The results reveal a frequent use of AI tools in software development, primarily to support learning and perform routine tasks, along with a generally positive perception of increased productivity. However, notable differences were identified across experience levels, with less experienced programmers showing greater trust in and reliance on AI, which influences the use of analytical skills, creativity, and technological autonomy. The study concludes that AI has a significant effect on developers' technical skills, enhancing operational efficiency in specific contexts, while also transforming how logical reasoning and autonomous problem-solving are practiced, particularly during the early stages of professional training.

Keywords: Dependence, productivity, programming, software.

Introducción

La inteligencia artificial (IA), a través de nuevos métodos de automatización y apoyo, ha cambiado de manera significativa la forma tradicional en que se desarrolla el software (Huet, 2024), transformando procesos como la generación automática de código, depuración inteligente y el proceso decisorio dentro del ciclo de vida del desarrollo de software, con mejoras relevantes en cuanto a productividad y calidad del producto final. Históricamente, los programadores han tenido que buscar soluciones en foros, documentos técnicos y plataformas colaborativas (Demera et al., 2023). Con la llegada de las herramientas de IA como *GitHub Copilot*, impulsadas por modelos de lenguaje natural, los programadores son ahora capaces de crear proyectos insertando descripciones en lenguaje natural y generando de manera automática fragmentos completos de código, que antes se hacían progresivamente (Pedraza, 2023). Se presenta entonces una pregunta ética de fondo: ¿Cómo la IA está cambiando fundamentalmente el trabajo de un programador y sus interacciones con el ciclo de vida del desarrollo de software?

Este ha sido un tema abordado desde distintas perspectivas; no obstante, la literatura vigente aún no presenta un consenso concreto en torno a la incidencia real de la IA acerca de las capacidades técnicas del programador, en especial, en términos de independencia y pensamiento crítico. En una primera fase evolutiva, los recursos aislados se limitaban generalmente a funciones básicas como el autocompletado de código y el “debugueo” simple, las cuales tenían un dominio que predomina en los ambientes de desarrollo (Bolaño & Duarte, 2023).

Por su parte, durante la década de los 2000, el enfoque no estaba centrado en la generación automática de soluciones completas, sino en determinar las bases del apoyo computacional al programador, impulsándose de forma principal en sistemas de sugerencias y autocompletado de código, demostrando incrementos focales en la eficacia operativa, pero sin reemplazar el razonamiento lógico de los programadores (IBM, 2024). En esta fase, las herramientas de IA ofrecían respuestas fragmentadas y limitadas, por lo que el desarrollador debía mantener un rol activo en cuanto a la búsqueda, validación e inclusión de la información, invirtiendo un esfuerzo notable al consultar foros, documentos técnicos y otras herramientas disponibles en Internet (Galván, 2023).

Al pasar el tiempo, los recursos de IA empezaron a mostrar un progreso notable con la aparición de modelos de lenguaje de gran escala entrenados para actividades de programación,

cuya capacidad para generar código funcional ha sido valorada de manera empírica en investigaciones vigentes (Alenezi & Akour, 2025; Chen et al., 2021). En este sentido, modelos documentados ampliamente como GPT - 3 y GPT - 4 impulsaron herramientas que dejaron de limitarse al autocompletado de líneas aisladas y avanzaron hacia la generación de fragmentos de código, incidiendo en la manera en que los desarrolladores escriben, revisan y conceptualizan el software (Pombo, 2022). A su vez, revisiones sistemáticas actuales en ingeniería de software revelan que versiones más emergentes y avanzadas, como GPT - 5, se inscriben dentro de esta misma línea evolutiva, anticipando un incremento en la automatización cognitiva del desarrollo, pese a que su impacto empírico directo aún se halla en etapa de valoración científica (Galván, 2023).

Posteriormente en el año 2021, la compañía *OpenAI* lanzó la herramienta *GitHub Copilot*, que fue un momento clave marcado por el empleo del modelo Codex centrado en GPT - 3; este recurso no solo recomienda fragmentos de código, sino que además es capaz de elaborar funciones completas a partir de lenguaje natural, representando un cambio enorme en la manera de programar. Esta clase de herramientas está transformando el desarrollo de software, al momento de permitir que los programadores se enfoquen más en elementos creativos o estratégicos del proyecto, en lugar de centrarse en detalles minuciosos en su código (Pedraza, 2023).

A la par, la automatización creciente de la IA en el campo de la programación ha suscitado preocupaciones por la falta de criterio y pensamiento crítico en los desarrolladores de software (Guerra & Ruales, 2024). *Thoughtworks* (2024) señala que, si bien la IA puede incrementar la productividad, también existe una desigualdad en la programación cuando se trata de resolver problemas complejos de manera autónoma. Este fenómeno ha sido descrito como dependencia u *overreliance* en sistemas de IA, comprendida como la tendencia a confiar y adoptar de forma automática las recomendaciones de la inteligencia artificial, reduciendo los procesos de verificación crítica (Villacreses et al., 2022; Klingbeil et al., 2024); y se ha conceptualizado como un síndrome de dependencia tecnológica, vinculado a la pérdida progresiva de autonomía cognitiva, cuando el empleo de tecnologías inteligentes se vuelve usual (Barke et al., 2022). En este artículo, la dependencia será medida por medio de la frecuencia de recursos de IA y el nivel de aceptación directa del código generado, permitiendo valorar de manera empírica este constructo en los programadores.

Se puede notar en las tendencias de adopción que los programadores menos experimentados tienden a confiar más en las herramientas de IA, mientras que los programadores más avanzados suelen usarlas como alternativas en lugar de soluciones independientes (Mikova, 2024). Tal como se evidencia en diversas investigaciones, aunque los desarrolladores principiantes acentúan la IA como un recurso poderoso para completar sus tareas en poco tiempo, los programadores con mayor experiencia adoptan un contexto más crítico, donde examinan y modifican los códigos generados por la IA (Erazo et al., 2023; Pastor, 2025). Además, estudios vigentes confirman estas distinciones, evidenciando que los programadores juniors presentan mayores niveles de confianza y dependencia en las recomendaciones generadas por recursos de IA, mientras que los seniors mantienen un mayor control cognitivo y criterio técnico durante su empleo (Passi & Vorvoreanu, 2024; Vaithilingam et al., 2022).

Uno de los inconvenientes principales abordados se encuentra asociado a la IA y cómo cambia el enfoque de los programadores en cuanto a la resolución de problemas mientras desarrollan software (Salazar & Velastegui, 2024). No hay duda de que integrar herramientas de IA ha demostrado ser útil para mejorar la productividad y la eficiencia; sin embargo, se debe

cuestionar si hacerlo disminuye o no la necesidad de habilidades y conocimientos fundamentales de programación (Hernández, 2022). Así mismo, se pretende comprender de qué forma el uso de la IA influye en el pensamiento autónomo y las capacidades para resolver código de los desarrolladores.

