

Relaciones entre la matemática, el pensamiento algorítmico y el pensamiento computacional

Relationship between mathematics, algorithmic thinking and computational thinking

Eduardo Adam Navas López

RESUMEN

En un mundo cada vez más influenciado por las herramientas computacionales disponibles y la aceleración en el desarrollo de estas, es necesario también incrementar la investigación epistemológica sobre las interacciones entre la matemática y las ciencias de la computación. Por ello se considera relevante estudiar las relaciones entre la habilidad de algoritmizar, el pensamiento matemático, el pensamiento algorítmico y el pensamiento computacional. Sin embargo, múltiples definiciones de pensamiento computacional no incorporan explícitamente al pensamiento algorítmico ni enfatizan su dimensión matemática. La proposición que se plantea en este ensayo científico es que existen relaciones, a veces implícitas, muy importantes entre los conceptos de algoritmización, pensamiento matemático, pensamiento algorítmico y pensamiento computacional. Y el objetivo es identificar las relaciones entre esos conceptos, y con ello fortalecer la literatura científica que busca abordarlos de forma integrada. Se concluye que el pensamiento algorítmico es una parte fundamental del pensamiento computacional, pero también es un tipo de pensamiento matemático; que la habilidad de algoritmizar es básica en la matemática y es fundamental en el pensamiento algorítmico y en el computacional, y que el pensamiento computacional brinda importantes ayudas en la exploración y descubrimiento de la matemática.

Palabras clave: Algoritmos, enseñanza de la matemática, estilos de pensamiento, habilidades específicas, pensamiento abstracto.

ABSTRACT

In a world increasingly influenced by the available computational tools, and the acceleration in their development, it is also necessary to increase epistemological research on the interactions between mathematics and computer science. Therefore, it is considered relevant to study the relationship between the ability to algorithmize, mathematical thinking, algorithmic thinking and computational thinking. However, multiple definitions of computational thinking do not explicitly incorporate algorithmic thinking or emphasize its mathematical dimension. The proposition raised in this scientific essay is that there are very important relationships, sometimes implicit, between the concepts of algorithmization, mathematical thinking, algorithmic thinking and computational thinking. And the aim is to identify the relationship between these concepts, and thereby strengthen the scientific literature that seeks to address them in an integrated manner. It is concluded that algorithmic thinking is a fundamental part of computational thinking, but it is also a type of mathematical thinking; that the ability to algorithmize is basic in mathematics and is fundamental in algorithmic and computational thinking; and that computational thinking provides important aids in exploring and discovering mathematics.

Keywords: Algorithms, mathematics teaching, thinking styles, specific skills, abstract thinking.

INTRODUCCIÓN

Según Pyzara (2012), la naturaleza de la matemática es dual, ya que comprende tanto aspectos conceptuales como algorítmicos. Estas dos facetas están intrínsecamente vinculadas y se complementan mutuamente, siendo igualmente esenciales, como sostiene Sfard (1991). En la opinión de Davis y McGowen (2001), “si las definiciones son la base de nuestra fundamentación matemática, entonces los algoritmos son los ladrillos en el puente de nuestra ruta matemática” (p. 311). Rasmussen et al. (2005) afirman que las habilidades de simbolizar, algoritmizar y definir son las prácticas centrales que abarcan todos los dominios matemáticos.

Se reconoce la presencia de un tipo particular de pensamiento denominado *pensamiento algorítmico*, estrechamente ligado a las habilidades matemáticas, incluida la habilidad de algoritmizar (Stephens y Kadijevich, 2020). Este tipo de pensamiento se considera de gran relevancia tanto en el ámbito educativo (Doğan, 2020) como en el ámbito laboral y económico (Hurlburt, 2018). Se afirma que el pensamiento algorítmico es una arista importante del pensamiento matemático (Modeste, 2012b; Lockwood et al., 2016) y de la matemática en sí misma (Lovász, 1988; Sfard, 1991; Pyzara, 2012). Después de todo, “la matemática es una ciencia de patrones y orden, [y] su dominio no son moléculas o células, sino los números, el azar, la forma, los algoritmos, y el cambio” (NRC, 1989, p. 31). El pensamiento algorítmico es de gran relevancia no solamente en el ámbito de la matemática sino también en diversos aspectos de la vida cotidiana en la actualidad (Sadykova y Usolzev, 2018; Sadykova e Il’bahtin, 2019). Algunas publicaciones, como las de Kátai (2015), Sadykova e Il’bahtin (2019) y Doğan (2020), afirman que el pensamiento algorítmico es una habilidad importante en una sociedad basada en la información, que todas las personas deberían poseer.

Este pensamiento algorítmico constituye un componente esencial de otro enfoque cognitivo conocido como *pensamiento computacional* (Selby y Woollard, 2013; Kalelioğlu et al., 2016; Korkmaz et al., 2017). Del mismo modo, Wing (2006) y Grover y Pea (2018) destacan que el pensamiento computacional es una destreza crucial no solo para los profesionales de las ciencias de la computación sino para todas las personas. Según Wing (2006, 2008, 2017) y Barr et al. (2011), el pensamiento computacional debería ser incluido como contenido esencial obligatorio desde la primaria para preparar a la población ante las demandas del mundo tecnológico en el que ya vivimos.

Eduardo Adam Navas López. Profesor universitario de la Facultad de Ciencias Naturales y Matemática de la Universidad de El Salvador. Es Maestro en Didáctica de la Matemática por la Universidad de El Salvador y Licenciado en Ciencias de la Computación por la Universidad Centroamericana “José Simeón Cañas”. Ha participado como ponente en diversos congresos internacionales sobre didáctica de la matemática y ciencias de la computación. Sus áreas principales de interés incluyen el pensamiento algorítmico, el pensamiento computacional, el arte matemático generado por computadora, los modelos cognitivos de comprensión de conceptos matemáticos, la etnomatemática y sus intersecciones. Correo electrónico: eduardo.navas@ues.edu.sv. ID: <https://orcid.org/0000-0003-3684-2966>.

En esta línea de pensamiento, D'Ambrosio (1999) propone incluso un nuevo *trivium*, un nuevo tipo de currículo universal para la educación de las nuevas generaciones de un mundo que se mueve hacia una civilización planetaria. Este nuevo plan de estudios propuesto se compone de las competencias *literacy*, *matheracy* y *technoracy*. Por *literacy*, D'Ambrosio (1999) entiende la capacidad de comprender e interpretar la información, gráficos, tablas, códigos, etc.; por *matheracy* entiende la habilidad de llegar a conclusiones a partir de datos y cálculos, realizar inferencias y plantear hipótesis, y por *technoracy* entiende la capacidad de comprender las ideas básicas, posibilidades y riesgos de los dispositivos tecnológicos, así como los aspectos morales inherentes de su uso.

Schoenfeld (2013) afirma que la presencia de las herramientas computacionales tiene el potencial de cambiar radicalmente la forma del conocimiento matemático al que el alumnado tiene acceso en las aulas de clase, y los modos en los que pueden operar con ese conocimiento. Por ello, en la actualidad existe una clara tendencia en la investigación de la enseñanza integrada del pensamiento computacional con la matemática (Broley et al., 2017; Broley et al., 2018; Israel y Lash, 2019; Waterman et al., 2018).

Por otro lado, la investigación de Lockwood et al. (2016) concluye que la noción de pensamiento computacional no está clara en el lenguaje cotidiano de los matemáticos profesionales, pero la noción de pensamiento algorítmico sí tiene presencia entre las habilidades que ellos valoran en la práctica de hacer matemática. En la encuesta desarrollada por Buteau et al. (2014) en personal académico se descubrió que mientras que el 43% del personal matemático encuestado reportó usar programación de computadoras en sus investigaciones, solo el 18% indicó que la usan en su práctica docente. Esto indica que la construcción de algoritmos es una habilidad profesional que forma parte del repertorio de casi la mitad del personal matemático. Sin embargo, su inclusión en la formación de los estudiantes de matemática se limita a menos de una quinta parte.

