

# La IA y las aulas

*Retos y oportunidades en la formación de programadores*

DOI: 10.29236/sistemas.n172a8

## Resumen

La inteligencia artificial (IA) ha generado un profundo impacto en diversas industrias, y la educación no es la excepción. Este artículo examina cómo la IA generativa está transformando el aula, especialmente en la enseñanza de la programación. Muchas universidades alrededor del mundo ya han implementado con éxito herramientas de IA generativa en sus cursos, como el uso de asistentes de código y el desarrollo de nuevas habilidades como la ingeniería de prompts y la capacidad de realizar pruebas de software. Si bien estas herramientas aceleran el aprendizaje, también plantean retos significativos, como la posible dependencia tecnológica de los estudiantes y la necesidad de que los profesores implementen estrategias pedagógicas que fomenten el pensamiento crítico.

Finalmente, se reflexiona sobre la importancia de que los estudiantes no solo dominen habilidades de bajo nivel, sino que también aprendan a utilizar estas herramientas de IA de manera ética y responsable, en sintonía con las exigencias de una industria que ya espera que los nuevos profesionales sean capaces de aprovechar estas tecnologías emergentes.

## Palabras claves

Desarrollo de software, Inteligencia artificial, Nuevas habilidades, IA generativa, Tecnología Educativa

## Introducción

La inteligencia artificial es un concepto complejo que a menudo genera dos sentimientos interrelacionados en las personas: incompreensión sobre su funcionamiento y temor por su impacto, imaginado no solo como inmenso, sino también acompañado de la posibilidad de que esta llegue a superar las capacidades humanas. Estos sentimientos de incertidumbre y miedo han sido comunes en cada revolución tecnológica. Sin embargo, a lo largo de la historia, hemos aprendido a adaptarnos y prosperar frente a estos cambios.

La IA actual se identifica como el corazón de la 4ª revolución industrial. Quizá la diferencia fundamental de esta revolución con las tres anteriores, es que las tecnologías emergentes de aquel entonces reemplazaban aspectos físicos de los humanos, relacionados con trabajo repetitivo, peligroso o que requería mucha precisión; sin embargo, lo que está sucediendo hoy evidencia un potencial reemplazo de las capacidades de razonamiento, aprendizaje y creatividad, que considerábamos únicas en nosotros. Qué tanto nos impactará es aún una pregunta abierta que respondemos un poco más cada día que pasa.

Ahora, es importante anotar que la inteligencia artificial no es un concepto nuevo. Desde las primeras

ideas propuestas por John Von Neumann y Alan Turing en los años 40, cuando propusieron la posibilidad de que las máquinas fueran capaces de aprender y razonar como los humanos, los científicos han soñado con imitar la inteligencia humana. Durante los años 60, la IA vivió su primer auge, con esperanzas de crear sistemas capaces de aprender. Conceptos como las redes neuronales nacieron en esta época. Sin embargo, la falta de poder computacional, la baja disponibilidad de datos en formato digital y unas altas expectativas no alcanzadas, llevaron a un estancamiento en los 70s y 80s, conocido como el "Invierno de la IA". Fue apenas en los últimos 20 años que, con el aumento en las capacidades de procesamiento, tanto en procesadores generales como en el uso de tarjetas gráficas para acelerar el cómputo de los modelos, y la generación y almacenamiento masivo de datos que permiten entrenar los modelos complejos, se han podido desarrollar nuevas iniciativas de analíticas de datos, como Big Data, Deep Learning, LLMs, entre otros, lo que permitió que la IA resurgiera con fuerza. Sin embargo, para el público en general, esta tecnología no era algo muy tangible. Muchos sabían que los filtros de cámara en sus aplicaciones móviles estaban relacionados con ese concepto, pero la interacción era pasiva y unidireccional. Dado que la mayoría no tenemos el conocimiento de cómo

alterar fotografías o videos, con toda naturalidad le cedíamos ese trabajo a la IA, quedándonos solo con el poder de aceptar o no el resultado final.