Ante lo expuesto, este trabajo busca explorar la influencia de la IA en relación con el rol de un programador de sistemas, generalmente en cuanto a la resolución de problemas en el desarrollo. En función de este propósito, se plantea como pregunta de investigación: ¿Hay una relación entre el empleo de recursos de IA y los niveles de dependencia, autonomía y capacidades de resolución de problemas en el desarrollo de software? Por ello, se realizará el análisis de los datos de encuestas recopilados a programadores para establecer qué cambios han ocurrido en relación con su comportamiento de búsqueda, la dependencia de herramientas de IA y la autopercepción de creatividad y autonomía en la programación.

En este contexto, pese a que diversos estudios internacionales han explorado la relación entre el uso, experiencia y confianza en recursos de IA en programación, existe escasa evidencia empírica en el contexto de Ecuador, por ende, el presente trabajo posee como objetivo validar en el ámbito nacional los resultados declarados en estudios previos, aportando evidencia contextualizada que permita entender cómo se revelan estas dinámicas en los programadores ecuatorianos.

Materiales y Métodos

Diseño y enfoque de la investigación

La presente investigación aplicó un diseño de tipo mixto al combinar el análisis documental con la encuesta a programadores informáticos. El componente cualitativo se centró en un análisis temático de la literatura científica, permitiendo identificar enfoques, tendencias y preocupaciones habituales acerca del impacto de la IA en el desarrollo de software, mientras que, el componente cuantitativo se fundamentó en la aplicación de encuestas. De esta manera, se logró tener un contraste más completo respecto al impacto que la IA posee sobre el desarrollo de software, tanto desde la práctica como desde la teoría recopilada de los especialistas en materia.

Población y muestra

La población se conformó por programadores de software con diversos niveles de experiencia, en especial, aquellos que han utilizado herramientas de IA para desarrollar sistemas informáticos. La muestra fue no probabilística por conveniencia, compuesta por 30 participantes, por lo que los resultados obtenidos se deben interpretar de forma exploratoria, pues no permiten una generalización estadística a la totalidad de la población de programadores, aunque sí brindan insumos importantes para entender tendencias iniciales en el empleo de IA en el desarrollo de software.

Procedimiento para la recolección de datos

El proceso de recolección de datos se realizó en dos etapas:

Análisis documental

Se efectuó un análisis de literatura basado en artículos científicos y otros documentos como informes técnicos alojados en páginas webs y trabajos anteriores, que trataban sobre la implantación de recursos de IA en programación. En particular, se valoraron los estudios de 10 años previos incluidos en repositorios IEEE Xplore, Scopus y Google Scholar. Este análisis buscaba identificar los patrones, los beneficios y los problemas más destacados relacionados con la IA en la programación.

Para la selección de la literatura se aplicaron palabras clave como “inteligencia artificial”, “IA en la programación”, “desarrollo de software”, “creación de sistemas informáticos con IA”, entre otros. Los informes seleccionados incluyeron artículos científicos, informes técnicos, trabajos de titulación, proyectos investigativos y todos los documentos que aportaban evidencia empírica y un estudio detallado acerca de la implantación y efectos de la IA en el área de programación.

Encuestas a programadores

Para complementar el análisis anterior, se diseñó una encuesta estructurada con una serie de preguntas dirigidas a desarrolladores de software con distintos niveles de experiencia, siendo principiantes, intermedios o seniors, quienes dieron su consentimiento informado para participar en la encuesta de manera anónima y voluntaria. Para su elaboración, se identificaron temáticas clave y se seleccionaron ítems relevantes con la guía de estudios previos y consultas con expertos en el área, donde se incluyeron preguntas cerradas y abiertas, enfocadas en:

- Frecuencia y clase de recursos de IA empleados en la programación.
- Comparación entre la búsqueda de soluciones por medio de la web y el uso de la IA.
- Opinión acerca del impacto de la IA en el pensamiento crítico y creativo al momento de solucionar problemas.
- Dependencia y confianza en las recomendaciones proporcionadas por la IA.

La consistencia interna del cuestionario se evaluó por medio del coeficiente *alpha* de *Cronbach*, obteniéndose un valor de $\alpha = 0,82$, lo que señala una fiabilidad apropiada del instrumento. El cuestionario de encuesta fue distribuido en línea por medio de *Google Form* durante febrero y marzo del 2025, y estuvo dirigido a una comunidad activa de programadores de sistemas. Se utilizaron técnicas de muestreo por conveniencia de no probabilidad para elegir participantes con experiencia previa utilizando *GitHub Copilot*, *ChatGPT* o herramientas similares. Dicho esto, la muestra estuvo conformada por 30 personas con los siguientes criterios:

Criterios de inclusión

- Profesionales graduados en áreas de desarrollo de software
- Experiencia en el área de programación.
- Uso de recursos de IA.
- Mayores de edad (>18 años).
- Disponibilidad para completar el cuestionario.
- Nacionalidad ecuatoriana.
- Brindaron su consentimiento informado.

Criterios de exclusión

- Inexperiencia en el uso de IA.
- Participación incompleta.
- Personas que no trabajen en áreas tecnológicas.
- Profesionales de otros países.
- No brindaron su consentimiento informado.

Análisis de datos

Los datos que se obtuvieron mediante encuestas, estos fueron procesados a través de técnicas estadísticas empleando Microsoft Excel. A su vez, para la interpretación de hallazgos, se efectuó una comparativa mediante un análisis descriptivo que incorporaba cálculo de frecuencias relativas y absolutas, además de descriptivas entre variables clave, cuyos hallazgos fueron contrastados con la información de la revisión bibliográfica.

Resultados y Discusión

Análisis documental sobre la inteligencia artificial en la programación

El análisis documental demuestra que el impacto de la IA en el desarrollo de software ha pasado de un enfoque teórico y experimental hacia una aplicación extendida y práctica, impulsada por los progresos acelerados de los últimos años (Serrahima, 2022). Por su parte, la literatura coincide en que, durante la década de los 2000, los programadores centraban su trabajo en la consulta de documentación técnica y foros especializados como *Stack Overflow* o *GitHub Discussions*, los cuales cumplían una función esencial en la resolución de problemas y el aprendizaje independiente (Dávila & Aguero, 2023; Xatruch, 2025).

A su vez, investigaciones más vigentes reportan que las herramientas centradas en IA, como *ChatGPT*, han introducido un paradigma novedoso al brindar explicaciones contextuales y fragmentos de código en tiempo real, transformando los procesos convencionales de búsqueda y resolución de problemas en programación (Coca & Llivina, 2021). En este contexto, Basantes (2023) destaca que recursos como *CodeWhisperer*, *ChatGPT* y *GitHub Copilot* han automatizado solicitudes recurrentes de la programación, contribuyendo con mejoras operacionales en términos de eficacia y productividad.

