

Integrando el desarrollo del pensamiento computacional y las competencias STEM en la educación de ingeniería: una revisión de la literatura

Pedro S. Gómez Álvarez & Julio Ariel Hurtado Alegría

Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, Popayán, Colombia.

pgomez@unicauca.edu.co, ahurtado@unicauca.edu.co

Resumen— El pensamiento computacional ocupa hoy un lugar central en la formación de ingenieros, sobre todo en programas que apuestan por integrar las competencias STEM (Ciencia, Tecnología, Ingeniería y Matemáticas). Partiendo de esa premisa, este trabajo analiza la literatura científica sobre la integración del pensamiento computacional y las competencias STEM en los primeros semestres de los programas de ingeniería mediante un mapeo sistemático que siguió el protocolo PRISMA y empleó el marco PICOC para formular las preguntas de investigación. La búsqueda abarcó bases de datos académicas de los campos de la educación, la ingeniería y la informática, de donde se seleccionaron y analizaron 26 estudios primarios publicados entre 2013 y 2025. Los hallazgos muestran que la integración del pensamiento computacional y las competencias STEM se articula principalmente a través de metodologías activas —aprendizaje basado en problemas y en proyectos— con apoyo de herramientas tecnológicas y entornos digitales. Habilidades como la descomposición y el diseño algorítmico reciben mayor atención en los estudios revisados, mientras que la abstracción y la generalización siguen siendo procesos difíciles de consolidar en etapas tempranas. A eso se suman vacíos que la literatura no ha resuelto: pocos estudios tienen carácter longitudinal, los métodos de evaluación del pensamiento computacional distan de estar estandarizados y la investigación desde contextos latinoamericanos es todavía escasa. Frente a ese panorama, el estudio ofrece una lectura actualizada del campo y plantea la necesidad de investigaciones más situadas que puedan orientar el diseño curricular y pedagógico de la formación inicial en ingeniería en América Latina.

Palabras Clave— Pensamiento computacional, competencias STEM, educación en ingeniería, mapeo sistemático de la literatura, primeros semestres.

Recibido: 2 de junio de 2025. Revisado: 24 de junio de 2026. Aceptado: 28 de julio de 2026.

Integrating computational thinking and stem competency development in engineering education: A review of the literature

Abstract— Computational thinking has become a central element in engineering education, particularly in programs committed to integrating STEM competencies (Science, Technology, Engineering, and Mathematics). Building on that premise, this study examines the scientific literature on the integration of computational thinking and STEM competencies in the early semesters of engineering programs through a systematic mapping that followed the PRISMA protocol and employed the PICOC framework to structure the research questions. The search covered academic databases in the fields of education, engineering, and computer science, from which 26 primary studies published between 2013 and 2025 were identified and analyzed. The findings show that Computational thinking –STEM integration is carried out primarily through active learning methodologies —problem-based and project-based learning— supported by technological tools and digital environments. Skills such as decomposition and algorithmic design receive more attention in the

reviewed studies, while abstraction and generalization remain difficult to consolidate in early stages of training. Added to this are gaps that the literature has yet to address: few studies are longitudinal in nature, computational thinking assessment methods are far from standardized, and research from Latin American contexts remains scarce. Against this backdrop, the study offers an updated reading of the field and underscores the need for more contextualized research to inform curricular and pedagogical design for early engineering education in Latin America.

Keywords— Computational thinking, STEM competencies, engineering education, systematic literature mapping, early semesters.

1 Introducción

El pensamiento computacional (*computational thinking*, CT) es un concepto cuyo origen se encuentra principalmente en el trabajo pionero de Seymour Papert en el MIT a lo largo de la década de 1980 [1], y que posteriormente fue conceptualizado y difundido a partir del trabajo de Jeannette M. Wing (2006). En el caso de esta investigación, el pensamiento computacional se comprende explícitamente en el marco de la competencia cognitiva transversal de una persona, enfocada en la capacidad de resolución de problemas y la formulación de soluciones con base en ciertos principios fundamentales de la informática, relevantes para el análisis y el diseño de algunos sistemas complejos de ingeniería. Por lo tanto, tal y como postula Wing, el pensamiento computacional es una competencia del siglo XXI que trasciende la asociación con el conocimiento de la programación [2]. En este sentido, el pensamiento computacional implica un enfoque estructurado para la resolución de problemas que integra procesos como la abstracción, la descomposición y el diseño algorítmico, ampliamente reconocidos en la literatura como componentes centrales de esta competencia [3].

En los últimos años, cada vez es más habitual la introducción de la informática, y por tanto también el pensamiento computacional, en diferentes contextos de enseñanza, sobre todo aquellos vinculados a la educación STEM [4]. El término STEM adopta en este estudio un enfoque educativo integral, que va más allá de un simple agregado de disciplinas (ya sean de ciencias, tecnología, ingeniería o matemáticas) y va encaminado al desarrollo de competencias en solución de problemas complejos dentro de contextos reales.

Distintas investigaciones han evidenciado que el trabajo con contenidos y contextos STEM ayuda en el desarrollo del pensamiento computacional y, por otra parte, que la incorporación del pensamiento computacional potencia la enseñanza en esas áreas [5]. Desde esta perspectiva, las competencias STEM incluyen habilidades como el pensamiento crítico, la resolución de problemas, el trabajo colaborativo y la aplicación integrada del conocimiento científico y tecnológico, aspectos especialmente relevantes en la formación en ingeniería.

Integrar el pensamiento computacional en la formación en ingeniería no solo enriquece el proceso de aprendizaje, sino que también potencia habilidades clave como la resolución de problemas y el diseño de soluciones, aspectos esenciales para que los estudiantes se preparen mejor ante los desafíos académicos y profesionales [6], [7].

Con la perspectiva de este estudio se plantea la integración del pensamiento computacional y las competencias STEM como un proceso educativo y sistémico intencional, que involucra al estudiante de ingeniería como sujeto de aprendizaje, al docente como mediador pedagógico y al currículo como estructura articuladora de dichas competencias.

En cambio, aunque cada vez son más quienes reconocen la importancia del pensamiento computacional en la educación en ingeniería, la bibliografía demuestra que todavía no hay una revisión sistemática de cómo se concreta esta integración al comienzo de los programas de ingeniería —y este vacío, tanto investigativo como curricularmente hablando— es relevante.

En esta misma línea de razonamiento, Kelley y Knowles (2016) conciben la educación STEM como un enfoque educativo integrador, basado en la articulación intencional de una o más disciplinas STEM —ciencia, tecnología, ingeniería y matemáticas— en entornos de enseñanza-aprendizaje orientados a la resolución de problemas del mundo real, promoviendo un aprendizaje más aplicado, contextualizado y significativo [8]. Desde esta perspectiva, la educación STEM no solo busca integrar contenidos disciplinares, sino también favorecer el desarrollo de competencias transversales relevantes para la formación en ingeniería. Según Mohtadi et al. (2013), muchos estudiantes que ingresan en carreras de ingeniería presentan vacíos significativos en sus competencias STEM; esto influye en su rendimiento académico inicial y muestra la conveniencia de investigar estrategias pedagógicas que integren el pensamiento computacional como una de ellas [9]. En este contexto, el presente artículo analiza la literatura científica relacionada con la integración del pensamiento computacional y las competencias STEM en la formación en ingeniería.

Para ello se realizó un mapeo sistemático de la literatura bajo las directrices del protocolo PRISMA y el marco PICOC, lo que permitió definir con precisión las preguntas de investigación, los criterios de selección y el procedimiento de análisis de los estudios primarios. El estudio busca identificar, clasificar y sintetizar los enfoques, estrategias pedagógicas, herramientas y brechas que la literatura reporta sobre la integración del pensamiento computacional y las competencias STEM en la educación en ingeniería, con atención particular a los primeros semestres de las carreras. Vale señalar además que este interés no es ajeno al momento que atraviesa la disciplina, Gonzales-Ttito et al. (2025) advierten que, ante la expansión de la

inteligencia artificial y el procesamiento masivo de datos, el pensamiento computacional cobra una relevancia estratégica que la formación en ingeniería no puede ignorar [10].

El artículo se estructura así: la sección 2 detalla la metodología del mapeo —diseño del estudio, preguntas de investigación derivadas del marco PICOC, estrategia de búsqueda, criterios de inclusión y exclusión, proceso de selección y diagrama PRISMA—; la sección 3 expone los resultados organizados por pregunta de investigación; la sección 4 discute los hallazgos en torno a la integración del pensamiento computacional y las competencias STEM en la educación en ingeniería; y la sección 5 cierra con las conclusiones, las limitaciones identificadas y las líneas de trabajo futuro.

