

01



INTEGRACIÓN

**DE LA INTELIGENCIA ARTIFICIAL GENERATIVA PARA EL
APRENDIZAJE DE FUNDAMENTOS DE PROGRAMACIÓN: UNA
REVISIÓN SISTEMÁTICA DE LA LITERATURA**

INTEGRACIÓN

DE LA INTELIGENCIA ARTIFICIAL GENERATIVA PARA EL APRENDIZAJE DE FUNDAMENTOS DE PROGRAMACIÓN: UNA REVISIÓN SISTEMÁTICA DE LA LITERATURA

INTEGRATING GENERATIVE ARTIFICIAL INTELLIGENCE FOR LEARNING PROGRAMMING FUNDAMENTALS: A SYSTEMATIC LITERATURE REVIEW

Betty Margarita Chávez-Boza¹

E-mail: bchavez@uteq.edu.ec

ORCID: <https://orcid.org/0000-0002-1870-9080>

Orlando Ramiro Erazo-Moreta¹

E-mail: oerazo@uteq.edu.ec

ORCID: <https://orcid.org/0000-0001-5642-9920>

¹ Universidad Técnica Estatal de Quevedo. Ecuador.

Cita sugerida (APA, séptima edición)

Chávez-Boza, B. M., & Erazo-Moreta, O. (2024). Integración de la inteligencia artificial generativa para el aprendizaje de fundamentos de programación: una revisión sistemática de la literatura. *Revista Mexicana de Investigación e Intervención Educativa*, 3(2), 5-17.

RESUMEN

La Inteligencia Artificial Generativa (IAG) ha mostrado un potencial significativo para revolucionar el aprendizaje de la programación en niveles educativos desde primaria hasta educación superior. Por ello, con esta revisión sistemática de literatura se analiza cómo la IAG se integra en la enseñanza-aprendizaje de fundamentos de programación, destacando tanto sus ventajas como los desafíos asociados. Las herramientas de IAG, incluyendo sistemas de tutoría inteligente y entornos de programación interactivos, ofrecen personalización del aprendizaje y retroalimentación inmediata, lo que facilita un entorno educativo más adaptativo y atractivo. Sin embargo, la revisión de literatura revela brechas en la implementación práctica y en la evaluación crítica de estas tecnologías, sugiriendo la necesidad de un enfoque más holístico que considere aspectos técnicos y humanísticos en el diseño de soluciones educativas. Así, este estudio subraya la importancia de una colaboración multidisciplinaria para explorar efectivamente el uso ético y eficiente de la IAG en la educación en programación.

Palabras clave:

Inteligencia Artificial Generativa, aprendizaje, competencias, programación.

ABSTRACT

Generative Artificial Intelligence (GAI) has shown significant potential to revolutionize the learning of programming at educational levels from primary to higher education. This systematic literature review evaluates how GAI is integrated into the teaching of programming fundamentals, highlighting both its advantages and associated challenges. IAG tools, including intelligent tutoring systems and interactive programming environments, offer personalization of learning and immediate feedback, facilitating a more adaptive and engaging learning environment. However, the literature review reveals gaps in the practical implementation and critical evaluation of these technologies, suggesting the need for a more integrated approach that considers both technical and humanistic aspects in the design of educational solutions. Thus, this study underlines the importance of multidisciplinary collaboration to effectively explore the ethical and efficient use of AGI in programming education.

Keywords:

Generative Artificial Intelligence, learning, competences, programming.

INTRODUCCIÓN

El desarrollo de habilidades de programación (para desarrollar software) se ha establecido como un elemento esencial en la formación profesional, pero también se observa su incursión en niveles de primaria y secundaria, impulsado por la creciente necesidad de competencias tecnológicas en el panorama laboral moderno. Dado que la programación demanda capacidades avanzadas de abstracción y razonamiento lógico, los estudiantes se enfrentan frecuentemente con obstáculos que pueden comprometer la eficacia de su proceso educativo (Insuasti Portilla, 2016; Sinchi & Gómez Ceballos, 2018).

Al mismo tiempo, la Inteligencia Artificial Generativa (IAG) ha emergido como una herramienta poderosa para revolucionar diversos campos, incluida la educación, y en particular el aprendizaje de programación. La capacidad de la IAG para generar contenido nuevo y adaptativo la convierte en un recurso valioso para personalizar el aprendizaje y abordar desafíos complejos en la enseñanza de la programación. Las ventajas de la IAG en el aprendizaje de programación son significativas. Herramientas como los sistemas de tutoría inteligentes y chatbots pueden proporcionar retroalimentación instantánea y personalizada, facilitando un aprendizaje adaptable que responda a las necesidades individuales de cada estudiante. Estos avances tienen el potencial de hacer que el aprendizaje de la programación sea más interactivo, atractivo y eficiente, permitiendo a los estudiantes progresar a su propio ritmo y de acuerdo con su estilo de aprendizaje (Chang et al., 2022; Harry, 2023).

Sin embargo, a pesar de sus beneficios, existen brechas significativas en la literatura sobre cómo integrar eficazmente la IAG en la educación en programación. La literatura existente tiende a enfocarse en aplicaciones específicas de la IAG y en el desarrollo de software (Petrovska et al., 2024), pero existe la necesidad de explorar más a fondo su aplicación en la enseñanza de programación.

La implementación de la IAG en el aprendizaje de programación no está exenta de desafíos. Becker et al. (2023), destacan la urgencia de la comunidad educativa para evaluar cómo estas herramientas pueden ser aprovechadas de manera efectiva, manteniendo un equilibrio entre las oportunidades y los desafíos que presentan. Además, autores como Petrovska et al. (2024), destacan el potencial de la IAG para fomentar la evaluación crítica de la tecnología en la educación en desarrollo de software, junto con la necesidad de investigar cómo estas evaluaciones críticas podrían implementarse en el aprendizaje de programación básica. De hecho, la naturaleza compleja y a menudo incierta de los problemas en la educación

de programación requiere un enfoque más holístico que considere tanto los aspectos técnicos como los humanos de la IAG (Gautam et al., 2022).

A pesar de estos desafíos, el potencial de la IAG para enriquecer la educación en programación es inmenso. La clave para desbloquear este potencial reside en una colaboración efectiva entre expertos en educación, desarrolladores de IA e investigadores en gestión de datos para responder de manera proactiva al panorama educativo global en rápida evolución (Khosravi et al., 2023).

