CC BY NC ND

DESARROLLO DEL PENSAMIENTO COMPUTACIONAL: UNA ESTRUCTURA LÓGICA QUE SOBREPASA LA PROGRAMACIÓN

Luz Yarney Caicedo Hernández[1]
yarneycaicedo@gmail.com
https://orcid.org/0009-0005-5809-1712
Universidad Pedagógica Experimental Libertador

Recibido: 27 de septiembre 2025 **Aprobado:** 01 de diciembre 2025

RESUMEN

En este artículo de revisión teórica se analizó, desde una perspectiva crítica, el desarrollo del pensamiento computacional como una estructura lógica y cognitiva que sobrepasa la programación, con el propósito de mostrar que constituye un marco más amplio y robusto para potenciar el razonamiento y la argumentación matemática. Apoyado en aportes fundacionales de Papert (1980) y Wing (2009), así como en contribuciones recientes de autores como Li et al. (2020), Polanco et al. (2021), Hernández et al. (2023), Jiménez y Carmona (2023) y Ocampo et al. (2024); el trabajo se caracterizó por una revisión documental sistemática en bases de datos académicas, con criterios definidos de búsqueda, selección y categorización de la literatura. Los hallazgos permiten afirmar que, aunque la programación opera como un medio privilegiado para desplegar habilidades del pensamiento computacional, este se configura como un constructo multidimensional y multidisciplinar que integra procesos de descomposición, reconocimiento de patrones, abstracción, pensamiento algorítmico, codificación y automatización, aplicables con y sin uso de tecnologías digitales, y estrechamente afines a la lógica del pensamiento matemático. Asimismo, se evidencian vacíos importantes: escasez de estudios longitudinales, integración curricular aún incipiente (especialmente en Latinoamérica) y necesidad de marcos conceptuales más unificados. En conclusión, este artículo sostiene que el Pensamiento Computacional mantiene una relación estrecha pero no equivalente con la programación, se erige como una forma de organización estructurada del pensamiento transferible a múltiples disciplinas y exige avanzar en políticas, diseños curriculares y formación docente que lo incorporen de manera sistemática para transformar la enseñanza y el aprendizaje de las matemáticas en coherencia con las demandas del siglo XXI y los marcos internacionales como el ODS 4 y el Plan Nacional Decenal de Educación.

Palabras clave: Pensamiento Computacional, Programación, Lógica.

[1]Profesional vinculada a la Universidad Tecnológica del Chocó "Diego Luis Córdoba" (UTCH), con trayectoria en el ámbito académico y administrativo en Quibdó, Colombia. Universidad de Adscripción: Universidad Pedagógica Experimental Libertador, Venezuela.

DEVELOPMENT OF COMPUTATIONAL THINKING: A LOGICAL STRUCTURE THAT GOES BEYOND PROGRAMMING

ABSTRACT

In this theoretical review article, the development of computational thinking was analyzed from a critical perspective as a logical and cognitive structure that goes beyond programming, with the aim of showing that it constitutes a broader and more robust framework for strengthening mathematical reasoning and argumentation. Drawing on foundational contributions by Papert (1980) and Wing (2009), as well as recent work by authors such as Li et al. (2020), Polanco et al. (2021), Hernández et al. (2023), Jiménez and Carmona (2023), and Ocampo et al. (2024), the study was characterized by a systematic documentary review in academic databases, using defined criteria for the search, selection, and categorization of the literature. The findings indicate that, although programming operates as a privileged means for deploying computational thinking skills, computational thinking itself emerges as a multidimensional and multidisciplinary construct that integrates processes of decomposition, pattern recognition, abstraction, algorithmic thinking, coding, and automation, applicable with and without digital technologies and closely aligned with the logic of mathematical thinking. Likewise, important gaps are identified: a scarcity of longitudinal studies, still incipient curricular integration (especially in Latin America), and the need for more unified conceptual frameworks. In conclusion, this article argues that Computational Thinking maintains a close but non-equivalent relationship with programming, constitutes a form of structured organization of thought that is transferable to multiple disciplines, and calls for further progress in policies, curriculum design, and teacher education that incorporate it systematically in order to transform the teaching and learning of mathematics in line with the demands of the twenty-first century and international frameworks such as SDG 4 and the National Ten-Year Education Plan.

Keywords: Computational Thinking, Programming, Logic.

Introducción

En el panorama educativo contemporáneo, de manera general, la educación enfrenta nuevos desafíos de acuerdo con lo propuesto en la agenda 2030, específicamente en el Objetivo de Desarrollo Sostenible 4 (ODS4), el cual resalta la necesidad de garantizar una educación inclusiva, equitativa y de calidad, promoviendo oportunidades de aprendizaje a lo largo de la vida.

Dentro de este marco, la alfabetización digital y el desarrollo del pensamiento crítico se reconocen como competencias esenciales para enfrentar los desafíos de la sociedad contemporánea. Por ello, es un reto responder a las transformaciones derivadas de esta digitalización, la expansión de la inteligencia artificial y la emergencia de nuevas formas de interacción cognitiva; por ello la enseñanza de la matemática no debe ser ajena a esta realidad.

