

## Programación Visual. Aplicaciones y Perspectivas en la Educación

Guzmán, Eduardo<sup>1\*</sup>; Chaparro, Evelio<sup>1</sup>; Castillo, Julián<sup>1</sup>

<sup>1</sup> *Institución Universitaria Digital de Antioquia, Facultad de Ingeniería y Ciencias Agropecuarias, Medellín, Colombia*

**Resumen:** El pensamiento computacional emerge como una apremiante habilidad a desarrollar en los estudiantes de grados K-12 quienes se enfrentan a la necesidad de adaptarse cada vez mejor y más rápido a entornos altamente digitalizados. El presente artículo corresponde a una reflexión documentada sobre trabajos de investigación publicados entre los años 2017 y 2021, sobre el uso de la programación visual como herramienta de apoyo pedagógico para el desarrollo del pensamiento computacional en ese grupo de estudiantes y que evidenció principalmente, la necesidad de realizar aportes para estructurar marcos de referencia y estándares de evaluación del uso de estas herramientas en el aula, así como la necesidad de abordar y aplicar enfoques que no se circunscriban al uso de la computadora como único medio de enseñanza y aprendizaje del pensamiento computacional.

**Palabras clave:** Pensamiento computacional. Educación en grados K-12. Programación con juegos. Programación visual.

Recibido: 5 de junio de 2022. Aceptado: 30 de noviembre de 2022

Received: June 5th, 2022. Accepted: November 30th, 2022

### Visual Programming. Applications and Perspectives in Education

**Abstract:** Computational thinking emerges as a compelling skill to develop in K-12 students who are faced with having to adapt better and faster to highly digitized environments. This article is a documented reflection based on published papers between 2017 and 2021 about the use of visual programming as a pedagogical tool for the development of computational thinking in this group of students. The review mainly demonstrated the need to make contributions to constructing a reference framework and evaluation standards for the use of these tools in the classroom, as well as the need to address and apply approaches, not limited to the use of the computer as the only means of learning computational thinking.

**Keywords:** Computational thinking. K-12 education. Programming with games. Visual programming.

---

(\*) [eduardo.guzman@est.iudigital.edu.co](mailto:eduardo.guzman@est.iudigital.edu.co)

## 1. INTRODUCCIÓN

El pensamiento computacional, es un término que fue relacionado inicialmente a comienzos del siglo diecinueve con el uso de análisis cuantitativo en el desarrollo de la actividad científica y más tarde al hacer énfasis en la enseñanza del razonamiento en el estudio de la aritmética (Kong & Abelson 2019). Por otro lado, se reconoce que el pensamiento computacional se define apoyándose en los trabajos de Wing (2006), para la comprensión y solución de problemas, utilizando los conceptos básicos fundamentales de la informática ya existentes (Selby & Woollard 2013). En la actualidad, el término ha sido adoptado como modo de señalar una cierta forma de pensar que propicia el análisis y la relación de ideas para la organización y la representación lógica de procedimientos (Zapata-Ros 2015). La importancia relativa reciente de esta forma de pensamiento particular se deriva de las exigencias propias de los nuevos escenarios laborales y formas de producción de la denominada industria 4.0 (Avent 2017) que ha puesto sobre la mesa, la necesidad de desarrollar e implementar una nueva alfabetización desde la óptica de lo digital.

En ese sentido, se han incrementado en la última década las discusiones sobre métodos y procedimientos de enseñanza y aprendizaje del pensamiento computacional con el fin de adaptar los currículos a la formación de los nuevos profesionales cualificados requeridos por la nueva industria, con mayor énfasis en aquellas destrezas básicas que supone el desarrollo del pensamiento computacional, tales como las relacionadas con la creatividad, la recursividad, el pensamiento abstracto, la resolución de problemas y la aplicación de métodos colaborativos para la solución de problemas, como los más importantes a desarrollar desde la edad temprana (Zapata-Ros 2015).

Las dificultades propias del proceso de aprendizaje de estas habilidades representan uno de sus mayores retos en todos los niveles del proceso de enseñanza aprendizaje, particularmente críticos en los primeros años de educación, donde se ven comprometidos todos los componentes formativos fundamentales de la persona (Sáez & Cózar 2017). El uso de diversas herramientas de programación ha demostrado impactos positivos en estudiantes en edad escolar de hasta grado doce o los denominados grados K-12 (Kindergarten – 12), particularmente con el uso de las denominadas herramientas de programación visual (Sun & Zhou 2021) y para temas transversales como las ciencias sociales (Sáez López, J. M., & Cózar Gutiérrez, R. 2017). Si bien es cierto que ya Papert (1980) había establecido las bases de una propuesta educativa fundamentada en el desarrollo del pensamiento computacional a través del lenguaje de programación “LOGO” (Furzeig, Papert, & Solomon, 1967), la explosión de la internet a finales de los noventa del siglo pasado y la masificación de las computadoras han supuesto la aparición de numerosas herramientas de programación visual, entendidas como aquellas que utilizan elementos gráficos o pictóricos para la creación de programas de computadora, en lugar del uso de sintaxis de programación formalmente rígidas (Míguez & Neira 2020).

Alves, Von Wangenheim & Hauck (2019) destacan el uso de herramientas de programación visual como Scratch (MIT Media Lab, 2008), Blockly (Blocky Team, 2022),

BYOB/Snap! (Mönig & Harvey, 2022) y App Inventor (MIT Center for Mobile Learning, 2013), como apoyo en el desarrollo de pensamiento computacional en educación en los grados K-12 (Šiaulyš 2021), característico entre estas herramientas, la aplicación del sentido lúdico como fundamento de su estructuración y funcionamiento, lo que las hace marcadamente muy efectivas, sobre todo en estudiantes en edades tempranas (Vera 2020).

En ese sentido, cabe destacar el interés que ha propiciado iniciativas como la CoolThink@JC, creada y fundada por la Hong Kong Jockey Club Charities Trust, la Universidad de Hong Kong y el Instituto Tecnológico de Massachusetts enfocada en el estudio del pensamiento computacional en la educación (CTE 2019) y que en su más reciente conferencia en el año 2019 expusieron trabajos alrededor de experiencias educativas de diversas partes del mundo, agrupadas en dieciséis sub temas que van desde la enseñanza de la codificación en niños menores de doce años, tópicos de Ciencia de Datos hasta propuestas de políticas gubernamentales aplicadas al sector de la educación. Esto plantea muy interesantes interrogantes alrededor del alcance y la viabilidad de incorporar estas metodologías en los currículos de educación, para el desarrollo del pensamiento computacional en el contexto latinoamericano y colombiano en particular.

Por consiguiente, se propone documentar, de forma general, propuestas y experiencias de aplicación de los lenguajes de programación visual como herramienta de apoyo en el desarrollo de procesos de enseñanza aprendizaje del pensamiento computacional en estudiantes de grados K-12, representados en artículos revisados y publicados entre los años 2017 y 2021 como forma de abordar la temática en una discusión reflexiva.

