

SSTSDS-Vol.4. N1. 072

Aplicación del pensamiento computacional en educación: estrategias para el desarrollo de competencias del siglo XXI

Application of computational thinking in education: strategies for the development of 21st-century competencies

Autores:

Jorge Eduardo Fernández Acevedo
Universidad Nacional de Chimborazo
Riobamba – Ecuador

jorgefernandez@unach.edu.ec
<https://orcid.org/0009-0003-0833-032X>

José Luis García
Universidad Nacional de Chimborazo
Riobamba – Chimborazo, Ecuador
josel.garcia@unach.edu.ec
<https://orcid.org/0009-0001-3741-3680>

Anabel Lisbeth Naranjo Brito
Unidad Educativa Simón Bolívar
Naranjal – Ecuador
anabelnaranjo1998@gmail.com
<https://orcid.org/0009-0009-5024-3952>

Karolayn Alejandra Sánchez Ronquillo
Investigadora Independiente
Guayaquil – Ecuador
karitosan08@gmail.com
<https://orcid.org/0009-0004-8382-014X>

Félix Javier Gómez Freire
Institución Educativa Fiscal Guayllabamba
Quito – Ecuador
felixj.gomez@educacion.gob.ec
<https://orcid.org/0009-0005-2147-2095>

Autor de correspondencia: Jorge Eduardo Fernández Acevedo, jorgefernandez@unach.edu.ec

Recepción: 27-abril-2026

Aceptación: 29-mayo-2026

Publicación: 28-junio-2026



Cómo citar este artículo:

Fernández Acevedo, J. E. ., García, J. L. ., Naranjo Brito, A. L. ., Sánchez Ronquillo, K. A. ., & Gómez Freire, F. J. . (n.d.). Aplicación del pensamiento computacional en educación: estrategias para el desarrollo de competencias del siglo XXI. *Sage Sphere of Technology, Sciences, Discoveries And Society*. <https://doi.org/10.63688/806c5n20>

© 2026; Los autores. Este es un artículo en acceso abierto, distribuido bajo los términos de una licencia Creative Commons (<https://creativecommons.org/licenses/by/4.0>) que permite el uso, distribución y reproducción en cualquier medio, siempre que la obra original sea correctamente citada.



RESUMEN

El pensamiento computacional constituye una competencia fundamental para enfrentar los desafíos educativos, sociales y tecnológicos del siglo XXI, porque permite formular problemas, organizar información, identificar patrones y diseñar soluciones lógicas y sistemáticas. El objetivo de este artículo fue analizar la aplicación del pensamiento computacional en la educación y reconocer las estrategias pedagógicas que contribuyen al desarrollo del pensamiento crítico, la creatividad, la colaboración, la competencia digital y la resolución de problemas. La investigación se desarrolló mediante una revisión sistemática de literatura con enfoque cualitativo, documental y analítico, siguiendo las directrices PRISMA 2020. Se consultaron bases académicas como Scopus, Web of Science, ScienceDirect, ACM Digital Library, SpringerLink, ERIC y Google Scholar, y se incluyeron 21 estudios publicados entre 2006 y 2024. Los resultados evidencian que la programación visual, la robótica educativa, las actividades desenchufadas, el aprendizaje basado en problemas y los proyectos interdisciplinarios favorecen la descomposición, la abstracción, el diseño algorítmico y la depuración. Se concluye que su integración requiere formación docente, criterios de evaluación pertinentes y acceso equitativo a recursos tecnológicos.

Palabras clave: pensamiento computacional, educación, competencias del siglo XXI, programación, resolución de problemas.

ABSTRACT

Computational thinking is a fundamental competency for addressing the educational, social, and technological challenges of the 21st century because it enables learners to formulate problems, organize information, identify patterns, and design logical and systematic solutions. The objective of this article was to analyze the application of computational thinking in education and identify the pedagogical strategies that contribute to the development of critical thinking, creativity, collaboration, digital competence, and problem-solving skills. The study was conducted through a systematic literature review with a qualitative, documentary, and analytical approach, following the PRISMA 2020 guidelines. Academic databases such as Scopus, Web of Science, ScienceDirect, ACM Digital Library, SpringerLink, ERIC, and Google Scholar were consulted, and 21 studies published between 2006 and 2024 were included. The findings show that visual programming, educational robotics, unplugged activities, problem-based learning, and interdisciplinary projects strengthen decomposition, abstraction, algorithmic design, and debugging. It is concluded that effective integration requires teacher training, appropriate assessment criteria, and equitable access to technological resources.

Keywords: computational thinking, education, 21st-century competencies, programming, problem solving.



1. INTRODUCCIÓN

El pensamiento computacional constituye un proceso intelectual mediante el cual las personas formulan problemas, organizan información y diseñan soluciones que pueden ser ejecutadas por seres humanos, computadoras o mediante la combinación de ambos. Esta forma de razonamiento no se limita al manejo de dispositivos tecnológicos, sino que comprende habilidades relacionadas con la descomposición de problemas, la identificación de patrones, la abstracción, el diseño de algoritmos y la evaluación de soluciones. Aunque el concepto adquirió mayor reconocimiento dentro del campo de las ciencias de la computación, actualmente se considera una competencia aplicable a distintos contextos educativos y profesionales. En este sentido, su integración en la educación permite que los estudiantes desarrollen capacidades para comprender situaciones complejas y responder de manera lógica, organizada y creativa ante los desafíos de la sociedad contemporánea (Wing, 2006; Grover & Pea, 2013).

El desarrollo del pensamiento computacional implica que el estudiante pueda dividir un problema complejo en componentes más pequeños, seleccionar la información relevante, establecer relaciones entre los datos y diseñar procedimientos ordenados para obtener una solución. También incluye la capacidad de detectar errores, modificar estrategias y transferir los conocimientos adquiridos a nuevas situaciones. Estas prácticas pueden incorporarse en asignaturas como Matemáticas, Ciencias Naturales, Lengua, Estudios Sociales y Arte, debido a que el pensamiento computacional no pertenece exclusivamente a la programación. Por ejemplo, los estudiantes pueden organizar los acontecimientos de una historia mediante secuencias, analizar datos de un fenómeno natural, construir modelos matemáticos o representar procesos sociales por medio de diagramas. Esta perspectiva interdisciplinaria fortalece la resolución de problemas, la modelización, el análisis de datos y el pensamiento sistémico (Weintrop et al., 2016; Shute et al., 2017).

