

Estrategias de enseñanza de la codificación en la educación primaria: una revisión sistemática

Teaching strategies for coding in primary education: a PRISMA systematic review

 Jorge Luis Córdova López ¹

 Juan Francisco Orbegoso Orozco ²

 Dajanna Stephany Uribe Godoy ³

 Belinda Huamán Mendoza ⁴

Resumen

Introducción: La enseñanza de la codificación en educación primaria es crucial para el desarrollo del pensamiento computacional, competencia esencial en la formación integral de los estudiantes. **Objetivo:** Analizar las estrategias de enseñanza de la codificación en educación primaria y su impacto en el desarrollo del pensamiento computacional. **Metodología:** Se realizó una revisión sistemática siguiendo PRISMA, integrando 24 estudios publicados entre 2022 y 2025. Se evaluaron herramientas pedagógicas, metodologías, integración curricular y recursos tecnológicos aplicados en contextos escolares primarios. **Resultados:** Las herramientas visuales y tangibles, como Scratch, favorecen habilidades de descomposición y abstracción. El andamiaje pedagógico potencia los aprendizajes, mientras que la integración curricular genera mejores resultados que la instrucción aislada. La gamificación incrementa la motivación, aunque persisten brechas de género en la participación. Los sistemas inteligentes se identifican como una frontera pedagógica prometedora para personalizar y fortalecer el aprendizaje. **Conclusiones:** La efectividad de las estrategias de enseñanza de codificación depende del andamiaje pedagógico, la integración curricular y la formación docente. Estos factores son determinantes para el desarrollo sostenido del pensamiento computacional en los primeros niveles educativos, recomendando enfoques didácticos inclusivos y contextualizados que maximicen la participación y la adquisición de competencias digitales tempranas.

Palabras clave: Pensamiento computacional; Codificación; Educación primaria; Estrategias de enseñanza; Programación por bloques

Abstract

Introduction: Teaching coding in primary education is pivotal for the development of computational thinking—an essential competency for the holistic development of students in the digital age. **Objective:** To analyze coding instruction strategies in primary education and their impact on the development of computational thinking skills. **Methodology:** A systematic review was conducted following the PRISMA guidelines, integrating 24 studies published between 2022 and 2025. The review evaluated pedagogical tools, methodologies, curricular integration, and technological resources implemented within primary school contexts. **Results:** Visual and tangible tools, such as Scratch, facilitate decomposition and abstraction skills. Pedagogical scaffolding significantly enhances learning outcomes, while curricular integration yields superior results compared to isolated instruction. Furthermore, gamification increases student motivation, although gender gaps in participation persist. Intelligent systems are identified as a promising

¹⁻²Universidad César Vallejo, Lima, Perú, ³Universidad Tecnológica del Perú, Lima, Perú, ⁴Investigador independiente. Lima, Perú
Artículo recibido 23 de abril 2026 | Aceptado 28 de mayo 2026 | Publicado 1 de julio 2026

Autor de correspondencia: jlcordova@ucvvirtual.edu.pe

Conflicto de interés: Los autores declaran no tener conflicto de intereses.

Como citar: Córdova López, J. L., Orbegoso Orozco, J. F., Uribe Godoy, D. S., & Huamán Mendoza, B. (2026). Estrategias de enseñanza de la codificación en la educación primaria: una revisión sistemática. *Tribunal, Revista en Ciencias de la Educación y Ciencias Jurídicas*, 6(16), 1-19. <https://doi.org/10.59659/revistatribunal.v6i16.498>

pedagogical frontier for personalizing and strengthening the learning experience. **Conclusions:** The effectiveness of coding instruction strategies depends on pedagogical scaffolding, seamless curricular integration, and robust teacher training. These factors are decisive for the sustained development of computational thinking in early educational levels. The study recommends inclusive and contextualized didactic approaches to maximize student engagement and the acquisition of early digital competencies.

Keywords: Computational thinking; Coding; primary education; Teaching strategies; Block-based programming

Introducción

La incorporación de la codificación en la educación primaria adquirió relevancia creciente como respuesta a las demandas de una sociedad digitalizada. Los currículos de diversos países integraron la programación como competencia fundamental desde los primeros años de escolaridad, con el propósito de desarrollar el Pensamiento Computacional (PC) en los estudiantes (Falloon, 2024). Esta tendencia responde a la necesidad de formar ciudadanos capaces de resolver problemas complejos mediante la descomposición, la abstracción y el diseño de algoritmos. No obstante, la implementación de estas políticas educativas presenta desafíos vinculados con la formación del profesorado, la equidad en el acceso y la pertinencia de las estrategias didácticas elegidas (El-Hamamsy et al., 2023). Los marcos curriculares varían entre regiones, lo cual dificulta la identificación de enfoques pedagógicos efectivos y replicables en contextos diversos.

En el ámbito de las herramientas y plataformas para la enseñanza de la programación, la investigación reciente demostró que los ambientes de programación visual por bloques favorecen los resultados de aprendizaje en estudiantes de educación básica y secundaria (Yu et al., 2025). Estos entornos permiten a los niños construir programas mediante la manipulación de bloques interconectados, lo cual reduce la carga sintáctica y facilita la concentración en la lógica algorítmica. Las rúbricas de evaluación de proyectos de codificación creativa en niños pequeños también ganaron atención como instrumentos para valorar el producto y el proceso de programación (Unahalekhaka y Bers, 2022). Sin embargo, la diversidad de herramientas disponibles —desde Scratch hasta plataformas robóticas— plantea interrogantes sobre cuáles resultan más efectivas según la edad, el contexto sociocultural y los objetivos de aprendizaje perseguidos. La selección inadecuada de la herramienta puede obstaculizar el desarrollo del PC en lugar de potenciarlo.

Unido a lo anterior, el desarrollo del PC en los primeros niveles escolares depende de factores que trascienden la mera elección de la herramienta digital. Las herramientas de programación inciden en el desarrollo del razonamiento espacial y las funciones ejecutivas de los niños, según evidencia reciente en distintas etapas del aprendizaje (Pellas, 2025). Estos hallazgos indican que la interacción física con dispositivos programables ofrece ventajas cognitivas que las interfaces de pantalla sin manipulación física no proporcionan. En paralelo, la validación de instrumentos de evaluación del PC para estudiantes de los primeros grados adquirió prioridad investigativa, pues las pruebas existentes carecen de adecuación psicométrica para esas edades (Zhang y Wong, 2023).

La ausencia de evaluaciones confiables limita la posibilidad de medir con rigor el impacto de las intervenciones pedagógicas y de comparar resultados entre estudios de distintas regiones. Esta carencia metodológica constituye un obstáculo para el avance del campo.

