



UNIVERSIDAD ESTATAL DE MILAGRO

REPÚBLICA DEL ECUADOR

**UNIVERSIDAD ESTATAL DE MILAGRO
FACULTAD DE POSGRADO**

VICERRECTORADO DE INVESTIGACIÓN Y POSGRADO

**ARTÍCULOS PROFESIONALES DE ALTO NIVEL
PREVIO A LA OBTENCIÓN DEL TÍTULO DE:**

**MAGÍSTER EN TECNOLOGÍAS DE LA INFORMACIÓN CON
MENCIÓN EN TRANSFORMACIÓN DIGITAL E INNOVACIÓN**

TEMA:

**Modelo AREM: Una Propuesta Pedagógica para la IA
Generativa en la Enseñanza de Programación**

Autor: Ph.D. Roger Marcelo Freire Avilés

Tutor: MSc. Jorge Luis Vinueza Martínez

Milagro, 2026

Revista digital de investigación

5
9
3

Digital Publisher

CEIT

Volumen 11
Número 4-1
Año 2026/julio
ISSN 2588-0705

Edición Especial

Ecuador | Julio



Modelo AREM: Una Propuesta Pedagógica para la IA Generativa en la Enseñanza de Programación

doi.org/10.33386/593dp.2026.4-1.4232

Roger Marcelo Freire-Avilés*
Jorge Luis Vinueza-Martínez**
Delia Isabel Carrión-León***

Resumen

La integración de la inteligencia artificial generativa (IAG) en los sistemas de tutoría inteligente (STI) ha ampliado el acceso a apoyo personalizado en la enseñanza de programación, pero la evidencia disponible muestra que la sofisticación conversacional de estos sistemas avanza más rápido que su fundamentación pedagógica explícita, una fragmentación que, según la literatura, obstaculiza la consolidación del campo como cuerpo teórico acumulativo. Este estudio sintetiza, a partir de un corpus derivado de una revisión sistemática registrada en PROSPERO (protocolo PRISMA 2020, Scopus) y una búsqueda complementaria centrada en enfoques pedagógicos, los modelos y estrategias reportados en la literatura 2020-2025 sobre tutores inteligentes basados en modelos de lenguaje de gran escala. Se identifican seis patrones recurrentes -andamiaje sin solución directa, retroalimentación formativa multinivel, aprendizaje autorregulado, marcos instruccionales adaptados, aprendizaje experiencial y equidad digital- que rara vez se integran en un marco coherente. A partir de esta síntesis se propone el Modelo AREM (Andamiaje, Retroalimentación, Equidad, Mediación metacognitiva), cuyo aporte no se limita a articular estos patrones, sino que introduce una innovación metodológica: trasladar la lógica configuracional de factores higiénicos y motivadores, desarrollada en la investigación sobre calidad de servicios institucionales, al diseño pedagógico de la tutoría individual, con potencial de transferencia a otros dominios de la IAG educativa. El modelo, pendiente de validación empírica, es pertinente para la educación superior latinoamericana y estructuralmente aplicable a otros sistemas con recursos limitados a nivel internacional, para los que se plantea una agenda de investigación comparada entre países.

Palabras clave: inteligencia artificial generativa; sistemas de tutoría inteligente; modelo pedagógico; enseñanza de programación; retroalimentación formativa.

* Facultad de Posgrado, Escuela de Salud, Maestría en Tecnologías de la información con mención en transformación digital e innovación. Docente e investigador de la Facultad de Ciencias e Ingeniería de la Universidad Estatal de Milagro. Universidad Estatal de Milagro. E-mail: rfreira2@unemi.edu.ec. ORCID: <https://orcid.org/0000-0001-9069-4787> (Autor de Correspondencia)

** Magister en Gerencia de Tecnologías de la Información, Profesor e investigador de la Universidad Estatal de Milagro. Universidad Estatal de Milagro. E-mail: jvinuezam@unemi.edu.ec. ORCID: <https://orcid.org/0000-0003-1966-4395>

*** Magíster en Sistemas de Información Gerencial por la Escuela Superior Politécnica del Litoral (ESPOL). Se desempeña como docente e investigadora en la Facultad de Ciencias e Ingeniería de la Universidad Estatal de Milagro. Universidad Estatal de Milagro. E-mail: dcarrionl@unemi.edu.ec. ORCID: <https://orcid.org/0000-0003-1147-8045>

The AREM Model: A Pedagogical Proposal for Generative AI in Programming Education

Abstract

The integration of generative artificial intelligence (GenAI) into intelligent tutoring systems (ITS) has expanded access to personalized support in programming education, yet available evidence shows that the conversational sophistication of these systems is advancing faster than their explicit pedagogical grounding, a fragmentation that, per the literature, hinders the field's consolidation as a cumulative body of theory. This study synthesizes, from a corpus derived from a systematic review registered in PROSPERO (PRISMA 2020 protocol, Scopus) and a complementary search focused on pedagogical approaches, the models and strategies reported in the 2020-2025 literature on large language model-based intelligent tutors. Six recurring patterns are identified, scaffolding without direct solutions, multilevel formative feedback, self-regulated learning, adapted instructional frameworks, experiential learning, and digital equity, that are rarely integrated into a coherent framework. Building on this synthesis, the study proposes the AREM Model (Scaffolding, feedback, Equity, Metacognitive mediation), whose contribution extends beyond articulating these patterns to introduce a methodological innovation: transferring the configurational logic of hygiene and motivator factors, developed in institutional service-quality research, to the pedagogical design of individual tutoring, with potential transfer to other domains of educational GenAI. The model, pending empirical validation, is relevant to Latin American higher education and structurally applicable to other resource-constrained systems internationally, for which a cross-country comparative research agenda is outlined.

Key words: generative artificial intelligence; intelligent tutoring systems; pedagogical model; programming education; formative feedback.

Introducción y Estado de la Cuestión

La enseñanza de los fundamentos de programación continúa siendo uno de los retos pedagógicos más persistentes de la educación superior contemporánea, debido a las barreras cognitivas asociadas al pensamiento abstracto, la lógica computacional y la depuración sistemática de errores (Elnaffar et al., 2026). Frente a la imposibilidad de ofrecer retroalimentación individualizada a grupos numerosos, los sistemas de tutoría inteligente (STI) han emergido como alternativa para extender el apoyo pedagógico más allá del aula. La irrupción de los modelos de lenguaje de gran escala (LLM) ha transformado las capacidades conversacionales de estos sistemas, acercándolos a la experiencia de un tutor humano experto (Batsaikhan & Correia, 2024).

Esta transformación se comprende mejor a la luz de la trayectoria histórica de los STI. Los sistemas de primera generación, como AutoTutor, ISNAP e ITAP, se organizaban en torno a reglas expertas y representaciones simbólicas estáticas; demostraron que la retroalimentación estructurada y el andamiaje adaptativo mejoran los resultados en programación introductoria, pero su alcance permanecía acotado por la rigidez de las reglas codificadas manualmente (Batsaikhan & Correia, 2024). Los LLM impulsan una evolución hacia arquitecturas híbridas y dinámicas, donde el modelo generativo genera explicaciones contextuales y diálogos socráticos en tiempo real. Sin embargo, esta capacidad ampliada no resuelve limitaciones bien documentadas, como alucinaciones, sesgos en el código generado y dificultades con dialectos locales de error (Ding et al., 2026).

Sin embargo, esta sofisticación técnica no ha estado acompañada de un desarrollo equivalente en su fundamentación pedagógica. La literatura reciente documenta de manera consistente que el diseño de los tutores inteligentes con inteligencia artificial generativa (IAG) prioriza la fluidez conversacional y la generación rápida de respuestas por encima de un diseño instruccional explícito y teóricamente fundamentado (Ding et al., 2026). Esta tensión

entre capacidad técnica y profundidad pedagógica constituye, precisamente, el problema central que aborda el presente artículo.