2 Marco teórico y conceptual

2.1 *Pensamiento computacional en la educación en ingeniería*

El término pensamiento computacional fue mencionado inicialmente en las décadas de 1980 y 1990 en el trabajo de Seymour Papert con el lenguaje de programación LOGO y el desarrollo del pensamiento procedimental en niños a través de la programación [1]. Posteriormente, en 2006, Jeannette Wing reutiliza y populariza el pensamiento computacional para referirse a una habilidad fundamental que, junto con la lectura, la escritura y la aritmética, debería hacer parte de la capacidad analítica a desarrollar por todas las personas [2]. En este acercamiento inicial, ella sostiene que el pensamiento computacional involucra la solución de problemas, el diseño de sistemas y la comprensión del comportamiento humano con base en los conceptos fundamentales de la informática, y comprende —pero no se limita a— la reformulación de problemas, el pensamiento recursivo, el procesamiento paralelo, la interpretación de código como información y viceversa, la abstracción, la descomposición y la modularización, y el razonamiento heurístico. A partir de esta visión originaria, han surgido diversos intentos por obtener una definición concreta y certera de lo que es el pensamiento computacional, sin que hasta el momento se haya llegado a un consenso al respecto [11].

La misma Wing refina más adelante su concepción del pensamiento computacional para visualizarlo como un marco para el pensamiento y, más precisamente, como “el proceso de pensamiento involucrado en la formulación de un problema y la expresión de su(s) solución(es), de tal manera que una computadora —humana o máquina— pueda ejecutarla efectivamente” [2], destacando la abstracción como su esencia y resaltando, al igual que otros investigadores, que no se trata solamente de solucionar un problema, sino también de formularlo [3].

La Sociedad Internacional para la Tecnología en la Educación (ISTE) y la Sociedad de Maestros de Ciencias de la Computación (CSTA) elaboraron conjuntamente una definición operacional de conceptos, que provee un marco y un vocabulario para el pensamiento computacional [9].

El pensamiento computacional incluye los procesos de pensamiento implicados en la formulación de problemas y de sus soluciones, de modo que queden representados de una manera que pueda ser abordada de forma efectiva por un

agente-procesador de la información, como una computadora” [1]. En este tipo de pensamiento se enmarcan los siguientes conceptos fundamentales [12]:

- Abstracción: es la habilidad de identificar y extraer información relevante de un problema o una solución.
- Descomposición: habilidad de separar los datos, procesos, soluciones o problemas en partes más pequeñas y manejables.
- Reconocimiento de patrones: la transferencia de un proceso de resolución de problemas a una amplia variedad de problemas.
- Algoritmo: desarrolla una serie ordenada de instrucciones que una máquina o un individuo debe seguir al ejecutarlo, resolverlo o ejecutar una tarea.
- Programación basada en bloques: incluye el ensamblaje de piezas prefabricadas, como armar un rompecabezas, de modo que hay una lista de instrucciones o pasos para resolver un problema definido explícitamente o creado por el usuario. Es un método conveniente y visual de programación en entornos gráficos.

Las primeras definiciones de pensamiento computacional giraban casi exclusivamente en torno a la resolución de problemas mediante conceptos propios de la informática. Con el tiempo, esa visión se fue ampliando. Palop et al. (2025) proponen un marco holístico que organiza el pensamiento computacional en tres dimensiones: la formulación de problemas, el diseño de algoritmos y el tratamiento de datos [13]. Bajo esa lectura, el pensamiento computacional deja de ser sinónimo de programación para convertirse en un conjunto de procesos cognitivos y metodológicos orientados a estructurar problemas complejos, identificar patrones, modelar soluciones y analizar información; una competencia transversal, en definitiva, aplicable a múltiples campos del conocimiento y especialmente pertinente en la formación en ingeniería y en contextos STEM. Esa trayectoria conceptual ha sido documentada por Lodi y Martini (2021), quienes rastrean la evolución del término desde los trabajos originales de Papert hasta su consolidación en la literatura reciente. Su análisis muestra cómo el pensamiento computacional fue desplazando su centro de gravedad: de una habilidad técnica ligada a la programación hacia una competencia cognitiva más amplia, vinculada al razonamiento algorítmico y a la resolución estructurada de problemas en distintas disciplinas [14]. Todo ello lleva a entender el pensamiento computacional no como una aptitud instrumental para escribir código, sino como una competencia cognitiva de fondo, relevante para el ejercicio profesional en ingeniería [12].

2.2 Educación STEM

El enfoque STEM (abreviatura en inglés de Ciencia, Tecnología, Ingeniería y Matemáticas) se ha convertido en las últimas décadas en una importante respuesta educativa a la creciente necesidad de profesionales capaces de abordar problemas complejos de forma interdisciplinaria [4]. Según Kelley y Knowles (2016), lo caracterizan como un modelo integrador que vincula deliberadamente dos o más de esas disciplinas en experiencias de aprendizaje ancladas en problemas reales, donde el conocimiento se pone a prueba en

contexto y no como abstracción teórica [8].

Lo que distingue al enfoque STEM de una simple yuxtaposición de materias es su apuesta por el desarrollo de competencias: conjuntos de conocimientos, habilidades y actitudes que permiten actuar con eficacia frente a situaciones complejas. Entre las competencias STEM que la literatura menciona con mayor frecuencia aparecen el pensamiento crítico, la resolución de problemas, el trabajo colaborativo, la comunicación eficaz y la capacidad de aplicar de forma integrada el conocimiento científico y tecnológico [4]. En ingeniería, estas capacidades no son un complemento deseable sino una exigencia real: los estudiantes deben enfrentarse cotidianamente a problemas abiertos, ambiguos y sin una solución única.

El problema es que muchos de quienes ingresan a esas carreras llegan con debilidades notorias en esas mismas competencias, lo que afecta tanto su rendimiento académico inicial como su continuidad en el programa. Frente a ese diagnóstico, varios investigadores han propuesto la integración del pensamiento computacional como estrategia articuladora: una competencia que, al conectar los distintos componentes del currículo, puede potenciar el desarrollo de las competencias STEM desde el comienzo de la formación [9]. Masaquiza y otros (2024) aportan evidencia en esa dirección desde el contexto latinoamericano, donde la educación STEM ha mostrado resultados favorables en el fortalecimiento de habilidades como el pensamiento crítico y la resolución de problemas [15].

De ahí que la convergencia entre pensamiento computacional y educación STEM no sea solo teóricamente coherente, sino metodológicamente útil para analizar propuestas educativas en ingeniería, en especial durante los primeros semestres, cuando se construyen las bases conceptuales y cognitivas sobre las que se apoyará toda la formación profesional. El pensamiento computacional, en ese esquema, aporta estructuras de pensamiento que facilitan la articulación de las competencias STEM y justifica la necesidad de revisiones sistemáticas que mapeen lo que se ha hecho, lo que ha funcionado y lo que aún falta por explorar.

3 Metodología del mapeo sistemático

3.1 Diseño del estudio

El enfoque metodológico de este trabajo es el mapeo sistemático de la literatura. La elección no es arbitraria: este tipo de revisión es especialmente útil cuando lo que se busca no es medir la efectividad de una intervención concreta, sino trazar un panorama del campo —qué se ha investigado, bajo qué enfoques, con qué herramientas y con qué resultados—. Preguntas de ese carácter exploratorio son precisamente las que guían este estudio sobre la integración del pensamiento computacional y las competencias STEM en los primeros semestres de la formación en ingeniería.

Para darle rigor al proceso y permitir que otros puedan replicarlo, se siguieron los lineamientos del protocolo PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*) [16]. Eso supuso definir por adelantado los criterios

de inclusión y exclusión de los estudios, así como establecer un procedimiento claro para cada fase: identificación de fuentes, selección de estudios y análisis de los primarios incluidos.

3.2 Objetivos

El objetivo general fue identificar, clasificar y analizar las metodologías, estrategias pedagógicas, herramientas tecnológicas y resultados reportados en estudios e investigaciones orientadas a la integración del pensamiento computacional y las competencias STEM en las actividades formativas de los primeros semestres de programas de ingeniería. De otra parte, los objetivos específicos fueron:

- Identificar y clasificar las estrategias reportadas en la literatura para introducir el pensamiento computacional y las competencias STEM en los primeros semestres de programas de ingeniería.
- Identificar las habilidades de pensamiento computacional y las competencias STEM más abordadas en los estudios que analizan su integración en la educación en ingeniería.
- Identificar las herramientas y tecnologías empleadas para apoyar la integración del pensamiento computacional y STEM en las actividades formativas iniciales de la carrera de ingeniería.
- Revisar las modalidades de evaluación utilizadas para determinar la efectividad de las intervenciones orientadas a la integración del pensamiento computacional y STEM en los primeros semestres de la formación en ingeniería.

3.3 Uso de la estructura de marco PICOC en la formulación de preguntas de investigación

Con el fin de estructurar de manera sistemática las preguntas de investigación y delimitar el alcance del estudio, se empleó el marco PICOC (*Population, Intervention, Comparison, Outcome, Context*), ampliamente utilizado en estudios de revisión de la literatura:

- Población: estudiantes de los primeros semestres de programas de ingeniería, así como docentes que imparten asignaturas en dichos programas.
- Intervención: estrategias, metodologías o enfoques pedagógicos para la enseñanza del pensamiento computacional y las competencias STEM.
- Comparación: diferentes estrategias o enfoques de integración del pensamiento computacional y STEM en la enseñanza de la ingeniería.
- Resultados: efectos sobre el aprendizaje del pensamiento computacional, el desarrollo de competencias STEM, el rendimiento académico, la motivación y el compromiso estudiantil.
- Contexto: instituciones de educación superior con programas de ingeniería, particularmente en los primeros semestres de formación.

