



## El aprendizaje inteligente en la programación: una revisión sistemática de la literatura

### Smart learning in programming: a systematic literature review

#### Autores

\*Edison Amador Miguez Gordillo

✉ [edison.miguez@unach.edu.ec](mailto:edison.miguez@unach.edu.ec)



Patricio Ricardo Humanante Ramos

✉ [phumanante@unach.edu.ec](mailto:phumanante@unach.edu.ec)



Universidad Nacional de Chimborazo,  
Facultad de Ciencias de la Educación,  
Humanas y Tecnologías, Riobamba,  
Chimborazo, Ecuador

\*Autor para correspondencia

#### Comó citar el artículo:

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>

Enviado: 18/05/2026

Aceptado: 03/06/2026

Publicado: 04/06/2026

#### Resumen

El aprendizaje inteligente en la educación de programación ha experimentado un crecimiento sostenido debido a la incorporación de inteligencia artificial, sistemas adaptativos y herramientas generativas destinadas a personalizar el aprendizaje. El objetivo de este estudio fue analizar los avances, enfoques, herramientas y desafíos asociados con la aplicación de tecnologías inteligentes en contextos de educación de programación. Para ello, se realizó una revisión sistemática de la literatura (RSL) basada en la metodología propuesta por Kitchenham, estructurada en fases de planificación, desarrollo y documentación. Se recuperó información de las bases de datos Scopus e IEEE Xplore, aplicando criterios de inclusión y exclusión, así como una evaluación de la calidad metodológica, para seleccionar estudios científicos publicados entre 2022 y 2026. Los resultados muestran un predominio de los Sistemas de Tutoría Inteligente (STI), modelos basados en grandes modelos de lenguaje (LLM), plataformas adaptativas y entornos gamificados inmersivos. Estas tecnologías permiten retroalimentación inmediata, adaptación al rendimiento del estudiante y apoyo personalizado a lo largo del proceso de aprendizaje. Los estudios reportan mejoras en el rendimiento académico, la comprensión conceptual, la resolución de problemas, la depuración de código, la motivación y el pensamiento computacional. Las principales limitaciones identificadas son la dependencia excesiva de herramientas generativas, la posibilidad de respuestas inexactas y la limitada evidencia longitudinal sobre sus efectos en diferentes contextos educativos. La integración de la inteligencia artificial y la mediación del profesor fomenta experiencias de aprendizaje más flexibles y autónomas, centradas en las necesidades individuales de los estudiantes de programación.

**Palabras clave:** aprendizaje inteligente; programación; aprendizaje adaptativo.

#### Abstract

Smart learning in programming education has experienced sustained growth due to the incorporation of artificial intelligence, adaptive systems, and generative tools aimed to personalizing learning. The objective of this study was to analyze the advances, approaches, tools, and challenges associated with the application of intelligent technologies in programming education contexts. To this end, a systematic literature review (SLR) was conducted based on the methodology proposed by Kitchenham, structured into planning, development, and documentation phases. Information was retrieved from the Scopus and IEEE Xplore databases, applying inclusion and exclusion criteria, as well as methodological quality assessment, to select scientific studies published between 2022 and 2026. The results show a predominance of Intelligent Tutoring System (ITS), models based on large language models (LLMs), adaptive platforms, and immersive gamified environments. These technologies enable immediate feedback, adaptation to student performance, and personalized support throughout the learning process. The studies report improvements in academic performance, conceptual understanding, problem solving, code debugging, motivation, and computational thinking as well. The main limitations identified are the excessive dependence on generative tools, the possibility of inaccurate responses, and the limited longitudinal evidence on their effects across different educational contexts. The integration of artificial intelligence and teacher mediation fosters more flexible and autonomous learning experiences centered on the individual needs of programming students.

**Keywords:** smart learning; programming; adaptive learning.



## 1. Introducción

En la última década, la inteligencia artificial (IA) ha adquirido un papel central en el ámbito educativo al impulsar enfoques orientados a la personalización del aprendizaje y al uso estratégico de datos para la toma de decisiones pedagógicas, lo que ha permitido superar esquemas homogéneos de enseñanza y atender con mayor precisión la diversidad estudiantil, tal como señalan Zawacki-Richter et al. (2022), quienes destacan su contribución en la transición hacia modelos centrados en el estudiante.

El aprendizaje inteligente se entiende como un enfoque educativo que integra tecnologías digitales, analítica de datos y herramientas de inteligencia artificial para crear entornos de aprendizaje adaptativos, capaces de mejorar los procesos educativos mediante la personalización y la innovación pedagógica, facilitando así una transformación significativa en la enseñanza (Amo-Filva, 2023). Dentro de este panorama, el aprendizaje inteligente se vuelve una propuesta que articula capacidades tecnológicas con fundamentos pedagógicos, integrando sistemas adaptativos y analítica de aprendizaje con el propósito de ajustar los contenidos, ritmos y estrategias de enseñanza; en esta línea, Míguez Gordillo (2025) sostiene que este enfoque se basa en la mediación de la IA para identificar necesidades individuales y orientar intervenciones docentes pertinentes.

La enseñanza de la programación representa un campo de especial interés para este tipo de enfoques debido a las exigencias cognitivas que implica, entre ellas la abstracción, la lógica formal y la resolución estructurada de problemas, lo que ha sido asociado con dificultades persistentes en estudiantes de distintos niveles, estudios recientes confirman que el aprendizaje de la programación continúa siendo un desafío significativo para principiantes y que identificar factores que influyen en el rendimiento resulta complejo en cursos introductorios (Köhler et al., 2023).

Ante este escenario, diversas investigaciones han explorado el uso de herramientas basadas en IA como mecanismos de apoyo al aprendizaje de la programación, incluyendo tutores inteligentes, sistemas de retroalimentación automática y asistentes de codificación, los cuales ofrecen acompañamiento continuo y adaptado al desempeño del estudiante, generando mejoras en la comprensión conceptual y en la práctica autónoma (Finnie-Ansley et al., 2022; Topham et al., 2026). No obstante, la incorporación de estas tecnologías en contextos educativos reales presenta desafíos que trascienden lo técnico, ya que involucran aspectos pedagógicos, institucionales y éticos; en este sentido López-Vasco, Angulo-Álvarez y Sosa-Zúñiga (2025), advierten que el uso de modelos de IA exige una adecuada formación

docente y criterios claros para su integración, a fin de evitar usos superficiales o descontextualizados.

De igual forma, la literatura evidencia una tendencia a analizar herramientas de manera aislada, priorizando sus características funcionales antes que su impacto en los procesos de aprendizaje, lo que limita la comprensión del fenómeno en su conjunto, Kasneci et al. (2023) señalan que esta dispersión dificulta establecer comparaciones y definir criterios comunes para evaluar su efectividad.

Partiendo de los aspectos descritos anteriormente, el estudio tuvo como propósito analizar los avances sobre el aprendizaje inteligente en la programación a través de una revisión sistemática de la literatura (SLR), alineado a esto, la investigación buscó responder cuatro preguntas orientadas a identificar: i) las principales tendencias del aprendizaje inteligente aplicadas a la programación, ii) las herramientas y enfoques usados en el aprendizaje inteligente, iii) los beneficios y limitaciones reportados en los procesos de enseñanza y aprendizaje, y iv) los desafíos y perspectivas futuras para su implementación en contextos educativos.

La presente revisión sistemática aporta una visión actualizada sobre el aprendizaje inteligente aplicado a la enseñanza de la programación, diferenciándose de estudios previos como los de Mustafa et al. (2024a) y Tlili (2024a) al abordar el vacío de investigación relacionado con la escasa evidencia específica sobre cómo los sistemas adaptativos, la analítica de aprendizaje y los asistentes basados en inteligencia artificial son implementados y utilizados en los procesos de enseñanza y aprendizaje de la programación. En este sentido, el estudio busca contribuir a una comprensión profunda de los avances, aplicaciones y desafíos del aprendizaje inteligente en este campo educativo particular.

### 1.1 Marco Teórico

#### 1.1.1 Fundamentos del aprendizaje inteligente

El aprendizaje inteligente (smart learning) constituye un modelo educativo basado en la integración de tecnologías digitales, inteligencia artificial y análisis de datos para personalizar la enseñanza y mejorar la experiencia de aprendizaje, este enfoque promueve entornos adaptativos capaces de responder a las necesidades individuales de los estudiantes mediante retroalimentación en tiempo real y recursos digitales interactivos. Asimismo, el aprendizaje inteligente se relaciona estrechamente con las Tecnologías de la Información y Comunicación (TIC), permitiendo modalidades educativas flexibles, colaborativas y centradas en el estudiante, estudios recientes señalan que

los sistemas inteligentes favorecen el aprendizaje autónomo y el rendimiento académico mediante el uso de algoritmos adaptativos y plataformas educativas inteligentes (Zizoune et al., 2025; Villegas-Ch et al., 2025).

### 1.1.2 Enseñanza y aprendizaje de la programación

La enseñanza de la programación representa uno de los principales desafíos en la educación tecnológica debido a la complejidad lógica y abstracta que implica para los estudiantes principiantes, pues, diversos estudios evidencian que muchos estudiantes presentan dificultades relacionadas con la comprensión de algoritmos, sintaxis y resolución de problemas computacionales, ante esta situación, se han implementado estrategias pedagógicas apoyadas en plataformas virtuales, gamificación y sistemas de tutoría inteligente que buscan mejorar la motivación y el aprendizaje significativo. Además, el pensamiento computacional se ha convertido en una competencia fundamental para desarrollar habilidades analíticas y de resolución de problemas en contextos educativos digitales (Cao, 2023; Fan et al., 2023).

### 1.1.3 Aplicaciones del aprendizaje inteligente en la programación

Las aplicaciones del aprendizaje inteligente en la programación han permitido desarrollar sistemas capaces de ofrecer tutoría personalizada, retroalimentación automática y adaptación dinámica de contenidos educativos, los Sistema de Tutoría Inteligente (STI) o Intelligent Tutoring Systems (ITS) en inglés, utilizan inteligencia artificial y aprendizaje automático para identificar errores, monitorear el progreso del estudiante y recomendar actividades acordes con su nivel de desempeño. Investigaciones recientes destacan que estas tecnologías

incrementan la participación, motivación y comprensión en cursos de programación, aunque también presentan desafíos relacionados con la privacidad de datos, dependencia tecnológica y capacitación docente. Del mismo modo, las tendencias actuales apuntan hacia el uso de modelos de lenguaje, analítica de aprendizaje y sistemas adaptativos avanzados en entornos educativos de programación (Lai & Lin, 2025; Mittelhessen et al., 2023).