Además, las aprobaciones en cursos introductorios de programación de computadoras a nivel mundial son en general muy bajas (McCracken et al., 2001; Guzdial y Soloway, 2002; Kurland et al., 2002; Gayo et al., 2003; Kinnunen y Malmi, 2006). Estos malos resultados se deben a diferentes razones, como las altas demandas cognitivas (Arévalo-Mercado et al., 2019; Kurland et al., 2002) y de abstracción (Dijkstra, 1974; Ramadhan, 2000), la falta de contenidos curriculares apropiados en la matemática escolar para desarrollar las habilidades necesarias (Dijkstra, 1974; Modeste, 2016), problemas didácticos (Doğan, 2020) y obstáculos epistemológicos propios de la interacción entre el álgebra escolar y la programación de computadoras (Kilhamn y Bråting, 2019; Modeste, 2012b, 2016). Blanco-Hamad et al. (2016) identificaron que los estudiantes no perciben la algoritmización como una parte fundamental del proceso

de programación de computadoras. La mayoría de los estudiantes no realizan una modelización previa utilizando algún pseudocódigo o diagramas de flujo; en cambio, creen que la situación problemática debe resolverse directamente en un lenguaje de programación. Esta falta de enfoque dificulta notablemente el desarrollo del pensamiento algorítmico, ya que los estudiantes no le otorgan la importancia requerida.

En lo que se refiere a la integración de estos conceptos en la literatura, no todos los investigadores están explícitamente de acuerdo con la múltiple inclusión que plantean Stephens y Kadijevich (2020) de algoritmización en pensamiento algorítmico y pensamiento algorítmico en pensamiento computacional. Por ejemplo las descripciones de pensamiento computacional de Wing (2006), Grover y Pea (2013), y Weintrop et al. (2016) no mencionan al pensamiento algorítmico. La habilidad de algoritmizar tampoco está explícita en muchas de las definiciones de pensamiento algorítmico, como en Futschek (2006) y en Sadykova y Usolzev (2018).

Modeste (2016) argumenta que la investigación epistemológica sobre las interacciones entre la matemática y las ciencias de la computación es crucial para enriquecer la investigación didáctica sobre la enseñanza y el aprendizaje de la matemática en la actualidad, influidas por el avance de las ciencias de la computación. De manera similar, la revisión bibliográfica de Czerkowski y Lyman (2015) muestra que falta mucho que investigar sobre la noción de pensamiento computacional en las dimensiones curricular, meta-cognitiva y evaluativa, especialmente en su interacción con otras disciplinas además de las ciencias de la computación. Por ello se considera relevante estudiar las relaciones epistemológicas entre la habilidad de algoritmizar, el pensamiento matemático, el pensamiento algorítmico y el pensamiento computacional.

DESARROLLO

La proposición que se plantea en este ensayo científico es que existen relaciones muy importantes entre los conceptos de algoritmización, pensamiento matemático (PM), pensamiento algorítmico (PA) y pensamiento computacional (PC), así que el objetivo del mismo es identificar las relaciones entre dichos conceptos –que a pesar de estar fuertemente relacionados se suelen abordar de forma aislada– y con ello fortalecer la literatura científica que busca abordarlos de forma integrada.

Esta sección está organizada de la siguiente manera:

1. Algunas definiciones de algoritmo y de la habilidad de algoritmizar.
2. Compendio de tipos de pensamiento asociados con la matemática.
3. El origen de la noción de PA.
4. Algunas definiciones conceptuales de PA.
5. Algunas definiciones conceptuales de PC.
6. El rol del PA en el PC.
7. La interacción entre el PC y el desarrollo del PM.

8. Aclaraciones necesarias para evitar posibles confusiones entre estos conceptos.
9. Diferencias específicas entre el PA y el PC.

Algoritmo, algoritmizar y algoritmización

Lo primero que hay que mencionar es el hecho de que los algoritmos son objetos matemáticos distinguibles de su implementación como programa de computadora (Lovász, 1988). Para Pyzara (2012), un algoritmo es “un esquema o descripción precisa de la solución a un problema expresado con operaciones que un ejecutante entiende y es capaz de llevar a cabo” (p. 53). Por otra parte, Knuth (1974) define algoritmo como “una secuencia precisa y definida de reglas que dicen cómo producir información de salida especificada a partir de información de entrada dada en un número finito de pasos” (p. 323). Misfeldt y Ejsing-Duun (2015) definen un algoritmo como una descripción sistemática de estrategias para la construcción y resolución de problemas, y descripción de relaciones causa-efecto y eventos. De manera similar, Modeste (2012a) afirma que un algoritmo es “un procedimiento para resolver problemas, que puede ser aplicado a una familia de instancias del problema y que produce, en un número finito de etapas constructivas, no ambiguas y organizadas, la respuesta al problema para toda instancia de la familia” (p. 246).

Algoritmizar, para Delgado (1998), es plantear una sucesión estricta de operaciones matemáticas que describan un procedimiento conducente a la solución de determinado problema. Y consecuentemente el sustantivo “algoritmización” se entiende como la acción o efecto de algoritmizar.

Tipos de pensamiento matemático

Existen diferentes modos de pensamiento, estos son aspectos individuales de habilidades cognitivas de las personas que aparecen en diferentes esferas de actividades mentales y actividades basadas en objetos (Sadykova y Usolzev, 2018).

Knuth (1985) llega a la conclusión de que no existe un único tipo de PM, sino que los matemáticos utilizan una variedad de enfoques mentales, algunos de los cuales se superponen con los utilizados en las ciencias de la computación. Sin embargo, a pesar de parecer un término muy común, no se encontró una definición conceptual u operacional reciente de PM, sin embargo, los siguientes tipos de pensamiento pueden considerarse modos de PM según la literatura científica, aunque no son exclusivos de los matemáticos:

- Pensamiento cuantitativo (Ruiz y Valdemoros, 2004).
- Pensamiento numérico y aritmético (Gallardo y González, 2007; Congacha et al., 2018).
- Pensamiento crítico (Kalelioğlu y Gülbahar, 2014; y Doleck et al., 2017).
- Pensamiento creativo (Doleck et al., 2017; Hidayat et al., 2018).

- Pensamiento lógico (Coscarelli et al., 1976).
- Pensamiento algebraico (Kilhamn y Bråting, 2019).
- Pensamiento geométrico (Thom y McGarvey, 2015).
- Pensamiento combinatorio (Fernández, 2013).
- Pensamiento estadístico (Batanero, 2013).
- Pensamiento matemático avanzado (Tall, 2002; Rasmussen et al., 2005).
- Pensamiento variacional (Martínez-López y Gualdrón-Pinto, 2018).
- Pensamiento algorítmico (Lockwood et al., 2016; Sadykova y Usolzev, 2018).

Muchos de estos modos o tipos de pensamiento están interrelacionados en diferentes áreas de la matemática y en otros contextos.

Orígenes del pensamiento algorítmico en la literatura científica

Dijkstra (1974) argumenta que la mayor parte del trabajo matemático comparte características comunes con la programación de computadoras. En la misma línea, Knuth (1974) reflexiona sobre la relación entre la matemática y las –por entonces todavía jóvenes– ciencias de la computación, y menciona algunos de los múltiples aportes que se han hecho mutuamente entre científicos de la computación haciendo matemática y matemáticos haciendo ciencias de la computación, a veces sin darse cuenta. Más adelante, Lovász (1988) escribió un ensayo en el que presenta su punto de vista sobre la relación entre el lado algorítmico y el lado estructural de la matemática. Aquí aparece la misma dualidad algoritmo-concepto de la matemática que Sfard (1991) y Pyzara (2012) describen.

Knuth (1985) también reflexiona sobre la relación entre lo que él llama (sin definirlo explícitamente) el *pensamiento algorítmico* y el *pensamiento matemático*, y sugiere que se necesita una investigación más profunda para aclarar las diferencias entre estos dos tipos de pensamiento. Modeste (2012b) está de acuerdo con Knuth (1974; 1985) en que el PA puede verse como parte del PM y con que se ha demostrado cómo este PA influye en el PM.