Sin embargo, la literatura además advierte que este progreso digital conlleva retos relacionados. Bayas (2024) revela que el empleo inapropiado o excesivo de herramientas de IA puede derivar en una reducción del razonamiento crítico y creativo del desarrollador, promoviendo enfoques mecánicos y poco reflexivos en la resolución de problemas. Haga clic o pulse aquí para escribir texto. stos problemas se han abordado en trabajos que señalan que la automatización cognitiva no reemplaza las capacidades analíticas esenciales, sino que necesita una inspección consciente por parte del programador.

De la misma forma, diferentes estudios documentales enfatizan que la adopción de recursos de IA no es uniforme y se influencia por el grado de experiencia del desarrollador (Barragán, 2023). Dans (2024) declara que los programadores juniors suelen utilizar la IA de forma más directa y dependiente en la generación de código. Por el contrario, los desarrolladores seniors emplean estos recursos como soporte, dando prioridad a la validación y modificación de las sugerencias generadas (Csic, 2019). De tal modo, la evidencia documental señala que el impacto de la inteligencia artificial en la programación se condiciona por el nivel de automatización y el criterio técnico de los usuarios.

En contraste, artículos recientes como el de Nascimento et al. (2023) brindan evidencia empírica y conceptual en torno a la incidencia de la IA en el desarrollo de software, por ejemplo, se comparó de forma empírica recursos de IA con ingenieros de software en actividades de programación, hallando variaciones en métricas de calidad y productividad que dependen del contexto de la adopción de esta herramienta informática. De forma adicional, estudios como el de Sauvola et al. (2024) revelan que la IA generativa cambia de manera crítica los procesos convencionales de software, automatizando tareas repetitivas y definiendo desafíos éticos y de propiedad intelectual.

Para finalizar, revisiones más recientes como la de Kam et al. (2025) indagan lo que los desarrolladores profesionales necesitan para incluir de manera eficaz la IA en su práctica profesional, detectando tanto ventajas potenciales como limitaciones asociadas con formación y confianza dirigida hacia estas herramientas.

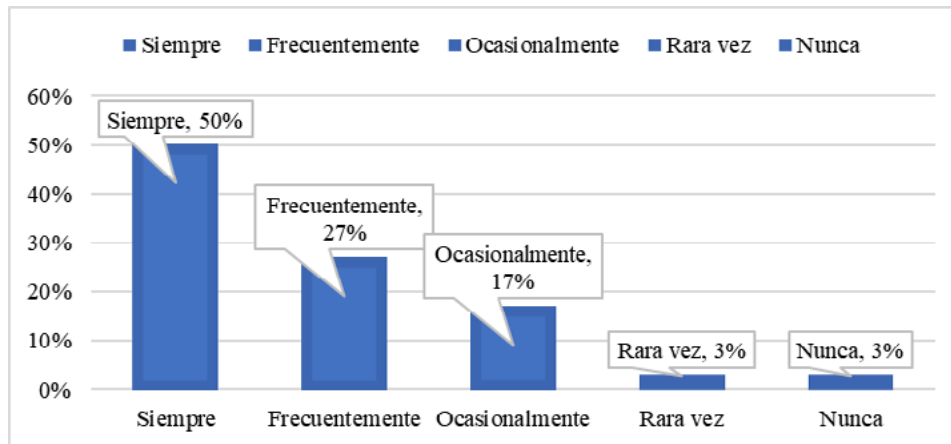
Resultados de la encuesta dirigida a desarrolladores de software

Los resultados de la encuesta a programadores de software (<https://forms.gle/zqvrkwyxyXfMnrw69>) se describen a continuación:

1. ¿Con qué frecuencia emplea recursos de inteligencia artificial para codificar o resolver tareas de programación?

Figura 1

Frecuencia de uso de recursos IA para programar



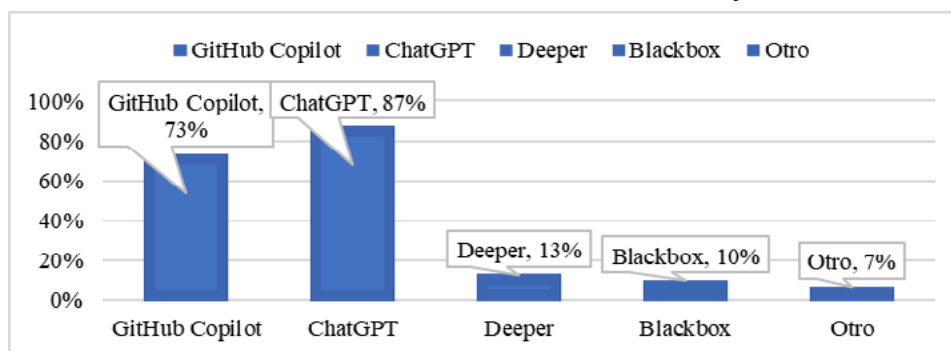
Nota: Resultados de encuesta aplicada a programadores (n = 30), 2025.

Análisis e interpretación: La frecuencia de uso de herramientas de IA se estudió por medio de una escala ordinal de cinco categorías (nunca - siempre), cuyos hallazgos revelaron que el 50% de los participantes informó emplearlas siempre y el 27% frecuentemente, lo que demostró un uso habitual de dichas tecnologías en la práctica del desarrollo. Este resultado permitió analizar el nivel de integración de la IA en las actividades profesionales; sin embargo, la medición se limitó al empleo declarado y no a indicadores objetivos de productividad, como tiempo de desarrollo o calidad del código. Investigaciones recientes evidencian que herramientas de IA pueden mejorar la velocidad de desarrollo en actividades específicas, pese a que sus ventajas son variables de acuerdo a la clase de tarea y la experiencia del programador (Barke et al., 2022; Vaithilingam et al., 2022).

2. ¿Cuáles de las siguientes herramientas IA utiliza con mayor frecuencia en su trabajo? Puede escoger más de una opción.

Figura 2

Herramienta IA más utilizada en el trabajo



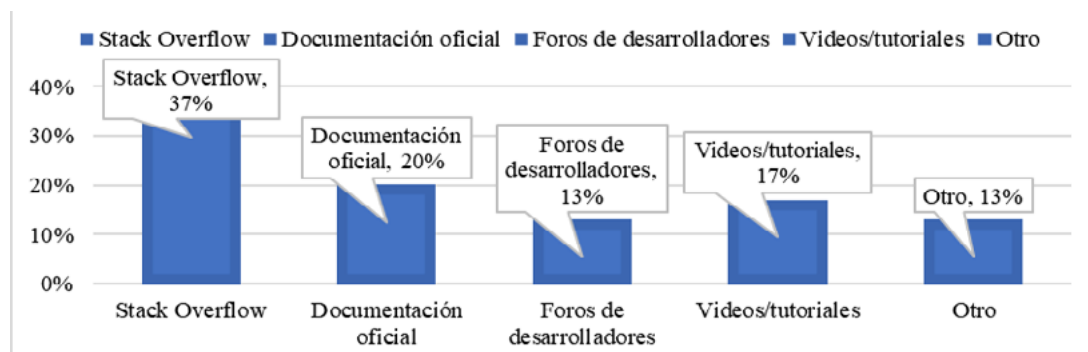
Nota: Resultados de encuesta aplicada a programadores (n = 30), 2025.

