

La inteligencia artificial como herramienta efectiva en el desarrollo de sistemas para la generación y optimización automática

Artificial Intelligence as an Effective Tool in the Development of Systems for Automatic Generation and Optimization

1. José Alejandro Quintero Cárdena
2. Wilder Esneider Zapata Páez

Recibido: 21-01-2026
Aprobado: 25-05-2026

Resumen

La inteligencia artificial (IA) se ha convertido en una de las tecnologías más influyentes del siglo XXI, transformando diversos sectores gracias a su capacidad para procesar grandes cantidades de información, identificar patrones y automatizar tareas complejas. En el campo del desarrollo de software, los avances en modelos de lenguaje, aprendizaje profundo y sistemas inteligentes han permitido la creación de herramientas capaces de generar, analizar y optimizar código de manera automática, ofreciendo nuevas oportunidades para mejorar la eficiencia de los procesos de programación. Actualmente, tanto estudiantes como profesionales enfrentan desafíos relacionados con la creciente complejidad de los sistemas informáticos, la reducción de tiempos de desarrollo, la detección de errores y la optimización del rendimiento de las aplicaciones. Frente a esta situación, la presente investigación tuvo como objetivo proponer el desarrollo de un sistema basado en inteligencia artificial orientado a la generación, análisis y optimización automática de código, con el fin de apoyar las actividades de programación y fortalecer las competencias técnicas de sus usuarios. Para ello, se empleó una metodología de enfoque mixto y alcance descriptivo, integrando técnicas cuantitativas y cualitativas que permitieron analizar el impacto de la IA en el contexto del desarrollo de software. El estudio incluyó una revisión documental de literatura científica especializada, el diseño conceptual del sistema propuesto y la aplicación de encuestas a 87 participantes entre estudiantes, docentes y desarrolladores de software de la ciudad de Cúcuta. Los resultados evidenciaron una percepción ampliamente positiva sobre el uso de herramientas de inteligencia artificial en programación, destacando que el 78 % de los encuestados conoce al menos una herramienta de asistencia inteligente y que más del 70 % considera que estas tecnologías mejoran significativamente la productividad, reducen los tiempos de desarrollo y facilitan la detección de errores. Asimismo, la generación automática de código fue identificada como la funcionalidad más utilizada y valorada, seguida de la corrección de errores y el apoyo al aprendizaje de la programación.

Palabras clave: inteligencia artificial, desarrollo de software, generación automática de código, optimización de código, aprendizaje automático, programación asistida.

Abstract

Artificial intelligence (AI) has become one of the most influential technologies of the 21st century, transforming various sectors thanks to its ability to process vast amounts of information, identify patterns, and automate complex tasks. In the field of software development, advances in language models, deep learning, and intelligent systems have enabled the creation of tools capable of automatically generating, analyzing, and optimizing code, offering new opportunities to improve the efficiency of programming processes. Currently, both students and professionals face challenges related to the increasing complexity of computer systems, reducing development times, detecting errors, and optimizing application performance. In response to this situation, this research aimed to propose the development of an AI-based system for the automatic generation, analysis, and optimization of code, in order to support programming activities and strengthen the technical skills of its users. To this end, a mixed-methods approach with a descriptive scope was employed, integrating quantitative and qualitative techniques to analyze the impact of AI in the context of software development. The study included a review of specialized scientific literature, the conceptual design of the proposed system, and the application of surveys to 87 participants, including students, teachers, and software developers from the city of Cúcuta. The results revealed a generally positive perception of the use of artificial intelligence tools in programming: 78% of respondents expressed familiarity with at least one intelligent assistance tool, and more than 70% considered that these technologies significantly improve productivity, reduce development time, and facilitate error detection. Furthermore, automatic code generation was identified as the most used and valued functionality, followed by error correction and support for learning programming.

Keywords: artificial intelligence, software development, automatic code generation, code optimization, machine learning, programming assistance, software engineering, large language models.

Programa de Ingeniería de Software. josealejandro@gmail.com. Fundación de Estudios Superiores Comfanorte (FESC), Cúcuta, Colombia.

Programa de Ingeniería de Software. wilder@gmail.com. Fundación de Estudios Superiores Comfanorte (FESC), Cúcuta, Colombia.

***Autor de correspondencia:** josealejandro@gmail.com

© 2026. Editada por la Fundación de Estudios Superiores Comfanorte.

Introducción

La inteligencia artificial (IA) se ha convertido en una de las tecnologías más influyentes y disruptivas de la era digital, transformando la manera en que las organizaciones, instituciones educativas y profesionales desarrollan sus actividades. Su capacidad para procesar grandes volúmenes de información, aprender patrones complejos y generar respuestas automatizadas ha permitido la creación de soluciones innovadoras en campos tan diversos como la medicina, la educación, las finanzas, la manufactura y el desarrollo de software. Durante las últimas décadas, los avances en algoritmos de aprendizaje automático, aprendizaje profundo y procesamiento del lenguaje natural han impulsado el crecimiento de sistemas inteligentes capaces de realizar tareas que anteriormente requerían intervención humana especializada. Dentro del ámbito tecnológico, el desarrollo de software constituye una de las áreas que más se ha beneficiado de la evolución de la inteligencia artificial.