Así que podemos convencernos de que existen modos de pensamiento (o razonamiento) que son comunes entre los matemáticos y los científicos de la computación. Y es de esta intersección entre la matemática y las ciencias de la computación de donde surge la noción moderna de PA.

Algunas definiciones conceptuales de pensamiento algorítmico

Una definición general y sencilla de PA es que se trata de “una forma lógica y organizada de pensar usada para descomponer un objetivo complicado en una serie de pasos (ordenados) o etapas usando las herramientas disponibles” (Lockwood et al., 2016, p. 1591).

Existen otras definiciones más sencillas e informales de PA. Por ejemplo, Kátai (2015) sostiene simplemente que el PA es el pensamiento que está vinculado al concepto de crear y procesar algoritmos, mientras que Misfeldt y Ejsing-Duun (2015) lo definen sucintamente como la habilidad de trabajar con algoritmos. Por otro lado, la definición de Mingus y Grassl (1998) es más detallada y considerablemente más exigente:

El pensamiento algorítmico es un método de pensar y guiar los procesos de pensamiento que utiliza procedimientos paso a paso, requiere entradas y produce salidas, requiere decisiones sobre la calidad y la idoneidad de la información que entra y de la que sale, y monitorea los procesos de pensamiento como un medio para controlar y dirigir el proceso de pensamiento. En esencia, el pensamiento algorítmico es al mismo tiempo un método de pensamiento y un medio para pensar en el propio pensamiento [p. 34].

Con una exigencia similar, Modeste (2012b) coincide en que el PA “sería entonces una forma de abordar un problema tratando de sistematizar su resolución, de cuestionar la forma en que los algoritmos podrían o no resolverlo” (p. 472), y destaca que el PA no se limita únicamente a aprender algoritmos e implementarlos, sino que también involucra su producción, comprensión y estudio. Un poco menos exigente es la definición de Ziatdinov y Musa (2012): “[El PA es] la habilidad de resolver problemas de diversos orígenes, que requieren un plan de acción para lograr el resultado deseado” (p. 1109). Estas dos definiciones enfatizan que se trata de resolución de problemas.

Grozdev y Terzieva (2011) afirman que el PA es una forma de pensar, que proporciona una solución para una tarea específica a través de una sucesión de acciones elementales. Según Hoyles y Noss (2015), el PA se refiere a la tendencia a visualizar las tareas en términos de pasos discretos conectados más pequeños. Estas dos definiciones presentan el problema de referirse a acciones elementales y pasos discretos, los cuales no son absolutos, sino que dependen de la tecnología (o lenguaje) subyacente en el que se exprese el algoritmo.

En la publicación de Sadykova y Usolzev (2018), el PA se define como “un sistema de operaciones mentales y procedimientos con componentes reales y abstractos basados en objetos que conducen al desarrollo de un algoritmo optimizado para la solución de un problema dado” (p. 5). Más adelante, Sadykova e Il’bahtin (2019) afirman que “el pensamiento algorítmico es un sistema de métodos de pensamiento que es necesario para construir una secuencia de resultados intermedios, planificando la estructura de acciones y su implementación, conduciendo al logro de una meta” (p. 421). Estas dos definiciones, como la de Mingus y Grassl (1998), indican que se trata muy específicamente de una forma de pensar.

Más recientemente, Juškevičienė (2020) define sucintamente al PA como la “habilidad de formular problemas que transforman una entrada en la salida deseada usando algoritmos” (p. 148). Aquí se enfatiza la idea de que se trata de formular problemas y

no de resolverlos. Esto tiene sentido ya que al construir un algoritmo no se resuelve el problema, sino que hay que ejecutar el algoritmo para resolver el problema, lo cual se hace en un momento posterior.

La definición de Lockwood et al. (2016) no implica que la persona ejecutante deba conocer y comprender el concepto de algoritmo previamente, es decir, es más cercana a lo que se esperaría de estudiantes que todavía no saben programar o que están comenzando a aprender a programar, a diferencia de la de Mingus y Grassl (1998), que parece describir más bien a expertos en PA. La definición de Lockwood et al. (2016) también aclara el tema de las acciones elementales de las que hablan Grozdev y Terzieva (2011) y Hoyles y Noss (2015), refiriéndose en su lugar a las herramientas disponibles, lo cual es más apropiado ya que deja claro que los pasos dependen de la tecnología concreta en la que se expresan esos pasos, y no que estos son absolutos.

Futschek (2006) afirma que el PA es una habilidad que puede ser desarrollada independientemente de aprender a programar computadoras. Esta hipótesis es explorada por Ziatdinov y Musa (2012) y por Kátai (2015). De hecho, para Ziatdinov y Musa (2012), el desarrollo del PA puede ser iniciado en las clases de matemática de la escuela primaria. Y según Douadi et al. (2012), puede ser desarrollado independientemente de las tecnologías, implementación o lenguajes de programación específicos.

Definiciones conceptuales de pensamiento computacional

Freiman et al. (2018) sostienen que el PC es un concepto relativamente reciente que está ganando protagonismo en los debates educativos, y que el número de publicaciones relacionadas con este tema está en aumento. Investigadores como Grover y Pea (2013), Voogt et al. (2015), Weintrop et al. (2016), Denning (2017), Broley et al. (2017), Freiman et al. (2018), Israel y Lash (2019), etc., identifican los inicios de la noción de PC en el trabajo de Papert (1980), precisamente aplicado para enseñar matemática a niños y niñas.

Para Aho (2012), el PC es “el proceso de pensamiento involucrado en la formulación de problemas tal que su solución puede ser representada como pasos computacionales y algoritmos” (p. 832), mientras que para Wing (2017), el PC “son los procesos de pensamiento involucrados en la formulación de problemas y en la expresión de sus soluciones de tal forma que una computadora –humana o mecánica– puede llevar a cabo efectivamente” (p. 8). De forma similar, Waterman et al. (2018) consideran que el PC “es una forma de pensar que involucra la formulación y descomposición de problemas, y la estructuración y comunicación de las soluciones tal que pueden ser entendidas por humanos y procesadas por máquinas” (p. 283).

Las tres definiciones conceptuales son en esencia equivalentes, pero la de Waterman et al. (2018) es más explícita que la de Aho (2012), y más fácil de entender que la de Wing (2017). Aho (2012) utiliza el término “paso computacional”, que es explicado

en su artículo pero que no es de uso común. En ese sentido es menos explícita que la de Waterman et al. (2018). Por otro lado, Wing (2017) habla de computadoras humanas refiriéndose a cuando una persona ejecuta un algoritmo concreto con precisión y sin errores. Por ello, aunque tal idea es clara, no es tan fácil de entender como la de Waterman et al. (2018). Claro que la definición de Waterman et al. (2018) no aclara de qué clase de humanos y de qué clase de máquinas se trata, por lo que la definición queda en función de los humanos disponibles y las máquinas disponibles.

Sin embargo, la definición de Wing (2017) indica, tal como ella misma lo explica, tres cosas importantes: primero, que las personas pueden computar; segundo, que las personas pueden aprender PC sin una máquina, y tercero, que el PC no trata solamente de resolución de problemas, sino también de (re)formulación de problemas.

Aunque estas definiciones conceptuales son parecidas a las de PA, puede verse que las definiciones de PC indican que las soluciones deben poder ser procesadas o ejecutadas por máquinas, mientras que las definiciones de PA no.

Existen también unas pocas definiciones operacionales de PA, de Futschek (2006), Grozdev y Terzieva (2015) y Sadykova y Usolzev (2018), sin embargo son muy extensas para presentarlas aquí.

