Usabilidad y aportes al aprendizaje de las metáforas de interfaz basadas en pensamiento algorítmico: una revisión sistemática de literatura

Usability and learning contributions of interface metaphors based on algorithmic thinking: a systematic literature review

MSc. Deixy Ximena Ramos Rivadeneira[1], **PhD. Javier Alejandro Jiménez Toledo**[1]

[1] **Universidad Centro de Estudios Superiores María Goretti (CESMAG),** Facultad de Ingeniería, Programa de Ingeniería de Sistemas, Grupo de Investigación Tecnofilia y Grupo de investigación Luca Paccioli, San Juan de Pasto, Nariño, Colombia.

Correspondencia: dxramos@unicesmag.edu.co

Recibido: 07 octubre 2026. **Revisado:** 02 noviembre 2026. **Aceptado:** 16 noviembre 2026. **Publicado:** 23 noviembre 2026.

Cómo citar: D. X. Ramos Rivadeneira y J. A. Jiménez Toledo, "Usabilidad y aportes al aprendizaje de las metáforas de interfaz basadas en pensamiento algorítmico: una revisión sistemática de literatura", *Revista Colombiana de Tecnologías de Avanzada (RCTA)*, vol. 2, no. 48, pp. 296–309, nov. 2026, doi: 10.24054/rcta.v2i48.4724

Esta obra está bajo una licencia internacional
Creative Commons Atribución-NoComercial 4.0.

Resumen: Las metáforas de interfaz, articuladas con el pensamiento algorítmico, pueden hacer visibles procesos abstractos durante el aprendizaje. Este artículo presenta una revisión sistemática de literatura, orientada por SEGRESS y reportada conforme a PRISMA 2020, cuyo objetivo fue analizar las características, las formas de evaluación de usabilidad y los aportes y limitaciones reportados de las metáforas de interfaz basadas en pensamiento algorítmico para el aprendizaje. Se consultaron Scopus, Web of Science, IEEE Xplore, ScienceDirect y Redalyc (2019–2026) y se sintetizaron 21 estudios. Los resultados caracterizan estas interfaces mediante cinco componentes (representación visual, organización secuencial, interacción directa, retroalimentación y correspondencia metáfora–procedimiento), identifican evaluaciones de usabilidad basadas en instrumentos estandarizados, observación, registros de interacción y evaluación heurística, y muestran aportes en la exteriorización del razonamiento y en habilidades de pensamiento computacional, con limitaciones asociadas a la complejidad visual y al predominio de estudios en educación infantil, primaria y secundaria.

Palabras clave: metáforas de interfaz, pensamiento algorítmico, usabilidad, tecnologías educativas, revisión sistemática.

Abstract: Interface metaphors, articulated with algorithmic thinking, can make abstract processes visible during learning. This article presents a systematic literature review, guided by SEGRESS and reported according to PRISMA 2020, whose objective was to analyse the characteristics, the usability evaluation approaches, and the reported contributions and limitations of interface metaphors based on algorithmic thinking for learning. Scopus, Web of Science, IEEE Xplore, ScienceDirect and Redalyc were searched (2019–2026) and 21 studies were synthesised. The results characterise these interfaces through five components (visual representation, sequential organisation, direct interaction, feedback, and metaphor–procedure correspondence), identify usability evaluation based on

standardised instruments, observation, interaction logs and heuristic evaluation, and show contributions to the externalisation of reasoning and to computational thinking skills, with limitations related to visual complexity and to the predominance of studies in early childhood, primary and secondary education.

Keywords: interface metaphors, algorithmic thinking, usability, educational technology, systematic review.

1. INTRODUCCIÓN

La incorporación de tecnologías digitales en los procesos educativos ha ampliado las posibilidades de interacción entre los estudiantes y los contenidos de aprendizaje. Sin embargo, disponer de una herramienta tecnológica no garantiza por sí mismo una experiencia adecuada, ya que su aprovechamiento también depende de la facilidad con que los usuarios puedan comprenderla, navegar por sus componentes y realizar las tareas propuestas. En este contexto, la usabilidad adquiere especial relevancia en el diseño de tecnologías educativas, al considerar aspectos como la efectividad, la eficiencia y la satisfacción durante la interacción con recursos y entornos digitales [1].

Dentro de la Interacción Humano-Computador, las metáforas de interfaz constituyen un recurso de diseño que permite relacionar elementos y acciones de un sistema digital con conceptos previamente conocidos por los usuarios. Esta asociación puede facilitar la construcción de modelos mentales y favorecer la comprensión de la forma en que se organiza y utiliza una interfaz; sin embargo, su efectividad depende de factores como las características del usuario, la complejidad de las tareas y la correspondencia entre la metáfora empleada y el funcionamiento del sistema. Por ello, una metáfora no debe valorarse únicamente por su representación visual, sino también por la manera en que contribuye a una interacción comprensible y usable [2].

El pensamiento algorítmico adquiere especial importancia en los procesos de aprendizaje que requieren organizar información, establecer secuencias y plantear procedimientos para resolver problemas. Su desarrollo implica habilidades como la descomposición, la abstracción, la algoritmización y la depuración, que permiten al estudiante estructurar progresivamente una solución y comprender las relaciones existentes entre sus diferentes etapas. En escenarios educativos, estas habilidades pueden favorecer una aproximación más organizada a problemas que demandan razonamiento lógico y construcción de procedimientos, por lo que su incorporación trasciende el aprendizaje exclusivo de la programación [3].

La integración del pensamiento algorítmico con metáforas de interfaz plantea, por tanto, una oportunidad para representar procesos abstractos mediante estructuras visuales e interactivas que faciliten su comprensión durante el aprendizaje. No obstante, el valor educativo de estas representaciones no depende únicamente de incorporar elementos visuales o secuencias algorítmicas, sino también de que la interfaz permita al estudiante reconocer su funcionamiento, anticipar las acciones posibles y desarrollar las tareas sin que la interacción con la tecnología se convierta en una dificultad adicional. Aunque la relación entre metáforas de interfaz y aprendizaje ha sido estudiada desde hace varias décadas, la literatura continúa mostrando la necesidad de considerar conjuntamente el diseño de la metáfora, las características de los usuarios y la usabilidad de la interacción en tecnologías educativas [4].

1.1 Delimitación conceptual

Dado que el objeto de la revisión se sitúa en la intersección de varios campos, resulta necesario precisar los conceptos que orientan la selección y el análisis de los estudios. En primer lugar, una *metáfora de interfaz* es un dispositivo de diseño que presenta las funciones de un sistema en términos de un dominio ya conocido por el usuario (por ejemplo, el escritorio, la pieza de rompecabezas o el diagrama de flujo), de modo que el conocimiento previo se transfiere al nuevo dominio. Esta noción se fundamenta en la teoría de la metáfora conceptual, según la cual la comprensión de dominios abstractos se estructura a partir de dominios concretos de la experiencia [5], y en la perspectiva del diseño centrado en el usuario, para la cual una buena metáfora proporciona un modelo conceptual coherente que hace visibles las posibilidades de acción (*affordances*) y la correspondencia entre controles y efectos [6]. En segundo lugar, las

interfaces visuales de programación (por bloques, tangibles o basadas en diagramas) son implementaciones concretas de metáforas de interfaz: el bloque encajable, el objeto físico manipulable o el diagrama de flujo son metáforas que representan instrucciones, secuencias y estructuras de control. Por esta razón, en esta revisión los estudios sobre programación por bloques, programación tangible y visualización de algoritmos se consideran pertinentes cuando describen y evalúan la representación mediante la cual el estudiante construye o sigue un procedimiento, y no cuando se limitan a medir resultados de aprendizaje sin caracterizar la interfaz.