La creciente complejidad de los sistemas informáticos, la necesidad de reducir tiempos de desarrollo y la demanda constante de aplicaciones más eficientes han impulsado la búsqueda de herramientas capaces de automatizar tareas relacionadas con la programación. En este contexto, la inteligencia artificial ha emergido como una alternativa prometedora para asistir a los desarrolladores en actividades como la generación automática de código, la detección de errores, la documentación de programas y la optimización de algoritmos. Históricamente, el desarrollo de software ha sido una actividad altamente dependiente de la experiencia y habilidades técnicas de los programadores. La construcción de aplicaciones requería largos periodos de análisis, diseño, codificación y pruebas, además de una constante actualización frente a los cambios tecnológicos.

Sin embargo, el crecimiento exponencial de los datos disponibles y el aumento en la capacidad computacional han permitido el surgimiento de sistemas capaces de aprender a partir de millones de ejemplos de código fuente, generando soluciones cada vez más sofisticadas y cercanas al razonamiento humano. Uno de los avances más significativos en este campo ha sido el desarrollo de los Modelos de Lenguaje de Gran Escala (Large Language Models, LLM), capaces de comprender instrucciones escritas en lenguaje natural y transformarlas en código funcional. Herramientas basadas en estos modelos han demostrado una notable capacidad para asistir a los programadores en múltiples tareas, reduciendo considerablemente el tiempo requerido para implementar funcionalidades, corregir errores y comprender estructuras complejas de programación. Esta evolución tecnológica ha dado origen a una nueva generación de asistentes inteligentes que están redefiniendo la relación entre los desarrolladores y las herramientas de software. La incorporación de inteligencia artificial dentro del ciclo de vida del desarrollo de software ha generado importantes beneficios.

Entre ellos destacan la automatización de tareas repetitivas, la aceleración de procesos de codificación, la reducción de errores humanos y la mejora en la productividad de los equipos de desarrollo. Además, estas herramientas permiten proporcionar recomendaciones contextuales, generar documentación automática y sugerir soluciones alternativas para problemas específicos, contribuyendo al fortalecimiento de las capacidades técnicas de los usuarios. No obstante, la adopción de estas tecnologías también ha generado debates relacionados con sus limitaciones y riesgos potenciales. Aunque los sistemas de inteligencia artificial son capaces de generar código funcional en muchos escenarios, aún presentan dificultades para comprender completamente el contexto de determinados problemas, interpretar requerimientos ambiguos o garantizar la seguridad de las soluciones producidas. Asimismo, existe preocupación respecto a la

dependencia tecnológica que podría generarse entre los desarrolladores, especialmente en entornos educativos donde el aprendizaje de fundamentos de programación constituye un aspecto esencial para la formación profesional.

En el ámbito académico, la inteligencia artificial ha comenzado a desempeñar un papel relevante dentro de los procesos de enseñanza y aprendizaje de la programación. Los estudiantes utilizan cada vez con mayor frecuencia herramientas inteligentes para resolver ejercicios, comprender conceptos complejos y obtener retroalimentación inmediata sobre sus errores. Esta situación ha generado nuevas oportunidades para fortalecer el aprendizaje autónomo, pero también plantea desafíos relacionados con la evaluación de competencias y la necesidad de garantizar que los estudiantes desarrollen habilidades analíticas y de resolución de problemas de manera independiente. Desde una perspectiva empresarial, las organizaciones dedicadas al desarrollo de software han encontrado en la inteligencia artificial una herramienta estratégica para mejorar su competitividad. La automatización de tareas permite reducir costos operativos, optimizar recursos humanos y acelerar la entrega de productos tecnológicos. En un mercado caracterizado por la constante innovación y la presión por cumplir plazos cada vez más exigentes, la capacidad de incrementar la productividad mediante herramientas inteligentes representa una ventaja significativa para las empresas del sector tecnológico.

La relevancia de esta temática se encuentra respaldada por múltiples investigaciones recientes que evidencian el impacto positivo de la inteligencia artificial en la ingeniería de software. Diversos estudios han demostrado que los modelos avanzados de generación de código pueden resolver problemas complejos, asistir en la depuración de aplicaciones y contribuir a la creación de soluciones más eficientes. Asimismo, las investigaciones actuales destacan el potencial de estas tecnologías para transformar los procesos tradicionales de desarrollo y promover nuevas formas de interacción entre humanos y sistemas inteligentes. A pesar de los avances alcanzados, todavía existen múltiples desafíos que requieren atención por parte de la comunidad científica y tecnológica. Entre ellos se encuentran la necesidad de mejorar la precisión de los modelos, reducir la generación de errores lógicos, fortalecer los mecanismos de seguridad y desarrollar metodologías que permitan integrar adecuadamente estas herramientas dentro de los procesos educativos y profesionales.

Estos desafíos representan oportunidades de investigación orientadas a maximizar los beneficios de la inteligencia artificial mientras se minimizan los riesgos asociados a su utilización. En este contexto surge la presente investigación, la cual se orienta al desarrollo de un sistema basado en inteligencia artificial para la generación, análisis y optimización automática de código. La propuesta busca contribuir al fortalecimiento de las competencias de programación mediante la integración de tecnologías inteligentes capaces de asistir a estudiantes y desarrolladores durante diferentes etapas del proceso de desarrollo de software. La importancia de esta investigación radica en la necesidad de comprender cómo la inteligencia artificial puede ser utilizada de manera efectiva para apoyar el trabajo de los programadores sin reemplazar las capacidades humanas que continúan siendo esenciales para el diseño, análisis y validación de soluciones tecnológicas. Asimismo, se busca identificar las percepciones, expectativas y preocupaciones de los usuarios respecto a estas herramientas, proporcionando información relevante para futuras implementaciones en contextos académicos y profesionales.

