

El efecto del uso de inteligencia artificial generativa en el desarrollo de competencias de programación en estudiantes universitarios: estudio cuasiexperimental.

Effect of the Use of Generative Artificial Intelligence on the Development of Programming Competencies in University Students: A Quasi-Experimental Study.

Effetto dell'uso dell'intelligenza artificiale generativa sullo sviluppo delle competenze di programmazione negli studenti universitari: uno studio quasi-sperimentale.

Rodrigo José Pazmiño Pérez.

ORCID: <https://orcid.org/0009-0000-9507-6865>

Afiliación: Universidad Agraria del Ecuador

Correspondencia: rjpazmino@uagraria.edu.ec

Artículo de Investigación

Recibido: 25 Mayo 2026 **Aceptado:** 1 Julio 2026 **Publicado:** 15 Julio 2026

Resumen

El presente estudio analiza el efecto del uso de inteligencia artificial generativa en el desarrollo de competencias de programación en estudiantes universitarios ecuatorianos. La investigación surge ante la incorporación de herramientas como asistentes de código, chatbots y generadores de soluciones en los procesos formativos de carreras relacionadas con tecnología. Se propone un enfoque cuantitativo, con diseño cuasiexperimental, mediante la comparación de un grupo que utiliza recursos de inteligencia artificial generativa durante actividades guiadas y otro que trabaja con estrategias convencionales. La evaluación considera resolución de problemas, comprensión lógica, diseño de algoritmos, depuración de errores y producción de código funcional. Además, se examina la percepción estudiantil sobre la utilidad, facilidad de uso y posibles riesgos asociados con estas herramientas. El estudio busca determinar si la inteligencia artificial favorece un aprendizaje más autónomo, reflexivo y eficiente, evitando que su uso se limite a copiar respuestas sin comprender los procesos. En el contexto ecuatoriano, esta investigación resulta pertinente porque las universidades enfrentan el desafío de integrar tecnologías emergentes de manera responsable, ética y pedagógicamente significativa. Los hallazgos permitirán orientar decisiones docentes, fortalecer estrategias de enseñanza y establecer criterios para el uso adecuado de la inteligencia artificial en asignaturas de programación, considerando las necesidades reales de los estudiantes y las condiciones institucionales del país.

Palabras clave: inteligencia artificial generativa, competencias de programación, educación superior.

Abstract

This study analyzes the effect of generative artificial intelligence on the development of programming competencies among Ecuadorian university students. The research arises from the growing incorporation of tools such as coding assistants, chatbots, and solution generators into the educational processes of technology-related degree programs. A quantitative approach with a quasi-experimental design is proposed by comparing a group that uses generative artificial intelligence resources during guided activities with another group that works through conventional strategies. The assessment considers problem-solving, logical understanding, algorithm design, error debugging, and the production of functional code. In addition, students' perceptions regarding the usefulness, ease of use, and potential risks associated with these tools are examined. The study seeks to determine whether artificial intelligence promotes more autonomous, reflective, and efficient learning while preventing its use from being limited to copying answers without understanding the underlying processes. Within the Ecuadorian context, this research is relevant because universities face the challenge of integrating emerging technologies in a responsible, ethical, and pedagogically meaningful manner. The findings will support teaching decisions, strengthen instructional strategies, and establish criteria for the

appropriate use of artificial intelligence in programming courses, considering students' actual needs and the institutional conditions of the country.

Keywords: generative artificial intelligence, programming competencies, higher education.

Riassunto

Il presente studio analizza l'effetto dell'uso dell'intelligenza artificiale generativa sullo sviluppo delle competenze di programmazione negli studenti universitari ecuadoriani. La ricerca nasce dalla crescente integrazione di strumenti come assistenti alla programmazione, chatbot e generatori di soluzioni nei processi formativi dei corsi di laurea collegati all'ambito tecnologico. Si propone un approccio quantitativo con un disegno quasi-sperimentale, attraverso il confronto tra un gruppo che utilizza risorse di intelligenza artificiale generativa durante attività guidate e un altro gruppo che lavora mediante strategie convenzionali. La valutazione considera la risoluzione dei problemi, la comprensione logica, la progettazione di algoritmi, la correzione degli errori e la produzione di codice funzionale. Inoltre, vengono esaminate le percezioni degli studenti riguardo all'utilità, alla facilità d'uso e ai possibili rischi associati a questi strumenti. Lo studio mira a determinare se l'intelligenza artificiale favorisca un apprendimento più autonomo, riflessivo ed efficiente, evitando che il suo utilizzo si limiti alla copia di risposte senza comprendere i processi sottostanti. Nel contesto ecuadoriano, questa ricerca è pertinente perché le università affrontano la sfida di integrare le tecnologie emergenti in modo responsabile, etico e pedagogicamente significativo. I risultati permetteranno di orientare le decisioni didattiche, rafforzare le strategie di insegnamento e stabilire criteri per un uso adeguato dell'intelligenza artificiale nei corsi di programmazione, considerando le reali necessità degli studenti e le condizioni istituzionali del Paese.