El fenómeno actual de la inteligencia artificial generativa, catalizado por herramientas como ChatGPT, Copilot, Gemini, Claude, y muchas otras, representa la fase más reciente de esta evolución, acercando las capacidades de la IA a un público más amplio. En poco más de un año, la inteligencia artificial se ha convertido en un término familiar para un buen porcentaje de la población y parte de la conversación en diversos ámbitos. Aun cuando muchos no se han atrevido a usarla, es improbable que no hayan escuchado sobre estas herramientas.

Esto no solo ha pasado con individuos: todas las grandes empresas del mundo ya han integrado esta tecnología en su core de negocios y han creado servicios que le permiten monetizarla, y muchas otras están justamente en ese proceso.

Esta es quizá una de las revoluciones que más rápido han impactado al mundo, gracias a las ventajas que ofrece de poder acelerar ciertos procesos, incluidas actividades creativas, que ahorran tiempo y costos.

Una de estas áreas, sin duda, es el desarrollo de software. Los modelos masivos de lenguaje (o LLM por sus siglas en inglés) son capaces

de ser entrenados para entender no solo idiomas sino lenguajes de programación. Con suficiente contexto, una IA puede generar código funcional a partir de una descripción en lenguaje natural. Muchos expertos consideran a este el fin de la programación como la conocemos, y que esto genera una depreciación directa de la labor del programador. Si en esta frase se cambiaran el contexto de programación por el de la fabricación de telas cuando aparecieron los telares mecánicos de Cartwright, o el uso de robots y sistemas de control en la industria automotriz japonesa, serían afirmaciones válidas para la distante primera revolución industrial en Europa en los 1800 y para la tercera revolución industrial de los años 60, respectivamente.

La historia nos ha enseñado que la respuesta a esto avances es siempre la misma, y no consiste en satanizar las nuevas tecnologías y prohibirlas, pues siempre termina siendo adoptada debido a las ventajas que ofrece en términos de productividad y eficiencia, que es el al final el objetivo de cualquier negocio.

Por el contrario, todos aquellos que la adoptan de manera temprana y logran integrarla, desarrollan una ventaja competitiva; los riesgos terminan siendo controlados por la regulación y la estandarización, y se continúa con el desarrollo de la tecnología, mediado por la ciencia; la academia diseña procesos de formación para masificar el manejo de

la misma en las nuevas generaciones. Al final del proceso, la adopción de la nueva tecnología es completa y nadie siquiera se cuestiona si era una opción no tomarla, y los viejos modos se olvidan o se transforman simplemente en anécdotas.

Encontrar la manera óptima de llevar a cabo este proceso de manera óptima y responsable es una labor directa de educadores, investigadores y las sociedades profesionales, de la mano con la industria y el gobierno.

### La IA en el aula

El aula de clase es el espacio de experimentación por excelencia para que los estudiantes se entrenen tanto en los aspectos fundamentales como en las nuevas tendencias, ojalá alineado con las necesidades de la industria y con la formalidad necesaria para que los estudiantes logren desarrollar habilidades útiles para su ejercicio profesional idóneo.

A pesar de su reciente aparición, en muchas universidades alrededor del mundo, la IA generativa ya se está utilizando en cursos de programación para facilitar el aprendizaje de los estudiantes. Se han realizado estudios interesantes sobre su experiencia como (Arora et al., 2024), (Zamfirescu - Pereira, 2023), (Lyu et al., 2024).