El pensamiento algorítmico en el pensamiento computacional

Aunque se ha hablado del PC desde la década de 1980, fue el ensayo de Wing (2006) el que inició las discusiones científicas recientes sobre este tema. En ese ensayo Wing (2006) describe que el PC implica la resolución de problemas, el diseño de sistemas y la comprensión del comportamiento humano, y abarca la amplitud del campo de las ciencias de la computación; además indica que abarca una variedad de herramientas mentales como, entre otras, las siguientes:

- Transformar un problema aparentemente difícil en uno solucionable, posiblemente mediante simplificación, transformación o simulación.
- Recurrir al pensamiento recursivo.
- Emplear la abstracción y la descomposición al enfrentarse a tareas grandes y complejas.
- Seleccionar una representación adecuada para un problema o modelizar los aspectos relevantes del problema para hacerlo más manejable.
- Utilizar el razonamiento heurístico para descubrir posibles soluciones.
- Gestionar tareas de manera concurrente.
- Interpretar el código como datos y los datos como código.
- Considerar estrategias para prevenir, protegerse y recuperarse de los peores escenarios mediante la redundancia, la mitigación de daños y la corrección de errores.
- Reconocer tanto los beneficios como los riesgos asociados con el uso de un alias o la asignación de múltiples nombres a un individuo o entidad.

- Negociar y equilibrar consideraciones relacionadas con el tiempo y el espacio, así como con la capacidad de procesamiento y la capacidad de almacenamiento.

Los primeros cinco elementos son análogos a algunos de los componentes operacionales del PA identificados por Futschek (2006), Grozdev y Terzieva (2015) y Sadykova y Usolzev (2018).

Más recientemente, la International Society for Technology in Education y la Computer Science Teachers Association (ISTE y CSTA, 2011) han elaborado de forma consensuada una definición operacional del PC:

Es un proceso de resolución de problemas que incluye (pero no se limita a) las siguientes características: (a) Formular problemas de una manera que permita usar una computadora y otras herramientas para ayudar a resolverlos; (b) Organizar y analizar datos de forma lógica; (c) Representar datos a través de abstracciones como modelos y simulaciones; (d) Automatizar soluciones a través del pensamiento algorítmico; (e) Identificar, analizar e implementar posibles soluciones con el objetivo de lograr la combinación más eficiente y efectiva de pasos y recursos; (f) Generalizar y transferir este proceso de resolución de problemas a una amplia variedad de problemas [p. 1].

De esta definición operacional del PC se destaca claramente que el PA es uno de sus componentes fundamentales. Esta idea también se ve respaldada por la revisión sistemática de literatura realizada por Selby y Woollard (2013), quienes consideran al PC como una actividad enfocada en el producto, relacionada con –aunque no limitada a– la resolución de problemas. Se trata de un proceso cognitivo o de pensamiento que involucra:

- la abstracción,
- la descomposición de problemas en subproblemas,
- el pensamiento algorítmico,
- la evaluación de algoritmos y sistemas computacionales, y
- reutilización de algoritmos en contextos diferentes a los originales.

Grover y Pea (2013) también realizaron una revisión de la definición de PC y afirman que el PC está compuesto por los siguientes elementos:

- Flujo de ejecución.
- Descomposición de problemas (o modularización).
- Lógica condicional.
- Depuración y detección sistemática de errores.
- Abstracción y generalización.
- Procesamiento sistemático de información.
- Sistemas de símbolos.
- Iteración, recursión y paralelización.
- Consideraciones sobre desempeño y eficiencia.

Similar a la lista de Wing (2006), los primeros cuatro elementos de esta lista son análogos a algunos de los componentes identificados del PA por Futschek (2006), Grozdev y Terzieva (2015) y Sadykova y Usolzev (2018).

Weintrop et al. (2016) presentan una taxonomía de prácticas y habilidades que componen el PC pensando en su incorporación en las aulas de clase de ciencias y matemática de educación básica y secundaria:

1. Sobre los datos: recolección, creación, manipulación, análisis, visualización.
2. Modelización y simulación (es decir, crear modelos computacionales para entender un concepto, crear modelos computacionales para encontrar y verificar soluciones, evaluar diferentes modelos computacionales).
3. Resolución de problemas computacionales (reformular problemas para obtener soluciones computacionales, programación de computadoras, selección efectiva de herramientas computacionales, evaluar y valorar diferentes aproximaciones o soluciones a un problema, desarrollar soluciones computacionales modulares, crear abstracciones computacionales, solución de problemas y depuración).
4. Pensamiento en sistemas (investigar un sistema complejo como un todo, entender las relaciones dentro de un sistema, pensamiento en niveles, comunicar información sobre un sistema, definir sistemas y administrar la complejidad)

Aunque esta taxonomía no incluye explícitamente el PA, sí hace énfasis en la habilidad de resolución de problemas por medio de soluciones y abstracciones, tal como indican Futschek (2006) y Grozdev y Terzieva (2015), y también hace énfasis en la comprensión de sistemas complejos, como hacen Sadykova y Usolzev (2018). Por ello se considera que implícitamente Weintrop et al. (2016) incorporan el PA en su taxonomía de habilidades de PC.

En la revisión sistemática de literatura de Kalelioğlu et al. (2016) se identificó que las características más comunes que forman el PC son: abstracción, resolución de problemas, pensamiento algorítmico, reconocimiento de patrones, y pensamiento basado en diseño.

Para Korkmaz et al. (2017) y Doleck et al. (2017), las habilidades del PC son: pensamiento algorítmico, cooperatividad, creatividad, pensamiento crítico, resolución de problemas.

En la investigación de Shute et al. (2017) el PC se compone de los siguientes elementos: descomposición, abstracción, diseño de algoritmos (algoritmización), depuración, iteración, generalización.

Para Stephens y Kadjevich (2020), las piedras angulares del PA son la descomposición, la abstracción y la algoritmización; mientras que las del PC son la descomposición, la abstracción, la algoritmización, y la automatización.

Tal como sucede con otros constructos psicológicos, no hay una única definición operacional universalmente aceptada y las que han sido propuestas siguen en evolución, como muestran las revisiones sistemáticas de literatura de Selby y Woollard (2013), Voogt et al. (2015), Czerkowski y Lyman (2015), Shute et al. (2017), Freiman et

al. (2018) y Tang et al. (2020), y los meta-análisis de Grover y Pea (2013) y Kalelioğlu et al. (2016) sobre la literatura científica que trata sobre el PC en los años recientes.

Respecto a este aspecto, Denning (2017) señala que tanto la definición como los métodos de evaluación y las afirmaciones sobre su beneficio universal son áreas problemáticas en la investigación del PC. Sin embargo, para efectos de este ensayo, el PA es claramente una parte importante del PC. Esto está sustentado tanto en la definición operacional de ISTE y CSTA (2011) como en las revisiones de literatura que realizan Selby y Woollard (2013), Grover y Pea (2013), Kalelioğlu et al. (2016) y Shute et al. (2017).

Pensamiento computacional y pensamiento matemático

En esta sección se presenta la relación entre el PC y el desarrollo del PM para finalizar el ciclo de relaciones entre PM, PA y PC.

Ya a mediados de los años 70s Dijkstra (1974) reflexionaba que la programación de computadoras es una disciplina de naturaleza matemática, y “como una ciencia de objetos abstractos, la matemática se basa en la lógica en lugar de la observación como su medida de verdad, [pero] también emplea la observación, la simulación, e incluso la experimentación como medio para descubrir la verdad” (NRC, 1989, p. 31). De ahí que el PA y el PC sean recursos útiles para el desarrollo de diferentes tipos de PM.

Bubnó y Takács (2014) describen muchas analogías entre la resolución de problemas matemáticos de texto y de problemas de programación de computadoras. En ambos casos hay que entender el problema, usar variables, modelizar, planificar, ejecutar el plan, realizar pruebas, resolver el problema inicial, etc. También identifican dos diferencias importantes: la notación y las reglas sintácticas.

El estudio de Kilhamn y Bråting (2019) examina cómo las actividades de programación de computadoras pueden influir de diversas maneras en el desarrollo del pensamiento algebraico en los estudiantes. Entre otras interacciones, revela que el aprendizaje del álgebra puede enfrentar obstáculos cuando se realiza después de actividades introductorias a la programación, principalmente debido a las diferencias en la simbología utilizada. Esto no implica que sea necesariamente desaconsejable aprender a programar antes que aprender álgebra, ya que Modeste (2012b, 2016) señala que lo contrario también puede ser cierto. Simplemente tienen diferentes notaciones y reglas sintácticas, como describen Bubnó y Takács (2014), por lo que es natural que puedan surgir interferencias entre el aprendizaje del álgebra y la programación de computadoras.