A partir de ese marco se definieron las preguntas que orientan el estudio.

3.4 Preguntas de investigación

La pregunta central del mapeo es: ¿cómo se ha abordado la integración del pensamiento computacional y las competencias STEM en las actividades formativas de los primeros semestres de la educación en ingeniería? Para desagregarla en aspectos concretos y manejables, se formularon seis preguntas específicas:

- PI-1: ¿Qué estrategias se han propuesto en la literatura para integrar el pensamiento computacional y las competencias STEM en los primeros semestres de la formación en ingeniería?
- PI-2: ¿Cuáles son las habilidades de pensamiento computacional y las competencias STEM más abordadas en los estudios analizados?
- PI-3: ¿Qué herramientas o tecnologías se han utilizado para apoyar esa integración en las actividades formativas de ingeniería?
- PI-4: ¿De qué manera se ha evaluado la eficacia de las intervenciones que combinan pensamiento computacional y STEM?
- PI-5: ¿Qué elementos de las estrategias identificadas podrían trasladarse a la formación en los primeros semestres de ingeniería?
- PI-6: ¿Qué brechas y retos señala la literatura respecto a la integración del pensamiento computacional y STEM en la educación en ingeniería?

En conjunto, estas preguntas orientan la búsqueda, la selección y el análisis de los estudios, con el propósito de construir una visión coherente del estado actual de la integración del pensamiento computacional y las competencias STEM en la educación superior en ingeniería.

3.5 Estrategia de búsqueda

La búsqueda se orientó a localizar estudios sobre la integración del pensamiento computacional y las competencias STEM en la educación en ingeniería. Se incluyeron tanto artículos científicos como tesis académicas para ampliar la cobertura del mapeo y no restringirlo únicamente a publicaciones en revistas. Las fuentes consultadas fueron ACM Digital Library, IEEE Xplore, ERIC, Scopus, SpringerLink, SpringerOpen y ASEE PEER. Elegir las bases de datos no fue un paso menor. Cada una cubre un flanco distinto de la literatura que este mapeo necesitaba rastrear, y usarlas en conjunto fue lo que permitió no dejar fuera trabajos relevantes por estar publicados en comunidades disciplinares distintas.

ACM Digital Library e IEEE Xplore fueron el punto de entrada natural para todo lo relacionado con las ciencias de la computación e ingeniería de sistemas, campos en los que el pensamiento computacional tiene más arraigo y donde se ha desarrollado buena parte de su base conceptual. ERIC, por su parte, cubre el territorio que las otras dos no alcanzan: es la base de mayor alcance en investigación educativa a nivel internacional, y sin ella habría sido difícil recuperar los estudios centrados en estrategias pedagógicas y desarrollo de competencias STEM. Scopus y SpringerLink ofrecen cobertura multidisciplinar amplia con revisión por pares, lo que permite

capturar trabajos en la intersección entre educación, ingeniería e informática que no siempre aparecen en bases especializadas. SpringerOpen amplía esa cobertura hacia publicaciones en acceso abierto, con mayor representación de investigaciones provenientes de contextos no anglosajones. Finalmente, ASEE PEER es el repositorio específico de la American Society for Engineering Education, que concentra contribuciones sobre innovación pedagógica en ingeniería y es una fuente de referencia obligada para este tipo de revisiones.

La decisión de incluir términos de búsqueda en inglés y español respondió a la intención deliberada de no excluir producción académica latinoamericana, que con frecuencia queda fuera de los mapeos cuando la búsqueda se restringe a un solo idioma.

La cadena de búsqueda combinó términos en inglés y español mediante operadores booleanos: [("*engineering education*" or "*educación en ingeniería*") and ("*computational thinking*" or "*pensamiento computacional*" or "*ct*") and ("*stem competences*" or "*stem skills*" or "*competencias stem*") and ("*integrating*" or "*integración*")].

La decisión de incluir términos en ambos idiomas respondió a la intención de capturar también producción académica latinoamericana, que con frecuencia queda fuera de los mapeos cuando la búsqueda se restringe al inglés.

3.6 Criterios de inclusión y exclusión

Los criterios que dan validez y calidad a los estudios seleccionados están en relación a los objetivos del apartado para identificar y analizar estudios relevantes que se centran en la integración del pensamiento computacional y las competencias STEM en las actividades de aprendizaje de los primeros semestres de estudios de ingeniería. Los criterios de inclusión fueron:

- CI1: estudios que analicen estrategias, metodologías o enfoques para la integración del pensamiento computacional y las competencias STEM en los primeros semestres de programas de ingeniería.
- CI2: estudios que evalúen el impacto de dicha integración en el aprendizaje, el desarrollo de competencias, el rendimiento académico o la motivación estudiantil.
- CI3: estudios comparativos entre distintas estrategias de integración pensamiento computacional–STEM.
- CI4: estudios realizados en instituciones de educación superior con programas de ingeniería.
- CI5: estudios publicados entre 2013 y 2025.

Muestras los criterios de exclusión

- CE1: publicaciones duplicadas identificadas en más de una base de datos.
- CE2: estudios cuyo texto completo no se encontrará disponible.
- CE3: publicaciones que no aborden la educación en ingeniería o no consideren explícitamente pensamiento computacional y STEM.
- CE4: estudios que, aunque aborden pensamiento computacional o STEM, no presenten diseño metodológico explícito —ya sea empírico, experimental, de caso o de revisión sistemática—,

es decir, trabajos de opinión, editoriales o reflexiones sin sustento en datos o evidencia.

- CE5: publicaciones en las que pensamiento computacional o STEM aparezcan de forma tangencial o instrumental, sin constituir un objeto de estudio central ni ser analizados de manera sistemática dentro del trabajo.

3.7 Proceso de selección de estudios

El proceso de selección se desarrolló siguiendo los lineamientos del protocolo PRISMA, con el fin de garantizar rigor y transparencia en la identificación y selección de los estudios relevantes. Este proceso comprendió cuatro fases: identificación, filtrado inicial, elegibilidad e inclusión final. Finalmente, un total de 26 estudios primarios cumplieron con todos los criterios establecidos y fueron incluidos en el análisis del mapeo sistemático. El diagrama de flujo PRISMA presentado en la Fig. 1 resume el proceso de selección y el número de estudios en cada fase.

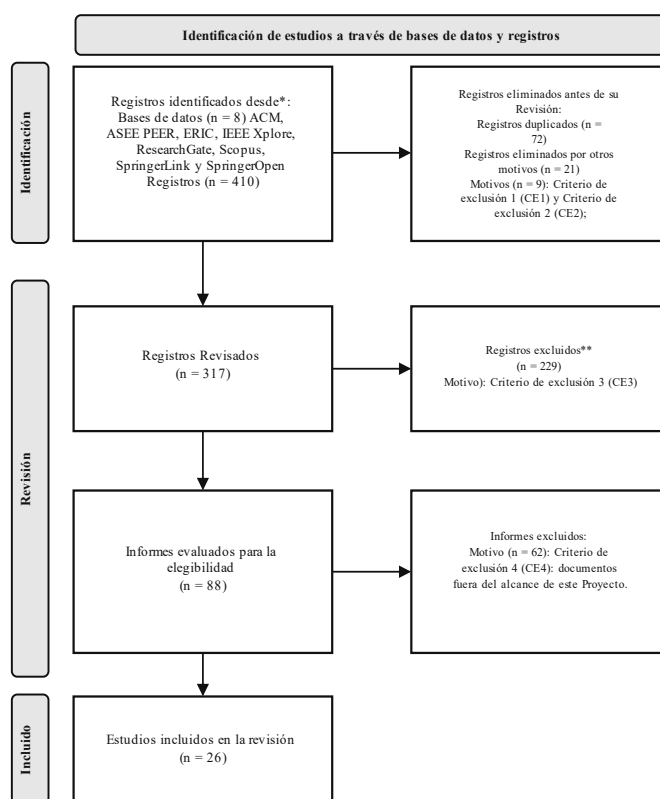


Figura 1. Diagrama de flujo Prisma
Fuente: Elaboración propia.

4 Resultados

Los resultados del mapeo sistemático se presentan organizados de acuerdo con las preguntas de investigación formuladas en la sección 3, con el fin de ofrecer una visión estructurada y coherente de los enfoques, estrategias y hallazgos reportados en la literatura sobre la integración del pensamiento computacional y las competencias STEM en la educación en ingeniería.

4.1 Descripción general de los estudios seleccionados

Aplicados los criterios de inclusión y exclusión, el corpus final quedó conformado por 26 estudios primarios. Una mirada al conjunto permite distinguir dos tipos de trabajos: los de orientación teórica —revisiones de literatura, marcos conceptuales y propuestas de modelos— y los de orientación práctica —estudios de caso, diseños curriculares, evaluaciones de impacto e implementaciones en contextos educativos reales—. Los primeros representan el 42 % de la muestra (11 estudios) y los segundos el 58 % restante (15 estudios), lo que indica un campo con producción empírica activa, aunque con una base teórica todavía en construcción.