## 2. MARCO TEÓRICO

Hay pocas dudas hoy en día en relación con la importancia que representa la educación en ciencias de la computación y lo que el pensamiento computacional aporta para su desarrollo en la actualidad; importancia enfatizada aún más con las necesidades de adaptación económica y social exigidas derivadas del manejo de la pandemia de COVID -19 y que dejaron en evidencia los complejos problemas generados por la inequidad en el acceso de tecnologías de la información y la comunicación, en especial de muchos estudiantes alrededor del mundo (CSTA, & ECEP Alliance 2020).

Como referencia de este artículo se han revisado diferentes fuentes de información científica de amplio reconocimiento a nivel global, así como se ha acogido el marco de la International Conference on Computational Thinking Education 2019 (CTE 2019), cuyos aportes, ideas, intercambios y hallazgos en torno al desarrollo del pensamiento computacional se consideran de relevancia en el estudio del pensamiento computacional en la educación, en especial con el uso creativo de herramientas digitales para su desarrollo, por lo que resulta conveniente estructurar los criterios de búsqueda y categorización de los artículos, basados en cuatro de sus criterios ajustados, a saber: *a. Desarrollo docente*, *b. Programación visual -educación STEM*, *c. Programación visual, aprendizaje y enseñanza*, *d.*

*Experiencias con Scratch en el desarrollo del pensamiento computacional y habilidades de programación.*

Por definición y bajo estos criterios, se propone revisar documentos que sobre este marco han nutrido diversos estudios científicos y docentes alrededor del mundo. Esta propuesta estima los resultados de actividades experimentales sobre jóvenes estudiantes en torno a la formación del pensamiento computacional y el entorno de trabajo que le ha permitido tener resultados de una u otra manera significativos para la mejora de los procesos educativos actuales. Es un trabajo descriptivo donde las muestras estadísticas son tomadas de jóvenes en edad escolar en varios países sin que el objetivo sea mostrar resultados segregados por rangos de edad o grado de formación académica.

Sin embargo, al margen del debate sobre su conveniencia, se proponen discusiones sobre las formas, métodos y enfoques más eficaces para la implementación de programas de enseñanza y aprendizaje de ciencias de la computación, orientados fundamentalmente a generar o desarrollar el pensamiento computacional desde los primeros años de educación (Vourletsis & Politis 2020). Múltiples enfoques pueden ser vistos en contextos latinoamericanos y particularmente en el caso colombiano como un punto de referencia para la construcción de políticas educativas adaptadas a nuestras realidades y proyectadas a planes de desarrollo a nivel local y nacional en el sector de la educación. Particular interés genera las propuestas que apelan al atractivo visual y lúdico propios de la programación visual como forma de estructurar nuevos patrones de enseñanza del pensamiento computacional (Gao, Lu, Zhao, & Li 2019). Si bien es cierto que existe una variedad de herramientas de programación visual como apoyo pedagógico para el desarrollo del pensamiento computacional, Scratch (MIT Media Lab, 2008) aparece como la más utilizada y referenciada; este es un lenguaje de programación visual con una interfaz sencilla y de libre acceso, para la creación de historietas, juegos y animaciones de forma digital, orientado a la promoción del desarrollo de las habilidades propias del pensamiento computacional en niños y adolescentes. Estudios recientes realizados sobre experiencias en el uso de Scratch en el aula dan validez a su uso en la apropiación del conocimiento en diversos contextos (Rose, Jay & Jacob 2019).

De acuerdo con Jiang & Li (2021) quienes realizaron estudios en estudiantes de quinto grado en grupos de zonas urbanas y rurales de China, con Scratch como herramienta integrada al aula, investigaron su efecto no solo en el desarrollo de las habilidades básicas del pensamiento como recordar, comprender y aplicar sino en las denominadas habilidades de pensamiento de orden superior, tales como análisis, pensamiento predictivo y evaluación, compara los resultados con el test desarrollado sobre la base del Computational Thinking Scale o CTS (Tsai, Liang, & Hsu 2021) aplicados antes y después del desarrollo de la experiencia, se encontró evidencia empírica significativa que permite correlacionar el uso de lenguajes de programación visual como Scratch y el aprendizaje del pensamiento computacional, con un impacto positivo evidente en el desarrollo de habilidades como la creatividad y pensamiento crítico, no tanto para el desarrollo del pensamiento algorítmico y la resolución de problemas, destaca el hecho que se encontraron resultados importantes en

el desarrollo del aprendizaje colaborativo entre los estudiantes para la búsqueda de soluciones a problemas, principalmente en grupos de niñas, lo que abre las posibilidades para nuevos estudios de este tipo, sobre grupos más grandes, con elementos de evaluación ajustados a cada contexto de país y región.

La tendencia del acercamiento a las bondades del uso de herramientas software con medios visuales como Scratch la enfatizan Jiang, Zhao, Gu & Yin (2021) al observar que este brinda oportunidades de comunicación e interacción, claramente un elemento a favor frente al modelo individual que se destaca en la etapa de formación en grados K-12 tradicional. Para Zhang & Nouri (2019), el pensamiento computacional no es exclusivo a ingenieros y técnicos de sistemas de información, señala en su investigación que han usado experimentalmente Scratch con el ánimo de explorar instancias de las preguntas “¿qué enseñar?” y “¿qué se puede aprender?” dentro de grupos de estudiantes de grados K-12 sustentados en el marco de trabajo de Brennan & Resnick (2012) y de Zhang & Nouri (2019) quienes esbozan que los estudiantes acogen conceptos de “entrada – salida”, lectura, interpretación y comunicación de código, uso de medios multimodales, pensamiento predictivo e interacción humano – computadora. De manera concluyente, el autor sugiere algunos estudios complementarios para el cierre de brechas de investigaciones actuales y expone los criterios que justifican el uso de Scratch para potenciar conductas de investigación y aprendizaje en estudiantes de grados K-9.

De acuerdo con Lowe & Brophy (2019) quienes usaron el lenguaje gráfico de bloques ‘Scratch Jr’, exploran un estudio con aprendices de primer nivel que construyen historias gráficas con animación de personajes en ‘Scratch’; en principio no es el aprendizaje de código sino el estímulo de la historia gráfica que busca analizar la progresión del pensamiento en la medida que el estudiante quiere orientar el argumento de la historia. La intencionalidad de la práctica buscó madurar el pensamiento computacional en estudiantes de nivel inicial y logró ver cómo ellos deben construir ‘artefactos’ propios de tareas computacionales sin que la esencia sea formar constructores de código.