Ante este panorama, las dificultades para la integración de la codificación en las aulas de primaria radican en la preparación insuficiente del profesorado. Los docentes carecen de formación específica en programación y PC, lo cual compromete la calidad de la instrucción que ofrecen (Kravik et al., 2022). Esta deficiencia formativa genera una dependencia excesiva hacia recursos prefabricados que no siempre se adaptan a las necesidades de cada grupo de estudiantes. Además, la exposición temprana a la programación influye de modo determinante en el desarrollo de habilidades de resolución de problemas, siempre que la intervención pedagógica posea estructura y progresión adecuadas (Wang et al., 2026). Los programas aislados y desvinculados del currículo producen efectos limitados y poco sostenibles. La vinculación entre la formación docente, el diseño curricular y la selección de estrategias didácticas resulta un factor decisivo para el éxito de estas iniciativas.

A partir de los elementos anteriores, resulta evidente que la enseñanza de la codificación en la educación primaria constituye un campo en fortalecimiento que aún requiere sistematización. Las investigaciones disponibles ofrecen resultados prometedores sobre el impacto de diversas herramientas y estrategias, pero la fragmentación de los hallazgos impide una comprensión completa de los factores que determinan la efectividad de las intervenciones. Los estudios tienden a concentrarse en herramientas específicas sin establecer comparaciones rigurosas entre alternativas didácticas. Asimismo, las variables de contexto como edad, género, formación docente y recursos disponibles reciben una atención desigual en la literatura, lo cual dificulta la formulación de recomendaciones generalizables. La ausencia de una síntesis que integre estos hallazgos representa una laguna para la toma de decisiones educativas fundamentadas en evidencia. Esta carencia justifica la realización de la presente investigación, donde se definen las siguientes interrogantes: ¿Qué estrategias de enseñanza de la codificación se emplean en la educación primaria? ¿Son efectivas en el proceso educativo?

En consecuencia, un estudio en este sentido contribuiría a la identificación de las dimensiones temáticas predominantes, evaluación de la efectividad de las aproximaciones pedagógicas y el examen de los factores que condicionan su implementación en contextos educativos diversos. Además, ofrecería una visión panorámica del estado del conocimiento. Los resultados de esta síntesis podrían orientar a investigadores, responsables de políticas educativas y docentes en la selección de estrategias fundamentadas para la enseñanza de la programación en los primeros niveles de escolaridad. Debido a esto, la presente revisión sistemática tuvo como objetivo analizar las estrategias de enseñanza de la codificación en la educación primaria.

Método

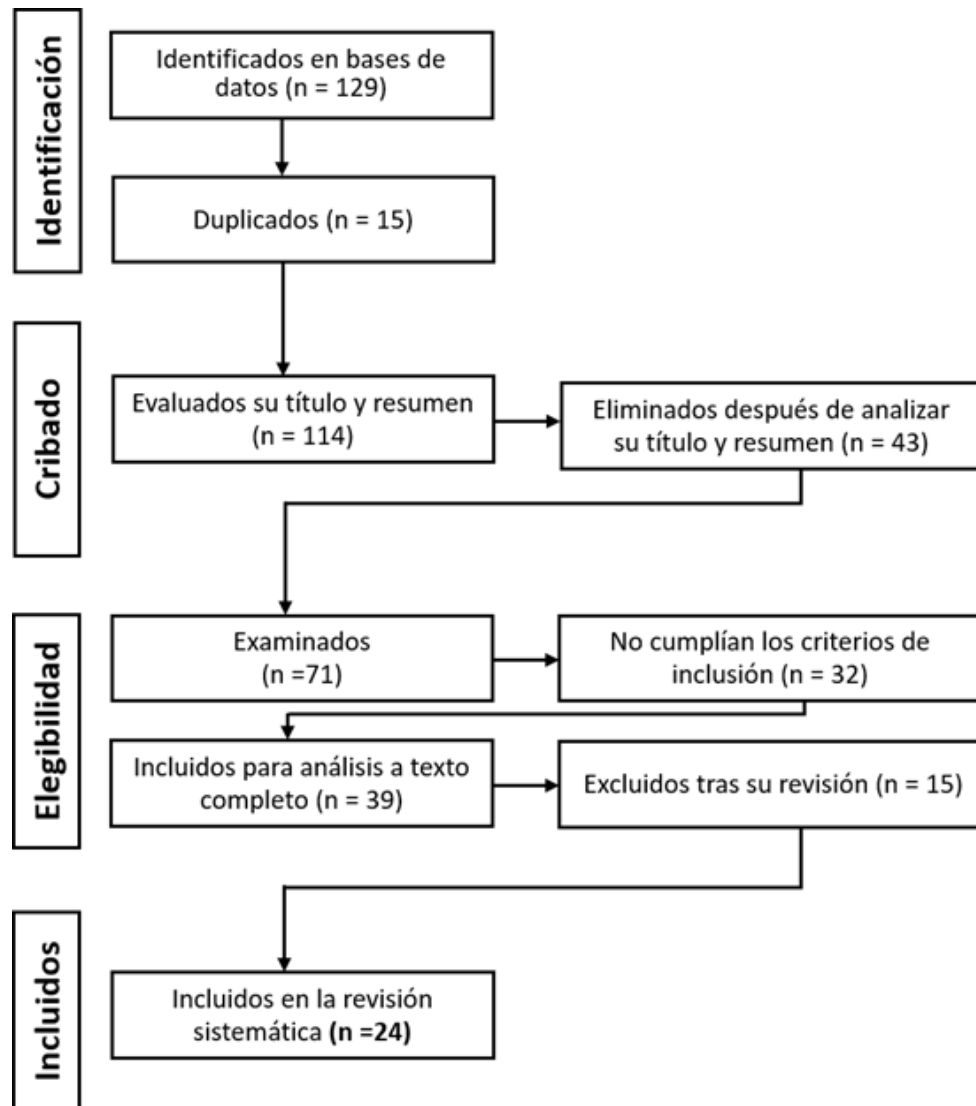
El estudio adoptó una metodología cualitativa basada en una revisión sistemática de literatura especializada en las estrategias de enseñanza de la codificación en la educación primaria. Para ello, se localizaron, seleccionaron y examinaron investigaciones

representativas sobre esta temática. La búsqueda de información se realizó en las bases de datos académicas Scopus, Web of Science y Scielo, seleccionadas por su cobertura interdisciplinaria, su indexación en reconocidos catálogos académicos y la rigurosidad de sus registros indexados. Esta selección permitió acceder a estudios recientes y relevantes.

En la fase de selección de los estudios se diseñó el protocolo de búsqueda con cadenas estandarizadas que combinaron términos en inglés y español: («computational thinking» OR «coding» OR «programming») AND («primary education» OR «elementary school») AND («teaching strategies» OR «pedagogical approaches»). Además, se aplicaron filtros que delimitaron los resultados según el año de publicación, el idioma y el tipo de documento. No se aplicó restricción geográfica, con la intención de obtener una visión completa de los resultados obtenidos en diferentes contextos con el empleo de estrategias de enseñanza de la codificación en la educación primaria. La búsqueda tuvo lugar en abril de 2026.