La evidencia empírica disponible ilustra ambas caras de este fenómeno: los asistentes basados en LLM muestran efectos positivos en el compromiso estudiantil y la comprensión conceptual (Kazemitabaar et al., 2024), pero la sobredependencia y el aprendizaje superficial se han consolidado como uno de los riesgos más documentados, presente en hasta el 65,52 % de los estudios revisados por Elnaffar et al. (2026). Prather et al. (2024) muestran que la IAG puede introducir nuevas dificultades metacognitivas, incluyendo una falsa sensación de competencia, mientras que Dawson et al. (2025) documentan, mediante entrevistas, una disonancia cognitiva entre el uso intensivo de LLM y el reconocimiento de que dicho uso no resulta óptimo para el aprendizaje profundo.

Diversos marcos pedagógicos han comenzado a proponerse para responder a esta brecha. Nathaniel et al. (2025a) desarrollan y validan empíricamente el marco GenAI-Ped, que combina la taxonomía de Bloom, el aprendizaje autorregulado, el diseño universal para el aprendizaje y la zona de desarrollo próximo de Vygotsky, con ganancias en pensamiento crítico y lógica de programación sin comprometer habilidades de orden superior. Liao et al. (2024) y Gomez-Jaramillo et al. (2025) integran de forma similar andamiaje, constructivismo y aprendizaje adaptativo en sistemas con ChatGPT. No obstante, Proust-Androwkha et al. (2025), a partir de 38 estudios, concluyen que existe una carencia generalizada de estrategias pedagógicas explícitamente diseñadas para integrar la IAG en la educación en ciencias de la computación.

Ante este panorama, el presente artículo persigue dos objetivos complementarios. En primer lugar, sintetizar los enfoques y modelos pedagógicos reportados en la literatura reciente (2020-2025) sobre STI con IAG orientados a la enseñanza de programación. En segundo lugar, proponer un modelo pedagógico integrador — el Modelo AREM— que articule los patrones convergentes identificados en dicha síntesis y

ofrezca un marco orientador para el diseño de la próxima generación de tutores inteligentes, con especial atención a su pertinencia para contextos de educación superior latinoamericana, donde la evidencia empírica específica continúa siendo escasa (Vinueza-Martínez, 2026).

Material y Métodos

Enfoque y Fuentes Documentales

Este artículo no reporta una investigación empírica primaria: corresponde a una propuesta pedagógica derivada de una síntesis narrativa de literatura y de un diálogo con marcos teóricos afines, tipología que, conforme a las normas editoriales de la revista, admite epígrafes flexibles distintos de la estructura IMRD, reservada para investigación empírica, o IDC. El estudio se elabora a partir de un corpus documental derivado de dos procesos de búsqueda complementarios. El primero corresponde a una revisión sistemática de literatura registrada prospectivamente en PROSPERO (CRD420261428002), que sigue el protocolo PRISMA 2020 con criterios de elegibilidad estructurados mediante el marco PICO, ejecutada en Scopus el 18 de junio de 2026 (N = 887 registros; N final = 24 estudios primarios).

El segundo proceso corresponde a una búsqueda complementaria ejecutada en Scopus el 24 de junio de 2026, que añadió a la ecuación original un bloque conceptual centrado explícitamente en modelos y enfoques pedagógicos (andamiaje, retroalimentación formativa, autorregulación, metacognición, entre otros descriptores), de la cual se recuperaron 36 registros adicionales. Tras excluir los registros ya presentes en el corpus original y cribar por título y resumen aquellos cuyo dominio no correspondía específicamente a la enseñanza de programación, se evaluaron 22 informes para elegibilidad; de estos, 20 estudios cumplieron el criterio de presentar un modelo, marco o estrategia pedagógica explícita y se incorporaron a la síntesis (ver figura 1).

La ecuación de búsqueda combina cuatro bloques conceptuales mediante el operador

booleano AND, aplicados sobre el campo TITLE-ABS-KEY de Scopus (título, resumen y palabras clave), con los términos de cada bloque unidos internamente mediante el operador OR. Los tres primeros bloques, población, intervención y dominio, reproducen la ecuación original registrada en PROSPERO; el cuarto bloque, centrado explícitamente en enfoques pedagógicos, se añadió específicamente para la búsqueda complementaria de la cual proviene el subconjunto de 20 estudios analizado en este artículo. La tabla 1 detalla los cuatro bloques y sus operadores.

Tabla 1

Bloques conceptuales de la ecuación de búsqueda en Scopus

Bloque	Función	Términos (TITLE-ABS-KEY)
P1	Población (STI)	"intelligent tutoring system" OR "ITS" OR "cognitive tutor" OR "adaptive tutoring" OR "AI tutoring"
P2	Intervención (IAG)	"generative AI" OR "large language model" OR "LLM" OR "GPT" OR "ChatGPT" OR "chatbot" OR "conversational agent"
P3	Dominio (programación)	"programming education" OR "computer science education" OR "coding" OR "introductory programming" OR "CS1" OR "programming tutor"
P4	Enfoque pedagógico (bloque añadido en la búsqueda complementaria)	"pedagogical model" OR "pedagogical framework" OR "pedagogical approach" OR "instructional design" OR "scaffolding" OR "formative feedback" OR "self-regulated learning" OR "metacognition"

Nota: los bloques P1-P3 reproducen la ecuación registrada en PROSPERO (CRD420261428002); el bloque P4 es específico de la búsqueda complementaria de este artículo. Filtros aplicados en ambas búsquedas: PUBYEAR > 2019 AND PUBYEAR < 2026; tipo de documento: artículos, revisiones y actas de conferencia; idioma: inglés y español.

La selección de Scopus como fuente única se sustenta en cuatro criterios: su cobertura interdisciplinaria en ciencias de la computación, ingeniería y educación; su rigor de indexación; su precedente en revisiones sistemáticas del dominio de la IAG en educación; y la superposición sustancial con otras bases especializadas al

aplicar filtros metodológicos rigurosos. Este último punto encuentra respaldo en el análisis de sensibilidad de Vinueza-Martínez (2026), quien documenta un 91 % de superposición entre Scopus/Web of Science y ERIC, con los registros no duplicados excluidos mayoritariamente por carecer de datos empíricos cuantificables, lo que sugiere un sesgo de cobertura mínimo.

El marco PICO estructuró la pregunta que orienta este artículo: Población, estudiantes de programación en educación superior que interactúan con STI basados en IAG; Intervención, STI que incorporan LLM como componente central de su arquitectura pedagógica; Comparación, entre enfoques pedagógicos y contextos de implementación; Resultado, modelos y estrategias reportadas, con énfasis en retroalimentación, andamiaje, autorregulación y equidad. Los criterios de elegibilidad exigieron estudios indexados en Scopus, publicados entre 2020 y 2025, en inglés o español, que reportaran explícitamente un modelo o estrategia pedagógica vinculada al uso de IAG; se excluyeron benchmarks técnicos sin participantes humanos y publicaciones sin arbitraje verificable.