La aplicación del pensamiento computacional en el aula requiere estrategias pedagógicas activas que permitan a los estudiantes participar en la construcción de sus conocimientos. Entre las principales estrategias se encuentran la programación visual por bloques, la robótica educativa, el aprendizaje basado en problemas, el aprendizaje por proyectos, los juegos de lógica, las simulaciones y las actividades desenchufadas. La programación mediante



plataformas visuales permite crear historias, animaciones y videojuegos mientras se aprenden conceptos como secuencias, ciclos, condiciones y variables. Por su parte, las actividades desenchufadas utilizan tarjetas, movimientos corporales, laberintos, instrucciones y objetos cotidianos para desarrollar habilidades computacionales sin depender de una computadora. Estas estrategias favorecen la participación de estudiantes de diferentes edades y pueden adaptarse a instituciones con acceso limitado a recursos tecnológicos (Lye & Koh, 2014; Hsu et al., 2018).

La robótica educativa y los proyectos colaborativos también permiten relacionar el pensamiento computacional con situaciones concretas. Al construir y programar un robot, por ejemplo, los estudiantes deben plantear objetivos, distribuir responsabilidades, diseñar instrucciones, comprobar resultados y corregir errores. Este proceso fortalece la autonomía, la perseverancia, la creatividad y el trabajo en equipo. De igual manera, la programación puede generar interés y una mayor percepción de capacidad entre los estudiantes cuando las actividades tienen un propósito significativo y permiten observar resultados concretos. No obstante, para alcanzar estos beneficios es necesario que la tecnología sea utilizada como una herramienta para investigar, crear y solucionar problemas, y no solamente como un recurso destinado a reproducir información (Kong et al., 2018; Nouri et al., 2020).

El pensamiento computacional mantiene una relación directa con las competencias del siglo XXI, debido a que fomenta el pensamiento crítico, la comunicación, la colaboración, la creatividad y la competencia digital. Los estudiantes no solo aprenden a seguir instrucciones, sino también a formular preguntas, comparar alternativas, justificar decisiones y diseñar soluciones originales. Estas habilidades resultan necesarias en una sociedad caracterizada por el crecimiento de la inteligencia artificial, la automatización, el análisis de datos y las tecnologías digitales. Sin embargo, la incorporación de la programación no produce automáticamente el desarrollo del pensamiento computacional. Las actividades deben contar con objetivos pedagógicos definidos, acompañamiento docente y oportunidades para reflexionar sobre el proceso seguido, los errores encontrados y las decisiones adoptadas (Nouri et al., 2020; Tikva & Tambouris, 2021).

La evaluación representa otro componente importante dentro de la enseñanza del pensamiento computacional. Las pruebas tradicionales pueden resultar insuficientes para valorar procesos como la creatividad, la depuración de errores, el razonamiento



algorítmico y la capacidad para aplicar soluciones en diferentes contextos. Por esta razón, se recomienda emplear proyectos, portafolios, observaciones, rúbricas, análisis de programas y actividades de resolución de problemas. La evaluación debe considerar tanto el producto final como el procedimiento utilizado por el estudiante, sus intentos, las modificaciones realizadas y la explicación de sus decisiones. La diversidad de instrumentos existentes demuestra los avances alcanzados en este campo, pero también evidencia la necesidad de establecer criterios más claros y válidos para medir el pensamiento computacional en los diferentes niveles educativos (Tang et al., 2020).

La implementación efectiva del pensamiento computacional también depende de la preparación de los docentes. Los educadores necesitan comprender sus componentes, conocer estrategias para integrarlo en las diferentes asignaturas y seleccionar herramientas adecuadas para las características de sus estudiantes. La falta de formación, el acceso desigual a los dispositivos, la conectividad limitada y la ausencia de orientaciones curriculares pueden dificultar su incorporación. Por ello, los programas de capacitación deben combinar conocimientos tecnológicos, pedagógicos y disciplinares, además de promover el diseño colaborativo de actividades. La integración curricular debe desarrollarse de manera progresiva y contextualizada, evitando reducir el pensamiento computacional a la enseñanza mecánica de lenguajes de programación (Grover & Pea, 2013; Hsu et al., 2018).

En este escenario, resulta necesario analizar las estrategias educativas utilizadas para integrar el pensamiento computacional, así como sus aportes al desarrollo de las competencias requeridas en el siglo XXI. La revisión sistemática de la literatura permite identificar las prácticas pedagógicas más frecuentes, las herramientas empleadas, los métodos de evaluación, los beneficios alcanzados y las limitaciones que todavía existen. En consecuencia, el presente estudio tiene como objetivo analizar, mediante una revisión sistemática, la aplicación del pensamiento computacional en la educación, identificando las estrategias pedagógicas que contribuyen al desarrollo del pensamiento crítico, la creatividad, la colaboración, la competencia digital y la resolución de problemas en los estudiantes.

2. MARCO TEÓRICO

Fundamentos cognitivos del pensamiento computacional



El pensamiento computacional no debe confundirse con el simple uso de computadoras o programas digitales. Denning (2017) sostiene que este concepto pierde precisión cuando se aplica a cualquier actividad de resolución de problemas. Para que exista pensamiento computacional, el estudiante debe representar una situación, organizar sus componentes y construir un procedimiento que pueda ser comprobado o automatizado. Esta postura permite evaluar no solo el producto elaborado, sino también el razonamiento empleado.

Desde una perspectiva procesual, Palts y Pedaste (2020) proponen etapas como formulación del problema, descomposición, abstracción, diseño algorítmico, evaluación y generalización. Dichas fases muestran que esta competencia se desarrolla progresivamente y requiere oportunidades para analizar errores, modificar soluciones y transferir conocimientos.

A su vez, Román-González et al. (2017) identificaron relaciones entre pensamiento computacional, razonamiento lógico, capacidad espacial y resolución de problemas. Este aporte sugiere que las actividades educativas deben avanzar desde experiencias concretas hacia representaciones abstractas, considerando las capacidades cognitivas del alumnado.

En síntesis, el pensamiento computacional puede comprenderse como una competencia para formular problemas, seleccionar información relevante, diseñar soluciones y evaluar su funcionamiento. Su valor educativo reside en formar estudiantes capaces de razonar y no únicamente de manejar tecnología.

Estrategias didácticas para su integración curricular

Las actividades desconectadas constituyen una alternativa para desarrollar habilidades computacionales sin depender de dispositivos electrónicos. Chen et al. (2023) determinaron que los juegos, tarjetas, movimientos corporales y objetos físicos pueden favorecer la comprensión de secuencias, patrones y algoritmos. Su aplicación resulta especialmente pertinente en instituciones con recursos tecnológicos limitados.

No obstante, estas actividades alcanzan mayor profundidad cuando se combinan con programación visual y proyectos digitales. El estudiante puede representar primero un algoritmo con acciones físicas y trasladarlo posteriormente a un entorno de programación. Esta transición facilita la comprensión de la relación entre una instrucción y el resultado que produce.

En otro plano, Witherspoon et al. (2017) destacan el aporte de la robótica educativa al desarrollo de la descomposición y la depuración. Cuando un robot no ejecuta



correctamente una acción, el estudiante debe revisar sus instrucciones, detectar el error y ajustar la solución. Así, la equivocación se convierte en una oportunidad de aprendizaje.