En sintonía con este planteamiento, el Plan Nacional Decenal de Educación (2016–2026) en Colombia y los Estándares Básicos de Competencias del Ministerio de Educación Nacional (MEN), acentúan la necesidad de formar estudiantes capaces de razonar y argumentar, más allá de memorizar procedimientos. Debido a que durante décadas la enseñanza de la matemática en muchos contextos escolares ha estado marcada por la repetición mecánica de algoritmos y la resolución de ejercicios rutinarios. Sin embargo, los cambios sociales y tecnológicos exigen un aprendizaje centrado en el desarrollo de competencias cognitivas superiores, entre ellas la capacidad de abstraer, analizar, argumentar y resolver problemas en contextos diversos.

En esta dinámica, el pensamiento computacional emerge como un enfoque fundamental que trasciende el ámbito de la programación y los asuntos informáticos (el cuál fue su propósito inicial), para ser comprendido como una estructura cognitiva, aplicable a la resolución de problemas de cualquier área del conocimiento (Wing, 2009). Y, en consecuencia, su aplicabilidad es capaz de fortalecer el razonamiento lógico y la argumentación matemática, porque su estructura presenta como un modelo que articula la lógica de los procesos computacionales con los fundamentos de la matemática escolar, ofreciendo un puente entre ambos campos.

Así mismo, autores de la comunidad académica, como Papert (1980) y Wing (2006, 2009), destacan la necesidad de establecer el Pensamiento Computacional como una habilidad transversal que debe formar parte de la alfabetización básica. Papert además, puntualiza en el papel preponderante de

los materiales manipulativos y la creatividad en la construcción del pensamiento matemático, mientras que Wing amplió esta perspectiva; al considerarlo una habilidad indispensable para todos los ciudadanos del siglo XXI, equiparable a leer, escribir y calcular. Estas visiones han abierto un debate amplio, que en la actualidad se refleja en investigaciones, revisiones sistemáticas y experiencias empíricas desarrolladas en distintos niveles educativos y en diversos contextos geográficos.

Es así como el presente artículo, se propone analizar críticamente el desarrollo del Pensamiento Computacional como una estructura lógica que trasciende la programación, para comprender su incidencia en el razonamiento y la argumentación matemática. Para ello se aborda desde cuatro apartados principales: origen y la conceptualización del Pensamiento Computacional; la identificación de sus dimensiones y su relación con la matemática escolar; la revisión de evidencias empíricas recientes en contextos internacionales y latinoamericanos; y finalmente, una reflexión sobre sus implicaciones curriculares, investigativas y pedagógicas.

Marco Teórico

En este apartado se desarrolla el marco teórico sobre el pensamiento computacional, entendido como una estructura lógica y cognitiva que va más allá de la programación. Se recuperan sus principales definiciones y aportes iniciales, desde Papert y Wing hasta aproximaciones recientes que lo sitúan como competencia clave del siglo XXI. Asimismo, se cuestiona la visión reduccionista que lo limita a la codificación y se argumenta su carácter multidimensional y multidisciplinar, integrando procesos propios del desarrollo del Pensamiento Computacional. Finalmente, se vinculan estos componentes con el razonamiento y la argumentación matemática, para mostrar cómo el pensamiento computacional ofrece un referente sólido para reconfigurar la enseñanza y el aprendizaje de la matemática.

Orígenes y evolución del pensamiento computacional

Aunque el término Pensamiento Computacional aparece formalmente en décadas recientes, muchas de sus prácticas subyacentes como el razonamiento algorítmico, la abstracción, la descomposición de problemas, la representación simbólica y lógica; tienen una larga tradición en matemáticas, lógica, ciencias y filosofía. Estas operaciones mentales han acompañado a la humanidad desde que se desarrollaron métodos sistemáticos para resolver problemas, modelar fenómenos y formalizar razonamientos; es decir, mucho antes de la computación como disciplina técnica.

En este sentido, el campo del pensamiento computacional encuentra sus raíces en la evolución histórica de las ciencias de la computación y su articulación con la educación, y la resolución de problemas. En la década de 1950, hitos fundacionales como la conferencia de Dartmouth Workshop, liderada por investigadores pioneros como Alan Perlis y John McCarthy, introdujeron la idea de sistematizar procesos para que agentes humanos o máquinas procesaran información mediante algoritmos. Posteriormente, entre los años 60 y 70, aportes como el desarrollo de conceptos de alfabetización algorítmica en el trabajo de Donald Knuth, particularmente desde su obra The Art of Computer Programming, consolidaron el razonamiento basado en procedimientos formales como un modo legítimo de construcción de conocimiento.

Estos antecedentes sentaron las bases epistémicas de un enfoque que trasciende la mera programación, investigando cómo las habilidades de descomposición, abstracción y estructuración lógica de los problemas permiten a los sujetos construir soluciones generales, transferibles y susceptibles de automatización; elementos que hoy caracterizan los orígenes disciplinares del pensamiento computacional en la academia. Es así como desde la perspectiva educativa, el origen del pensamiento computacional como constructo didáctico, emerge de la influencia del construccionismo propuesto por Seymour Papert quien, en 1980 desarrolló el lenguaje visual interactivo Logo, cuyo impacto fue ampliamente sistematizado en su obra clásica Mindstorms.