Evaluación de Calidad y Síntesis de Datos

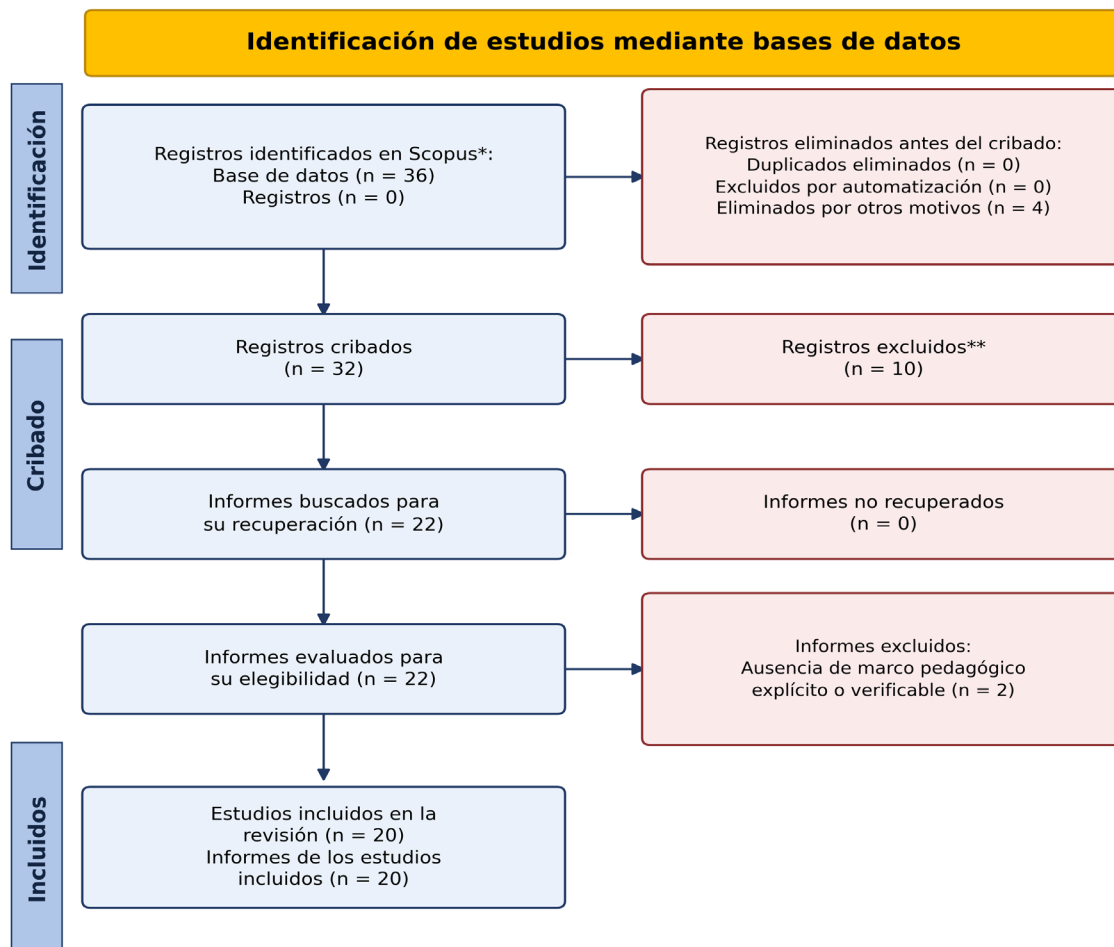
Dado que este artículo constituye una propuesta derivada y acotada, la selección final de los 20 estudios se realizó mediante lectura de título y resumen; y, de texto completo cuando fue necesario, aplicando como criterio la presencia explícita de un marco o estrategia pedagógica verificable, en línea con el umbral MMAT (versión 2018) del corpus original. La extracción y codificación se realizó mediante una plantilla estandarizada, coherente con las doce variables empleadas en la revisión matriz (identificador, autoría, país, tipo de publicación, arquitectura tecnológica, tipo de STI, marco pedagógico declarado, tipo de retroalimentación, muestra, resultados de aprendizaje, desafíos y puntuación de calidad), a la que se añadió el patrón y componente AREM de cada estudio (tabla 4). La codificación inicial se realizó a partir de título y resumen, revisada por los coautores contra la definición operacional de cada categoría.

Para fortalecer su confiabilidad, se verificó de forma independiente a texto completo 15 de los 20 estudios (75,0 %), los de acceso abierto; los 5 restantes permanecen clasificados solo a nivel de título y resumen, limitación declarada. Un segundo codificador ciego obtuvo Kappa de Cohen = 1,0 a nivel de componente AREM (15/15) y Kappa = 0,91 a nivel de patrón (14/15), ambos casi perfectos (Landis y Koch, 1977). La única discrepancia (Proust-Andrewkha et al., 2025, inicialmente patrón 4) se resolvió por segunda lectura, reclasificándose como patrón 1 sin alterar su componente A. La submuestra no incluyó el componente E (equidad), pues su único estudio (Park et al., 2025) está entre los cinco de acceso restringido, lo que limita la generalización a esa categoría.

La heterogeneidad metodológica entre los estudios incluidos, que combinan diseños cualitativos, cuantitativos no aleatorizados y de métodos mixtos, impide la agregación estadística; por ello, la síntesis de datos sigue las directrices SWiM (Synthesis Without Meta-Analysis), empleando una síntesis narrativa estructurada que permite identificar patrones convergentes sin forzar homogeneidad artificial entre estudios de diseño diverso (Campbell et al., 2020). El presente artículo sintetiza, con redacción propia, el subconjunto pedagógicamente relevante de ambos corpus, sin reproducir las tablas de síntesis completas de la revisión matriz.

Figura 1

Diagrama de flujo PRISMA 2020 de la búsqueda complementaria (elaborado sobre la estructura oficial, Page et al., 2021)



* Búsqueda ejecutada exclusivamente en Scopus (bloque P4, búsqueda complementaria).

** Registros excluidos por no corresponder al dominio de programación o ciencias de la computación.

Nota: proceso de cribado correspondiente a la búsqueda complementaria centrada en enfoques pedagógicos (bloque P4), independiente del cribado de la revisión sistemática matriz. Elaborado a partir de la estructura oficial del diagrama de flujo PRISMA 2020, disponible en <http://www.prisma-statement.org/>, bajo licencia CC BY 4.0.

Análisis y Resultados

Distribución de los Estudios Incluidos

El corpus analizado combina el subconjunto pedagógicamente relevante del corpus original (24 estudios) con 20 estudios adicionales de la búsqueda complementaria (bloque P4). Estos 20 estudios se distribuyen de forma creciente en el tiempo: 2 (10,0 %) en 2023, 5 (25,0 %) en 2024 y 13 (65,0 %) en 2025, confirmando que la producción se concentra en el último año analizado. Predominan las actas de

conferencia (12, 60,0 %), seguidas de artículos de revista (6, 30,0 %) y revisiones (2, 10,0 %), lo que sugiere que la innovación pedagógica se comunica primero en foros de divulgación rápida. La tabla 2 resume esta distribución según el componente AREM predominante, anticipando la síntesis de los apartados siguientes.

Tabla 2
Distribución de los 20 estudios de la búsqueda complementaria por año, tipo de publicación y componente AREM predominante

Componente AREM predominante	n	%	Estudios representativos
A — Andamiaje sin solución directa	12	60,0	Liao et al. (2024); Nathaniel et al. (2025a)
R — Retroalimentación formativa multinivel	3	15,0	Azaiz et al. (2023); Edwards et al. (2025)
E — Equidad y alfabetización digital previa	1	5,0	Park et al. (2025)
M — Mediación metacognitiva	4	20,0	Wang et al. (2025); Martinez et al. (2025)

Nota: la asignación de componente predominante es indicativa; varios estudios abordan más de un componente simultáneamente.

Fundamentos Pedagógicos Aplicables a la Tutoría Generativa

El diseño de los STI con IAG se apoya, de manera explícita o implícita, en un conjunto reducido de marcos pedagógicos recurrentes, algunos de ellos anteriores en varias décadas a la IAG. El andamiaje sociocultural propuesto por Vygotsky (1978) continúa siendo el referente más citado: su noción de zona de desarrollo próximo, entendida como la distancia entre lo que un estudiante puede resolver de forma autónoma y lo que puede lograr con apoyo experto, se reinterpreta en el contexto de la IAG como la capacidad del sistema de ofrecer apoyo progresivo sin resolver la tarea por el estudiante (Nathaniel et al., 2025a). Esta lógica de andamiaje se extiende también a la generación de explicaciones de código: MacNeil et al. (2023) muestran que las explicaciones línea por línea generadas por LLM se perciben como útiles para comprender fragmentos de código de distinta complejidad, funcionando como un andamiaje conceptual complementario a la resolución directa de problemas.