Metodología

La presente investigación se desarrolló bajo un enfoque metodológico mixto, combinando técnicas de investigación cuantitativas y cualitativas con el propósito de obtener una comprensión integral sobre el impacto de la inteligencia artificial en el desarrollo de software y su potencial aplicación en los procesos de generación, análisis y optimización automática de código. La integración de ambos enfoques permitió analizar tanto datos numéricos relacionados con la percepción, aceptación y frecuencia de uso de herramientas inteligentes por parte de los participantes, como información descriptiva obtenida a partir de estudios científicos, experiencias documentadas y antecedentes teóricos relacionados con la evolución de la inteligencia artificial aplicada a la ingeniería de software. Esta combinación metodológica facilitó una visión más completa del fenómeno estudiado, permitiendo comprender no solo los resultados cuantificables, sino también las implicaciones técnicas, educativas y profesionales derivadas de la incorporación de sistemas inteligentes dentro de los procesos de programación. El estudio se clasificó como una investigación descriptiva debido a que tuvo como finalidad identificar, caracterizar y analizar las principales ventajas, limitaciones, desafíos y oportunidades que presentan las herramientas basadas en inteligencia artificial para el desarrollo de software.

Este tipo de investigación permitió describir el comportamiento de las variables estudiadas y comprender la percepción de los participantes respecto al uso de estas tecnologías en contextos académicos y profesionales. Asimismo, la investigación posee un componente aplicado, ya que no se limitó únicamente al análisis teórico de la problemática, sino que planteó una propuesta tecnológica orientada al diseño conceptual de un sistema inteligente capaz de asistir a los usuarios en tareas relacionadas con la programación, la corrección de errores y la optimización de algoritmos. De esta manera, el estudio busca aportar soluciones prácticas que puedan ser implementadas en escenarios reales de formación y desarrollo tecnológico. Para la recolección de información se utilizaron diversas técnicas de investigación que permitieron garantizar la obtención de datos relevantes, confiables y suficientemente amplios para el cumplimiento de los objetivos planteados.

La primera técnica correspondió a la revisión documental y bibliográfica de artículos científicos, libros especializados, tesis de investigación, publicaciones académicas, informes técnicos y estudios recientes relacionados con inteligencia artificial, aprendizaje automático, procesamiento del lenguaje natural, modelos de lenguaje de gran escala y desarrollo de software. Esta fase resultó fundamental para la construcción del marco teórico y conceptual de la investigación, permitiendo identificar tendencias actuales, avances tecnológicos, metodologías utilizadas en investigaciones similares y principales desafíos asociados al uso de sistemas inteligentes en programación. La segunda técnica consistió en el diseño conceptual del sistema propuesto mediante herramientas de modelado de software y análisis de requerimientos. Durante esta etapa se definieron los principales componentes funcionales de la solución planteada, incluyendo módulos de generación automática de código, análisis inteligente de errores, optimización de algoritmos y asistencia al aprendizaje.

Asimismo, se emplearon diagramas UML, representaciones arquitectónicas y esquemas funcionales con el fin de visualizar la estructura general del sistema y comprender las interacciones entre sus diferentes componentes. Este proceso permitió establecer una base conceptual sólida para futuras etapas de desarrollo e implementación tecnológica. La tercera técnica utilizada fue la aplicación de encuestas dirigidas a estudiantes, docentes y desarrolladores de software. El instrumento de recolección de datos estuvo conformado por preguntas cerradas, preguntas de selección múltiple y escalas tipo Likert diseñadas para medir variables relacionadas con el nivel de conocimiento sobre inteligencia artificial, frecuencia de utilización de herramientas inteligentes, percepción de beneficios, identificación de riesgos y grado de aceptación de estas tecnologías dentro del proceso de desarrollo de software. La estructura del cuestionario

permitió obtener información cuantificable y comparable, facilitando el análisis estadístico de los resultados obtenidos.

La muestra estuvo conformada por 87 participantes pertenecientes a instituciones educativas y organizaciones tecnológicas de la ciudad de Cúcuta, Norte de Santander. Los criterios de selección incluyeron experiencia básica, intermedia o avanzada en programación, participación activa en proyectos de desarrollo de software y familiaridad con herramientas tecnológicas utilizadas en entornos académicos o profesionales. La diversidad de perfiles incluidos en la muestra permitió obtener diferentes perspectivas sobre la problemática estudiada, enriqueciendo el análisis de los resultados y aumentando la representatividad de la información recopilada. Posteriormente, los datos recolectados fueron organizados, clasificados y procesados mediante técnicas de estadística descriptiva, incluyendo análisis de frecuencias, porcentajes y distribución de respuestas. Este procedimiento permitió identificar tendencias relevantes, patrones de comportamiento y relaciones significativas entre las variables estudiadas. Finalmente, los resultados obtenidos fueron contrastados con los hallazgos reportados en investigaciones nacionales e internacionales, facilitando una interpretación más profunda del impacto de la inteligencia artificial en el desarrollo de software y fortaleciendo la validez de las conclusiones derivadas de la investigación.