En tercer lugar, el *pensamiento computacional* se entiende como el conjunto de procesos de formulación y resolución de problemas que recurren a conceptos fundamentales de la computación, tales como abstracción, descomposición y automatización [7]; su estudio en contextos educativos ha sido operacionalizado mediante conceptos (secuencias, bucles, condicionales), prácticas (ser incremental, probar y depurar, reutilizar) y perspectivas computacionales [8]. El *pensamiento algorítmico* constituye un componente del pensamiento computacional centrado en la construcción, representación y ejecución ordenada de secuencias de pasos para resolver un problema [3]. Dado que la mayor parte de la literatura empírica utiliza el término pensamiento computacional, ambos descriptores se incluyeron en la estrategia de búsqueda, y en el análisis se privilegiaron las dimensiones algorítmicas (secuenciación, diseño de algoritmos, depuración). Finalmente, la *usabilidad* se define como el grado en que un sistema puede ser usado por usuarios específicos para alcanzar objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso determinado [9], y su evaluación recurre tanto a métodos de inspección, como la evaluación heurística, como a pruebas con usuarios e instrumentos estandarizados [10].

1.2 Objetivo de la revisión

En este escenario, resulta pertinente sistematizar la evidencia disponible para comprender cómo se ha abordado la usabilidad de las metáforas de interfaz vinculadas con el pensamiento algorítmico en contextos de aprendizaje. La literatura se encuentra distribuida entre estudios de Interacción Humano-Computador, diseño de interfaces y tecnologías educativas, lo que dificulta reconocer de manera integrada las características de estas metáforas, los criterios utilizados para evaluar su usabilidad y los aportes reportados en los procesos de aprendizaje. Por esta razón, el presente estudio tiene como objetivo analizar, mediante una revisión sistemática de literatura, la usabilidad y los aportes al aprendizaje de las metáforas de interfaz basadas en pensamiento algorítmico, identificando sus características, formas de evaluación y principales aportes y limitaciones reportados en la literatura científica. El artículo se organiza de la siguiente manera: la sección 2 describe el método; la sección 3 presenta los resultados organizados por pregunta de investigación; la sección 4 discute los hallazgos frente a la literatura previa y expone las limitaciones del estudio; y la sección 5 presenta las conclusiones.

2. METODOLOGÍA

La investigación se desarrolló mediante una revisión sistemática de literatura orientada a identificar y analizar estudios relacionados con la usabilidad de metáforas de interfaz basadas en pensamiento algorítmico para el aprendizaje. Para la planificación y la ejecución de la revisión se siguieron las directrices SEGRESS (*Software Engineering Guidelines for REporting Secondary Studies*) [11], que adaptan las recomendaciones PRISMA al ámbito de la ingeniería de software y de los sistemas interactivos. En concreto, SEGRESS se aplicó como lista de verificación para estructurar y reportar las siguientes etapas: formulación de preguntas de investigación, definición de la estrategia de búsqueda (fuentes, cadena y fechas), criterios de elegibilidad, procedimiento de selección, evaluación de calidad, extracción de datos y método de síntesis. El proceso de selección se reporta, además, mediante el diagrama de flujo PRISMA 2020 [12] (ver Fig. 1).

2.1 Preguntas de Investigación

Como punto de partida se definieron las preguntas de investigación, procurando establecer correspondencia entre el propósito de la revisión, la estrategia de búsqueda y las categorías que posteriormente orientarían la extracción y síntesis de la información. Las preguntas se formularon alrededor de tres componentes centrales del estudio: las características de las metáforas de interfaz basadas en pensamiento algorítmico, los mecanismos empleados para evaluar su usabilidad y los aportes y limitaciones reportados en contextos de aprendizaje (Tabla 1). Para reducir el solapamiento entre RQ2 y RQ3, RQ2 se circunscribe a los *métodos, criterios e instrumentos de evaluación de usabilidad y experiencia de usuario*, mientras que RQ3 se circunscribe a los

resultados de aprendizaje y a las *limitaciones* reportadas.

Tabla 1: *Preguntas de investigación de la revisión sistemática*

Código	Pregunta de investigación
RQ1	¿Qué características presentan las metáforas de interfaz basadas en pensamiento algorítmico utilizadas para el aprendizaje?
RQ2	¿Qué métodos, criterios e instrumentos se utilizan para evaluar la usabilidad y la experiencia de usuario de estas metáforas de interfaz?
RQ3	¿Qué aportes al aprendizaje y qué limitaciones se reportan sobre el uso de estas metáforas en los procesos de aprendizaje?

Fuente: elaboración propia.

2.2 Estrategia de Búsqueda

Una vez definidas las preguntas de investigación, se estableció la estrategia de búsqueda a partir de los conceptos centrales del estudio: metáforas de interfaz, pensamiento algorítmico, usabilidad y aprendizaje. Para ampliar la recuperación de publicaciones relacionadas, se incorporaron términos equivalentes asociados con pensamiento computacional, experiencia de usuario, interacción y educación (Tabla 2). Los términos se formularon en inglés y se combinaron mediante los operadores booleanos AND y OR, considerando su presencia en título, resumen y palabras clave.

Tabla 2: *Descriptores utilizados en la estrategia de búsqueda*

Componente	Descriptores
Metáforas de interfaz	"interface metaphor", "interface metaphors", "visual metaphor"
Pensamiento algorítmico	"algorithmic thinking", "computational thinking"
Usabilidad	"usability", "user experience", "interaction"
Aprendizaje	"learning", "education", "teaching"

Fuente: elaboración propia.

A partir de estos descriptores, se construyó la cadena de búsqueda genérica:

("interface metaphor" OR "interface metaphors" OR "visual metaphor") AND ("algorithmic thinking" OR "computational thinking") AND ("usability" OR "user experience" OR "interaction") AND ("learning" OR "education" OR "teaching")

La combinación de los cuatro bloques mediante AND se adoptó de manera deliberada para exigir la coocurrencia de los cuatro conceptos que definen el objeto de la revisión y, con ello, reducir la recuperación de estudios sobre pensamiento computacional o sobre usabilidad que no abordan metáforas de interfaz. Se reconoce que esta decisión favorece la precisión sobre la exhaustividad; para compensarla, en los bloques de usabilidad y aprendizaje se incluyeron términos amplios ("interaction", "education"), cuyo ruido fue controlado en la fase de cribado por título y resumen, y se incluyó el descriptor "visual metaphor", de uso frecuente en la literatura educativa. La Tabla 3 presenta la adaptación de la estrategia de búsqueda a las características de cada base de datos, especificando las cadenas utilizadas, los campos consultados y las fechas de ejecución correspondientes a septiembre de 2026.

Tabla 3: *Adaptación de la estrategia de búsqueda por base de datos*

Base de datos	Cadena / sintaxis	Campos	Fecha
Scopus	TITLE-ABS-KEY(("interface metaphor" OR "interface metaphors" OR "visual metaphor") AND ("algorithmic thinking" OR "computational thinking") AND ("usability" OR "user experience" OR "interaction") AND ("learning" OR "education" OR "teaching")) AND PUBYEAR > 2018	Título, resumen, palabras clave	02/09/2026
Web of Science	TS=(("interface metaphor*" OR "visual metaphor") AND ("algorithmic thinking" OR "computational thinking") AND ("usability" OR "user experience" OR "interaction") AND ("learning" OR "education" OR "teaching")), 2019–2026	Tema (título, resumen, palabras clave)	04/09/2026
IEEE Xplore	Cadena genérica en "All Metadata", 2019–2026	Metadatos	08/09/2026
ScienceDirect	Cadena genérica en "Title, abstract or author-specified keywords" (adaptada al límite de ocho operadores booleanos del buscador), 2019–2026	Título, resumen, palabras clave	10/09/2026
Redalyc	Equivalentes en español y portugués de los descriptores: ("metáfora de interfaz" OR "metáforas visuales" OR "metáfora de interface") AND ("pensamiento algorítmico" OR "pensamiento computacional" OR "pensamento computacional") AND ("usabilidad" OR	Búsqueda general	14/09/2026

	"usabilidade" OR "experiencia de usuario")

Fuente: elaboración propia.