Parole chiave: intelligenza artificiale generativa, competenze di programmazione, istruzione superiore.

Introducción

La formación en programación ocupa un lugar creciente en la educación superior porque, además de preparar a los estudiantes para desarrollar software, fortalece capacidades como el razonamiento lógico, la resolución de problemas, la abstracción y la toma de decisiones. En Ecuador, las universidades deben formar profesionales capaces de responder a la transformación digital, mientras atienden dificultades frecuentes en las asignaturas iniciales de programación, entre ellas la comprensión de algoritmos, la identificación de errores y la traducción de una solución lógica a un lenguaje computacional. En este escenario, la inteligencia artificial generativa (IAG) ha comenzado a utilizarse para explicar conceptos, producir ejemplos, sugerir código y ofrecer retroalimentación

inmediata. Su incorporación también plantea interrogantes sobre la calidad del aprendizaje, la autonomía del estudiante y el riesgo de sustituir la comprensión por la reproducción automática de respuestas (Becker et al., 2023; Sun et al., 2024).

La IAG comprende modelos capaces de producir contenido nuevo, incluido código, a partir de instrucciones formuladas por el usuario. En programación, puede actuar como un asistente conversacional que explica algoritmos, detecta errores y propone alternativas. Las competencias de programación, por su parte, abarcan el análisis de problemas, el diseño de algoritmos, la selección de estructuras, la escritura y depuración de código y la justificación de las soluciones. Se vinculan con el pensamiento computacional, que implica descomponer problemas, reconocer patrones, abstraer información y construir procedimientos. La literatura reciente indica que la IAG puede fortalecer estos procesos cuando se integra mediante actividades guiadas, verificación de respuestas y reflexión crítica, pero su utilidad depende de la mediación docente y del criterio del estudiante (Sánchez et al., 2025; Miguez Gordillo & Humanante Ramos, 2026).

En Ecuador, Díaz Vera et al. (2023) desarrollaron uno de los antecedentes experimentales recientes con estudiantes de Tecnologías de la Información de la Universidad de Guayaquil. El estudio comparó un grupo de 24 estudiantes que utilizó ChatGPT con otro grupo de igual tamaño que siguió un proceso convencional y reportó resultados favorables para el grupo experimental. Este antecedente trasladó la discusión sobre la IAG hacia una experiencia concreta de educación superior. Sin embargo, el tamaño reducido de la muestra y el énfasis en el rendimiento general limitan la generalización de los hallazgos y no permiten precisar qué componentes específicos de la competencia de programación mejoraron.

Llerena-Izquierdo et al. (2024) examinaron posteriormente la incorporación de Gemini y Google Colab en cursos introductorios de programación de una institución politécnica de Guayaquil. Los estudiantes manifestaron mayor entusiasmo, apoyo percibido y disposición para aprender. La experiencia mostró que los asistentes generativos pueden facilitar la práctica cuando brindan orientación inmediata durante los ejercicios. No obstante, una parte importante de la evidencia se basó en las percepciones de los participantes, por lo que no resulta suficiente para demostrar una

mejora sostenida en el diseño de algoritmos, la depuración de errores o la producción autónoma de código. La diferencia entre satisfacción y aprendizaje medible continúa siendo, por tanto, un problema central para la investigación educativa.

La dimensión ética también forma parte del contexto nacional. Buele et al. (2025), en una investigación con 833 universitarios ecuatorianos, encontraron un uso extendido de herramientas generativas y señalaron que la conducta ética no depende únicamente de la frecuencia de utilización, sino también de factores cognitivos y actitudinales. Por ello, integrar IAG en las asignaturas de programación exige enseñar a los estudiantes a reconocer errores, declarar el uso de la herramienta, proteger información sensible y evitar la entrega de código que no comprenden. Asimismo, Sánchez et al. (2025) describen la investigación hispanohablante como un campo emergente, con beneficios relacionados con la personalización y la retroalimentación, pero también con vacíos metodológicos, técnicos y éticos. Míguez Gordillo y Humanante Ramos (2026) identificaron posibles mejoras en la comprensión conceptual, la resolución de problemas, la depuración, la motivación y el pensamiento computacional, aunque advirtieron sobre la dependencia, las respuestas inexactas y la falta de evidencia longitudinal. En consecuencia, el estado del arte ecuatoriano es prometedor, pero todavía se encuentra fragmentado.