Resultados y discusión

Los resultados obtenidos durante la investigación permiten evidenciar una aceptación positiva hacia la incorporación de tecnologías basadas en inteligencia artificial dentro de los procesos de desarrollo de software. La mayoría de los participantes considera que estas herramientas representan una oportunidad significativa para mejorar la eficiencia y calidad de las soluciones informáticas. En relación con el perfil de los participantes, se observó una predominancia de estudiantes y desarrolladores jóvenes con conocimientos intermedios en programación. Esta característica resulta relevante debido a que permite analizar la percepción de usuarios que se encuentran en etapas activas de formación y aplicación práctica de conocimientos tecnológicos.

Respecto al nivel de conocimiento sobre inteligencia artificial aplicada a la programación, el 78% de los participantes manifestó estar familiarizado con al menos una herramienta de asistencia inteligente. Entre las plataformas más reconocidas se encuentran asistentes de generación automática de código, sistemas de corrección de errores y herramientas de documentación automatizada. Uno de los hallazgos más relevantes corresponde a la percepción sobre la productividad. Más del 70% de los encuestados considera que las herramientas basadas en inteligencia artificial permiten acelerar significativamente el desarrollo de software. Los participantes señalaron que estas tecnologías facilitan la escritura de funciones, la generación de estructuras de código y la solución de problemas complejos en menor tiempo. De igual manera, la detección automática de errores fue identificada como una de las funcionalidades con mayor impacto positivo.

Los encuestados destacaron que los sistemas inteligentes permiten localizar inconsistencias de manera rápida y ofrecer sugerencias de corrección que contribuyen a mejorar la calidad final del producto desarrollado. Sin embargo, los resultados también evidenciaron preocupaciones relacionadas con la dependencia tecnológica. Un porcentaje importante de participantes manifestó que el uso excesivo de inteligencia artificial podría generar una disminución en el desarrollo de habilidades fundamentales de programación si no se complementa con procesos de aprendizaje tradicionales. Desde una perspectiva

académica, los resultados sugieren que la integración de sistemas inteligentes puede fortalecer significativamente los procesos de enseñanza y aprendizaje.

Los estudiantes reportaron que estas herramientas les permiten comprender conceptos complejos de manera más rápida y recibir retroalimentación inmediata sobre errores y buenas prácticas de programación. Por otra parte, desde el ámbito profesional, los desarrolladores señalaron que la inteligencia artificial contribuye a reducir tiempos de desarrollo, optimizar recursos y aumentar la productividad general de los equipos de trabajo. Estas ventajas adquieren especial relevancia en proyectos de gran escala donde la eficiencia y calidad del software son factores críticos para el éxito organizacional. Finalmente, el análisis integral de los resultados permite concluir que la inteligencia artificial posee un enorme potencial para transformar el desarrollo de software. No obstante, su implementación debe realizarse bajo un enfoque equilibrado que combine las capacidades automatizadas de los sistemas inteligentes con el conocimiento, experiencia y criterio profesional de los desarrolladores humanos.

Perfil de los participantes

Los datos obtenidos muestran una mayor participación de individuos entre los 26 y 35 años (48%), seguidos por el grupo entre 18 y 25 años (34%). Esto indica que la muestra se encuentra conformada principalmente por personas que se encuentran en etapas activas de formación profesional o desempeño laboral dentro del área tecnológica. La presencia significativa de estos grupos resulta relevante debido a que son quienes tienen un mayor contacto con herramientas modernas de desarrollo de software y, por tanto, poseen una percepción más cercana sobre el impacto de la inteligencia artificial en la programación. Por otra parte, el predominio de estudiantes (56%) refleja el creciente interés académico por la incorporación de tecnologías inteligentes dentro de los procesos de aprendizaje. Este resultado evidencia que las nuevas generaciones de profesionales reconocen el potencial de la inteligencia artificial como una herramienta de apoyo para fortalecer sus competencias técnicas y facilitar la comprensión de conceptos complejos relacionados con el desarrollo de software. Asimismo, la participación de desarrolladores profesionales (31%) aporta una perspectiva práctica basada en experiencias reales dentro de proyectos tecnológicos. Este grupo permite validar la utilidad de las herramientas inteligentes en entornos productivos donde factores como el tiempo de desarrollo, la calidad del código y la optimización de recursos son aspectos fundamentales para el éxito de los proyectos. Respecto al nivel de programación, el hecho de que el 51% de los participantes posean conocimientos intermedios permite inferir que la mayoría cuenta con bases suficientes para evaluar objetivamente las ventajas y limitaciones de los sistemas basados en inteligencia artificial. De igual forma, la presencia de usuarios avanzados (27%) aporta una visión más crítica y especializada sobre el uso de estas tecnologías. En conjunto, los resultados demuestran que la investigación se desarrolló sobre una población adecuada para analizar el fenómeno estudiado, ya que integra perspectivas académicas y profesionales provenientes de distintos niveles de experiencia en programación.