El aprendizaje activo constituye un segundo eje central: Hou et al. (2024) demuestran, mediante CodeTailor, que los ejercicios personalizados tipo Parsons Puzzle sostienen el compromiso cognitivo más eficazmente que la simple entrega de una solución generada por

el modelo. Un tercer eje es la retroalimentación formativa, entendida, siguiendo a Hattie y Timperley (2007), como aquella que responde a tres preguntas del estudiante (hacia dónde voy, cómo voy, hacia dónde ir después) y que es oportuna, específica y orientada al proceso; Azaiz et al. (2023) muestran que los modelos GPT generan retroalimentación de calidad en corrección de código, aunque con limitaciones en la localización precisa de errores, mientras Li et al. (2025) integran visualización dinámica de algoritmos con retroalimentación personalizada en un MOOC.

En la misma línea, Xue et al. (2024) reportan mejoras en las tasas de completitud de tareas en programación introductoria cuando la retroalimentación conversacional está disponible durante todo el proceso, no solo al final. Revisiones en subdominios específicos, como la instrucción en bases de datos, confirman que las herramientas de IA se categorizan según modelos de diseño instruccional, roles y desafíos pedagógicos (Gaitantzi & Kazanidis, 2025). Finalmente, la metacognición y la autorregulación, entendida según Zimmerman (2002) como un proceso cíclico de planificación, ejecución y autorreflexión, constituyen el eje menos desarrollado: Looi y Jia (2025) advierten que los LLM aún carecen de la capacidad de construir perfiles de aprendizaje dinámicos o seleccionar autónomamente estrategias pedagógicas óptimas equivalentes a las de un tutor experto.

Taxonomía de Enfoques Pedagógicos Identificados

La síntesis de ambos corpus permite identificar seis patrones pedagógicos recurrentes, resumidos en la tabla 3 y desarrollados a continuación. Primero, el andamiaje sin solución directa, presente en sistemas que evitan revelar la respuesta completa y en su lugar ofrecen pistas progresivas, pseudocódigo o puzzles estructurados, documentado en CodeTailor (Hou et al., 2024), en pistas de siguiente paso (Roest et al., 2024) y en el andamiaje gamificado de Cao (2023). En depuración de errores de compilación, Pankiewicz y Baker (2024) muestran que las

pistas generadas por GPT mejoran la persistencia estudiantil sin resolver la tarea automáticamente, extendiendo el principio de andamiaje a un tipo de dificultad frecuente entre programadores principiantes.

Segundo, la retroalimentación formativa multinivel, que combina corrección inmediata de sintaxis con orientación diferida sobre diseño y lógica; Ouyang et al. (2024) muestran que esta retroalimentación, aunque valiosa, difiere de la retroalimentación docente en profundidad y personalización, mientras que Edwards et al. (2025) demuestran que los LLM pueden sintetizar datos de proceso de programación, como registros de pulsaciones de teclado, para generar retroalimentación formativa automatizada centrada en el desarrollo incremental del código. Persson et al. (2024) reportan menores tiempos de completitud, mayores tasas de éxito y mejores puntuaciones entre estudiantes asistidos por LLM, aunque advierten que las ganancias dependen de la calidad de la retroalimentación, no solo de su disponibilidad.

Tercero, el aprendizaje autorregulado, eje central en Wang et al. (2025) y López-Pernas et al. (2025), quienes documentan que los estudiantes emplean con frecuencia instrucciones de monitoreo del proceso, pero rara vez despliegan estrategias metacognitivas más profundas como la reflexión o la autoevaluación. Cuarto, los marcos instruccionales previos a la IAG (taxonomía de Bloom, diseño universal para el aprendizaje, modelo 5E) se adaptan activamente al contexto generativo (Nathaniel et al., 2025a, 2025b; Liu et al., 2025; Alphinás y Tambo, 2024). Un patrón adicional es la integración de perfiles de usuario diferenciados en sistemas de evaluación automática: Frankford et al. (2024) documentan cómo un LLM operando como AI-Tutor identifica patrones de interacción distintos, sentando las bases para una personalización más fina del andamiaje.

Quinto, el aprendizaje experiencial y afectivo, ejemplificado por el tutor virtual JULIUS de Martínez et al. (2025), que integra conectivismo y aprendizaje basado en retos para atender el desarrollo técnico y la resiliencia

emocional. Sexto, y de manera transversal, la equidad y la alfabetización digital previa emergen como variable determinante más que la mera disponibilidad de la herramienta: Park et al. (2025) documentan que el 40,6 % de los estudiantes manifestó incertidumbre sobre el uso responsable de la IA, mientras que el corpus original confirma que la alfabetización en IA y el conocimiento previo determinan la efectividad real. Lai y Lin (2025), en el sistema ITS-CAL, documentan que los estudiantes de menor nivel dependen más de la función de pistas, reforzando la necesidad de un componente de equidad explícito.

La dimensión metacognitiva encuentra respaldo adicional en los patrones de búsqueda de ayuda: Sheese et al. (2024) documentan que las solicitudes de ayuda a asistentes de programación basados en LLM varían sustancialmente según el nivel de autorregulación del estudiante, mientras que Haindl y Weinberger (2024) reportan que un grupo de estudiantes evita deliberadamente el uso de ChatGPT por temor a no desarrollar competencia real, una forma explícita, aunque informal, de autorregulación metacognitiva. En la dimensión de integridad y confianza, Stoyanova et al. (2025) encuentran que los docentes expresan mayor preocupación que los estudiantes por la integridad académica, aunque ambos grupos valoran estas herramientas como complemento y no como sustituto de la tutoría humana.

Tabla 3
Síntesis de los seis patrones pedagógicos identificados en la literatura

Patrón pedagógico	Descripción breve	Estudios representativos
1. Andamiaje sin solución directa	Pistas progresivas o pseudocódigo en lugar de la solución completa	Hou et al. (2024); Roest et al. (2024); Cao (2023); Pankiewicz y Baker (2024)
2. Retroalimentación formativa multinivel	Corrección inmediata de sintaxis combinada con orientación diferida sobre diseño y lógica	Ouyang et al. (2024); Edwards et al. (2025); Persson et al. (2024)
3. Aprendizaje autor-regulado	Monitoreo del proceso más frecuente que estrategias metacognitivas profundas	Wang et al. (2025); López-Pernas et al. (2025)
4. Marcos instruccionales estructurados adaptados	Taxonomía de Bloom, diseño universal para el aprendizaje y modelo 5E adaptados al contexto generativo	Nathaniel et al. (2025a, 2025b); Liu et al. (2025); Alphinay y Tambo (2024); Frankford et al. (2024)
5. Aprendizaje experiencial y afectivo	Integración de conectivismo y aprendizaje basado en retos con atención al componente emocional	Martinez et al. (2025)
6. Equidad y alfabetización digital previa	La alfabetización en IA y el conocimiento previo determinan la efectividad real de la herramienta	Park et al. (2025); Lai y Lin (2025); Sheese et al. (2024); Haindl y Weinberger (2024); Stoyanova et al. (2025)

Nota: los patrones no son mutuamente excluyentes; varios estudios se vinculan a más de un patrón.

Matriz de Asignación de Estudios a Patrones y Componentes

Con el fin de hacer explícito el proceso de categorización mediante el cual los seis patrones pedagógicos se agruparon en los cuatro componentes del Modelo AREM, la tabla 4 detalla la asignación individual de cada uno de los 20 estudios de la búsqueda complementaria a su patrón predominante (P1 a P6, definidos en la tabla 3) y al componente correspondiente.