## 1.2 Organización del estudio

El estudio se organiza en cinco secciones, 1) la introducción donde se presenta el problema, los objetivos y la base teórica sobre inteligencia artificial y educación; 2) la metodología describe el desarrollo de la revisión sistemática de literatura basada en Kitchenham; 3) los resultados obtenidos en las bases de datos seleccionadas; 4) la discusión interpreta los resultados comparándolos con estudios previos sobre aprendizaje inteligente; y por último 5) conclusiones.

## 2. Materiales y Métodos

El estudio se desarrolló bajo un enfoque cualitativo, de tipo descriptivo y documental, para ello, se adoptó una SLR, ya que este tipo de metodología permite organizar de manera estructurada la evidencia existente y analizar tendencias emergentes en un campo de estudio, la revisión sistemática es entendida como un proceso riguroso que permite identificar, evaluar e interpretar toda la evidencia disponible relacionada con una pregunta de investigación específica (Kitchenham, 2007a), garantizando así un análisis objetivo y confiable de la información, la misma se desarrolló a través de tres fases las que se visualizan en la Figura 1.

**Figura 1.** Diagrama de la metodología SLR

Fuente: Adaptada por los autores, fases de la metodología SLR de Kitchenham 2007.



## 2.1 Fase 1: Planificación de la revisión

En esta fase se definieron las preguntas de investigación, las bases de datos y se formuló el protocolo de revisión, el cual integra los criterios de inclusión y exclusión y los criterios de evaluación de calidad de los estudios, permitiendo organizar el proceso de búsqueda, filtrado y análisis de la evidencia, pues, una adecuada planificación garantiza la coherencia y la reproducibilidad del proceso, como señalan Mustafa et al., (2024b) alineado a esto, los criterios de inclusión y exclusión establecen los límites del estudio y permiten decidir qué artículos forman parte de la revisión, lo que aporta claridad en la selección de la evidencia (Biblioteca Universidad de Navarra, 2024) a continuación, se desarrollaron en las siguientes etapas de esta fase:

### 2.1.1 Identificación de la necesidad del estudio

La enseñanza de la programación presenta desafíos en la era contemporánea, debido a las exigencias cognitivas que implica, lo que repercute en el desempeño de estudiantes y en las estrategias pedagógicas empleadas por los docentes, de acuerdo con Adil Shah (2025) la evidencia sobre el uso de inteligencia artificial en educación se encuentra dispersa en distintos enfoques y contextos, lo que dificulta obtener una visión clara de su impacto, J. Yang et al. (2025) indican que, si bien estas tecnologías ofrecen ventajas como la personalización del aprendizaje, también plantean desafíos pedagógicos y éticos que requieren un análisis sistemático.

### 2.1.2 Preguntas de investigación

En este contexto, se planteó las siguientes preguntas de investigación:

**PI1:** ¿Qué enfoques de aprendizaje inteligente se han aplicado en la enseñanza de la programación y cuáles son sus características principales?

**PI2:** ¿Cuál es el impacto del aprendizaje inteligente en el rendimiento académico y la adquisición de habilidades de programación en estudiantes?

**PI3:** ¿Qué desafíos, limitaciones y oportunidades se han identificado en la implementación del aprendizaje inteligente en la enseñanza de la programación?

**PI4:** ¿Qué herramientas de aprendizaje inteligente se han utilizado en la enseñanza de la programación?

### 2.1.3 Desarrollo del protocolo de revisión

Este protocolo funciona como una guía que orienta la identificación, selección y análisis de la evidencia, permitiendo que el proceso siga reglas claras desde el inicio, según

Kitchenham, (2007b) el protocolo especifica los métodos que se utilizarán en la revisión y forma parte esencial de la fase de planificación, ya que asegura que el estudio se ejecute con coherencia y posibilidad de réplica, se detallan los criterios de selección de estudios como los de evaluación de calidad a continuación:

### 2.1.4 Criterios de selección de estudios

El desarrollo de los criterios de inclusión se realizó con el fin de establecer reglas claras para seleccionar los estudios que formarían parte de la revisión, considerando aspectos como el tipo de documento, el periodo de publicación y la relación con el tema de estudio, los mismos que permiten definir qué investigaciones cumplen con las condiciones necesarias para ser analizadas y facilitan un proceso de selección coherente, además, su definición previa ayuda a delimitar el alcance del estudio y a reducir sesgos en la selección de la evidencia, según el manual del Joanna Briggs Institute (JBI), los criterios de inclusión deben estar bien definidos y servir como guía para decidir qué fuentes se integran en la revisión, lo que aporta consistencia al proceso (Aromataris et al., 2024), los criterios de selección se muestra en la Tabla 1:

**Tabla 1.** Criterios de selección de estudios  
Fuente: Los autores.

Tipo de criterio	Descripción
Inclusión	Estudios publicados entre 2022–2026
Inclusión	Se incluirán únicamente artículos científicos originales publicados en revistas indexadas
Inclusión	Investigaciones sobre aprendizaje inteligente en educación y programación
Inclusión	Artículos revisados por pares
Exclusión	Estudios fuera del contexto educativo
Exclusión	Se excluirán todos los documentos que no correspondan a artículos científicos originales, tales como tesis, libros, revisiones, editoriales y resúmenes de congresos
Exclusión	Documentos no científicos
Exclusión	Artículos sin acceso completo

### 2.1.5 Criterios de evaluación de calidad

Se definieron criterios de evaluación de calidad que permiten analizar la validez, coherencia y aporte de cada investigación seleccionada, en este proceso se revisaron aspectos como el diseño de investigación, la relación entre objetivos y resultados, la pertinencia del estudio con el tema de la inteligencia artificial

en la enseñanza de la programación, así como la claridad en la presentación de los hallazgos, se consideró la calidad de la fuente y el reconocimiento de limitaciones por parte de los autores.

Este enfoque se respalda en estudios como el de Bolaño-García & Duarte-Acosta (2023), quienes señalan que la evaluación de calidad permite identificar beneficios, tendencias y limitaciones del uso de la inteligencia artificial en educación, lo que fortalece la interpretación de los resultados y el valor de las conclusiones, en este sentido, y con el fin de organizar estos criterios de manera estructurada, se estableció una escala de valoración de cinco niveles que clasifica las publicaciones según su rigor metodológico y nivel de evidencia, como se presenta en la Tabla 2:

**Tabla 2.** Criterios de evaluación de calidad

Fuente: Los autores.

Puntuación	Criterio
1	Constituyen estudios descriptivos o conceptuales sobre aprendizaje inteligente en la enseñanza de la programación, sin una base metodológica clara ni evidencia empírica.
2	Presentan aportes teóricos o revisiones generales sobre herramientas o enfoques de aprendizaje inteligente aplicados a la programación, con escasa profundidad metodológica.
3	Incluyen propuestas metodológicas, diseños de intervención o aplicaciones de herramientas de aprendizaje inteligente en contextos de enseñanza de la programación, debidamente fundamentadas.
4	Presentan experiencias prácticas de implementación de herramientas de aprendizaje inteligente en la enseñanza de la programación en contextos reales de aprendizaje, con análisis de resultados.
5	Además de lo contemplado en el nivel 4, evalúan rigurosamente la implementación de herramientas de aprendizaje inteligente IA, mostrando resultados claros, comparables y replicables sobre el aprendizaje de la programación.

### 2.1.6 Fuentes de información

La búsqueda de información se realizó en bases de datos científicas reconocidas como Scopus e IEEE Xplore. El uso combinado de estas fuentes permite garantizar una mayor cobertura y calidad en la selección de estudios, reduciendo posibles sesgos, tal como señalan Martínez Lobo & Martínez Romero (2022). Estas bases de datos fueron seleccionadas por su prestigio académico, la rigurosidad de sus procesos de indexación y la amplia cobertura de literatura científica en las áreas de estudio abordadas. Por esta razón, no se consideraron otras bases de datos, ya que Scopus e IEEE Xplore proporcionan un conjunto de publicaciones suficiente, confiable y pertinente para alcanzar los objetivos de la investigación.

### 2.1.7 Estrategia de búsqueda

La estrategia de búsqueda se basó en la combinación de palabras clave relacionadas con el aprendizaje inteligente y la enseñanza de la programación, utilizando operadores booleanos como AND y OR, se muestra en la Tabla 3:

**Tabla 3.** Estrategia de búsqueda por base de datos

Fuente: Los autores.

Base de datos	Ecuación de búsqueda	URL
Scopus	TITLE-ABS-KEY ( ( “artificial intelligence” OR “AI” OR “machine learning” OR “deep learning” ) AND ( “programming education” OR “computer programming” OR coding OR “computer science education” ) AND ( “intelligent tutoring system*” OR “adaptive learning” OR “personalized learning” OR “adaptive system*” OR “smart learning” ) ) AND PUBYEAR > 2021 AND PUBYEAR < 2027 AND ( LIMIT-TO ( OA , “all” ) ) AND ( LIMIT-TO ( DOCTYPE , “ar” ) )	<a href="https://www.scopus.com">https://www.scopus.com</a>
IEEE Xplore	(“All Metadata”: (“artificial intelligence” OR “AI” OR “machine learning” OR “deep learning”) AND (“programming education” OR “computer programming” OR coding OR “computer science education”) AND (“intelligent tutoring system” OR “adaptive learning” OR “personalized learning” OR “adaptive systems” OR “smart learning”))	<a href="https://ieeexplore.ieee.org">https://ieeexplore.ieee.org</a>

## 2.2 Fase 2: Realizar la revisión sistemática

Se realizó la búsqueda en las bases de datos seleccionadas, seguida de un proceso de filtrado por título, resumen y texto completo, este procedimiento permitió seleccionar los estudios relevantes, reduciendo sesgos y asegurando la calidad de la información analizada, se describen las etapas siguientes de este proceso:

### 2.2.1 Identificación de estudios relevantes

La identificación de estudios se realizó a través de la ejecución de búsquedas sistemáticas utilizando las palabras clave definidas previamente, las mismas fueron refinadas mediante la aplicación de filtros específicos, con el fin de ajustar los resultados a los criterios de inclusión establecidos y mejorar su relevancia, como resultado de este proceso, se obtuvo un conjunto preliminar de estudios pertinentes, que fueron considerados para las siguientes fases de evaluación y análisis dentro de la revisión sistemática.