Weintrop et al. (2016) enfrentan el desafío de definir el PC en el ámbito de la educación en ciencias naturales y matemática, y proporcionan una base teórica sobre cómo debería manifestarse este tipo de pensamiento en las aulas. Abogan por la idea de integrar el PC en lugar de simplemente añadirlo a los contextos del aula

de matemática y ciencias, argumentando que esto es necesario debido a la creciente naturaleza computacional de las matemáticas y las ciencias naturales modernas. Broley et al. (2017) muestran que se valida el modelo de Weintrop et al. (2016) con un diverso conjunto de prácticas profesionales auténticas en el contexto de la matemática, y aseguran que el poder de la computadora ha tenido un gran impacto en la forma en la que los profesionales en CTIM –ciencia, tecnología, ingeniería y matemática– pueden hacer y hacen su trabajo.

El análisis cualitativo de Broley et al. (2018) revela que los profesores e investigadores matemáticos logran una variedad de tareas con la programación de computadoras, como modelar fenómenos complejos para resolver problemas del mundo real; realizar cálculos y simulaciones imposibles de hacer a mano para realizar matemática experimental; construir demostraciones; desarrollar herramientas tecnológicas, etc.

La forma en que el profesorado universitario utiliza la programación de computadoras para la enseñanza de la matemática es muy diversa, pero la investigación de Broley et al. (2018) identificó seis tipos o niveles de interacción que el profesorado ofrece a sus estudiantes:

1. Solo observación. El docente ejecuta.
2. Ejecución (ex-aula) de un programa existente.
3. Lectura (y tal vez análisis) del código fuente de un programa.
4. Modificación de un programa existente (es decir, editar su código fuente) para lograr algo distinto.
5. Construcción de un programa, con base en especificaciones del docente (incluso dado el algoritmo).
6. El estudiante desarrolla un programa (o varios), y esto puede incluir la validación o depuración.

Shute et al. (2017) examinan el campo del PC en la educación y comparan el PC con el PM describiendo habilidades de cada uno, comunes y no comunes:

- Las habilidades comunes entre el PC y el PM: resolución de problemas, modelización, análisis e interpretación de datos, y estadística y probabilidad.
- Las habilidades del PC que no pertenecen al PM: simulación, minería de datos, redes, recolección automatizada de datos, juegos, razonamiento algorítmico, robótica y programación.
- Las habilidades del PM que no pertenecen al PC: conteo, aritmética, álgebra, geometría, cálculo, teoría de conjuntos y topología.

Freiman et al. (2018) discutieron la pregunta sobre cómo se podría conceptualizar el PC en la investigación sobre la enseñanza de la matemática. Este grupo llegó a la conclusión de que hay una multiplicidad de perspectivas sobre lo que es el PC, y que estas perspectivas pueden ser útiles dependiendo de los objetivos de los usuarios, como los propósitos específicos de un equipo de investigación. También señalaron

que estas diferentes perspectivas pueden corresponderse con diferentes tipos de tareas. Ellos afirman que incluso entre un grupo pequeño de investigadores especializados en el tema, cada uno define el PC de manera ligeramente diferente: para algunos se trata de las prácticas de una persona (Broley et al., 2018), para otros es un artefacto o herramienta (Buteau et al., 2016), mientras que para otros es una de las muchas modalidades de aprendizaje (Shute et al., 2017).

Siguiendo el razonamiento de Freiman et al. (2018), se puede afirmar que los contextos matemáticos pueden ser utilizados como medios para integrar el PC en la enseñanza y el aprendizaje, lo que proporciona oportunidades fructíferas para llevar a cabo investigaciones significativas. Esto puede conducir a una mayor comprensión de la matemática, así como a una comprensión más profunda del papel del PC y las posibilidades que ofrecen las herramientas computacionales en la enseñanza y aprendizaje de la matemática.

Aclaraciones sobre posibles confusiones

Existe una marcada tendencia por asociar el PC con programación de computadoras (Wing, 2006; Voogt et al., 2015; Tang et al., 2020). Es cierto que ambas “constituyen competencias clave que deben ser adquiridas por los jóvenes estudiantes, y cada vez más por los trabajadores, en una amplia gama de actividades industriales y profesionales” (Bordignon y García-Marín, 2018, p. 180), sin embargo, el PC comprende otros componentes además de la programación de computadoras, como indican Wing (2006), ISTE y CSTA (2011) y Weintrop et al. (2016).

También es cierto que encontrar e inventar algoritmos apropiados para un problema dado es un requisito previo necesario para la programación de computadoras (Futschek y Moschitz, 2010; Juškevičienė, 2020), y que la programación de computadoras es una herramienta importante para ayudar a desarrollar habilidades de PC (Voogt et al., 2015). Por otro lado, “la programación [de computadoras] es un tipo específico de actividad humana y su realización exitosa requiere no solamente la aplicación práctica de conocimientos y habilidades, sino también un tipo específico de pensamiento [el PAJ]” (Grozdev y Terzieva, 2015, p. 471).

Sadykova y Usolzev (2018) apoyan esta idea, ya que afirman que “el pensamiento algorítmico bien desarrollado es muy importante para el desarrollo de programas informáticos. Si un estudiante no tiene las habilidades del pensamiento algorítmico, incluso el conocimiento de uno o diferentes lenguajes de programación no será de utilidad” (p. 2).

También es importante destacar que el pensamiento computacional (PC) no se refiere a pensar como una computadora, dado que las computadoras no piensan; en realidad, el término se refiere a pensar de manera similar a como lo hace alguien que trabaja en ciencias de la computación (Grover y Pea, 2018).

Algunas influyentes organizaciones relacionadas a la enseñanza de las ciencias de la computación aseguran que el PC puede tener lugar sin una computadora, y también que el uso de una computadora en clase no necesariamente construye o promueve PC (ACM et al., 2016). Esta idea es compartida por Wing (2017), que afirma que las personas pueden desarrollar PC sin una máquina. Voogt et al. (2015) van más allá al afirmar que el PC no necesariamente requiere del uso de la programación de computadoras, al igual que se afirma sobre el PA (Futschek, 2006; Ziatdinov y Musa, 2012; Douadi et al., 2012; Kátai, 2015).

Diferencias entre pensamiento algorítmico y pensamiento computacional

Se debe tener cuidado de no confundir el PC con el PA a pesar de que sus definiciones en la literatura sean similares. En la actualidad se considera que el PA es el principal componente operacional del PC (Selby y Woollard, 2013; Grover y Pea, 2013; Kalelioğlu et al., 2016, Stephens y Kadijevich, 2020), a pesar de que muchas definiciones operacionales de PC no lo incluyan explícitamente.

A partir de las definiciones ya descritas, podemos concluir que el PA se enfoca en la creación, análisis, manipulación, ejecución y evaluación de algoritmos. Sin embargo, de las descripciones y definiciones de PC se puede apreciar que este va más allá de dichas actividades, por ejemplo, este incluye la habilidad de “interpretar el código como datos y los datos como código”, la habilidad de “pensar en términos de prevención, protección y recuperación [...] a través de la redundancia, la contención de daños y la corrección de errores” (Wing, 2006, pp. 33-34); también la habilidad de “representar datos a través de abstracciones como modelos y simulaciones” (ISTE y CSTA, 2011, p. 1), o la “selección efectiva de herramientas computacionales” (Weintrop et al., 2016, p. 139).

Los estándares educativos para la enseñanza de las ciencias de la computación que plantea la CSTA (2017), además de algoritmos, programación, abstracción y modelización, incluye muchos aspectos como la ciber-seguridad, los impactos de la computación en la cultura, los impactos de la computación en las leyes y la ética; lo cual es coincidente con la noción de *technoracy* que plantea D’Ambrosio (1999).