Para abordar estas brechas, es crucial explorar más a fondo cómo la IAG puede ser utilizada para desarrollar habilidades de programación fundamentales de manera efectiva y ética, teniendo en cuenta las preocupaciones sobre la integridad académica y la equidad en el acceso a la tecnología. Pensando en ese potencial y relevancia, así como en las brechas existentes, surge esta propuesta. La finalidad de este trabajo es realizar una revisión exhaustiva de la literatura tratando de organizar el conocimiento sobre la manera en que la IAG puede ser aprovechada en el aprendizaje de programación. Se espera que esta revisión ofrezca un mapa de las oportunidades y desafíos de la IAG en este ámbito de la educación, sirviendo a su vez de base para estudios posteriores.

METODOLOGÍA

Esta investigación toma como marco metodológico la revisión sistemática de literatura (RSL) para planificar, ejecutar y generar la información requerida sobre el tema de investigación. Para realizar el presente trabajo se utilizó el protocolo PRISMA que consta de 3 fases: identificación, cribado e inclusión (Page et al., 2021).

En primer lugar, se efectuó una búsqueda de información a partir de diversas fuentes. Tras concretar el propósito y las preguntas del estudio, se diseñaron los criterios de inclusión y exclusión. Luego, se categorizó la información utilizando un proceso de selección para este fin. A continuación, se discutieron los resultados y se redactó el documento.

Para recopilar información de bases de datos o buscadores bibliográficos, es esencial definir términos vinculados al área de estudio. Estos términos, derivados de las preguntas de investigación y las palabras clave identificadas se incluyen en la matriz PICOC (Población, Intervención, Comparación, Resultados y Contexto). Las palabras clave son fundamentales para formular la cadena de búsqueda efectiva. La tabla 1 detalla las palabras clave que sirvieron de base para la generación de la cadena de búsqueda.

Tabla 1 Elementos de la matriz PICOC.

Ámbito	Palabra clave	Sinónimo(s)
Población:	Alumnos universitarios	
Intervención:	Inteligencia artificial	IA
Comparación:	Estrategia de enseñanza	Didáctica
Resultados:	Competencias de Programación	Fundamentos de programación, aprendizaje de Programación, Programación
Contexto:	Primer año sistemas/software/computación	

Se emplearon las palabras clave y sus sinónimos para formular la cadena de consulta. Se utilizó el operador lógico OR para conectar cada sinónimo con su palabra clave correspondiente y el operador lógico AND para unir estas expresiones compuestas por las palabras clave y sus sinónimos entre sí, obteniendo como resultado la siguiente cadena de consulta:

("Alumnos universitarios") AND ("inteligencia artificial" OR "IA") AND ("estrategia de enseñanza" OR "Didáctica") AND ("competencia de programación" OR "Fundamentos de programación" OR "aprendizaje de programación" OR "programación").

Se tradujo al inglés la cadena de consulta y se eliminó el término ("Alumnos universitarios") para ampliar la búsqueda en caso de no obtener resultados en el contexto universitario, pero que signifiquen aportes para su implementación en este nivel de formación. La expresión resultante es:

("artificial intelligence" OR "AI") AND ("teaching strategy" OR "Didactics") AND ("programming competence" OR "Programming fundamentals" OR "programming learning" OR "programming")

De acuerdo con los lineamientos para desarrollar una RSL, en la Tabla 2 se presentan las bases de datos científicas multidisciplinarias y especializadas de gran impacto mundial que se utilizaron en la investigación.

Tabla 2. Bases de datos utilizadas para la investigación.

Base de Datos	Dirección Web
ACM Digital Library	https://dl.acm.org/
Science@Direct	http://www.sciencedirect.com

La cadena de consulta se adaptó a cada base de datos, según se detalla en la tabla 3:

Tabla 3. Bases de datos utilizadas para la investigación.

Base de Datos	Dirección Web	Filtros aplicados
ACM Digital Library	<i>("artificial intelligence" OR "AI") AND ("teaching strategy" OR "Didactics") AND ("programming competence" OR "Programming fundamentals" OR "programming learning" OR "programming")</i>	<ul style="list-style-type: none"> » Últimos 5 años » Research Article
Science@Direct	<i>("artificial intelligence" OR "AI") AND ("teaching strategy" OR "Didactics") AND ("programming competence" OR "Programming fundamentals" OR "programming learning" OR "programming")</i>	<ul style="list-style-type: none"> » Últimos 5 años » Artículos de investigación » Títulos - Computers and Education: Artificial Intelligence

El protocolo para revisión sistemática establece directrices para minimizar sesgos y asegurar la consistencia en el análisis. Este protocolo especifica los criterios para la selección de artículos, evalúa la calidad de la información y detalla el método para el procesamiento de datos. Por ello, para aceptar una publicación como parte de la investigación se consideró los siguientes criterios de inclusión:

IC1: Las publicaciones se encuentran en el rango de los últimos 5 años.

IC2: Las publicaciones son del contexto de la investigación.

Para rechazar una publicación se consideraron los siguientes criterios de exclusión:

EC1: Las publicaciones se encuentran fuera del rango del 2019 al 2024.

EC2: Las publicaciones son artículos de revisión.

EC3: Las publicaciones no son del contexto de investigación.

Para el proceso de cribado de los estudios se utilizaron tres análisis. El primer análisis consistió en leer el título de los artículos obtenidos con base en las palabras clave y cadenas de búsqueda, teniendo en cuenta los criterios de inclusión y exclusión. Posteriormente, se efectuó el segundo análisis, procediendo a revisar el resumen, aplicando los mismos criterios de inclusión y exclusión. Finalmente, para el tercer análisis, se descargaron los artículos candidatos, continuando con la lectura del documento completo, con el fin de filtrar los trabajos pertinentes al objeto de estudio. La Figura 1 muestra todo el procedimiento con todos los criterios de inclusión y exclusión.

La aplicación de las expresiones de búsqueda en las bases de datos permitió encontrar 44 documentos en ACM y 32 documentos en Science Direct, sin duplicados. Las bases de datos permitieron filtrar directamente la búsqueda en el rango de años establecidos en los criterios de inclusión y exclusión. Este proceso también puede observarse en la Figura 1.