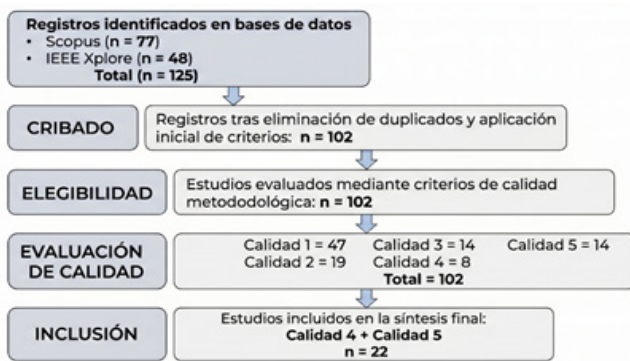
**2.2.2 Proceso de selección de estudios**

La búsqueda inicial identificó 125 registros provenientes de Scopus (n = 77) e IEEE Xplore (n = 48). Posteriormente, se aplicaron los criterios de inclusión y exclusión definidos en el protocolo de revisión, obteniéndose 102 estudios potencialmente relevantes. Estos artículos fueron sometidos a una evaluación de calidad metodológica mediante una escala de cinco niveles. Por último, se seleccionaron únicamente los estudios clasificados en los niveles 4 y 5, obteniéndose una muestra final de 22 investigaciones para el análisis y síntesis de resultados.

Con el propósito de garantizar la transparencia, trazabilidad y reproducibilidad del proceso de selección de estudios, se incorporó el diagrama de flujo PRISMA 2020, el cual permite documentar de manera sistemática las diferentes etapas de identificación, cribado, elegibilidad e inclusión de los registros recuperados. Como se observa en la Figura 2, este procedimiento facilita la comprensión del proceso seguido para la conformación de la muestra final de estudios analizados y fortalece el rigor metodológico de la revisión sistemática al reducir posibles sesgos durante la selección de la evidencia científica.

**Figura 2.** Diagrama de flujo PRISMA 2020 del proceso de selección de estudios.

Fuente: Elaboración propia basada en las directrices PRISMA 2020 (Page et al., 2021).



**2.2.3 Extracción de datos**

Previo a la extracción de datos, la información se organiza mediante la elaboración de una tabla estructurada, en la cual se registran los datos generales de cada documento, conocidos como “datos de pantalla”, la misma se puede visualizar en la Tabla 4, permite sistematizar información clave como el título, objetivo de estudio, tipo de documento, autor(es), año de publicación, país de origen y enlace de acceso (URL o DOI).

El propósito de esta organización inicial es facilitar la

identificación, clasificación y comparación de las fuentes seleccionadas, asegurando un manejo ordenado y coherente de la información, pues, en esta etapa contribuye a mejorar la trazabilidad de los documentos analizados y sienta las bases para un proceso de extracción de datos preciso y eficiente.

**Tabla 4.** Datos de pantalla  
 Fuente: Los autores.

Campo	Descripción
Título	Nombre del documento o estudio analizado
Objetivo de estudio	Propósito de realizar el estudio
Tipo de documento	Deben ser artículos
País	Lugar de procedencia o publicación del documento
URL / DOI	Enlace web o identificador digital del documento
Año de publicación	Fecha en la que fue publicado el documento
Autor(es)	Nombre(s) de quien(es) elaboraron el documento
Base de datos / Fuente	Plataforma o repositorio donde se encontró la información
Resumen	Breve descripción del contenido del documento
Palabras clave	Términos principales que describen el tema del documento

Después del proceso descrito anteriormente se diseñó una matriz de extracción como se ve en la Tabla 5, su propósito fue recopilar información de los estudios seleccionados tras aplicar criterios de evaluación de calidad, lo que permitió organizar los datos de manera estructurada, pues, este proceso facilita la identificación de patrones y tendencias en el uso de tecnologías inteligentes, tal como destaca Adil Shah (2025).

**Tabla 5.** Criterios para la matriz de extracción de datos  
 Fuente: Los autores.

Variable	Descripción
Código	Identificador dentro de la matriz
Título	Conservar el título original
Autor/Año	Identificación del estudio
Nivel de aprendizaje inteligente usado	Adaptativo, generativo, tutor inteligente
Contexto	Nivel educativo
Beneficios	Resultados positivos
Limitaciones	Problemas identificados

## 2.2.4 Síntesis y análisis de la información

La información recopilada fue analizada mediante un enfoque analítico-sintético, lo que permitió identificar tendencias en el aprendizaje inteligente en la enseñanza de la programación, a más de esto, se aplicó un análisis de contenido con un enfoque cualitativo, orientado a interpretar y categorizar la información obtenida, profundizando en los significados, percepciones y patrones emergentes presentes en los datos, lo que facilitó una comprensión más integral del fenómeno estudiado.

## 2.3 Fase 3: Documentar la revisión

Los resultados fueron organizados de manera estructurada para dar respuestas a las preguntas de investigación, Mustafa et al. (2024b), resaltan que este tipo de síntesis es imprescindible para comprender el fenómeno estudiado.

## 3. Resultados y Discusión

### 3.1 Resultados

#### 3.1.1 Caracterización de los estudios recuperados

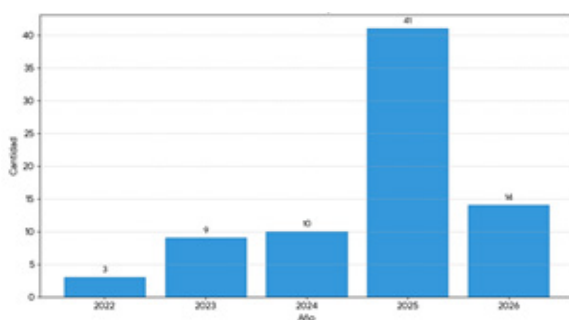
Con el propósito de identificar la evolución de la investigación relacionada con el aprendizaje inteligente aplicado a la enseñanza de la programación, se realizó un análisis descriptivo de la producción científica recuperada en las bases de datos Scopus e IEEE Xplore durante el periodo 2022-2026.

#### Scopus

El análisis de la productividad científica indexada en Scopus evidencia una tendencia de crecimiento sostenido durante el periodo analizado. Como se observa en la Figura 3, el número de publicaciones pasó de 3 documentos en 2022 a 9 en 2023 y 10 en 2024. El incremento más pronunciado se registró en 2025, año en el que se identificaron 41 publicaciones. Para el primer cuatrimestre de 2026 se recuperaron 14 investigaciones, considerando como fecha de corte el 30 de abril de 2026. La base de datos Scopus aportó 77 documentos, constituyéndose como la principal fuente de evidencia científica considerada en esta revisión.

**Figura 3.** Publicaciones por año Scopus

Fuente: Los autores.



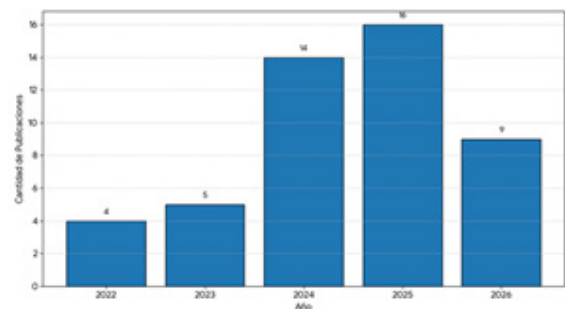
La evolución observada refleja un aumento significativo del interés académico por la incorporación de tecnologías inteligentes, inteligencia artificial y sistemas adaptativos en la enseñanza de la programación. El crecimiento registrado a partir de 2024 sugiere una fase de expansión del campo de estudio asociada al desarrollo de modelos generativos, sistemas tutoriales inteligentes y plataformas de aprendizaje adaptativo.

#### IEEE Xplore

La producción científica identificada en IEEE Xplore presenta una evolución similar. Como se aprecia en la Figura 4, las publicaciones aumentaron de 4 estudios en 2022 a 5 en 2023. Durante 2024 se registraron 14 publicaciones y en 2025 se alcanzaron 16 investigaciones. Para el primer cuatrimestre de 2026 se identificaron 9 estudios utilizando la misma fecha de corte. IEEE Xplore aportó 48 documentos relacionados con el objeto de estudio.

**Figura 4.** Publicaciones por año IEEE Xplore

Fuente: Los autores.



La tendencia observada confirma el creciente interés de la comunidad científica por el desarrollo de soluciones basadas en inteligencia artificial orientadas al fortalecimiento de los procesos de enseñanza y aprendizaje de la programación. La presencia constante de publicaciones recientes evidencia que se trata de una línea de investigación activa, caracterizada por la incorporación de sistemas tutoriales inteligentes, modelos de lenguaje, aprendizaje adaptativo y entornos inmersivos.

#### 3.1.2 Resultados de la evaluación de calidad

Una vez concluido el proceso de selección de estudios descrito en la Figura 2, se procedió a la evaluación de calidad metodológica de los 102 artículos potencialmente relevantes. Para ello, se aplicó la escala de valoración de cinco niveles definida en el protocolo de revisión, con el propósito de determinar el rigor metodológico, la pertinencia y la solidez de la evidencia aportada por cada investigación.

Los resultados muestran una distribución heterogénea de la calidad metodológica de los estudios analizados. Como se observa en la Tabla 6, 47 artículos fueron clasificados en el nivel 1, 19 en el nivel 2, 14 en el nivel 3, 8 en el nivel 4 y 14 en el nivel 5. La concordancia entre los evaluadores fue determinada mediante el coeficiente Kappa de Cohen, obteniéndose un valor de  $\kappa = 0.82$ ,

indicador de una concordancia casi perfecta. Los casos en los que existieron discrepancias fueron revisados nuevamente hasta alcanzar consenso en la clasificación definitiva.

**Tabla 6.** Resultados de aplicar criterios de calidad  
Fuente: Los autores.

Criterios de Calidad	Número de Artículos
1	47
2	19
3	14
4	8
5	14
Total	102

La distribución observada en la Tabla 6, evidencia que 47 de los 102 estudios evaluados (46,08 %) fueron clasificados en el nivel más bajo de calidad metodológica. Este resultado sugiere que una proporción importante de la producción científica disponible aún se concentra en estudios descriptivos o conceptuales que presentan escasa evidencia empírica y limitada validación en contextos educativos reales. Este comportamiento es característico de áreas de investigación emergentes que atraviesan procesos de consolidación teórica y metodológica.