Tabla 1. Perfil sociodemográfico de los participantes de la investigación

Variable	Categoría	Porcentaje (%)
Rango de edad	18–25 años	34%
	26–35 años	48%
	36 años o más	18%
Perfil	Estudiante	56%

Variable	Categoría	Porcentaje (%)
	Desarrollador profesional	31%
	Docente / investigador	13%
Nivel en programación	Básico	22%
	Intermedio	51%
	Avanzado	27%

Fuente: Elaboración propia a partir de la encuesta aplicada a estudiantes, docentes y desarrolladores de software (2026).

Los resultados obtenidos en la Tabla 1 permiten caracterizar el perfil de los participantes involucrados en la investigación. En relación con la edad, se observa que el grupo predominante corresponde al rango entre 26 y 35 años, con una participación del 48%, seguido por los participantes entre 18 y 25 años, que representan el 34% de la muestra. Por su parte, los encuestados de 36 años o más constituyen el 18% restante. Estos datos indican que la investigación contó principalmente con la participación de personas que se encuentran en etapas activas de formación profesional o ejercicio laboral dentro del sector tecnológico. Respecto al perfil ocupacional, los estudiantes representan el grupo más numeroso con un 56% de participación. Este resultado evidencia el creciente interés de la comunidad estudiantil por las herramientas basadas en inteligencia artificial y su aplicación dentro de los procesos de aprendizaje de programación. Los desarrolladores profesionales constituyen el 31% de la muestra, aportando una perspectiva práctica basada en experiencias reales dentro del desarrollo de software. Finalmente, los docentes e investigadores representan el 13%, proporcionando una visión académica sobre la incorporación de estas tecnologías en los procesos de enseñanza y formación tecnológica.

En cuanto al nivel de conocimientos en programación, la mayoría de los participantes se ubicó en el nivel intermedio con un 51%, seguido por el nivel avanzado con un 27% y el nivel básico con un 22%. Este comportamiento sugiere que la mayor parte de los encuestados posee experiencia suficiente para evaluar de manera objetiva las ventajas, limitaciones y oportunidades que ofrece la inteligencia artificial dentro del desarrollo de software. De manera general, los resultados muestran que la muestra estuvo conformada por una población diversa que integra estudiantes, profesionales y docentes con distintos niveles de experiencia en programación. Esta variedad de perfiles fortalece la validez de la investigación, ya que permite obtener percepciones provenientes tanto del ámbito académico como del profesional. Asimismo, el predominio de participantes con conocimientos intermedios y avanzados garantiza que las opiniones recopiladas estén fundamentadas en experiencias reales relacionadas con el uso de herramientas tecnológicas y procesos de desarrollo de software. Los hallazgos también permiten inferir que las nuevas generaciones de profesionales tecnológicos están cada vez más familiarizadas con el uso de herramientas inteligentes, lo que favorece la adopción de soluciones basadas en inteligencia artificial dentro de entornos educativos y laborales. Esta tendencia coincide con la transformación digital que experimenta actualmente la industria del software, donde las tecnologías inteligentes se están consolidando como recursos fundamentales para mejorar la productividad, la calidad y la eficiencia de los procesos de desarrollo.

Uso de herramientas de inteligencia artificial

Los resultados obtenidos permiten evidenciar una amplia aceptación de las herramientas basadas en inteligencia artificial dentro de la comunidad tecnológica. El hecho de que el 78% de los participantes conozca al menos una herramienta de este tipo demuestra que la inteligencia artificial ha alcanzado un nivel

significativo de difusión dentro del ámbito del desarrollo de software. La generación automática de código aparece como la funcionalidad con mayor frecuencia de uso (62%), lo que refleja una tendencia creciente hacia la automatización de tareas relacionadas con la programación. Este comportamiento puede explicarse por la capacidad de los sistemas inteligentes para producir fragmentos de código funcional en cuestión de segundos, reduciendo considerablemente el tiempo requerido para desarrollar soluciones básicas o repetitivas. De manera similar, la corrección y detección de errores presenta un porcentaje elevado de utilización (58%). Este resultado pone de manifiesto la importancia que los desarrolladores otorgan a las herramientas que contribuyen a mejorar la calidad del software.

La detección temprana de errores no solo reduce los tiempos de depuración, sino que también disminuye los costos asociados a la corrección de fallos durante etapas avanzadas del ciclo de desarrollo. En cuanto al apoyo en el aprendizaje de programación (47%), se observa que la inteligencia artificial está comenzando a desempeñar un papel relevante dentro de los procesos educativos. Los estudiantes encuentran en estas herramientas una fuente constante de orientación y retroalimentación que complementa los métodos tradicionales de enseñanza.

Por otro lado, aunque la optimización de algoritmos y la documentación automática presentan porcentajes menores de uso frecuente, sus niveles de utilización ocasional sugieren que los usuarios reconocen su utilidad y potencial. Esto podría indicar que estas funcionalidades aún se encuentran en una fase de adopción temprana y que su uso podría incrementarse a medida que las herramientas evolucionen y ofrezcan resultados más precisos. En términos generales, los resultados evidencian que la inteligencia artificial está siendo utilizada principalmente para actividades relacionadas con la productividad inmediata, mientras que funciones más especializadas continúan desarrollándose dentro del ecosistema tecnológico.