En cuanto a la distribución temporal, los 26 estudios seleccionados abarcan el período 2013–2025. De ellos, 11 (42 %) fueron publicados entre 2020 y 2025, lo que refleja un crecimiento notable de la producción académica sobre integración pensamiento computacional –STEM en los últimos años y garantiza que el estado del arte incorpora desarrollos recientes del campo. Los 15 estudios restantes, publicados entre 2013 y 2019, aportan el contexto histórico y conceptual necesario para entender la evolución de los enfoques analizados. Esta distribución permite combinar consolidación teórica con evidencia actualizada, lo que fortalece la validez del mapeo como fotografía del estado actual de la investigación.

El proceso de selección se detalla en el diagrama PRISMA; la distribución de los documentos por categoría se presenta en la Fig. 2, Y la clasificación completa puede consultarse en el Anexo A.

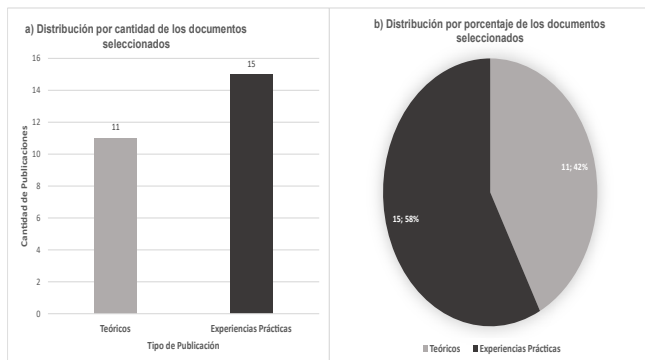


Figura 2. Distribución a) por cantidad y b) por porcentaje de los documentos seleccionados para revisión.

Fuente: Elaboración propia.

4.2 Resultados por pregunta de investigación

PI-1: Estrategias propuestas para integrar pensamiento computacional y competencias STEM en los primeros semestres de ingeniería. La literatura revisada evidencia que las estrategias de integración pensamiento computacional–STEM en educación en ingeniería se apoyan predominantemente en metodologías activas orientadas a la resolución de problemas auténticos. Entre los enfoques más recurrentes se encuentran el aprendizaje basado en problemas, el aprendizaje basado en proyectos, el aprendizaje basado en retos, y enfoques asociados al constructivismo y al movimiento *maker/thinking*, los cuales facilitan la integración interdisciplinaria y el desarrollo

de competencias STEM en contextos aplicados [4], [17], [18],[19]. Asimismo, se reportan propuestas de integración curricular y de rediseño formativo que buscan introducir el pensamiento computacional como componente transversal de la formación en ingeniería desde etapas tempranas [9], [20]. También se identifican estrategias que combinan metodologías activas con el uso de programación visual, robótica educativa y entornos digitales, con el propósito de favorecer el aprendizaje de pensamiento computacional mediante actividades prácticas y colaborativas [1], [21]. La heterogeneidad en la forma de operacionalizar la integración pensamiento computacional–STEM es uno de los rasgos más notorios del corpus revisado: las propuestas difieren tanto en profundidad como en enfoque, lo que hace difícil comparalas entre sí y apunta a que el campo todavía requiere mayor claridad conceptual y metodológica [4], [22]. Este hallazgo coincide con lo que Kelley y Knowles (2016) ya anticipaban al definir la educación STEM como un enfoque que exige anclar el aprendizaje en problemas del mundo real [8]. Lo que la literatura revisada añade es que esa aspiración no siempre se traduce en diseños coherentes: la brecha entre el enfoque declarado y su operacionalización concreta sigue siendo uno de los problemas no resueltos del campo.

PI-2: Habilidades de pensamiento computacional y competencias STEM más abordadas en los estudios. Las habilidades de pensamiento computacional que aparecen con mayor frecuencia en los estudios son la abstracción, la descomposición, el diseño algorítmico, el reconocimiento de patrones y la modularidad. Ahora bien, que sean las más mencionadas no significa que se trabajen de la misma manera: su tratamiento cambia según la estrategia pedagógica y las herramientas que cada propuesta utiliza. En contextos de robótica educativa, los estudiantes tienden a incorporar con más facilidad nociones como algoritmos, modularidad y descomposición; la abstracción y la generalización, en cambio, suelen resistirse más, especialmente en etapas iniciales [1], [21]. También se nota un interés creciente por incorporar el modelado, la simulación y la alfabetización digital como componentes del pensamiento computacional aplicado a la ingeniería [7]. En cuanto a las competencias STEM, la resolución de problemas, el trabajo colaborativo, el pensamiento crítico y la aplicación integrada del conocimiento científico-tecnológico son las que más peso tienen en los estudios que trabajan desde una perspectiva interdisciplinaria [18], [19]. Algunos trabajos reportan además algo que tiene implicaciones curriculares directas: hay asociaciones positivas entre las capacidades de pensamiento computacional y el rendimiento en ciencias en poblaciones preuniversitarias, lo que refuerza la idea de que el pensamiento computacional no opera de forma aislada sino como una competencia que atraviesa y potencia el desempeño en las disciplinas STEM [23]. Lodi y Martini (2021) ya advertían que el pensamiento computacional ha tendido históricamente a privilegiar los componentes más procedimentales —algoritmos, descomposición— sobre los más abstractos [14]. Que los estudios revisados reflejen exactamente ese patrón sugiere que la enseñanza del pensamiento computacional en ingeniería reproduce una limitación conceptual que viene de antes, y que no basta con incorporar el pensamiento

computacional al currículo si no se aborda deliberadamente el desarrollo de las habilidades cognitivas de mayor complejidad.

PI-3: Herramientas y tecnologías utilizadas para apoyar la integración pensamiento computacional –STEM. En la literatura revisada se identifica un uso recurrente de entornos de programación visual, actividades basadas en robótica educativa, y herramientas orientadas a la resolución de problemas mediante simulación y modelado. Los lenguajes por bloques han mostrado ser una entrada útil para el desarrollo del pensamiento computacional, particularmente en etapas de iniciación donde las actividades prácticas y colaborativas son el eje de la enseñanza [1], [23]. También se reportan escenarios donde el uso sostenido de herramientas de software en el currículo —desde el primer año— favorece la comprensión progresiva de conceptos matemáticos, programación e ingeniería, y potencia la autonomía del estudiante en procesos de indagación y diseño [9].

Esto es consistente con lo que Mohtadi et al. (2013) señalaban al proponer la integración del pensamiento computacional en ingeniería: la tecnología es un medio, no un fin, y su valor depende de la intencionalidad con que se incorpora al currículo [9]. Una década después, los estudios revisados confirman que esa advertencia sigue siendo pertinente.

PI-4: Modalidades de evaluación de intervenciones que integran pensamiento computacional –STEM. Evaluar el pensamiento computacional no es sencillo, y la literatura lo refleja: los estudios revisados emplean aproximaciones bastante distintas entre sí. Se reportan enfoques basados en tareas y pruebas estandarizadas, evaluaciones *pretest–posttest*, cuestionarios, observación en aula e instrumentos específicos para medir pensamiento computacional. Por ejemplo, se describen evaluaciones basadas en tareas vinculadas a programación visual y resolución de problemas, donde se analizan diferencias según tipos de tareas, habilidades objetivo y niveles de dificultad [24]. También se reporta el uso de pruebas basadas en tareas tipo Bebras, consideradas útiles para medir pensamiento computacional sin requerir conocimientos previos avanzados, y que han evidenciado brechas en preparación en estudiantes de primer año [25]. Asimismo, se identifican instrumentos validados específicamente para evaluar pensamiento computacional, como el test propuesto por Román-González [26]. En conjunto, la literatura señala una falta de estandarización en métodos de evaluación, lo cual limita la comparabilidad de resultados entre estudios [4].

PI-5: Elementos adaptables a los primeros semestres de ingeniería. Entre los elementos identificados con potencial de adaptación a los primeros semestres, se encuentran: (i) el uso de metodologías activas (problemas, proyectos, juegos) como estrategia para integrar pensamiento computacional en STEM [4], [11], (ii) el uso gradual de herramientas de software a lo largo de la trayectoria formativa para fortalecer aprendizaje de ingeniería y matemáticas [9], y (iii) enfoques basados en robótica educativa para promover habilidades específicas del pensamiento computacional en escenarios prácticos [1].