Un poco en la misma línea, investigaciones recientes han documentado la integración del pensamiento computacional y el uso de herramientas de programación visual a la enseñanza y aprendizaje de áreas STEM (Science, Technology, Engineering And Mathematics, en español Ciencias, Tecnologías, Ingenierías y Matemáticas) en la educación (Martínez, Rodríguez, Roanes, Fernández, & Cujó 2016) y cuya importancia en edad temprana se ha visto reforzada por el hecho que representa la generación de nuevo conocimiento, en términos de innovación técnica y tecnológica como uno de los activos más valiosos de la sociedad moderna e incluso como ventaja comparativa entre países. De igual forma, autores han agregado la educación artística a las cuatro áreas STEM iniciales, con aplicaciones en áreas STEAM (Science, Technology, Engineering, Arts and Mathematics, en español Ciencias, Tecnologías, Ingenierías Artes y Matemáticas) (Huang & Starcic 2020). Tien & Yen (2019) encontraron que el modelo de instrucción más utilizado está enfocado en la solución de problemas, siendo el diseño de juegos con bloques de Scratch, una forma de apropiar conceptos como las consecuencias del cambio climático, así como aplicaciones básicas en robótica, de especial interés y motivación en niños

y niñas, como las herramientas más utilizadas, se recalca el hecho de reconocer que aún no existe un consenso generalizado propiamente dicho sobre metodologías y contenidos replicables en diferentes contextos, por lo que se esperan mejores esfuerzos en lograr una definición concreta de objetivos a alcanzar y una integralidad entre el desarrollo del pensamiento computacional, el uso de herramientas visuales y la enseñanza aprendizaje de áreas STEM, con especial énfasis en estudiantes de escuelas en grados K-12.

De igual forma, el aporte de tecnologías STEM ha sido considerado un medio transversal para el logro de competencias básicas en resolución de problemas con enfoques de ciencias, ingeniería, tecnología, matemática y procedimientos con un común denominador en la informática, como lo señala Mecca (2021). Para ello, con la asistencia del software ‘Diogene-CT’ (Svelto Tech, 2022) se han apropiado conceptos fundamentales de los servicios de software actuales, con programación por procedimientos y orientada a objetos con una porción de lúdica, ejercicio de colaboración y con técnicas de metáforas visuales conocidas como ‘ACME’ (Animación de código y metáforas avanzadas), que han impulsado el uso de iniciativas nativas de la codificación, la ingeniería y las matemáticas en procesos de aprendizaje lógico o computacional.

Por otra parte, dado que aún no se documentan suficientes experiencias en el desarrollo de programas de educación del pensamiento computacional aplicadas en diferentes contextos, se han propuesto metodologías que buscan orientar al docente en su desarrollo. Gao, Lu, Zhao & Li (2019) proponen un nuevo método que denominan PBL-VP (Problem-Based Learning and the Visual Programming) que aplicaron a la enseñanza de los fundamentos de computación, proponiendo como objetivos a alcanzar, las habilidades reconocidas para el desarrollo del pensamiento computacional (creatividad, pensamiento algorítmico, comunicación, colaboración y pensamiento crítico) en concurso con programación visual por bloques con la herramienta Blockly (Blockly Team, 2022), se enfatiza en la aproximación de los problemas de la vida real y con datos reales, mediante su división para su solución por partes, para luego mediante el uso de la abstracción y la modelación, el diseño del algoritmo, la implementación del programa y la generación del código, así como plantear soluciones de fomento al desarrollo del pensamiento lógico, en un proceso que fue medido en contraste evaluaciones previas y posteriores al desarrollo del método. El estudio destaca el papel del docente como facilitador y guía, en fomento de la discusión y solución de los problemas en los grupos de trabajo, donde prima la importancia de desarrollar las habilidades de abstracción en los estudiantes, imprescindible antes de llevar cualquier solución a un algoritmo.

Asimismo, las garantías en la implementación de programas de educación en general dependen en gran medida de preparación y coordinación de los docentes a cargo de desarrollarlos. Los estándares de la Sociedad Internacional para Tecnología en la educación (ISTE 2016) señalan para los docentes aspectos relacionados con el aprendizaje permanente, el liderazgo, el tener una ciudadanía comprometida, un espíritu de colaboración, ser facilitadores, ser diseñadores y analistas. Sin embargo, al margen de las cualidades y características que se pudiesen señalar en un estándar en particular, se deben

establecer criterios de desarrollo, seguimiento y evaluación de los programas educativos, siendo aún más importante en aquellos para los cuales se empieza a reconocer su verdadera importancia, como lo representan los programas de enseñanza y aprendizaje del pensamiento computacional, en especial en escuelas de grados K-12.

De hecho, la forma de abordar las competencias y habilidades que exige la enseñanza y aprendizaje del pensamiento computacional en estudiantes en grados K-12 y sus formas de aproximación han sido base de trabajos de recopilación reciente. Da Cruz, Gresse & Hauck (2019) señalan que ante evidentes deficiencias en términos de criterios de evaluación y retroalimentación institucional, se evalúan atributos medibles a través de los códigos ensamblados o escritos por los estudiantes y el diseño de rúbricas descriptivas que luego son convertidas a escalas numéricas que muestran el grado de conocimiento sobre habilidades o competencias del pensamiento computacional, con interés de descubrir mejores formas de fomentar y evaluar la creatividad, hecho que en definitiva soporta la necesidad de adelantar mucha más investigación en la forma de medir y evaluar los programas educativos orientados al desarrollo del pensamiento computacional, con especial interés en estudiantes de grados K-12.

De igual importancia resulta el hecho de considerar estrategias de enseñanza y aprendizaje del pensamiento computacional en entornos educativos con limitaciones de infraestructura, técnicas y tecnológicas, muy comunes en países en vías de desarrollo. En ese sentido, investigaciones sobre la aplicación de metodologías basadas en juegos de mesa que utilizan ejercicios lúdicos como las denominadas “coding pocker cards” (WU, 2019) o juegos de mesa para el reconocimiento de patrones (Tseng, Doll, & Varma 2019) en estudiantes de grados K-12 pueden resultar como enormes herramientas de apoyo en los procesos de enseñanza y aprendizaje a muy bajo coste, aunadas a actividades extracurriculares que fomenten y motiven el interés de los estudiantes, podrían generar enormes avances en el desarrollo de habilidades propias del pensamiento computacional (Agyei, Laru, & Mäkitalo 2019).

### 3. METODOLOGÍA

Se planteó la documentación de publicaciones revisadas que recojan propuestas y experiencias de aplicación de la programación visual como herramienta de apoyo en el desarrollo de procesos de enseñanza y aprendizaje del pensamiento computacional, con sus diversos enfoques y principales características, con el objetivo de identificar aplicaciones y perspectivas de la programación visual en la educación.

En consecuencia, se planteó la búsqueda de artículos publicados y revisados entre los años 2017 y 2021, relacionados con las palabras clave “pensamiento computacional” (computational AND thinking) “programación visual” (visual AND programming) y “educación” (AND education) y publicaciones relacionadas, con la siguiente cadena de búsqueda:

TS = TITLE – ABS –

KEY ( computational AND thinking AND visual  
AND programming AND education ) AND ( LIMIT –  
TO ( PUBSTAGE , "final" ) ) (1)

Se examinaron todas las publicaciones disponibles en Scopus y el IEEE (Institute of Electrical and Electronics Engineers) resultantes de la cadena de búsqueda propuesta, con énfasis en los resultados presentados por Scopus, por considerarse la base de datos de publicaciones más extensa, confiable y referenciada, se comprueba duplicidad de artículos con el resto de las bases de datos señaladas, con el fin de consolidar la totalidad de las publicaciones por medio de filtros manuales y automatizados.