El protocolo de selección definió criterios de inclusión que requirieron que las investigaciones presentaran datos empíricos sobre estrategias de enseñanza de la codificación en estudiantes de educación primaria. Estos estudios debían ser artículos científicos originales, publicados en revistas con revisión por pares, accesibles en línea, y redactados en español o inglés, comprendidos entre 2022 a 2025. Los criterios de exclusión descartaron revisiones teóricas sin datos primarios, estudios centrados en educación secundaria o superior, informes de conferencias sin revisión por pares, artículos no disponibles en texto completo y registros duplicados.

El diseño metodológico siguió los protocolos establecidos en la metodología PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*) para garantizar la transparencia y la reproducibilidad del proceso de búsqueda, selección y análisis de la literatura. Esta guía proporciona un conjunto de ítems que orientan la identificación, el tamizado, la elegibilidad y la inclusión de estudios empíricos de manera rigurosa y documentada. En la Figura 1 se puede ver el proceso de selección, que comenzó con la identificación de 129 estudios; tras la eliminación de 15 duplicados, quedaron 114 para tamizado por título y resumen; de estos, 71 avanzaron a la evaluación del cumplimiento de todos los criterios de inclusión, proceso en el que fueron descartados 32 para quedar 39 para la revisión del texto completo, donde 24 investigaciones constituyeron el corpus de análisis de la presente revisión sistemática.

Figura 1. Diagrama de flujo PRISMA del proceso de selección de estudios

Junto con lo anterior, se realizó la evaluación de la calidad metodológica de los estudios incluidos mediante una matriz de criterios adaptada de las escalas de valoración para investigaciones educativas experimentales y cuasiexperimentales. Se examinaron aspectos como la claridad de los objetivos, la adecuación del diseño metodológico, la validez de los instrumentos de medición, el tamaño de la muestra y la coherencia entre los hallazgos y las conclusiones. La extracción de los datos se realizó mediante un formulario que registró los autores, año, país, objetivo, metodología, participantes, intervención, instrumento, hallazgos clave y dimensión temática. Dos revisores llevaron a cabo la extracción de forma independiente y resolvieron las discrepancias mediante consenso. Este procedimiento aseguró la confiabilidad de los datos extraídos y minimizó los sesgos de interpretación en la etapa posterior de análisis temático.

Resultados

Conforme al procedimiento metodológico antes descrito, los resultados de la

sistematización del corpus de estudios que formaron parte de la revisión sistemática se organizaron en torno a la Tabla 1. Se recogió de cada investigación los datos de los autores y año de publicación, país de origen, objetivo perseguido, metodología empleada, participantes, tipo de intervención, instrumento de evaluación, principales resultados y dimensión temática asignada. Los elementos resumidos permiten la comparación horizontal entre estudios y la identificación de patrones transversales en las estrategias de enseñanza de la codificación. De esta manera se visibiliza la diversidad de enfoques pedagógicos y la posibilidad de establecer relaciones entre el tipo de intervención, los resultados obtenidos y el contexto de aplicación.

En cuanto a la distribución temporal y geográfica de los estudios, se puede apreciar una concentración de publicaciones en el bienio 2024-2025, con diez y siete estudios en cada año, mientras que 2022 aportó seis investigaciones y 2023 solo una. Esta tendencia indica un interés creciente por el tema en los últimos años. En términos geográficos, Asia lideró con diez estudios (China (5), Taiwán (4), Kazajistán (1)), seguida por América Latina con cinco (Ecuador (3), Colombia (1), Chile, Perú y Ecuador (1)), Europa con seis (España (2), Austria (1), Finlandia (1), Italia (1), Luxemburgo (1)), y contribuciones de Malta (1), Turquía (1) y un estudio multinacional europeo. La predominancia asiática refleja las políticas gubernamentales de integración temprana de la programación en esa región. La representación latinoamericana, aunque menor, evidencia un impulso investigativo emergente. La escasa presencia africana y de Oceanía señala carencia en la producción académica que merecen atención futura.

Respecto a los objetivos de los estudios revisados, la mayoría de las investigaciones persiguió evaluar o desarrollar el PC mediante distintas intervenciones de codificación. Once estudios tuvieron como propósito medir el impacto de una herramienta o estrategia sobre el PC, seis buscaron comparar la efectividad de dos o más enfoques, tres apuntaron a identificar diferencias por edad o género en el desarrollo de habilidades computacionales, y los restantes se propusieron diseñar marcos pedagógicos, explorar actitudes de los estudiantes o validar instrumentos de evaluación. Esta distribución refleja que el campo privilegia la evaluación de resultados sobre la comprensión de los procesos cognitivos subyacentes al aprendizaje de la programación. Los objetivos comparativos ganaron terreno reciente, lo cual indica una maduración del campo hacia la indagación sobre la eficacia relativa de las estrategias disponibles en lugar de la mera demostración de su utilidad.

En lo que concierne a las metodologías empleadas, el cuasiexperimental predominó con nueve estudios, seguido del experimental con siete, el descriptivo con dos, y el mixto, cualitativo y computacional con una contribución cada uno. La hegemonía del diseño cuasiexperimental responde a la dificultad de asignar al azar participantes en contextos escolares reales, donde los grupos naturales de clase constituyen la unidad de análisis. Los estudios experimentales correspondieron a investigaciones con mayor control de variables. Las aproximaciones cualitativas aportaron comprensión de los factores contextuales que influyen en la implementación de las estrategias. La presencia de una investigación computacional con modelos Long Short-Term Memory (LSTM) señala una incorporación incipiente de técnicas de Inteligencia Artificial (IA) en el análisis de datos educativos. Esta diversidad metodológica enriquece el campo, aunque también dificulta la comparación directa de los hallazgos entre estudios.

En relación con los participantes, en los estudios se involucra varios estudiantes de educación primaria, con muestras entre 24 y 312 participantes. Vidal et al. (2024) trabajaron con 195 estudiantes en tres países latinoamericanos. La mediana fue de 88 estudiantes por estudio. En 19 estudios trabajaron con estudiantes de primaria, tres incluyeron niños de 5 y 6 años, y los restantes abarcaron rangos de edad más amplios. La participación de docentes como sujetos de investigación resultó limitada, pues solo dos estudios incorporaron la perspectiva del profesorado. Esta concentración en los alumnos refleja un enfoque centrado en los resultados de aprendizaje más que en los procesos de enseñanza. Las muestras de mayor tamaño correspondieron a los estudios transversales y cuasiexperimentales y los de caso y cualitativos operaron con grupos más reducidos, pero con mayor profundidad analítica. La edad de los participantes influyó en el tipo de herramienta seleccionada.