Análisis e interpretación: La interrogante fue de selección múltiple, permitiendo a los encuestados marcar más de una herramienta de IA empleada en su trabajo. Los resultados evidenciaron que ChatGPT fue el recurso más usado (87%), principalmente para buscar soluciones y tener apoyo conceptual, seguida de GitHub Copilot (73%), empleada sobre todo para la asistencia directa en la escritura y autocompletado de código; en consecuencia, recursos como Deeper y Blackbox presentaron una adopción menor con 13% cada una. Esta distinción por clase de uso coincide con trabajos recientes, que demuestran que los desarrolladores utilizan asistentes conversacionales para razonamiento y comprensión, mientras que los asistentes integrados al IDE se usan para codificar de forma instantánea (Amin et al., 2024).

3. Antes de usar recursos de IA, ¿dónde solía buscar soluciones a los problemas de programación?

Figura 3

Búsqueda de soluciones a problemas de programación



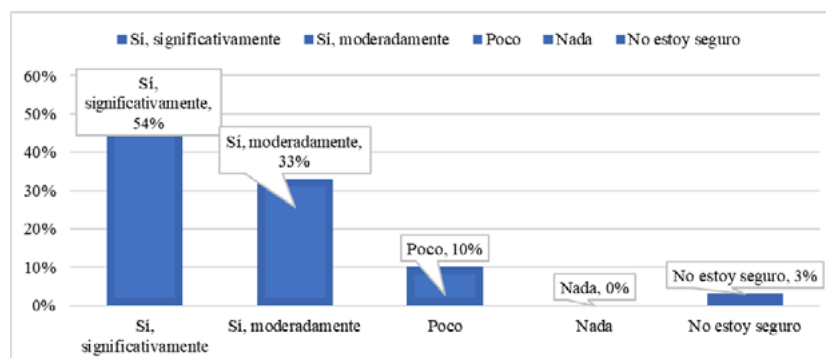
Nota: Resultados de encuesta aplicada a programadores (n = 30), 2025.

Análisis e interpretación: Los datos demostraron que, antes de utilizar la IA, los programadores buscaban información en foros como Stack Overflow, y en documentos oficiales técnicos, evidenciando un cambio en cuanto a fuentes de consulta hacia soluciones con automatización. Esta variación se traduce en una modificación en los hábitos de búsqueda e implicaciones en el aprendizaje autónomo, puesto que la disminución de la lectura técnica y la indagación activa puede afectar la formación de competencias analíticas y de entendimiento profundo del código fuente (Bayas, 2024). Diversos autores señalan que, si bien los recursos de IA facilitan la resolución inmediata de problemas, pueden disminuir la exploración deliberada y aprendizaje activo, en especial, en programadores con menos experiencia (Weber et al., 2024).

4. En su opinión, ¿ha tenido una incidencia positiva la utilización de recursos de IA en tu productividad en el desarrollo de sistemas?

Figura 4

Incidencia del uso de recursos de IA



Nota: Resultados de encuesta aplicada a programadores (n = 30), 2025.

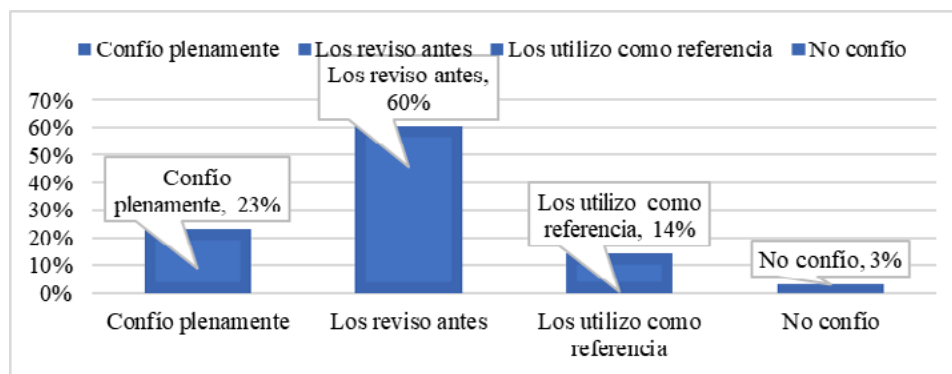
Análisis e interpretación: La productividad se evaluó mediante un indicador de productividad percibida, el cual se midió a través de una escala ordinal en la que los participantes valoraron la incidencia del empleo de recursos de IA en su rendimiento. Dicho esto, más del 80% de los programadores percibió un impacto favorable de la IA en su productividad, relacionado a la automatización de actividades repetitivas y de bajo nivel, como la generación de código a partir de descripciones de lenguaje natural y autocompletado de fragmentos; esta clase de tareas tiende a liberar tiempo que puede dedicarse a aspecto más creativos o tácticos del desarrollo.

En contraste, se logró apreciar una tendencia donde los programadores de nivel principiante/intermedio declararon mejoras percibidas; no obstante, los desarrolladores de nivel avanzado que emplean la IA solo lo hacen para apoyo puntual. Esta situación reveló que el beneficio de la IA está concretado en actividades de rutina, en lugar de decisiones de arquitectura, enlazándose con estudios que muestran mejoras productivas por utilización de modelos de lenguaje en acciones de codificación. En definitiva, estos hallazgos se asemejan a trabajos vigentes que evidencian incrementos, especialmente en tareas de código de bajo nivel (Xatruch, 2025; Yavich, 2025).

5. ¿Cuánto confías en las respuestas o fragmentos de código proporcionados por herramientas de IA?

Figura 5

Confianza en las respuestas proporcionadas de la IA



Nota: Resultados de encuesta aplicada a programadores (n = 30), 2025.

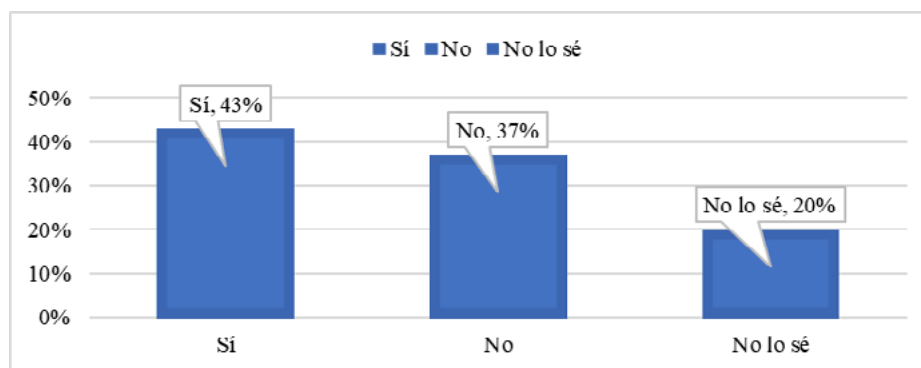
Análisis e interpretación: En la encuesta, el 60% confirmó que revisa y modifica las recomendaciones, lo que demostró una actitud responsable y racional hacia el uso de IA. Por otro lado, el 23% confía totalmente en las respuestas generadas, lo que planteó preguntas sobre la validación del código y el peligro de la dependencia excesiva en la automatización.