La integración con otras asignaturas también amplía sus posibilidades. Ye et al. (2023) explican que el pensamiento matemático y el computacional pueden fortalecerse mutuamente, ya que la programación permite modelar y comprobar conceptos matemáticos. Bajo esta lógica, el pensamiento computacional debe incorporarse mediante problemas disciplinares y no como un contenido aislado.

Por tanto, las estrategias más pertinentes son aquellas que combinan actividades desconectadas, programación, robótica y resolución de problemas. La tecnología adquiere valor cuando permite investigar, crear y comprobar soluciones, en lugar de utilizarse solamente para reproducir información.

Competencias del siglo XXI y evaluación del aprendizaje

El pensamiento computacional se relaciona con habilidades como creatividad, cooperación, pensamiento crítico y resolución de problemas. Korkmaz et al. (2017) plantean que esta competencia posee una naturaleza multidimensional, por lo que no puede evaluarse únicamente mediante el dominio técnico de la programación. Un estudiante puede utilizar códigos correctamente y, sin embargo, presentar dificultades para explicar sus decisiones o proponer soluciones originales.

En relación con la evaluación, Allsop (2019) recomienda combinar observaciones, conversaciones, productos digitales y reflexiones del alumnado. Este enfoque permite diferenciar a quien comprende el procedimiento de quien se limita a copiar instrucciones. La evaluación debe examinar la planificación, las pruebas realizadas, los errores detectados y las mejoras incorporadas.

La creatividad tampoco surge automáticamente por programar. Israel-Fishelson et al. (2021) encontraron vínculos entre creatividad y pensamiento computacional, aunque sus resultados muestran que dichas relaciones dependen de las características de la actividad y del estudiante. Por esa razón, los ejercicios excesivamente guiados pueden restringir la iniciativa y la búsqueda de alternativas.

A partir de estos planteamientos, las tareas educativas deben permitir varias soluciones posibles. Las rúbricas, los portafolios y los proyectos facilitan la valoración de procesos como



abstracción, depuración, colaboración y transferencia. De esta manera, la evaluación reconoce tanto el resultado como la evolución del razonamiento.

Formación docente y condiciones de implementación

La formación docente es esencial para integrar el pensamiento computacional con una intención pedagógica clara. Yadav et al. (2014) comprobaron que la participación de futuros profesores en módulos especializados modifica sus concepciones y les permite reconocer que esta competencia puede aplicarse en diferentes asignaturas.

Sin embargo, aprender a utilizar una plataforma no garantiza una enseñanza adecuada. El docente necesita identificar las habilidades que pretende desarrollar, diseñar problemas significativos y establecer evidencias de aprendizaje. El propósito no es formar programadores profesionales, sino educadores capaces de orientar procesos de descomposición, abstracción y evaluación.

Dong et al. (2024) señalan que la preparación del profesorado obtiene mejores resultados cuando combina programación, robótica, aprendizaje basado en proyectos, colaboración y diseño de actividades. Este hallazgo demuestra que la formación debe integrar conocimientos tecnológicos, pedagógicos y disciplinares.

Por último, la implementación requiere continuidad, acceso a recursos, tiempo de planificación y apoyo institucional. Las capacitaciones breves pueden despertar interés, pero difícilmente producen cambios permanentes si los docentes no cuentan con acompañamiento y oportunidades para aplicar lo aprendido. En este marco, el desarrollo del pensamiento computacional depende tanto de la estrategia utilizada como de las condiciones educativas que permiten sostenerla.

Síntesis teórica

Los estudios examinados permiten establecer que el pensamiento computacional supera el aprendizaje técnico de la programación. Su desarrollo implica formular problemas, organizar información, construir algoritmos, detectar errores y transferir soluciones a otros contextos. Las actividades desconectadas, la robótica, la programación y la integración interdisciplinaria pueden fortalecer estas habilidades cuando se aplican mediante problemas significativos. A la vez, la evaluación debe considerar los procesos de razonamiento, la creatividad, la colaboración y las modificaciones realizadas durante la actividad.



La formación docente conecta estos fundamentos con la práctica educativa. Un profesorado preparado puede convertir los recursos tecnológicos en medios para investigar, crear y resolver problemas, favoreciendo competencias necesarias para desenvolverse en la sociedad digital.

3. METODOLOGÍA

Diseño de la investigación

El presente estudio se desarrolló mediante una revisión sistemática de literatura científica orientada al análisis de la aplicación del pensamiento computacional en la educación y de las estrategias empleadas para fortalecer las competencias del siglo XXI. Esta metodología permitió identificar, evaluar y sintetizar evidencia relacionada con los fundamentos del pensamiento computacional, las estrategias didácticas utilizadas, los mecanismos de evaluación y la formación docente necesaria para su integración curricular.

La investigación adoptó un enfoque cualitativo de carácter documental y analítico. El proceso metodológico siguió las directrices de la declaración PRISMA 2020, con la finalidad de presentar de manera transparente las etapas de identificación, cribado, elegibilidad e inclusión de los artículos científicos (Page et al., 2021). Además, se consideró la propuesta de Okoli (2015), quien plantea que una revisión sistemática debe partir de un protocolo previamente definido para reducir sesgos y garantizar la trazabilidad de la búsqueda.

Fuentes de información

La búsqueda bibliográfica se realizó en bases de datos y plataformas académicas reconocidas por la calidad y cobertura de sus publicaciones, entre ellas Scopus, Web of Science, ScienceDirect, ACM Digital Library, SpringerLink, ERIC y Google Scholar.

Estas fuentes fueron seleccionadas porque concentran investigaciones sobre educación, ciencias de la computación, programación, robótica educativa, formación docente, evaluación del aprendizaje y competencias digitales.

Estrategia de búsqueda

La estrategia de búsqueda se estructuró mediante términos en español e inglés combinados con los operadores booleanos **AND** y **OR**. Las palabras clave utilizadas fueron: “pensamiento computacional”, “educación”, “competencias del siglo XXI”, “programación educativa”, “robótica educativa”, “actividades desconectadas”, “evaluación del pensamiento



computacional”, “formación docente”, “computational thinking”, “computational thinking education”, “21st century skills”, “programming education”, “educational robotics”, “unplugged activities”, “computational thinking assessment” y “teacher education”.

Las ecuaciones fueron adaptadas a las características de cada base de datos para mejorar la precisión y recuperación de documentos. Esta organización sistemática permitió establecer criterios explícitos de búsqueda, selección y síntesis, tal como recomiendan Xiao y Watson (2019).