En lo referente a las intervenciones implementadas, la programación por bloques con Scratch constituyó la estrategia más recurrente, presente en nueve estudios. Caiza et al. (2025) evaluaron Micro:bit para el desarrollo del PC y Molano (2024) exploró la programación de robots educativos. Schwartz et al. (2024) investigaron la plataforma tangible Kniwwelino, y Kastner et al. (2022) combinaron programación por bloques con computación física mediante Arduino. Le siguieron intervenciones con Python gamificada en dos estudios, y experiencias con sistemas tutores inteligentes y modelos de IA. Las intervenciones oscilaron entre sesiones únicas y programas de hasta diez sesiones, con una mediana de seis. Scratch dominó como herramienta principal por su accesibilidad, aunque la emergencia de plataformas basadas en IA y robótica tangible señala una diversificación de las opciones didácticas para la enseñanza de la codificación en la educación primaria.

En relación con los instrumentos de evaluación, los test de PC representaron la herramienta de medición más utilizada, con presencia en 17 estudios. Entre estos destacan el Test de Bebras, el Beginners' CT Test y pruebas de elaboración propia. Cuatro estudios combinaron test estandarizados con cuestionarios de motivación o engagement, tres emplearon escalas de autoeficacia o usabilidad, y dos utilizaron entrevistas como complemento cualitativo. Los estudios de caso y cualitativos recurrieron a la observación, el análisis de patrones de interacción y los modelos de aprendizaje automático. La diversidad de los instrumentos limita la comparabilidad de los resultados entre investigaciones, pues cada test mide dimensiones distintas del PC. La ausencia de un marco de evaluación unificado constituye una debilidad del campo que afecta la acumulación de evidencia. No obstante, la combinación de instrumentos cuantitativos y cualitativos en varios estudios aporta una visión más completa de los fenómenos educativos investigados.

Con respecto a los principales resultados, en 22 estudios reportaron mejoras en el PC o en habilidades asociadas. Angamarca y Andrade (2022) documentaron una mejora significativa en el razonamiento lógico con Scratch y Gamito et al. (2022) confirmaron avances notables en abstracción con la misma herramienta. Yang y Lin (2024) demostraron que los organizadores gráficos potencian la descomposición algorítmica. Las mejoras más consistentes se observaron en descomposición y abstracción, seguidas por depuración y pensamiento algorítmico. Tres estudios identificaron interacciones de género y experiencia previa con resultados divergentes. Un estudio halló que la robótica

generó efectos más sostenidos que la programación desconectada. La motivación y el engagement emergieron como mediadores relevantes del aprendizaje. Estos resultados coinciden en señalar que la codificación, bajo condiciones pedagógicas adecuadas, contribuye al desarrollo cognitivo de los estudiantes de primaria, aunque la magnitud del efecto varía según la estrategia y el contexto.

A este aspecto se une las dimensiones temáticas, los estudios se agruparon en seis categorías: herramientas digitales (siete estudios), estrategias pedagógicas (cinco), programación por bloques (tres), sistemas inteligentes (tres), gamificación y programación desconectada (dos en cada caso) y brechas de género junto con aprendizaje colaborativo (dos y uno en cada caso). La categoría de herramientas digitales concentró el mayor número de estudios, lo cual indica que la investigación prioriza la evaluación de tecnologías específicas sobre el análisis de los procesos de enseñanza subyacentes. Las estrategias pedagógicas —andamiaje, organizadores gráficos y progresión escalonada— recibieron atención creciente en los años más recientes. Los sistemas inteligentes, con modelos LSTM y tutores adaptativos, representan una frontera emergente. La gamificación y la programación desconectada ofrecieron contribuciones relevantes sobre la motivación y la reducción de la carga cognitiva en cada ámbito respectivo.

.

Tabla 1. Síntesis de los estudios incluidos en la revisión sistemática que estudian estrategias de enseñanza de la codificación en la educación primaria

N.º	Autores (año) / País	Objetivo	Metodología / Participantes	Intervención	Instrumento	Principales resultados	Dimensión temática
1	Caiza et al. (2025) / Ecuador	Evaluar Micro:bit para desarrollo del PC.	Cuasiexperimental / 58 estudiantes de 7-14 años y 9 profesores.	Programación con Micro:bit.	Test de PC Bebras.	Mejora en descomposición y abstracción.	Herramientas digitales.
2	Gao et al. (2025) / China	Efecto de mapas mentales como andamiaje en PC.	Cuasiexperimental / 104 estudiantes de educación primaria.	Mapas mentales + programación bloques.	Test de PC.	Andamiaje con mapas mentales eleva CT.	Estrategias pedagógicas.
3	Hsu (2025) / Taiwán	Desarrollo de PC mediante IA generativa y prompting.	Experimental / 108 estudiantes de educación primaria.	Actividades con LLM y prompting.	Test de PC y entrevistas.	Prompting con IA desarrolla habilidades de PC.	Sistemas inteligentes.
4	Lai y Lin (2025) / Taiwán	Analizar comportamientos con sistema tutor inteligente Intelligent Tutoring System for Coding and Learning (ITS-CAL).	Cuasiexperimental / 143 estudiantes de educación primaria.	Sistema tutor inteligente ITS-CAL.	Análisis logs de aprendizaje.	ITS adaptativo mejora rendimiento en codificación.	Sistemas inteligentes.
5	Sun y Liu (2025) / China	Comparar CodeCombat Python y Scratch en PC de sexto grado.	Experimental / 87 estudiantes de sexto grado.	CodeCombat Python vs. Scratch.	Test de PC.	Mejoras significativas de las habilidades de PC de los estudiantes.	Herramientas digitales.
6	Vassallo (2025) / Malta	Fomentar PC con enfoque iterativo.	Cualitativo-iterativo / 52 estudiantes de educación primaria.	Progresión unplugged-conectada.	Observación y evaluación.	Progresión escalonada reduce carga cognitiva.	Programación desconectada.
7	Zhang et al. (2025) / China	Comparar unplugged vs. robótica en preescolar.	Experimental / 80 niños de 5-6 años.	Actividades unplugged vs. robótica.	Test de PC y funciones ejecutivas.	Robótica tuvo impacto más sostenido que unplugged.	Programación desconectada.
8	Barradas et al. (2024) / Europa (múltiples países)	Robótica móvil basada en Arduino para PC.	Experimental / 168 estudiantes de educación primaria.	Robótica Arduino con aprendizaje basado en problemas.	Test de PC	Robótica con aprendizaje basado en problemas mejora PC.	Herramientas digitales.