En 2006, el pensamiento computacional alcanzó una consolidación conceptual decisiva gracias al artículo pionero de Jeannette Wing, divulgado en la revista insignia de la Communications of the ACM. En este trabajo, su autora lo define como un proceso cognitivo que permite formular problemas y estructurar sus soluciones de modo que puedan ser interpretadas y ejecutadas por un agente que procesa información, ya sea un ser humano o un sistema computacional, lo que otorga al concepto un carácter transversal y aplicable tanto en contextos digitales como no digitales.

Posteriormente, trabajos auspiciados por organizaciones como Carnegie Mellon University y divulgados a través de plataformas académicas abiertas, permitieron la expansión del constructo hacia áreas no informáticas como las matemáticas, fortaleciendo su inserción curricular en educación media y superior. Más recientemente, trabajos de síntesis como el de Shute et al., (2017) evidencian la expansión y diversificación de definiciones y modelos de Pensamiento computacional en distintos niveles educativos, al tiempo que subrayan el reto de conceptualizar y evaluar de manera coherente con su origen como forma particular de pensamiento para resolver problemas de manera eficiente, con o sin la ayuda directa de un computador. En suma, el pensamiento computacional se sitúa en el centro de las discusiones sobre alfabetización digital, STEM y, en particular, sobre el desarrollo del razonamiento y la resolución de problemas en contextos educativos.

Pensamiento Computacional

La definición de pensamiento computacional tiene sus raíces en los trabajos de Papert (1980), aunque no usa la expresión “pensamiento computacional” lo concibe el como una forma de razonamiento procedimental y reflexivo que se desarrolla cuando los sujetos construyen y manipulan artefactos intelectuales mediante lenguajes computacionales, proceso en el que el computador no es solo una herramienta operativa, sino un medio para pensar sobre el propio pensamiento y modelar problemas de manera abstracta, lógica y algorítmica.

Para Papert aprender con el computador implica construir conocimiento a través de la creación, depuración y socialización de programas, simulaciones o modelos. Dicho de otra forma, el pensamiento computacional es un proceso cognitivo en el que la programación y la construcción de artefactos funcionan como ventanas para el razonamiento formal, la argumentación y la autoconciencia metacognitiva, permitiendo que los estudiantes descompongan problemas, generalicen, formulen conjeturas y validen procedimientos dentro de un ciclo iterativo de diseño y depuración.

Décadas más tarde, Jeannette Wing consolidó el pensamiento computacional en la comunidad académica a través de la revista insignia de la ACM y, particularmente, mediante su artículo seminal en Communications of the ACM (2006), donde lo plantea como una competencia cognitiva básica para formular y resolver problemas complejos desde la abstracción y la automatización. Wing lo define como un proceso mental que permite formular problemas y estructurar sus soluciones de forma que puedan ser interpretadas y ejecutadas por un agente que procesa información, sea humano o máquina, subrayando así su carácter transversal y no restringido a la programación ni a la informática. Esta visión propició que instituciones como la Carnegie Mellon University y organismos internacionales como la UNESCO lo reconocieran como una competencia esencial para el siglo XXI, impulsando su adopción en la educación escolar y su proyección interdisciplinar como marco cognitivo para la resolución de problemas en cualquier disciplina.

Desde entonces, el concepto se ha diversificado y enriquecido en la literatura académica; por ejemplo, la The Royal Society (2012) lo describe como un proceso de pensamiento orientado a formular problemas y estructurar sus soluciones de modo que puedan representarse y ser ejecutadas por un agente que procesa información, idea originalmente difundida en debates científicos publicados tras el reporte Shut down or restart. Bajo esta perspectiva, el pensamiento computacional comenzó a incorporarse en marcos curriculares de diversos países y, paralelamente, fue ganando solidez como

campo investigativo impulsado desde iniciativas como el Computing At School, lo que ha favorecido su consolidación y expansión como área de investigación interdisciplinar en continuo crecimiento.

En América Latina, el pensamiento computacional se ha posicionado como un constructo de creciente interés en la última década, debido a su potencial para dinamizar el desarrollo de competencias digitales y fortalecer procesos cognitivos asociados al razonamiento lógico y argumentativo en la educación básica y media. No obstante, como señalan Quiroz, et al. (2021), su integración en los sistemas educativos de la región sigue siendo desigual, con brechas significativas entre países e instituciones. Estos autores advierten que, en múltiples escenarios escolares, el pensamiento computacional se implementa mediante proyectos aislados de robótica o programación, frecuentemente extracurriculares y fragmentados, sin lograr una articulación profunda ni sostenida dentro del currículo formal.

Esta falta de incorporación curricular limita su consolidación como competencia transversal e interdisciplinar, reduciendo su impacto formativo, especialmente en disciplinas que demandan razonamiento formal, como las matemáticas. Así, aunque su valor para la resolución de problemas complejos es ampliamente reconocido, permanece vigente el desafío de asegurar su inclusión educativa con rigor pedagógico, continuidad didáctica y sostenibilidad curricular, a fin de potenciar su alcance cognitivo en la formación escolar latinoamericana.