Los niveles 4 y 5 concentraron 22 investigaciones caracterizadas por presentar implementaciones reales, análisis sistemáticos de resultados y evidencia empírica sólida sobre la aplicación de tecnologías inteligentes en la enseñanza de la programación. Estas investigaciones proporcionan resultados comparables, verificables y metodológicamente consistentes, condiciones necesarias para sustentar una síntesis rigurosa de la evidencia científica.

Los estudios clasificados en el nivel 3 fueron considerados durante la evaluación de calidad debido a que presentaban propuestas metodológicas fundamentadas y aplicaciones potencialmente relevantes. Sin embargo, dichos estudios no fueron incorporados a la matriz de extracción debido a que, en la mayoría de los casos, no reportaban validaciones empíricas suficientemente robustas ni resultados comparables con los observados en los niveles superiores. La exclusión de estos documentos respondió al propósito de fundamentar la síntesis exclusivamente en investigaciones respaldadas por evidencia práctica y procedimientos metodológicos consolidados.

### 3.1.3 Estudios seleccionados para la síntesis de evidencia

La muestra definitiva quedó conformada por 22 estudios

clasificados en los niveles 4 y 5 de calidad metodológica, los mismo fueron seleccionados debido a que presentaban implementaciones reales de tecnologías de aprendizaje inteligente, procedimientos metodológicos claramente definidos y resultados respaldados por evidencia empírica obtenida en contextos educativos. La selección de estudios con los niveles más altos de calidad permitió garantizar una base sólida para el análisis, reduciendo el riesgo de incorporar evidencia insuficientemente validada o con limitaciones metodológicas significativas.

La información recuperada fue sistematizada mediante la matriz de extracción de datos presentada en la Tabla 7, pues, este procedimiento permitió organizar de manera estructurada variables relacionadas con los enfoques de aprendizaje inteligente implementados, los contextos educativos de aplicación, las herramientas tecnológicas utilizadas, los beneficios obtenidos y las limitaciones reportadas por los autores. La utilización de esta matriz de extracción facilitó la comparación entre investigaciones y permitió identificar similitudes, diferencias y tendencias recurrentes dentro de la literatura analizada.

Los estudios seleccionados proceden de diversos contextos educativos y tecnológicos, incluyendo educación secundaria, educación superior, formación profesional y entornos de aprendizaje asistidos por inteligencia artificial. Asimismo, abarcan diferentes enfoques de aprendizaje inteligente, tales como sistemas de tutoría inteligente, plataformas adaptativas, modelos basados en grandes modelos de lenguaje, sistemas de recomendación personalizados y entornos inmersivos apoyados por realidad virtual y gamificación. Esta diversidad permitió obtener una visión amplia sobre las formas en que las tecnologías inteligentes están siendo incorporadas en la enseñanza de la programación.

El análisis de los estudios también permitió identificar patrones comunes relacionados con los efectos del aprendizaje inteligente sobre el rendimiento académico, la motivación, el pensamiento computacional, la resolución de problemas y las habilidades de programación. De igual manera, se identificaron limitaciones recurrentes asociadas con la dependencia excesiva de herramientas generativas, la precisión de las respuestas proporcionadas por los sistemas basados en inteligencia artificial y la necesidad de una adecuada mediación docente para garantizar un uso pedagógicamente pertinente de estas tecnologías.

La evidencia sintetizada a partir de los 22 estudios seleccionados proporcionó información suficiente para responder las preguntas de investigación planteadas en el estudio. En particular, permitió identificar los principales enfoques de aprendizaje inteligente aplicados a la programación, analizar su impacto en los procesos

de enseñanza y aprendizaje, reconocer las oportunidades y desafíos asociados a su implementación y determinar las

herramientas tecnológicas que actualmente presentan mayor presencia en este campo de investigación.

**Tabla 7.** Matriz de extracción de datos  
Fuente: Los autores.

Código	Título en Español	Autor/Año	Nivel de aprendizaje inteligente usado	Contexto	Beneficios	Limitaciones
A1	SP-TeachLLM: Un marco impulsado por LLM para la educación personalizada y adaptativa en programación	Huang et al. (2025)	Generativo y adaptativo	Educación en programación / computer science	Mejora el dominio de conocimiento, la resolución de problemas y el desempeño docente; propone tutoría inteligente personalizada y adaptable	El problema identificado es que muchas aplicaciones con LLM siguen siendo limitadas, con poca planificación instruccional sistemática y escasa adaptación a perfiles diversos de estudiantes
A2	Un juego adaptativo de realidad virtual para la enseñanza de programación utilizando mapas cognitivos difusos y modelos pedagógicos	Maroukias et al. (2025)	Adaptativo	Educación en programación con realidad virtual para Java	Mejora el rendimiento en tareas, los resultados de conocimiento y reduce errores mediante personalización en tiempo real	Se señala que muchas experiencias VR no integran bien modelos pedagógicos ni se adaptan al estilo de aprendizaje individual
A3	Mejorando el rendimiento de los estudiantes en programación: un enfoque integrado de mapas mentales y chatbot de IA generativa	Ye et al. (2025)	Generativo	Estudiantes de séptimo grado en China; aprendizaje de programación	Mejora el rendimiento en programación y el pensamiento computacional; el mapeo progresivo resultó más eficaz	Se advierte el riesgo de sobredependencia del chatbot, lo que puede debilitar el pensamiento independiente
A4	Aprendizaje de programación apoyado por inteligencia artificial generativa: efectividad del aprendizaje y competencias fundamentales	Li et al. (2025)	Generativo	Curso de programación en una escuela vocacional de China	Mejora significativamente los resultados de aprendizaje, la motivación, la autoeficacia y las competencias 5C	El estudio parte del problema de una enseñanza centrada en lo técnico, con poca guía práctica y escasa personalización para estudiantes con niveles distintos
A5	Educación adaptativa en programación basada en IA para estudiantes socialmente desfavorecidos: reduciendo la brecha digital	Katona & Gyonyoru (2025)	IA adaptativa basada en ChatGPT y algoritmos de aprendizaje automático	Educación en programación para estudiantes socialmente desfavorecidos en universidades de Hungría	Mejoró el rendimiento académico, el compromiso cognitivo, emocional y conductual; personalización del aprendizaje; retroalimentación inmediata	Duración corta del estudio (13 semanas), posible sesgo de selección, dependencia de datos auto reportados y riesgo de sesgo algorítmico
A6	PyChatAI: Mejorando las habilidades de programación en Python, un estudio empírico de un sistema de aprendizaje inteligente	Alanazi, Soh, et al. (2025)	Generativo y tutor inteligente	Python para principiantes en Arabia Saudita	Da retroalimentación en tiempo real, apoyo personalizado y mejora especialmente la comprensión teórica, escritura de código y depuración	El problema de fondo es que los principiantes tienen dificultades con la depuración y muchas herramientas existentes no son proactivas ni están alineadas al currículo
A7	Examinando la influencia de la IA en la educación de programación en Python: un estudio empírico y análisis de la aceptación estudiantil mediante TAM3	Alanazi, Li, et al. (2025)	Generativo y tutor inteligente	Estudiantes mujeres de informática en Jouf University, Arabia Saudita	Muestra que la utilidad percibida y la facilidad de uso impulsan la intención de uso, valida la aceptación de herramientas bilingües de IA	El alcance es específico, contexto bilingüe y femenino, por lo que la generalización a otros grupos puede ser limitada, además, el impacto de la influencia social y factores cognitivos fue bajo



A8	PyTutor: eficacia de un sistema de tutoría inteligente basado en ChatGPT para la enseñanza de programación	A. C. M. Yang et al. (2024)	Tutor inteligente	Estudiantes principiantes de Python	El estudio ofrece pistas estructuradas, explicaciones y guía continua, mejora participación, tasas de completitud y éxito, sobre todo en estudiantes con bases débiles	Se identifica el riesgo de sobredependencia del sistema, lo que puede frenar el desarrollo de la resolución independiente de problemas
A9	Una aplicación móvil inteligente basada en ChatGPT para mejorar las habilidades de programación en C# en estudiantes de instituciones educativas	Atta et al. (2025)	Generativo y tutor inteligente	Educación superior, aprendizaje de C# en instituciones educativas	Proporciona asistencia personalizada, depuración en tiempo real y mejora el rendimiento estudiantil con apoyo móvil accesible	El artículo indica la necesidad de un uso guiado y controlado de IA, porque el acceso libre a ChatGPT puede no favorecer un aprendizaje estructurado
A10	PerFuSIT: estrategias personalizadas de lógica difusa para la tutoría inteligente en programación	Chrysafiadi & Virvou (2024)	Adaptativo y tutor inteligente	Estudiantes universitarios de informática en Grecia	Selecciona estrategias de tutoría según desempeño, errores y tiempo, mejora el rendimiento y la eficacia de interacción con el Intelligent Tutoring System	El reto principal es modelar datos complejos y vagos del estudiante, así como diagnosticar distintos tipos de error y ajustar la tutoría en tiempo real
A11	Evaluación de la experiencia del usuario, adaptabilidad y resultados de aprendizaje de un sistema de tutoría inteligente basado en lógica difusa para programación en estudiantes universitarios de Grecia	Chrysafiadi et al. (2023)	Adaptativo y tutor inteligente	Estudiantes universitarios de informática en Grecia	Mejora satisfacción, aprendizaje, precisión de recomendaciones, eficacia de interacción, compromiso y motivación	El campo de programación exige actualización frecuente y atiende a estudiantes con perfiles heterogéneos, lo que complica la adaptación de la tutoría
A12	Despliegue integral del Centro Educativo de IA para el aprendizaje personalizado y el compromiso estudiantil	Sajja et al. (2025)	Generativo y adaptativo	Educación en ciencias ambientales y multidisciplinar	Mejora la comprensión, el acceso a información compleja y el engagement mediante asistentes conversacionales personalizados	Dependencia de precisión en recuperación de información y riesgos como alucinaciones en respuestas del modelo
A13	Marco de recomendación personalizada utilizando LLM y prompting Chain-of-Thought	Hongthong et al. (2025)	Generativo y adaptativo	Curso de programación	Mejora la personalización, motivación y calidad de recomendaciones mediante razonamiento paso a paso (CoT)	Falta de integración de preferencias multimodales y dependencia de datos estructurados para personalización efectiva
A14	Redes de memoria difusa y esquemas contextuales: mejorando las respuestas de ChatGPT	Troussas et al. (2025)	Generativo, adaptativo y tutor inteligente	Educación en programación (Java, nivel universitario)	Mejora la personalización, memoria contextual y calidad de respuestas adaptativas	Los sistemas previos carecen de memoria contextual y priorización de información, afectando la adaptabilidad
A15	Aprendizaje adaptativo con factores humanos e inteligencia artificial	Lin et al. (2025)	Adaptativo	Educación en programación	Mejora motivación, interés y rendimiento mediante integración de factores psicológicos y ergonomía	No establece causalidad fuerte ni comparación con sistemas no adaptativos