Ecuaciones de búsqueda por base de datos

Scopus

TITLE-ABS-KEY ("computational thinking") AND TITLE-ABS-KEY ("education" OR "21st century skills" OR "programming")

Web of Science

TS=("computational thinking") AND TS=("education" OR "programming education" OR "21st century skills")

ScienceDirect

("computational thinking" OR "computational skills") AND ("education" OR "programming" OR "student competencies")

ACM Digital Library

("computational thinking") AND ("K-12 education" OR "teacher education" OR "programming")

SpringerLink

"computational thinking" AND ("education" OR "educational robotics" OR "mathematics education")

ERIC

("computational thinking" OR "programming education") AND ("students" OR "teachers" OR "21st century skills")

Google Scholar

- "computational thinking" AND "education"
- "computational thinking" AND "21st century skills"
- "computational thinking" AND "teacher education"



- "computational thinking" AND "educational robotics"
- "pensamiento computacional" AND "estrategias educativas"

Período de estudio

Se incluyeron investigaciones publicadas entre 2006 y 2024. El año inicial se estableció para incorporar el trabajo fundacional de Wing sobre pensamiento computacional, mientras que el límite final permitió integrar evidencia reciente relacionada con actividades desconectadas, robótica educativa, evaluación, creatividad y formación docente.

La amplitud temporal permitió analizar la evolución del concepto desde sus primeras formulaciones hasta las propuestas contemporáneas para su incorporación en diferentes niveles educativos. También permitió comparar investigaciones conceptuales, estudios empíricos y revisiones sistemáticas desarrolladas durante este periodo.

Criterios de inclusión y exclusión

Criterios de inclusión

- Artículos científicos publicados en revistas indexadas.
- Estudios sometidos a revisión por pares.
- Investigaciones publicadas entre 2006 y 2024.
- Documentos redactados en español o inglés.
- Estudios relacionados directamente con el pensamiento computacional en contextos educativos.
- Investigaciones sobre programación, robótica, actividades desconectadas, evaluación o formación docente.
- Publicaciones con DOI o información bibliográfica verificable.
- Documentos disponibles a texto completo o con información metodológica suficiente.

Criterios de exclusión

- Registros duplicados.
- Tesis de grado o posgrado.
- Libros y capítulos de libros.
- Actas, resúmenes y ponencias de congresos.



- Artículos de opinión y documentos no arbitrados.
- Estudios centrados exclusivamente en aspectos técnicos de informática.
- Investigaciones que mencionaran el pensamiento computacional de manera secundaria.
- Documentos sin relación directa con procesos educativos.
- Publicaciones sin información metodológica suficiente.

Proceso de selección de los estudios

El proceso de selección se desarrolló de acuerdo con las etapas propuestas por PRISMA 2020. Inicialmente, se identificaron 110 registros en las bases de datos académicas consultadas. Después de eliminar 18 documentos duplicados, quedaron 92 publicaciones para la revisión de títulos, resúmenes y palabras clave.

Durante el cribado se excluyeron 58 registros porque no abordaban directamente la aplicación educativa del pensamiento computacional, se concentraban únicamente en aspectos informáticos o no cumplían los criterios establecidos.

A continuación, se evaluaron 34 artículos a texto completo. En esta fase se excluyeron 13 documentos: cinco por no analizar directamente el pensamiento computacional en contextos educativos, tres por presentar información metodológica insuficiente, tres por no corresponder a artículos revisados por pares y dos por carecer de información bibliográfica verificable.

Finalmente, 21 estudios fueron incluidos en la revisión sistemática y conformaron la muestra utilizada para la síntesis cualitativa. No se realizó metaanálisis debido a las diferencias existentes entre los diseños metodológicos, niveles educativos, características de los participantes, estrategias pedagógicas e instrumentos de evaluación.

Distribución de los registros por base de datos

Con el propósito de mantener la trazabilidad del proceso de búsqueda, se registró la cantidad de documentos recuperados en cada fuente de información. En total se identificaron 110 registros.

Tabla 1

Distribución de registros identificados por fuente de información



Fuente de información	Registros identificados
Scopus	22
Web of Science	16
ScienceDirect	18
ACM Digital Library	17
SpringerLink	14
ERIC	11
Google Scholar	12
Total	110

Nota. Elaboración propia a partir del proceso de búsqueda bibliográfica realizado para la revisión sistemática.

Extracción y organización de la información

Los estudios seleccionados se organizaron en una matriz de sistematización que incluyó las siguientes categorías:

- Autor y año de publicación.
- Título del estudio.
- Objetivo de la investigación.
- Diseño metodológico.
- Nivel educativo.
- Participantes.
- Estrategia pedagógica aplicada.
- Componentes del pensamiento computacional.
- Competencias desarrolladas.
- Instrumentos de evaluación.
- Resultados principales.
- Limitaciones del estudio.
- Aportes a la investigación.
- DOI.



La matriz facilitó la comparación entre las investigaciones y permitió reconocer coincidencias, diferencias, tendencias y vacíos en el conocimiento disponible.

Análisis de la información

La información se analizó mediante un proceso de categorización temática. Este procedimiento permitió organizar los resultados y construir una síntesis interpretativa, evitando limitar la revisión a una descripción individual de cada publicación. Snyder (2019) señala que una revisión de literatura rigurosa debe integrar, comparar y analizar críticamente la evidencia para producir una comprensión organizada del campo investigado.

Los artículos se distribuyeron en cuatro categorías principales:

1. Fundamentos conceptuales y cognitivos

Esta categoría reunió estudios relacionados con la definición, delimitación y componentes del pensamiento computacional, como la descomposición, abstracción, identificación de patrones, diseño algorítmico, depuración y generalización.

2. Estrategias didácticas e integración curricular

En este eje se analizaron investigaciones sobre programación visual, robótica educativa, actividades desconectadas, aprendizaje basado en proyectos y aplicación del pensamiento computacional en áreas como Matemáticas y Ciencias.

3. Competencias del siglo XXI y evaluación

Esta categoría comprendió estudios relacionados con creatividad, pensamiento crítico, colaboración, resolución de problemas, competencia digital e instrumentos empleados para evaluar el pensamiento computacional.