Asimismo, el análisis identifica la presencia de modelos específicos orientados al desarrollo del pensamiento computacional. En particular, se describe el modelo

ChildProgramming como un programa evolutivo y continuo que se expande a través de extensiones asociadas a colaboración, gamificación, diversidad de género, memoria transactiva y robótica educativa. Así, por ejemplo, aunque el concepto de pensamiento computacional no aparecía de forma explícita en el modelo original de Hurtado et al. [27], sí ha hecho parte de trabajos posteriores como el de Chimunja et al. [28], llamado ChildProgramming-C, y el estudio de Zúñiga et al. [29]. El modelo se ha expandido a través de algunas extensiones tales como: ChildProgramming-C, dirigida a satisfacer la dimensión colaborativa del modelo, ChildProgramming-G [30], enfocada en la gamificación como forma de incorporar el aspecto lúdico al mismo. Child Gender, encaminada a extender ChildProgramming con un aspecto de diversidad de género [31]. ChildProgramming-MT, dirigida a estudiar el modelo para encontrar elementos que permitan desarrollar un sistema de Memoria Transactiva [32], ChildProgramming Evolution, se hace una revisión general de la evolución de la metodología ChildProgramming, [33]. ChildProgramming-RE, enfocada a incentivar la participación y motivación hacia el desarrollo del pensamiento computacional en niños entre 10 y 12 años, a partir de los conceptos de STEM relacionados con la Robótica Educativa [11]. ChildDebugging. Se extraen las principales estrategias y recursos utilizados por los equipos de niños para depurar programas basados en el paradigma de bloques [34]. Block-Based Design es una propuesta que aplica mecanismos de abstracción propios del pensamiento computacional para enseñar a los niños a construir, descomponer e integrar bloques de programación [35]. En esa misma dirección, el modelo CT4E aborda el pensamiento computacional desde la perspectiva específica de la formación de ingenieros en contextos STEM [12]. Tomados en conjunto, estos trabajos configuran una línea metodológica con identidad propia dentro del corpus revisado: propuestas que buscan desarrollar el CT de manera progresiva y aplicada, partiendo de experiencias concretas para avanzar hacia niveles de mayor complejidad.

4.2.1 PI-6: Brechas y retos reportados por la literatura

El conjunto de estudios revisados no deja solo un mapa de lo que se ha hecho, sino también de lo que falta. Entre los vacíos más señalados está la escasez de estudios longitudinales; sin ese tipo de evidencia, es difícil saber qué pasa realmente con el CT y su incidencia sobre las competencias STEM una vez que termina la intervención, y si los efectos se sostienen o se diluyen con el tiempo. A eso se suma la poca investigación sobre adaptaciones curriculares pensadas específicamente para el primer semestre de ingeniería, que es precisamente donde las bases se construyen.

Otro flanco débil es la evaluación: las metodologías para medir el pensamiento computacional siguen siendo heterogéneas y poco estandarizadas, lo que hace casi imposible comparar la efectividad de estrategias aplicadas en contextos distintos. Y aunque menos visible, hay un vacío igual de importante en la formación docente: pocos programas preparan específicamente a los profesores para enseñar pensamiento computacional en etapas tempranas. A todo esto, se añade que

la diversidad —género, acceso tecnológico, condiciones socioeconómicas— rara vez se considera de forma explícita en los diseños estudiados, lo que pone en cuestión la equidad de los enfoques pensamiento computacional –STEM tal como se han implementado hasta ahora.

Masaquiza y otros (2024) y Gonzales-Ttito (2025) son de los pocos referentes que sitúan esta discusión en América Latina [10] y [15], lo que refuerza la escasez señalada y la urgencia de construir una base empírica regional que no dependa de extrapolar hallazgos producidos en contextos muy distintos.

5 Discusión

La presente discusión interpreta los principales hallazgos del mapeo sistemático a la luz del marco teórico expuesto y de las preguntas de investigación planteadas, con el propósito de analizar cómo se ha abordado la integración del pensamiento computacional (CT) y las competencias STEM en los primeros semestres de la educación en ingeniería, así como de identificar sus implicaciones, limitaciones y oportunidades de investigación futura.

5.1 *Discusión de los resultados en relación con las estrategias de integración CT–STEM (PI-1)*

Buena parte de los estudios revisados recurre a metodologías activas —aprendizaje basado en problemas y aprendizaje basado en proyectos— como vía principal para integrar el pensamiento computacional y las competencias STEM en la formación en ingeniería. La tendencia tiene sentido: estas metodologías parten de situaciones problemáticas reales y sitúan al estudiante como constructor activo de su propio conocimiento, lo que las alinea naturalmente con los principios constructivistas que sustentan el marco teórico de este trabajo.

El problema aparece cuando se mira más de cerca cómo se aplican. Que un estudio use aprendizaje basado en proyectos no garantiza que la integración pensamiento computacional –STEM sea profunda ni sistemática; de hecho, en muchos casos el pensamiento computacional se incorpora de forma tangencial a las actividades STEM sin que se explique con claridad cómo se conectan unas habilidades con otras. Esa falta de explicitación tiene consecuencias prácticas: las propuestas se vuelven difíciles de replicar y aún más difíciles de transferir a contextos educativos distintos al original.

5.2 *Discusión sobre las habilidades de pensamiento computacional y competencias STEM priorizadas (PI-2)*

Entre las habilidades de pensamiento computacional, la descomposición, el diseño de algoritmos y la modularidad concentran la mayor atención en los estudios revisados. La abstracción y la generalización, en cambio, aparecen con menos frecuencia y los datos sugieren por qué: son procesos cognitivamente más exigentes que los estudiantes en etapas iniciales los encuentran difíciles de dominar. Esto coincide con

lo que investigaciones previas ya habían advertido: estas habilidades no se desarrollan espontáneamente, sino que requieren mayor madurez cognitiva y un andamiaje pedagógico sostenido.

En cuanto a las competencias STEM, la resolución de problemas, el pensamiento lógico y el trabajo colaborativo son las que más peso tienen en los estudios analizados, lo cual es coherente con las demandas propias de la formación en ingeniería. Sin embargo, hay un punto ciego que vale la pena señalar: pocas investigaciones se detienen a explicar cómo estas competencias se desarrollan de forma articulada con el pensamiento computacional, no de manera paralela sino integrada. Ese vacío no es menor; mientras no haya modelos teóricos y metodológicos que den cuenta de esa relación, la integración pensamiento computacional –STEM seguirá siendo más una aspiración declarada que una práctica sistemática.

5.3 *Discusión sobre el papel de las herramientas y tecnologías (PI-3)*

Los entornos de programación visual y la robótica educativa aparecen en los estudios como facilitadores reales de la integración del pensamiento computacional en contextos STEM. Su valor principal está en que reducen la carga cognitiva inicial: al simplificar la sintaxis o hacer tangible lo abstracto, le permiten al estudiante concentrarse en la lógica del problema y en los procesos propios del pensamiento computacional, sin que la herramienta en sí se convierta en un obstáculo.

Dicho esto, los datos también dejan claro que la tecnología no hace el trabajo sola. Una herramienta bien diseñada puesta en manos de un currículo sin intención pedagógica clara produce resultados limitados. La efectividad de estos recursos depende de cómo se integran en el proceso formativo y de si están alineados con los objetivos del programa de ingeniería, algo que varios de los estudios revisados no terminan de explicitar.

5.4 *Discusión sobre la evaluación del pensamiento computacional (PI-4)*

Medir el pensamiento computacional no es tarea sencilla, y los estudios revisados lo confirman: las formas de evaluarlo van desde pruebas estandarizadas hasta tareas de desempeño y observación directa en el aula, sin que haya un enfoque dominante.

Esa variedad no es casual; responde a la naturaleza misma del pensamiento computacional, que es una competencia compleja y difícil de capturar con un solo instrumento. Pero también tiene un costo: cuando cada estudio evalúa a su manera, comparar resultados entre investigaciones se vuelve casi imposible.

Mientras no existan instrumentos con mayor aceptación en la comunidad, las conclusiones sobre qué intervenciones pensamiento computacional –STEM funcionan mejor en la educación en ingeniería seguirán siendo parciales. No se trata de uniformar todo, sino de construir referentes comunes que den más solidez a la evidencia acumulada.

5.5 *Discusión sobre modelos y elementos adaptables a los primeros semestres (PI-5)*

ChildProgramming y sus extensiones representan uno de los desarrollos más sistemáticos dentro de la literatura revisada: proponen una progresión estructurada para el desarrollo del pensamiento computacional e incorporan elementos como el andamiaje, la gamificación y el trabajo colaborativo que, sobre el papel, tienen potencial para la formación inicial en ingeniería.

Ahora bien, hay un dato de contexto que no puede pasarse por alto: estos modelos nacieron y se validaron en etapas previas a la educación superior. Eso no significa que no sirvan, pero sí que no pueden trasladarse sin más. Los primeros semestres de ingeniería tienen sus propias exigencias cognitivas y disciplinares, y lo que funciona con niños o adolescentes no necesariamente funciona igual con estudiantes universitarios que ya traen expectativas y marcos de referencia distintos. Ahí está la pregunta que la investigación todavía no ha respondido bien: no si estos modelos son útiles, sino cómo habría que adaptarlos para que lo sean en ese contexto específico.

5.6 *Discusión de las brechas y retos identificados (PI-6)*

Los vacíos que deja la literatura no son menores ni fáciles de resolver. La escasez de estudios longitudinales sigue siendo uno de los más críticos: sin ese tipo de evidencia, es difícil saber si los efectos del pensamiento computacional sobre las competencias STEM perduran o simplemente se evaporan al terminar la intervención. A eso se suman la poca atención que se le ha dado a la formación docente específica para pensamiento computacional y la tendencia generalizada a ignorar variables de diversidad y equidad en el diseño de las propuestas.