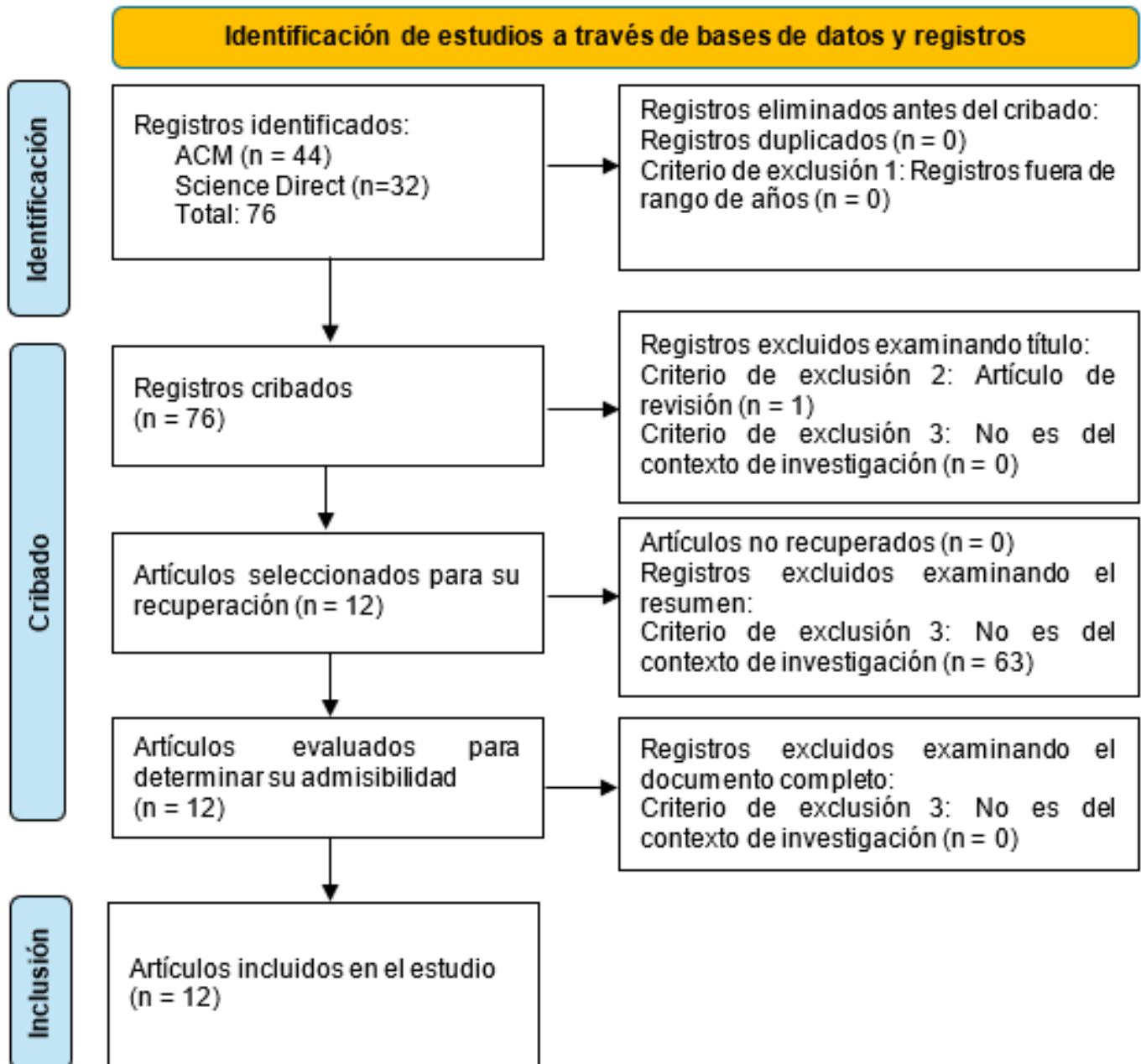


Figura 1. Diagrama de Flujo PRISMA del estudio.

DESARROLLO

En las siguientes subsecciones se exponen los resultados obtenidos del análisis de los documentos válidos. Estas subsecciones se enfocan en cada una de las preguntas específicas de investigación.

Enfoques encontrados en las fuentes bibliográficas sobre IAG y competencias de programación

Figueiredo & Garcia-Peñalvo (2020), abordan el uso de Sistemas de Tutoría Inteligente (ITS, por su significado en inglés) en cursos introductorios de programación. Se describe una arquitectura típica de un ITS, que incluye módulos de conocimiento experto, modelo del estudiante, tutoría e interfaz de usuario. El estudio se centra en la construcción de un modelo para estrategias de instrucción que ayuden a los estudiantes a aprender programación de manera efectiva. Se menciona la importancia de reducir las tasas de fracaso y falta de motivación de los estudiantes en cursos introductorios de programación. Además, se detalla un estudio realizado con 204 estudiantes de un curso de programación introductoria en Portugal. El objetivo principal es predecir la probabilidad de retención de los estudiantes en el curso. Se hace referencia a varios estudios relacionados con la enseñanza de la programación y el pensamiento computacional.

Sun et al. (2020), analizan cómo los programadores principiantes se benefician de la enseñanza combinada SPOC (Pequeño Curso en Línea Abierto y Masivo) a través de un enfoque basado en datos. Se comparan el rendimiento académico y el comportamiento de aprendizaje en una clase de enseñanza SPOC combinada con una clase de enseñanza tradicional para un curso introductorio de programación en lenguaje "C". Los resultados muestran que los estudiantes en la clase combinada SPOC tienen un mejor rendimiento académico y esfuerzos de aprendizaje, especialmente para los estudiantes con bases de programación intermedias-bajas.

Kong et al. (2021), describen un estudio sobre un curso de alfabetización en inteligencia artificial para estudiantes universitarios con diversos antecedentes de estudio. Los resultados mostraron que el curso de siete horas ayudó tanto a participantes masculinos como femeninos a mejorar significativamente su comprensión de conceptos de IA y alfabetización en ella. No hubo diferencias significativas entre los géneros en las puntuaciones de las pruebas de conceptos de IA o alfabetización en IA, y la brecha de género en la percepción de empoderamiento en IA se redujo después del curso. Además, el curso eliminó las brechas de habilidades entre participantes con antecedentes educativos diversos. Tampoco se encontraron diferencias significativas en el rendimiento de los participantes en las pruebas, independientemente de si tenían antecedentes relacionados con la informática o no. Los resultados sugieren que el curso de alfabetización en IA puede ser exitosamente impartido a estudiantes

universitarios de todos los géneros y antecedentes de estudio, ayudando a reducir la brecha digital de género en el campo de la IA.

En otro trabajo, Repenning et al. (2021), examinan el impacto de un módulo obligatorio de educación en Ciencias de la Computación introducido por la Escuela de Educación en FHNW en Suiza, centrándose en maestros en formación de nivel primario. El módulo, compuesto por dos cursos (FW y FD), ha capacitado con éxito a 1000 maestros desde 2017. Se enfoca en el Desarrollo del Pensamiento Computacional y cómo la introducción de conceptos computacionales puede cambiar las percepciones y actitudes de los maestros hacia la informática. Se recopilaron datos de evaluación de actitudes en diferentes momentos para calcular los tamaños de efecto y se observaron cambios positivos en la comprensión y actitudes hacia la informática entre los participantes.