4. Formación docente y condiciones de implementación

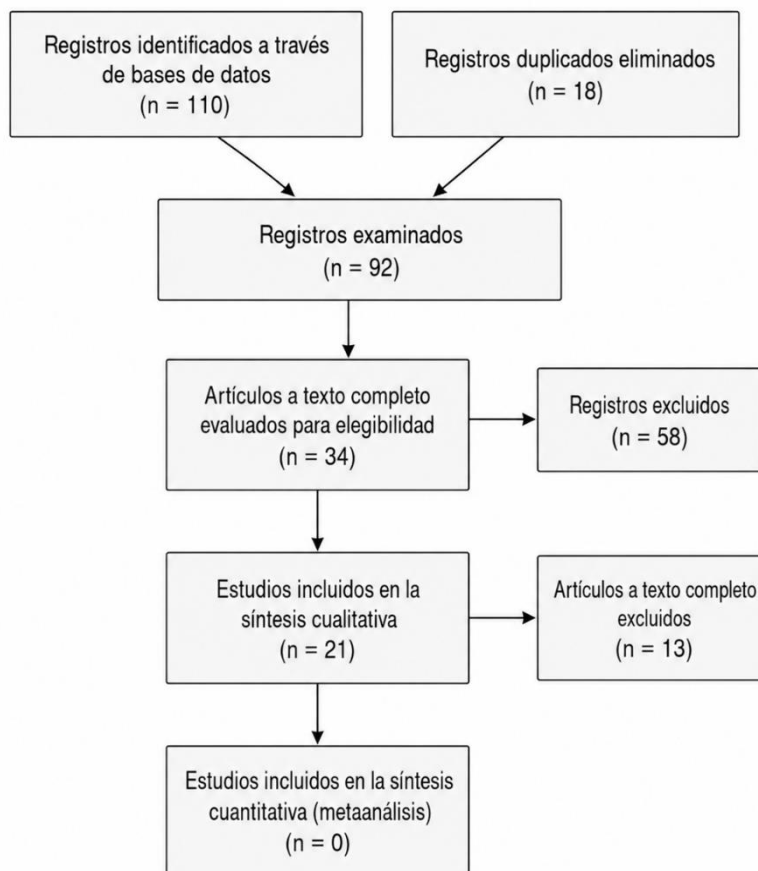
El último eje agrupó investigaciones sobre preparación del profesorado, conocimientos tecnológicos y pedagógicos, percepción docente, disponibilidad de recursos y condiciones institucionales para integrar el pensamiento computacional en el aula.

Posteriormente, se compararon los hallazgos reportados por los diferentes autores para identificar acuerdos, diferencias, aportes y limitaciones. Los resultados se integraron mediante una síntesis narrativa y comparativa, debido a la heterogeneidad de los contextos educativos y metodologías de los estudios seleccionados.

Figura 1

Diagrama de flujo del proceso de selección según PRISMA 2020





Nota. Adaptado de Page et al. (2021). El diagrama representa las etapas de identificación, depuración, cribado, elegibilidad e inclusión. Se identificaron 110 registros, se eliminaron 18 duplicados y se examinaron 92 publicaciones. Después de excluir 58 registros durante el cribado, se evaluaron 34 artículos a texto completo. Finalmente, 21 estudios fueron incluidos en la síntesis cualitativa. Elaboración propia.

4. RESULTADOS

A partir de la revisión sistemática, los 21 estudios seleccionados se organizaron en cuatro categorías temáticas: fundamentos conceptuales y cognitivos del pensamiento computacional; estrategias didácticas e integración curricular; competencias del siglo XXI y evaluación del aprendizaje; y formación docente y condiciones de implementación. Esta clasificación permitió comparar las contribuciones de la literatura científica e identificar los



principales avances, estrategias, instrumentos y desafíos relacionados con la incorporación del pensamiento computacional en el ámbito educativo.

Tabla 2

Fundamentos conceptuales y cognitivos del pensamiento computacional

Autor	Enfoque	Aporte a la investigación
Wing (2006)	Fundamentación conceptual	Plantea el pensamiento computacional como una competencia para resolver problemas, diseñar sistemas y comprender situaciones mediante principios propios de la computación.
Grover y Pea (2013)	Revisión del campo educativo	Delimitan sus componentes principales y reconocen vacíos relacionados con la enseñanza, la evaluación y la integración del pensamiento computacional en la educación escolar.
Denning (2017)	Análisis conceptual	Advierte que el concepto no debe utilizarse para cualquier actividad de resolución de problemas y destaca la necesidad de mantener una definición precisa y evaluable.
Shute et al. (2017)	Revisión teórica	Organizan el pensamiento computacional en procesos como descomposición, abstracción, diseño algorítmico, depuración, iteración y generalización.
Palts y Pedaste (2020)	Modelo de desarrollo	Proponen una secuencia progresiva para formular problemas, abstraer información, diseñar soluciones, evaluarlas y transferirlas a nuevos contextos.

Nota. Principales aportes de los estudios relacionados con la delimitación conceptual, los componentes y los procesos cognitivos del pensamiento computacional. Elaboración propia. Los estudios evidencian que el pensamiento computacional no se limita al uso de computadoras ni al aprendizaje de programación. Su desarrollo implica comprender un problema, seleccionar información relevante, construir procedimientos, detectar errores y comprobar la utilidad de las soluciones planteadas.

Tabla 3



Estrategias didácticas e integración curricular

Autor	Estrategia analizada	Aporte a la investigación
Lye y Koh (2014)	Programación educativa	Identifican que la programación puede fortalecer el razonamiento algorítmico, la resolución de problemas y la depuración cuando se aplica con una orientación pedagógica adecuada.
Chen et al. (2023)	Actividades desconectadas	Demuestran que los juegos, tarjetas, movimientos y materiales físicos pueden favorecer el pensamiento computacional sin depender exclusivamente de dispositivos electrónicos.
Witherspoon et al. (2017)	Robótica virtual	Evidencian que la programación de robots facilita la descomposición de problemas, la creación de instrucciones y la corrección de errores.
Weintrop et al. (2016)	Matemáticas y Ciencias	Proponen prácticas computacionales relacionadas con datos, modelización, simulación, resolución de problemas y pensamiento sistémico para su aplicación interdisciplinaria.
Ye et al. (2023)	Integración con Matemáticas	Determinan que el pensamiento computacional puede apoyar la comprensión matemática mediante programación, representación de problemas y construcción de modelos.

Nota. Estudios incluidos en la revisión sistemática que analizan estrategias como programación, robótica, actividades desconectadas e integración del pensamiento computacional en distintas asignaturas. Elaboración propia.

Los resultados muestran que las estrategias producen mejores oportunidades de aprendizaje cuando los estudiantes diseñan, experimentan y revisan sus propias soluciones. En este sentido, la tecnología debe funcionar como un recurso para razonar y crear, y no únicamente como un medio para recibir información.

Tabla 4



Competencias del siglo XXI y evaluación del aprendizaje

Autor	Proceso analizado	Aporte a la investigación
Kong et al. (2018)	Interés y colaboración	Relacionan la educación en pensamiento computacional con el interés estudiantil, la actitud colaborativa y el empoderamiento para aprender programación.
Nouri et al. (2020)	Competencias digitales	Vinculan el aprendizaje de programación con el desarrollo del pensamiento computacional, la competencia digital, la colaboración y la resolución de problemas.
Israel- Fishelson et al. (2021)	Creatividad	Identifican asociaciones entre pensamiento computacional y creatividad, aunque señalan que estas pueden variar según las características personales y educativas.
Korkmaz et al. (2017)	Medición de habilidades	Presentan una escala que considera creatividad, pensamiento algorítmico, cooperación, pensamiento crítico y resolución de problemas.
Tang et al. (2020)	Evaluación sistemática	Reconocen distintos instrumentos para evaluar el pensamiento computacional y resaltan la necesidad de comprobar su validez y confiabilidad.
Allsop (2019)	Evaluación múltiple	Propone combinar observaciones, conversaciones, productos digitales y reflexiones para valorar tanto el resultado como el proceso seguido por el estudiante.
Román- González et al. (2017)	Habilidades cognitivas	Relacionan el desempeño en pensamiento computacional con capacidades de razonamiento, resolución de problemas y habilidades espaciales.