Tabla 4
Asignación de los 20 estudios a patrones pedagógicos y componentes del Modelo AREM

Estudio	Patrón (ver tabla 3)	Componente AREM
Nathaniel et al. (2025b)	P4	A
Nathaniel et al. (2025a)	P4	A
Liu et al. (2025)	P4	A
Gomez-Jaramillo et al. (2025)	P4	A
Alphinay y Tambo (2024)	P4	A
Sankaranarayanan y Mithun (2024)	P4	A
Gaitantzi y Kazanidis (2025)	P4	A
Liao et al. (2024)	P1	A
Cao (2023)	P1	A
Farreras (2025)	P1	A
Penney et al. (2024)	P1	A
Proust-Andrewkha et al. (2025)	P1	A
Azaiz et al. (2023)	P2	R
Li et al. (2025)	P2	R
Edwards et al. (2025)	P2	R
Park et al. (2025)	P6	E
Wang et al. (2025)	P3	M
Prather et al. (2024)	P3	M
López-Pernas et al. (2025)	P3	M
Martinez et al. (2025)	P5	M

Nota: P1-P6 remiten a los patrones definidos en la tabla 3. La asignación refleja el patrón predominante identificado a partir de título y resumen, verificado a texto completo en el 75,0 % de los casos (Material y métodos); algunos estudios abordan más de un patrón, pero se asignó una única categoría dominante.

Propuesta: Modelo AREM

A partir de los seis patrones identificados y de las brechas persistentes señaladas en la literatura, entre ellas la ausencia de marcos pedagógicos explícitos, el riesgo de sobredependencia y de aprendizaje superficial, y la desatención sistemática de la dimensión metacognitiva y de equidad, se propone el Modelo AREM, un marco pedagógico integrador de cuatro componentes interdependientes para el diseño de STI con IAG orientados a la enseñanza de programación. La transición de los seis patrones hacia estos cuatro componentes siguió un análisis comparativo de función pedagógica dominante, evaluando cada patrón según el momento de intervención respecto a la resolución

de la tarea (antes o durante, frente a después) y el objeto sobre el que actúa (el andamiaje cognitivo externo frente a la autorregulación interna del estudiante): los patrones 1 (andamiaje sin solución directa) y 4 (marcos instruccionales estructurados adaptados) confluyen en el componente A, al compartir la función de estructurar el apoyo cognitivo sin resolver la tarea; el patrón 2 (retroalimentación formativa multinivel) constituye por sí solo el componente R; el patrón 6 (equidad y alfabetización digital previa) constituye por sí solo el componente E; y los patrones 3 (aprendizaje autorregulado) y 5 (aprendizaje experiencial y afectivo) confluyen en el componente M, por su carácter interno y metacognitivo-afectivo compartido. Este criterio, aplicado de manera consistente y documentado en la tabla 4, hace el proceso de categorización trazable y replicable por otros investigadores (ver tabla 5 y figura 2).

Tabla 5
Componentes del Modelo AREM

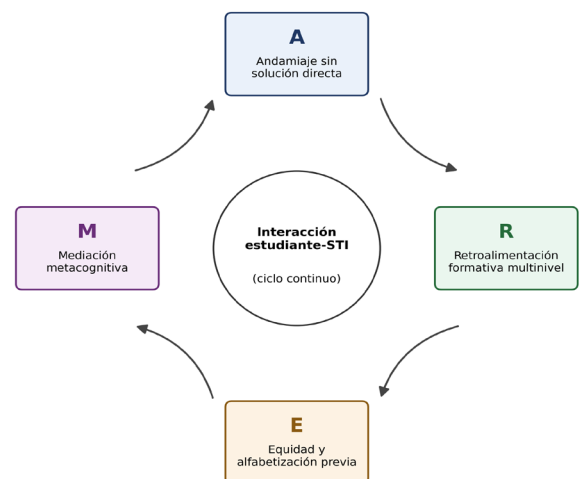
Componente	Principio pedagógico	Implementación esperada en el STI
A — Andamiaje sin solución directa	Zona de desarrollo próximo (Vygotsky); el sistema guía sin resolver la tarea por el estudiante	Pistas progresivas, pseudocódigo, preguntas socráticas antes de código ejecutable; límites configurables de revelación de solución
R — Retroalimentación formativa multinivel	Retroalimentación oportuna, específica y orientada al proceso, no solo al resultado	Combinación de retroalimentación inmediata (sintaxis, errores) y diferida (diseño, lógica); trazabilidad del progreso longitudinal
E — Equidad y alfabetización digital previa	El impacto del STI depende del nivel de alfabetización en IA y del conocimiento previo del estudiante, no solo de la disponibilidad de la herramienta	Módulo inicial de alfabetización en IA; perfiles de entrada diferenciados; monitoreo de patrones de uso por nivel de acceso previo
M — Mediación metacognitiva	Autorregulación y conciencia del propio proceso de aprendizaje frente al riesgo de dependencia irreflexiva	Prompts reflexivos post-interacción; indicadores visibles de uso acumulado para el propio estudiante

Nota: síntesis de los seis patrones pedagógicos identificados en la literatura revisada.

El Modelo AREM se concibe como un ciclo y no como una secuencia lineal. Cada interacción del estudiante con el sistema activa primero el componente de andamiaje, que determina el nivel de apoyo apropiado

sin resolver la tarea; genera después una retroalimentación formativa ajustada al nivel de competencia observado; se evalúa en función de indicadores de equidad de acceso y alfabetización digital previa del estudiante; y cierra con un momento de mediación metacognitiva que retroalimenta el perfil del estudiante para la siguiente interacción, tal como se ilustra en la figura 2. Esta circularidad busca responder directamente a la brecha, documentada de manera consistente en la literatura, entre la sofisticación conversacional de los sistemas actuales y su limitada profundidad pedagógica (Ding et al., 2026; Proust-Androwkha et al., 2025).

Figura 2
Representación cíclica del Modelo AREM



Nota: los cuatro componentes se activan de manera secuencial y recursiva en cada interacción entre el estudiante y el sistema de tutoría inteligente. Autoría propia.

Es necesario precisar que el Modelo AREM se presenta como una propuesta derivada de síntesis narrativa, sin validación empírica propia. Esta limitación se declara de manera explícita y se retoma en conclusiones como agenda futura.

La aplicabilidad del modelo puede ilustrarse con un ejemplo hipotético: ante un error de compilación, el sistema activaría primero el andamiaje ofreciendo una pista sobre la naturaleza del error sin mostrar la corrección

(Pankiewicz y Baker, 2024); generaría luego una retroalimentación adaptada al historial de errores previos; evaluaría si el estudiante completó el módulo de alfabetización en IA de su perfil de acceso, como exige la equidad; y cerraría con una pregunta reflexiva breve que active la mediación metacognitiva. Este recorrido, aunque hipotético, ilustra cómo los cuatro componentes se traducen en decisiones de diseño concretas y verificables.

Discusión y Conclusiones

Discusión

La distribución de los 20 estudios (tabla 2) complementa la síntesis temática previa. El componente A (andamiaje) concentra la mayor proporción (60,0 %), seguido por mediación metacognitiva (20,0 %), retroalimentación (15,0 %) y equidad (5,0 %). Esta concentración no implica que el andamiaje domine el campo en términos absolutos, sino que refleja, en parte, un artefacto de la ecuación de búsqueda: el bloque P4 incluyó explícitamente términos como “scaffolding” e “instructional design”, sobrerrepresentando estudios centrados en marcos estructurados de apoyo. La baja representación de la equidad (un único estudio) confirma, más bien, la brecha ya señalada en la literatura revisada.

El Modelo AREM se distingue de los marcos pedagógicos previos revisados, como GenAI-Ped (Nathaniel et al., 2025a), en que no se limita a adaptar marcos instruccionales preexistentes al contexto de la IAG, sino que articula explícitamente cuatro dimensiones que la literatura revisada aborda de manera dispersa: el andamiaje, la retroalimentación, la equidad y la metacognición rara vez se integran en un mismo marco de diseño. Esta integración resulta particularmente relevante frente al hallazgo transversal más consistente de la literatura revisada: la coexistencia de una recepción estudiantil mayoritariamente positiva hacia los asistentes de IAG con una preocupación persistente, tanto docente como estudiantil, por el desarrollo de un aprendizaje superficial (Dawson et al., 2025; Park et al., 2025).