N.º	Autores (año) / País	Objetivo	Metodología / Participantes	Intervención	Instrumento	Principales resultados	Dimensión temática
9	Belessova et al. (2024) / Kazajistán	Medir impacto de Scratch en engagement.	Cuasiexperimental / 210 estudiantes de educación primaria.	Programación Scratch en clases.	Encuesta engagement y notas.	Aumento de participación y rendimiento académico.	Programación por bloques.
10	Jaimez et al. (2024) / Ecuador	Evaluar app web para crear juegos y lógica.	Descriptivo / 45 estudiantes de educación primaria.	App web de creación de juegos.	Escala de usabilidad.	Mejora en comprensión lógica de programación.	Gamificación.
11	Kalemkuş y Kalemkuş (2024) / Turquía	Diseñar experimentos científicos con programación visual.	Cuasiexperimental / 76 estudiantes de educación primaria.	Diseño de experimentos con Scratch.	Escala autoeficacia CT.	Integración Scratch-ciencia mejora autoeficacia.	Estrategias pedagógicas.
12	Molano (2024) / Colombia	Desarrollar PC mediante programación de robots.	Mixto / 90 estudiantes.	Programación de robots educativos.	Test PC y entrevistas.	Incremento del PC y motivación estudiantil	Herramientas digitales.
13	Schwartz et al. (2024) / Luxemburgo	Enseñar PC con plataforma tangible Kniwwelino.	Estudio de campo / 64 estudiantes de educación primaria.	Programación tangible Kniwwelino.	Cuestionario y observación.	Plataforma tangible facilita comprensión de CT.	Herramientas digitales.
14	Su et al. (2024) / Taiwán	Cultivar creatividad y codificación con andamiaje.	Experimental / 96 estudiantes de educación primaria.	Andamiaje general y específico.	Test creatividad y codificación.	Andamiaje específico mejora más la codificación.	Estrategias pedagógicas.
15	Sun y Liu (2024) / China	Efectos de Python gamificado en PC por género.	Cuasiexperimental / 156 estudiantes de educación primaria.	Programación Python gamificada (CodeCombat).	Test de PC.	Mejora de PC y reducción de brecha de género.	Gamificación.
16	Vidal et al. (2024) / Chile, Ecuador, Perú	Desarrollar competencias de programación con herramientas de bloques.	Descriptivo / 195 estudiantes de educación primaria.	Scratch con sistema recomendador.	Evaluación de competencias.	Contexto sociocultural influye en desarrollo de competencias.	Herramientas digitales.
17	Yang y Lin (2024) / Taiwán	Mejorar PC con organizadores gráficos.	Experimental / 88 estudiantes de educación primaria.	Programación con organizadores gráficos.	Test de PC	Organizadores gráficos mejoran descomposición.	Estrategias pedagógicas
18	Imbernón et al. (2023) / España	Identificar necesidades de ayuda en Scratch con LSTM.	Computacional / 78 estudiantes de educación primaria.	Análisis de interacciones Scratch.	Modelo LSTM.	Predicción de dificultades con 85% de precisión.	Sistemas inteligentes.
19	Angamarca y Andrade (2022) / Ecuador	Enseñar programación con Scratch para razonamiento.	Cuasiexperimental / 38 niños de entre 8 a 15 años	Talleres Scratch (10 sesiones)	Test razonamiento lógico	Mejora en razonamiento lógico	Programación por bloques.

N.º	Autores (año) / País	Objetivo	Metodología / Participantes	Intervención	Instrumento	Principales resultados	Dimensión temática
20	Fagerlund et al. (2022) / Finlandia	PC en programación en pareja con Scratch.	Estudio de caso / 24 estudiantes de cuarto grado.	Programación en pareja Scratch.	Análisis de patrones CT.	Pair programming desarrolla planificación y depuración.	Aprendizaje colaborativo.
21	Gamito et al. (2022) / España	Valorar PC con experiencia Scratch.	Cuasiexperimental / 62 estudiantes de 8-12 años.	Talleres Scratch (8 sesiones).	Test PC y cuestionario.	Scratch mejora PC, en especial abstracción.	Herramientas digitales.
22	Jiang y Wong (2022) / China	Explorar diferencias de edad y género en PC.	Transversal / 312 estudiantes de educación primaria	Actividades de programación.	Test de PC por edad.	Diferencias por edad y género en PC.	Brechas de género.
23	Kastner et al. (2022) / Austria	Efectos combinados de programación por bloques y computación física.	Cuasiexperimental / 72 estudiantes de educación primaria.	Scratch + Arduino.	Test de PC.	Combinación de bloques y computación física mejora PC.	Estrategias pedagógicas.
24	Montuori et al. (2022) / Italia	Explorar diferencias de género en codificación.	Experimental / 96 niños al inicio de primaria.	Tareas de programación.	Tareas de codificación.	Sin diferencias significativas al inicio de primaria.	Brechas de género.

Discusión

Derivado del análisis de resultados, la primera dimensión que merece atención concierne al papel de las herramientas visuales y tangibles en el desarrollo del PC. Siete estudios del corpus demostraron que plataformas como Scratch, Micro:bit y Kniwwelino mejoraron las habilidades de descomposición, abstracción y pensamiento algorítmico en estudiantes de educación primaria. Los resultados indican que la manipulación directa de objetos programables facilita la comprensión de conceptos computacionales abstractos, mientras que las interfaces de pantalla ofrecen ventajas en la rapidez de iteración. Estos hallazgos coinciden con [Yang et al. \(2025\)](#), quienes argumentaron que la programación visual por bloques desarrolla habilidades de codificación distintas del PC y que ambas dimensiones requieren evaluaciones separadas. La distinción entre codificación y PC resulta crucial para evitar interpretaciones infladas sobre el alcance de las intervenciones con herramientas visuales.

En relación con lo anterior, el andamiaje pedagógico resultó un factor determinante para la efectividad de las intervenciones de codificación. Los estudios del corpus que implementaron andamiaje específico como organizadores gráficos, mapas mentales y progresiones estructuradas, reportaron mejoras superiores en comparación con las condiciones de descubrimiento libre. [Su et al. \(2024\)](#) demostraron que el andamiaje de dominio específico superó al de dominio general en la mejora de habilidades de codificación, y [Gao et al. \(2025\)](#) confirmaron que los mapas mentales como soporte incrementaron el PC de forma notable. Estos resultados alinean con [Hutchison et al. \(2025\)](#), quienes propusieron que el andamiaje a través de la lectoescritura digital en la plataforma Compose and Code facilita la integración de la codificación con habilidades lingüísticas. El andamiaje acelera la adquisición de competencias y reduce la frustración y promueve la autorregulación del aprendizaje en los estudiantes más jóvenes.

Unido a esto, la integración curricular de la codificación con otras áreas del conocimiento constituye una dimensión que trasciende la instrucción aislada de la programación. [Kalemkuş y Kalemkuş \(2024\)](#) demostraron que la integración de Scratch con el diseño de experimentos científicos mejoró la autoeficacia en PC y [Barradas et al. \(2024\)](#) evidenciaron que la robótica con aprendizaje basado en problemas potenció el PC de modo considerable. Estas aproximaciones interdisciplinarias superan la visión de la codificación como actividad extracurricular y la posicionan como competencia transversal. [Tsortanidou et al. \(2023\)](#) documentaron que la programación desconectada integrada en las asignaturas del currículo ordinario de kindergarten a sexto grado desarrolla el PC sin requerir recursos tecnológicos sofisticados. Sin embargo, la integración efectiva demanda planificación docente y coherencia entre los objetivos de la codificación y los de las áreas curriculares vinculadas.