Por su parte, Kay et al. (2022), exploran cómo los modelos de aprendizaje abiertos (OLM por las siglas en inglés) pueden mejorar la enseñanza a través del diseño de datos centrados en el aprendizaje. Se destaca la importancia de integrar conceptos de Inteligencia Artificial en la Educación para apoyar a los docentes en el diseño de datos de aprendizaje que mejoren su práctica educativa. El estudio muestra cómo los OLM pueden ayudar a los docentes a acceder y utilizar datos de aprendizaje de manera efectiva, así como el papel crucial de los modelos de aprendizaje y la *scrutability* en el fomento del aprendizaje autorregulado para los estudiantes. Además, se discute cómo el uso de ontologías en analítica de aprendizaje puede beneficiar a los docentes en el diseño de experiencias de aprendizaje personalizadas para los estudiantes.

Otro análisis es el de Yadav et al. (2022), que proporciona detalles de cómo se preparan académicamente los profesores de informática en diferentes países. Se discuten modelos comunes de preparación docente en educación en informática, así como enfoques de diferentes países para la formación y certificación de profesores de informática. Además, se ofrecen recomendaciones basadas en evidencia para mejorar la educación de los profesores de informática, centrándose en el desarrollo de conocimientos y habilidades para enseñar informática de manera efectiva a estudiantes de primaria y secundaria.

Otro enfoque es el de Yildirim-Erbasli & Bulut (2023), que exploran el uso de agentes conversacionales en evaluaciones educativas para mejorar el esfuerzo de los estudiantes durante los exámenes. Se destaca la importancia de estos agentes en la educación y se discuten los beneficios potenciales de la evaluación basada en conversaciones. Se sugiere que futuras investigaciones se centren en el impacto del entorno interactivo de evaluación centrada en conversaciones en el esfuerzo de los estudiantes durante los exámenes. Además, se menciona que el futuro de la educación implicará un mayor uso de agentes

conversacionales para guiar a los estudiantes en todas las etapas educativas.

Recientemente, Bressane et al. (2024), investigaron la influencia de las estrategias de estudio y las discapacidades de aprendizaje en el rendimiento académico de los estudiantes. A través de herramientas de inteligencia artificial, se buscó desarrollar un sistema de apoyo a la toma de decisiones para recomendar mejoras en los enfoques educativos. Emplearon una red neuronal artificial para identificar características con mayor poder explicativo basada en datos empíricos y construyeron un sistema de inteligencia artificial basado en lógica difusa para ofrecer recomendaciones sobre intervenciones educativas efectivas. El estudio buscó identificar patrones y correlaciones que puedan informar sobre la toma de decisiones basada en evidencia en contextos educativos, con el objetivo de mejorar los resultados académicos para todos los estudiantes, independientemente de sus perfiles cognitivos o desafíos de aprendizaje.

Por último, Chiu (2024), examina cómo la inteligencia artificial generativa, como ChatGPT, está transformando la educación superior. Se enfoca en el impacto de la IA en los resultados de aprendizaje, la pedagogía y la evaluación en la educación superior. Se destaca la importancia de considerar las opiniones de los estudiantes en la implementación de la IA en la educación y se señala la necesidad de investigaciones adicionales sobre el diseño efectivo de evaluaciones basadas en IA. Además, se mencionan limitaciones del estudio, como la necesidad de investigar cómo la IA afecta a diferentes regiones culturales y la importancia de recopilar las opiniones de los instructores. Se resalta la relevancia de la ética en el uso de la IA en la educación y se enfatiza la importancia de la participación estudiantil en la discusión sobre el futuro de la educación superior en un entorno dominado por la IA generativa.

¿Qué tipos de herramientas basadas en Inteligencia Artificial Generativa se han utilizado para apoyar el aprendizaje de fundamentos de programación?

En primer lugar, se puede considerar el trabajo de Figueiredo & Garcia-Peñalvo (2020). Ellos utilizaron diversas herramientas basadas en Inteligencia Artificial Generativa para apoyar el aprendizaje de fundamentos de programación. Algunas de ellas son:

- **Sistemas de Tutoría Inteligente (ITS):** Estos sistemas utilizan técnicas de inteligencia artificial para adaptar la enseñanza a las necesidades individuales de los estudiantes, brindando retroalimentación personalizada y apoyo en el proceso de aprendizaje.
- **Entornos de programación interactivos:** Herramientas que permiten a los estudiantes escribir y ejecutar código, ofreciendo sugerencias y correcciones automáticas para mejorar la comprensión de la sintaxis y semántica del código.

- **Analizadores de programas:** Herramientas que identifican errores y bugs en el código desarrollado por los estudiantes, ayudándoles a mejorar sus habilidades de programación.

Estas herramientas basadas en Inteligencia Artificial Generativa han demostrado ser útiles para mejorar la enseñanza y el aprendizaje de programación, proporcionando un entorno interactivo y personalizado para los estudiantes

Por otro lado, Sun et al. (2020), mencionan haber utilizado una herramienta de apoyo llamada *rainy classroom* que se basa en inteligencia artificial generativa para respaldar el aprendizaje de fundamentos de programación en la enseñanza combinada con SPOC. Esta herramienta toma nota del estado de registro de los estudiantes, el tiempo de aprendizaje en línea, la frecuencia de interacción con el profesor, la puntuación en cuestionarios en clase y los datos de advertencia temprana de los estudiantes. Estos datos objetivos se utilizan en la evaluación formativa de la práctica docente, lo que ilustra las ventajas de la aplicación de Big Data en educación.

En la propuesta de Kong et al. (2021), se utilizaron herramientas basadas en Inteligencia Artificial Generativa, como chatbots y sistemas de tutoría inteligente, para apoyar el aprendizaje de fundamentos de programación en el curso de alfabetización en inteligencia artificial para estudiantes universitarios con diversos antecedentes de estudio. Estas herramientas ayudaron a los participantes a desarrollar una comprensión conceptual de la IA y a mejorar su alfabetización en IA, incluso si no tenían conocimientos previos de programación.

También, Repenning et al. (2021), mencionan el uso de herramientas basadas en Inteligencia Artificial Generativa, como el *Generative Adversarial Networks* (GANs), para apoyar el aprendizaje de fundamentos de programación. Estas herramientas permiten a los estudiantes crear juegos y simulaciones de manera interactiva, lo que facilita la comprensión de conceptos computacionales y fomenta la creatividad en el proceso de aprendizaje. En el mismo sentido, Kay et al. (2022), usan GANs, para apoyar el aprendizaje de fundamentos de programación. En este caso se utilizan para generar ejemplos de código y proporcionar retroalimentación a los estudiantes en su proceso de aprendizaje de programación.