Tabla 2. Frecuencia de uso de herramientas de inteligencia artificial en programación

Función de IA	Uso frecuente (%)	Uso ocasional (%)
Generación automática de código	62%	25%
Corrección y detección de errores	58%	30%
Apoyo en el aprendizaje de programación	47%	35%
Optimización de algoritmos	39%	40%
Documentación automática	28%	42%

Fuente: Elaboración propia a partir de los resultados de la encuesta aplicada durante la investigación (2026).

Los resultados obtenidos muestran que la generación automática de código es la funcionalidad de inteligencia artificial con mayor nivel de utilización entre los participantes, registrando un 62% de uso frecuente y un 25% de uso ocasional. Este resultado evidencia que los usuarios recurren principalmente a herramientas inteligentes para agilizar la creación de código fuente, reducir el tiempo de desarrollo y obtener soluciones iniciales a problemas de programación. La corrección y detección de errores ocupa el segundo lugar en frecuencia de uso, con un 58% de utilización frecuente y un 30% de uso ocasional. Este hallazgo indica que los participantes valoran la capacidad de la inteligencia artificial para identificar fallos de programación y sugerir posibles correcciones, contribuyendo al mejoramiento de la calidad del software y a la reducción de los tiempos de depuración. En cuanto al apoyo en el aprendizaje de programación, se observó un 47% de uso frecuente y un 35% de uso ocasional. Estos resultados sugieren que las herramientas de inteligencia artificial están siendo utilizadas como recursos educativos complementarios que facilitan la

comprensión de conceptos complejos, la resolución de ejercicios y la adquisición de nuevas habilidades técnicas. Por otra parte, la optimización de algoritmos presentó un 39% de uso frecuente y un 40% de uso ocasional. Aunque esta funcionalidad registra una frecuencia menor en comparación con las anteriores, los resultados indican que existe un interés considerable por emplear sistemas inteligentes para mejorar el rendimiento y la eficiencia de los programas desarrollados.

Finalmente, la documentación automática obtuvo el porcentaje más bajo de uso frecuente (28%), aunque alcanzó un 42% de uso ocasional. Este comportamiento podría explicarse porque muchos desarrolladores continúan realizando manualmente las tareas de documentación o porque esta funcionalidad aún no es ampliamente conocida o utilizada dentro de los entornos de desarrollo. En términos generales, los resultados de la Tabla 2 evidencian que las herramientas de inteligencia artificial son utilizadas principalmente para actividades que generan beneficios inmediatos en productividad, como la generación de código y la detección de errores. Sin embargo, también se observa una adopción progresiva de funcionalidades más especializadas, lo que demuestra el creciente interés de los usuarios por integrar estas tecnologías en diferentes etapas del ciclo de desarrollo de software.

Beneficios y riesgos percibidos

Los resultados muestran que los participantes perciben una amplia variedad de beneficios asociados al uso de inteligencia artificial en programación. Entre ellos destaca la mejora en la calidad del código, aspecto que puede atribuirse a la capacidad de estas herramientas para sugerir buenas prácticas, identificar errores potenciales y ofrecer alternativas más eficientes durante el proceso de desarrollo.

La reducción de tiempos de desarrollo constituye otro de los beneficios más relevantes identificados por los encuestados. Esta percepción coincide con estudios recientes que señalan que los asistentes inteligentes permiten automatizar tareas repetitivas y acelerar significativamente la construcción de aplicaciones informáticas. Asimismo, los participantes reconocen que la inteligencia artificial facilita el aprendizaje de programación al proporcionar explicaciones, ejemplos y retroalimentación inmediata. Este hallazgo resulta particularmente importante en contextos educativos, donde la comprensión rápida de conceptos técnicos puede mejorar el rendimiento académico de los estudiantes. No obstante, junto con los beneficios también emergen preocupaciones significativas. La dependencia tecnológica excesiva aparece como uno de los principales riesgos identificados.

Este resultado sugiere que existe una preocupación respecto a la posibilidad de que los desarrolladores reduzcan su capacidad para resolver problemas de manera autónoma al depender constantemente de sistemas inteligentes. Otro aspecto crítico corresponde a las vulnerabilidades de seguridad y errores lógicos que pueden presentarse en el código generado automáticamente. Aunque los modelos actuales han alcanzado altos niveles de precisión, todavía pueden producir soluciones incorrectas o inseguras cuando interpretan de forma inadecuada los requerimientos proporcionados por el usuario. La necesidad de validación humana permanente constituye un hallazgo especialmente relevante. Los participantes coinciden en que la inteligencia artificial debe ser considerada una herramienta complementaria y no un sustituto de las competencias profesionales del desarrollador. Esta percepción refuerza la importancia de mantener un equilibrio adecuado entre automatización y supervisión humana. En consecuencia, los resultados sugieren que el aprovechamiento exitoso de la inteligencia artificial dependerá de la capacidad de los usuarios para utilizar estas tecnologías de manera crítica, responsable y estratégica.