Como criterios de selección y calidad se consideraron artículos en inglés y español revisados y publicados entre los años 2017 a 2021, relacionados con la enseñanza y aprendizaje del pensamiento computacional en la educación con el uso de la programación visual como herramienta pedagógica. Se excluyeron, como primer filtro, aquellas publicaciones relacionadas con el aprendizaje de un lenguaje de programación de alto nivel, de desarrollo de código de programación puramente sintáctico o de construcción de algoritmos, enseñanza y aprendizaje para el desarrollo de aplicaciones móviles, publicaciones técnicas o artículos de revistas con encuestas o entrevistas relacionadas con experiencias metodológicas en la educación sin un trasfondo investigativo, así como tesis o trabajos de grado.

Se propone una categorización de los artículos en cuatro grandes grupos basados en el marco conceptual propuesto para el estudio del desarrollo del pensamiento computacional en la educación de la International Conference on Computational Thinking Education 2019 (CTE 2019) adaptados y numerados en este caso de la siguiente manera:

1. Desarrollo Docente, Enseñanza y Aprendizaje de programación y desarrollo de Pensamiento Computacional con el uso de programación visual.
2. Programación visual y Educación en STEM.
3. Programación visual, Aprendizaje y Enseñanza de programación y desarrollo de Pensamiento Computacional en grados K-12.
4. Experiencias con Scratch en el desarrollo del Pensamiento Computacional y habilidades de programación.

De esa forma, se realizaron los análisis y las estadísticas básicas de la bibliometría encontrada, con el fin de determinar los tópicos o grupos de tópicos más estudiados, aplicados o referenciados y en ese mismo sentido se presentarán los resultados, las discusiones y las recomendaciones para futuros estudios que así se pudiesen derivar.

#### 4. RESULTADOS Y/O DISCUSIÓN

En relación con los resultados de la búsqueda, desde el punto de vista cuantitativo inicialmente, luego de hacer las consultas con las dos cadenas de búsqueda utilizadas tanto en la base de datos de Scopus como en la de IEEE (Institute of Electrical and Electronics Engineers), se obtuvieron 165 y 55 publicaciones respectivamente y es la inclusión de la palabra clave “educación” (“education”) el primer ejercicio de filtrado o depuración de búsqueda, donde quedan 104 artículos elegibles de la base de datos Scopus y 54 artículos elegibles de la base de datos de la IEEE.

A partir de este número de artículos (n = 158), se identificaron tanto de manera automatizada como manual, aquellos artículos duplicados en cada una de las bases de datos y de aquellos

duplicados en la base de datos de la IEEE con respecto a los reseñados por Scopus (n = 20) para luego con los artículos restantes (n = 138) realizar los análisis cualitativos, con verificación de contenidos y alcances de cada trabajo y descartando aquellos pertenecientes a conferencias – simposios sin publicación final (n = 5), aquellos relacionados con educación visual o apreciación musical o educación en población con deficiencias visuales y/o cognitivas (n = 11), aquellos dedicados a trabajos sobre el desarrollo y/o codificación de videojuegos o Apps sin contenido educativo importante (n = 7) y aquellos artículos cuya finalidad se inscribe en la discusión de tópicos de enseñanza y aprendizaje o aplicación de experiencias en lenguajes de programación de alto nivel tradicionales (n = 23) quedando con los artículos elegibles para análisis cualitativo (n = 92) definitivos tal y como se muestran en el Diagrama 1.

La totalidad de artículos filtrados fueron analizados y agrupados en cada una de las categorías establecidas para el estudio, destaca el hecho que más del 80% de estos artículos tienen como áreas o sujetos de estudio las ciencias de la computación donde se encuentra el estudio del pensamiento computacional y las ciencias sociales, y se incluye la pedagogía como rama específica de trabajo.

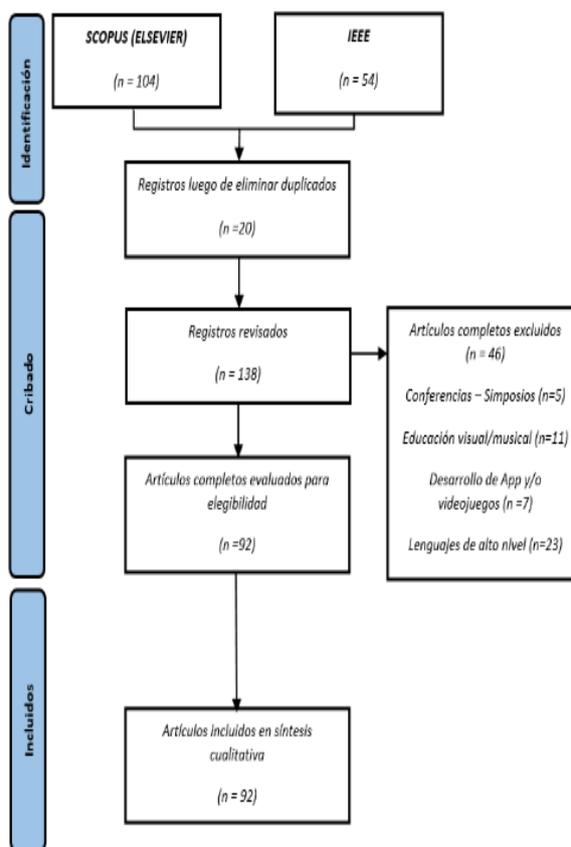


Diagrama 1. Diagrama de Resultados – Matriz prisma del estudio. Elaboración propia.

La búsqueda generó resultados estadísticos importantes con supremacía de temas: desarrollo docente y programación

visual - aprendizaje - enseñanza marcando las mayores apariciones en la consulta (según lo demuestra la Gráfica 1). La Tabla 1 refleja el amplio número de artículos que citan temas comunes al próposito de la búsqueda.

**Tabla 1.** Temas de amplia aparición en búsquedas

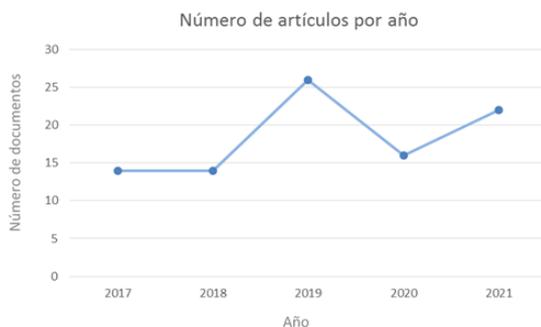
Tema	Artículos que citan el tema
Education	50
Computational thinking	39
Visual programming	15
Scratch	11



**Gráfica 1.** Distribución de Resultados de búsqueda de acuerdo a categorizaciones. Elaboración propia.

En resumen, las categorías Desarrollo Docente, Enseñanza y Aprendizaje de programación y desarrollo de Pensamiento Computacional con el uso de programación visual representan el mayor número de artículos con un 42%, mientras que categoría Programación visual, Aprendizaje y Enseñanza de programación y desarrollo de Pensamiento Computacional en grados K-12 agruparon un 26% de la totalidad de artículos analizados.