La comparación con marcos consolidados de integración tecnológico-pedagógica refuerza esta distinción. El modelo TPACK (Mishra & Koehler, 2006), referente clásico para pensar la integración de la tecnología en la enseñanza, describe la interacción entre conocimiento tecnológico, pedagógico y disciplinar, pero fue concebido para la planificación docente general y no anticipa el caso de un sistema que genera contenido de manera autónoma y en tiempo real. El Modelo AREM no sustituye a TPACK, sino que lo complementa: mientras TPACK orienta qué debe saber un docente para integrar tecnología, AREM orienta qué debe hacer el propio sistema generativo, en la interacción momento a momento con el estudiante, para que esa integración sea pedagógicamente robusta.

Diálogo con la Literatura Adyacente sobre Calidad Educativa Digital (Nota Metodológica)

Es importante precisar, antes de continuar, que esta sección incorpora literatura adyacente mediante lectura dirigida y no mediante el proceso de cribado sistemático descrito en Material y métodos: su función es estrictamente conceptual y comparativa, no evidencial dentro del corpus analizado en el apartado anterior. Con esa salvedad, la investigación reciente sobre calidad de servicios educativos digitales en universidades públicas ofrece un marco teórico complementario que permite refinar la arquitectura del Modelo AREM. Vinueza-Martínez (2026), en una revisión sistemática cuantitativa de 15 estudios sobre calidad de e-learning en universidades públicas (N = 9.847 estudiantes, 10 países), propone el Public University E-Learning Quality Model (PUEQM), que distingue dimensiones universales que operan como factores higiénicos, en el sentido clásico de Herzberg (1966) y de su revisión conceptual posterior (Alshmemri et al., 2017), cuya ausencia genera insatisfacción pero cuya presencia no garantiza aprendizaje superior más allá de un umbral funcional, de dimensiones específicas del sector público que funcionan como motivadores diferenciadores.

Esta lógica configuracional resulta directamente aplicable a la arquitectura del

Modelo AREM. Dos de sus componentes, el andamiaje sin solución directa y la equidad y alfabetización digital previa, operan de manera análoga a un factor higiénico: su ausencia compromete estructuralmente la experiencia de aprendizaje, ya que un estudiante sin alfabetización digital previa difícilmente puede beneficiarse de ningún nivel de sofisticación conversacional del sistema, independientemente de su calidad técnica, pero su sola presencia no garantiza aprendizaje profundo. Los otros dos componentes, la retroalimentación formativa multinivel y la mediación metacognitiva, funcionan en cambio como factores diferenciadores: una vez asegurada la condición de acceso equitativo y de andamiaje básico, es la calidad de estos dos componentes la que determina si la interacción deriva en aprendizaje profundo o en dependencia superficial. La tabla 6 resume esta relectura configuracional en dos niveles funcionales.

Tabla 6

Modelo AREM: condiciones necesarias y factores diferenciadores

Nivel funcional	Componentes AREM	Analogía con el PUEQM (Vinueza-Martínez, 2026)
Condiciones necesarias (factor higiénico): su ausencia compromete la experiencia; su presencia no garantiza por sí sola aprendizaje profundo	A — Andamiaje sin solución directa; E — Equidad y alfabetización digital previa	Infraestructura tecnológica y soporte administrativo (dimensiones universales, Tier 1)
Factores diferenciadores (motivadores): determinan la profundidad del aprendizaje una vez aseguradas las condiciones necesarias	R — Retroalimentación formativa multinivel; M — Mediación metacognitiva	Articulación de valor percibido y dimensiones específicas del sector público (Tier 2)

Nota: relectura configuracional del Modelo AREM inspirada en la arquitectura por niveles del Public University E-Learning Quality Model (Vinueza-Martínez, 2026).

La comparación entre ambos dominios revela además una tensión estructural común: así como las dimensiones técnicas y administrativas concentran el 93,3 % de la frecuencia investigativa en la literatura de e-learning frente a apenas 20,0 % para equidad institucional (Vinueza-Martínez, 2026), la literatura sobre STI con IAG muestra un patrón equivalente, con el desarrollo técnico

superando consistentemente a la fundamentación pedagógica explícita (Ding et al., 2026; Proust-Androwkha et al., 2025). Esta convergencia entre dos dominios distintos sugiere que la primacía de lo técnico sobre lo pedagógico es un patrón sistémico más amplio en tecnología educativa, lo que refuerza la pertinencia de marcos como el Modelo AREM.

Limitaciones y Alcance Teórico

La analogía entre el Modelo AREM y el PUEQM debe interpretarse con cautela metodológica: ambos operan en unidades de análisis distintas, el PUEQM describe determinantes institucionales agregados a nivel de universidad, mientras que AREM describe componentes de diseño pedagógico a nivel de interacción individual estudiante-IAG. La distinción entre factores higiénicos y motivadores se traslada como marco conceptual orientador, no como equivalencia empírica verificada, y queda sujeta a la misma agenda de validación declarada para el modelo. El hallazgo de heterogeneidad geográfica extrema de Vinueza-Martínez (2026), con coeficientes entre $\beta = 0,197$ y $\beta = 0,768$ según el país, es una advertencia y no una transferencia directa: sugiere que el componente de equidad probablemente requiera calibración contextual antes que aplicación universal.

Los cuatro componentes admiten además distintos escenarios de implementación: en un entorno masivo tipo MOOC, el componente E podría operacionalizarse mediante un módulo de autodiagnóstico inicial automatizado, mientras que en un curso presencial reducido bastaría con una breve entrevista diagnóstica docente-estudiante; esta variabilidad es, en sí misma, una hipótesis pendiente de contrastación empírica. Una validación rigurosa enfrentaría además desafíos concretos: operacionalizar cada componente en indicadores observables (por ejemplo, tasa de revelación de solución para A, tiempo y especificidad de respuesta para R, perfiles de acceso previo para E, frecuencia de prompts reflexivos para M), diseñar un estudio comparativo con grupo de control frente a un tutor conversacional convencional, y controlar

variables confusoras como la alfabetización previa en IA de los participantes.

Implicaciones para la Formación Docente y el Diseño Curricular

La adopción del Modelo AREM conlleva implicaciones para la formación docente, dimensión que la literatura aborda de forma insuficiente. Ismail et al. (2024) documentan que los docentes de programación en C emplean ChatGPT para elaborar ítems de evaluación eficientemente, pero requieren supervisión cuidadosa para que la facilidad de generación no sustituya el juicio experto; esto sugiere que la retroalimentación del Modelo AREM no puede delegarse por completo en el sistema generativo. A ello se suma una dimensión afectiva poco explorada: Kuhail et al. (2024) documentan, en 99 programadores profesionales, preocupación por la seguridad laboral ante la IA generativa, lo que sugiere que los propios instructores pueden aproximarse a la integración de IAG con ambivalencia. Incorporar espacios de formación docente que aborden esta ambivalencia constituye una condición institucional implícita para la agenda futura.

Conclusiones y Agenda Futura

La síntesis realizada evidencia que, pese a la rápida proliferación de STI con IAG en programación, los marcos pedagógicos que sustentan su diseño continúan siendo mayoritariamente implícitos o adaptados ad hoc de teorías preexistentes. Se identificaron seis patrones recurrentes -andamiaje sin solución directa, retroalimentación formativa multinivel, aprendizaje autorregulado, marcos instruccionales adaptados, aprendizaje experiencial y equidad digital- que rara vez se integran en un marco coherente. Esta fragmentación constituye, según Ding et al. (2026) y Proust-Andrewkha et al. (2025), uno de los principales obstáculos para que la investigación madure hacia un cuerpo teórico acumulativo. La contribución central de este artículo es, por tanto, la respuesta directa a esa fragmentación mediante el Modelo AREM, desarrollado en Análisis y resultados.