Nota. Investigaciones que examinan la relación del pensamiento computacional con la creatividad, la colaboración, la competencia digital, el razonamiento y los métodos utilizados para evaluar el aprendizaje. Elaboración propia.



Los hallazgos indican que el pensamiento computacional puede contribuir al desarrollo de competencias necesarias para la sociedad digital. No obstante, estas capacidades no surgen automáticamente por utilizar tecnología; requieren actividades abiertas, acompañamiento pedagógico y evaluaciones que consideren el razonamiento, la creatividad y las decisiones adoptadas.

Tabla 5
Formación docente y condiciones de implementación

Autor	Enfoque	Aporte a la investigación
Yadav et al. (2014)	Formación inicial docente	Evidencian que los módulos formativos pueden ampliar la comprensión del profesorado sobre el pensamiento computacional y sus posibilidades de aplicación en distintas asignaturas.
Dong et al. (2024)	Revisión sobre futuros docentes	Identifican que la preparación docente mejora cuando combina programación, robótica, proyectos, colaboración y diseño de actividades educativas.
Hsu et al. (2018)	Enseñanza y aprendizaje	Sistematizan estrategias para enseñar pensamiento computacional y resaltan la importancia de adaptar los métodos a la edad, el contexto y los objetivos educativos.
Tikva y Tambouris (2021)	Integración en educación escolar	Proponen un modelo conceptual para organizar los componentes del pensamiento computacional desarrollados mediante programación en los niveles educativos K–12.

Nota. Estudios que analizan la preparación del profesorado y las condiciones pedagógicas y curriculares requeridas para incorporar el pensamiento computacional en el ámbito educativo. Elaboración propia.

Las investigaciones coinciden en que la implementación efectiva depende de la formación del profesorado, la planificación curricular y el apoyo institucional. El acceso a dispositivos o programas resulta insuficiente cuando el docente no cuenta con conocimientos para diseñar problemas significativos, acompañar la exploración y evaluar los procesos desarrollados por los estudiantes.



Síntesis de los resultados

En conjunto, los estudios revisados muestran que el pensamiento computacional constituye una competencia transversal vinculada con la formulación de problemas, la abstracción, el razonamiento algorítmico, la creatividad y la evaluación de soluciones. Las estrategias más utilizadas incluyen programación, robótica educativa, actividades desconectadas, proyectos y prácticas integradas en Matemáticas y Ciencias.

De igual manera, los resultados evidencian que su incorporación puede favorecer la competencia digital, el pensamiento crítico, la colaboración y la resolución de problemas. Sin embargo, los beneficios dependen de la calidad de las actividades, de los métodos de evaluación y de la mediación pedagógica.

Finalmente, la formación docente representa una condición esencial para que el pensamiento computacional se integre de forma sostenida en el currículo. La capacitación debe superar el aprendizaje técnico de plataformas y orientar al profesorado hacia el diseño de experiencias contextualizadas, progresivas y centradas en la participación activa del estudiante.

5. DISCUSIÓN

Los resultados obtenidos evidencian que el pensamiento computacional no constituye únicamente una habilidad técnica asociada con la programación, sino una competencia multidimensional que integra procesos cognitivos, estrategias pedagógicas, capacidades digitales y habilidades para la resolución de problemas. Los estudios revisados coinciden en su importancia dentro de la educación; sin embargo, presentan diferencias respecto a su definición, los componentes que lo conforman y las formas utilizadas para promoverlo en el aula. Wing (2006) lo plantea como una manera de formular problemas y diseñar soluciones basadas en principios computacionales, mientras que Grover y Pea (2013) amplían esta perspectiva al incorporar procesos como abstracción, descomposición y automatización. Esta comparación permite reconocer que el concepto ha evolucionado desde una formulación general hacia modelos educativos más estructurados.

En el plano conceptual, Denning (2017) advierte que el pensamiento computacional pierde precisión cuando se utiliza para describir cualquier proceso de resolución de problemas. Su planteamiento complementa el trabajo de Shute et al. (2017), quienes organizan esta competencia mediante dimensiones observables, como descomposición, abstracción, diseño



algorítmico, depuración y generalización. A su vez, Palts y Pedaste (2020) proponen una secuencia progresiva que permite comprender cómo el estudiante avanza desde la formulación del problema hasta la evaluación y transferencia de la solución. Aunque estos autores emplean enfoques distintos, coinciden en que el pensamiento computacional debe demostrarse mediante procesos de razonamiento verificables y no únicamente por el uso de herramientas digitales.

Desde una perspectiva cognitiva, Román-González et al. (2017) relacionan el pensamiento computacional con habilidades de razonamiento, capacidad espacial y resolución de problemas. Este hallazgo sugiere que su desarrollo depende de procesos cognitivos que pueden fortalecerse mediante actividades progresivas. Sin embargo, esta relación no implica que el pensamiento computacional sea una capacidad fija, ya que la práctica educativa y la mediación docente pueden ampliar las posibilidades de aprendizaje. Por esta razón, las propuestas pedagógicas deben considerar la edad, los conocimientos previos y el nivel de complejidad de los problemas planteados.

En cuanto a las estrategias didácticas, Lye y Koh (2014) sostienen que la programación puede favorecer el razonamiento algorítmico, la depuración y la resolución de problemas. No obstante, sus beneficios dependen de que las actividades trasciendan la repetición de códigos y permitan que el estudiante diseñe y evalúe sus propias soluciones. Hsu et al. (2018) coinciden con esta perspectiva al identificar metodologías como el aprendizaje basado en problemas, los proyectos y la programación visual. En ambos casos, la tecnología adquiere valor educativo cuando se utiliza para construir conocimientos, experimentar y tomar decisiones.

Chen et al. (2023) incorporan una perspectiva diferente al demostrar que las actividades desconectadas también pueden favorecer el pensamiento computacional. Juegos de lógica, tarjetas, movimientos corporales y materiales manipulables permiten trabajar secuencias, patrones y algoritmos sin utilizar dispositivos electrónicos. Este aporte resulta relevante para instituciones con acceso tecnológico limitado. Pese a ello, dichas actividades alcanzan mayor profundidad cuando se articulan posteriormente con programación, robótica o simulaciones, ya que el estudiante puede relacionar las acciones concretas con representaciones simbólicas y digitales.