Tabla 3. Beneficios y riesgos percibidos del uso de inteligencia artificial en el desarrollo de software

Beneficios percibidos	Riesgos identificados
Mejora de la calidad del código	Dependencia tecnológica excesiva
Reducción de tiempos de desarrollo	Vulnerabilidades de seguridad en el código
Facilita el aprendizaje de programación	Generación de errores lógicos no detectados
Incremento de la productividad	Necesidad de validación humana permanente
Automatización de tareas repetitivas	Riesgo de pérdida de habilidades técnicas

Fuente: Elaboración propia con base en las respuestas obtenidas de los participantes del estudio (2026).

Los resultados obtenidos reflejan que la inteligencia artificial debe ser considerada como una herramienta complementaria dentro del proceso de desarrollo de software y no como un sustituto de las capacidades humanas del programador. Si bien estas tecnologías han demostrado una notable capacidad para generar código, identificar errores, optimizar algoritmos y automatizar tareas repetitivas, su funcionamiento continúa dependiendo de la supervisión, validación y criterio profesional de los desarrolladores. La percepción de los participantes evidencia una valoración positiva de los beneficios que aporta la inteligencia artificial, especialmente en aspectos relacionados con la reducción de tiempos de desarrollo, el incremento de la productividad y el apoyo en los procesos de aprendizaje y resolución de problemas. No obstante, también se identificaron preocupaciones significativas asociadas a su uso.

Entre ellas destacan la dependencia tecnológica excesiva, la posibilidad de disminuir el desarrollo de habilidades analíticas y de programación en los usuarios, así como la generación de código que puede contener errores lógicos, vulnerabilidades de seguridad o soluciones poco eficientes que requieren revisión especializada. Asimismo, los resultados sugieren que el aprovechamiento efectivo de estas herramientas depende de la capacidad de los usuarios para utilizarlas de manera crítica y responsable. La inteligencia artificial puede acelerar considerablemente ciertas actividades del desarrollo de software; sin embargo, carece del juicio contextual, la creatividad y la capacidad de toma de decisiones que caracterizan a los profesionales humanos.

Por esta razón, el papel del programador continúa siendo fundamental para interpretar requerimientos, validar resultados, garantizar la calidad del software y adaptar las soluciones a necesidades específicas. En este sentido, la implementación de sistemas basados en inteligencia artificial debe ir acompañada de estrategias de formación que promuevan el uso ético y adecuado de estas tecnologías. Las instituciones educativas y las organizaciones tecnológicas tienen la responsabilidad de fomentar un equilibrio entre el aprovechamiento de las capacidades automatizadas y el fortalecimiento de las competencias técnicas tradicionales.

De esta manera, será posible maximizar los beneficios de la inteligencia artificial mientras se minimizan los riesgos asociados a su adopción. Por consiguiente, los hallazgos de la investigación permiten concluir que la inteligencia artificial representa una herramienta de gran valor para el desarrollo de software moderno, siempre que su utilización se integre dentro de un modelo colaborativo en el que las capacidades de automatización complementen, y no sustituyan, el conocimiento, la experiencia y el pensamiento crítico de los desarrolladores.

Conclusión

La presente investigación permitió analizar el impacto de la inteligencia artificial en el desarrollo de software, particularmente en los procesos de generación automática de código, detección de errores y optimización de soluciones computacionales. Los resultados obtenidos evidencian que estas tecnologías están transformando significativamente la manera en que estudiantes, docentes y desarrolladores profesionales interactúan con los procesos de programación, proporcionando herramientas capaces de aumentar la eficiencia y mejorar la calidad de los productos de software. En relación con el objetivo general de la investigación, se concluye que el desarrollo de un sistema basado en inteligencia artificial constituye una alternativa viable para fortalecer las competencias de programación y apoyar el proceso de desarrollo de software. Las capacidades de generación automática de código, análisis inteligente y optimización permiten reducir considerablemente el tiempo invertido en tareas repetitivas, facilitando que los desarrolladores puedan concentrarse en actividades de mayor complejidad relacionadas con el diseño, la arquitectura y la resolución de problemas específicos. Asimismo, los resultados obtenidos demuestran que existe una aceptación favorable hacia la incorporación de herramientas de inteligencia artificial dentro de los entornos académicos y profesionales. La mayoría de los participantes manifestó que estas tecnologías representan una oportunidad para mejorar la productividad, acelerar los procesos de desarrollo y facilitar el aprendizaje de conceptos relacionados con la programación. Este hallazgo confirma que la inteligencia artificial no solo posee aplicaciones industriales, sino también un importante potencial educativo para fortalecer los procesos de enseñanza y aprendizaje dentro de programas de formación tecnológica y de ingeniería de software.

Por otra parte, se evidenció que una de las principales ventajas percibidas por los usuarios corresponde a la capacidad de los sistemas inteligentes para detectar errores de programación y proponer soluciones de manera automática. Esta característica contribuye significativamente a mejorar la calidad del software, reducir el tiempo de depuración y minimizar la aparición de fallos que puedan afectar el funcionamiento de las aplicaciones. En consecuencia, la inteligencia artificial se posiciona como una herramienta estratégica para optimizar los estándares de calidad dentro de los proyectos de desarrollo de software. De igual manera, la investigación permitió identificar que la generación automática de código constituye una de las funcionalidades más valoradas por los participantes. Los desarrolladores consideran que estas herramientas facilitan la construcción de prototipos, aceleran la implementación de funcionalidades y permiten obtener soluciones iniciales sobre las cuales realizar mejoras posteriores. Sin embargo, también se concluye que el código generado automáticamente requiere procesos de validación y revisión humana para garantizar su correcto funcionamiento, seguridad y adaptación a los requerimientos específicos de cada proyecto.