La IA ha mejorado el desempeño de los programadores al facilitar y optimizar tareas; no obstante, los niveles altos de confianza identificados en una parte de los participantes evidenciaron un riesgo de dependencia cuando las sugerencias no son validadas de manera crítica. Dicho comportamiento suele ser más habitual en programadores junior, quienes pueden aceptar soluciones generadas por IA con menos cuestionamiento, a su vez, los programadores senior tienden a revisar y ajustar el código. Investigaciones actuales han definido que, si bien la IA puede apoyar el aprendizaje y la eficiencia, una dependencia excesiva puede inhibir el desarrollo de habilidades críticas y la toma de decisiones autónomas, afectando las capacidades de evaluación y análisis profundo (Dávila & Aguero, 2023; Tong, 2025).

6. ¿Crees que el uso constante de IA podría impedirte ejercer tus habilidades fundamentales de programación (lógica, sintaxis y resolución de problemas)?

Figura 6

Impedimento para ejercer las habilidades de programación



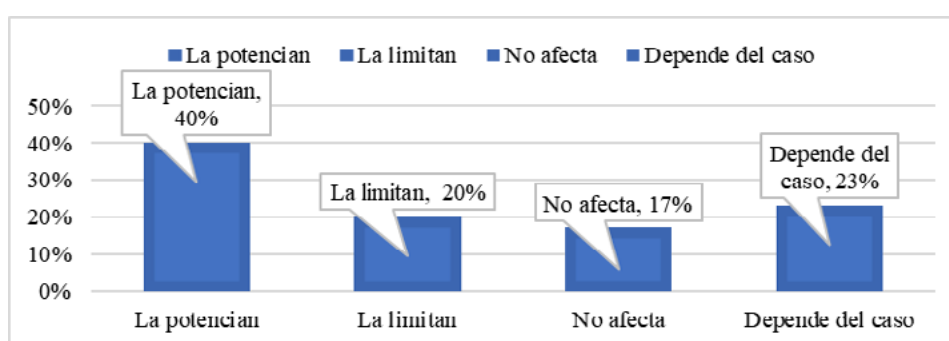
Nota: Resultados de encuesta aplicada a programadores (n = 30), 2025.

Análisis e interpretación: El 43% sostuvo que la implementación constante de herramientas de IA le resta alguna capacidad básica, en tanto que el 37 % consideró que no hay efectos negativos, lo que reflejó una percepción dividida acerca del impacto de la IA en habilidades esenciales de programación. Esta opinión muestra correlación empírica con investigaciones vigentes donde se ha evidenciado que, en tareas familiares, programadores seniors asistidos por IA tardaron más en completar actividades que aquellos que no la emplearon, debido al tiempo adicional invertido en revisar y corregir sugerencias automatizadas, lo cual puede afectar la práctica de capacidades cognitivas profundas y el pensamiento crítico en tareas manuales complejas. A su vez, estos efectos parecen variar de acuerdo al nivel de experiencia, puesto que la familiaridad con el código y las tácticas de resolución inciden en la manera en que cada programador incluye e inspecciona las recomendaciones de IA (Weisz et al., 2025).

7. ¿Crees que las herramientas de IA representan una restricción o un aliciente a la creatividad que se posee para solucionar problemas de programación?

Figura 7

Restricción o aliciente a la creatividad



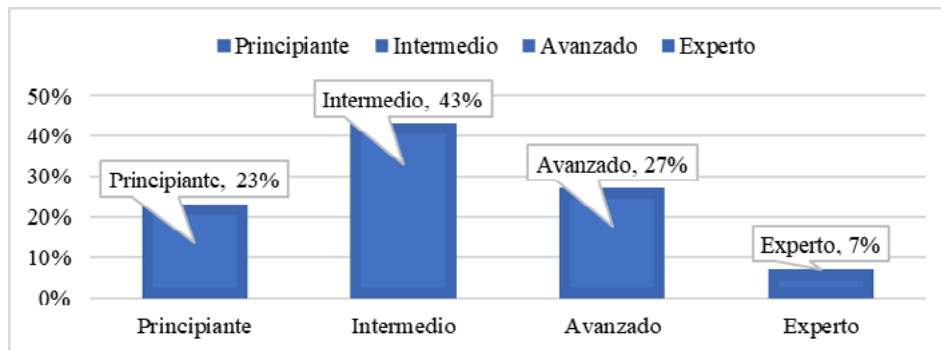
Nota: Resultados de encuesta aplicada a programadores (n = 30), 2025.

Análisis e interpretación: Hasta un 40% consideró que la IA mejora su creatividad cuando se enfoca en la lógica del problema, mientras que el 20% sintió que la restringe. Esto indicó que hay una división en la percepción en lo que respecta al impacto creativo: algunos desarrolladores son capaces de aprovechar la automatización, mientras que otros sienten que hay una falta de iniciativa y exploración autónomas. Por un lado, se ha demostrado que los asistentes de IA liberan a los programadores de actividades rutinarias, como generar documentación o escribir código, permitiéndoles dedicar mayor tiempo y energía a la ideación de soluciones, diseños de

algoritmos y exploración de marcos innovadores. En contraste, estudios revelan que cuando la IA genera soluciones automáticamente, puede reducir la necesidad de un razonamiento autónomo, disminuyendo la práctica de habilidades cognitivas profundas y limitando la exploración independiente del desarrollador, en especial, en actividades de alto nivel (Holzner et al., 2025; Joshi et al., 2025).

8. ¿Cómo calificaría su nivel de habilidad en la programación?

Figura 8
Nivel de habilidad en la programación



Nota: Resultados de encuesta aplicada a programadores (n = 30), 2025.

Análisis e interpretación: La mayoría de la población encuestada se consideraron programadores en rango intermedio (43%) o avanzado (27%), reflejando una confianza general en sus conocimientos y una disposición a estudiar nuevas tecnologías, como los recursos de IA. No obstante, este resultado necesita un análisis con mayor detalle, pues un cruce entre el nivel de habilidad y las respuestas acerca de la confianza en las herramientas de IA, la dependencia en sus sugerencias y la percepción de la creatividad podría brindar una visión con mayor precisión de cómo los distintos niveles de experiencia influyen en su uso. Por ejemplo, los programadores juniors suelen fiarse ciegamente en las recomendaciones de la IA, mientras que los de nivel senior son más selectivos y críticos en su empleo; dichas tendencias de comportamiento fortalecen la literatura que sugiere que los desarrolladores con más experiencia utilizan la IA como complemento táctico, mientras que los que tienen menos experiencia pueden depender más de ella para actividades rutinarias (Pastor, 2025).