La robótica educativa ofrece otra posibilidad de aprendizaje. Witherspoon et al. (2017) señalan que los entornos de robótica virtual permiten crear instrucciones, comprobar resultados y corregir errores. A diferencia de una actividad basada únicamente en explicaciones teóricas, el comportamiento del robot ofrece una respuesta inmediata sobre la efectividad de la solución. De esta forma, el error se transforma en una oportunidad para analizar, ajustar y mejorar el procedimiento. Sin embargo, el recurso tecnológico no garantiza por sí mismo el desarrollo del pensamiento computacional, pues una tarea excesivamente guiada puede reducir la participación del estudiante.

Respecto a la integración curricular, Weintrop et al. (2016) identifican prácticas computacionales aplicables en Matemáticas y Ciencias, entre ellas el análisis de datos, la modelización, la simulación y el pensamiento sistémico. Ye et al. (2023) profundizan esta relación al señalar que el pensamiento computacional puede mejorar el aprendizaje matemático cuando los estudiantes representan problemas y construyen modelos. Estos hallazgos muestran que su enseñanza no debería limitarse a una asignatura de informática. Por el contrario, puede funcionar como una herramienta transversal para explorar fenómenos, organizar información y comprobar hipótesis en diferentes áreas del conocimiento.

En relación con las competencias del siglo XXI, Nouri et al. (2020) vinculan la programación con el desarrollo de competencia digital, colaboración y resolución de problemas. Kong et al. (2018), por su parte, destacan la influencia del pensamiento computacional sobre el interés, la actitud colaborativa y el empoderamiento para aprender. La diferencia entre ambos estudios se encuentra en el énfasis otorgado a los resultados: el primero se concentra en competencias digitales y cognitivas, mientras que el segundo incorpora dimensiones motivacionales y sociales. En conjunto, estos aportes indican que el pensamiento computacional puede favorecer tanto habilidades intelectuales como disposiciones para participar activamente en el aprendizaje.

La creatividad constituye otra dimensión relevante. Israel-Fishelson et al. (2021) identifican una relación entre creatividad y pensamiento computacional, aunque advierten que esta asociación puede variar según las características personales y educativas del estudiante. Esto significa que programar no genera automáticamente creatividad. Para promoverla, las actividades deben admitir distintas soluciones, permitir la experimentación y ofrecer espacios para modificar o ampliar los productos elaborados.



Los estudios sobre evaluación también presentan aportes complementarios. Korkmaz et al. (2017) plantean una escala que integra creatividad, cooperación, pensamiento algorítmico, pensamiento crítico y resolución de problemas. Tang et al. (2020) analizan diferentes instrumentos y reconocen la ausencia de un procedimiento único para valorar esta competencia. Allsop (2019) propone una evaluación múltiple basada en observaciones, conversaciones, productos y reflexiones. La comparación de estas investigaciones demuestra que una prueba aislada resulta insuficiente para valorar procesos complejos como abstracción, depuración, colaboración y transferencia.

Tikva y Tambouris (2021) aportan un modelo conceptual para organizar las dimensiones del pensamiento computacional desarrolladas mediante programación. Su trabajo permite relacionar actividades, habilidades y resultados de aprendizaje. Pese a ello, la diversidad de modelos e instrumentos dificulta la comparación directa entre investigaciones. Algunos estudios valoran conocimientos de programación, mientras que otros examinan razonamiento, actitudes, creatividad o productos digitales. Esta heterogeneidad limita la posibilidad de establecer conclusiones generales sobre la efectividad de todas las estrategias educativas.

La formación docente representa una condición decisiva para la implementación. Yadav et al. (2014) muestran que la preparación inicial puede modificar las concepciones del profesorado y ampliar su comprensión sobre las posibilidades del pensamiento computacional. Dong et al. (2024) complementan este planteamiento al señalar que la formación resulta más efectiva cuando combina programación, robótica, trabajo colaborativo, proyectos y diseño de actividades. Estos hallazgos sugieren que aprender a utilizar una plataforma es insuficiente si el docente no comprende los procesos cognitivos que debe promover ni los criterios necesarios para evaluarlos.

Otro aspecto crítico es la desigualdad en el acceso a dispositivos, conectividad y recursos educativos. Aunque las actividades desconectadas ofrecen una alternativa, la implementación sostenida requiere apoyo institucional, tiempo de planificación y materiales adaptados a cada contexto. También es necesario evitar que el pensamiento computacional se convierta en una propuesta exclusiva para instituciones con mayores recursos o para estudiantes considerados previamente competentes en tecnología.



Entre los vacíos identificados se encuentra la necesidad de realizar más estudios longitudinales que permitan observar la evolución del pensamiento computacional durante periodos prolongados. Gran parte de las investigaciones analiza intervenciones breves y grupos específicos, lo que dificulta determinar si las habilidades adquiridas se mantienen y transfieren a nuevas situaciones. Asimismo, se requieren instrumentos de evaluación comparables, investigaciones en niveles educativos iniciales y estudios realizados en contextos latinoamericanos.

En conjunto, la evidencia analizada permite afirmar que el pensamiento computacional constituye una competencia relevante para la educación contemporánea. Sus beneficios se relacionan con el razonamiento, la creatividad, la colaboración, la competencia digital y la resolución de problemas. Aun así, estos resultados dependen de la calidad de las actividades, la mediación del docente, los criterios de evaluación y las condiciones institucionales. Su incorporación efectiva requiere una visión pedagógica que supere el aprendizaje técnico de programación y lo vincule con problemas significativos de diferentes áreas del conocimiento.

6. CONCLUSIÓN

La revisión sistemática de los 21 artículos científicos permitió concluir que el pensamiento computacional constituye una competencia educativa multidimensional relacionada con la capacidad de formular problemas, descomponer situaciones complejas, reconocer patrones, abstraer información, diseñar algoritmos y evaluar soluciones. Aunque su desarrollo suele vincularse con la programación, los hallazgos muestran que puede promoverse mediante diferentes estrategias y áreas curriculares.

Respecto a sus fundamentos, los estudios evidencian que el pensamiento computacional no debe reducirse al manejo de computadoras. Su valor se encuentra en los procesos de razonamiento desarrollados por los estudiantes. La programación, la robótica y las actividades desconectadas funcionan como medios para favorecer estas capacidades, siempre que las tareas permitan tomar decisiones, experimentar, detectar errores y proponer mejoras. En cuanto a las estrategias didácticas, la revisión identificó beneficios asociados con la programación visual, la robótica educativa, los proyectos, la resolución de problemas y las actividades sin tecnología. La integración con Matemáticas y Ciencias demuestra que el pensamiento computacional puede aplicarse de manera transversal. No obstante, su



efectividad depende de la orientación pedagógica y del nivel de participación otorgado al estudiante.