Por lo tanto, de manera concisa, se puede afirmar que el PC, a diferencia del PA, no se limita únicamente a los algoritmos, sino que aborda las características y limitaciones de la tecnología subyacente a estos algoritmos (paradigmas de programación, lenguajes, concurrencia, redes, capacidad de almacenamiento, etc.), así como las implicaciones sociales del uso de esa tecnología (robo de identidad, respeto y responsabilidad en el ciber-espacio, cambios culturales debido a la intrusión de las tecnologías computacionales, licencias digitales, etc.).

CONCLUSIONES

Con base en la gran cantidad de relaciones mutuas entre los conceptos estudiados, es clara la necesidad de realizar más investigación sobre las interacciones del PA y el PC en los diferentes tipos de PM. Esto es válido en todos los ámbitos, epistemológicos, praxeológicos, psicológicos, cognitivos, fenomenológicos, etc.

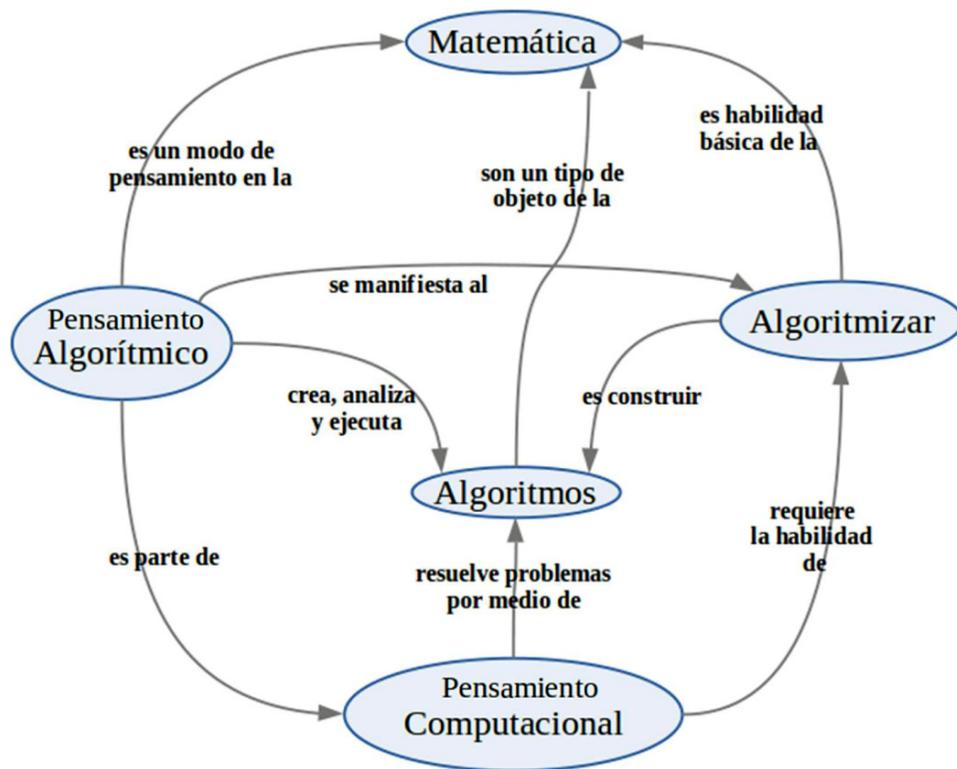
Puesto que las definiciones conceptuales y operacionales de los constructos PA y PC no son universalmente aceptadas, es importante seguir disertando sobre ellas. Sin embargo, tal como sugieren Voogt et al. (2015), tal vez no deberíamos buscar unas definiciones definitivas sino más bien encontrar similitudes y relaciones.

A pesar de que en la literatura científica aún no se ha delineado claramente cómo incorporar y menos aún cómo integrar el PA y el PC en la enseñanza de la matemática, se reconoce que los métodos de enseñanza y aprendizaje de la matemática han experimentado una profunda expansión en la actualidad gracias a las oportunidades que ofrecen las herramientas computacionales; por lo tanto, es necesario promover tanto el PC como el PA en toda la población para poder aprovechar esta expansión.

A manera de resumen esquemático se presenta en la Figura 1 un mapa de las relaciones entre los principales conceptos de este ensayo.

Figura 1

Resumen de las relaciones entre los principales conceptos presentados en este ensayo



Fuente: Construcción personal.

REFERENCIAS

- ACM [Association for Computing Machinery], Code.org, Computer Science Teachers Association, Cyber Innovation Center, y National Math and Science Initiative (2016). *K-12 Computer Science Framework*. <https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf>
- Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835. <https://doi.org/10.1093/comjnl/bxs074>
- Arévalo-Mercado, C. A., Estrada-Rentería, B. G., y Muñoz-Andrade, E. L. (2019). El efecto de la teoría de carga cognitiva en el aprendizaje de la programación básica. *Entorno*, 67, 169-176. <https://doi.org/10.5377/entorno.v0i67.7500>
- Barr, D., Harrison, J., y Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6), 20-23. <https://eric.ed.gov/?id=EJ918910>
- Batanero Bernabeu, M. d. C. (2013). Sentido estadístico: componentes y desarrollo. *Probabilidad Condicionada: Revista de Didáctica de la Estadística*, 1, 55-64. <https://document.unirioja.es/servlet/articulo?codigo=4770161>
- Blanco-Hamad, A., Salgado-Castillo, A., y Alonso-Berenguer, I. (2016). Habilidades para la algoritmización computacional en la Licenciatura en Educación: especialidad Educación Laboral-Informática. *Maestro y Sociedad. Revista Electrónica para Maestros y Profesores*, 13(1), 18-30. <https://maestrosociedad.uo.edu.cu/index.php/MyS/article/view/986>
- Bordignon, F., y García-Marín, D. (2018). Principios del pensamiento computacional. En R. Aparici y D. García-Marín (eds.), *Comunicar y educar en el mundo que viene* (2a. ed., pp. 177-191). Gedisa.
- Broley, L., Buteau, C., y Muller, E. (2017). (Legitimate peripheral) computational thinking in mathematics. *Tenth Congress of the European Society for Research in Mathematics Education (CERME 10)*. <https://hal.archives-ouvertes.fr/hal-01946353>
- Broley, L., Caron, F., y Saint-Aubin, Y. (2018). Levels of programming in mathematical research and university mathematics education. *International Journal of Research in Undergraduate Mathematics Education*, 4(1), 38-55. <https://doi.org/10.1007/s40753-017-0066-1>
- Bubnó, K. T., y Takács, V. L. (2014). Solving word problems by computer programming. En A. Ambrus y É. Vásárhelyi (eds.), *15th ProMath Problem Solving in Mathematics Education Conference* (pp. 193-208). Mathematics Teaching and Education Center. <https://helda.helsinki.fi/bitstream/handle/10138/230168/ProMath2013.pdf>
- Buteau, C., Jarvis, D. H., y Lavicza, Z. (2014). On the integration of Computer Algebra Systems (CAS) by Canadian mathematicians: Results of a national survey. *Canadian Journal of Science, Mathematics and Technology Education*, 14(1), 35-57. <https://doi.org/10.1080/14926156.2014.874614>
- Buteau, C., Muller, E., Marshall, N., Sacristán, A. I., y Mgombelo, J. (2016). Undergraduate mathematics students appropriating programming as a tool for modelling, simulation, and visualization: A case study. *Digital Experiences in Mathematics Education*, 2(2), 142-166. <https://doi.org/10.1007/s40751-016-0017-5>
- CSTA [Computer Science Teachers Association] (2017). *CS standards*. <https://www.csteachers.org/page/standards>
- Congacha Aushay, E. P., Santillán Castillo, J. R., Guerra Salazar, J. E., y Barba Vera, R. G. (2018). Empleo de una aplicación informática como estrategia didáctica para el desarrollo de aptitudes académicas. *Revista Educación*, 42(2), 398-413. <https://doi.org/10.15517/revedu.v42i2.27472>
- Coscarelli, W. C., Visscher, M. O., y Schwen, T. M. (1976). Algorithmization and transfer of learning. *Annual Meeting of the American Educational Research Association*, 19-23. <http://files.eric.ed.gov/fulltext/ED121608.pdf>
- Czerkawski, B. C., y Lyman, E. W. (2015). Exploring issues about computational thinking in higher education. *TechTrends*, 59(2), 57-65. <https://doi.org/10.1007/s11528-015-0840-3>
- D'Ambrosio, U. (1999). Literacy, matheracy, and technoracy: A trivium for today. *Mathematical Thinking and Learning*, 1(2), 131-153. https://doi.org/10.1207/s15327833mtl0102_3
- Davis, G. E., y McGowen, M. A. (2001). Jennifer's journey: Seeing and remembering mathematical connections in a pre-service elementary teachers course.