Yadav et al. (2022), fundamentan su propuesta con el uso de herramientas basadas en Inteligencia Artificial Generativa, como ChatGPT de OpenAI, para apoyar el aprendizaje de fundamentos de programación. Estas herramientas permiten a los estudiantes interactuar con un sistema de IA para recibir retroalimentación y ayuda en sus procesos de aprendizaje en programación.

Yildirim-Erbaşlı & Bulut (2023), gestionaron el uso de grandes modelos de lenguaje, como ChatGPT, para mejorar la eficiencia del diálogo en la tutoría y apoyar el aprendizaje

de fundamentos de programación. Estos modelos de lenguaje generativo han demostrado ser útiles en la generación de texto humano similar, resúmenes de texto y recuperación de respuestas en formato de conversación. Su capacidad para aprender reglas, patrones y estructuras del lenguaje a partir de grandes cantidades de datos los hace herramientas prometedoras para revolucionar diversos campos, incluida el aprendizaje de programación.

Según Bressane et al. (2024), la aplicación de herramientas de inteligencia artificial generativa, como ChatGPT, resultaron útiles para mejorar la redacción y la legibilidad del contenido. Estas herramientas han sido utilizadas para apoyar el proceso de escritura y revisión, con el objetivo de mejorar la calidad del lenguaje utilizado en la investigación sobre el papel de las estrategias de estudio y las discapacidades de aprendizaje en el rendimiento académico de los estudiantes.

También, Chiu (2024), utilizó algunas herramientas basadas en Inteligencia Artificial Generativa que se han empleado para apoyar el aprendizaje de fundamentos de programación. Algunas de estas herramientas incluyen ChatGPT, Midjourney y GitHub Copilot. Estas aplicaciones se utilizan para producir aprendizaje personalizado, crear o revisar materiales de enseñanza, desarrollar código para investigaciones y redactar correos electrónicos para responder a consultas de estudiantes.

Los artículos antes mencionados destacan el uso diverso y enriquecedor de herramientas basadas en IAG, evidenciado en las investigaciones revisadas. Estas herramientas abarcan desde tutores inteligentes hasta sistemas avanzados de tutoría, pasando por la generación de pseudocódigo y la aplicación de tecnologías de visión por computadora para fines educativos.

Contrastando la información obtenida de la revisión de literatura con criterios de otros autores para consolidar las herramientas más utilizadas, se tiene los siguientes detalles:

- Chatbot: Los chatbots pueden actuar como asistentes virtuales ayudando a los estudiantes con preguntas concretas y ofreciendo correcciones automáticas. Facilitan el aprendizaje autodirigido y pueden adaptarse al ritmo y estilo de aprendizaje de cada alumno, mejorando la comprensión y la retención de conocimientos (Lin, 2022). Otra ventaja es la eficiencia en la gestión del tiempo, tanto para los estudiantes como para los instructores, que les permite centrarse en aspectos más complejos del proceso educativo mientras los chatbots se encargan de las interacciones rutinarias y repetitivas (Dimitriadis, 2020). Estos aspectos ponen de relieve cómo los chatbots pueden ser herramientas valiosas en la enseñanza de la programación, contribuyendo significativamente al proceso de aprendizaje moderno (Kong et al., 2021).
- Analizadores de programas: El uso de GitHub Copilot es potenciado por el modelo de lenguaje Codex de OpenAI, para sugerir código en tiempo real. GitHub Copilot es una herramienta de generación de código desarrollada por GitHub y OpenAI, conocida como un *AI pair programmer* (Prather et al., 2023). Utiliza un modelo de lenguaje entrenado en miles de millones de líneas de código de proyectos de código abierto, lo que le permite generar código en varios lenguajes de programación a partir de descripciones en lenguaje natural (Yetistiren et al., 2022). Esta herramienta representa un significativo apoyo práctico para los estudiantes, al ofrecer sugerencias de código basadas en las entradas de los usuarios, lo que estimula el aprendizaje interactivo y la comprensión de patrones de programación (Chiu, 2024).
- Grandes Modelos de Lenguaje: La aplicación de grandes modelos de lenguaje, como GPT-3 o GPT-4 en el contexto educativo, ofrece la posibilidad de generar texto que imita al humano, lo cual se ha explorado en tareas de resumen, traducción, y respuesta a preguntas, entre otras, proporcionando un amplio espectro de aplicaciones en la enseñanza de la programación (Prather et al., 2023; Yildirim-Erbasli & Bulut, 2023; Bressane et al., 2024; Chiu, 2024).
- Sistemas de Tutoría Inteligente: Estos sistemas son alimentados por algoritmos de aprendizaje profundo y son capaces de interactuar con los estudiantes mediante la respuesta a preguntas y la provisión de ejemplos de código, facilitando así el proceso de aprendizaje de la programación (Roldán-Álvarez & Mesa, 2023). También ofrecen retroalimentación adaptativa a los estudiantes basándose en su desempeño. Aunque su aplicación suele estar limitada a tareas específicas, su capacidad para adaptarse y responder a las necesidades individuales de aprendizaje los convierte en una herramienta valiosa en la educación en programación (Prather et al., 2023). Este modelo de sistema incluye un módulo de conocimiento experto con algoritmos y un módulo predictivo para ayudar en las decisiones del profesor al sugerir nuevas actividades. Además, se destaca la importancia de que el profesor cuente con la ayuda de un sistema de inteligencia artificial para mejorar la enseñanza y el aprendizaje de programación introductoria (Figueiredo & García-Peñalvo, 2020; Kong et al., 2021).
- Generación Automática de Código o Pseudocódigo: Duong et al. (2022), señalan el uso de herramientas de IAG para la generación automática de pseudocódigo a través de técnicas de traducción automática. Esta innovación apoya a los estudiantes al ofrecer explicaciones y sugerencias en pseudocódigo, lo que facilita la comprensión de diversos enfoques para resolver problemas de programación.

- Entornos de programación interactivos: Desempeñan un papel crucial en la educación de la programación al permitir a los estudiantes escribir y ejecutar código, ofreciendo sugerencias y correcciones automáticas. Estas herramientas mejoran la comprensión de la sintaxis y la semántica del código, lo que es esencial para los aprendices. Herramientas interactivas como IntelliCode Compose utilizan modelos transformadores generativos para predecir secuencias de tokens de código, mejorando así la completitud del código hasta generar líneas enteras de código sintácticamente correcto, lo que demuestra su eficacia en entornos de desarrollo integrados modernos (Svyatkovskiy et al., 2020). Estas herramientas permiten a los estudiantes escribir y ejecutar código, ofreciendo sugerencias y correcciones automáticas para mejorar la comprensión de la sintaxis y semántica del código (Figueiredo & García-Peñalvo, 2020)

¿Qué estrategias de aprendizaje se utilizan cuando se implementa IA en cursos de programación?