Los estudios también muestran aportes al desarrollo de competencias del siglo XXI, como creatividad, pensamiento crítico, colaboración, competencia digital y resolución de problemas. Estas habilidades no aparecen automáticamente por utilizar dispositivos o programas. Se requiere diseñar experiencias abiertas, contextualizadas y progresivas que permitan al estudiante construir, explicar y evaluar sus soluciones.

En materia de evaluación, se concluye que no existe un único instrumento capaz de medir todas las dimensiones del pensamiento computacional. Por esta razón, resulta pertinente combinar pruebas, rúbricas, observaciones, proyectos, productos digitales y reflexiones del alumnado. Esta variedad permite valorar tanto el resultado final como el proceso de razonamiento.

La formación docente constituye una condición esencial para integrar el pensamiento computacional en el currículo. Los profesores necesitan conocimientos tecnológicos, pedagógicos y disciplinares que les permitan seleccionar estrategias, plantear problemas significativos y acompañar el proceso de aprendizaje. Asimismo, las instituciones deben proporcionar recursos, tiempo de planificación y apoyo continuo.

A partir de los hallazgos revisados, se reconoce la necesidad de desarrollar investigaciones longitudinales, validar instrumentos de evaluación y ampliar los estudios en contextos educativos diversos. También es necesario analizar cómo las condiciones socioeconómicas, el acceso tecnológico y las características del alumnado influyen en su implementación.

En conclusión, el pensamiento computacional ofrece oportunidades relevantes para fortalecer la educación y preparar a los estudiantes frente a los desafíos de la sociedad digital. Su integración debe desarrollarse como una competencia transversal orientada a la comprensión y resolución de problemas, evitando limitarla al aprendizaje mecánico de códigos o plataformas tecnológicas.

REFERENCIAS BIBLIOGRÁFICAS

- Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach. *International Journal of Child-Computer Interaction*, 19, 30–55. <https://doi.org/10.1016/j.ijcci.2018.10.004>



- Chen, P., Yang, D., Metwally, A. H. S., Lavonen, J., & Wang, X. (2023). Fostering computational thinking through unplugged activities: A systematic literature review and meta-analysis. *International Journal of STEM Education*, 10, Article 47. <https://doi.org/10.1186/s40594-023-00434-7>
- Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33–39. <https://doi.org/10.1145/2998438>
- Dong, W., Li, Y., Sun, L., & Liu, Y. (2024). Developing pre-service teachers' computational thinking: A systematic literature review. *International Journal of Technology and Design Education*, 34, 191–227. <https://doi.org/10.1007/s10798-023-09811-3>
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43. <https://doi.org/10.3102/0013189X12463051>
- Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296–310. <https://doi.org/10.1016/j.compedu.2018.07.004>
- Israel-Fishelson, R., HersHKovitz, A., Eguíluz, A., Garaizar, P., & Guenaga, M. (2021). The associations between computational thinking and creativity: The role of personal characteristics. *Journal of Educational Computing Research*, 58(8), 1415–1447. <https://doi.org/10.1177/0735633120940954>
- Kong, S.-C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers & Education*, 127, 178–189. <https://doi.org/10.1016/j.compedu.2018.08.026>
- Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales. *Computers in Human Behavior*, 72, 558–569. <https://doi.org/10.1016/j.chb.2017.01.005>
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K–12? *Computers in Human Behavior*, 41, 51–61. <https://doi.org/10.1016/j.chb.2014.09.012>



- Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K–9. *Education Inquiry*, 11(1), 1–17. <https://doi.org/10.1080/20004508.2019.1627844>
- Okoli, C. (2015). A guide to conducting a standalone systematic literature review. *Communications of the Association for Information Systems*, 37, Article 43. <https://doi.org/10.17705/1CAIS.03743>
- 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, Article n71. <https://doi.org/10.1136/bmj.n71>
- Palts, T., & Pedaste, M. (2020). A model for developing computational thinking skills. *Informatics in Education*, 19(1), 113–128. <https://doi.org/10.15388/infedu.2020.06>
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678–691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104, 333–339. <https://doi.org/10.1016/j.jbusres.2019.07.039>
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, Article 103798. <https://doi.org/10.1016/j.compedu.2019.103798>
- Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K–12 education: A conceptual model based on a systematic literature review. *Computers & Education*, 162, Article 104083. <https://doi.org/10.1016/j.compedu.2020.104083>



- Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. <https://doi.org/10.1007/s10956-015-9581-5>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Witherspoon, E. B., Higashi, R. M., Schunn, C. D., Baehr, E. C., & Shoop, R. (2017). Developing computational thinking through a virtual robotics programming curriculum. *ACM Transactions on Computing Education*, 18(1), Article 4, 1–20. <https://doi.org/10.1145/3104982>
- Xiao, Y., & Watson, M. (2019). Guidance on conducting a systematic literature review. *Journal of Planning Education and Research*, 39(1), 93–112. <https://doi.org/10.1177/0739456X17723971>
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, 14(1), Article 5, 1–16. <https://doi.org/10.1145/2576872>
- Ye, H., Liang, B., Ng, O. L., & Chai, C. S. (2023). Integration of computational thinking in K–12 mathematics education: A systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education*, 10, Article 3. <https://doi.org/10.1186/s40594-023-00396-w>

Conflicto de Intereses: Los autores afirman que no existen conflictos de intereses en este estudio y que se han seguido éticamente los procesos establecidos por esta revista. Además, aseguran que este trabajo no ha sido publicado parcial ni totalmente en ninguna otra revista.

FINANCIAMIENTO

Los autores no recibieron financiamiento para el desarrollo de esta investigación.

CONTRIBUCIÓN DE AUTORÍA:

CONTRIBUCIÓN DE AUTORÍA

Nombres de autores e iniciales: Jorge Eduardo Fernández Acevedo (JEFA), José Luis García (JLG), Anabel Lisbeth Naranjo Brito (ALNB), Karolayn Alejandra Sánchez Ronquillo (KASR), Félix Javier Gómez Freire (FJGF).



1. Conceptualización: (JEFA) (JLG) (ALNB)
2. Curación de datos: (ALNB) (KASR) (FJGF)
3. Análisis formal: (JEFA) (JLG) (KASR)
4. Adquisición de fondos: No aplica
5. Investigación: (JEFA) (JLG) (ALNB) (KASR) (FJGF)
6. Metodología: (JEFA) (JLG) (ALNB)
7. Administración del proyecto: (JEFA)
8. Recursos: (JLG) (ALNB) (FJGF)
9. Software: (ALNB) (KASR)
10. Supervisión: (JEFA) (JLG)
11. Validación: (JEFA) (JLG) (FJGF)
12. Visualización: (KASR) (FJGF)
13. Redacción – Borrador original: (JEFA) (ALNB) (KASR)
14. Redacción – Revisión y edición: (JEFA) (JLG) (FJGF)