En cuanto a la cobertura idiomática, la cadena principal se formuló en inglés porque Scopus, Web of Science, IEEE Xplore y ScienceDirect indexan título, resumen y palabras clave en inglés incluso para artículos publicados en español o portugués, de manera que estos documentos pueden recuperarse mediante descriptores en inglés. Para reducir el sesgo idiomático, Redalyc se consultó con los equivalentes en español y portugués de los descriptores (Tabla 3). Se reconoce, como limitación, que la literatura en español y portugués no indexada con metadatos en inglés pudo quedar subrepresentada (ver sección 4.3).

2.3 Fuentes de Información

La búsqueda de literatura se realizó en Scopus, Web of Science, IEEE Xplore y ScienceDirect, seleccionadas por su cobertura de publicaciones relacionadas con Interacción Humano-Computador, tecnologías educativas, informática y educación. De manera complementaria, se consideró Redalyc con el propósito de ampliar la identificación de producción científica iberoamericana relacionada con el objeto de estudio.

2.4 Criterios de Inclusión y Exclusión

La selección de los estudios se realizó mediante criterios de inclusión y exclusión definidos previamente, con el propósito de conservar publicaciones relacionadas directamente con el objeto de la revisión y reducir la incorporación de documentos que no aportaran evidencia suficiente para responder las preguntas de investigación. Los criterios consideraron la pertinencia temática, el contexto educativo, el tipo de publicación, el periodo de publicación y la disponibilidad del texto completo (Tabla 4).

Tabla 4: *Criterios de inclusión y exclusión de los estudios*

Tipo	Criterios
Inclusión	CI1: estudios publicados entre 2019 y 2026; CI2: artículos científicos o trabajos de congresos revisados por pares; CI3: investigaciones que describen y evalúan una interfaz (metáfora de interfaz, interfaz visual de programación, interfaz tangible o visualización de algoritmos) vinculada con el pensamiento algorítmico o computacional en contextos de aprendizaje; CI4: publicaciones en español, inglés o portugués con texto completo disponible.
Exclusión	CE1: estudios duplicados; CE2: publicaciones sin relación directa con las preguntas de investigación; CE3: documentos sin revisión por pares (editoriales, resúmenes, capítulos no arbitrados); CE4: estudios sobre metáforas sin relación con interfaces digitales; CE5: estudios sobre pensamiento algorítmico o computacional sin componente de interacción (por ejemplo, actividades exclusivamente desconectadas) o que no caracterizan la interfaz utilizada.

Fuente: elaboración propia.

2.5 Proceso de Selección de los Estudios

Los registros recuperados en las bases de datos fueron exportados a un gestor de referencias y depurados mediante la eliminación de duplicados. Posteriormente, se realizó una selección preliminar a partir de la revisión del título, resumen y palabras clave, aplicando los criterios de inclusión y exclusión establecidos. Los estudios que superaron esta etapa fueron revisados a texto completo para determinar su pertinencia frente al objetivo y las preguntas de investigación. Finalmente, los documentos seleccionados fueron sometidos a evaluación de calidad antes de realizar la extracción y síntesis de la información. El cribado por título y resumen y la revisión a texto completo fueron realizados de manera independiente por los dos autores, y las discrepancias identificadas se resolvieron mediante discusión hasta alcanzar un consenso. La Fig. 1 presenta el flujo de identificación, cribado, elegibilidad e inclusión conforme a PRISMA 2020, mientras que la Tabla 5 detalla las cifras correspondientes a cada fuente de información.

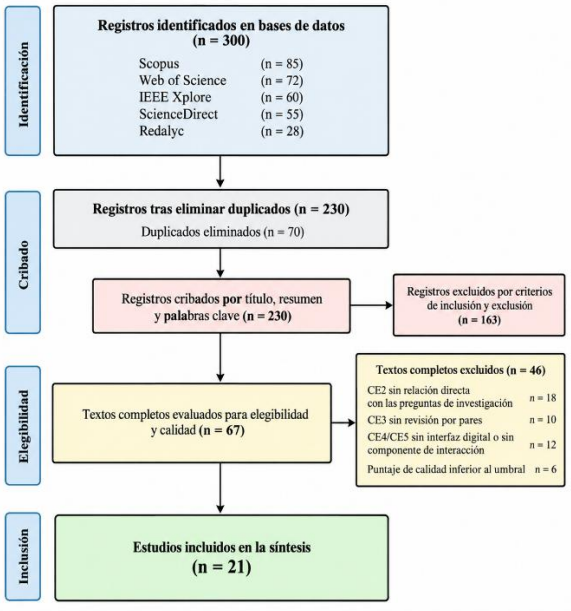

Fig. 1. *Diagrama de flujo PRISMA 2020 del proceso de selección.* ***Fuente:*** *elaboración propia.*

Tabla 5: *Registros recuperados, cribados e incluidos por fuente de información*

Fuente	Registros recuperados	Duplicados eliminados	Registros cribados (título/resumen)	Textos completos evaluados	Estudios incluidos
Scopus	85	15	70	22	8
Web of Science	72	20	52	17	5
IEEE Xplore	60	12	48	13	4
Science Direct	55	18	37	10	3
Redalyc	28	5	23	5	1
Total	300	70	230	67	**21**

Fuente: elaboración propia. Los estudios excluidos en la fase de texto completo deben reportarse con sus razones (CE2 a CE5) en el registro del protocolo; el total de estudios incluidos corresponde a los 21 estudios sintetizados en la Tabla 7.

2.6 Evaluación de Calidad

La calidad de los estudios seleccionados se evaluó mediante criterios relacionados con la claridad del objetivo, descripción metodológica, pertinencia de los resultados y correspondencia del estudio con las preguntas de investigación (Tabla 6). Cada criterio se valoró mediante una escala de tres niveles: 1 (cumple), 0,5 (cumple parcialmente) y 0 (no cumple), con un puntaje máximo de 4 puntos por estudio.

Se estableció un puntaje mínimo de 2,5 sobre 4 para la inclusión en la síntesis, equivalente al 62,5 % del puntaje máximo. Los estudios que obtuvieron una calificación inferior a este umbral fueron excluidos de la revisión. La evaluación fue realizada de manera independiente por los dos autores y las discrepancias se resolvieron mediante consenso.

Tabla 6: *Criterios para la evaluación de calidad*

Código	Criterio de evaluación
EC1	¿El objetivo del estudio está claramente definido?
EC2	¿La metodología se encuentra suficientemente descrita?
EC3	¿El estudio aporta evidencia relacionada con la usabilidad o la interacción con la interfaz?
EC4	¿Los resultados contribuyen a responder alguna de las preguntas de investigación?

Escala de valoración: 1 = cumple; 0,5 = cumple parcialmente; 0 = no cumple. **Fuente:** elaboración propia.