En los trabajos (Kazemitabaar et al., 2023) y (Kazemitabaar et al., 2024) se presentan estudios sobre

cómo los aprendices novatos utilizaban e interactuaban con generadores de código basados en LLMs al aprender programación en Python. Su estudio se centró en identificar los contextos en los que los estudiantes utilizan generadores de código, los tipos de indicaciones que crean y cómo modifican y verifican el código generado. Las conclusiones apuntaron a que cuando los estudiantes solicitaban que la IA generara el código tenían notas altas en las tareas de creación de código, pero mostraban limitaciones cuando el trabajo requería modificaciones al código. En este caso es evidente que, cuando los estudiantes le ceden enteramente la creación del algoritmo a la IA, no están desarrollando las habilidades para poder crearlo por sí mismos, debido a una falta de comprensión de la lógica subyacente, y corren el riesgo de generar una dependencia en la herramienta. Esto muestra que una estrategia que se reduzca a actividades de generación sería definitivamente nociva para los estudiantes.

En el trabajo de (Denny et al., 2024) se exploran los desafíos y oportunidades de integrar IA generativa en la educación informática. Los autores enfatizan la necesidad de estrategias pedagógicas actualizadas que se centren en nuevas habilidades, como la ingeniería de prompts y la evaluación de código, que son muy importantes para poder usar de manera exitosa estas nuevas herramientas.

En un segundo trabajo, (Denny et al., 2024b) introdujeron el concepto de "Problemas de Prompts" (Prompt Problems), un nuevo tipo de ejercicio de programación diseñado para ayudar a los estudiantes a aprender cómo escribir indicaciones efectivas para generadores de código de IA. Desarrollaron una herramienta web llamada *Promptly* para presentar estos problemas en donde se hace una descripción visual del objetivo final del algoritmo y los requerimientos, y los estudiantes deben escribir los prompts adecuados para suplir esa necesidad. Los autores descubrieron que los estudiantes estaban entusiasmados con ellos, apreciando el uso de sus habilidades de pensamiento computacional y la exposición a nuevos constructos de programación. Sin embargo, resaltan también que la formación en los aspectos fundamentales es muy importante para que puedan enfrentarse a estos problemas con una mejor preparación.

En el trabajo de (Aillon et. al., 2023), los autores presentan el resultado de una experiencia en clase en la que los estudiantes de últimos semestres tuvieron que desarrollar una aplicación móvil utilizando exclusivamente herramientas de IA.

El artículo muestra el proceso iterativo que siguieron, un análisis de los resultados de cada paso y la justificación para los cambios necesarios para el siguiente, y una evaluación final de la experiencia. La

construcción de una bitácora les permite a los estudiantes reflexionar sobre el proceso, sobre las decisiones que tomaron, los problemas a los que se enfrentaron, y esto le da valor a la experiencia en el desarrollo de las habilidades.

Otra metodología de trabajo con IA en el salón de clase consiste en proponer versiones alternativas de juegos ampliamente conocidos, llamado "Twisted Games" (Wightman, 2024). El objetivo de esta actividad es los estudiantes de primer semestre puedan aprovechar soluciones existentes de estos juegos en la IA, pero que deban incluir las modificaciones de estos juegos a través de los prompts claros. Algunos ejemplos de propuestas de juegos son: Batalla naval para 3 personas (A, B, C) en donde se atacan uno a uno en orden ascendentes, pero si se hunde un barco, deben cambiar de orden y atacarse de manera descendente; Buscaminas con un tablero triangular, tal que el usuario puede escoger entre tableros equiláteros, isósceles o escaleno; 4-en-línea que también soporta figuras en L de  $3 \times 1$ ,  $2 \times 2$  y  $1 \times 3$ , etc.

En estos casos finales, la escritura del programa no es el objetivo principal de la evaluación, sino si capacidad de construir instrucciones claras para la IA, la realización de pruebas para evaluar si el software generado cumple con los requerimientos y su habilidad de pensamiento crítico sobre el proceso.