A16	Marco de aprendizaje basado en LLM para retroalimentación adaptativa en entornos XR gamificados	Gianni et al. (2025)	Generativo y adaptativo	Educación en informática con XR y gamificación	Incrementa motivación (22%) y reduce errores mediante feedback en tiempo real	Sistemas actuales carecen de personalización dinámica y feedback adaptativo en entornos XR
A17	Un enfoque basado en conocimiento para la creación personalizada de exámenes mediante IA	Wu & Wan (2025)	Adaptativo	Educación en programación (evaluación)	Genera evaluaciones personalizadas según nivel del estudiante, mejora aprendizaje	Sistemas tradicionales no relacionan dificultad con habilidades del estudiante ni adaptan dinámicamente
A18	ITS-CAL: sistema de tutoría inteligente para codificación y aprendizaje	Lai & Lin (2025b)	Generativo y tutor inteligente	Educación superior en programación	Proporciona feedback inmediato y mejora tasas de aprobación cuando se usa moderadamente	Uso excesivo reduce habilidades de resolución independiente y dependencia del sistema
A19	La inteligencia artificial apoyando el aprendizaje independiente de los estudiantes	Hartley et al. (2024)	Generativo	Aprendizaje autónomo de programación	Facilita aprendizaje autodirigido, planificación y comprensión de conceptos	Limitaciones en evaluación, interacción y dependencia de habilidades metacognitivas del estudiante
A20	¿Puede la inteligencia artificial ayudar en la educación de ciencias de la computación? Un enfoque de metaanálisis	Tlili (2024b)	Tutor inteligente y adaptativo	Educación en ciencias de la computación (meta-análisis)	Evidencia un gran efecto positivo de la IA (g=1.36), especialmente STI	Falta de estudios específicos en CS y variabilidad según duración y contexto
A21	Revisión sistemática de literatura sobre aplicaciones de LLM en la evaluación de la enseñanza de programación	Pereira & Ferreira Mello (2025)	Generativo	Evaluación en educación en programación	Mejora feedback automático, debugging y aprendizaje personalizado	Riesgo de sobredependencia y necesidad de supervisión humana para aprendizaje profundo
A22	ProgTutor: un marco basado en robótica para apoyar la enseñanza y aprendizaje de fundamentos de programación	Ortega-Morla et al. (2025)	Adaptativo y tutor inteligente	Educación secundaria en programación con robótica	Mejora aprendizaje práctico, feedback automático y rutas personalizadas	Aún en fase piloto; falta evidencia de impacto a largo plazo y adopción en sistemas educativos formales

### 3.1.4 Respuestas a preguntas de investigación

#### PI1: ¿Qué enfoques de aprendizaje inteligente se han aplicado en la enseñanza de la programación y cuáles son sus características principales?

Los enfoques identificados se agrupan en cinco grandes líneas; 1) los tutores inteligentes basados en LLM y chatbots ofrecen explicación de código, pistas graduadas, retroalimentación inmediata y tutoría conversacional, con especial eficacia para principiantes y para escenarios donde el acompañamiento humano no es constante (Yang et al., 2024; Alanazi et al., 2025a; Alanazi et al., 2025b; Hartley et al., 2024; Pereira & Mello, 2025), 2) los sistemas adaptativos con lógica difusa o modelos de conocimiento priorizan la personalización mediante perfiles de desempeño, errores, frecuencia de ayuda y tiempo de resolución, lo que permite seleccionar estrategias instruccionales más finas y explicar mejor la decisión del sistema (Chrysafiadi & Virvou, 2024; Chrysafiadi et al., 2023; Troussas et al., 2025; Wu & Wan, 2025), 3) los entornos inmersivos VR/XR con gamificación introducen aprendizaje experiencial, retos progresivos, motivación y apoyo contextual en tiempo real (Maroungkas et al., 2025; Gianni et al., 2025), 4) las plataformas de recomendación y evaluación personalizada ajustan recursos, actividades y pruebas

a las necesidades del estudiante, incluyendo CoT prompting, bancos de preguntas adaptativos y sistemas de feedback automatizado (Hongthong et al., 2025; Wu & Wan, 2025; Pereira & Mello, 2025), 5) aparecen propuestas de augmentación docente y simulación aplicada, donde la IA no reemplaza al profesor, sino que lo asiste mediante rutas dinámicas, evaluación automática y apoyo al trabajo práctico (Ortega-Morla et al., 2025; Lin et al., 2025).

**Tabla 8.** Características de los enfoques usados en la enseñanza de la programación  
Fuente: Los autores.

Categoría	Enfoque	Características principales	Documento
1	LLM/chatbots/ STI	Retroalimentación inmediata, explicación de código, tutoría conversacional, pistas graduadas, soporte continuo	Huang et al., 2025; Alanazi et al., 2025a; Yang et al., 2024; Hartley et al., 2024; Pereira & Mello, 2025



**Tabla 9.** Impacto del aprendizaje inteligente  
 Fuente: Los autores.

Resultado observado	Hallazgo principal	Evidencia cuantitativa reportada	Estudios
Rendimiento académico	Mejora en puntajes, logro y aprobación	Tasa de aprobación de 72.22 % con uso moderado de ITS-CAL	Ye et al., 2025; Alanazi et al., 2025a; Atta et al., 2025; Li et al., 2025; Lai & Lin, 2025
Habilidades técnicas	Mejor escritura de código, depuración y resolución de problemas	Evidencia positiva, aunque los estudios revisados no reportan tamaños de efecto comparables en los fragmentos analizados	Yang et al., 2024; Alanazi et al., 2025a; Hartley et al., 2024; Pereira & Mello, 2025
Variables afectivas	Aumento de motivación, autoeficacia, interés y compromiso	Incremento de motivación de 22 % y reducción de reintentos superior al 40 %	Li et al., 2025; Lin et al., 2025; Gianni et al., 2025; Sajja et al., 2025
Personalización efectiva	Mayor beneficio para estudiantes con menor base o necesidades específicas	Utilidad percibida → intención de uso ( $\beta = 0.446$ ; $p < .001$ ); facilidad de uso → intención de uso ( $\beta = 0.243$ ; $p = .005$ )	Yang et al., 2024; Lai & Lin, 2025; Chrysafiadi et al., 2023; Alanazi et al., 2025b
Efecto global	La IA muestra un impacto muy grande en CS; los STI son los más potentes	Tamaño de efecto global $g = 1.36$ ( $p < .001$ ); ITS $g = 1.45$ ; meta-análisis de 28 estudios y 2765 participantes	Tlili, 2024

Los tutores inteligentes también muestran efectos importantes, aunque con matices, PyTutor benefició especialmente a estudiantes con bases débiles, elevando la participación y el éxito en tareas, aunque el estudio advierte riesgo de dependencia (Yang et al., 2024), ITS-CAL (Intelligent Tutoring System for Coding and Learning) mostró que el uso moderado se asoció con la mejor tasa de aprobación, mientras que el uso excesivo redujo la autonomía del estudiante (Lai & Lin, 2025), PerFuSIT y el ITS difuso evaluado en Grecia mejoraron la eficacia de la interacción y la satisfacción, confirmando que la adaptación bien

2	Adaptación difusa y basada en conocimiento	Modelado del estudiante, manejo de incertidumbre, selección de estrategias, memoria contextual	Chrysafiadi & Virvou, 2024; Chrysafiadi et al., 2023; Troussas et al., 2025; Wu & Wan, 2025
3	VR/XR y gamificación	Inmersión, motivación, balance retro-habilidad, feedback en tiempo real	Maroungkas et al., 2025; Gianni et al., 2025
4	Recomendación y evaluación personalizada	CoT prompting, generación de pruebas, feedback automático, ajuste al nivel del alumno	Hongthong et al., 2025; Wu & Wan, 2025; Pereira & Mello, 2025
5	Augmentación docente y simulación	Evaluación automática, rutas personalizadas, foco en el profesor como mediador	Lin et al., 2025; Ortega-Morla et al., 2025

Se debe decir que el análisis que se observa en la Tabla 8, no presenta un único modelo de “aprendizaje inteligente”, sino una convergencia de arquitecturas que comparten cuatro rasgos: personalización, retroalimentación inmediata, adaptación al desempeño y soporte escalable, pues, el aporte más consistente es que la IA se usa para convertir la enseñanza de programación en una experiencia situada, más interactiva y menos rígida que la instrucción tradicional (Huang et al., 2025; Ye et al., 2025; Li et al., 2025; Alanazi et al., 2025b; Lin et al., 2025; Ortega-Morla et al., 2025).

**PI2: ¿Cuál es el impacto del aprendizaje inteligente en el rendimiento académico y la adquisición de habilidades de programación en estudiantes?**

La evidencia es mayoritariamente favorable, ya que en los estudios experimentales y cuasi experimentales, el aprendizaje inteligente mejora el rendimiento académico, la comprensión de conceptos, la escritura de código, la depuración, el pensamiento computacional, la autoeficacia y la motivación, un ejemplo de esto es, el enfoque combinado de mapeo mental y GenAI produjo mejores resultados que el modelo tradicional en programación y pensamiento computacional (Ye et al., 2025), PyChatAI mejoró de forma significativa la comprensión teórica, la escritura y la depuración en Python (Alanazi et al., 2025a), mientras que la aplicación móvil con ChatGPT para C# reportó mejoras estadísticamente significativas en el desempeño estudiantil (Atta et al., 2025), alineado a esto, la intervención de GAI en programación elevó aprendizaje, motivación, autoeficacia y competencias 5C (Li et al., 2025), estos resultados se presentan en la Tabla 9.

calibrada si favorece el aprendizaje (Chrysaftadi & Virvou, 2024; Chrysaftadi et al., 2023), a escala agregada, el metaanálisis de Tlili (2024c) encontró un tamaño de efecto grande de la IA sobre el aprendizaje en ciencias de la computación, siendo alto para los sistemas tutoriales inteligentes.