Figueiredo & García-Peñalvo (2020), señalan que, cuando se implementa Inteligencia Artificial (IA) en cursos de programación, se utilizan diversas estrategias de aprendizaje para mejorar la experiencia educativa de los estudiantes. Algunas de las estrategias comunes incluyen:

- Personalización del aprendizaje: La IA permite adaptar el contenido y la dificultad de las lecciones según las necesidades individuales de cada estudiante, brindando un enfoque más personalizado y efectivo.
- Retroalimentación instantánea: Los sistemas basados en IA pueden proporcionar retroalimentación inmediata sobre el desempeño de los estudiantes, identificando errores y ofreciendo sugerencias para mejorar, lo que facilita el proceso de aprendizaje.
- Tutoría inteligente: Los sistemas de tutoría inteligente utilizan IA para simular la interacción con un tutor humano, guiando a los estudiantes a través de los conceptos clave, resolviendo dudas y proporcionando apoyo individualizado.
- Análisis de datos y seguimiento del progreso: La IA puede usarse para analizar el rendimiento de los estudiantes, identificar patrones de aprendizaje y ofrecer recomendaciones para optimizar el proceso educativo, permitiendo un seguimiento detallado del progreso de cada estudiante.

Estas estrategias de aprendizaje, potenciadas por la IA, contribuyen a mejorar la eficacia y la eficiencia de la

enseñanza de la programación, fomentando un aprendizaje más interactivo, personalizado y centrado en el estudiante.

Por su parte, Sun et al. (2020), señalan que cuando se implementa IA en cursos de programación, se utilizan estrategias de aprendizaje como la extracción de patrones del comportamiento de los estudiantes a través de datos de interacción registrados, con el objetivo de evaluar el rendimiento de los estudiantes. Además, se recopilan datos de comportamiento de los estudiantes en plataformas de aprendizaje en línea y se utilizan técnicas de minería de datos educativa y análisis de aprendizaje para procesar e interpretar estos datos. Estas estrategias permiten evaluar el rendimiento académico de los estudiantes y ajustar las estrategias de enseñanza de manera oportuna.

Figueiredo & García-Peñalvo (2020), mencionan que se ha utilizado un modelo de Sistemas de Tutoría Inteligente (ITS) para apoyar el aprendizaje de fundamentos de programación. Este modelo incluye un módulo de conocimiento experto con algoritmos y un modelo predictivo para ayudar en las decisiones del profesor al sugerir nuevas actividades. Además, se destaca la importancia de que el profesor cuente con la ayuda de un sistema de inteligencia artificial para mejorar la enseñanza y el aprendizaje de programación introductoria.

Se destaca la propuesta de Kong et al. (2021), quienes implementaron una estrategia de aprendizaje conocida como enfoque de aula invertida (flipped classroom approach) en el curso de alfabetización en inteligencia artificial para estudiantes universitarios. Esta estrategia implicaba que los participantes revisaran el material de estudio antes de la clase, lo que les permitía dedicar más tiempo en el aula a actividades prácticas y de aplicación de los conceptos aprendidos. Esta metodología facilitó un aprendizaje más interactivo y participativo, lo que contribuyó al desarrollo de una comprensión más profunda de los fundamentos de la programación y la inteligencia artificial.

Repenning et al. (2021), refieren estrategias que incluyen el uso de herramientas de Pensamiento Computacional, la enseñanza de abstracciones y algoritmos a través de Patrones de Pensamiento Computacional, el diseño de juegos y simulaciones interactivas para promover la creatividad, y la exposición a conceptos como la visualización de datos y la programación de inteligencia artificial en proyectos de juegos. Además, se destaca la importancia de combinar actividades desenchufadas y enchufadas para enseñar conceptos de programación de manera efectiva.

Bressane et al. (2024), utilizan estrategias de aprendizaje personalizado, tutoría automatizada, evaluación del conocimiento personalizada y automatización de tareas. Estas estrategias buscan empoderar a los educadores para brindar una mejor orientación, mientras los estudiantes

disfrutan de una mayor autonomía en su proceso de aprendizaje. Además, se destaca que las herramientas de inteligencia artificial pueden actuar como sistemas de apoyo metafóricos para los instructores, mejorando así la experiencia educativa en el campo de la programación.

El uso de Inteligencia Artificial (IA) en cursos de programación, propuesto por Yildirim-Erbaşlı & Bulut (2023), aplica estrategias de aprendizaje que incluyen la personalización del aprendizaje para adaptarse a las necesidades individuales de los estudiantes, la provisión de retroalimentación inmediata y la interacción en tiempo real a través de agentes conversacionales. Estas estrategias buscan mejorar la experiencia de aprendizaje de los estudiantes al ofrecerles un entorno de aprendizaje interactivo y adaptativo que fomente la participación y la comprensión de los conceptos de programación.

También, Kay et al. (2022), utilizan estrategias de aprendizaje centradas en el estudiante, como el aprendizaje autorregulado y la retroalimentación personalizada. Estas estrategias buscan mejorar la experiencia de aprendizaje de los estudiantes al proporcionarles herramientas y recursos que se adapten a sus necesidades individuales, fomentando así un aprendizaje más efectivo y significativo.

De igual manera, Chiu (2024), menciona estrategias de aprendizaje que se centran en maximizar las oportunidades que ofrece la IA generativa. Estas estrategias incluyen el fomento de un aprendizaje personalizado, interacciones más efectivas y conversaciones, así como retroalimentación mejorada. La introducción de la IA generativa en la educación superior altera las pedagogías al cambiar el énfasis de la transferencia de conocimientos al procesamiento de conocimientos y de un aprendizaje disciplinario a uno interdisciplinario.

Yadav et al. (2022), utilizan estrategias de aprendizaje como la retroalimentación personalizada, la tutoría adaptativa y la generación automática de ejercicios. Estas estrategias permiten a los estudiantes recibir un apoyo individualizado y adaptado a sus necesidades, lo que mejora su proceso de aprendizaje en programación.

Los hallazgos extraídos de los artículos revisados revelan un enfoque multifacético en la aplicación de estrategias de aprendizaje, que abarca desde la personalización, adaptabilidad y el aprendizaje colaborativo. A continuación, se presenta una síntesis de las estrategias de aprendizaje identificadas:

- **Aprendizaje Adaptativo y Personalizado:** Roldán-Álvarez & Mesa (2023), subrayan la implementación de estrategias de aprendizaje adaptativas y personalizadas a través de tutores inteligentes que ofrecen retroalimentación instantánea y recursos complementarios, permitiendo a los estudiantes aprender de sus errores y profundizar en temas de interés según su Zona de Desarrollo Próximo.