Hay además un sesgo geográfico que merece atención. La producción académica sobre integración pensamiento computacional-STEM se concentra en contextos norteamericanos, europeos y asiáticos; América Latina aparece en los márgenes. Eso es problemático porque los sistemas educativos, las condiciones curriculares y los contextos socioeconómicos de la región tienen particularidades propias que no pueden simplemente asumirse como equivalentes a los de otras latitudes. Países como Colombia necesitan investigación situada, no solo modelos importados, para construir propuestas de integración pensamiento computacional –STEM que respondan a sus realidades concretas.

En particular, Novoa y otros (2025), aportan evidencia relevante en ese sentido: sus hallazgos muestran que los estudiantes universitarios presentan dificultades marcadas en abstracción, reconocimiento de patrones y formulación de algoritmos, lo que refuerza la urgencia de trabajar estas competencias desde el comienzo de la formación [36]. A eso se suma que casi no existen adaptaciones curriculares pensadas específicamente para el primer semestre de ingeniería, lo que revela una brecha entre el reconocimiento teórico de la importancia del pensamiento computacional y su incorporación efectiva en los planes de estudio. Cerrar esa brecha no es solo una tarea académica; también es una oportunidad para que las

políticas educativas tomen decisiones más informadas sobre la formación inicial en ingeniería.

5.7 *Implicaciones para la educación en ingeniería*

Los resultados del mapeo sugieren que la integración del pensamiento computacional y las competencias STEM en los primeros semestres de ingeniería requiere un enfoque intencional, sistémico y curricular, que involucre no solo al estudiante, sino también al docente y al diseño del currículo. Las metodologías activas y el uso de tecnologías educativas emergen como estrategias prometedoras, siempre que se articulen con objetivos formativos claros y con procesos de evaluación coherentes.

5.8 *Limitaciones del estudio*

Como todo mapeo sistemático, este trabajo tiene límites que conviene tener presentes al leer sus resultados.

El primero es de cobertura: la búsqueda se restringió a bases de datos específicas y a un período determinado, por lo que es posible que investigaciones relevantes publicadas en otros idiomas o indexadas en fuentes no consultadas hayan quedado fuera.

El segundo tiene que ver con la heterogeneidad de los estudios incluidos: las diferencias en enfoques, contextos y formas de reportar resultados hacen que las comparaciones directas entre propuestas sean, en el mejor de los casos, aproximadas.

Y el tercero es inherente al tipo de revisión: un mapeo ofrece una fotografía amplia del campo, útil para identificar tendencias y vacíos, pero no está diseñado para juzgar en profundidad la efectividad de intervenciones concretas.

5.9 *Proyecciones de investigación futura*

Los hallazgos de este trabajo apuntan a varias direcciones que vale la pena explorar. La más urgente es quizás la de los estudios longitudinales: entender cómo evoluciona el pensamiento computacional y su relación con las competencias STEM a lo largo de la formación en ingeniería, y no solo en el corto plazo de una intervención, sigue siendo una deuda pendiente del campo. En paralelo, hace falta investigación que no se limite a proponer estrategias generales, sino que desarrolle modelos curriculares pensados específicamente para los primeros semestres, donde las bases se construyen y los estudiantes son más vulnerables al abandono. Dos frentes adicionales merecen atención: la evaluación y la formación docente.

Sin instrumentos estandarizados para medir el pensamiento computacional, comparar resultados entre estudios seguirá siendo un ejercicio impreciso; avanzar en ese terreno fortalecería la evidencia empírica del campo. Y sin docentes preparados para integrar el pensamiento computacional en sus prácticas de enseñanza, las mejores propuestas curriculares corren el riesgo de quedarse en el papel.

6 Conclusiones y futuras líneas de investigación

6.1 Conclusiones

Este mapeo sistemático analizó 26 estudios primarios sobre la integración del pensamiento computacional y las competencias STEM en los primeros semestres de la educación en ingeniería. Lo que emerge del conjunto es un campo en movimiento: hay un interés creciente por incorporar el pensamiento computacional como competencia cognitiva transversal en la formación inicial de ingenieros, reconociendo su capacidad para fortalecer la resolución de problemas, el razonamiento lógico y la articulación interdisciplinaria que el enfoque STEM demanda.

Las metodologías activas —aprendizaje basado en problemas y en proyectos, apoyadas en herramientas tecnológicas y entornos digitales— han sido el camino más transitado para implementar esa integración. Pero que muchos estudios recurran a las mismas estrategias no significa que las apliquen de la misma manera: la forma en que se articulan varía considerablemente, y esa disparidad revela algo importante: no existe todavía un modelo pedagógico ampliamente consensado que oriente de forma sistemática la integración CT-STEM en los primeros semestres.

En cuanto a las habilidades específicas, la descomposición, el diseño algorítmico y la modularidad concentran la atención de la literatura, mientras que la abstracción y la generalización siguen siendo un escollo en etapas tempranas. Esto no es un dato menor: indica que las experiencias de aprendizaje necesitan andamiajes pedagógicos progresivos, calibrados a las demandas cognitivas y disciplinares reales de la ingeniería.

Las herramientas tecnológicas y la robótica educativa facilitan la integración del pensamiento computacional en contextos STEM, pero con una condición que los estudios revisados no siempre cumplen: que estén respaldadas por una intención pedagógica clara, coherencia curricular y docentes preparados para usarlas bien. La diversidad de enfoques de evaluación identificados habla de la complejidad del pensamiento computacional como competencia, y también de la urgencia de avanzar hacia metodologías de medición más consistentes que permitan comparar resultados entre estudios.

Hay además un sesgo geográfico que este trabajo ayuda a visibilizar: la producción académica sobre pensamiento computacional y STEM en ingeniería se concentra en contextos norteamericanos, europeos y asiáticos. América Latina, y Colombia en particular, aparecen poco, a pesar de que sus sistemas educativos, condiciones institucionales y desafíos curriculares tienen características propias que no pueden ignorarse. Fortalecer la investigación situada en la región no es solo una cuestión de representación; es una condición para que las propuestas pedagógicas y curriculares que se diseñen sean pertinentes para quienes las necesitan.

Los vacíos que quedan —escasez de estudios longitudinales, formación docente insuficiente, evaluación poco estandarizada, ausencia de adaptaciones curriculares para los primeros semestres— no son solo limitaciones: son una agenda de investigación. Este estudio ofrece una síntesis actualizada del campo y un marco de referencia para quienes,

desde la investigación o desde las instituciones, buscan fortalecer la integración del pensamiento computacional y las competencias STEM en la formación inicial de ingenieros, con los pies puestos en la realidad latinoamericana.

6.2 Resumen de los hallazgos más importantes

El recorrido por la literatura deja varios puntos claros. El primero es metodológico: el aprendizaje basado en problemas, el aprendizaje basado en proyectos y la gamificación han sido los caminos más utilizados para integrar el pensamiento computacional en la educación STEM, precisamente porque permiten anclar el pensamiento computacional en situaciones propias de la ingeniería y no tratarlo como un contenido aislado. A esto se suma un uso creciente de entornos digitales y herramientas tecnológicas como soporte para la enseñanza del pensamiento computacional, especialmente en las etapas iniciales de la formación universitaria donde el andamiaje es más necesario.

El segundo punto es sobre quién enseña. Los estudios revisados son consistentes en señalar que la formación del profesorado no es un factor secundario: la calidad de cualquier intervención pensamiento computacional-STEM depende en buena medida de las decisiones pedagógicas del docente y de su dominio conceptual del tema. Sin embargo, ese es precisamente uno de los vacíos más notorios que deja la literatura: poco se sabe sobre el impacto sostenido del pensamiento computacional en el desarrollo de competencias STEM a largo plazo, lo que hace difícil evaluar si las intervenciones producen efectos duraderos o solo resultados inmediatos.

Los estudios más recientes no dejan mucha duda sobre hacia dónde va el campo: el pensamiento computacional se ha vuelto una competencia difícil de ignorar en la formación de ingenieros, sobre todo en programas que apuestan por la innovación tecnológica y la resolución de problemas complejos. Integrarlo dentro de los enfoques STEM no es un agregado cosmético; incide en habilidades cognitivas y analíticas que los ingenieros van a necesitar en un mundo donde la transformación digital no da señales de detenerse [13], [10].

6.3 Limitaciones de la revisión

Este trabajo tiene tres limitaciones que conviene reconocer. La primera es temporal: la literatura revisada ofrece muy pocos estudios longitudinales, lo que hace difícil seguirle la pista al impacto del pensamiento computacional sobre las competencias STEM más allá del momento en que termina una intervención. La evidencia disponible es mayormente de corto plazo, y eso deja sin responder preguntas que importan.

La segunda es de contexto: una proporción importante de los estudios analizados se desarrolla en entornos educativos muy específicos, lo que complica extender sus conclusiones a otros escenarios institucionales o culturales. Los hallazgos son válidos dentro de sus propios marcos, pero su transferibilidad es limitada.