Fases del Pensamiento Computacional

El pensamiento computacional constituye una metodología de resolución de problemas que trasciende el ámbito técnico para convertirse en una competencia cognitiva de naturaleza sistémica. De acuerdo con Wing (2006), este enfoque permite abordar desafíos complejos mediante un conjunto de procesos estructurados que organizan el razonamiento de manera lógica. A continuación, se describen las etapas fundamentales que integran este

paradigma, las cuales permiten transformar una dificultad global en una solución ejecutable y eficiente.

En primera instancia, la descomposición se presenta como la habilidad esencial de fragmentar un problema complejo en partes más pequeñas, manejables y comprensibles que pueden abordarse de forma independiente. Según Zapata (2019), este paso inicial facilita la comprensión de la estructura interna del desafío, permitiendo organizar la respuesta de manera segmentada. Al dividir la carga cognitiva, el individuo logra diseñar estrategias específicas para cada componente sin sentirse abrumado por la magnitud del problema original.

Posteriormente, el reconocimiento de patrones permite identificar regularidades, tendencias y similitudes dentro de los datos o procesos analizados. Esta capacidad cognitiva es crucial para inferir la lógica subyacente que organiza un fenómeno, permitiendo detectar qué elementos son invariantes y recurrentes. Para García (2018), gracias a este análisis, es posible generar conjeturas fundamentadas y orientar la modelación de soluciones que puedan ser generalizables, aumentando la eficiencia al no tener que resolver cada detalle como si fuera único.

Aunado a lo anterior, la abstracción actúa como un filtro intelectual que permite identificar los elementos esenciales de una situación mientras se omiten los detalles irrelevantes. Mediante este proceso, se construye una representación simplificada y universal del problema, condensando reglas y relaciones en conceptos o modelos abstractos. Como señalan Adell et al. (2019), la importancia de esta fase radica en que facilita la formulación de soluciones transferibles y verificables que no dependen de un contexto particular, permitiendo razonar sobre el núcleo del problema.

Seguidamente, el pensamiento algorítmico se encarga de diseñar y estructurar secuencias lógicas y finitas de pasos para resolver la situación planteada de forma precisa y replicable. Esta habilidad implica planificar el flujo

operativo bajo el esquema de entrada, proceso y salida, integrando decisiones condicionales e iteraciones. El objetivo primordial es construir una ruta de acción ordenada que pueda ser evaluada y, eventualmente, automatizada (Bocconi et al., 2022).

Por otra parte, la codificación consiste en traducir el modelo abstracto o la secuencia algorítmica a un lenguaje formal, ya sea simbólico o computacional, para que sea interpretado por un sistema. Este paso requiere organizar las reglas bajo una sintaxis específica que elimine ambigüedades, asegurando funcionalidad y coherencia (Hwang et al., 2023). De esta manera, el razonamiento diseñado se materializa en una herramienta técnica que permite la validación práctica de la solución ideada.

Finalmente, la automatización se consolida como la fase donde se transforma el conjunto de pasos organizados en un proceso capaz de ejecutarse repetidamente con mínima intervención humana. Según Shute et al. (2017), esta habilidad permite escalar la solución, garantizando que los resultados sean consistentes y fiables. Al optimizar los tiempos de ejecución y asegurar la fiabilidad operativa, la automatización cierra el ciclo del pensamiento computacional, convirtiendo una idea abstracta en una respuesta tecnológica de alto impacto.

Multidimensionalidad y multidisciplinariedad del pensamiento computacional

El pensamiento computacional no debe entenderse como un constructo unidimensional, pues la literatura converge en reconocerlo como un marco cognitivo complejo, dinámico y multidimensional; está compuesto por dimensiones conceptuales, procedimentales y actitudinales que se entrelazan en la práctica e integra habilidades interdependientes para comprender, formular y resolver problemas en diversos contextos disciplinares (Polanco, et al., 2021). De esta manera, se consolida como una competencia cognitiva de alto orden, cuya riqueza conceptual radica precisamente en la articulación de

diversos procesos mentales que operan de forma integrada para la resolución de problemas complejos, permitiendo su transferencia interdisciplinar y fortaleciendo la construcción de razonamientos estructurados en diferentes campos del conocimiento.

Además, en la actualidad este se entiende como una competencia de carácter claramente multidisciplinar, ya que, aunque surge de la informática, se proyecta como un modo de resolver problemas que atraviesan las matemáticas, las ciencias naturales, las ciencias sociales y las humanidades. Informes de política educativa como el de la Royal Society sobre educación en computación y las revisiones de Voogt y colaboradores muestran que el pensamiento computacional permite reconocer rasgos de la computación en fenómenos naturales y sociales, y aplicar herramientas conceptuales de la ciencia de la computación para comprender y modelar sistemas muy diversos. Así, el pensamiento computacional deja de ser un contenido aislado y se configura como un lenguaje común que conecta saberes disciplinares y favorece formas de razonamiento y argumentación transferibles a problemas complejos de distintas disciplinas.