2.7 Extracción y Síntesis de la Información

Los estudios que superaron la evaluación de calidad fueron organizados mediante una matriz de extracción de datos. Para cada publicación se registraron datos bibliográficos, contexto de aplicación, características de la metáfora de interfaz, relación con el pensamiento algorítmico o computacional, criterios de usabilidad evaluados, métodos e instrumentos empleados, principales resultados y limitaciones reportadas. Posteriormente, la información se agrupó de acuerdo con las preguntas de investigación para realizar una síntesis descriptiva y temática de los hallazgos. La Tabla 7 presenta la matriz resumida de los 21 estudios incluidos, que constituye la base de los resultados de la sección 3.

Tabla 7: *Matriz de los estudios incluidos en la síntesis (n = 21)*

Ref.	Estudio (autores, año)	Nivel educativo	Tipo de interfaz / metáfora	Evaluación de usabilidad o experiencia	Aporte al aprendizaje / RQ
[13]	Molina-Ayuso et al., 2022	Formación docente (educación superior)	Programación por bloques (Scratch)	No reportada como objetivo principal	Introducción al pensamiento computacional / RQ1, RQ3
[14]	Gao et al., 2025	Primaria	Bloques con andamiaje de mapas mentales	Percepción de los estudiantes	Pensamiento computacional / RQ1, RQ3
[15]	Moon et al., 2022	Educación superior (estudiantes universitarios de pregrado)	Programación por bloques	Dificultades de uso y comprensión	Conceptos y prácticas difíciles de PC / RQ1, RQ3
[16]	Budiyanto et al., 2025	K-12	Programación visual (Code.org)	Análisis de actividades con marco Brennan –Resnick	Conceptos y prácticas de PC / RQ1
[17]	Pellas, 2025	Educación infantil y otras etapas	Programación tangible	Observación y medidas de desempeño	PC, razonamiento espacial, función ejecutiva / RQ1, RQ3
[18]	Chen et al., 2025	Aula escolar (K-12)	Programación visual	Estudio de usuarios,	Apoyo a proyect

			con apoyo de IA generativa (MindScratch)	cuestionarios	os de programación / RQ1, RQ2
[19]	Lin et al., 2025	Educación superior (gestión)	Metáfora visual en diseño instruccional	Carga cognitiva y percepción	Comprensión del Business Model Canvas / RQ1, RQ3
[20]	Adorni et al., 2024	Escolar (3 a 16 años)	Multi-interfaz: bloques y comandos gestuales (Virtual CAT)	Diseño e implementación; retroalimentación en tiempo real	Evaluación del pensamiento algorítmico / RQ1, RQ2
[21]	Végh et al., 2026	Educación superior (programación introductoria)	Visualización interactiva de algoritmos basada en matrices	Percepción de los estudiantes (cuestionario)	Comprensión de la ejecución de algoritmos / RQ2, RQ3
[22]	Tsai et al., 2025	Aprendices novatos	Programación por bloques	Seguimiento ocular y análisis de secuencias	Comprensión de código / RQ2
[23]	Fu et al., 2025	Primaria	Visualización de algoritmos	Motivación y comportamiento	Desempeño en aprendizaje de algoritmos / RQ2, RQ3
[24]	Gong et al., 2025	Educación secundaria (séptimo grado)	Tangible, bloques y papel y lápiz (comparación)	Carga cognitiva y disfrute	PC y razonamiento espacial / RQ2, RQ3
[25]	Fanchamps et al., 2024	Educación primaria (estudiantes de 10 a 12 años)	Programación SRA con salida tangible o en pantalla	Aspectos de aprendizaje identificados	PC / RQ1, RQ3
[26]	Misirli y Komis, 2023	Educación infantil	Robot tangible programable	Observación de la interacción	Depuración / RQ1, RQ3
[27]	Wu et al., 2023	Aprendices de programación visual	Lenguaje de programación visual	Registros de progreso (trazas de interacción)	Seguimiento del progreso en PC / RQ2
[28]	Piedade y Dorotea, 2023	Primaria (4.º grado)	Scratch	No reportada como objetivo principal	Habilidades de PC / RQ3
[29]	Zhao et al., 2022	Primaria	Mapas mentales en programación visual	No reportada como objetivo principal	Habilidades de PC / RQ3
[30]	Katchapakirin et al., 2022	Educación primaria	Scratch con apoyo conversacional (ScratchThAI)	Evaluación del marco de apoyo	Desarrollo de PC / RQ1, RQ3
[31]	Boom et al., 2022	Educación superior (docentes en formación inicial)	Programación (bloques/texto)	No reportada como objetivo principal	Relación entre PC y calidad de programas / RQ3
[32]	Oyelere et al., 2022	Transversal	Herramientas de apoyo al PC (marco de evaluación)	Marco pedagógico de evaluación	Criterios de evaluación de herramientas / RQ2
[33]	Su et al., 2022	Primaria (niños)	Mapas mentales en Scratch	No reportada como objetivo principal	Pensamiento creativo / RQ3

PC = pensamiento computacional.
Fuente: elaboración propia a partir de los estudios incluidos.

3. RESULTADOS

3.1 Caracterización cuantitativa del corpus

De los 21 estudios incluidos, 7 fueron publicados en 2022, 3 en 2023, 2 en 2024, 8 en 2025 y 1 en 2026, lo que evidencia una concentración reciente de la producción. En cuanto al tipo de interfaz, 12 estudios se centran en entornos de programación por bloques (Scratch, Code.org y variantes con andamiaje o apoyo inteligente), 4 en interfaces tangibles o robots programables, 2 en visualización interactiva de algoritmos, 1 en una plataforma multi-interfaz que combina bloques y comandos gestuales, 1 en una metáfora visual aplicada al diseño instruccional y 1 corresponde a un marco de

evaluación transversal. Respecto al nivel educativo, de los 13 estudios cuyo nivel se identifica en el título o resumen, 2 se sitúan en educación infantil, 5 en educación primaria, 3 en educación secundaria o K-12 en general y 3 en educación superior (formación docente, gestión y programación introductoria); los 8 restantes no explicitan el nivel o son transversales. Esta distribución muestra un predominio de la educación infantil, primaria y secundaria, por lo que los hallazgos que siguen deben leerse principalmente en ese contexto, y su extrapolación a la educación superior requiere evidencia adicional. Finalmente, solo 1 estudio [19] utiliza explícitamente el término *metáfora visual*; los demás describen metáforas de interfaz implícitas (bloques encajables, objetos tangibles, diagramas y visualizaciones), lo que confirma la necesidad de la delimitación conceptual presentada en la sección 1.1.

3.2 RQ1: Características de las metáforas de interfaz basadas en pensamiento algorítmico utilizadas para el aprendizaje

La revisión de los estudios permitió identificar que las interfaces orientadas al desarrollo del pensamiento algorítmico recurren principalmente a representaciones visuales que hacen perceptibles procesos que, en principio, son abstractos. Estas representaciones incluyen bloques gráficos, diagramas de flujo, mapas mentales, animaciones y visualizaciones dinámicas de algoritmos [13], [14], [23]. Su característica común es que la información no se presenta únicamente mediante instrucciones textuales, sino a través de elementos que permiten reconocer relaciones, secuencias y transformaciones durante la ejecución de una tarea. Los estudios sobre visualización de algoritmos muestran precisamente esta transición desde una representación abstracta hacia estructuras gráficas que permiten seguir el comportamiento de los procedimientos y comprender su organización [21], [23].