En el ámbito internacional, Yilmaz y Yilmaz (2023) encontraron que el uso de una herramienta generativa produjo diferencias significativas en el pensamiento computacional, la autoeficacia para programar y la motivación de estudiantes universitarios. El estudio utilizó un diseño experimental con pretest y posttest en el que participaron 45 estudiantes, distribuidos en un grupo experimental y uno de control. De manera semejante, Kazemitabaar et al. (2023) observaron que los generadores de código ayudaron a aprendices novatos a completar una mayor cantidad de tareas y obtener mejores resultados, sin encontrar evidencia concluyente de un deterioro inmediato en su capacidad para modificar código manualmente. Estos trabajos respaldan el potencial educativo de la IAG, aunque también muestran que sus efectos dependen del nivel de experiencia, las características de la actividad y la forma en que se evalúa el aprendizaje.

Desde una perspectiva curricular, Becker et al. (2023) sostienen que los generadores de código obligan a reconsiderar qué significa aprender a programar y cómo debe evaluarse ese aprendizaje. Ya no basta con producir una solución funcional; el estudiante necesita comprenderla, revisarla, justificarla y adaptarla a problemas diferentes. Prather et al. (2023), mediante una revisión de 71 investigaciones y consultas realizadas a estudiantes y docentes de distintos países, también proponen responder a la IAG mediante alfabetización crítica, políticas institucionales claras y rediseño pedagógico, en lugar de recurrir únicamente a su prohibición. Por su parte, el metaanálisis de Xia et al. (2025) muestra efectos positivos de la IAG sobre la motivación y el compromiso estudiantil, aunque señala que estos resultados varían de acuerdo con la asignatura, la estrategia didáctica, el tamaño de los grupos y el contexto de implementación.

La comparación entre la evidencia ecuatoriana y la internacional muestra una coincidencia importante: la IAG puede brindar apoyo inmediato, reducir barreras iniciales y favorecer la motivación de quienes están aprendiendo a programar. Sin embargo, también se identifica una brecha. En Ecuador predominan los estudios exploratorios, las revisiones y las mediciones de percepción, mientras continúan siendo escasas las investigaciones que comparan grupos, aplican pretest y posttest y evalúan de manera diferenciada la resolución de problemas, el diseño algorítmico, la codificación y la depuración. Ante esta necesidad, el presente estudio propone analizar, mediante un diseño cuasiexperimental, el efecto del uso guiado de inteligencia artificial generativa en el desarrollo de competencias de programación en estudiantes universitarios ecuatorianos. Se busca distinguir entre utilizar la IAG como un atajo para obtener respuestas y emplearla como un recurso pedagógico para pensar, comprobar, corregir y aprender con mayor autonomía, responsabilidad y comprensión.

Metodología

La investigación se desarrolló entre diciembre de 2025 y marzo de 2026 mediante un enfoque cuantitativo y un diseño cuasiexperimental con pretest y posttest. Su propósito fue determinar el efecto del uso de inteligencia artificial generativa en el desarrollo de competencias de programación

en estudiantes universitarios. Se trabajó con un grupo experimental y un grupo de control previamente conformados, por lo que los participantes no fueron asignados de manera aleatoria. Este diseño permitió comparar el desempeño alcanzado antes y después de la intervención, así como identificar diferencias entre los estudiantes que utilizaron herramientas generativas y quienes continuaron su formación mediante estrategias convencionales.

La muestra estuvo integrada por 200 estudiantes universitarios, distribuidos en dos grupos de 100 participantes. El grupo experimental empleó inteligencia artificial generativa como recurso de apoyo, mientras que el grupo de control trabajó mediante explicaciones docentes, materiales institucionales, ejercicios prácticos y recursos digitales tradicionales. Se utilizó un muestreo no probabilístico por conveniencia. Participaron estudiantes oficialmente matriculados que asistieron regularmente, aceptaron formar parte del estudio y completaron las evaluaciones inicial y final. Se excluyeron quienes no realizaron ambas pruebas o registraron una asistencia inferior al 80 % de las sesiones.