En el ámbito iberoamericano, Polanco et al., (2021) sostienen que el pensamiento computacional debe asumirse como una competencia compleja que articula componentes lógicos, matemáticos, tecnológicos y didácticos, superando visiones reducidas a la programación. Esta visión multidimensional es reforzada por estudios que lo conceptualizan como competencia del siglo XXI, enfatizando la integración de conocimientos, habilidades y actitudes. Entendido así, el pensamiento computacional ofrece un marco robusto para analizar cómo, en la resolución de problemas, los estudiantes movilizan abstracción, reconocimiento de patrones, descomposición y codificación, y cómo estas operaciones sostienen formas más rigurosas de pensamiento lógico, razonamiento y argumentación.

La lógica propia del Pensamiento Computacional

La relación entre la lógica y el desarrollo del pensamiento computacional es fundamental; la lógica aparece como el andamiaje interno que hace posible que el Pensamiento Computacional se exprese como un modo ordenado, coherente y justificable de razonar. En las revisiones teóricas y empíricas se insiste en que los procesos nucleares de este pensamiento (descomposición, reconocimiento de patrones, abstracción y diseño de algoritmos), exigen establecer relaciones claras entre condiciones y acciones, anticipar consecuencias y garantizar la no ambigüedad de los pasos; es decir, obligan a estructurar el razonamiento en términos de secuencias, condicionales, bucles y relaciones de causa–efecto, todos ellos propios del pensamiento lógico-formal (Li et al., 2020).

En el campo de la educación matemática, varios trabajos sostienen que el Pensamiento Computacional funciona como soporte lógico del razonamiento matemático; al pedir al estudiante que modele una situación en términos de variables, reglas y procedimientos, se promueve la formulación de conjeturas, la generalización de patrones y la sistematización de métodos de resolución, incluso sin escribir código. Así, el diseño de algoritmos y la comparación de soluciones se convierten en espacios privilegiados para ejercitar la coherencia interna del razonamiento, la consistencia de los pasos y la justificación de por qué una estrategia es válida o más eficiente que otra (Argoti, 2024).

Programación: origen y definiciones

La programación, en sentido estricto, designa hoy una práctica técnico-intelectual que se apoya en una larga evolución histórica y semántica. Desde el punto de vista etimológico, el término programa proviene del griego programma que traduce; aviso escrito, plan público. En latín tardío la programación pasa a significar proclamación y, ya en el siglo XX, se especializa en informática como una secuencia de instrucciones codificadas que dirigen la operación de un computador. Sobre esta base, diversos autores y organismos coinciden en definir la programación de computadores como un proceso sistemático de diseño, construcción, prueba y mantenimiento de programas

ejecutables que especifican, mediante algoritmos, la secuencia de acciones que un sistema computacional debe realizar para resolver problemas o ejecutar tareas concretas.

Históricamente, el origen de la programación se vincula, de un lado, con dispositivos mecánicos “programables” como el telar de Jacquard, que empleaban tarjetas perforadas para controlar patrones de tejido, y con los diseños del Motor Analítico de Charles Babbage en el siglo XIX; en ese contexto, las notas de Ada Lovelace sobre dicho motor, donde especifica un procedimiento detallado para calcular números de Bernoulli, son ampliamente reconocidas como el primer programa de computador publicado y la consolidan como una de las pioneras de la disciplina. Con el advenimiento de los computadores electrónicos en el siglo XX, la programación se formaliza primero en lenguaje máquina y ensamblador, y luego mediante lenguajes de alto nivel como FORTRAN, desarrollado en IBM bajo el liderazgo de John Backus en la década de 1950 y considerado el primer lenguaje de alto nivel de uso masivo para cálculo científico, seguido por COBOL y otros lenguajes orientados a dominios específicos.

De otro lado, el glosario de la Network of the National Library of Medicine describe la programación como el desarrollo de instrucciones o código que el computador ejecuta para llevar a cabo tareas o resolver problemas, incluyendo actividades de prueba, depuración y actualización del software. De forma convergente, recursos de síntesis como Google Arts y Culture y Science Direct, señalan que programar implica diseñar y construir programas ejecutables, a partir del análisis del problema, la generación de algoritmos, la estimación de su eficiencia y su implementación en un lenguaje de programación, además de las etapas de verificación, depuración y documentación.

Convergencias y divergencias entre pensamiento computacional y programación

El pensamiento computacional y la programación aparecen estrechamente vinculados, en la medida en que programar se reconoce como uno de los medios privilegiados para favorecer el desarrollo de habilidades propias del Pensamiento Computacional. Adicionalmente, los estudios que integran entornos de programación visual, robótica educativa o plataformas como Arduino, Scratch y App Inventor; muestran que estas herramientas se utilizan para promover procesos como el pensamiento algorítmico, la descomposición de problemas y la depuración, mientras el alumnado aprende a escribir y ejecutar código. Por lo anterior estos procesos coinciden en identificar un núcleo común de habilidades asociadas al Pensamiento Computacional y ponen de relieve que, en la práctica curricular, la programación suele operar como el vehículo preferente para hacer visibles y trabajables estos procesos en distintos niveles y áreas educativas (Ocampo et al., 2024).

Al mismo tiempo, la literatura señala que se trata de constructos diferenciados, en tanto que, la programación alude, fundamentalmente, al dominio de lenguajes y entornos específicos para escribir código ejecutable, mientras que el Pensamiento Computacional se configura como una forma de organización lógica del pensamiento orientada a la comprensión y resolución sistemática de problemas. Desde esta perspectiva, aprender a programar no garantiza, por sí solo, el desarrollo de Pensamiento Computacional: es posible escribir código de manera mecánica, replicando estructuras sintácticas sin comprender los procesos de descomposición, abstracción, diseño y evaluación que subyacen a una solución.