## La historia se repite

La naturaleza cíclica de la historia es innegable, y se confirma por hecho como que esta no es la primera vez en la historia que se dan este tipo de discusiones se dan. Por ejemplo, hace unas décadas atrás, se dejó de enseñar lenguaje ensamblador en los programas de ingeniería de sistemas, ya que en su momento se identificó que las probabilidades de que un estudiante llegara a necesitar lenguaje de bajo nivel eran muy pocas, gracias a que los compiladores de lenguajes de alto nivel habían alcanzado un alto nivel de sofisticación para generar código suficientemente eficiente, en poco tiempo. A pesar de que el desarrollo a tan bajo nivel tiene ventajas como la eficiencia y el tamaño de los programas, no cumple con las necesidades de la industria hoy, que principalmente implica tiempos de desarrollo cortos, código seguro, multiplataforma, etc.

Hoy, en el desarrollo de software, nadie pone en duda la entera confianza que tenemos en los compiladores, y mirar hacia atrás ni siquiera se discute.

Otra discusión similar ocurrió más recientemente sobre los frameworks web como Django, NodeJS, React, entre muchos otros, ya que se convirtieron en las herramientas preferida para desarrolladores quienes. En muy poco tiempo, se podían generar plataformas genéricas funcionales y que incluían buenas prácticas, dejando el trabajo de

personalización y de ajuste de detalles a los desarrolladores. En lugar de tener que desarrollar cada línea de código en HTML, CCS y Javascript, los framework ofrecen soluciones predefinidas, con alto nivel de parametrización y personalización, que requerían muy poca intervención. La diferencia con los compiladores es que nunca vemos su resultado final, mientras que en programación web existe la posibilidad de que se tenga que entrar al código para realizar algún ajuste, lo que aún obliga, al menos, conocer los lenguajes.

La IA funciona, hasta cierto punto, de manera similar que los frameworks, aunque con un nivel de variabilidad muy alto, pues cada prompt va a generar soluciones diferentes a diferencia de código prehecho personalizado. Por eso, conocer los aspectos fundamentales de la programación para poder entender el código, probarlo e identificar la posible causa del problema son habilidades críticas para los desarrolladores, al menos mientras la IA sigue sin poder solucionar todas las preguntas sobre casos de programas complejos, con una alta probabilidad éxito.

## Conclusiones

La inteligencia artificial no es solo la última moda en computación. Esta herramienta representa un cambio de paradigma profundo frente a la manera como interactuamos con nosotros mismos, y nos hace cuestionarnos nuestro rol en las activi-

dades que considerábamos propias solo de los humanos. Quedarse pasivos esperando a que alguien más defina esto, no puede ser la estrategia, especialmente debido a la velocidad con que están ocurriendo estos cambios. Descubrir y participar activamente en la definición este papel lo antes posible es crítico para garantizar el uso correcto de estas herramientas y la relevancia de los ingenieros en el mercado mundial.

En el aula es posible integrar efectivamente la IA para apoyar los procesos de enseñanza-aprendizaje de programación. Es fundamental que tanto educadores como estudiantes abracen esta transformación, desarrollando las competencias necesarias para navegar en este nuevo panorama tecnológico.

Sin embargo, es muy importante que la definición de las reglas sea clara. Es conveniente que existan espacios en donde la IA no sea una opción, pues los estudiantes deben ser capaces de desarrollar sus propias habilidades para generar soluciones por sí mismos; por ejemplo, trabajo en clase, algunos exámenes, en donde el profesor siga siendo un apoyo y pueda contribuir en el desarrollo de los procesos de aprendizaje en los estudiantes.

Al mismo tiempo, también debe existir espacios en donde los estudiantes puedan experimentar con estas herramientas, de una manera independiente, con lineamientos

claros, para que desarrollen las habilidades de definición de requerimientos y pruebas, que son igual de valiosas que la misma creación.

Los profesores deberían ser usuarios frecuentes de estas herramientas, que conozcan sus capacidades y reconozcan sus debilidades para que puedan guiar a los estudiantes en su uso ético y donde se resalte la importancia de que se den la oportunidad de desarrollar sus propias habilidades y no depender enteramente de la inteligencia artificial, con una confianza ciega en el resultado de estas herramientas, corriendo riesgos como el plagio.