En otra dimensión de análisis, la progresión de lo desconectado a lo conectado representa una estrategia pedagógica que reduce la carga cognitiva de los estudiantes en las etapas iniciales del aprendizaje de la programación. [Vassallo \(2025\)](#) demostró que la progresión escalonada desde actividades desconectadas hasta la programación con herramientas digitales disminuyó la sobrecarga cognitiva en estudiantes de primaria. [Zhang et al. \(2025\)](#) compararon actividades desconectadas y con robótica en niños de 5 y 6 años, y

concluyeron que la robótica produjo un impacto más sostenido, aunque ambas aproximaciones mejoraron el PC. Estos resultados dialogan con [Montuori et al. \(2023\)](#), quienes combinaron entrenamiento desconectado y robótica educativa para promover el PC y las habilidades cognitivas en preescolares, con resultados favorables en ambas dimensiones. La progresión gradual permite a los niños construir representaciones mentales de los conceptos computacionales antes de enfrentar la complejidad de los lenguajes formales.

Por otro lado, la narrativa y la gamificación potenciaron la motivación de los estudiantes hacia la codificación, aunque con matices relevantes. [Sun y Liu \(2024\)](#) demostraron que la programación de Python gamificada con CodeCombat mejoró el PC y redujo la brecha de género en la participación. [Jaimez et al. \(2024\)](#) confirmaron que las aplicaciones web de creación de juegos incrementaron la comprensión lógica de la programación. [Belessova et al. \(2024\)](#) reportaron aumentos de engagement y rendimiento académico con Scratch en escuelas primarias. Sin embargo, [Nevrelova et al. \(2024\)](#) advirtieron que la realidad aumentada requiere un diseño pedagógico cuidadoso para evitar que la novedad tecnológica opaque los objetivos de aprendizaje. [Dong et al. \(2025\)](#) añadieron que las insignias digitales mejoran la motivación y el PC, pero su efecto depende del diseño del sistema de retroalimentación. La gamificación precisa una articulación deliberada con los propósitos formativos.

No menos importante, las brechas de género en la codificación persisten como desafío pese a los avances en las estrategias de enseñanza. [Jiang y Wong \(2022\)](#) identificaron diferencias por edad y género en el PC de estudiantes de primaria, con ventaja para los varones. [Montuori et al. \(2022\)](#), en contraste, no hallaron diferencias de género al inicio de la primaria, lo cual sugiere que la brecha emerge con la exposición a estereotipos socioculturales. [Sun y Liu \(2025\)](#) reportaron efectos de interacción entre género y experiencia previa en la comparación de CodeCombat y Scratch. Estos hallazgos convergen con [Master et al. \(2023\)](#), quienes demostraron inequidades de género en la preparación motivacional desde la infancia temprana, y con [Jacob et al. \(2022\)](#), quienes documentaron barreras adicionales para los estudiantes multilingües en ciencias de la computación. Las intervenciones deben incorporar una perspectiva de género desde su diseño.

A este aspecto se une, que la evaluación del PC y los sistemas inteligentes constituyen una dimensión relevante con implicaciones pedagógicas y tecnológicas. [Imbernón et al. \(2023\)](#) lograron predecir dificultades en Scratch con un 85% de precisión mediante modelos LSTM y [Lai y Lin \(2025\)](#) demostraron que el sistema tutor inteligente ITS-CAL mejora el rendimiento de forma adaptativa. [Hsu \(2025\)](#) mostró que el prompting con IA generativa desarrolla habilidades de PC. Estos avances precisan marcos de evaluación sólidos: [Kim et al. \(2024\)](#) exploraron consideraciones de diseño curricular de PC en educación infantil, [Ukkonen et al. \(2025\)](#) evidenciaron limitaciones en la comprensión docente de la evaluación, [El-Hamamsy et al. \(2022\)](#) validaron pruebas de PC para primeros grados, [Aristizábal et al. \(2024\)](#) diseñaron un test para esos niveles, [Kong y Lai \(2022\)](#) aplicaron teoría de respuesta al ítem para validar pruebas en educación primaria.

Cabe destacar que la formación docente constituye el factor crítico que condiciona la implementación efectiva de todas las estrategias identificadas. Los resultados del corpus reflejan que las intervenciones exitosas requieren docentes con competencias

disciplinarias y pedagógicas en codificación, condición que no siempre se cumple. [Fagerlund et al. \(2022\)](#) demostraron que la programación en pareja con Scratch desarrolla habilidades de planificación y depuración, pero su implementación exige docentes capacitados para facilitar la colaboración. [Killen et al. \(2023\)](#) documentaron que los programas de formación docente para integrar el PC en ciencias de primaria mejoran tanto la autoeficacia como la práctica pedagógica. [Kapoor et al. \(2022\)](#) añadieron que el desarrollo profesional virtual apoya el conocimiento y la autoeficacia de los docentes de primeros grados en codificación. Sin inversión sostenida en formación del profesorado, las políticas de integración de la codificación carecen de viabilidad real en los contextos escolares ordinarios.

Conclusión

A partir de los estudios sistematizados se pudo apreciar que las estrategias de enseñanza de la codificación en la educación primaria se articulan en torno a seis dimensiones temáticas: herramientas digitales, programación por bloques, estrategias pedagógicas, sistemas inteligentes, gamificación y programación desconectada. Las herramientas visuales y tangibles mejoran el PC, en particular las dimensiones de descomposición y abstracción, aunque su efectividad depende del andamiaje proporcionado. La integración curricular de la codificación con otras áreas del conocimiento produce resultados superiores a la instrucción aislada, y la progresión de lo desconectado a lo conectado reduce la carga cognitiva en los primeros niveles. La gamificación incrementa la motivación, pero no elimina las brechas de género persistentes en el PC. Los sistemas inteligentes y las evaluaciones adaptativas representan una oportunidad emergente, aunque su implementación requiere marcos de evaluación validados y docentes con formación adecuada para aprovechar su potencial pedagógico.

En cuanto a las implicaciones y las direcciones futuras, los resultados indican que las políticas educativas deben priorizar la formación continua del profesorado en codificación y PC, condición indispensable para la implementación efectiva de cualquier estrategia. Las investigaciones futuras necesitan diseños longitudinales que evalúen la sostenibilidad de los efectos más allá del período inmediato. La comparación rigurosa entre estrategias con asignación aleatoria constituye una prioridad para avanzar en la identificación de las aproximaciones más efectivas según la edad y el contexto. Asimismo, resulta imperativo desarrollar instrumentos de evaluación estandarizados para los primeros grados. La perspectiva de género y la exploración de contextos poco representados completan la agenda investigativa.