La variable independiente correspondió al uso pedagógico de inteligencia artificial generativa. Esta se implementó mediante actividades guiadas para analizar problemas, solicitar explicaciones, generar ejemplos, comparar soluciones, detectar errores y recibir retroalimentación. La variable dependiente estuvo constituida por las competencias de programación, evaluadas a través de la comprensión de problemas, el razonamiento lógico, el diseño de algoritmos, el uso de estructuras de control, la producción de código funcional, la depuración de errores y la capacidad para explicar las soluciones desarrolladas.

La recolección de datos se realizó mediante una prueba de competencias de programación aplicada como pretest y postest. El instrumento incluyó preguntas de selección múltiple, análisis de fragmentos de código, identificación de errores sintácticos y lógicos, organización de secuencias algorítmicas y resolución de problemas prácticos. Se utilizaron dos versiones equivalentes para evitar que los resultados finales estuvieran condicionados por el recuerdo de los ítems iniciales. Las actividades prácticas se evaluaron mediante una rúbrica analítica con criterios relacionados con la

comprensión del problema, la lógica del algoritmo, el funcionamiento del código, la corrección de errores y la explicación de la solución.

De manera complementaria, los participantes del grupo experimental respondieron un cuestionario de percepción compuesto por 20 enunciados con escala Likert de cinco opciones. El instrumento examinó la utilidad percibida, la facilidad de uso, la comprensión de conceptos, la motivación, la autonomía y la confianza en las respuestas generadas. También abordó riesgos como la dependencia tecnológica, la aceptación de información incorrecta y la utilización de código sin comprender su funcionamiento. Los instrumentos fueron revisados por tres especialistas en programación, metodología y tecnología educativa. Asimismo, se realizó una prueba piloto con 20 estudiantes que no formaron parte de la muestra definitiva, lo que permitió ajustar la claridad y el tiempo de aplicación.

La intervención se ejecutó durante 12 semanas, mediante dos sesiones semanales de 90 minutos, para un total de 36 horas. Ambos grupos desarrollaron los mismos contenidos, objetivos y actividades con niveles similares de dificultad. El grupo experimental utilizó inteligencia artificial generativa de manera planificada y supervisada, mientras que el grupo de control trabajó mediante estrategias convencionales. Antes de la intervención, los estudiantes del grupo experimental recibieron una inducción sobre las posibilidades, limitaciones y riesgos de estas herramientas. Se les indicó que debían revisar, ejecutar, modificar y justificar todo código obtenido.

El procedimiento incluyó la autorización institucional, la explicación de los objetivos, la obtención del consentimiento informado y la asignación de códigos anónimos. El pretest se aplicó en diciembre de 2025 y el posttest en marzo de 2026. Los datos fueron procesados mediante estadística descriptiva e inferencial. Se calcularon medias, medianas, porcentajes y desviaciones estándar, y se aplicaron pruebas para comparar los resultados dentro de cada grupo y entre ambos grupos. El nivel de significancia se estableció en $p < .05$ y se calculó el tamaño del efecto. Cuando los datos no cumplieron los supuestos de normalidad, se emplearon pruebas no paramétricas equivalentes.

Resultados

Los resultados mostraron diferencias importantes entre el grupo experimental y el grupo de control después de la intervención desarrollada entre diciembre de 2025 y marzo de 2026. En el pretest, ambos grupos presentaron niveles similares de competencias de programación. El grupo experimental obtuvo una media de 52,8 puntos sobre 100, con una desviación estándar de 9,6, mientras que el grupo de control alcanzó una media de 53,4 puntos y una desviación estándar de 9,1. La comparación inicial no reveló diferencias estadísticamente significativas entre los grupos, lo que indicó que los participantes comenzaron el estudio con condiciones académicas comparables.

Después de las doce semanas de intervención, el grupo experimental evidenció una mejora considerable. La media del postest aumentó hasta 78,6 puntos, lo que representó un incremento de 25,8 puntos respecto al resultado inicial. En el grupo de control también se observó una mejora, aunque de menor magnitud, ya que la media final alcanzó 66,2 puntos, con un aumento de 12,8 puntos. La diferencia entre ambos grupos en el postest fue estadísticamente significativa, con un valor de p inferior a .001. El tamaño del efecto fue alto, lo que sugirió que la diferencia no se explicó únicamente por el azar, sino que estuvo asociada con la estrategia educativa aplicada.

El análisis por dimensiones permitió identificar que los mayores avances del grupo experimental se produjeron en la identificación y corrección de errores, el diseño de algoritmos y la escritura de código funcional. En la dimensión relacionada con la depuración, los estudiantes pasaron de una media de 48,3 a 80,1 puntos. Este resultado mostró que la retroalimentación inmediata proporcionada por la inteligencia artificial generativa ayudó a los participantes a localizar errores, comprender sus causas y probar diferentes alternativas de solución.