En términos pedagógicos, el impacto no se limita a “saber más código”, pues, varios estudios muestran mejora en autorregulación, motivación sostenida, manejo de errores y confianza para resolver tareas, esto sugiere que el aprendizaje inteligente no solo acelera el rendimiento, sino que también fortalece condiciones psicológicas y cognitivas necesarias para persistir en programación, un campo reconocido por su alta carga cognitiva (Hartley et al., 2024; Lin et al., 2025; Gianni et al., 2025; Hongthong et al., 2025).

PI3: ¿Qué desafíos, limitaciones y oportunidades se han identificado en la implementación del aprendizaje inteligente en la enseñanza de la programación?

Las limitaciones y oportunidades se muestran en la Tabla 10, pues, el principal desafío es la sobredependencia, estudios advierten que, aunque la IA ayuda, un uso excesivo puede debilitar la resolución independiente de problemas, especialmente en principiantes (Ye et al., 2025; Yang et al., 2024; Lai & Lin, 2025; Pereira & Mello, 2025).

A esto se suma el hecho de que la programación requiere razonamiento gradual, práctica deliberada y corrección de errores, por lo que una tutoría automática demasiado directa puede dar respuestas correctas sin desarrollar comprensión profunda, en este punto, Hartley et al. (2024), indican que la eficacia de ChatGPT depende en gran medida de las habilidades metacognitivas del estudiante para pedir ayuda y evaluar límites del sistema.

**Tabla 10.** Limitaciones y oportunidades en la implementación del aprendizaje inteligente en la enseñanza de la programación  
Fuente: Los autores.

Desafío o limitación	Descripción	Oportunidad asociada
Dependencia excesiva	El estudiante puede usar las herramientas de aprendizaje inteligente como atajo y no como apoyo	Diseñar uso guiado, con andamiaje y reflexión
Personalización incompleta	No todos los sistemas capturan bien estilo, nivel o contexto	Modelos más finos con analítica, memoria y datos multimodales
Precisión y alucinaciones	Algunos modelos generan respuestas no fiables	Integrar validación, recuperación y supervisión docente
Falta de evidencia a largo plazo	Muchos estudios son pilotos o casos puntuales	Realizar ensayos longitudinales y comparativos
Brecha de acceso	El soporte inteligente no llega igual a todos los grupos	Ampliar cobertura a contextos desfavorecidos y bilingües

Otra limitación reiterada es la precisión del sistema, Sajja et al. (2025) muestra que la recuperación de información y la generación de respuestas deben controlarse para evitar alucinaciones y respuestas no recuperables, en la misma línea, los estudios basados en lógica difusa y memoria contextual reconocen que todavía hay dificultad para modelar incertidumbre, distinguir errores diversos y sostener adaptaciones robustas en tiempo real (Chrysaftadi & Virvou, 2024; Chrysaftadi et al., 2023; Troussas et al., 2025), Lin et al. (2025) añaden que, aunque la motivación y el interés se asocian con mejores resultados, aún hacen falta estudios causales comparativos para afirmar superioridad frente a sistemas no adaptativos.

Aun así, las oportunidades son muy claras, la IA permite escalar la tutoría, reducir la carga docente, dar soporte a cursos masivos, mejorar la inclusión lingüística y cultural y atender a estudiantes con menor capital académico o tecnológico. PyChatAI y el estudio TAM3 muestran que una herramienta bilingüe y culturalmente situada puede ser especialmente valiosa en contextos específicos (Alanazi et al., 2025a, 2025b), en paralelo, el trabajo de AI-based adaptive programming education para estudiantes socialmente desfavorecidos demuestra que la IA puede contribuir a cerrar brechas socioeducativas en programación (Katona & Katonane Gyonyoru, 2025), programas como ProgTutor, además, muestran una oportunidad estratégica: la IA no sustituye al docente, sino que lo aumenta y le permite concentrarse en los casos que más apoyo requieren (Ortega-Morla et al., 2025).

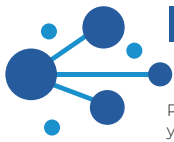
PI4: ¿Qué herramientas de aprendizaje inteligente se han utilizado en la enseñanza de la programación?

Las herramientas y técnicas utilizadas son, 1) los LLM y chatbots: ChatGPT, GPT-4 y sistemas derivados como PyChatAI, PyTutor e ITS-CAL, que se usan para explicar código, responder dudas, generar pistas, apoyar depuración y ofrecer tutoría personalizada (Alanazi et al., 2025a; Yang et al., 2024; Lai & Lin, 2025; Hartley et al., 2024; Pereira & Mello, 2025), 2) aparecen modelos de lógica difusa y memoria contextual, empleados para priorizar información, seleccionar estrategias y adaptar el sistema al progreso del estudiante (Chrysaftadi & Virvou, 2024; Chrysaftadi et al., 2023; Troussas et al., 2025), 3) se utilizan técnicas de recomendación y razonamiento guiado, como chain-of-thought prompting, bancos de preguntas personalizados y sistemas de generación adaptativa de pruebas (Hongthong et al., 2025; Wu & Wan, 2025).

**Tabla 10.** Limitaciones y oportunidades en la implementación del aprendizaje inteligente en la enseñanza de la programación  
Fuente: Los autores.

Herramienta / técnica	Uso en programación	Estudios
ChatGPT / GPT-4 / LLMs	Tutoría conversacional, explicación, hints, evaluación, planificación	Yang et al., 2024; Hartley et al., 2024; Pereira & Mello, 2025; Alanazi et al., 2025b





PyChatAI	Soporte bilingüe y depuración para Python	Alanazi et al., 2025a, 2025b
Lógica difusa	Selección de estrategias y personalización	Chrysaftadi & Virvou, 2024; Chrysaftadi et al., 2023
Fuzzy memory / schemas	Memoria contextual y priorización de interacciones	Troussas et al., 2025
CoT prompting	Razonamiento paso a paso en recomendaciones	Hongthong et al., 2025
MAML, gamificación, XR	Personalización adaptativa en entornos inmersivos	Gianni et al., 2025
Algoritmos genéticos	Generación personalizada de pruebas	Wu & Wan, 2025
Robótica simulada y 3D	Aprendizaje aplicado y rutas adaptativas	Ortega-Morla et al., 2025
RAG y módulos de recuperación	Respuesta más contextual y menos alucinatoria	Huang et al., 2025; Sajja et al., 2025

En la Tabla 11, se observan herramientas de realidad virtual y realidad extendida con gamificación, las cuales fusionan inmersión, motivación y feedback dinámico, así como sistemas de simulación robótica y de aumentación docente que trasladan la IA desde la respuesta textual hacia la interacción práctica y contextualizada (Maroungkas et al., 2025; Gianni et al., 2025; Ortega-Morla et al., 2025), pues las mismas no se limitan al chatbot, estas abarcan desde motores de recomendación hasta módulos de memoria, análisis afectivo, simulación y generación automática de pruebas, lo que confirma una convergencia técnica hacia ecosistemas de aprendizaje inteligente cada vez más integrados (Huang et al., 2025; Lin et al., 2025; Sajja et al., 2025).

### 3.2 Discusión

Los hallazgos de esta revisión muestran que el aprendizaje inteligente en programación se orienta hacia modelos centrados en personalización, tutoría adaptativa y retroalimentación inmediata, tendencia que coinciden con Wang et al. (2024), los que sostienen que la inteligencia artificial educativa ha pasado de sistemas automatizados básicos hacia ecosistemas que integran analítica de aprendizaje, modelos predictivos y adaptación pedagógica en tiempo real, donde los sistemas tutoriales inteligentes representan la línea de investigación con mayor crecimiento.

De forma similar, Hariyanto et al. (2025) identifican que los modelos adaptativos basados en IA mejoran la adecuación del contenido al perfil del estudiante mediante aprendizaje supervisado, refuerzo y análisis multimodal, aspecto que coincide con los resultados encontrados en esta revisión, donde herramientas como PerFuSIT, ITS-CAL y SP-TeachLLM utilizan desempeño, errores y tiempo de respuesta para modificar estrategias instruccionales, Garzón et al. (2025) afirma que los STI y los sistemas adaptativos se han convertido en los enfoques predominantes dentro de la AIED debido a su capacidad para responder a diferencias individuales y ampliar la cobertura educativa, resultado que guarda relación con la convergencia de arquitecturas observada en los estudios seleccionados.

En relación con el impacto académico, los resultados obtenidos coinciden con estudios recientes que reportan mejoras significativas en rendimiento, compromiso y desarrollo de habilidades de programación cuando existe integración estructurada de IA. Nathaniel et al. (2025), encontraron que el uso guiado de GenAI fortalece el pensamiento de orden superior y la lógica de programación cuando las actividades incorporan andamiaje pedagógico y retroalimentación progresiva, pues, estos efectos mantienen relación con los hallazgos de esta revisión, donde PyChatAI, PyTutor y los sistemas con CoT prompting mejoraron depuración, resolución de problemas y comprensión conceptual, alineado a esto, Phung et al. (2023), demostraron que GPT-4 alcanza niveles cercanos al desempeño de tutores humanos en escenarios de programación introductoria, aunque todavía presenta dificultades en razonamiento complejo y corrección contextual.

Respecto a las limitaciones y desafíos, la evidencia encontrada presenta coincidencias importantes con investigaciones actuales sobre riesgos asociados al uso de IA generativa en educación, Franklin et al. (2025) señalan que muchos ejercicios tradicionales de programación pueden resolverse con herramientas generativas sin que exista aprendizaje profundo, situación que obliga a replantear métodos de evaluación y diseño instruccional. Este planteamiento guarda relación con los resultados de esta revisión, donde estudios como ITS-CAL, PyTutor y los sistemas basados en ChatGPT advierten que la dependencia excesiva puede afectar autonomía y pensamiento crítico, pues, Zastudil et al. (2023), identificaron preocupación tanto en docentes como en estudiantes sobre el uso indiscriminado de modelos generativos, especialmente por problemas relacionados con integridad académica, sobre-confianza y disminución del razonamiento independiente, a esto se suman problemas de precisión y alucinaciones, situación descrita por Łodzickowski et al. (2023), quienes sostienen que la IA generativa requiere mecanismos de verificación, transparencia y control pedagógico para evitar

respuestas incorrectas o sesgadas dentro de entornos educativos.