Acerca de

Contribución de los autores: Los autores contribuyeron a la conceptualización del estudio, al desarrollo metodológico, al análisis e interpretación de los datos, a la redacción del manuscrito y a la revisión crítica de su contenido intelectual.

Financiamiento: Los autores declaran que no recibieron financiamiento para esta investigación.

Certificación ética: La naturaleza de la presente investigación, consistente en una revisión sistemática de literatura científica publicada, no requiere de certificación ética, por cuanto no involucra participantes humanos ni datos sensibles.

Referencias

- Angamarca, O., y Andrade, D. (2022). Enseñanza de programación a niños de edad escolar utilizando Scratch para mejora del razonamiento lógico. *ProSciences: Revista de Producción, Ciencias e Investigación*, 6(42), 111-121. <https://doi.org/10.29018/issn.2588-1000vol6iss42.2022pp111-121>
- Aristizábal, J. H., Gutiérrez Posada, J. E., y Diago, P. D. (2024). Design and validation of a computational thinking test for children in the first grades of elementary education. *Multimodal Technologies and Interaction*, 8(5), 39. <https://doi.org/10.3390/mti8050039>
- Barradas, R., Lencastre, J. A., Soares, S. P., y Valente, A. (2024). Arduino-based mobile robotics for fostering computational thinking development: An empirical study with elementary school students using problem-based learning across Europe. *Robotics*, 13(11), 159. <https://doi.org/10.3390/robotics13110159>
- Belessova, D., Ibashova, A., Zhidebayeva, A., Shaimerdenova, G., y Nakhipova, V. (2024). The impact of 'Scratch' on student engagement and academic performance in primary schools. *Open Education Studies*, 6(1), 20220228. <https://doi.org/10.1515/edu-2022-0228>
- Caiza, F. A., Oña, J. P., y Salazar, C. E. (2025). Micro:bit para desarrollar el pensamiento computacional en estudiantes entre 7 y 14 años. *MENTOR Revista de Investigación Educativa y Deportiva*, 4(12), 523-542. <https://doi.org/10.56200/mried.v4i12.10302>
- Dong, Y., Ma, H., Li, H., Jing, B., y Liu, H. (2025). Effects of digital badges on pupils' computational thinking and learning motivation in computer science. *Acta Psychologica*, 254, 104824. <https://doi.org/10.1016/j.actpsy.2025.104824>
- El-Hamamsy, L., Bruno, B., Audrin, C., Chevalier, M., Avry, S., y Dehler Zufferey, J. (2023). How are primary school computer science curricular reforms contributing to equity? Impact on student learning, perception of the discipline, and gender gaps. *International Journal of STEM Education*, 10, 60. <https://doi.org/10.1186/s40594-023-00438-3>
- El-Hamamsy, L., Zapata, M., Marcelino, P., Bruno, B., Dehler Zufferey, J., Martín, E. y Román, M. (2022). Comparing the psychometric properties of two primary school Computational Thinking (CT) assessments for grades 3 and 4: The Beginners' CT test (BCTt) and the competent CT test (cCTt). *Frontiers in Psychology*, 13, 1082659. <https://doi.org/10.3389/fpsyg.2022.1082659>
- Fagerlund, J., Vesisenaho, M., y Häkkinen, P. (2022). Fourth grade students' computational thinking in pair programming with Scratch: A holistic case analysis. *International Journal of Child-Computer Interaction*, 33, 100511. <https://doi.org/10.1016/j.ijcci.2022.100511>

- Falloon, G. (2024). Advancing young students' computational thinking: An investigation of structured curriculum in early years primary schooling. *Computers & Education*, 216, 105045. <https://doi.org/10.1016/j.compedu.2024.105045>
- Gamito, R., Aristizabal, P., Basasoro, M., y León, I. (2022). El desarrollo del pensamiento computacional en educación: valoración basada en una experiencia con Scratch. *Innoeduca. International Journal of Technology and Educational Innovation*, 8(1), 59-74. <https://doi.org/10.24310/innoeduca.2022.v8i1.12093>
- Gao, X., Yang, Y., Du, Y., y Sun, D. (2025). Effect of mind mapping-based scaffolding on elementary students' computational thinking in block-based programming. *Journal of Educational Computing Research*, 63(1), 236-271. <https://doi.org/10.1177/07356331241303067>
- Hsu, H. P. (2025). From programming to prompting: Developing computational thinking through large language model-based generative artificial intelligence. *TechTrends*, 69, 485-506. <https://doi.org/10.1007/s11528-025-01052-6>
- Hutchison, A., Si, Q., Colwell, J., Kaya, E., Jakeway, E., Miller, B., Gutierrez, K., Regan, K., y Evmenova, A. (2025). Scaffolding coding instruction through literacy via the Compose and Code digital platform and curriculum. *Journal of Computer Assisted Learning*, 41, e13115. <https://doi.org/10.1111/jcal.13115>
- Imbernón, L. E., Manjarrés, Á., y de la Paz, F. (2023). Using LSTM to identify help needs in primary school Scratch students. *Applied Sciences*, 13(23), 12869. <https://doi.org/10.3390/app132312869>
- Jacob, S., Montoya, J., Nguyen, H., Richardson, D. y Warschauer, M. (2022). Examining the what, why, and how of multilingual student identity development in computer science. *ACM Transactions on Computing Education*, 22(3), 1-33. <https://doi.org/10.1145/3500918>
- Jaimez, C. R., García, B., y Erazo, J. (2024). Evaluación de aplicación web para la creación de juegos que fomentan el aprendizaje de la lógica de programación. *Revista Dilemas Contemporáneos: Educación, Política y Valores*, 12, ed. especial. <https://doi.org/10.46377/dilemas.v12i.4421>
- Jiang, S., y Wong, G. K. W. (2022). Exploring age and gender differences of computational thinkers in primary school: A developmental perspective. *Journal of Computer Assisted Learning*, 38(1), 60-75. <https://doi.org/10.1111/jcal.12591>
- Kalemkuş, J., y Kalemkuş, F. (2024). The effects of designing scientific experiments with visual programming language on learning outcomes. *Science & Education*, 34, 2409–2430. <https://doi.org/10.1007/s11191-024-00546-8>
- Kapoor, M., Yang, Z., y Bers, M. U. (2022). Supporting early elementary teachers' coding knowledge and self-efficacy through virtual professional development. *Journal of Technology and Teacher Education*, 30(4), 461-491. <https://eric.ed.gov/?id=EJ1378127>
- Kastner, O., Tengler, K., Sabitzer, B., y Lavicza, Z. (2022). Combined effects of block-based programming and physical computing on primary students' computational thinking skills. *Frontiers in Psychology*, 13, 875382. <https://doi.org/10.3389/fpsyg.2022.875382>