Finalmente, es vital mantener una vigilancia permanente sobre esta tecnología, no por temor a escenarios apocalípticos, sino porque estamos en el amanecer de una IA que se acerca cada vez más a la inteligencia artificial general, con habilidades cognitivas superiores y capaz de ofrecer soluciones innovadoras y sofisticadas. Como toda tecnología revolucionaria, se convertirá en un recurso invaluable para quienes logren adaptarse y desarrollen las habilidades para manejarla y aprovechar sus ventajas competitivas, o en una amenaza para aquellos que no lo consigan a tiempo. La decisión final es suya

### **Agradecimientos**

Apartes de este artículo fueron escritos con el apoyo de herramientas de inteligencia artificial generativa.

## Referencias

- Aillon, S., García, A., Velandia, N., Zarate, D., & Wightman, P. (2023). Empirical evaluation of automated code generation for mobile applications by AI tools. En 2023 IEEE Colombian Caribbean Conference (C3) (pp. 1–6).
- Arora, C., Venaik, U., Singh, P., Goyal, S., Tyagi, J., Goel, S., Singhal, U., & Kumar, D. (2024). Analyzing LLM usage in an advanced computing class in India.
- Denny, P., Prather, J., Becker, B. A., Finnie-Ansley, J., Hellas, A., Leinonen, J., Luxton-Reilly, A., Reeves, B. N., Santos, E. A., & Sarsa, S. (2024). Computing education in the era of generative AI. *Communications of the ACM*, 67(2), 56–67. <https://doi.org/10.1145/3624720>
- Denny, P., Leinonen, J., Prather, J., Luxton-Reilly, A., Amarouche, T., Becker, B. A., & Reeves, B. N. (2024b). Prompt problems: A new programming exercise for the generative AI era. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024)*, 296–302. New York, NY: Association for Computing Machinery. <https://doi.org/10.1145/3626252.3630909>
- Kazemitabaar, M., Hou, X., Henley, A., Ericson, B. J., Weintrop, D., & Grossman, T. (2023). How novices use LLM-based code generators to solve CS1 coding tasks in a self-paced learning environment.
- Kazemitabaar, M., Ye, R., Wang, X., Henley, A. Z., Denny, P., Craig, M., & Grossman, T. (2024). CodeAid: Evaluating a classroom deployment of an LLM-based programming assistant that balances student and educator needs. En *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. New York, NY: Association for Computing Machinery. <https://doi.org/10.1145/3613904.3642773>
- Lyu, W., Wang, Y., Chung, T. (Rachel), Sun, Y., & Zhang, Y. (2024). Evaluating the effectiveness of LLMs in introductory computer science education: A semester-long field study.
- Wightman, P. (2024). Twisted Games: A First Experience of Inclusion of AI tools in First Year Programming Classes. En *Latin American Conference On Computational Intelligence*. (Aceptada para publicación).
- Zamfirescu-Pereira, J., Qi, L., Hartmann, B., Denero, J., & Norouzi, N. (2023). Conversational programming with LLM-powered interactive support in an introductory computer science course. En *Generative AI for Education (GAIED): Advances, Opportunities, and Challenges - NeurIPS 2023* (pp. 1–10). Retrieved from <https://gaied.org/neurips2023/files/32/32-paper.pdf>

**Pedro Wightman.** *Ingeniero de Sistemas y Doctor en Ciencias de la Computación. Profesor Asociado de la Escuela de Ingeniería, Ciencia y Tecnología de la Universidad del Rosario, Bogotá, Colombia. Investigador senior y consultor de industria y gobierno en transformación digital e innovación, realidades extendidas, aplicaciones de blockchain, IoT, ciudades inteligentes y sistemas de información basados en localización, principalmente en las áreas de privacidad de ubicación y geocodificación. Miembro senior de IEEE y miembro del comité de EPICS in IEEE.*