9. ¿Qué ideas tienes sobre el avance y el futuro de la programación con la inclusión de herramientas de IA?

Las respuestas de los entrevistados demostraron que los desarrolladores perciben a la IA como un elemento focal en el futuro de la programación; sin embargo, esta adopción viene acompañada de una preocupación constante por la dependencia. Diversos participantes manifestaron que una automatización excesiva podría limitar su habilidad para escribir, entender o depurar código sin asistencia, coincidiendo con trabajos que advierten que una dependencia sostenida de recursos generativos puede disminuir la práctica independiente de capacidades técnicas esenciales (Coca & Llivina, 2021).

Con respecto a la creatividad, los entrevistados revelaron percepciones divididas; algunos creen que la IA actúa como un aliciente al permitirles centrarse en la lógica del problema y estudiar soluciones con mayor agilidad, mientras que otros perciben que la generación automática de código puede limitar la iniciativa personal y la exploración autónoma. Dicha tensión ha sido documentada en la literatura vigente, donde se indica que la IA puede apoyar procesos creativos cuando se emplea como recurso de co-creación, pero también puede minimizar la diversidad de ideas si domina el proceso generativo de soluciones.

De la misma manera, los cambios en los hábitos de aprendizaje declaran una transición hacia formas más rápidas de adquisición de conocimientos, centradas en la observación y transformación de código generado por IA. Pese a que este enfoque agiliza el aprendizaje práctico, los entrevistados manifestaron preocupación por la pérdida de hábitos investigativos y de entendimiento profundo, en especial en fases tempranas formativas. Investigaciones recientes sobre desarrollo de software advierten que este fenómeno puede afectar la construcción de habilidades de análisis si no existe una inspección consciente del empleo de la inteligencia artificial (Jackson et al., 2025).

Es así que, dichos hallazgos mostraron que el progreso de la programación en la era de la inteligencia artificial es una oportunidad, así como un desafío. De tal modo, resulta esencial promover una formación técnica y con ética que permita a los desarrolladores integrar estos recursos e IA de forma crítica y óptima, considerando sus ventajas sin erosionar la independencia, sus habilidades de análisis y la calidad de la programación.

Análisis bivariado entre el nivel de experiencia y variables clave

Con el propósito de fundamentar de forma empírica las diferencias evidenciadas en los análisis descriptivos, se efectuaron pruebas chi-cuadrado de independencia entre el grado de experiencia del programador y variables clave como el nivel de confianza en las respuestas generadas por recursos de IA y la percepción de mejora en la productividad.

En este sentido, el estudio entre grado de experiencia y nivel de confianza demostró una asociación significativa de forma estadística ($\chi^2 = 10,24$; $gl = 9$; $p = 0.0037$), revelando que el análisis de frecuencias cruzadas permitió detectar que la relación entre las dos variables es inversamente proporcional, puesto que a medida que incrementa el grado de experiencia, se reduce la aceptación directa y automática de las sugerencias generadas por la IA; es decir, los programadores de nivel principiante/intermedio presentan mayores grados de aceptación directa de las recomendaciones de la IA, mientras que los desarrolladores avanzados suelen revisar y ajustar el código antes de implementarlo.

En segundo lugar, se exploró la relación entre el grado de experiencia y percepción de mejora en la productividad, hallando que de igual forma existe una asociación estadística significativa ($\chi^2 = 11,08$; $gl = 9$; $p = 0.026$), siendo inversamente proporcional, ya que, los hallazgos indicaron que los participantes con menos experiencia informan mayores grados de mejora percibida, mientras que los programadores seniors emplean la IA como recurso puntual de apoyo.

Dichos resultados fortalecen de forma cuantitativa las tendencias detalladas en los análisis descriptivos previos, demostrando que el grado de experiencia actúa como una variable moduladora en la percepción y utilización de herramientas de IA en la programación de sistemas.

Igualmente, tomando en cuenta que los participantes del estudio fueron desarrolladores de nacionalidad ecuatoriana, es viable revelar que aspectos formativos y laborales propios del contexto nacional podrían incidir en los hallazgos obtenidos. En Ecuador, la formación en programación suele enlazar enseñanza académica convencional con aprendizaje autodidacta apoyado en herramientas digitales lo que podría explicar la adopción ágil de recursos de IA como apoyo práctico. Del mismo modo, el mercado laboral tecnológico en el territorio nacional, distinguido por elevada demanda de eficacia y optimización de tiempos en proyectos de desarrollo, podría incentivar el empleo habitual de asistentes inteligentes para la mejora de la productividad. Por ende, los resultados se deben interpretar dentro de este plano regional, reforzando el aporte contextual del trabajo.

Amenazas a la validez y confiabilidad del estudio

El presente estudio posee diversas limitaciones que se deben tomar en cuenta al interpretar los hallazgos. Con respecto a la validez interna, al tratarse de un diseño no experimental y de corte transversal centrado en encuestas auto informadas, los datos revelaron percepciones subjetivas de los encuestados, lo que implica que variables como mejora en la productividad o confianza en la IA no se midieron por medio de indicadores objetivos, como métricas de tiempo o de calidad del código, por lo cual, se podría introducir sesgos de sobreestimación o percepción.

Así mismo, en el marco de la validez externa, el grupo poblacional del que se extrajo la muestra fue la de desarrolladores ecuatorianos, por lo que sus resultados se deben contextualizar en el entorno formativo y laboral ecuatoriano; a su vez, aspectos académicos, culturales y digitales pueden inferir en otro contexto, limitando la generalidad directa de los datos recabados.

Además, en cuanto a la validez del constructo, los conceptos explorados se operacionalizaron a través de escalas ordinales de percepción, lo que puede no capturar totalmente la dificultad de constructos como confianza tecnológica o productividad.

Finalmente, en función de la validez de conclusión, pese a que las pruebas estadísticas que se aplicaron demostraron asociaciones significativas entre las variables, dichos enlaces no implican causalidad, dado el enfoque transversal de la investigación.

Conclusiones

El presente estudio evidencia que la IA ha adquirido una función esencial en las prácticas vigentes de desarrollo de software, siendo empleada de manera habitual por los programadores como apoyo para la codificación, resolución de problemas y aprendizaje técnico. En este sentido, la encuesta demuestra que la mayor parte de los participantes percibe una incidencia favorable de los recursos de IA en su productividad, en especial, en tareas rutinarias y de bajo nivel, como autocompletado de código y generación de fragmentos en función de descripciones en lenguaje natural. De este modo, la principal contribución de esta investigación es validar hallazgos internacionales dentro del contexto de Ecuador, demostrando que las tendencias observadas en el ámbito global en torno al uso y confianza en recursos de IA también se visualizan en desarrolladores nacionales, pese a que poseen particularidades propias del ambiente formativo y laboral de la nación.