La tercera afecta directamente la comparabilidad: las metodologías de evaluación del pensamiento computacional

varían tanto entre estudios que contrastar resultados se vuelve un ejercicio impreciso. Mientras no haya mayor estandarización en cómo se mide el pensamiento computacional, las conclusiones sobre qué estrategias de integración funcionan mejor seguirán siendo tentativas.

6.4 Propuestas para futuras investigaciones

Los hallazgos y limitaciones de este trabajo apuntan a líneas concretas que la investigación debería explorar. La más urgente sigue siendo la de los estudios longitudinales: comprender cómo el pensamiento computacional incide en el desarrollo de competencias STEM a lo largo de toda la formación en ingeniería —y no solo en el tramo que cubre una intervención— es una deuda pendiente del campo que ningún mapeo puede saldar por sí solo.

Una segunda línea igualmente necesaria es la formación docente. Diseñar programas específicamente orientados a preparar a los profesores de ingeniería para enseñar pensamiento computacional no es un lujo; es una condición para que cualquier propuesta curricular tenga posibilidades reales de funcionar en el aula.

Hay además territorios poco explorados que merecen atención. Uno es el de las asignaturas de ingeniería que no tienen relación directa con la programación: si el pensamiento computacional es realmente una competencia transversal, debería poder integrarse en esos espacios también, y vale la pena investigar cómo. Otro es el de la equidad: género, situación socioeconómica y acceso tecnológico son variables que rara vez aparecen en los estudios revisados, pero que condicionan de manera real quién se beneficia y quién queda fuera de los enfoques pensamiento computacional-STEM.

Finalmente, el campo necesita ponerse de acuerdo en cómo medir el pensamiento computacional. Sin metodologías de evaluación estandarizadas, cada estudio habla su propio idioma y la evidencia acumulada pierde fuerza. Avanzar en esa dirección no es solo una cuestión técnica; es lo que permitiría construir conclusiones más sólidas sobre qué funciona, en qué contextos y para quién.

Referencias

- [1] S. Atmatzidou y S. Demetriadis, «How to Support Students' Computational Thinking Skills in Educational Robotics Activities», jul. 2014. doi: 10.1016/j.robot.2015.10.008.
- [2] J. Wing, «Computational Thinking», *Commun. ACM*, vol. 49, pp. 33-35, mar. 2006, doi: 10.1145/1118178.1118215.
- [3] J. Wing, «Computational thinking's influence on research and education for all», *Ital. J. Educ. Technol.*, vol. 25, n.º 2, pp. 7-14, 2017.
- [4] C. Wang, J. Shen, y J. Chao, «Integrating Computational Thinking in STEM Education: A Literature Review», *Int. J. Sci. Math. Educ.*, nov. 2021, doi: 10.1007/s10763-021-10227-5.
- [5] P. Sengupta et al., «Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework», *Educ. Inf. Technol.*, vol. 18, pp. 351-380, jun. 2013, doi: 10.1007/s10639-012-9240-x.
- [6] M. Hacker, «Integrating Computational Thinking into Technology and Engineering Education», *Technol. Eng. Teach.*, vol. 77, n.º 4, pp. 8-14, 2018.
- [7] J. Wu, Y. Wang, H. Kong, y L. Zhu, «How to Cultivate Computational Thinking-Enabled Engineers: A Case Study on the Robotics Class of Zhejiang University», presentado en 2019 ASEE Annual Conference & Exposition, jun. 2019. doi: 10.18260/1-2--32901.
- [8] T. R. Kelley y J. G. Knowles, «A Conceptual Framework for Integrated STEM Education», *Int. J. STEM Educ.*, vol. 3, 2016, doi: 10.1186/s40594-016-0046-z.
- [9] C. Mohtadi, M. Kim, y J. Schlosser, «Why integrate computational thinking into a 21 st century engineering curriculum?», ene. 2013. doi: 10.1109/educon.2014.6826132.
- [10] Y. M. Gonzales-Tito y otros, «Inteligencia artificial y pensamiento computacional en estudiantes de una universidad privada de Lima (Perú)», *Form. Univ.*, vol. 18, n.º 3, pp. 1-10, jun. 2025, doi: 10.4067/s0718-50062025000300001.
- [11] I. Mejía, J. A. Hurtado, R. F. Z. Muñoz, y B. G. S. España, «Robótica educativa como herramienta para el desarrollo del pensamiento computacional. Una revisión de la literatura», *Rev. Educ. En Ing.*, vol. 17, n.º 33, Art. n.º 33, ene. 2022, doi: 10.26507/rei.v17n33.1216.
- [12] P. Gómez A. y Á. Gómez N., «Integrando el desarrollo del pensamiento computacional y los conceptos T&E STEM en el enfoque metodológico de ChildProgramming», feb. 2023, Accedido: 20 de junio de 2023. [En línea]. Disponible en: <http://repositorio.unicauca.edu.co:8080/xmlui/handle/123456789/6279>
- [13] B. Palop, I. Díaz, L. J. Rodríguez-Muñiz, y J. J. Santaengracia, «Redefining computational thinking: A holistic framework and its implications for K-12 education», *Educ. Inf. Technol.*, vol. 30, n.º 10, pp. 13385-13410, jul. 2025, doi: 10.1007/s10639-024-13297-4.
- [14] M. Lodi y S. Martini, «Computational Thinking, Between Papert and Wing», *Sci. Educ.*, vol. 30, n.º 4, pp. 883-908, ago. 2021, doi: 10.1007/s11191-021-00202-5.
- [15] R. R. C. Masaquiza, K. L. R. Arce, D. I. L. Pozo, y O. I. P. Gualoto, «Desarrollo de habilidades del siglo XXI a través de la educación STEM», *Rev. Imaginario Soc.*, vol. 7, n.º 2, abr. 2024, doi: 10.59155/is.v7i2.191.
- [16] M. J. Page et al., «Declaración PRISMA 2020: una guía actualizada para la publicación de revisiones sistemáticas», *Rev. Esp. Cardiol.*, vol. 74, n.º 9, pp. 790-799, sep. 2021, doi: 10.1016/j.recesp.2021.06.016.
- [17] D. Yang, Y. Baek, Y.-H. Ching, S. Swanson, B. Chittoori, y S. Wang, «Infusing Computational Thinking in an Integrated STEM Curriculum: User Reactions and Lessons Learned», *Eur. J. STEM Educ.*, vol. 6, n.º 1, 2021, doi: 10.20897/ejsteme/9560.
- [18] E. S. Ludeña, «La educación STEAM y la cultura "maker"», *Rev. Padres Maest. J. Parents Teach.*, n.º 379, pp. 45-51, 2019, doi: 10.14422/pym.i379.y2019.008.
- [19] D. D. Yepes Miranda, «Stem y sus oportunidades en el ámbito educativo», jun. 2020, Accedido: 2 de noviembre de 2022. [En línea]. Disponible en: <https://repositorio.unicordoba.edu.co/handle/ucordoba/2774>
- [20] E. F. Crawley, A. «Peko» Hosoi, y A. «Babi» Mitra, «Redesigning Undergraduate Engineering Education at MIT – the New Engineering Education Transformation (NEET) initiative», presentado en 2018 ASEE Annual Conference & Exposition, jun. 2018. doi: 10.18260/1-2--30923.
- [21] S. Atmatzidou y S. Demetriadis, «Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences», *Robot. Auton. Syst.*, vol. 75, pp. 661-670, ene. 2016, doi: 10.1016/j.robot.2015.10.008.
- [22] H. Shoaib y S. P. Brophy, «A Literature-based Perspective Towards Learning and Pedagogy of Computational Thinking», presentado en 2020 ASEE Virtual Annual Conference Content Access, jun. 2020. doi: 10.18260/1-2--34011.
- [23] M. A. Cannady, M. A. Collins, T. Hurt, R. Montgomery, E. Greenwald, y R. Dorph, «Computational Thinking for Science Positions Youth to Be Better Science Learners», *Educ. Sci.*, vol. 15, n.º 1, Art. n.º 1, ene. 2025, doi: 10.3390/educsci15010105.
- [24] T. Djambong y V. Freiman, *Task-Based Assessment of Students' Computational Thinking Skills Developed through Visual Programming or Tangible Coding Environments*. International Association for the Development of the Information Society, 2016. Accedido: 5 de mayo de 2022. [En línea]. Disponible en: <https://eric.ed.gov/?id=ED571389>
- [25] V. Dolgoplovas, T. Jevsikova, L. Savulionienė, y V. Dagiene, «On Evaluation of Computational Thinking of Software Engineering Novice Students», jul. 2015. doi: 10.13140/RG.2.1.2855.9206.