- En M. van den Heuvel-Panhuizen (ed.), *Proceedings of the 25th Conference of the International Group for the Psychology of Mathematics Education (PME 25)* (vol. 2, pp. 305-312). International Group for the Psychology of Mathematics Education. <https://files.eric.ed.gov/fulltext/ED466960.pdf>
- Delgado Rubí, J. R. (1998). Las habilidades generales matemáticas. En H. Hernández (ed.), *Cuestiones de didáctica de la matemática*. Homo Sapiens.
- Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39. <https://doi.org/10.1145/2998438>
- Dijkstra, E. W. (1974). Programming as a discipline of mathematical nature. *The American Mathematical Monthly*, 81(6), 608-612. <https://doi.org/10.1080/0029890.1974.11993624>
- Doğan, A. (2020). Algorithmic thinking in primary education. *International Journal of Progressive Education*, 16(4), 286-301. <https://doi.org/10.29329/ijpe.2020.268.18>
- Doleck, T., Bazalais, P., Lemay, D. J., Saxena, A., y Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: Exploring the relationship between computational thinking skills and academic performance. *Journal of Computers in Education*, 4(4), 355-369. <https://doi.org/10.1007/s40692-017-0090-9>
- Douadi, B., Tahar, B., y Hamid, S. (2012). Smart edutainment game for algorithmic thinking. *Procedia - Social and Behavioral Sciences*, 31, 454-458. <https://doi.org/10.1016/j.sbspro.2011.12.085>
- Fernández Millán, E. (2013). Razonamiento combinatorio y el currículo español. *Probabilidad Condicionada: Revista de Didáctica de la Estadística*, 1, 539-545. <https://dialnet.unirioja.es/servlet/articulo?codigo=5487251>
- Freiman, V., Broley, L., Buteau, C., y Vasilyeva, N. (2018). Report from working group focussed on research-based understandings of the interplay between the affordances of computational thinking and mathematics. En *Online Proceedings of the Computational Thinking in Mathematics Education Symposium*. University of Ontario Institute of Technology. http://ctmath.ca/wp-content/uploads/2018/10/Symposium_Working-Report_-Frieman-Et-al.pdf
- Futschek, G. (2006). Algorithmic thinking: The key for understanding computer science. En R. T. Mittermeir (ed.), *Informatics education – The bridge between using and understanding computers. ISSEP 2006. Lecture Notes in Computer Science* (vol. 4226, pp. 159-168). Springer-Verlag. https://doi.org/10.1007/11915355_15
- Futschek, G., y Moschitz, J. (2010). Developing algorithmic thinking by inventing and playing algorithms. En *Proceedings of the 2010 Constructionist Approaches to Creative Learning, Thinking and Education: Lessons for the 21st Century (Constructionism 2010)* (pp. 1-10). http://publik.tuwien.ac.at/files/PubDat_187461.pdf
- Gallardo Romero, J., y González, J. L. (2007). Una aproximación operativa al diagnóstico y la evaluación de la comprensión del conocimiento matemático. *PNA. Revista de Investigación en Didáctica de la Matemática*, 1(1). <http://revistaseug.ugr.es/index.php/pna/article/view/6217>
- Gayo Avello, D., Cernuda del Río, A., Cueva Lovelle, J. M., Díaz Fondón, M. Á., García Fuente, M. del P. A., y Redondo López, J. M. (2003). Reflexiones y experiencias sobre la enseñanza de POO como único paradigma. En *IX Jornadas sobre la Enseñanza Universitaria de la Informática (JENUI'2003)*. <http://hdl.handle.net/10651/30804>
- Grover, S., y Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>
- Grover, S., y Pea, R. (2018). Computational thinking: A competency whose time has come. En S. Sentance, E. Barendsen y C. Schulte (eds.), *Computer science education: Perspectives on teaching and learning in school*. Bloomsbury Publishing.
- Grozdev, S., y Terzieva, T. (2011). Research of the concept of algorithmic thinking in teaching computer science. En *The International Scientific-Practical Conference "Informatization of Education-2011"* (pp. 112-119).
- Grozdev, S., y Terzieva, T. (2015). A didactic model for developmental training in computer science. *Journal of Modern Education Review*, 5(5), 470-480. [https://doi.org/10.15341/jmer\(2155-7993\)/05.05.2015/005](https://doi.org/10.15341/jmer(2155-7993)/05.05.2015/005)
- Guzdial, M., y Soloway, E. (2002). Teaching the Nintendo generation to program. *Communications of the ACM*, 45(4), 17-21. <https://doi.org/10.1145/505248.505261>
- Hidayat, T., Susilaningsih, E., y Kurniawan, C. (2018). The effectiveness of enrichment test instruments

- design to measure students' creative thinking skills and problem solving. *Thinking Skills and Creativity*, 29, 161-169. <https://doi.org/10.1016/j.tsc.2018.02.011>
- Hoyles, C., y Noss, R. (2015). Revisiting programming to enhance mathematics learning. En *Math+Coding Symposium*.
- Hurlburt, G. F. (2018). Thinking critically about algorithmic thinking. *IT Professional*, 20(2), 5-10. <https://doi.org/10.1109/MITP.2018.021921644>
- Israel, M., y Lash, T. (2019). From classroom lessons to exploratory learning progressions: Mathematics + computational thinking. *Interactive Learning Environments*, 28(3), 362-382. <https://doi.org/10.1080/10494820.2019.1674879>
- ISTE, y CSTA [International Society Technology Education, y Computer Science Teachers Association] (2011). *Operational Definition of Computational Thinking for K-12 Education*. https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf
- Juškevičienė, A. (2020). Developing algorithmic thinking through computational making. En D. Gintautas, B. Jolita y K. Janusz (eds.), *Data science: New issues, challenges and applications* (vol. 869, pp. 183-197). Springer. https://doi.org/10.1007/978-3-030-39250-5_10
- Kalelioğlu, F., y Gülbahar, Y. (2014). The effect of instructional techniques on critical thinking and critical thinking dispositions in online discussion. *Educational Technology & Society*, 17(1), 248-258. <https://www.jstor.org/stable/jeductechsoci.17.1.248>
- Kalelioğlu, F. K., Gülbahar, Y., y Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal Modern Computing*, 4(3), 583-596. https://www.bjmc.lu.lv/fileadmin/user_upload/lu_portal/projekti/bjmc/Contents/4_3_15_Kalelioglu.pdf
- Kátai, Z. (2015). The challenge of promoting algorithmic thinking of both sciences-and humanities-oriented learners. *Journal of Computer Assisted Learning*, 31(4), 287-299. <https://doi.org/10.1111/jcal.12070>
- Kilhamn, C., y Bråting, K. (2019). Algebraic thinking in the shadow of programming. En U. T. Jankvist, M. van den Heuvel-Panhuizen y M. Veldhuis (eds.), *Eleventh Congress of the European Society for Research in Mathematics Education (CERME 11)* (vol. TWG03, núm. 7). Freudenthal Group. <https://hal.archives-ouvertes.fr/hal-02429028>
- Kinnunen, P., y Malmi, L. (2006). Why Students drop out CS1 course? En *Proceedings of the Second International Workshop on Computing Education Research* (pp. 97-108). Association for Computing Machinery. <https://doi.org/10.1145/1151588.1151604>
- Knuth, D. E. (1974). Computer science and its relation to mathematics. *The American Mathematical Monthly*, 81(4), 323-343. <https://doi.org/10.1080/00029890.1974.11993556>
- Knuth, D. E. (1985). Algorithmic thinking and mathematical thinking. *The American Mathematical Monthly*, 92(3), 170-181. <https://doi.org/10.1080/00029890.1985.11971572>
- Korkmaz, Ö., Çakir, R., y Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558-569. <https://doi.org/10.1016/j.chb.2017.01.005>
- Kurland, D. M., Clement, C. A., Mawby, R., y Pea, R. D. (1987). Mapping the cognitive demands of learning to program. En D. N. Perkins, J. Lochhead y J. C. Bishop (eds.), *Thinking: The Second International Conference*. Routledge.
- Lockwood, E., DeJarnette, A. F., Asay, A., y Thomas, M. (2016). Algorithmic thinking: An initial characterization of computational thinking in mathematics. En M. B. Wood, E. E. Turner, M. Civil y J. A. Eli (eds.), *Proceedings of the 38th Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education (PMENA 38)* (pp. 1588-1595). The University of Arizona. <https://files.eric.ed.gov/fulltext/ED583797.pdf>
- Lovász, L. (1988). Algorithmic mathematics: An old aspect with a new emphasis. En *Proceedings of 6th International Congress on Mathematics Education* (pp. 67-78). <https://web.cs.elte.hu/~%20lovasz/icme.pdf>
- Martínez-López, L. G., y Gualdrón-Pinto, E. (2018). Fortalecimiento del pensamiento variacional a través de una intervención mediada con TIC en estudiantes de grado noveno. *Revista de Investigación, Desarrollo e Innovación*, 9(1), 91-102. <https://doi.org/10.19053/20278306.v9.n1.2018.8156>
- McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-D., Laxer, C., Thomas, L., Utting, I., y Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of program-