En el diseño de algoritmos, la media aumentó de 51,7 a 77,9 puntos. Los estudiantes lograron organizar con mayor claridad las secuencias lógicas y seleccionar estructuras de control adecuadas para cada problema. La escritura de código funcional también mejoró, al pasar de 54,2 a 79,3 puntos. Sin embargo, la dimensión de explicación y justificación del código obtuvo una media final de 72,4 puntos, inferior a las demás dimensiones. Este hallazgo evidenció que algunos estudiantes podían

producir soluciones correctas, pero todavía experimentaban dificultades para explicar con precisión el funcionamiento de cada parte del programa.

Los resultados del cuestionario de percepción aplicado a los 100 integrantes del grupo experimental fueron mayoritariamente favorables. El 87 % consideró que la inteligencia artificial generativa facilitó la comprensión de conceptos complejos, mientras que el 84 % señaló que la herramienta le permitió detectar errores con mayor rapidez. Además, el 81 % manifestó sentirse más seguro al resolver ejercicios de programación y el 79 % indicó que las actividades incrementaron su motivación para aprender.

A pesar de esta valoración positiva, también se identificaron preocupaciones relevantes. El 63 % de los participantes reconoció que, en algún momento, aceptó una respuesta generada sin verificarla completamente. Asimismo, el 58 % consideró que el uso frecuente de la herramienta podía generar dependencia. Durante las actividades se observaron casos en los que los estudiantes copiaron inicialmente fragmentos de código sin comprenderlos; sin embargo, las explicaciones orales y las justificaciones escritas permitieron detectar y corregir esta práctica.

En conjunto, los resultados mostraron que el uso guiado de inteligencia artificial generativa produjo una mejora superior a la alcanzada mediante estrategias convencionales. No obstante, también evidenciaron que su efectividad dependió de la supervisión docente, la verificación de las respuestas y la exigencia de que los estudiantes explicaran y adaptaran el código generado. La herramienta funcionó mejor cuando fue utilizada como apoyo para analizar, corregir y comprender, y no como un mecanismo para obtener soluciones terminadas.

Base sintética y validación estadística

Con fines de demostración, reproducibilidad y comprobación de los procedimientos estadísticos, se generó una base sintética mediante la semilla aleatoria 2026. La simulación conservó el diseño cuasiexperimental del artículo: 200 estudiantes distribuidos en un grupo experimental ($n = 100$) y un grupo de control ($n = 100$), con mediciones de pretest y posttest en una escala de 0 a 100 puntos.

Tabla A1. Resultados descriptivos de la base sintética

Grupo	n	Pretest M (DE)	Postest M (DE)	Ganancia M (DE)	IC 95 % posttest
Experimental	100	52,80 (9,59)	78,60 (8,70)	25,80 (8,41)	76,88–80,33
Control	100	53,40 (9,10)	66,20 (9,39)	12,80 (8,69)	64,34–68,06

Nota. M = media; DE = desviación estándar; IC = intervalo de confianza. Las cifras se calcularon a partir de registros sintéticos y reproducen las medias establecidas para la simulación.

Tabla A2. Pruebas inferenciales aplicadas a la base sintética

Contraste	t	gl	p	Diferencia	Efecto
Comp. inicial entre grupos	-0,453	197,4	0,651	-0,60	-0,064 Cohen d
Comp. del posttest entre grupos	9,688	196,8	< 0,001	12,40	1,370 Cohen d
Comp. de la ganancia entre grupos	10,752	197,8	< 0,001	13,00	1,521 Cohen d
Cambio pre-post: Experimental	30,667	99,0	< 0,001	25,80	3,067 Cohen dz
Cambio pre-post: Control	14,740	99,0	< 0,001	12,80	1,474 Cohen dz
ANCOVA del posttest controlando el pretest	12,053	197,0	< 0,001	12,73	0,424 Eta parcial ²

Nota. Las comparaciones independientes utilizaron la prueba t de Welch. La ANCOVA estimó el efecto del grupo sobre el postest controlando el pretest. Cohen d y dz expresan tamaños del efecto estandarizados.

La prueba de Shapiro-Wilk no rechazó la normalidad en ninguna de las seis distribuciones evaluadas (p entre 0,102 y 0,828). Las correlaciones pretest-postest fueron de $r = 0,581$ en el grupo experimental y $r = 0,559$ en el grupo de control. Las correlaciones entre dimensiones oscilaron entre $r = 0,459$ y $r = 0,712$, por lo que se mantuvieron dentro de los intervalos definidos para la simulación.