Una segunda característica corresponde a la organización lógica y secuencial de las acciones. Las interfaces de programación visual permiten construir procedimientos mediante elementos que pueden seleccionarse, desplazarse y conectarse para establecer un orden de ejecución [15]. Esta forma de interacción disminuye la dependencia inicial de la sintaxis textual y hace visible la estructura del procedimiento que está construyendo el estudiante. En Code.org, por ejemplo, las actividades de programación visual permiten trabajar conceptos y prácticas vinculadas con el pensamiento computacional según el marco de Brennan y Resnick [16], mientras que los estudios de seguimiento ocular muestran que la manera en que los aprendices novatos localizan y leen los bloques constituye un componente relevante de la interacción con este tipo de interfaces [22].

También se identificó la interacción directa con las representaciones algorítmicas como una característica relevante. Las interfaces recientes permiten que el estudiante no se limite a observar una representación, sino que construya y modifique procedimientos mediante bloques, acciones de arrastrar y soltar, comandos gestuales u objetos manipulables. Virtual CAT, por ejemplo, combina una interfaz visual de programación basada en bloques con comandos gestuales para construir algoritmos [20], mientras que los entornos de programación tangible trasladan esta interacción hacia objetos físicos [17], [26]. Estas alternativas amplían las formas mediante las cuales los estudiantes pueden representar y manipular una secuencia algorítmica.

Otra característica corresponde a la retroalimentación durante la interacción. Algunas interfaces permiten observar inmediatamente los efectos de las acciones realizadas y proporcionan información sobre el progreso o las dificultades encontradas durante la construcción de una solución. Virtual CAT incorpora retroalimentación en tiempo real que se ajusta a las acciones del usuario [20], mientras que propuestas más recientes, como MindScratch, incorporan inteligencia artificial generativa para proporcionar apoyo durante la programación visual y orientar la realización de proyectos de acuerdo con objetivos de aprendizaje previamente establecidos [18]. La interfaz adquiere así un carácter dinámico, en el que representación, acción y retroalimentación forman parte de un mismo proceso de aprendizaje.

La literatura también evidencia que la representación visual debe mantener una correspondencia comprensible entre la metáfora utilizada y el concepto que pretende representar. En educación, las metáforas visuales pueden actuar como estructuras que relacionan representaciones gráficas con contenidos que presentan mayor nivel de abstracción. Un estudio reciente aplicado al aprendizaje del *Business Model Canvas* encontró que el diseño instruccional apoyado en metáforas visuales integra componentes gráficos y textuales para facilitar la comprensión del modelo y abordar la carga cognitiva asociada con su aprendizaje [19]. Esto indica que la utilidad de la metáfora no reside únicamente en incorporar elementos gráficos, sino

en la relación significativa que estos establecen con el contenido representado, en coherencia con la noción de correspondencia entre modelo conceptual y sistema planteada desde el diseño centrado en el usuario [6].

En conjunto, los hallazgos permiten caracterizar estas interfaces mediante cinco componentes principales: representación visual, organización secuencial, interacción directa, retroalimentación y correspondencia entre la representación y la lógica del procedimiento. Estas características muestran que la metáfora de interfaz no funciona únicamente como un recurso gráfico; puede constituirse en una estructura de interacción mediante la cual el estudiante representa, organiza, ejecuta y revisa procedimientos. Esta interpretación resulta consistente con los entornos recientes de visualización algorítmica, programación visual y programación tangible analizados en la literatura [20], [21], [17].

3.3 RQ2: Métodos, criterios e instrumentos utilizados para evaluar la usabilidad de estas metáforas de interfaz

La revisión evidencia que la usabilidad de las interfaces orientadas al pensamiento algorítmico se evalúa mediante una combinación de métodos cuantitativos y cualitativos. Los estudios no se limitan a determinar si el estudiante puede utilizar una herramienta, sino que consideran la experiencia que se produce durante la interacción, la facilidad para completar las actividades propuestas y las dificultades que aparecen durante su utilización. En este sentido, las evaluaciones combinan cuestionarios, observación de usuarios, entrevistas, registros de interacción y análisis del desempeño durante tareas de programación o resolución de problemas [21], [23], [27]. La Tabla 8 sintetiza los métodos e instrumentos identificados, diferenciando los que evalúan propiedades de la interfaz de los que evalúan resultados de aprendizaje.

Tabla 8: *Métodos, criterios e instrumentos de evaluación identificados en el corpus*

Componente evaluado	Criterios	Métodos e instrumentos	Estudios
Facilidad y eficiencia de interacción (propiedad de la interfaz)	Facilidad de uso, claridad, eficiencia, predictibilidad, velocidad	Cuestionarios estandarizados (UEQ, SUS), cuestionarios tipo Likert, evaluación heurística	[1]. [2], [18], [21], [32]
Comportamiento del usuario durante las tareas (propiedad de la interacción)	Errores, recordación, finalización de tareas, patrones de atención	Registros del sistema, grabaciones, seguimiento ocular, análisis de secuencias, observación	[22], [27], [23], [26]
Experiencia subjetiva (percepción del usuario)	Carga cognitiva, disfrute, motivación, percepción de utilidad	Escalas de carga cognitiva y motivación, entrevistas	[24], [19], [23], [21]
Desempeño asociado con el aprendizaje (resultado de aprendizaje; se reporta en RQ3)	Pensamiento computacional, razonamiento espacial, calidad de programas	Pruebas de PC, rúbricas, análisis de artefactos	[17], [24], [28], [29], [31]

Fuente: elaboración propia. a partir de los estudios incluidos en la revisión sistemática.

Entre los instrumentos estandarizados para evaluar la experiencia de usuario se encuentra el *User Experience Questionnaire* (UEQ), que permite valorar seis dimensiones: atractivo, perspicuidad, eficiencia, control, estimulación y novedad [1], en su estudio sobre Kniwwelino Classroom Kit, aplicaron el UEQ a docentes y complementaron la evaluación con cuestionarios previos y posteriores dirigidos a estudiantes, entrevistas semiestructuradas con docentes y registros de las sesiones. Los resultados evidenciaron una experiencia de usuario generalmente positiva, aunque también identificaron aspectos de la interfaz susceptibles de mejora, relacionados con su facilidad de uso y funcionamiento.

También se identifican evaluaciones centradas directamente en el comportamiento del usuario durante la interacción. En los entornos de programación visual y tangible se analizan aspectos como errores cometidos, facilidad para recordar el funcionamiento de la interfaz, conveniencia de uso y capacidad para completar las tareas propuestas. Estos datos pueden obtenerse mediante registros del sistema, observaciones, grabaciones de video, seguimiento ocular, entrevistas y cuestionarios posteriores a la interacción [22], [27], [26]. La combinación de estas fuentes permite contrastar lo que el usuario manifiesta sobre la interfaz con las dificultades que efectivamente presenta mientras la utiliza [23].

Otro grupo de estudios incorpora indicadores de experiencia subjetiva como parte de la evaluación de las interfaces. En estos trabajos se consideran variables como carga cognitiva, disfrute, motivación y percepción de utilidad. La

comparación entre herramientas tangibles, programación por bloques y actividades realizadas con papel y lápiz muestra que las modalidades de interacción pueden generar experiencias diferentes: las interfaces tangibles pueden reducir la carga cognitiva, mientras que los entornos basados en bloques pueden producir mayores niveles de disfrute y favorecer determinadas actividades relacionadas con el diseño de algoritmos [24]. Es importante precisar que, en varios de estos estudios, las medidas de experiencia se reportan junto con indicadores de desempeño en pensamiento computacional o razonamiento espacial; para efectos de esta revisión, los primeros se consideran evidencia de usabilidad y experiencia de usuario (RQ2), y los segundos, evidencia de aportes al aprendizaje (RQ3).