Metodología

En este artículo se adopta una metodología de revisión teórica rigurosa, necesaria para fundamentar con solidez la tesis de que el desarrollo del pensamiento computacional constituye un marco más robusto y amplio que la programación evitando lecturas reduccionistas centradas únicamente en el

código. Desde esta perspectiva, la metodología de revisión no se limita a compilar autores, sino que organiza, compara y triangula categorías, dimensiones y evidencias empíricas, para mostrar de manera argumentada cómo el pensamiento computacional integra procesos de abstracción, descomposición, reconocimiento de patrones y diseño algorítmico que trascienden la enseñanza de lenguajes de programación y se proyectan como un referente más amplio para el desarrollo del razonamiento y la argumentación en contextos educativos.

En el desarrollo de la arquitectura documental, la identificación de las fuentes se gestionó mediante consultas exhaustivas en repositorios de alto impacto científico tales como Scopus, Redalyc, Dialnet, Google Scholar y Scielo. Con el propósito de equilibrar la actualidad con la solidez teórica, se priorizaron hallazgos publicados en el rango temporal de 2000 a 2025, incorporando simultáneamente obras fundamentales de autores como Papert (1980) y Wing (2006, 2009), cuya relevancia es esencial para comprender la base del paradigma. Esta exploración se ejecutó empleando descriptores bilingües de gran precisión, entre los que destacan "pensamiento computacional" (computational thinking), "razonamiento matemático", "argumentación matemática" y el enfoque "STEM", garantizando así una captura de datos coherente con los objetivos de la investigación.

Bajo esta línea de acción, la discriminación de los materiales que integraron la revisión teórica se rigió por criterios de inclusión rigurosos que garantizaron la calidad del corpus. Se admitieron exclusivamente publicaciones en revistas arbitradas o libros académicos que presentaran estudios empíricos sobre el pensamiento computacional en el aula, así como revisiones de literatura y lineamientos curriculares oficiales. Aunado a ello, se consideraron investigaciones de origen latinoamericano e internacional con vinculación directa hacia la enseñanza de las matemáticas y el desarrollo cognitivo, descartando de manera tajante cualquier documento de carácter divulgativo o carente de respaldo académico verificado.

Para consolidar el análisis de la información recolectada, la estructura del estudio se sistematizó a través de seis categorías analíticas que permitieron diseccionar el fenómeno desde múltiples ángulos. Este ordenamiento contempló desde los orígenes y conceptualizaciones tanto del pensamiento computacional como de la programación, hasta sus fases y principios estructurales, sin dejar de lado la multidimensionalidad y la lógica intrínseca que define a este paradigma. Finalmente, mediante el contraste de las convergencias y divergencias entre ambos conceptos, se logró identificar tensiones y vacíos en la literatura especializada, ofreciendo así un panorama integrador y crítico sobre el estado actual del conocimiento en esta área del saber.

A pesar del rigor metodológico aplicado, es imperativo reconocer que el alcance del presente artículo se ve supeditado a ciertas restricciones intrínsecas de la literatura actual. Fundamentalmente, se observa que la mayor parte de la evidencia empírica proviene de entornos con alta disponibilidad tecnológica, lo cual restringe la posibilidad de extrapolar estos resultados a realidades educativas con infraestructura precaria. Asimismo, predomina el uso de muestras reducidas y diseños cuasi-experimentales de breve duración, factores que merman la capacidad de generalización de los hallazgos. Por otra parte, se detecta una notable carencia de estudios longitudinales que verifiquen la transferencia de habilidades hacia áreas como las matemáticas a largo plazo; esta situación se acentúa en el contexto latinoamericano, donde, pese a los progresos recientes, la integración curricular sistemática del pensamiento computacional todavía se encuentra en una etapa incipiente.

Resultados y Discusión

Dentro de la fase de hallazgos y contrastes, se ha logrado establecer que el pensamiento computacional constituye una estructura cognitiva de alta complejidad, cuya naturaleza multidimensional sobrepasa la ejecución técnica de la programación. Esta configuración intelectual se entrelaza de forma orgánica con el razonamiento matemático a través de dimensiones

fundamentales como la abstracción, la descomposición, la secuenciación y la automatización. Tales facultades no solo potencian la lógica y la creatividad, sino que blindan al estudiante con una capacidad superior para la resolución de problemas, consolidándose como un pilar del desarrollo cognitivo moderno.

Desde una perspectiva analítica, resalta que la afinidad entre el pensamiento computacional y la argumentación matemática no es meramente instrumental, sino esencialmente estructural. Ambos procesos convergen en el uso de una lógica rigurosa y una organización secuencial que permite fundamentar soluciones de manera coherente. En consecuencia, el pensamiento computacional actúa como un catalizador pedagógico, impulsando a los alumnos a trascender la resolución mecánica de ejercicios para alcanzar un pensamiento matemático crítico y reflexivo, donde la justificación y la argumentación se vuelven protagonistas del aprendizaje.