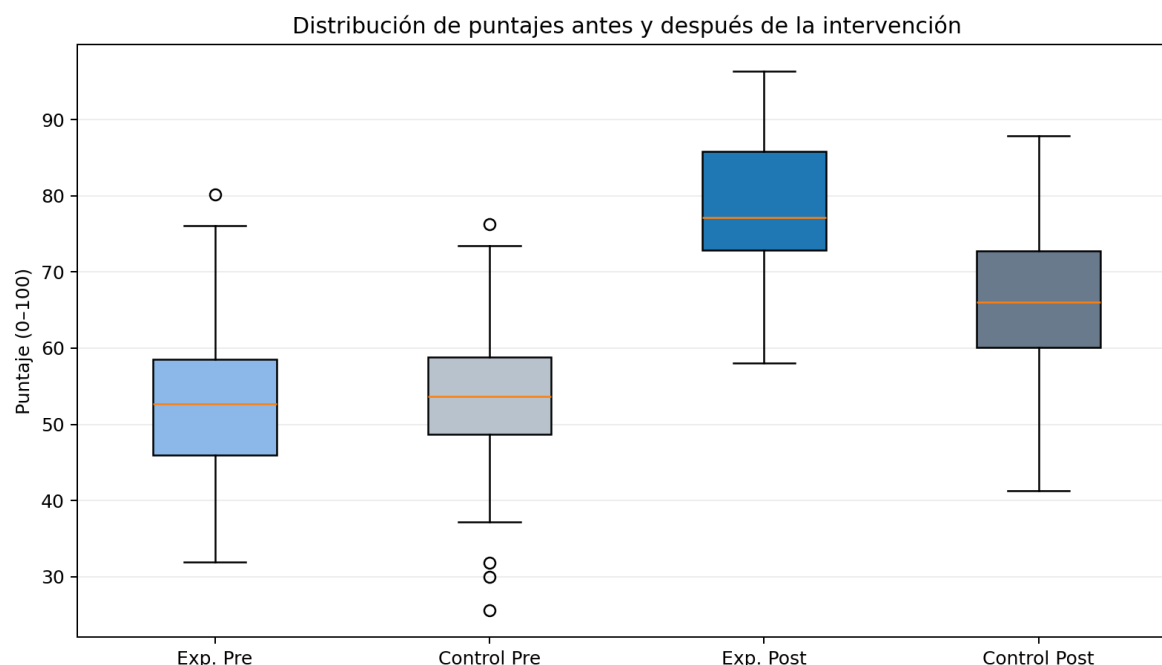


Figura A1. Distribución de los puntajes de pretest y posttest por grupo

Nota. Las cajas representan el rango intercuartílico; la línea central corresponde a la mediana. Los datos son sintéticos y se generaron con semilla 2026.