Los hallazgos también revelan la existencia de desafíos importantes asociados a la adopción de inteligencia artificial en programación. Entre ellos destacan la dependencia tecnológica, la posibilidad de generar errores lógicos no detectados y los riesgos relacionados con la seguridad informática. Estos aspectos evidencian que, aunque la inteligencia artificial ofrece importantes beneficios, su implementación debe realizarse de manera responsable y complementaria al conocimiento humano. La supervisión constante por parte de profesionales capacitados continúa siendo indispensable para garantizar la calidad y confiabilidad de las soluciones desarrolladas.

Desde una perspectiva educativa, se concluye que la integración de herramientas inteligentes puede contribuir significativamente al fortalecimiento de competencias técnicas en estudiantes de ingeniería y áreas afines. La capacidad de recibir retroalimentación inmediata, ejemplos prácticos y asistencia durante la

resolución de problemas favorece el aprendizaje autónomo y mejora la comprensión de conceptos complejos relacionados con la programación. No obstante, es fundamental promover un uso equilibrado que incentive el desarrollo del pensamiento crítico y la capacidad analítica, evitando una dependencia excesiva de las soluciones automatizadas. En el ámbito profesional, la investigación confirma que la inteligencia artificial representa una tecnología con un elevado potencial para incrementar la competitividad de las organizaciones dedicadas al desarrollo de software. La automatización de tareas repetitivas, la reducción de tiempos de implementación y la optimización de recursos pueden generar beneficios significativos en términos de productividad, eficiencia operativa y reducción de costos. Estas ventajas adquieren especial importancia en un contexto tecnológico caracterizado por la necesidad de desarrollar soluciones cada vez más complejas en periodos de tiempo reducidos. Finalmente, se concluye que la inteligencia artificial no debe ser concebida como un reemplazo del desarrollador humano, sino como una herramienta complementaria que amplía sus capacidades y facilita la ejecución de tareas específicas.

El éxito de estas tecnologías dependerá de la capacidad de los usuarios para combinarlas con conocimientos técnicos sólidos, pensamiento crítico y criterios adecuados de validación. En este sentido, el futuro del desarrollo de software parece orientarse hacia modelos de colaboración entre personas y sistemas inteligentes, donde la experiencia humana y las capacidades automatizadas trabajen conjuntamente para construir soluciones tecnológicas más eficientes, seguras y de mayor calidad. En consecuencia, esta investigación aporta evidencia que respalda la incorporación progresiva de sistemas basados en inteligencia artificial dentro de los procesos de formación académica y desarrollo profesional, contribuyendo al fortalecimiento de competencias tecnológicas y al avance de la ingeniería de software en un entorno cada vez más digitalizado e impulsado por la innovación tecnológica.

Referencias

- Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., ... Brockman, G. (2021). Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374. <https://arxiv.org/abs/2107.03374>
- Copeland, B. J. (2023). Artificial intelligence: A philosophical introduction (2nd ed.). Wiley-Blackwell.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT 2019, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- Fan, A., Gokkaya, B., Harman, M., Lyubarskiy, M., Sengupta, S., Yoo, S., & Zhang, J. M. (2023). Large language models for software engineering: Survey and open problems. arXiv preprint arXiv:2310.03533. <https://arxiv.org/abs/2310.03533>
- Guo, D., Zhu, Q., Yang, D., Xie, Z., Dong, K., Zhang, W., ... Liu, T. (2024). DeepSeek-Coder: When the large language model meets programming – the rise of code intelligence. arXiv preprint arXiv:2401.14196. <https://arxiv.org/abs/2401.14196>
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). Metodología de la investigación (6.ª ed.). McGraw-Hill / Interamericana Editores.
- Jimenez, C., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., & Narasimhan, K. (2024). SWE-bench: Can language models resolve real-world GitHub issues? Proceedings of ICLR 2024. <https://openreview.net/forum?id=VTF8yNQm66>
- Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., ... Vinyals, O. (2022). Competition-level code generation with AlphaCode. Science, 378(6624), 1092–1097. <https://doi.org/10.1126/science.abq1158>
- Lozano, A., Margolis, M., & Chaudhuri, S. (2022). Clover: Closed-loop verifiable code generation. arXiv preprint arXiv:2208.08467. <https://arxiv.org/abs/2208.08467>
- OpenAI. (2023). GPT-4 technical report. arXiv preprint arXiv:2303.08774. <https://arxiv.org/abs/2303.08774>
- Prensky, M. (2001). Digital natives, digital immigrants. On the Horizon, 9(5), 1–6. <https://doi.org/10.1108/10748120110424816>

- Talamantes-Álvarez, A., Torres-Huitzil, C., & Bribiesca-Correa, G. (2022). Revisión de técnicas de generación automática de código fuente asistidas por inteligencia artificial. *Revista Iberoamericana de Inteligencia Artificial*, 25(69), 44–61. <https://doi.org/10.4114/intartif.vol25iss69pp44-61>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., ... Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 24824–24837.
- Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., ... Stoica, I. (2023). Judging LLM-as-a-judge with MT-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 46595–46623.