No obstante, la revisión teórica pone de manifiesto una serie de vacíos y retos que la comunidad científica aún debe solventar para garantizar la sostenibilidad de estas prácticas. Se percibe una preocupante escasez de estudios longitudinales que evalúen el impacto real de este paradigma en la formación matemática a largo plazo, sumado a una producción investigativa todavía limitada en las etapas de educación inicial y primaria. Por tal motivo, resulta imperativo transitar hacia la construcción de un marco conceptual unificado que brinde claridad a los currículos y políticas públicas, orientando de forma efectiva la vinculación entre la lógica computacional y la enseñanza de las ciencias exactas.

Bajo este escenario, las implicaciones pedagógicas sugieren una ruptura definitiva con la visión reduccionista que confina el pensamiento computacional al uso exclusivo de dispositivos electrónicos. La integración efectiva en las matemáticas demanda la adopción de enfoques transversales que combinen estrategias desconectadas (unplugged) con proyectos interdisciplinarios y metodologías activas. Para que esta transformación sea viable, es vital robustecer la formación inicial y continua del profesorado, dotándolos de las

bases teóricas y las herramientas metodológicas necesarias para que puedan orquestar estas experiencias dentro del aula de manera autónoma.

Finalmente, la promoción de este modelo se encuentra plenamente alineada con los marcos internacionales y las políticas nacionales, como el Objetivo de Desarrollo Sostenible 4 (ODS4) y el Plan Nacional Decenal de Educación en Colombia. Al fomentar una ciudadanía crítica, creativa y técnicamente competente, el desarrollo del pensamiento computacional deja de ser solo una meta académica para convertirse en un imperativo sociopolítico. De este modo, se asegura que las nuevas generaciones cuenten con las competencias necesarias para navegar y transformar la realidad de la era digital, respondiendo así a las demandas de equidad y calidad educativa que la sociedad contemporánea exige.

Conclusiones

El análisis de la literatura empleada para la construcción de este artículo devela que la programación y el pensamiento computacional mantienen una relación estrecha pero no equivalente. Es así como en la mayoría de las experiencias, la programación se presenta como un medio privilegiado para desplegar habilidades del Pensamiento Computacional, ya sea mediante lenguajes visuales, robótica o entornos de hardware como Arduino (Marín et al., 2024). Sin embargo, varios estudios advierten que reducirlo al simple dominio de un lenguaje de programación implica una visión code-céntrica que empobrece su alcance formativo (Li et al., 2020). De hecho, se documentan experiencias en las que este, se desarrolla mediante actividades unplugged, en las que el estudiante resuelve problemas y explicita estrategias algorítmicas sin necesidad de usar computador ni escribir código, lo que evidencia que la lógica subyacente al Pensamiento Computacional no depende de la codificación en sí misma (Hernández et al., 2023).

Asimismo, investigaciones en educación STEM muestran que el Pensamiento Computacional se integra en matemática, ciencias y tecnología

como un marco de análisis y modelación de problemas, más que con el objetivo de enseñar programación (Jiménez y Carmona, 2023). En esta línea, la literatura coincide en caracterizar el pensamiento computacional como una estructura lógica de alto nivel que organiza la formulación, representación y resolución sistemática de problemas, mientras que la programación se entiende como uno de los lenguajes posibles para expresar e implementar las soluciones derivadas de dicha estructura.

Por consiguiente, el Pensamiento Computacional articula un ciclo que incluye comprender el problema, descomponerlo, seleccionar información relevante, construir modelos (frecuentemente de naturaleza matemática), diseñar algoritmos y contrastar la pertinencia de las soluciones, independientemente de si estas se codifican o no en un lenguaje específico (Ocampo et al., 2024). En consecuencia, la programación opera como una gramática formal que hace visible y ejecutable ese andamiaje lógico, pero no lo agota; debido a que es posible programar de manera mecánica sin desarrollar auténtico Pensamiento Computacional, así como es posible movilizarlo en contextos donde no se escriben códigos. Desde esta perspectiva, el Pensamiento Computacional sobrepasa la programación porque se configura como una forma de pensamiento estructurado, transferible y transversal a múltiples disciplinas, mientras que la programación constituye solo uno de los vehículos para su materialización.

Referencias

Adell, J., Laviña, J., & Pascual, M. A. (2019). *El pensamiento computacional en la educación obligatoria: Informe para el INTEF*. Ediciones Universidad de Cantabria / IBSTE. https://intef.es/wp-content/uploads/2019/03/Pensamiento-Computacional-en-la-Educacion-Obligatoria.pdf

Acevedo, J., Valverde, J., y Garrido, M. (2021). *Pensamiento computacional y tecnología educativa: Una revisión exploratoria de la literatura*. Educación y Tecnologías de la Información, 13(2), 45–62.