Más allá de la programación, el Modelo AREM introduce una innovación metodológica de mayor alcance: aplicar la lógica configuracional de factores higiénicos y motivadores, desarrollada en la investigación sobre calidad de servicios institucionales (Herzberg, 1966; Alshmemri et al., 2017; Vinueza-Martínez, 2026), al diseño pedagógico de un tutor a nivel de interacción individual. Este cruce no se documenta, hasta donde permite establecer la literatura revisada, en STI con IAG, y abre una vía replicable en subdominios afines -matemáticas, idiomas o razonamiento científico asistidos por IAG- para distinguir qué componentes deben tratarse como condiciones mínimas y cuáles como factores de diferenciación. El valor del modelo no se agota en la programación, sino en la arquitectura conceptual que ofrece para la IAG educativa en general.

La inclusión de la equidad como componente de primer orden responde a una brecha que la evidencia muestra como global y no exclusivamente latinoamericana: la literatura adyacente documenta una omisión sistemática del nivel socioeconómico en estudios de diez países de tres continentes, junto con heterogeneidad extrema en los efectos de la interacción digital según infraestructura y alfabetización (Vinueza-Martínez, 2026). El componente de equidad del Modelo AREM es, por diseño, aplicable a cualquier sistema con recursos limitados, incluyendo África subsahariana, sur y sudeste asiático, y regiones rurales de economías de altos ingresos. Esta generalidad estructural es una de las razones por las que el modelo aspira a relevancia en la conversación internacional sobre el diseño responsable de la IAG en educación.

El Modelo AREM articula estos patrones en un marco cíclico para guiar el diseño pedagógico de la próxima generación de tutores inteligentes con IAG. Surelectura configuracional en condiciones necesarias (andamiaje, equidad) y factores diferenciadores (retroalimentación, mediación metacognitiva), inspirada en el PUEQM (Vinueza-Martínez, 2026), ofrece una hoja de ruta de implementación por fases para instituciones con recursos limitados: asegurar primero el acceso equitativo y el andamiaje

básico como condiciones no negociables, y solo después invertir en los mecanismos que diferencian a un sistema pedagógicamente robusto de uno meramente conversacional.

Se trata, sin embargo, de una propuesta pendiente de validación empírica, limitación declarada explícitamente que constituye la principal línea de investigación futura: el diseño de un estudio piloto, idealmente multiinstitucional y replicado en más de un país, que permita operacionalizar los cuatro componentes del modelo y someter a prueba la hipótesis configuracional planteada. Una segunda línea consiste en integrar esta síntesis con el análisis de arquitecturas tecnológicas y resultados de aprendizaje, en un estudio de mayor alcance en desarrollo. Estas líneas sitúan al Modelo AREM no como un punto de llegada, sino como hipótesis de partida para una agenda comparable entre países, orientada a que el diseño pedagógico, y no solo el avance técnico, determine el futuro de la tutoría inteligente en la era de la IAG.

Referencias Bibliográficas

- Alphinas, R., & Tambo, T. (2024). Generative AI learning environment for non-computer science engineering students: Coding versus generative AI prompting. En Proceedings of the 4th International Conference on AI Research (ICAIR 2024) (pp. 20-29).
- Alshmemri, M., Shahwan-Akl, L., & Maude, P. (2017). Herzberg's two-factor theory. *Life Science Journal*, 14(5), 12-16.
- Azaiz, I., Deckarm, O., & Strickroth, S. (2023). AI-enhanced auto-correction of programming exercises: How effective is GPT-3.5? *International Journal of Engineering Pedagogy*, 13(8), 67-83. <https://doi.org/10.3991/ijep.v13i8.45621>
- Batsaikhan, B., & Correia, A.-P. (2024). The effects of Generative Artificial Intelligence on Intelligent Tutoring Systems in higher education: A systematic review. *Studies in Technology Enhanced Learning*, 4(1). <https://doi.org/10.21428/8c225f6e.33570bb1>
- Campbell, M., McKenzie, J. E., Sowden, A., Katikireddi, S. V., Brennan, S. E., Ellis, S., Hartmann-Boyce, J., Ryan, R., Shepperd, S., Thomas, J., Welch, V., & Thomson, H. (2020). Synthesis without meta-analysis (SWiM) in systematic reviews: Reporting guideline. *BMJ*, 368, l6890. <https://doi.org/10.1136/bmj.l6890>
- Cao, C. (2023). Scaffolding CS1 courses with a large language model-powered intelligent tutoring system. En Proceedings of the International Conference on Intelligent User Interfaces (pp. 229-232). ACM. <https://doi.org/10.1145/3581754.3584111>
- Dawson, M. G., Deer, R., & Boguslawski, S. (2025). Cognitive dissonance in programming education: A qualitative exploration of the impact of generative AI on application-directed learning. *Computers in Human Behavior Reports*, 19. <https://doi.org/10.1016/j.chbr.2025.100724>
- Ding, L., O'Berry, R., Tallent, H., Gharti, S., & Gapud, A. (2026). Learning with large language models: Beyond prompt engineering. *Education and Information Technologies*, 31, 2877-2901. <https://doi.org/10.1007/s10639-026-13924-2>
- Edwards, J., Hellas, A., & Leinonen, J. (2025). On the opportunities of large language models for programming process data. En Proceedings of the 27th Australasian Computing Education Conference (pp. 105-113). ACM. <https://doi.org/10.1145/3716640.3716652>
- Elnaffar, S., Rashidi, F., & Abualkishik, A. Z. (2026). Teaching with AI: A systematic review of chatbots, generative tools, and tutoring systems in programming education. *International Journal of Learning, Teaching and Educational Research*, 25(1), 1-28. <https://doi.org/10.26803/ijlter.25.1.1>
- Frankford, E., Sauerwein, C., Bassner, P., Krusche, S., & Breu, R. (2024). AI-tutoring in software engineering

- education: Experiences with large language models in programming assessments. En Proceedings of the International Conference on Software Engineering. ACM. <https://doi.org/10.1145/3639474.3640061>
- Gaitantzi, A., & Kazanidis, I. (2025). The role of artificial intelligence in computer science education: A systematic review with a focus on database instruction. *Applied Sciences*, 15(7). <https://doi.org/10.3390/app15073960>
- Gomez-Jaramillo, S., Cardona-Zapata, O., Valbuena-Henao, M., & Poliche, V. (2025). Guided learning with AI: A didactic strategy using sequential tutoring via ChatGPT for object-oriented programming. *International Journal of Learning, Teaching and Educational Research*, 24(7), 717-736. <https://doi.org/10.26803/ijlter.24.7.35>
- Haindl, P., & Weinberger, G. (2024). Students' experiences of using ChatGPT in an undergraduate programming course. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2024.3380909>
- Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research*, 77(1), 81-112. <https://doi.org/10.3102/003465430298487>
- Herzberg, F. (1966). *Work and the nature of man*. World Publishing Company.
- Hou, X., Wu, Z., Wang, X., & Ericson, B. J. (2024). CodeTailor: LLM-powered personalized Parsons puzzles for engaging support while learning programming. En Proceedings of the 11th ACM Conference on Learning @ Scale (pp. 51-62). <https://doi.org/10.1145/3657604.3662032>
- Ismail, S., Lynch, J., & Alghazo, R. (2024). Enhancing STEM education with ChatGPT: A case study on developing practice assessments for C programming. En 2024 IEEE 13th International Conference on Engineering Education. <https://doi.org/10.1109/ICEED62316.2024.10923780>
- Kazemitabaar, M., Ye, R., Wang, X., Henley, A. Z., Denny, P., Craig, M., & Grossman, T. (2024). CodeAid: Evaluating a classroom deployment of an LLM-based programming assistant that balances student and educator needs. En Proceedings of the CHI Conference on Human Factors in Computing Systems. ACM. <https://doi.org/10.1145/3613904.3642773>
- Kuhail, M. A., Mathew, S. S., Khalil, A., Berengueres, J., & Shah, S. J. H. (2024). "Will I be replaced?" Assessing ChatGPT's effect on software development and programmer perceptions of AI tools. *Science of Computer Programming*. <https://doi.org/10.1016/j.scico.2024.103111>
- Lai, C.-H., & Lin, C.-Y. (2025). 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). <https://doi.org/10.3390/app15041922>
- Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 159-174.
- Li, M., Wang, D., Purwanto, E., Selig, T., Zhang, Q., & Liang, H.-N. (2025). VisualCodeMOOC: A course platform for algorithms and data structures integrating a conversational agent for enhanced learning through dynamic visualizations. *SoftwareX*, 30. <https://doi.org/10.1016/j.softx.2025.102072>
- Liao, J., Zhong, L., Zhe, L., Xu, H., Liu, M., & Xie, T. (2024). Scaffolding computational thinking with ChatGPT. *IEEE Transactions on Learning Technologies*, 17, 1668-1682. <https://doi.org/10.1109/TLT.2024.3392896>
- Liu, Y., Zhu, J., Bai, S., Li, W., & Yang, C. (2025). A task-oriented artificial intelligence chatbot to support university students' design problem-solving: An exploratory case study. En Proceedings