A modo de cierre de la fase de discusión se puede decir que el futuro del aprendizaje inteligente en programación se dirige hacia modelos híbridos donde convergen IA generativa, aprendizaje adaptativo, simulación inmersiva y aumentación docente, Létourneau et al. (2025) indican que los sistemas tutoriales inteligentes actuales muestran mejores resultados cuando funcionan como apoyo pedagógico y no como reemplazo total de la interacción humana, aspecto que coincide con las propuestas de ProgTutor y los sistemas adaptativos encontrados en esta revisión, de manera complementaria, Latif et al. (2026) destacan que los STI y los robots tutoring systems representan una alternativa para enfrentar problemas globales de acceso, personalización y acompañamiento educativo, aunque recalcan la necesidad de validación longitudinal y contextos reales de aplicación.

#### 4. Conclusiones

La revisión sistemática permitió identificar que el aprendizaje inteligente en la enseñanza de la programación ha impulsado modelos educativos orientados a la personalización, adaptación y acompañamiento continuo del estudiante, donde las herramientas basadas en inteligencia artificial ajustan actividades, retroalimentación y recursos según el desempeño individual, pues, los estudios analizados muestran que este enfoque favorece procesos de aprendizaje más dinámicos y cercanos a las necesidades reales del alumnado, sobre todo en escenarios donde la programación representa una dificultad frecuente durante las etapas iniciales de formación académica.

Los enfoques con mayor presencia corresponden a sistemas tutoriales inteligentes, modelos adaptativos y herramientas generativas basadas en LLM, los cuales evidencian aportes importantes en comprensión conceptual, resolución de problemas, escritura de código y depuración de errores, la evidencia revisada también demuestra mejoras en motivación, confianza y participación estudiantil, aspecto que permite reconocer que el aprendizaje inteligente no solo impacta en el rendimiento académico, sino también en factores asociados con permanencia y continuidad dentro de cursos de programación.

Entre las principales limitaciones identificadas destacan la dependencia excesiva de herramientas generativas, la posibilidad de respuestas incorrectas o alucinaciones y la ausencia de estudios longitudinales que permitan validar impactos sostenidos en distintos contextos educativos, del mismo modo, la efectividad de estas tecnologías depende del diseño pedagógico y del acompañamiento docente, debido a que el uso sin orientación adecuada puede afectar el razonamiento independiente y reducir procesos de análisis crítico durante la resolución de problemas de programación.

El aprendizaje inteligente tiene un gran potencial para mejorar la enseñanza de la programación al integrar la inteligencia artificial

con el acompañamiento docente de manera complementaria, ya que esto permite ofrecer experiencias más personalizadas, ampliar el acceso al aprendizaje y brindar apoyo académico a los estudiantes, siempre que el uso de la IA esté guiado por objetivos pedagógicos claros que favorezcan la autonomía, la reflexión y el desarrollo progresivo del conocimiento en programación.

#### Agradecimientos

Expresamos nuestro más sincero agradecimiento a las familias de los autores por su apoyo incondicional, comprensión y motivación constante durante el desarrollo de esta investigación, siendo un pilar fundamental para culminar este proceso académico. Asimismo, agradecemos a la Universidad Nacional de Chimborazo por fomentar y fortalecer los procesos investigativos que contribuyen al desarrollo del conocimiento y la formación profesional.

#### Contribución de los autores

**Edison Amador Miguez Gordillo:** Conceptualización, Metodología, Investigación, Software de análisis, Redacción-revisión, edición del artículo, y borrador original del artículo.  
**Patricio Ricardo Humanante Ramos:** Conceptualización, Metodología, Redacción-revisión, edición del artículo.

#### Conflictos de interés

Los autores declaran no tener ningún conflicto de interés.

#### Referencias bibliográficas

- Adil Shah. (2025). Artificial Intelligence and Its Impact on Education and Digital Learning. *International Journal of Research & Technology*, 13(S4), 511–517. <https://doi.org/10.64882/ijrt.v13.iS4.824>
- Alanazi, M., Li, A., Samra, H., & Soh, B. (2025). Examining the Influence of AI on Python Programming Education: An Empirical Study and Analysis of Student Acceptance Through TAM3. *Computers*, 14(10), 411. <https://doi.org/10.3390/computers14100411>
- Alanazi, M., Soh, B., Samra, H., & Li, A. (2025). PyChatAI: Enhancing Python Programming Skills—An Empirical Study of a Smart Learning System. *Computers*, 14(5), 158. <https://doi.org/10.3390/computers14050158>
- Amo-Filva, D. (2023). Editorial Consolidar aulas inteligentes para configurar espacios de aprendizaje Editorial Consolidating smart classrooms for shaping learning spaces. (2), 3767. <https://doi.org/10.17345/ute.2023.3767>
- Aromataris, E., Lockwood, C., Porritt, K., Pilla, B., & Jordan, Z. (Eds.). (2024). *JB1 Manual for Evidence Synthesis*. JBI.



<https://doi.org/10.46658/JBIMES-24-01>

- Atta, A., Esmat, M., Amasha, N., Elayat, E., & ElSaid, W. K. (2025). A Smart ChatGPT Mobile Application for Improving C# Programming Skills for Students in Educational Institutions. *Qubahan Academic Journal*, 5(2), 49–62. <https://doi.org/10.48161/qaj.v5n2a1772>
- Biblioteca Universidad de Navarra. (2024). Revisiones sistemáticas: Ejemplos de criterios de inclusión y de exclusión. [https://biblioguias.unav.edu/revisionessistematicas/criterios\\_de\\_inclusion\\_y\\_exclusion](https://biblioguias.unav.edu/revisionessistematicas/criterios_de_inclusion_y_exclusion)
- Bolaño-García, M., & Duarte-Acosta, N. (2023). Una revisión sistemática del uso de la inteligencia artificial en la educación. *Revista Colombiana de Cirugía*. <https://doi.org/10.30944/20117582.2365>
- Cao, C. (2023). Scaffolding CS1 Courses with a Large Language Model-Powered Intelligent Tutoring System. *International Conference on Intelligent User Interfaces, Proceedings IUI*, 229–232. <https://doi.org/10.1145/3581754.3584111>
- Chrysafiadi, K., & Virvou, M. (2024). PerFuSIT: Personalized Fuzzy Logic Strategies for Intelligent Tutoring of Programming. *Electronics*, 13(23), 4827. <https://doi.org/10.3390/electronics13234827>
- Chrysafiadi, K., Virvou, M., Tsihrantzis, G. A., & Hatzilygeroudis, I. (2023). Evaluating the user's experience, adaptivity and learning outcomes of a fuzzy-based intelligent tutoring system for computer programming for academic students in Greece. *Education and Information Technologies*, 28(6), 6453–6483. <https://doi.org/10.1007/s10639-022-11444-3>
- Fan, Z., Noller, Y., Dandekar, A., & Roychoudhury, A. (2024). Software Engineering Educational Experience in Building an Intelligent Tutoring System. <http://arxiv.org/abs/2310.05472>
- Finnie-Ansley, J., Denny, P., Becker, B. A., Luxton-Reilly, A., & Prather, J. (2022). The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. *Proceedings of the 24th Australasian Computing Education Conference*, 10–19. <https://doi.org/10.1145/3511861.3511863>
- Franklin, D., Denny, P., Gonzalez-Maldonado, D. A., & Tran, M. (2025). *Generative AI in Computer Science Education*. Cambridge University Press. <https://doi.org/10.1017/9781009581738>
- Garzón, J., Patiño, E., & Marulanda, C. (2025). Systematic Review of Artificial Intelligence in Education: Trends, Benefits, and Challenges. *Multimodal Technologies and Interaction*, 9(8), 84. <https://doi.org/10.3390/mti9080084>
- Gianni, A. M., Nikolakis, N., & Antoniadis, N. (2025). An LLM based learning framework for adaptive feedback mechanisms in gamified XR. *Computers & Education: X Reality*, 7, 100116. <https://doi.org/10.1016/j.cexr.2025.100116>
- Hariyanto, Kristianingsih, F. X. D., & Maharani, R. (2025). Artificial intelligence in adaptive education: a systematic review of techniques for personalized learning. *Discover Education*, 4(1), 458. <https://doi.org/10.1007/s44217-025-00908-6>
- Hartley, K., Hayak, M., & Ko, U. H. (2024). Artificial Intelligence Supporting Independent Student Learning: An Evaluative Case Study of ChatGPT and Learning to Code. *Education Sciences*, 14(2), 120. <https://doi.org/10.3390/educsci14020120>
- Hongthong, T., Chondamrongkul, N., & Temdee, P. (2025). Personalized Recommendation Framework Using Large Language Model and Chain-of-thought Prompting: A Case Study of a Computer Programming Course. *Journal of Mobile Multimedia*. <https://doi.org/10.13052/jmm1550-4646.2165>
- Huang, S., Sun, Y., & Yu, X. (2025). SP-TeachLLM: An LLM-Driven Framework for Personalized and Adaptive Programming Education. *Information*, 16(12), 1045. <https://doi.org/10.3390/info16121045>
- Kasneçi, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., ... Kasneçi, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103, 102274. <https://doi.org/10.1016/j.lindif.2023.102274>
- Katona, J., & Gyonyoru, K. I. K. (2025). AI-based Adaptive Programming Education for Socially Disadvantaged Students: Bridging the Digital Divide. *TechTrends*, 69(5), 925–942. <https://doi.org/10.1007/s11528-025-01088-8>
- Kitchenham, B. & C. Stuart. (2007a). Guidelines for performing Systematic Literature Reviews in Software Engineering. [https://www.researchgate.net/publication/302924724\\_Guidelines\\_for\\_performing\\_Systematic\\_Literature\\_](https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_)