- **Retroalimentación Personalizada:** Prather et al. (2023), destacan la importancia del **feedback** (retroalimentación) personalizado proporcionado por herramientas de IA, como sugerencias de código en tiempo real, que guían a los estudiantes en la resolución de problemas de programación de manera específica y contextualizada.
- **Práctica Individual y Feedback Oportuno:** Duong et al. (2022), identifican la práctica individualizada y la retroalimentación automática y oportuna como estrategias clave, ofreciendo a los estudiantes explicaciones detalladas y sugerencias para mejorar sus habilidades de codificación y la resolución de problemas.

¿Cuáles son los beneficios de utilizar la IA generativa para enseñar programación?

La introducción de la IA generativa en la educación superior puede cambiar las pedagogías al enfocarse en el procesamiento del conocimiento y promover un aprendizaje interdisciplinario. Una vez consolidada la información de los trabajos seleccionados, se ha podido determinar que los beneficios de utilizar IAG en la enseñanza para el desarrollo de competencias de programación son:

- **Personalización del aprendizaje:** La IA generativa permite adaptar la enseñanza a las necesidades individuales de cada estudiante, ofreciendo un enfoque personalizado que facilita la comprensión de los conceptos de programación. La IA generativa, como ChatGPT, se destaca por su capacidad para ofrecer un aprendizaje más efectivo y conversaciones realistas (Figueiredo & García-Peñalvo, 2020; Chiu, 2024).
- **Retroalimentación inmediata:** Los sistemas basados en IA pueden proporcionar retroalimentación instantánea sobre el desempeño de los estudiantes, identificando errores y ofreciendo sugerencias para mejorar, lo que contribuye a un aprendizaje más efectivo y eficiente (Figueiredo & García-Peñalvo, 2020; Kong et al., 2021). Las herramientas de IA generativa pueden ofrecer retroalimentación inmediata sobre el código y los proyectos de los estudiantes, lo que les permite corregir errores y mejorar sus habilidades de programación de manera eficiente (Repenning et al., 2021).
- **Retroalimentación personalizada:** Las herramientas basadas en IA pueden proporcionar retroalimentación personalizada a los estudiantes, identificando áreas de mejora y ofreciendo sugerencias específicas para su desarrollo y la posibilidad de ofrecer apoyo individualizado en tiempo real (Kay et al., 2022a; Yadav et al., 2022).
- **Rendimiento académico:** Mejora significativa en las calificaciones de los estudiantes, aumentando la tasa de aprobación y el interés en las actividades de aprendizaje al proporcionar retroalimentación personalizada

y adaptar las estrategias de enseñanza según las necesidades individuales (Sun et al., 2020).

- Motivación y participación: La IA generativa puede fomentar la motivación de los estudiantes al ofrecer actividades interactivas, desafiantes y personalizadas, lo que aumenta la participación y el compromiso con el proceso de aprendizaje (Figueiredo & García-Peñalvo, 2020; Sun et al., 2020).
- Eficiencia en la enseñanza: Optimiza el tiempo y los esfuerzos de los docentes al proporcionar herramientas que facilitan la evaluación del progreso de los estudiantes y la identificación de áreas de mejora, permitiendo una enseñanza más eficiente (Figueiredo & García-Peñalvo, 2020; Sun et al., 2020).
- Apoyo personalizado: Las herramientas basadas en IA generativa, como chatbots y sistemas de tutoría inteligente, pueden ofrecer retroalimentación personalizada y adaptada a las necesidades individuales de los estudiantes, lo que mejora la comprensión de los conceptos de programación (Kong et al., 2021).
- Aprendizaje interactivo: La IA generativa puede fomentar un aprendizaje más interactivo, adaptativo y participativo al ofrecer explicaciones detalladas, orientación en tiempo real, recursos adicionales y actividades prácticas que involucren a los estudiantes de manera activa en su proceso de aprendizaje mientras trabajan en sus proyectos de programación (Kong et al., 2021a; Kay et al., 2022a).
- La inteligencia artificial generativa en la creación de juegos y simulaciones interactivas no solo facilita proyectos prácticos y atractivos para los estudiantes, sino que también promueve la creatividad al permitirles explorar y experimentar con conceptos de programación de manera innovadora. Además, esta tecnología mejora la redacción y legibilidad del contenido educativo, proporciona apoyo en el proceso de escritura y revisión del material, y empodera a los educadores para ofrecer una orientación más efectiva, mientras brinda autonomía a los estudiantes en su proceso de aprendizaje (Repenning et al., 2021).
- Generación de ejemplos de código: La IA generativa puede crear ejemplos de código automáticamente, lo que ayuda a los estudiantes a comprender mejor los conceptos de programación y a practicar para el desarrollo de sus habilidades (Kay et al., 2022).

CONCLUSIONES

La revisión sistemática realizada ha evidenciado que la Inteligencia Artificial Generativa representa una herramienta transformadora en el ámbito de la enseñanza-aprendizaje de la programación. A través del análisis

de diversas investigaciones, se ha demostrado que la IAG no solo facilita la personalización del aprendizaje, sino que también mejora significativamente la interacción y la retroalimentación en tiempo real, elementos cruciales para un aprendizaje efectivo de la programación.

Hemos observado que las herramientas basadas en IAG, como los Sistemas de Tutoría Inteligente y entornos de programación interactivos, proporcionan un soporte inestimable en el proceso educativo al adaptar el contenido a las necesidades individuales de cada estudiante y al ofrecer retroalimentación personalizada que guía y corrige a los estudiantes de manera eficiente. Estas herramientas no solo incrementan el rendimiento académico, sino que también potencian la motivación y el compromiso de los estudiantes hacia el aprendizaje de programación.

Sin embargo, también se reconoce la existencia de desafíos significativos que se deben abordar en la implementación efectiva de la IAG en educación. La necesidad de equilibrar la tecnología con intervenciones pedagógicas humanas es fundamental para evitar una dependencia excesiva en las soluciones automáticas y para garantizar una comprensión profunda de los fundamentos de la programación. Además, la ética y la integridad académica son preocupaciones que deben abordarse al integrar estas tecnologías avanzadas en los currículos educativos.

Por lo tanto, mientras se reconoce el potencial de la IAG para revolucionar la enseñanza de la programación, se recomienda una implementación cuidadosa que considere tanto las capacidades tecnológicas como las necesidades pedagógicas. La colaboración continua entre educadores, desarrolladores de IA y expertos en políticas educativas será esencial para maximizar los beneficios de la IAG y minimizar sus riesgos en entornos educativos.