En tanto la Gráfica 2 revela cómo se han emitido producciones científicas a través de la línea de tiempo en el lapso de años de interés.

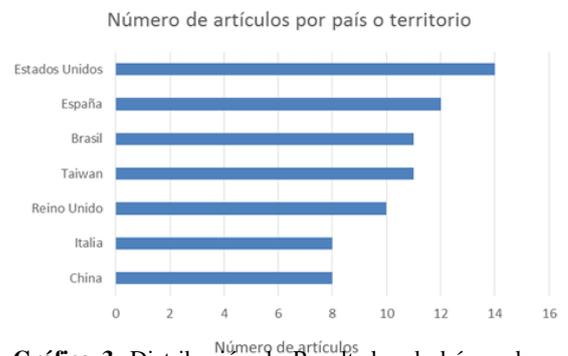


**Gráfica 2.** Distribución de Resultados de búsqueda por años de publicación. Elaboración propia.

En relación con el número de artículos publicados en el rango de años de publicación establecidos, la Gráfica 2 muestra que

el año 2019 con 26 publicaciones fue el de mayor producción, mientras 2017 y 2018 con 14 artículos cada uno, los de menor producción, mientras que se destaca el hecho que ya 2021 manifiesta lo que se esperaría un incremento sustancial en investigaciones pedagógicas en los años sucesivos al tener en cuenta las condiciones especiales generadas por la pandemia de COVID-19.

De igual importancia resulta el hecho de destacar que siete países concentran poco más del 80% de la producción investigativa en el mundo, liderada por Estados Unidos (Ver Gráfica 3). Para el caso latinoamericano, solamente Brasil se encuentra entre estos siete países y particularmente Colombia, aporta una publicación que utiliza datos y experiencias del Instituto Colombiano de Aprendizaje (INCAP) en Bogotá, con investigadores de afiliación española y estadounidense sobre experiencias con Scratch para el desarrollo del pensamiento computacional (Basogain, Olabe, Olabe, & Rico 2018).



**Gráfica 3.** Distribución de Resultados de búsqueda por países o territorios. Elaboración propia.

En general, la mayoría de investigadores coinciden en la necesidad de aumentar los esfuerzos tanto en el desarrollo de más investigaciones, así como en la aplicación y documentación de más experiencias de enseñanza y aprendizaje del pensamiento computacional con diferentes enfoques, a escala global y de forma mancomunada; las asociaciones entre fundaciones públicas o privadas y universidades, tanto en países asiáticos como en Norteamérica, son un buen paso en esa dirección, aún a pesar de sus diferencias políticas e ideológicas. De igual forma, se evidencian en muchos casos, de acuerdo con los investigadores, marcadas diferencias entre metodologías y procedimientos de evaluación de los programas educativos, que hacen que muchas veces los resultados de las experiencias de enseñanza aprendizaje del pensamiento computacional en general, no puedan ser comparables en muchos casos.

La Tabla 2 muestra referencias a publicaciones adicionales a considerar y que pueden ser parte de la base a discusiones en torno a la programación visual como herramienta pedagógica, que si bien no fueron incluidas por los filtros de búsqueda, señalan el desarrollo de investigaciones relacionadas con contenidos parciales en torno a programación visual y el pensamiento computacional, en especial con contenidos experimentales de la programación por bloques gráficos en diversos niveles de educación.

4.1 Otras referencias

**Tabla 2.** Otras referencias

<b>Autor</b>	<b>Año</b>	<b>Título</b>	<b>Descripción</b>	<b>Fuente</b>
Allsopp, B. B., & Misfeldt, M.	2019	Visual representations supporting implementation of a K12 programming curriculum in open and democratic educational institutions	Uso de una representación visual para apoyar la discusión del pensamiento computacional (Allsopp, B. B., & Misfeldt, M, 2019).	Scopus
Csapó, G.	2019	Placing event-action-based visual programming in the process of computer science education	Uso de programación visual basada en eventos y acciones como alternativa del pensamiento computacional (Csapó, G, (2019)	Scopus.
Eguíluz, A., Guenaga, M., Garaizar, P., & Olivares-Rodríguez, C.	2020	Exploring the Progression of Early Programmers in a Set of Computational Thinking Challenges via Clickstream Analysis	Bloques de codificación Kodetu para resolver desafíos (Eguíluz, A., Guenaga, M., Garaizar, P., & Olivares-Rodríguez, C, 2020).	IEEE.
Fragapane, V., & Standl, B.	2021	Work in progress: Creative coding and computer science education - From approach to concept	Trabajo creativo en términos de arte visual en la educación (Fragapane, V., & Standl, B, 2021).	IEEE
Jamil, H. M.	2017	Visual computational thinking using Patch	Uso del lenguaje visual Patch para expresar soluciones a problemas (Jamil, H. M, 2017).	Scopus
João, P., Nuno, D., Fábio, S. F., & Ana, P.	2019	A cross-analysis of block-based and visual programming apps with computer science student-teachers	Uso de aplicaciones de programación visual en la enseñanza (João, P., Nuno, D., Fábio, S. F., & Ana, P, 2019).	Scopus
Katchapakirin, K., & Anutariya, C.	2018	An Architectural Design of ScratchThAI A conversational agent for Computational Thinking Development using Scratch	Pensamiento computacional con ScratchThAI (Katchapakirin, K., & Anutariya, C, 2018).	Scopus
Kohen-Vacs, D., & Milrad, M.	2019	Computational thinking education for in-service elementary swedish teachers: Their perceptions and implications for competence development	Competencias y habilidades de los maestros en CT (Kohen-Vacs, D., & Milrad, M, 2019).	Scopus
Kong, S. -, & Wang, Y. -.	2019	Assessing programming concepts in the visual block-based programming course for primary school students	Importancia de la programación basada en bloques en el aprendizaje de programación (Kong, S. -, & Wang, Y. -, 2019)	Scopus
Kong, S. C., & Wang, Y. Q.	2021	Item response analysis of computational thinking practices: Test characteristics and students' learning abilities in visual programming contexts	Ejercicios del pensamiento computacional: depurar, reutilizar, abstraer (Kong, S. C., & Wang, Y. Q, 2021).	Scopus.
Kraleva, R., KraleV, V., & Kostadinova, D.	2019	A methodology for the analysis of block-based programming languages appropriate for children	Uso de Scratch y Code.org, lenguajes de bloques como impulso de la educación (Kraleva, R., KraleV, V., & Kostadinova, D, 2019).	Scopus.
Ntourou, V., Kalogiannakis, M., & Psycharis, S.	2021	A Study of the Impact of Arduino and Visual Programming In Self-Efficacy, Motivation, Computational Thinking and 5th Grade Students' Perceptions on Electricity	Uso de Arduino, Scratch y pensamiento computacional (Ntourou, V., Kalogiannakis, M., & Psycharis, S, 2021).	Scopus
Panskyi, T., & Rowińska	2021	A Holistic Digital Game-Based Learning Approach to Out-of-	Programación visual, robótica, electrónica y pensamiento	Scopus