Asimismo, algunas investigaciones utilizan procedimientos de evaluación heurística para identificar problemas de diseño antes o de manera complementaria a las pruebas con usuarios [10]. En entornos de programación por bloques, este procedimiento permite examinar aspectos de la interfaz que pueden dificultar la localización de elementos, la comprensión de las acciones disponibles o la transición hacia otras formas de representación de los algoritmos. La evaluación mediante heurísticas específicas del dominio, complementada con el análisis de problemas reportados por usuarios, permite convertir las dificultades encontradas en recomendaciones y criterios para el rediseño de las interfaces educativas [2]; en una línea similar, se han propuesto marcos pedagógicos para evaluar herramientas de apoyo al pensamiento computacional que integran criterios de usabilidad con criterios didácticos [32].

Los resultados muestran que la evaluación de usabilidad de estas interfaces puede organizarse alrededor de tres componentes propios de la interfaz y la interacción: facilidad y eficiencia de interacción, comportamiento del usuario durante las tareas y experiencia subjetiva. Los instrumentos empleados incluyen escalas estandarizadas como UEQ y SUS, cuestionarios tipo Likert, entrevistas, observación directa, registros de interacción, seguimiento ocular, análisis de errores y mediciones de carga cognitiva o satisfacción [18], [21]–[24], [27]. Esta diversidad evidencia que la usabilidad en contextos educativos requiere considerar simultáneamente las propiedades de la interfaz y la manera en que estas influyen en el desarrollo de las actividades de aprendizaje, lo que coincide con la definición de usabilidad como una propiedad dependiente del contexto de uso [9].

3.4 RQ3: Aportes al aprendizaje y limitaciones reportadas sobre el uso de estas metáforas

Los estudios analizados muestran como uno de los principales aportes la posibilidad de hacer visibles procesos de razonamiento que normalmente permanecen en un nivel abstracto. Las interfaces basadas en bloques, diagramas, representaciones tangibles y visualizaciones dinámicas permiten exteriorizar secuencias, relaciones y transformaciones durante la resolución de problemas. Esta representación facilita que el estudiante observe el procedimiento que está construyendo, identifique sus componentes y modifique progresivamente la solución, reduciendo inicialmente la dependencia de lenguajes formales o de estructuras sintácticas complejas [27], [14].

Otro aporte se relaciona con el fortalecimiento de habilidades vinculadas con el pensamiento computacional y algorítmico. Los resultados reportados en entornos de programación visual muestran mejoras principalmente en habilidades asociadas con construcción de secuencias, diseño de algoritmos y resolución estructurada de problemas [28], [29]. De manera similar, las interfaces tangibles han mostrado efectos favorables en habilidades como abstracción, descomposición de problemas, depuración y razonamiento espacial [17], [26], [24]. No obstante, estos resultados no son uniformes para todas las dimensiones del pensamiento computacional, lo que indica que el tipo de representación y las actividades propuestas condicionan las habilidades que pueden desarrollarse [28], [25].

La retroalimentación inmediata constituye otro aporte relevante. Las interfaces interactivas permiten representar los cambios producidos durante la ejecución de un algoritmo y relacionarlos con las decisiones adoptadas por el estudiante. Las visualizaciones paso a paso, el resaltado sincronizado de instrucciones y la representación de los estados intermedios facilitan el seguimiento de procedimientos cuya ejecución podría resultar difícil de representar mentalmente. Estudios recientes sobre visualización interactiva de algoritmos muestran una valoración favorable de estos mecanismos para comprender la secuencia de ejecución y relacionar el código con los cambios producidos sobre los datos [21], [23].

Los resultados también muestran efectos favorables sobre componentes afectivos y experienciales del aprendizaje. Las interfaces visuales y tangibles pueden incrementar el interés, la participación y el

disfrute durante actividades relacionadas con programación y pensamiento computacional [24], [23]. En particular, los entornos tangibles facilitan la manipulación directa de representaciones y pueden favorecer dinámicas de colaboración [17], mientras que los sistemas de programación visual, incluidos aquellos con apoyo conversacional o inteligente, permiten experimentar con diferentes soluciones sin que los errores sintácticos constituyan la principal barrera de entrada [30], [18]. Estas características amplían las posibilidades de participación de estudiantes con diferentes niveles de experiencia previa.

Sin embargo, la evidencia también permite identificar limitaciones. Una interfaz visual no garantiza por sí misma el desarrollo del pensamiento computacional. Estudios con programación basada en bloques han encontrado mejoras en habilidades específicas de programación o en la calidad de los programas sin observar necesariamente efectos equivalentes sobre medidas generales de pensamiento computacional [31], [28]. Esto sugiere que el aprendizaje depende no solamente de la representación utilizada, sino también del diseño pedagógico, las actividades propuestas, el acompañamiento docente y la correspondencia entre las acciones realizadas en la interfaz y los procesos cognitivos que se pretende desarrollar [32].

Otra limitación se relaciona con la complejidad que pueden alcanzar las propias representaciones visuales. A medida que aumenta el número de bloques, elementos, conexiones o estados representados, la interfaz puede incrementar la carga cognitiva y dificultar la localización de componentes [22], [24]. También se reportan problemas relacionados con predictibilidad, velocidad de interacción, organización de los elementos y transición entre representaciones visuales y lenguajes textuales [15]. Por ello, reducir las exigencias sintácticas no elimina necesariamente todas las dificultades de interacción, sino que puede trasladarlas hacia la organización espacial y la interpretación de los componentes visuales.

Finalmente, los estudios presentan limitaciones metodológicas relacionadas con la diversidad de contextos educativos, edades, instrumentos y periodos de intervención. Las investigaciones incluyen desde educación infantil hasta educación superior y utilizan diferentes indicadores para evaluar usabilidad, experiencia de usuario y pensamiento computacional, dificultando la comparación directa de los resultados [17], [24], [26], [21]. Asimismo, algunos trabajos evalúan prototipos o experiencias desarrolladas durante periodos relativamente cortos, por lo que todavía resulta necesario estudiar la permanencia de los aprendizajes y la transferencia de las habilidades adquiridas hacia problemas diferentes de aquellos utilizados durante la intervención [33], [28].

4. DISCUSIÓN

4.1 Contraste con la literatura previa

Los cinco componentes identificados en RQ1 (representación visual, organización secuencial, interacción directa, retroalimentación y correspondencia metáfora–procedimiento) son coherentes con los principios clásicos del diseño de metáforas de interfaz: la metáfora debe proporcionar un modelo conceptual que haga visibles las acciones posibles y mantenga una correspondencia natural entre controles y efectos [6], y su poder explicativo proviene de la proyección de un dominio concreto sobre uno abstracto [5]. La revisión muestra, sin embargo, que la literatura reciente rara vez utiliza explícitamente el lenguaje de las metáforas de interfaz y, en su lugar, describe entornos de programación por bloques, tangibles o de visualización de algoritmos. Esta brecha terminológica explica la dispersión de la evidencia entre los campos de Interacción Humano-Computador y tecnologías educativas, y sugiere la utilidad de recuperar el marco de las metáforas de interfaz para analizar de manera integrada estos entornos.

En RQ2, la diversidad de instrumentos identificada coincide con la definición de usabilidad de ISO 9241-11, que exige considerar efectividad, eficiencia y satisfacción en un contexto de uso específico [9], y con la complementariedad entre métodos de inspección y pruebas con usuarios propuesta por Nielsen [10]. Un hallazgo relevante es que pocos estudios reportan indicadores objetivos de efectividad y eficiencia (tiempo y tasa de finalización de tareas, errores), privilegiando medidas de percepción y experiencia; la incorporación de registros de interacción y seguimiento ocular [22], [27] constituye un avance en esa dirección. En RQ3, los aportes reportados se concentran en las dimensiones de conceptos y prácticas del marco de Brennan y Resnick [8] (secuencias, depuración, diseño incremental), mientras que la evidencia sobre perspectivas computacionales y sobre transferencia es escasa, lo que coincide con revisiones previas que señalan la heterogeneidad de los instrumentos de medición del pensamiento computacional [7], [3].