- [26] M. Román-González, J.-C. Pérez-González, y C. Jiménez-Fernández, «Test de Pensamiento Computacional: diseño y psicometría general [Computational Thinking Test: design & general psychometry]», oct. 2015. doi: 10.13140/RG.2.1.3056.5521.
- [27] J. Hurtado, C. C. Alberto, C. S. Tatiana, y R. O. Eduardo, «Child Programming: Una Estrategia de Aprendizaje y Construcción de Software Basada en la Lúdica, la Colaboración y la Agilidad», 2012.
- [28] A. M. Chimunja, C. A. Collazos, y J. A. Hurtado, «ChildProgramming-C: como una mejora de la dimensión colaborativa del modelo ChildProgramming», *Entre Cienc. E Ing.*, vol. 11, n.º 22, Art. n.º 22, 2017, doi: 10.31908/19098367.3551.
- [29] R. F. Zuñiga, J. A. H. Alegría, y C. A. C. Ordoñez, «Comprendiendo los procesos de abstracción computacional en los niños: un estudio de caso exploratorio», *T C-Rev. Investig. Tecnol. Cienc.*, vol. 1, n.º 8, pp. 57-66, 2014.
- [30] H. F. O. Muñoz, A. García, J. A. H. Alegría, y C. A. Collazos, «Analizando y aplicando la gamificación en el proceso ChildProgramming», *Rev. Colomb. Comput.*, vol. 14, n.º 2, pp. 7-23, 2013.
- [31] C. A. Manzano Quiñonez y J. C. Moreno Vásquez, «Child Gender : Extendiendo child programming con aspectos de diversidad de género», sep. 2017, Accedido: 20 de mayo de 2022. [En línea]. Disponible en: <http://repositorio.unicauca.edu.co:8080/xmlui/handle/123456789/1745>
- [32] J. A. Hurtado Alegría, V. Y. Gómez Calvache, y A. C. Zambrano Lasso, «Estudiando el modelo ChildProgramming-G para encontrar elementos que permitan desarrollar un sistema de memoria transactiva», *Campus Virtuales Rev. Científica Iberoam. Tecnol. Educ.*, 2017, doi: 10/8.pdf.
- [33] R. Zuñiga, J. Hurtado, C. Collazos, y H. Fardoun, «ChildProgramming Evolution, A Method to Increase the Computational Thinking Skills in School: 4th Iberoamerican Workshop, HCI-Collab 2018, Popayán, Colombia, April 23–27, 2018, Revised Selected Papers», 2019, pp. 57-69. doi: 10.1007/978-3-030-05270-6_5.
- [34] M. Tenorio Melenje María, M. A. Trujillo, J. A. Hurtado Alegría, y C. Collazos, «Debugging Block-Based Programs», en *Human-Computer Interaction*, V. Agredo-Delgado y P. H. Ruiz, Eds., en Communications in Computer and Information Science. Cham: Springer International Publishing, 2019, pp. 98-112. doi: 10.1007/978-3-030-05270-6_8.
- [35] J. Muñoz, «Block-Based Design: Un Método Para El Diseño de Programas Basados en Bloques», Trabajo de grado, Unicauca, 2022.
- [36] M. G. G. Novoa, M. del S. Pérez. Alcalá y R. M. Gamboa, «Habilidades identificadas de pensamiento computacional, programación y motivación académica en estudiantes de pregrado», *Transregiones*, n.º 9, pp. 13-40, feb. 2025.

J. A. Hurtado Alegría es Doctor en ciencias de la computación, Especialista en Redes y Servicios Telemáticos, Especialista en procesos para el desarrollo de Software, Ingeniero en electrónica y telecomunicaciones. Actualmente se desempeña como docente titular del Departamento de Sistemas de la Universidad del Cauca. ORCID: <https://orcid.org/0000-0002-2508-0962>

P. S. Gómez, es estudiante del Doctorado en Ciencias de la Electrónica, Universidad del Cauca, Ciclo I, Magíster en Computación, Ingeniero en Electrónica y Telecomunicaciones. Actualmente se desempeña como docente becario en la Universidad del Cauca. ORCID: <https://orcid.org/0009-0007-5382-2488>

7 Anexos

Anexo A. Caracterización de los estudios incluidos en el mapeo sistemático.

Autor(es)	Año	Nombre del Estudio	Metodología	Contexto	Principales Hallazgos
Wang et al.	2021	Integrating Computational Thinking in STEM Education: A Literature Review	Revisión semi-sistemática	Integración CT en STEM	Metodologías activas como aprendizaje basado en problemas y juegos
Yang et al.	2021	Infusing Computational Thinking in an Integrated STEM Curriculum: User Reactions and Lessons Learned	Estudio de implementación	Educación STEM	Evaluación positiva de programas CT en proyectos STEM
Mohtadi et al.	2013	Why integrate computational thinking into a 21st century engineering curriculum?	Análisis curricular	Ingeniería	Integración CT en ingeniería mejora resolución de problemas
Sánchez Ludeña	2019	La educación STEAM y la cultura “maker”	Revisión teórica	Educación STEM	Destaca la importancia del constructivismo en STEM y el aprendizaje por proyectos
Yepes Miranda	2020	STEM y sus oportunidades en el ámbito educativo	Revisión sistemática	Educación STEM	Identifica experiencias significativas en el enfoque STEM
Cannady et al.	2025	Computational Thinking for Science Positions Youth to Be Better Science Learners	Estudio empírico con 600 estudiantes	Educación secundaria	Correlación positiva entre CT y desempeño en ciencias
Zhou et al.	2024	Effect of Different Flow Design Approaches on Undergraduates’ Computational Thinking During Pair Programming	Análisis experimental	Educación en ingeniería	Impacto del andamiaje pedagógico en el desarrollo de CT
Wu et al.	2019	How to Cultivate Computational Thinking-Enabled Engineers	Revisión sistemática y entrevistas	Educación en ingeniería	Marco para alfabetización digital y modelado en ingeniería
Hacker	2018	Integrating Computational Thinking into Technology and Engineering Education	Encuesta a educadores	Educación en ingeniería	Modelo de instrucción basado en diseño informado
Shoaib y Brophy	2020	A Literature-based Perspective Towards Learning and Pedagogy of Computational Thinking	Revisión de literatura	Educación STEM	Estrategias pedagógicas para mejorar la enseñanza del CT
Gross et al.	2014	Fostering computational thinking in engineering education	Revisión bibliográfica	Educación en ingeniería	Buenas prácticas de integración de CT en planes de estudio
Miller et al.	2014	Integrating computational and creative thinking to improve learning and performance in CS1	Experimentos educativos	Ciencias de la computación	Relación entre pensamiento creativo y CT
Vergara et al.	2009	Aligning Computing Education with engineering workforce computational needs	Análisis curricular	Ingeniería	Alineación de educación informática con necesidades de la industria
Crawley et al.	2018	Redesigning Undergraduate Engineering Education at MIT – NEET initiative	Transformación curricular	Educación en ingeniería	Modelo NEET para innovación curricular
Atmatzidou y Demetriadis	2014	How to Support Students’ Computational Thinking Skills in Educational Robotics Activities	Investigación experimental	Robótica educativa	Desarrollo de habilidades de CT mediante actividades prácticas
Atmatzidou y Demetriadis	2016	A Didactical Model for Educational Robotics Activities	Seguimiento longitudinal	Educación en CT	Edad y género no influyen significativamente en el desarrollo de CT
Djambong y Freiman	2016	Task-Based Assessment of Students’ Computational Thinking Skills	Estudio de caso	Evaluación de CT	Evaluación de CT mediante programación visual
Dolgopolas et al.	2015	On Evaluation of Computational Thinking of Software Engineering Novice Students	Evaluación experimental	Ingeniería	Uso del concurso Bebras para evaluar CT
Román-González	2016	Test de Pensamiento Computacional	Desarrollo de instrumento	Educación en CT	Creación de test psicométrico para medir CT
Hurtado et al.	2012	ChildProgramming	Desarrollo metodológico	Educación en CT	Modelo pedagógico basado en colaboración y aprendizaje lúdico
Gómez et al.	2023	Integrando el desarrollo del pensamiento computacional y los conceptos T&E STEM	Estudio de caso	Modelo CT4E	Extensión del modelo ChildProgramming para ingeniería
Zúñiga et al.	2018	Comprendiendo los procesos de abstracción computacional en los niños	Estudio exploratorio	ChildProgramming	Impacto de ChildProgramming en el desarrollo de CT
Masaquiza et al.	2024	Desarrollo de habilidades del siglo XXI a través del enfoque STEM	Estudio educativo	Educación STEM	STEM fortalece pensamiento crítico, creatividad y resolución de problemas
Gonzales-Ttito et al.	2025	Inteligencia artificial y pensamiento computacional en la educación	Análisis conceptual	Educación tecnológica	Relación entre CT, inteligencia artificial y transformación digital
Novoa et al.	2025	Habilidades identificadas de pensamiento computacional en estudiantes universitarios	Estudio empírico	Educación superior	Dificultades en abstracción, secuencias y ciclos en estudiantes
Palop et al.	2025	Redefining Computational Thinking: A Holistic Framework	Marco conceptual	Educación en CT	Propone un marco holístico de CT basado en problemas, algoritmos y datos
Lodi y Martini	2021	Computational Thinking: Between Papert and Wing	Revisión teórica	Educación en CT	Análisis conceptual del desarrollo histórico del CT

Fuente: Elaboración propia.