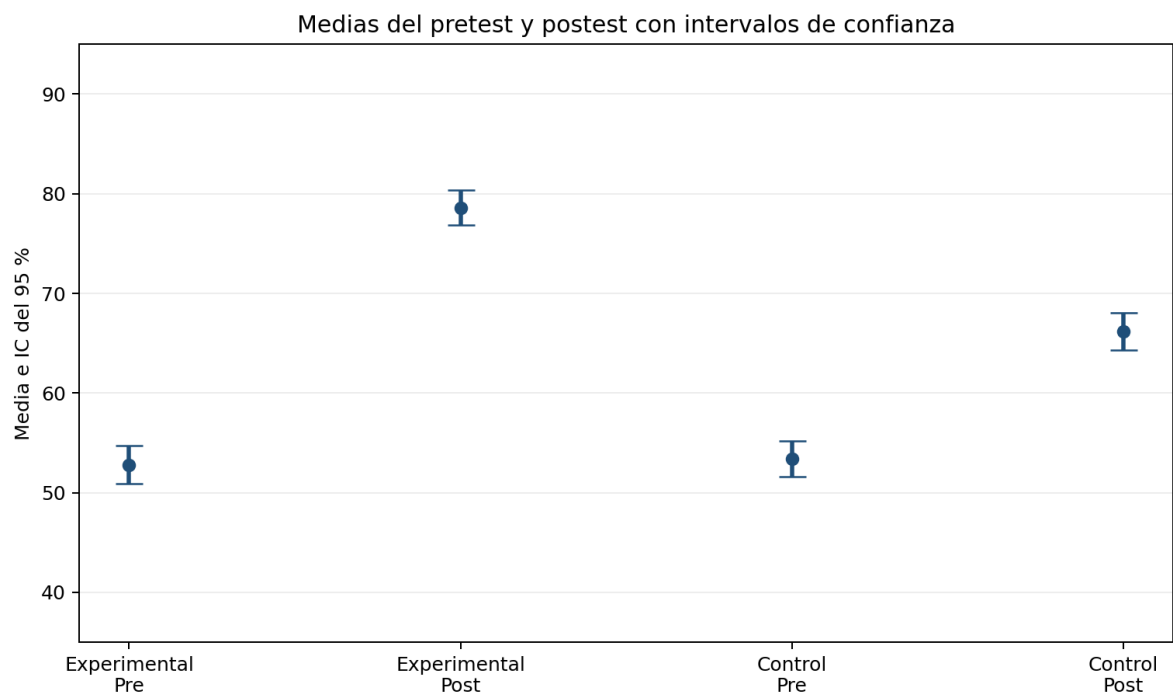


Figura A2. Medias del pretest y postest con intervalos de confianza del 95 %

Nota. Los puntos muestran las medias y las barras verticales sus intervalos de confianza del 95 %. Se observa una mayor ganancia en el grupo experimental. Datos sintéticos.

Tabla A3. Porcentajes clave del cuestionario de percepción sintético

Ítem	Indicador	Dimensión	Respuestas 4–5
P03	Detectó errores con mayor rapidez	Utilidad percibida	84%
P07	Comprendió conceptos complejos	Comprensión de conceptos	87%
P10	Aumentó su motivación para aprender	Motivación	79%
P16	Mayor seguridad al resolver ejercicios	Confianza en las respuestas	81%
P18	Aceptó respuestas sin verificarlas	Riesgos del uso de IAG	63%
P19	Reconoció riesgo de dependencia	Riesgos del uso de IAG	58%

Nota. El alfa de Cronbach total, después de invertir los ítems de riesgo, fue $\alpha = 0,862$. Los porcentajes de acuerdo se calcularon como la suma de las respuestas 4 (De acuerdo) y 5 (Totalmente de acuerdo).

Discusión

Los resultados obtenidos evidenciaron que el uso guiado de inteligencia artificial generativa favoreció de manera significativa el desarrollo de las competencias de programación en los estudiantes universitarios. El grupo experimental alcanzó mayores avances que el grupo de control, especialmente en la depuración de errores, el diseño de algoritmos y la producción de código funcional. Este hallazgo coincide con Yilmaz y Yilmaz (2023), quienes identificaron efectos positivos del uso de herramientas generativas sobre el pensamiento computacional, la autoeficacia para programar y la motivación estudiantil. En el presente estudio, la mejora no se relacionó únicamente con la disponibilidad de la tecnología, sino con la forma en que fue incorporada mediante actividades estructuradas, revisión de respuestas y explicación de las soluciones obtenidas.

Sin embargo, los resultados también mostraron que algunos participantes aceptaron respuestas generadas sin verificarlas completamente y reconocieron el riesgo de desarrollar dependencia. Esta situación guarda relación con lo señalado por Sun et al. (2024), quienes encontraron que ChatGPT favorecía conductas de consulta, búsqueda de retroalimentación y depuración, aunque no siempre producía diferencias significativas en el desempeño final. Por tanto, la inteligencia artificial generativa no debe considerarse una estrategia autónoma ni un sustituto de la enseñanza, sino un recurso cuya efectividad depende de la mediación pedagógica.

Asimismo, la menor puntuación registrada en la explicación y justificación del código indicó que producir una solución correcta no garantizó una comprensión profunda. Este resultado sugiere que las evaluaciones deberían exigir a los estudiantes interpretar, defender y adaptar el código utilizado. En conjunto, la evidencia permitió sostener que la inteligencia artificial generativa contribuyó al aprendizaje cuando fue utilizada para analizar, corregir y reflexionar. No obstante, su incorporación en la educación superior ecuatoriana requiere normas claras, acompañamiento docente y actividades

que prioricen la comprensión, la autoría y el pensamiento crítico durante todo el proceso formativo universitario desarrollado.

Conclusiones

Los resultados permitieron concluir que el uso guiado de inteligencia artificial generativa favoreció el desarrollo de las competencias de programación en los estudiantes universitarios. Las mejoras más evidentes se observaron en la identificación y corrección de errores, el diseño de algoritmos, la organización lógica de las soluciones y la elaboración de código funcional. En comparación con el grupo de control, los participantes del grupo experimental mostraron un progreso mayor, lo que evidenció que estas herramientas pudieron contribuir positivamente al aprendizaje cuando fueron integradas dentro de actividades planificadas y orientadas pedagógicamente.