- of the 5th International Conference on Educational Technology (pp. 158-162). IEEE. <https://doi.org/10.1109/ICET67421.2025.11380454>
- Looi, C.-K., & Jia, F. (2025). Personalization capabilities of current technology chatbots in a learning environment: An analysis of student-tutor bot interactions. *Education and Information Technologies*, 30, 14165-14195. <https://doi.org/10.1007/s10639-025-13369-z>
- López-Pernas, S., Misiejuk, K., Oliveira, E., & Saqr, M. (2025). The dynamics of the self-regulation process in student-AI interactions: The case of problem-solving in programming education. En *Proceedings of the 25th International Conference on Computing Education Research*. ACM. <https://doi.org/10.1145/3769994.3770043>
- MacNeil, S., Tran, A., Hellas, A., Kim, J., Sarsa, S., Denny, P., Bernstein, S., & Leinonen, J. (2023). Experiences from using code explanations generated by large language models in a web software development e-book. En *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*. <https://doi.org/10.1145/3545945.3569785>
- Martinez, J. R., Chong, M., Maya, S., Caicedo, M. A., & Luna, A. (2025). Enhancing algorithmic thinking and emotional resilience in programming education through AI powered virtual tutoring. En *Proceedings of the Frontiers in Education Conference*. IEEE. <https://doi.org/10.1109/FIE63693.2025.11328240>
- Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108(6), 1017-1054. <https://doi.org/10.1111/j.1467-9620.2006.00684.x>
- Nathaniel, J., Oyelere, S. S., Suhonen, J., & Tedre, M. (2025a). 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. <https://doi.org/10.1016/j.caeai.2025.100460>
- Nathaniel, J., Oyelere, S. S., Suhonen, J., & Tedre, M. (2025b). Literature review on the integration of generative AI in programming education. *International Journal of Artificial Intelligence in Education*, 35(5), 2724-2755. <https://doi.org/10.1007/s40593-025-00524-3>
- Ouyang, F., Guo, M., Zhang, N., Bai, X., & Jiao, P. (2024). Comparing the effects of instructor manual feedback and ChatGPT intelligent feedback on collaborative programming in China's higher education. *IEEE Transactions on Learning Technologies*. <https://doi.org/10.1109/TLT.2024.3486749>
- 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., ... Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372, n71. <https://doi.org/10.1136/bmj.n71>
- Pankiewicz, M., & Baker, R. S. (2024). Navigating compiler errors with AI assistance: A study of GPT hints in an introductory programming course. En *Proceedings of the 2024 Conference on Innovation and Technology in Computer Science Education V. 1*. ACM. <https://doi.org/10.1145/3649217.3653608>
- Park, S. M., Ho, M., Lin, M. P.-C., & Ryoo, J. (2025). Evaluating the impact of assistive AI tools on learning outcomes and ethical considerations in programming education. En *IEEE Global Engineering Education Conference*. <https://doi.org/10.1109/EDUCON62633.2025.11016517>
- Persson, M., Nouri, J., & Norén, E. (2024). The impact of large language models on programming education and student learning outcomes. *Applied Sciences*,

- 14(10). <https://doi.org/10.3390/app14104115>
- Prather, J., Reeves, B. N., Leinonen, J., MacNeil, S., Randrianasolo, A. S., Becker, B. A., Kimmel, B., Wright, J., & Briggs, B. (2024). The widening gap: The benefits and harms of generative AI for novice programmers. En *Proceedings of the 2024 ACM Conference on International Computing Education Research*, Volume 1 (pp. 469-486). ACM. <https://doi.org/10.1145/3632620.3671116>
- Proust-Androwkha, S., Psyché, V., Kyriakoglou, R., Boubaker, A., Lallé, S., Chartofylaka, L., Medeiros Machado, G., & El Mawas, N. (2025). Beyond programming assistance towards the integration of generative AI into computer science education: A scoping review. *Lecture Notes in Computer Science*, 15881, 380-389. https://doi.org/10.1007/978-3-031-98462-4_48
- Roest, L., Keuning, H., & Jeurig, J. (2024). Next-step hint generation for introductory programming using large language models. En *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3636243.3636259>
- Sheese, B., Liffiton, M., Savelka, J., & Denny, P. (2024). Patterns of student help-seeking when using a large language model-powered programming assistant. En *Proceedings of the 26th Australasian Computing Education Conference*. ACM. <https://doi.org/10.1145/3636243.3636249>
- Stoyanova, D., Stoyanova-Petrova, S., & Mileva, N. (2025). Exploring students' and teachers' perceptions about using ChatGPT in programming education. *International Journal of Engineering Pedagogy*. <https://doi.org/10.3991/ijep.v15i2.50607>
- Vinueza-Martínez, J. (2026). E-learning service quality and student satisfaction in public universities: A quantitative hierarchisation of determinants through systematic review. *Frontiers in Education*, 11. <https://doi.org/10.3389/educ.2026.1818648>
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.
- Wang, Z., Zou, D., Zhang, R., Lee, L.-K., Xie, H., & Wang, F. L. (2025). ChatGPT-enhanced self-regulated learning in programming education: Impacts on motivation, self-efficacy, and learning outcomes. *Interactive Learning Environments*. <https://doi.org/10.1080/10494820.2025.2559919>
- Xue, Y., Chen, H., Bai, G. R., Tairas, R., & Huang, Y. (2024). Does ChatGPT help with introductory programming? An experiment of students using ChatGPT in CS1. En *Proceedings of the International Conference on Software Engineering*. ACM. <https://doi.org/10.1145/3639474.3640076>
- Zimmerman, B. J. (2002). Becoming a self-regulated learner: An overview. *Theory Into Practice*, 41(2), 64-70. https://doi.org/10.1207/s15430421tip4102_2

Quito, 22 de julio del 2026

Estimados

Roger Marcelo Freire Avilés

Jorge Luis Vinueza Martínez

Delia Isabel Carrión León

V11-N4-1-4232

Presente

Reciban un cordial saludo del equipo de la revista 593 Digital Publisher CEIT, ISSN# 2588-0705, esta revista es indexada en Latindex con catálogo v2.0, su proceso contempla un arbitraje de pares ciegos y es multidisciplinar, evalúa la pertinencia en la calidad de investigación y sus dinámicas propias relacionadas al tema de estudio, con el fin de garantizar estándares de exigencia académica.

Este documento certifica que ha concluido la fase de revisión de pares, por lo tanto, el artículo fue aceptado para la publicación en el V11-N4-1, edición continua, por los autores **Roger Freire, Jorge Vinueza, Delia Carrión**, con el tema **"Modelo AREM: Una Propuesta Pedagógica para la IA Generativa en la Enseñanza de Programación"**, cuyos resultados obedecen a un proceso de investigación previo del/os autor/es.

doi.org/10.33386/593dp.2026.4-1.4232

Agradecemos su publicación y le deseamos éxitos en su carrera como investigadores.

Renato Revelo Dr.(c)
Editor General



Nota: La única forma de validar este documento es por medio del QR.

UNEMI

UNIVERSIDAD ESTATAL DE MILAGRO

¡Evolución académica!

@UNEMIEcuador