Sin embargo, los resultados además destacan distinciones importantes de acuerdo al nivel de experiencia del desarrollador. El análisis de asociación por medio de la prueba chi-cuadrado afirma un vínculo estadísticamente significativo entre el grado de experiencia y el nivel de confianza en los recursos de IA, evidenciando que los programadores principiantes e intermedios suelen confiar más en las soluciones generadas por la inteligencia artificial o emplearlas como referencia directa. En este contexto, los desarrolladores avanzados y expertos presentan una utilización más selectiva y crítica, caracterizándose por la revisión, validación y adaptación del código recomendado. Estos resultados sugieren que la experiencia actúa como un elemento modulador del impacto de la IA sobre las capacidades técnicas, usualmente en lo referente al razonamiento lógico, la habilidad analítica y la independencia en la resolución de problemas.

De la misma manera, se identifica un cambio en los hábitos de aprendizaje y búsqueda de información, donde los recursos convencionales, como la documentación técnica y foros especializados, han sido parcialmente desplazados por herramientas de IA que proporcionan respuestas con inmediatez. Si bien este cambio promueve la rapidez y el aprendizaje práctico,

los participantes revelaron preocupaciones acerca de la posible reducción de la exploración autónoma y del razonamiento profundo, lo cual puede afectar el desarrollo sostenido de capacidades técnicas esenciales si no existe una inspección consciente del uso de la inteligencia artificial.

En torno a esto, los resultados reflejan que la IA no sustituye las habilidades técnicas del desarrollador, pero sí incide en la manera en que estas se practican y desarrollan, potenciándolas en diversos contextos y debilitándolas en otros, en especial, cuando se genera dependencia en fases tempranas de formación. Desde esta perspectiva, la incidencia de la IA sobre las capacidades técnicas depende menos del recurso en sí y más del criterio técnico y la experiencia del usuario.

Para trabajos futuros, se recomienda dar profundidad con estudios que contengan muestras más amplias y diseños metodológicos que permitan indagar el progreso de las capacidades técnicas con el tiempo, además de incorporar indicadores de rendimiento, como la calidad del código que generan estos recursos o los tiempos que tarda en brindar respuestas automáticas. Así mismo, es clave explorar tácticas de inclusión pedagógica de la IA en la capacitación de los desarrolladores, focalizadas en el reforzamiento del pensamiento crítico, la independencia y la capacidad de la resolución de problemas, lo que asegura que la inteligencia artificial funcione como una herramienta de apoyo y no como reemplazo del razonamiento técnico por parte del desarrollador de sistemas.

Reconocimientos y Declaraciones

La autora expresa que no fue necesario contar con un financiamiento para elaborar la presente obra investigativa.

La autora declara que, en la elaboración del presente artículo, no se ha utilizado herramientas de inteligencia artificial.

Referencias

- Alenezi, M., & Akour, M. (2025). AI-driven innovations in software engineering: a review of current practices and future directions. *Applied Sciences*, 15(3). <https://doi.org/10.3390/app15031344>
- Amin, M., Mathew, S., Khalil, A., Berengueres, J., & Hussain, J. (2024). “¿Me reemplazarán?” evaluación del efecto de ChatGPT en el desarrollo de software y la percepción de los programadores sobre las herramientas de IA. *Science of Computer Programming*, 235(1). <https://doi.org/10.1016/j.scico.2024.103111>
- Barke, S., James, M., & Polikarpova, N. (2022). Grounded Copilot: how programmers interact with code-generating models. *Arxiv*, 3(1). <https://doi.org/10.48550/arXiv.2206.15000>
- Barragán, X. (2023). Situación de la Inteligencia Artificial en el Ecuador en relación con los países líderes de la región del Cono Sur. *Figempa: Investigación y Desarrollo*, 16(2). <https://doi.org/10.29166/revfig.v16i2.4498>
- Basantes, R. (2023). *Estudio comparativo de las herramientas de IA (Inteligencia Artificial) Code GPT y Github copilot como asistentes de programación en el desarrollo de software* [Tesis de grado, Universidad Técnica de Babahoyo]. <https://dspace.utb.edu.ec/bitstream/handle/49000/13900/E-UTB-FAFI-SIST-000409.pdf?sequence=1&isAllowed=y>
- Bayas, L. (2024). El impacto de la inteligencia artificial en la limitación del desarrollo del pensamiento lógico y crítico en estudiantes universitarios. *ULEAM*, 8(18). <https://doi.org/10.56124/scicam.v8i018.001>