		School Primary Programming Education	computacional (Panskyi, T., & Rowińska, 2021).	
Panskyi, T., Rowinska, Z., & Biedron, S.	2019	Out-of-school assistance in the teaching of visual creative programming in the game-based environment – Case study: Poland	Programación creativa en el entorno basado en juegos (., & Biedron, S, 2019)	Scopus.
Plaza, P., Castro, M., Saez-Lopez, J. M., Sancristobal, E., Gil, R., Menacho, A., . . . Ruiperez-Valiente, J. A.	2021	Promoting computational thinking through visual block programming tolos	Programación de bloques visuales en el aula (Plaza, P., Castro, M., Saez-Lopez, J. M., Sancristobal, E., Gil, R., Menacho, A., . . . Ruiperez-Valiente, J. A, 2021).	IEEE
Plaza, P., Peixoto, A., Sancristobal, E., Castro, M., Blazquez, M., Menacho, A., . . . Lopez-Rey, A.	2020	Visual block programming languages and their use in educational robotics	Programación visual, herramientas de robótica educativa (Plaza, P., Peixoto, A., Sancristobal, E., Castro, M., Blazquez, M., Menacho, A., . . . Lopez-Rey, A, 2020)	IEEE.
Polat, E., Hopcan, S., Kucuk, S., & Sisman, B	2021	A comprehensive assessment of secondary school students' computational thinking skills	Desempeño del pensamiento computacional en estudiantes de escuela secundaria (Polat, E., Hopcan, S., Kucuk, S., & Sisman, B, 2021).	British Educational Research Association
Rahman, M. M., & Paudel, R.	2019	Visual programming and interactive learning based dynamic instructional approaches to teach an introductory programming course	Educación en ciencias de la computación y retención de estudiantes (Rahman, M. M., & Paudel, R, 2019).	Scopus
Sáez López, J. M., Otero, R. B., & De Lara García-Cervigón, S.	2021	Introducing robotics and block programming in elementary education	Importancia de la programación visual de bloques en la educación primaria (Sáez López, J. M., Otero, R. B., & De Lara García-Cervigón, S, 2021).	Scopus
Sáez-López, J. -, Sevillano-García, M. -, & Vazquez-Cano, E.	2019	The effect of programming on primary school students' mathematical and scientific understanding: educational use of mBot	Lenguajes de programación visual como herramienta de educación en jóvenes estudiantes (Sáez-López, J. -, Sevillano-García, M. -, & Vazquez-Cano, E, 2019).	Scopus
Scullard, S., Tsibolane, P., & Garbutt, M.	2019	The role of scratch visual programming in the development of computational thinking of non-is majors	Uso de Scratch en la abstracción para el desarrollo del pensamiento computacional (Scullard, S., Tsibolane, P., & Garbutt, M, 2019).	Scopus
Shanmugam, L., Yassin, S. F., & Khalid, F.	2019	Enhancing students' motivation to learn computational thinking through mobile application development module (M-CT)	Desarrollo de aplicaciones para la motivación de estudiantes hacia el CT (Shanmugam, L., Yassin, S. F., & Khalid, F, 2019).	Scopus
Sherwin, J., Wang, Y., & Yang, Y.	2021	Build Bridges between Play, Computation, Visual Programming, and Text-based Programming in Early Childhood Education	Programación visual y lógica computacional (Sherwin, J., Wang, Y., & Yang, Y, 2021).	Scopus
Sung, Y. -, & Jeong, Y. -.	2019	Development and application of programming education model based on visual thinking strategy for pre-service teachers	Pensamiento visual para maestros (Sung, Y. -, & Jeong, Y. -, 2019)	Scopus

## 5. CONCLUSIONES

Desde la perspectiva del uso de la programación visual como herramienta pedagógica en la enseñanza y aprendizaje del pensamiento computacional, este ejercicio de documentación destacó resultados comunes a la mayoría de experiencias, siendo quizá el más importante el hecho que los estudiantes hayan reaccionado positivamente ante tareas de análisis y aplicación de soluciones en ambientes de trabajo colaborativo con entusiasmo, atraídos a resolver problemas y retos con ayuda de elementos visuales.

De igual forma, se evidencia la amplia aceptación que tiene la herramienta basada en bloques Scratch en los componentes curriculares de muchos países, para estudiantes de grados K-12 principalmente.

Las herramientas de programación basadas en bloques han sido comunes en muchas de las prácticas experimentales, en búsqueda de soluciones a retos mentales como mecanismo de prueba, con uso de diversas escalas estadísticas de respuesta, con criterio de edad, género o nivel académico. Varias de las conclusiones de estudios mostraron que los jóvenes resuelven retos con uso de software complejo, a la altura del que están acostumbrados a ver o usar, el obstáculo de la complejidad quizás se sublima con el hecho de ser usuarios nativos digitales y usar software de bloques asociado al ejercicio lúdico; tal ejercicio los ha conducido a fortalecer el trabajo en equipo, la abstracción y la comunicación desde edades tempranas con ingredientes muy importantes y deseables: la sensación de diversión, de bienestar, de reconocimiento propio y aceptación social, de motivación en el ámbito educativo.

Por otra parte, se evidenció que no existen consensos en la forma en la que se construyen e implementan las evaluaciones a los estudiantes en programas de enseñanza y aprendizaje del pensamiento computacional en general. Se discute el uso de cuestionarios de selección múltiple o el uso de escalas de evaluación de habilidades relacionadas con el pensamiento computacional (desarrollo de pensamiento lógico, abstracción y generalización), situación que por lo tanto revela la necesidad de aportar formas y aplicaciones en la construcción de un marco de referencia, así como una metodología de evaluación que permita comparaciones entre grupos heterogéneos de estudiantes.

En última instancia, resulta de especial atención el hecho de reconocer las dificultades propias de los países latinoamericanos, con brechas sociales enormes y que enfrentan la imposibilidad de contar con herramientas tecnológicas en entornos educativos ideales en el corto plazo para la mayoría de su población en edad escolar, lo que demanda la implementación de estrategias que se enfoquen en el desarrollo del pensamiento computacional, no solo desde la mera perspectiva de aumentar el conocimiento, sino la de enseñar a los estudiantes a lidiar con la complejidad, el desarrollo de la lógica y la abstracción a través de actividades didácticas, como juegos de mesa o cualquier otra herramienta lúdica tradicional, que motive su interés en desarrollar formas de vencer retos y problemas dentro de entornos colaborativos; objetivos que parecen poco, pero que en realidad constituyen todo un cambio de paradigma en los procesos de enseñanza y aprendizaje del pensamiento computacional.