## Reviews\_in\_Software\_Engineering

- Kitchenham, B. & C. Stuart. (2007b). Guidelines for performing Systematic Literature Reviews in Software Engineering. [https://www.researchgate.net/publication/302924724\\_Guidelines\\_for\\_performing\\_Systematic\\_Literature\\_Reviews\\_in\\_Software\\_Engineering](https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering)
- Köhler, J., Hidalgo, L., & Jara, J. L. (2023). Predicting Students' Outcome in an Introductory Programming Course: Leveraging the Student Background. *Applied Sciences*, 13(21), 11994. <https://doi.org/10.3390/app132111994>
- Lai, C.-H., & Lin, C.-Y. (2025a). Analysis of Learning Behaviors and Outcomes for Students with Different Knowledge Levels: A Case Study of Intelligent Tutoring System for Coding and Learning (ITS-CAL). *Applied Sciences*, 15(4), 1922. <https://doi.org/10.3390/app15041922>
- Lai, C.-H., & Lin, C.-Y. (2025b). Analysis of Learning Behaviors and Outcomes for Students with Different Knowledge Levels: A Case Study of Intelligent Tutoring System for Coding and Learning (ITS-CAL). *Applied Sciences*, 15(4), 1922. <https://doi.org/10.3390/app15041922>
- Latif, E., Liu, V., & Zhai, X. (2026). A systematic review of intelligent and robot tutoring systems: evolution, pedagogical design, and AI-driven classification. *Smart Learning Environments*, 13(1), 1. <https://doi.org/10.1186/s40561-025-00427-9>
- Létourneau, A., Deslandes Martineau, M., Charland, P., Karran, J. A., Boasen, J., & Léger, P. M. (2025). A systematic review of AI-driven intelligent tutoring systems (ITS) in K-12 education. *Npj Science of Learning*, 10(1), 29. <https://doi.org/10.1038/s41539-025-00320-7>
- Li, H.-J., Huang, Q.-R., Wen, L.-P., Chen, W., & Xu, Z.-Z. (2025). Generative Artificial Intelligence Supported Programming Learning: Learning Effectiveness and Core Competence. *SAGE Open*, 15(3). <https://doi.org/10.1177/21582440251377986>
- Lin, K.-Y., Li, M.-H., Lo, F., Huang, H.-C., Matsuno, K., & Watanabe, R. (2025). Adaptive learning with human factors and Artificial Intelligence: associations with training effectiveness in programming education. *International Journal of Industrial Ergonomics*, 110, 103834. <https://doi.org/10.1016/j.ergon.2025.103834>
- Łodzikowski, K., Foltz, P. W., & Behrens, J. T. (2023). Generative AI and Its Educational Implications. <https://doi.org/10.48550/arXiv.2401.08659>
- López-Vasco, F. E., Angulo-Álvarez, M. R. y Sosa-Zúñiga, D. I. (2025). Formación docente en IA Generativa: impacto ético y retos en educación superior. *Alteridad*, 20(2), 166-177. <https://doi.org/10.17163/alt.v20n2.2025.01p-ISSN:1390-325X / e-ISSN:1390-8642http://alteridad.ups.edu.ec>
- Marougkas, A., Troussas, C., Krouska, A., & Sgouropoulou, C. (2025). An adaptive virtual reality game for programming education using fuzzy cognitive maps and pedagogical models. *Smart Learning Environments*, 12(1), 62. <https://doi.org/10.1186/s40561-025-00392-3>
- Martínez Lobo, A. del P., & Martínez Romero, B. (2022). Bases de datos académicas-científicas, su uso para el desarrollo de la investigación y producción de las IES en el sector aeronáutico en Colombia. *Ciencia y Poder Aéreo*, 17(2). <https://doi.org/10.18667/cienciaypoderaereo.736>
- Miguez Gordillo, M. Á. (2025). La inteligencia artificial como recurso didáctico en el proceso de enseñanza y aprendizaje. *Revista Multidisciplinar Epistemología de Las Ciencias*, 2(2), 181–196. <https://doi.org/10.71112/vqb1zx28>
- Mittelhessen, T. H., Reid, S. A., Kammer, F., Kunz, J., Pellekorne, T., Siepermann, M., & Wölfer, J. (2023). ItsSQL: Intelligent Tutoring System for SQL ItsSQL: Intelligent Tutoring System for SQL Working Paper. <https://doi.org/10.48550/arXiv.2311.10730>
- Mustafa, M. Y., Tlili, A., Lampropoulos, G., Huang, R., Jandrić, P., Zhao, J., Salha, S., Xu, L., Panda, S., Kinshuk, López-Pernas, S., & Saqr, M. (2024a). A systematic review of literature reviews on artificial intelligence in education (AIED): a roadmap to a future research agenda. *Smart Learning Environments*, 11(1), 59. <https://doi.org/10.1186/s40561-024-00350-5>
- Mustafa, M. Y., Tlili, A., Lampropoulos, G., Huang, R., Jandrić, P., Zhao, J., Salha, S., Xu, L., Panda, S., Kinshuk, López-Pernas, S., & Saqr, M. (2024b). A systematic review of literature reviews on artificial intelligence in education (AIED): a roadmap to a future research agenda. *Smart Learning Environments*, 11(1), 59. <https://doi.org/10.1186/s40561-024-00350-5>
- Nathaniel, J., Oyelere, S. S., Suhonen, J., & Tedre, M. (2025). Investigating the impact of generative AI integration on the sustenance of higher-order thinking skills and understanding of programming logic. *Computers and Education: Artificial Intelligence*, 9, 100460. <https://doi.org/10.1016/j.caeai.2025.100460>
- Ortega-Morla, J., Leis, A., Mallo, A., Morán-Fernández, L., Guerreiro, S., Paz-López, A., Pérez-Sánchez, B., Sánchez-Maróño, N., Rodríguez-Arias, A., Fontenla-Romero, O., & Bellas, F. (2025). ProgTutor: A Robotic-Based Framework to Support Teaching and Learning of Programming Fundamentals. *IEEE Transactions on Learning Technologies*, 1–15. <https://doi.org/10.1109/TLT.2025.3598041>
- 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>
- Pereira, A. F., & Ferreira Mello, R. (2025). A Systematic Literature Review on Large Language Models Applications in Computer Programming Teaching Evaluation Process. *IEEE Access*, 13, 113449–113460. <https://doi.org/10.1109/ACCESS.2025.3584060>
- Phung, T., Pădurean, V.-A., Cambronero, J., Gulwani, S., Kohn, T., Majumdar, R., Singla, A., & Soares, G. (2023). Generative AI for Programming Education: Benchmarking ChatGPT, GPT-4, and Human Tutors. <http://arxiv.org/abs/2306.17156>
- Sajja, R., Sermet, Y., & Demir, I. (2025). End-to-End Deployment of the Educational AI Hub for Personalized Learning and Engagement: A Case Study on Environmental Science Education. *IEEE Access*, 13, 55169–55186. <https://doi.org/10.1109/ACCESS.2025.3554222>
- Tlili, A. (2024). ¿Puede ayudar la inteligencia artificial (IA) en la educación en ciencias de la computación? Un enfoque metaanalítico. *Revista Española de Pedagogía*, 82(289), 469–490. <https://doi.org/10.22550/2174-0909.4172>
- Topham, L., Atherton, P., Reynolds, T., Hussain, Y., Hussain, A., Kolivand, H., & Khan, W. (2026). Artificial Intelligence in Educational Technology: A Systematic Review of Datasets and Applications. *ACM Computing Surveys*, 58(3), 1–28. <https://doi.org/10.1145/3768312>
- Troussas, C., Krouska, A., Mylonas, P., Sgouropoulou, C., & Voyiatzis, I. (2025). Fuzzy Memory Networks and Contextual Schemas: Enhancing ChatGPT Responses in a Personalized Educational System. *Computers*, 14(3), 89. <https://doi.org/10.3390/computers14030089>
- Villegas-Ch, W., Buenano-Fernandez, D., Navarro, A. M., & Mera-Navarrete, A. (2025). Adaptive intelligent tutoring systems for STEM education: analysis of the learning impact and effectiveness of personalized feedback. *Smart Learning Environments*, 12(1), 41. <https://doi.org/10.1186/s40561-025-00389-y>
- Wang, S., Wang, F., Zhu, Z., Wang, J., Tran, T., & Du, Z. (2024). Artificial intelligence in education: A systematic literature review. *Expert Systems with Applications*, 252, 124167. <https://doi.org/10.1016/j.eswa.2024.124167>
- Wu, Z., & Wan, S. (2025). A Knowledge-Driven Approach to AI-Based Personalized Test Paper Creation in Programming Education. *International Journal of Knowledge Management*, 21(1), 1–21. <https://doi.org/10.4018/IJKM.369825>
- Yang, A. C. M., Lin, J.-Y., Lin, C.-Y., & Ogata, H. (2024). Enhancing python learning with PyTutor: Efficacy of a ChatGPT-Based intelligent tutoring system in programming education. *Computers and Education: Artificial Intelligence*, 7, 100309. <https://doi.org/10.1016/j.caeai.2024.100309>
- Yang, J., Shi, G., Zhu, W., & Sun, Y. (2025). Intelligent technologies in smart education: a comprehensive review of transformative pillars and their impact on teaching and learning methods. *Humanities and Social Sciences Communications*, 12(1), 1239. <https://doi.org/10.1057/s41599-025-05444-0>
- Ye, X., Zhang, W., Zhou, Y., Li, X., & Zhou, Q. (2025). Improving students' programming performance: an integrated mind mapping and generative AI chatbot learning approach. *Humanities and Social Sciences Communications*, 12(1), 558. <https://doi.org/10.1057/s41599-025-04846-4>
- Zastudil, C., Rogalska, M., Kapp, C., Vaughn, J., & MacNeil, S. (2023). Generative AI in Computing Education: Perspectives of Students and Instructors. <http://arxiv.org/abs/2308.04309>
- Zawacki-Richter, O., Müssens, W., & Marin, V. I. (2022). Quality Assurance of Open Educational Resources. In *Handbook of Open, Distance and Digital Education* (pp. 1–19). Springer Singapore. [https://doi.org/10.1007/978-981-19-0351-9\\_43-1](https://doi.org/10.1007/978-981-19-0351-9_43-1)
- Zizoune, A., Dakki, M., Zizoune, A., Hajhouj, S. E. El, Salaheddine, K., & Ziti, S. (2025). Smart Education and Intelligent Learning Systems (pp. 29–36). [https://doi.org/10.1007/978-3-031-74470-9\\_4](https://doi.org/10.1007/978-3-031-74470-9_4)