- Killen, H., Coenraad, M., Byrne, V. L., Cabrera, L., Mills, K., Ketelhut, D. J., y Plane, J. D. (2023). Teacher education to integrate computational thinking into elementary science: A design-based research study. *ACM Transactions on Computing Education*, 23(4), 1-36. <https://doi.org/10.1145/3618115>
- Kim, J., Leftwich, A., y Castner, D. (2024). Beyond teaching computational thinking: Exploring kindergarten teachers' computational thinking and computer science curriculum design considerations. *Education and Information Technologies*, 29, 15191–15227. <https://doi.org/10.1007/s10639-023-12406-z>
- Kong, S. C., y Lai, M. (2022). Validating a computational thinking concepts test for primary education using item response theory: An analysis of students' responses. *Computers & Education*, 185, 104562. <https://doi.org/10.1016/j.compedu.2022.104562>
- Kravik, R., K. Berg, T., y Siddiq, F. (2022). Teachers' understanding of programming and computational thinking in primary education – A critical need for professional development. *Acta Didactica Norden*, 16(4), 23. <https://doi.org/10.5617/adno.9194>
- Lai, C. H., y 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), 1922. <https://doi.org/10.3390/app15041922>
- Master, A., Tang, D., Forsythe, D., Alexander, T. M., Cheryan, S., y Meltzoff, A. N. (2023). Gender equity and motivational readiness for computational thinking in early childhood. *Early Childhood Research Quarterly*, 64, 242-254. <https://doi.org/10.1016/j.ecresq.2023.03.004>
- Molano, W. F. (2024). Desarrollo de pensamiento computacional en los estudiantes de educación utilizando la programación de robots. *Praxis Pedagógica*, 24(37), 186-210. <https://doi.org/10.26620/uniminuto.praxis.24.37.2024.186-210>
- Montuori, C., Pozzan, G., Padova, C., Ronconi, L., Vardanega, T., y Arfé, B. (2023). Combined unplugged and educational robotics training to promote computational thinking and cognitive abilities in preschoolers. *Education Sciences*, 13(9), 858. <https://doi.org/10.3390/educsci13090858>
- Montuori, C., Ronconi, L., Vardanega, T., y Arfé, B. (2022). Exploring gender differences in coding at the beginning of primary school. *Frontiers in Psychology*, 13, 887280. <https://doi.org/10.3389/fpsyg.2022.887280>
- Nevrelova, N., Koreňová, L., Lavicza, Z., Brůžková, N., y Schmid, A. (2024). Enhancing digital literacy in primary education through augmented reality. *Frontiers in Education*, 9, 1390491. <https://doi.org/10.3389/feduc.2024.1390491>
- Pellas, N. (2025). Enhancing computational thinking, spatial reasoning, and executive function skills: The impact of tangible programming tools in early childhood and across different learner stages. *Journal of Educational Computing Research*, 63(1), 3-32. <https://doi.org/10.1177/07356331241292767>
- Schwartz, L., Maquil, V., Johannsen, L., Moll, C., y Hermen, J. (2024). Teaching computational thinking with a tangible development platform: An exploratory field

study at school with Kniwwelino. *Education and Information Technologies*, 29, 4935-4967. <https://doi.org/10.1007/s10639-023-11983-3>

Su, S. W., Chen, L. X., Yuan, S. M., y Sun, C. T. (2024). Cultivating creativity and improving coding skills in primary school students via domain-general and domain-specific learning scaffoldings. *Education Sciences*, 14(7), 695. <https://doi.org/10.3390/educsci14070695>

Sun, L., y Liu, J. (2024). Effects of gamified Python programming on primary school students' computational thinking skills: A differential analysis of gender. *Journal of Educational Computing Research*, 62(3), 626-654. <https://doi.org/10.1177/07356331231225269>

Sun, L., y Liu, J. (2025). Comparative study of CodeCombat-based Python programming and Scratch programming effects on sixth graders' computational thinking: Interaction effects of gender and programming experience. *Thinking Skills and Creativity*, 57, 101852. <https://doi.org/10.1016/j.tsc.2025.101852>

Tsortanidou, X., Daradoumis, T., y Barberá, E. (2023). Unplugged computational thinking at K-6 education: Evidence from a multiple-case study in Spain. *Education 3-13*, 51(6), 948-965. <https://doi.org/10.1080/03004279.2022.2029924>

Ukkonen, A., Pajchel, K., y Mifsud, L. (2025). Teachers' understanding of assessing computational thinking. *Computer Science Education*, 35(4), 794-819. <https://doi.org/10.1080/08993408.2024.2365566>

Unahalekhaka, A., y Bers, M. U. (2022). Evaluating young children's creative coding: Rubric development and testing for ScratchJr projects. *Education and Information Technologies*, 27, 6577-6597. <https://doi.org/10.1007/s10639-021-10873-w>

Vassallo, D. (2025). Fostering computational thinking in early learners: An iterative approach in a Maltese primary school. *Discover Education*, 4, 126. <https://doi.org/10.1007/s44217-025-00553-z>

Vidal, C., Cárdenas, J., Tupac, M., Serrano, J., y Sánchez, A. (2024). Developing programming competencies in school-students with block-based tools in Chile, Ecuador, and Peru. *IEEE Access*, 12, 118924-118936. <https://doi.org/10.1109/access.2024.3449228>

Wang, X., Wan, F., y Dai, J. (2026). Enhancing computational thinking through coding education in primary school students: An experimental study on the impact of early programming exposure on problem-solving skills. *Frontiers in Psychology*, 17, 1734482. <https://doi.org/10.3389/fpsyg.2026.1734482>

Yang, T. C., y Lin, Z. S. (2024). Enhancing elementary school students' computational thinking and programming learning with graphic organizers. *Computers & Education*, 209, 104962. <https://doi.org/10.1016/j.compedu.2023.104962>

Yang, Z., Blake, J., Yang, D., y Bers, M. U. (2025). The impact of a block-based visual programming curriculum: Untangling coding skills and computational thinking. *Learning and Instruction*, 95, 102041. <https://doi.org/10.1016/j.learninstruc.2024.102041>

Yu, Q., Yu, K., y Li, B. (2025). Effects of block-based visual programming on K-12 students' learning outcomes. *Journal of Educational Computing Research*, 63(1), 64-98. <https://doi.org/10.1177/07356331241293163>

Zhang, S., y Wong, G. K. W. (2023). Development and validation of a computational thinking test for lower primary school students. *Educational Technology Research and Development*, 71, 1595-1630. <https://doi.org/10.1007/s11423-023-10231-2>

Zhang, X., Chen, Y., Hu, L., Hwang, G. J., y Tu, Y. F. (2025). Developing preschool children's computational thinking and executive functions: Unplugged vs. robot programming activities. *International Journal of STEM Education*, 12, 10. <https://doi.org/10.1186/s40594-024-00525-z>