## REFERENCIAS

- Ageyi, E. E. Y. F., Laru, J., & Mäkitalo, K. (2019). Designing unplugged activities for learning computational thinking in the context K-2 pupils' afterschool coding club. *CoolThink@ JC*, 70.
- Allsopp, B. B., & Misfeldt, M. (2019). Visual representations supporting implementation of a K12 programming curriculum in open and democratic educational institutions. Paper presented at the *Proceedings of the European Conference on e-Learning, ECEL*, 2019-November 24-31. doi:10.34190/EEL.19.072 Retrieved from [www.scopus.com](http://www.scopus.com)
- Alves, N. D. C., Von Wangenheim, C. G., & Hauck, J. C. (2019). Approaches to assess computational thinking competences based on code analysis in K-12 education: A systematic mapping study. *Informatics in Education*, 18(1), 17.
- Avent, R. (2017). *La riqueza de los humanos: El trabajo en el siglo XXI*. Editorial Ariel.
- Basogain, X., Olabe, M. Á., Olabe, J. C., & Rico, M. J. (2018). Computational Thinking in pre-university Blended Learning classrooms. *Computers in Human Behavior*, 80, 412-419.
- Blocky Team. (2022). *Blockly | Google Developers*. 2022. <https://developers.google.com/blockly>
- Brennan, K., & Resnick, M. (2012). Using artifact-based interviews to study the development of computational thinking in interactive media design. Paper presented at annual American Educational Research Association meeting, Vancouver, BC, Canada.
- Csapó, G. (2019). Placing event-action-based visual programming in the process of computer science education. *Acta Polytechnica Hungarica*, 16(2), 35-57. doi:10.12700/APH.16.2.2019.2.3
- CSTA, & ECEP Alliance. (2020). *2020 State of Computer Science Education: Illuminating Disparities*. Recuperado de <https://advocacy.code.org/stateofcs>
- CTE. 2019. *CoolThink@JC – Computational Thinking Education Programme*. Proceedings of International Conference on Computational Thinking Education 2019. 13-15 June 2019. Hong Kong. Recuperado de: <https://www.coolthink.hk/en/teaching-resources/>
- Da Cruz Alves, N., Gresse Von Wangenheim, C., & Hauck, J. C. R. (2019). Approaches to assess computational thinking competences based on code analysis in K-12 education: A systematic mapping study. *Informatics in Education*, 18(1), 17-39. doi:10.15388/infedu.2019.02
- Eguíluz, A., Guenaga, M., Garaizar, P., & Olivares-Rodríguez, C. (2020). Exploring the progression of early programmers in

- a set of computational thinking challenges via clickstream analysis. *IEEE Transactions on Emerging Topics in Computing*, 8(1), 256-261. doi:10.1109/TETC.2017.2768550
- Feurzeig, W., Papert, S., & Solomon, C. (1967). LOGO (version 1.0) [Software]. Instituto Tecnológico de Massachusetts (MIT). [https://el.media.mit.edu/logo-foundation/what\\_is\\_logo/index.html](https://el.media.mit.edu/logo-foundation/what_is_logo/index.html)
- Fragapane, V., & Standl, B. (2021). Work in progress: Creative coding and computer science education - from approach to concept. Paper presented at the *IEEE Global Engineering Education Conference, EDUCON, , 2021-April* 1233-1236. doi:10.1109/EDUCON46332.2021.9453951 Retrieved from [www.scopus.com](http://www.scopus.com)
- Gao, P., Lu, M., Zhao, H., & Li, M. (2019). A new teaching pattern based on PBL and visual programming in computational thinking course. Paper presented at the 14th International Conference on Computer Science and Education, ICCSE 2019, 304-308. doi:10.1109/ICCSE.2019.8845503 Retrieved from [www.scopus.com](http://www.scopus.com)
- Huang, Y. -, Huang, Y. -, & Starcic, A. I. (2020). Enhancing students' learning outcomes of a STEAM permutations course through a game based visual programming environment with qualifying rank strategy doi:10.1007/978-3-030-63885-6\_11 Retrieved from [www.scopus.com](http://www.scopus.com)
- ISTE. ISTE Standards: educators. International society for technology in education. ISTE; 2016. Retrieved from: <https://www.iste.org/es/standards/iste-standards-for-teachers>
- Jamil, H. M. (2017). *Visual computational thinking using patch* doi:10.1007/978-3-319-66733-1\_23 Retrieved from [www.scopus.com](http://www.scopus.com)
- Jiang, B., & Li, Z. (2021). Effect of scratch on computational thinking skills of chinese primary school students. *Journal of Computers in Education*, 8(4), 505-525. doi:10.1007/s40692-021-00190-z
- João, P., Nuno, D., Fábio, S. F., & Ana, P. (2019). A cross-analysis of block-based and visual programming apps with computer science student-teachers. *Education Sciences*, 9(3) doi:10.3390/educsci9030181
- Katchapakirin, K., & Anutariya, C. (2018). An architectural design of ScratchThAI A conversational agent for computational thinking development using scratch. Paper presented at the *ACM International Conference Proceeding Series*, doi:10.1145/3291280.3291787 Retrieved from [www.scopus.com](http://www.scopus.com)
- Kohen-Vacs, D., & Milrad, M. (2019). Computational thinking education for in-service elementary swedish teachers: Their perceptions and implications for competence development. Paper presented at the *Proceedings of International Conference on Computational Thinking Education*, 109-112. Retrieved from [www.scopus.com](http://www.scopus.com)
- Kong, S. C., & Abelson, H. (2019). Computational thinking education (p. 382). Springer Nature.
- Kong, S. -, & Wang, Y. -. (2019). Assessing programming concepts in the visual block-based programming course for primary school students. Paper presented at the *Proceedings of the European Conference on e-Learning, ECEL, , 2019-November* 294-302. doi:10.34190/EEL.19.035 Retrieved from [www.scopus.com](http://www.scopus.com)
- Kong, S. C., & Wang, Y. Q. (2021). Item response analysis of computational thinking practices: Test characteristics and students' learning abilities in visual programming contexts. *Computers in Human Behavior*, 122 doi:10.1016/j.chb.2021.106836
- Kraleva, R., KraleV, V., & Kostadinova, D. (2019). A methodology for the analysis of block-based programming languages appropriate for children. *Journal of Computing Science and Engineering*, 13(1), 1-10. doi:10.5626/JCSE.2019.13.1.1
- Lowe, T. A., & Brophy, S. P. (2019). Identifying computational thinking in storytelling literacy activities with scratch jr. (RTP). Paper presented at the ASEE Annual Conference and Exposition, Conference Proceedings, Retrieved from [www.scopus.com](http://www.scopus.com)
- Martínez-Zarzuelo, A., Rodríguez, J. M., Roanes, E., Fernández, M. J., & Cujó, J. (2016). Aplicación y evaluación de una metodología de programación visual para el desarrollo de competencias matemáticas.
- Mecca, G., Santoro, D., Sileno, N., & Veltri, E. (2021). Spice-up your coding lessons with the ACME approach. Paper presented at the CEUR Workshop Proceedings, 2994 Retrieved from [www.scopus.com](http://www.scopus.com).
- Míguez, Á. R., & Neira, R. H. (2020). Creación de un videojuego educativo sobre células para la enseñanza de la programación. *IE Comunicaciones: Revista Iberoamericana de Informática Educativa*, (32), 15-24.
- MIT Center for Mobile Learning. (2013). *MIT App Inventor* (No. 2). <http://appinventor.mit.edu/>
- MIT Media Lab. (2008). *Scratch*. <https://scratch.mit.edu/>
- Mönig, J., & Harvey, B. (2022). *Snap! Build Your Own Blocks*. <https://snap.berkeley.edu/credits>
- Ntourov, V., Kalogiannakis, M., & Psycharis, S. (2021). A study of the impact of arduino and visual programming in self-efficacy, motivation, computational thinking and 5th grade students' perceptions on electricity. *Eurasia Journal of Mathematics, Science and Technology Education*, 17(5), 1-11. doi:10.29333/ejmste/10842
- Panskyi, T., & Rowińska, Z. (2021). A holistic digital game-based learning approach to out-of-school primary