- Bolaño, M., & Duarte, N. (2023). Una revisión sistemática del uso de la inteligencia artificial en la educación. *Revista Colombiana de Cirugía*, 39(1). <https://doi.org/10.30944/20117582.2365>
- Chen, M., Tworek, J., Jun, H., Yuan, Q., & De Oliveira, H. (2021). Evaluating Large Language Models Trained on Code. *Arxiv*, 2(1). <https://doi.org/10.48550/arXiv.2107.03374>
- Coca, Y., & Llivina, M. (2021). *Desarrollo y retos de la inteligencia artificial* (I. Quesada, Ed.; 1st ed.). Editorial Educación Cubana. <https://www.entramar.mvl.edu.ar/wp-content/uploads/2023/09/2-Desarrollo-y-retos-de-la-IA.pdf>
- Csic. (2019, June 3). *El impacto de la Inteligencia Artificial en nuestra sociedad. Retos y oportunidades*. <https://www.csic.es/es/actualidad-del-csic/el-impacto-de-la-inteligencia-artificial-en-nuestra-sociedad-retos-y-oportunidades>
- Dans, E. (2024, April 18). *El interesante efecto de la IA sobre los desarrolladores*. <https://www.enriquedans.com/2024/04/el-interesante-efecto-de-la-ia-en-los-desarrolladores.html>
- Dávila, R., & Agüero, E. (2023). Desafíos éticos de la inteligencia artificial: implicaciones para la sociedad y la economía. *Conrado*, 19(94). http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1990-86442023000500137
- Demera, A., Sánchez, A., Franco, M., Espinoza, M., & Santana, G. (2023). Fundamentación teórica de la inteligencia artificial en el desarrollo de aplicaciones móviles en el Instituto de Admisión y Nivelación de la Universidad Técnica de Manabí. *TESLA Revista Científica*, 3(2). <https://tesla.puertomaderoeditorial.com.ar/index.php/tesla/article/view/223/276>
- Erazo, A., Ramos, F., Galarza, P., & Boné, M. (2023). La inteligencia artificial aplicada a la optimización de programas informáticos. *Ciencia Investigativa Multidisciplinaria*, 3(1). <https://doi.org/10.55813/gaea/jessr/v3/n1/61>
- Galván, P. (2023). *Impacto de la Inteligencia Artificial en el Desarrollo de Software*. <https://sg.com.mx/revista/56/inteligencia-artificial-desarrollo-software>
- Guerra, D., & Ruales, F. (2024). ¿Cómo influye el uso de la inteligencia artificial en el desarrollo de software? *Ciencia y Turismo*, 3(4). <https://doi.org/10.33262/ct.v3i4.53>
- Hernández, J. (2022, September 22). *Inteligencia artificial: qué aporta y qué cambia en el mundo del trabajo*. <https://blogs.iadb.org/trabajo/es/inteligencia-artificial-que-aporta-y-que-cambia-en-el-mundo-del-trabajo/>
- Holzner, N., Maier, S., & Feuerriegel, S. (2025). Generative AI and creativity: A systematic literature review and meta-analysis. *Cornell University*, 1(1). <https://doi.org/10.48550/arXiv.2505.17241>
- Huet, P. (2024, March 7). *Inteligencia artificial en desarrollo de software: Tendencias emergentes y futuro*. <https://openwebinars.net/blog/inteligencia-artificial-en-desarrollo-de-software/>
- IBM. (2024, October 7). *IA en el desarrollo de software*. <https://www.ibm.com/mx-es/think/topics/ai-in-software-development>
- Jackson, V., Vasilescu, B., Russo, D., Ralph, P., Prikladnicki, R., Izadi, M., D'Angelo, S., Inman, S., Andrade, A., & Der, A. (2025). The Impact of Generative AI on Creativity in Software Development: A Research Agenda. *ACM Transactions on Software Engineering and Methodology*, 34(133). <https://doi.org/10.1145/3708523>
- Joshi, O., Mundra, K., & Das, S. (2025). *AI coding assistants in competitive programming: empirical studies from India on human-AI interactions in learning, problem solving, and curiosity development Perspectives*. https://doi.org/10.1007/978-3-031-99261-2_17
- Kam, M., Miller, C., Wang, M., Tidwell, A., Lee, I., Malyn, J., Perret, B., Tiwari, V., & Kenitzer, J. (2025). What do professional software developers need to know to succeed in an age of Artificial Intelligence? *ACM*, 1(1). <https://doi.org/10.1145/3696630.3727251>

- Klingbeil, A., Grutzner, C., & Schereck, P. (2024). Trust and reliance on AI — An experimental study on the extent and costs of overreliance on AI. *Computers in Human Behavior*, 160(1). <https://doi.org/10.1016/j.chb.2024.108352>
- Mikova, K. (2024, December 12). *La verdad sobre el impacto de la IA en el desarrollo de software*. <https://www.appbuilder.dev/es/blog/impact-of-ai-on-software-development>
- Nascimento, N., Alencar, P., & Cowan, D. (2023). Artificial Intelligence vs. Software Engineers: An Empirical Study on Performance and Efficiency using ChatGPT. *ACM*, 1(1). <https://dl.acm.org/doi/abs/10.5555/3615924.3615927>
- Passi, S., & Vorvoreanu, M. (2024). Overreliance on AI: literature review. In *Scale Stanford* (Vol. 1, Number 1). <https://scale.stanford.edu/ai/repository/overreliance-ai-literature-review>
- Pastor, J. (2025, February 18). *Los jóvenes programadores ya no saben programar: la IA está provocando ahora lo mismo que provocó la calculadora hace medio siglo*. <https://www.xataka.com/robotica-e-ia/tenemos-a-primera-profesion-que-ia-ha-cambiado-para-siempre-programadores-no-necesitan-saber-programar>
- Pedraza, J. (2023). *La Inteligencia Artificial en la sociedad: Explorando su Impacto Actual y los Desafíos Futuros* [Universidad Politécnica de Madrid]. https://oa.upm.es/75068/1/TFG_JAROD_DAVID_PEDRAZA_CARO.pdf
- Pombo, C. (2022, December 7). *Los riesgos de la inteligencia artificial y algunas soluciones*. <https://blogs.iadb.org/conocimiento-abierto/es/riesgos-inteligencia-artificial/>
- Salazar, L., & Velastegui, D. (2024). Inteligencia Artificial y su Impacto en la Psicología Humana: Mini Revisión. *Revistas UTA*, 8(1). <https://doi.org/10.31243/mdc.uta.v8i1.2306.2024>
- Sauvola, J., Tarkoma, S., Klemettinen, M., Riekkki, J., & Doermann, D. (2024). Future of software development with generative AI. *Autom Softw Eng.*, 31(26). <https://doi.org/10.1007/s10515-024-00426-z>
- Serrahima, Á. (2022). *Avances y desafíos de la inteligencia artificial* [Tesis, Universidad Pontificia Comillas]. <https://repositorio.comillas.edu/rest/bitstreams/507551/retrieve>
- Thoughtworks. (2024). *El surgimiento de los codebots: Cómo la inteligencia artificial está cambiando el desarrollo de software*. https://www.thoughtworks.com/content/dam/thoughtworks/documents/report/tw_report_wired_ai_software_development_spanish.pdf
- Tong, A. (2025). *AI slows down some experienced software developers, study finds*. Reuters. <https://www.reuters.com/business/ai-slows-down-some-experienced-software-developers-study-finds-2025-07-10/>
- Vaithilingam, P., Zhang, T., & Glassman, E. (2022). Expectation vs. experience: evaluating the usability of code generation Tools powered by large language models. *Scite*, 1(1). <https://doi.org/10.1145/3491101.3519665>
- Villacreses, C., Chóez, J., Barreto, J., Figueroa, V., & Pin, L. (2022). La calidad del software asistida mediante inteligencia artificial. *Revista Científica Journal Techinnovation*, 1(1). <https://doi.org/10.47230/Journal.TechInnovation.v1.n1.2022.10-21>
- Weber, T., Brandmaier, M., Schmidt, A., & Mayer, S. (2024). Significant Productivity Gains through Programming with Large Language Models. *Proc. ACM Hum.-Comput. Interact.*, 8(256). <https://doi.org/10.1145/3661145>
- Weisz, J., Kumar, S., Muller, M., Browne, K., Heintze, E., & Bajpai, S. (2025). *Examining the use and impact of an AI code assistant on developer productivity and experience in the enterprise*. <https://arxiv.org/pdf/2412.06603>
- Xatruch, C. (2025). Desafíos y oportunidades de la inteligencia artificial en la ingeniería de software. *Technology Inside*, 12(1). <https://cpic-sistemas.or.cr/revista/index.php/technology-inside/article/view/110/107>
- Yavich, R. (2025). Will the Use of AI Undermine Students Independent Thinking? *MDPI*, 15(6). <https://doi.org/10.3390/educsci15060669>