4.2 Brechas de conocimiento e implicaciones

La revisión identifica tres brechas principales. Primero, el predominio de estudios en educación infantil, primaria y secundaria implica que la evidencia sobre metáforas de interfaz basadas en pensamiento algorítmico en educación superior, y en particular en disciplinas no informáticas, es limitada; los resultados de esta revisión no permiten afirmar que los aportes identificados se reproduzcan en esos contextos. Segundo, la evaluación de usabilidad y la evaluación de resultados de aprendizaje suelen reportarse de manera entrelazada, lo que dificulta establecer si las mejoras observadas se deben a las propiedades de la interfaz o al diseño pedagógico; se recomienda que los estudios futuros reporten por separado indicadores de usabilidad (ISO 9241-11) e indicadores de aprendizaje. Tercero, la correspondencia entre la metáfora y el dominio de conocimiento representado ha sido poco estudiada fuera de la programación; el estudio sobre el *Business Model Canvas* [19] sugiere que las metáforas visuales pueden aplicarse a contenidos procedimentales de otras disciplinas, pero esta línea requiere más evidencia empírica. Como implicación para el diseño, los resultados sugieren que una metáfora de interfaz orientada al pensamiento algorítmico debería integrar los cinco componentes identificados y ser evaluada con instrumentos que distingan propiedades de la interfaz, comportamiento del usuario y experiencia subjetiva.

4.3 Limitaciones del estudio

En primer lugar, no puede descartarse la omisión de estudios pertinentes debido a las limitaciones de cobertura de las fuentes consultadas. En segundo lugar, la cadena principal de búsqueda se formuló en inglés, por lo que la literatura en español y portugués sin metadatos en ese idioma pudo quedar subrepresentada, pese a la consulta de Redalyc. En tercer lugar, el corpus incluido presenta heterogeneidad en los niveles educativos, diseños de investigación e instrumentos de evaluación, lo que limitó la síntesis a un análisis descriptivo y temático e impidió realizar una síntesis cuantitativa de efectos. Finalmente, la evaluación de calidad se basó en cuatro criterios generales y no en un instrumento validado para evaluar el riesgo de sesgo, lo que constituye una limitación para valorar la solidez metodológica de la evidencia sintetizada.

5. CONCLUSIONES

En respuesta a RQ1, los 21 estudios incluidos permiten caracterizar las metáforas de interfaz basadas en pensamiento algorítmico para el aprendizaje mediante cinco componentes: representación visual de procesos abstractos, organización lógica y secuencial de las acciones, interacción directa con las representaciones, retroalimentación durante la interacción y correspondencia entre la metáfora y la lógica del procedimiento representado. Estas metáforas se materializan principalmente en interfaces de programación por bloques, interfaces tangibles, visualizaciones interactivas de algoritmos y, en menor medida, metáforas visuales aplicadas a contenidos procedimentales de otras disciplinas.

En respuesta a RQ2, la usabilidad de estas interfaces se evalúa mediante una combinación de instrumentos estandarizados de usabilidad y experiencia de usuario (UEQ, SUS, escalas tipo Likert), métodos centrados en el comportamiento del usuario (registros de interacción, observación, grabaciones y seguimiento ocular), medidas de experiencia subjetiva (carga cognitiva, disfrute, motivación) y métodos de inspección como la evaluación heurística. Se identificó que los indicadores objetivos de efectividad y eficiencia son menos frecuentes que las medidas de percepción, y que la evaluación de usabilidad suele reportarse de manera entrelazada con la evaluación de resultados de aprendizaje.

En respuesta a RQ3, los aportes reportados comprenden la exteriorización de procesos de razonamiento, el fortalecimiento de habilidades de secuenciación, diseño de algoritmos, depuración y razonamiento espacial, la retroalimentación inmediata y efectos favorables sobre el interés, la participación y el disfrute. Las limitaciones comprenden la ausencia de garantía de que una interfaz visual desarrolle por sí misma el pensamiento computacional, el incremento de la carga cognitiva cuando las representaciones se vuelven complejas, las dificultades de transición hacia lenguajes textuales y la heterogeneidad metodológica del corpus, que impide comparaciones directas y deja abiertas las preguntas sobre permanencia y transferencia de los aprendizajes.

Dado el predominio de estudios en educación infantil, primaria y secundaria, las conclusiones de esta revisión se circunscriben a esos contextos. Como trabajo futuro se propone el diseño y la

evaluación empírica de metáforas de interfaz basadas en pensamiento algorítmico en educación superior, reportando de forma diferenciada indicadores de usabilidad conforme a ISO 9241-11 e indicadores de aprendizaje, de manera que pueda establecerse en qué medida los cinco componentes identificados contribuyen a la comprensión de procedimientos en disciplinas no informáticas.

Contribución de los autores: Conceptualización: D.X.R.R. y J.A.J.T.; metodología: D.X.R.R.; investigación: D.X.R.R. y J.A.J.T.; análisis formal: D.X.R.R.; redacción del borrador original: D.X.R.R.; revisión y edición del manuscrito: J.A.J.T.; supervisión: J.A.J.T. Ambos autores han leído y aprobado la versión final del manuscrito.

Financiación: Esta investigación no recibió financiación externa.

Disponibilidad de materiales y datos: La matriz de extracción de datos correspondiente a los 21 estudios incluidos en la revisión sistemática se encuentra disponible en el repositorio Zenodo, versión 1.0, bajo el identificador digital permanente (DOI): https://doi.org/10.5281/zenodo.23844664.

Agradecimientos: Los autores agradecen a la Universidad CESMAG por el apoyo institucional brindado para el desarrollo de esta investigación.

REFERENCIAS

[1] O. Estrada Molina, D. R. Fuentes-Cancell, and A. García-Hernández, "Evaluating usability in educational technology: A systematic review from the teaching of mathematics," *LUMAT*, vol. 10, no. 1, Feb. 2022, doi: 10.31129/LUMAT.10.1.1686.

[2] Q. Li and Y. Luximon, "Navigating the mobile applications: The influence of interface metaphor and other factors on older adults' navigation behavior," *Int. J. Hum.–Comput. Interact.*, vol. 39, no. 5, pp. 1184–1200, Mar. 2023, doi: 10.1080/10447318.2022.2050540.

[3] T. H. Lehmann, "How current perspectives on algorithmic thinking can be applied to students' engagement in algorithmatizing tasks," *Math. Educ. Res. J.*, vol. 36, no. 3, pp. 609–643, Sep. 2024, doi: 10.1007/s13394-023-00462-0.

[4] M. Giannakos, *Experimental Studies in Learning Technology and Child–Computer Interaction* (SpringerBriefs in Educational Communications and Technology). Cham, Switzerland: Springer, 2022, doi: 10.1007/978-3-031-14350-2.

[5] G. Lakoff and M. Johnson, *Metaphors We Live By*. Chicago, IL, USA: Univ. of Chicago Press, 1980.

[6] D. A. Norman, *The Design of Everyday Things*, revised and expanded ed. New York, NY, USA: Basic Books, 2013.

[7] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, Mar. 2006, doi: 10.1145/1118178.1118215.