Finalmente, se sugiere la continuación de la investigación en este campo para explorar más a fondo cómo se pueden superar los desafíos actuales y cómo la IAG puede ser utilizada no solo como una herramienta de enseñanza, sino también como un medio para fomentar habilidades críticas de pensamiento y solución de problemas en los estudiantes. Con un enfoque integrado y ético, la IAG tiene el potencial de enriquecer significativamente la educación en programación, preparando a los estudiantes para enfrentar los desafíos del futuro digital.

REFERENCIAS BIBLIOGRÁFICAS

- 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. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, 1, 500–506. <https://doi.org/10.1145/3545945.3569759>

- Bressane, A., Zwirn, D., Essiptchouk, A., Saraiva, A. C. V., Carvalho, F. L. de C., Formiga, J. K. S., Medeiros, L. C. de C., & Negri, R. G. (2024). Understanding the role of study strategies and learning disabilities on student academic performance to enhance educational approaches: A proposal using artificial intelligence. *Computers and Education: Artificial Intelligence*, *6*. <https://doi.org/10.1016/j.caeai.2023.100196>
- Chang, Q., Pan, X., Manikandan, N., & Ramesh, S. (2022). Artificial Intelligence Technologies for Teaching and Learning in Higher Education. *International Journal of Reliability, Quality and Safety Engineering*, *29*(05). <https://doi.org/10.1142/S021853932240006X>
- Chiu, T. K. (2024). Future research recommendations for transforming higher education with generative AI. *Computers and Education: Artificial Intelligence*, *6*. <https://doi.org/https://doi.org/10.1016/j.caeai.2023.100197>
- Dimitriadis, G. (2020). Evolution in Education: Chatbots. *Homo Virtualis*, *3*(1). <https://doi.org/10.12681/homvir.23456>
- Figueiredo, J., & García-Peñalvo, F. J. (2020). Intelligent Tutoring Systems approach to Introductory Programming Courses. *Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality*, 34–39. <https://doi.org/10.1145/3434780.3436614>
- Gautam, S., Akgun, M., & Mitra, P. (2022). Exploring the challenges of AI experts to inform AI curriculum. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, *2*, 1338–1338. <https://doi.org/10.1145/3545947.3576284>
- Harry, A. (2023). Role of AI in Education. *Interdisciplinary Journal and Humanity (INJURITY)*, *2*(3), 260–268. <https://doi.org/10.58631/injurity.v2i3.52>
- Insuasti Portilla, J. (2016). Problemas de enseñanza y aprendizaje de los fundamentos de programación. *Educación y Desarrollo Social*, *10*(2), 234–246.
- Kay, J., Bartimote, K., Kitto, K., Kummerfeld, B., Liu, D., & Reimann, P. (2022). Enhancing learning by Open Learner Model (OLM) driven data design. *Computers and Education: Artificial Intelligence*, *3*. <https://doi.org/10.1016/j.caeai.2022.100069>
- Khosravi, H., Sadiq, S., & Amer-Yahia, S. (2023). Data management of AI-powered education technologies: Challenges and opportunities. *Journal of Learning Letters*. <https://doi.org/10.59453/XLUD7002>
- Kong, S.-C., Man-Yin Cheung, W., & Zhang, G. (2021). Evaluation of an artificial intelligence literacy course for university students with diverse study backgrounds. *Computers and Education: Artificial Intelligence*, *2*. <https://doi.org/10.1016/j.caeai.2021.100026>
- Lin, Y. H. (2022). Chatbot Script Design for Programming Language Learning. *2022 IEEE 5th Eurasian Conference on Educational Innovation (ECEI)*, 123–125. <https://doi.org/10.1109/ECEI53102.2022.9829460>
- Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., ... Alonso-Fernández, S. (2021). Declaración PRISMA 2020: una guía actualizada para la publicación de revisiones sistemáticas. *Revista Española de Cardiología*, *74*(9), 790–799. <https://doi.org/10.1016/j.recesp.2021.06.016>
- Petrovska, O., Clift, L., Moller, F., & Pearsall, R. (2024). Incorporating Generative AI into Software Development Education. *Proceedings of the 8th Conference on Computing Education Practice*, 37–40. <https://doi.org/10.1145/3633053.3633057>
- Prather, J., Reeves, B. N., Denny, P., Becker, B. A., Leinonen, J., Luxton-Reilly, A., Powell, G., Finnie-Ansley, J., & Santos, E. A. (2023). "It's Weird That it Knows What I Want": Usability and Interactions with Copilot for Novice Programmers. *ACM Transactions on Computer-Human Interaction*, *31*(1). <https://doi.org/10.1145/3617367>
- Repenning, A., Lamprou, A., & Basawapatna, A. (2021). Computing Effect Sizes of a Science-first-then-didactics Computational Thinking Module for Preservice Elementary School Teachers. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 274–280. <https://doi.org/10.1145/3408877.3432446>
- Roldán-Álvarez, D., & Mesa, F. J. (2023). Intelligent Deep-Learning Tutoring System to Assist Instructors in Programming Courses. *IEEE Transactions on Education*, *67*(1), 1–9. <https://doi.org/10.1109/te.2023.3331055>
- Sinchi, E. R., & Gómez Ceballos, G. P. (2018). Acceso y deserción en las universidades. Alternativas de financiamiento. *Alteridad*, *13*(2), 274–287. <https://doi.org/10.17163/alt.v13n2.2018.10>
- Sun, Q., Song, Y., & Tan, H. (2020). How do early programmers benefit from SPOC blended teaching. *Proceedings of the ACM Turing Celebration Conference - China*, 65–70. <https://doi.org/10.1145/3393527.3393539>
- Svyatkovskiy, A., Deng, S. K., Fu, S., & Sundaresan, N. (2020). IntelliCode compose: code generation using transformer. *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 1433–1443. <https://doi.org/10.1145/3368089.3417058>

- Yadav, A., Connolly, C., Berges, M., Chytas, C., Franklin, C., Hijón-Neira, R., Macann, V., Margulieux, L., Ottenbreit-Leftwich, A., & Warner, J. R. (2022). A Review of International Models of Computer Science Teacher Education. *Proceedings of the 2022 Working Group Reports on Innovation and Technology in Computer Science Education*, 65–93. <https://doi.org/10.1145/3571785.3574123>
- Yetistiren, B., Ozsoy, I., & Tuzun, E. (2022). Assessing the quality of GitHub copilot's code generation. *Proceedings of the 18th International Conference on Predictive Models and Data Analytics in Software Engineering*, 62–71. <https://doi.org/10.1145/3558489.3559072>
- Yildirim-Erbasli, S. N., & Bulut, O. (2023). Conversation-based assessment: A novel approach to boosting test-taking effort in digital formative assessment. *Computers and Education: Artificial Intelligence*, 4. <https://doi.org/10.1016/j.caeai.2023.100135>