Argoti, L. (2024). *El pensamiento computacional como soporte del pensamiento matemático*. Voces y Silencios, 15(1), 1–22. https://doi.org/10.18175/VyS15.1.2024.5

Bocconi, S., Chioccariello, A., & Earp, J. (2022). *Reviewing Computational Thinking in Compulsory Education*. Publications Office of the European Union. https://doi.org/10.2760/126955

Cabra, M., y Ramírez, S. (2021). *Desarrollo del pensamiento computacional y las competencias matemáticas en análisis y solución de problemas: una experiencia de aprendizaje con Scratch*. Revista Educación, 46(1), 1–30. https://doi.org/10.15517/revedu.v46i1.46721

Gamito, R., Basasoro, M., Aristizabal, P., y León, I. (2022). *El desarrollo del pensamiento computacional en educación: valoración basada en una experiencia con Scratch*. Innoeduca, 8(1), 59–74. https://doi.org/10.24310/innoeduca.2022.v8i1.13757

García-Peñalvo, F. J. (2018). El Pensamiento Computacional como una competencia transversal. *Education in the Knowledge Society (EKS)*, *19*(4), 7-12. https://doi.org/10.14201/eks2018194712

Hernández, C., Gamboa, A., y Prada, R. (2023). *Desarrollo de pensamiento computacional en educación media mediante actividades unplugged*. Revista Colombiana de Educación, 13(25), 74–90. https://doi.org/10.61799/2216-0388.1415

Hwang, G. J., Liang, J. C., & Hsu, K. Y. (2023). A review of computer-based computational thinking assessments. *Journal of Educational Computing Research*, *61*(2), 345-372. https://doi.org/10.1177/07356331221115645

Jiménez, J., y Carmona, E. (2023). *Construcción del pensamiento computacional mediante la incorporación de la educación STEM en el currículo de secundaria del departamento del Quindío (Colombia)*. Región

Knuth, D. (1997). *The art of computer programming* (3rd ed.). Addison-Wesley.

Li, Y., Schoenfeld, A., diSessa, A., Graesser, A., Benson, L., English, L., y Duschl, R. (2020). *On computational thinking and STEM education*. Journal for STEM Education Research, 3, 147–166. https://doi.org/10.1007/s41979-020-00044-w (rc.cienciasas.org)

Marín J., García, P., y Duo, P. (2024). *Computational thinking and programming with Arduino in education: A systematic review for secondary education*. Heliyon, 10(8), e29177. https://doi.org/10.1016/j.heliyon.2024.e29177 (PubMed)

Ministerio de Educación Nacional (MEN). (2004). *Estándares básicos de competencias en matemáticas*. Bogotá: MEN.

Ministerio de Educación Nacional (MEN). (2018). *Orientaciones pedagógicas para el desarrollo del pensamiento lógico-matemático en la educación inicial*. Bogotá: MEN.

Ocampo, L., Corrales M., y Cardona, S. (2024). *Definiciones del pensamiento computacional: Una revisión de la literatura*. Revista EIA, 21(42), 1–24. https://doi.org/10.24050/reia.v21i42.1716

Orozco, J., y Zapa, J. (2023). *Revisión sistemática sobre pensamiento computacional en Iberoamérica: avances y desafíos*. Revista Latinoamericana de Educación en Tecnología, 4(2), 23–41.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.

Plan Nacional Decenal de Educación. El camino hacia la calidad y hacia la equidad (2016-2026). https://www.mineducacion.gov.co/1780/articles-392871_recurso_1.pdf

Polanco, N., Ferrer, S., y Fernández, M. (2021). *Aproximación a una definición de pensamiento computacional*. Revista Iberoamericana de Educación a Distancia, 24(1), 55–65. https://doi.org/10.5944/ried.24.1.28081

Quiroz, D., Carmona, J., Castrillón, A., y Villa, J. (2021). *Integración del pensamiento computacional en la educación primaria y secundaria en Latinoamérica: una revisión sistemática*. RED. Revista de Educación a Distancia, 21(68), 1–25. https://doi.org/10.6018/red.424101

Shute, V., Sun, C., y Asbell, J. (2017). *Demystifying computational thinking.* Educational Research *Review, 22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Terroba, M., Ribera, J., y Lapresa, D. (2021). *Pensamiento computacional en la resolución de problemas en Educación Infantil*. Edma 0-6: Educación Matemática en la Infancia, 9(2), 73–92.

Tinedo, A. (2022). *ICTs and computational thinking to foster language skills: A proposal based on Scratch*. Comunicar, 30(70), 91–101. https://doi.org/10.3916/C70-2022-08

Voogt, J., Fisser, P., Good, J., Mishra, P., y Yadav, A. (2015). *Computational thinking in compulsory education: Towards an agenda for research and practice.* Education and Information Technologies, 20*(4), 715–728. https://doi.org/10.1007/s10639-015-9412-6*

Weller, J., Fernández, M., y Álvarez, P. (2021). *Pensamiento computacional y educación matemática: conexiones y perspectivas de investigación*. Revista Latinoamericana de Tecnología Educativa, 20(1), 15–34.

Wing, J. (2006). *Computational thinking*. Communications of the ACM, 49(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wing, J. (2009). *Computational thinking and thinking about computing*. Philosophical Transactions of the Royal Society A, 366(1881), 3717–3725. https://doi.org/10.1098/rsta.2008.0118

Zapata-Ros, M. (2019). El pensamiento computacional, una cuarta competencia básica. *RED. Revista de Educación a Distancia*, *19*(60). http://dx.doi.org/10.6018/red/60/11