[8] K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," in *Proc. Annu. Meeting Amer. Educ. Res. Assoc. (AERA)*, Vancouver, BC, Canada, 2012, pp. 1–25.

[9] *Ergonomics of Human-System Interaction—Part 11: Usability: Definitions and Concepts*, ISO 9241-11:2018, Int. Org. Standardization, Geneva, Switzerland, 2018.

[10] J. Nielsen, *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann, 1993.

[11] B. Kitchenham, L. Madeyski, and D. Budgen, "SEGRESS: Software engineering guidelines for reporting secondary studies," *IEEE Trans. Softw. Eng.*, vol. 49, no. 3, pp. 1273–1298, Mar. 2023, doi: 10.1109/TSE.2022.3174092.

[12] M. J. Page et al., "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *BMJ*, vol. 372, Art. no. n71, Mar. 2021, doi: 10.1136/bmj.n71.

[13] Á. Molina-Ayuso, N. Adamuz-Povedano, R. Bracho-López, and M. Torralbo-Rodríguez, "Introduction to computational thinking with Scratch for teacher training for Spanish primary school teachers in mathematics," *Educ. Sci.*, vol. 12, no. 12, Art. no. 899, Dec. 2022, doi: 10.3390/educsci12120899.

[14] X. Gao, Y. Yang, Y. Du, and D. Sun, "Effect of mind mapping-based scaffolding on elementary students' computational thinking in block-based programming," *J. Educ. Comput. Res.*, vol. 63, no. 1, pp. 236–271, Mar. 2025, doi: 10.1177/07356331241303067.

[15] H. Moon, J. Cheon, and K. Kwon, "Difficult concepts and practices of computational thinking using block-based programming," *Int. J. Comput. Sci. Educ. Sch.*, vol. 5, no. 3, pp. 3–16, May 2022, doi: 10.21585/ijcses.v5i3.129.

[16] C. W. Budiyanto, K. Fenyvesi, Y. I. Maharani, R. A. Yuana, P. K. Nashiroh, and R. Latifah, "Investigating computational thinking in K–12 visual programming activities on Code.org: A Brennan–Resnick framework approach," *Elinvo (Electron., Informat., Vocational*

Educ.), vol. 10, no. 1, pp. 54–62, Jul. 2025, doi: 10.21831/elinvo.v10i1.78486.

[17] N. Pellas, "Enhancing computational thinking, spatial reasoning, and executive function skills: The impact of tangible programming tools in early childhood and across different learner stages," *J. Educ. Comput. Res.*, vol. 63, no. 1, pp. 3–32, Mar. 2025, doi: 10.1177/07356331241292767.

[18] Y. Chen, S. Xiao, Y. Song, Z. Li, L. Sun, and L. Chen, "MindScratch: A visual programming support tool for classroom learning based on multimodal generative AI," *Int. J. Hum.–Comput. Interact.*, vol. 41, no. 21, pp. 13650–13668, Nov. 2025, doi: 10.1080/10447318.2025.2475991.

[19] Y.-L. Lin, W.-T. Wang, and I.-T. Lee, "Implementing visual metaphor-supported instructional design in management education: An example of learning a business model canvas," *Int. J. Manage. Educ.*, vol. 23, no. 3, Art. no. 101256, Dec. 2025, doi: 10.1016/j.ijme.2025.101256.

[20] G. Adorni, S. Piatti, and V. Karpenko, "Virtual CAT: A multi-interface educational platform for algorithmic thinking assessment," *SoftwareX*, vol. 27, Art. no. 101737, Sep. 2024, doi: 10.1016/j.softx.2024.101737.

[21] L. Végh, O. Takáč, K. Czakóová, and D. Demeter, "Design and evaluation of interactive matrix-based algorithm visualizations in introductory programming: A student perception study," *Educ. Sci.*, vol. 16, no. 7, Art. no. 1135, Jul. 2026, doi: 10.3390/educsci16071135.

[22] M.-J. Tsai, F. P. Chien, W.-T. Sun, and N. K. Jha, "How block-based programming supports novice learners' coding comprehension: Evidence from eye-tracking lag-sequential analysis," *Comput. Educ.*, vol. 239, Art. no. 105430, Dec. 2025, doi: 10.1016/j.compedu.2025.105430.

[23] Q. Fu, X. Zhou, Y. Zheng, and Z. Wang, "The effects of algorithm visualisation on elementary school students' algorithm-learning performance, motivation, and behaviour," *J. Comput. Assist. Learn.*, vol. 41, no. 3, Art. no. e70049, Jun. 2025, doi: 10.1111/jcal.70049.

[24] X. Gong, W. Xu, S. Yu, J. Ma, and A. Qiao, "Enhancing computational thinking and spatial reasoning skills in gamification programming learning: A comparative study of tangible, block and paper-and-pencil tools," *Brit. J. Educ. Technol.*, vol. 56, no. 1, pp. 80–102, Jan. 2025, doi: 10.1111/bjet.13482.

[25] N. Fanchamps, E. van Gool, L. Slangen, and P. Hennissen, "The effect on computational thinking and identified learning aspects: Comparing unplugged smartGames with SRA-programming with tangible or on-screen output," *Educ. Inf. Technol.*, vol. 29, no. 3, pp. 2999–3024, Feb. 2024, doi: 10.1007/s10639-023-11956-6.

[26] A. Misirli and V. Komis, "Computational thinking in early childhood education: The impact of programming a tangible robot on developing debugging knowledge," *Early Childhood Res. Quart.*, vol. 65, pp. 139–158, 2023, doi: 10.1016/j.ecresq.2023.05.014.

[27] T.-T. Wu, C.-J. Lin, S.-C. Wang, and Y.-M. Huang, "Tracking visual programming language-based learning progress for computational thinking education," *Sustainability*, vol. 15, no. 3, Art. no. 1983, Jan. 2023, doi: 10.3390/su15031983.

[28] J. Piedade and N. Dorotea, "Effects of Scratch-based activities on 4th-grade students' computational thinking skills," *Informat. Educ.*, vol. 22, no. 3, pp. 499–523, 2023, doi: 10.15388/infedu.2023.19.

[29] L. Zhao, X. Liu, C. Wang, and Y.-S. Su, "Effect of different mind mapping approaches on primary school students' computational thinking skills during visual programming learning," *Comput. Educ.*, vol. 181, Art. no. 104445, May 2022, doi: 10.1016/j.compedu.2022.104445.

[30] K. Katchapakirin, C. Anutariya, and T. Supnithi, "ScratchThAI: A conversation-based learning support framework for computational thinking development," *Educ. Inf. Technol.*, vol. 27, no. 6, pp. 8533–8560, Jul. 2022, doi: 10.1007/s10639-021-10870-z.

[31] K.-D. Boom, M. Bower, J. Siemon, and A. Arguel, "Relationships between computational thinking and the quality of computer programs," *Educ. Inf. Technol.*, vol. 27, no. 6, pp. 8289–8310, Jul. 2022, doi: 10.1007/s10639-022-10921-z.

[32] S. S. Oyelere, F. J. Agbo, and I. T. Sanusi, "Developing a pedagogical evaluation framework for computational thinking supporting technologies and tools," *Front. Educ.*, vol. 7, Art. no. 957739, Aug. 2022, doi: 10.3389/feduc.2022.957739.

[33] Y.-S. Su, M. Shao, and L. Zhao, "Effect of mind mapping on creative thinking of children in Scratch visual programming education," *J. Educ. Comput. Res.*, vol. 60, no. 4, pp. 906–929, Jul. 2022, doi: 10.1177/07356331211053383.