- programming education. *Informatics in Education*, 20(2), 1-22. doi:10.15388/infedu.2021.12
- Panskyi, T., Rowinska, Z., & Biedron, S. (2019). Out-of-school assistance in the teaching of visual creative programming in the game-based environment – case study: Poland. *Thinking Skills and Creativity*, 34 doi:10.1016/j.tsc.2019.100593
- Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. Brighton: Harvester Press.
- Plaza, P., Castro, M., Saez-Lopez, J. M., Sancristobal, E., Gil, R., Menacho, A., . . . Ruiperez-Valiente, J. A. (2021). Promoting computational thinking through visual block programming tools. Paper presented at the *IEEE Global Engineering Education Conference, EDUCON, , 2021-April* 1131-1136. doi:10.1109/EDUCON46332.2021.9453903 Retrieved from [www.scopus.com](http://www.scopus.com)
- Plaza, P., Peixoto, A., Sancristobal, E., Castro, M., Blazquez, M., Menacho, A., . . . Lopez-Rey, A. (2020). Visual block programming languages and their use in educational robotics. Paper presented at the *IEEE Global Engineering Education Conference, EDUCON, , 2020-April* 457-464. doi:10.1109/EDUCON45650.2020.9125219 Retrieved from [www.scopus.com](http://www.scopus.com)
- Polat, E., Hopcan, S., Kucuk, S., & Sisman, B. (2021). A comprehensive assessment of secondary school students' computational thinking skills. *British Journal of Educational Technology*, 52(5), 1965-1980. doi:10.1111/bjet.13092
- Rahman, M. M., & Paudel, R. (2019). Visual programming and interactive learning based dynamic instructional approaches to teach an introductory programming course. Paper presented at the *Proceedings - Frontiers in Education Conference, FIE, , 2018-October* doi:10.1109/FIE.2018.8658581 Retrieved from [www.scopus.com](http://www.scopus.com)
- Rose, S. P., Jay, T., & Jacob Habgood, M. P. (2019). Using pirate plunder to develop children's abstraction skills in scratch. Paper presented at the *Conference on Human Factors in Computing Systems - Proceedings*, doi:10.1145/3290607.3312871 Retrieved from [www.scopus.com](http://www.scopus.com).
- Sáez López, J. M., & Cózar Gutiérrez, R. (2017). Programación visual por bloques en Educación Primaria: Aprendiendo y creando contenidos en Ciencias Sociales. *Revista Complutense de Educación*, 28(2), 409-426.
- Sáez López, J. M., Otero, R. B., & De Lara García-Cervigón, S. (2021). Introducing robotics and block programming in elementary education. [La aplicación de la robótica y programación por bloques en la enseñanza elemental] *RIED-Revista Iberoamericana De Educacion a Distancia*, 24(1), 95-113. doi:10.5944/ried.24.1.27649
- Sáez-López, J. -, Sevillano-García, M. -, & Vazquez-Cano, E. (2019). The effect of programming on primary school students' mathematical and scientific understanding: Educational use of mBot. *Educational Technology Research and Development*, 67(6), 1405-1425. doi:10.1007/s11423-019-09648-5
- Scullard, S., Tsibolane, P., & Garbutt, M. (2019). The role of scratch visual programming in the development of computational thinking of non-is majors. Paper presented at the *Proceedings of the 23rd Pacific Asia Conference on Information Systems: Secure ICT Platform for the 4th Industrial Revolution, PACIS 2019*, Retrieved from [www.scopus.com](http://www.scopus.com)
- Selby, C., & Woollard, J. (2013). Computational thinking: the developing definition.
- Shanmugam, L., Yassin, S. F., & Khalid, F. (2019). Enhancing students' motivation to learn computational thinking through mobile application development module (M-CT). *International Journal of Engineering and Advanced Technology*, 8(5), 1293-1303. Retrieved from [www.scopus.com](http://www.scopus.com)
- Sherwin, J., Wang, Y., & Yang, Y. (2021). Build bridges between play, computation, visual programming, and text-based programming in early childhood education. Paper presented at the *SIGCSE 2021 - Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 1318. doi:10.1145/3408877.3439656 Retrieved from [www.scopus.com](http://www.scopus.com)
- Šiaulyš, T. (2021). Developing interactive visualizations focusing on computational thinking in K-12 computer science education. Paper presented at the *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, 680-681. doi:10.1145/3456565.3460025 Retrieved from [www.scopus.com](http://www.scopus.com)
- Sun, L., Hu, L., & Zhou, D. (2021). Which way of design programming activities is more effective to promote K-12 students' computational thinking skills? A meta-analysis. *Journal of Computer Assisted Learning*, 37(4), 1048-1062.
- Sung, Y. -, & Jeong, Y. -. (2019). Development and application of programming education model based on visual thinking strategy for pre-service teachers. *Universal Journal of Educational Research*, 7(5), 42-53. doi:10.13189/ujer.2019.071507
- Svelto Tech. (2022). *Svelto*. 2022. <https://www.svelto.tech/index.html#>
- Tien, H. -, & Yen, J. -. (2019). Computational thinking in STEM task design: Authentic, useful, experiential, and visual. Paper presented at the *Proceedings of International Conference on Computational Thinking Education*, 135-140. Retrieved from [www.scopus.com](http://www.scopus.com)
- Tsai, M. J., Liang, J. C., & Hsu, C. Y. (2021). The computational thinking scale for computer literacy education. *Journal of Educational Computing Research*, 59(4), 579-602.

Tseng, C. Y., Doll, J., & Varma, K. (2019). Exploring evidence that board games can support computational thinking. In 3rd International Conference on Computational Thinking Education, CTE 2019 (pp. 61-64). The Education University of Hong Kong.

Vera, M. D. M. S. (2020). La robótica, la programación y el pensamiento computacional en la educación infantil. *Revista Infancia, Educación y Aprendizaje*, 7(1), 209-234.

Vourletsis, I., & Politis, P. (2020). Effects of a computational thinking experimental course on students' perceptions of their problem-solving skills. Paper presented at the ACM International Conference Proceeding Series, 14-20. doi:10.1145/3383923.3383935 Retrieved from [www.scopus.com](http://www.scopus.com)

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

WU, S. Y. (2019). The Design and Development of Coding Poker Cards. *CoolThink@ JC*, 46.

Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. *Revista de Educación a Distancia (RED)*, (46).

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through scratch in K-9. *Computers and Education*, 141 doi:10.1016/j.compedu.2019.103607