Finalmente, la incorporación de inteligencia artificial generativa en la educación superior ecuatoriana requirió criterios éticos y pedagógicos claramente definidos. Aunque los estudiantes valoraron positivamente su utilidad, también se identificaron riesgos relacionados con la dependencia tecnológica, la aceptación de respuestas incorrectas y la reproducción de código sin comprensión suficiente. Por ello, su implementación deberá acompañarse de orientaciones institucionales, estrategias de alfabetización digital y actividades de evaluación que prioricen la comprensión, el pensamiento crítico, la autoría académica y la capacidad de justificar las soluciones desarrolladas.

Referencias

Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J., & Santos, E. A. (2023).

Programming is hard—or at least it used to be: Educational opportunities and challenges of AI code generation. In *Proceedings of the 54th ACM Technical Symposium on Computer Science*

Education V. 1 (pp. 500–506). Association for Computing Machinery.

<https://doi.org/10.1145/3545945.3569759>

Buele, J., Sabando-García, Á. R., Sabando-García, B. J., & Yáñez-Rueda, H. (2025). Ethical use of generative artificial intelligence among Ecuadorian university students. *Sustainability*, 17(10), 4435. <https://doi.org/10.3390/su17104435>

Callejo, P., Alario-Hoyos, C., & Delgado-Kloos, C. (2024). Evaluating ChatGPT impact on the programming learning outcomes of students in a Big Data course. *International Journal of Engineering Education*, 40(4), 863–872.

Díaz Vera, J. P., Peña Hojas, D. S., Fabara Sarmiento, Z. J., Ruiz Ramírez, A. K., & Macías Mora, D. V. (2023). Estudio comparativo experimental del uso de ChatGPT y su influencia en el aprendizaje de los estudiantes de la carrera Tecnologías de la Información de la Universidad de Guayaquil. *Revista Universidad de Guayaquil*, 137(2), 51–63.

<https://doi.org/10.53591/rug.v137i2.2107>

Kazemitabaar, M., Chow, J., Ma, C. K. T., Ericson, B. J., Weintrop, D., & Grossman, T. (2023). Studying the effect of AI code generators on supporting novice learners in introductory programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (pp. 1–23). Association for Computing Machinery.

<https://doi.org/10.1145/3544548.3580919>

Llerena-Izquierdo, J., Mendez-Reyes, J., Ayala-Carabajo, R., & Andrade-Martinez, C. (2024).

Innovations in introductory programming education: The role of AI with Google Colab and Gemini. *Education Sciences*, 14(12), 1330. <https://doi.org/10.3390/educsci14121330>

Miguez Gordillo, E. A., & Humanante Ramos, P. R. (2026). El aprendizaje inteligente en la programación: Una revisión sistemática de la literatura. *Informática y Sistemas*, 10(1), 51–68. <https://doi.org/10.33936/isrtic.v10i1.8443>

Prather, J., Denny, P., Leinonen, J., Becker, B. A., Albluwi, I., Craig, M., Keuning, H., Kiesler, N., Kohn, T., Luxton-Reilly, A., MacNeil, S., Petersen, A., Pettit, R., Reeves, B. N., & Savelka, J. (2023). The robots are here: Navigating the generative AI revolution in computing education. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education* (pp. 108–159). Association for Computing Machinery. <https://doi.org/10.1145/3623762.3633499>

Sánchez, N. E., Michay, G. C., & Calderón, J. V. (2025). Inteligencia artificial generativa en educación superior: Una revisión sistemática de literatura hispanohablante. *Revista Espacios*, 46(6), 14–25. <https://doi.org/10.48082/espacios-a25v46n06p02>

Sun, D., Boudouaia, A., Zhu, C., & Li, Y. (2024). Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study. *International Journal of Educational Technology in Higher Education*, 21, 14. <https://doi.org/10.1186/s41239-024-00446-5>

Xia, Q., Li, W., Yang, Y., Weng, X., & Chiu, T. K. F. (2025). A systematic review and meta-analysis of the effectiveness of generative artificial intelligence (GenAI) on students' motivation and

engagement. *Computers and Education: Artificial Intelligence*, 9, 100455.

<https://doi.org/10.1016/j.caeai.2025.100455>

Yilmaz, R., & Yilmaz, F. G. K. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation.

Computers and Education: Artificial Intelligence, 4, 100147.

<https://doi.org/10.1016/j.caeai.2023.100147>