- ming skills of first-year CS students. En *Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education* (pp. 125-180). <https://doi.org/10.1145/572133.572137>
- Mingus, T. T. Y., y Grassl, R. M. (1998). Algorithmic and recursive thinking: Current beliefs and their implications for the future. En L. J. Morrow y M. J. Kenney (eds.), *The teaching and learning of algorithms in school mathematics* (pp. 32-43). National Council of Teacher of Mathematics.
- Misfeldt, M., y Ejsing-Duun, S. (2015). Learning mathematics through programming: An instrumental approach to potentials and pitfalls. En K. Krainer y N. Vondrová (eds.), *Ninth Congress of the European Society for Research in Mathematics Education (CERME 9)* (pp. 2524-2530). Charles University in Prague/ERME. <https://hal.archives-ouvertes.fr/hal-01289367>
- Modeste, S. (2012a). *Enseigner l'algorithme pour quoi? Quelles nouvelles questions pour les mathématiques? Quels apports pour l'apprentissage de la preuve?* [Tesis doctoral]. Universidad de Grenoble. HAL Theses. Archivo tel-00783294. <https://tel.archives-ouvertes.fr/tel-00783294>
- Modeste, S. (2012b). La pensée algorithmique: Apports d'un point de vue extérieur aux mathématiques. En J.-L. Dorier y S. Gousseau Coutat (eds.), *Enseignement des Mathématiques et Contrat Social: Enjeux et Défis pour le 21e Siècle – Actes du Colloque Espace Mathématique Francophone 2012* (pp. 467-479). <https://www.emf.unige.ch/files/6514/5320/2307/EMF2012GT-3MODESTE.pdf>
- Modeste, S. (2016). Impact of informatics on mathematics and its teaching. En F. Gadducci y M. Tavosanis (eds.), *History and philosophy of computing* (pp. 243-255). Springer. https://doi.org/10.1007/978-3-319-47286-7_17
- NRC [National Research Council] (1989). *Everybody counts: A report to the nation on the future of mathematics education*. National Academies Press.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books. <https://dl.acm.org/doi/pdf/10.5555/1095592>
- Pyzara, A. (2012). Algorithmisation in teaching mathematics. *Didactica Mathematicae*, 34, 51-68. <https://doi.org/10.14708/dm.v34i0.417>
- Ramadhan, H. A. (2000). Programming by discovery. *Journal of Computer Assisted Learning*, 16(1), 83-93. <https://doi.org/10.1046/j.1365-2729.2000.00118.x>
- Rasmussen, C., Zandieh, M., King, K., y Teppo, A. (2005). Advancing mathematical activity: A practice-oriented view of advanced mathematical thinking. *Mathematical Thinking and Learning*, 7(1), 51-73. https://doi.org/10.1207/s15327833mtl0701_4
- Ruiz Ledesma, E. F., y Valdemoros Álvarez, M. E. (2004). Connections between qualitative and quantitative thinking about proportion: The case of Paulina. En M. J. Hoines y A. B. Fuglestad (eds.), *The 28th International Conference of the International Group for the Psychology of Mathematics Education*. International Group for the Psychology of Mathematics Education. <https://eric.ed.gov/?id=ED489542>
- Sadykova, O. V., e Il'bahtin, G. G. (2019). The definition of algorithmic thinking. En *Proceedings of the International Session on Factors of Regional Extensive Development (FRED 2019)* (pp. 419-422). Atlantis Press. <https://doi.org/10.2991/fred-19.2020.85>
- Sadykova, O. V., y Usolzev, A. (2018). On the concept of algorithmic thinking. *SHS Web Conferences – International Conference on Advanced Studies in Social Sciences and Humanities in the Post-Soviet Era (ICPSE 2018)*, 55, 03016. <https://doi.org/10.1051/shsconf/20185503016>
- Schoenfeld, A. H. (2013). Reflections on problem solving theory and practice. *The Mathematics Enthusiast*, 10(1), 9-34. <https://doi.org/10.54870/1551-3440.1258>
- Selby, C. C., y Woollard, J. (2013). Computational thinking: The developing definition. En *18th Annual Conference on Innovation and Technology in Computer Science Education*. <https://eprints.soton.ac.uk/356481/>
- Sfard, A. (1991). On the dual nature of mathematical conceptions: Reflections on processes and objects as different sides of the same coin. *Educational Studies in Mathematics*, 22(1), 1-36. <https://doi.org/10.1007/BF00302715>
- Shute, V. J., Sun, C., y Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Stephens, M., y Kadijevich, D. M. (2020). Computational/algorithmic thinking. En S. Lerman (ed.), *Encyclopedia of Mathematics Education* (pp. 117-123). Springer. https://doi.org/10.1007/978-3-030-15789-0_100044

- Tall, D. (ed.). (2002). *Advanced mathematical thinking*. Kluwer Academic Publishers.
- Tang, X., Yin, Y., Lin, Q., Hadad, R., y Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 1-22. <https://doi.org/10.1016/j.compedu.2019.103798>
- Thom, J. S., y McGarvey, L. M. (2015). The act and artifact of drawing(s): Observing geometric thinking with, in, and through children's drawings. *ZDM*, 47(3), 465-481. <https://doi.org/10.1007/s11858-015-0697-0>
- Voogt, J., Fisser, P., Good, J., Mishra, P., y Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715-728. <https://doi.org/10.1007/s10639-015-9412-6>
- Waterman, K., Goldsmith, L., Pasquale, M., Goldenberg, E. P., Malyn-Smith, J., DeMallie, A., y Lee, I. A. (2018). Integrating computational thinking into elementary mathematics and science curriculum materials and instruction. En Pixel (ed.), *Conference Proceedings: The Future of Education*. <https://conference.pixel-online.net/files/foe/ed0008/FP/4648-ICL3076-FP-FOE8.pdf>
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., y Wilensky, U. (2016). Defining Computational thinking for Mathematics and Science classrooms. *Journal of Science Education and Technology*, 25(1), 127-147. <https://doi.org/10.1007/s10956-015-9581-5>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, 366(1881), 3717-3725. <https://doi.org/10.1098/rsta.2008.0118>
- Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7-14. <https://doi.org/10.17471/2499-4324/922>
- Ziatdinov, R., y Musa, S. (2012). Rapid mental computation system as a tool for algorithmic thinking of elementary school students development. *European Researcher*, 25(7), 1105-1110. <http://www.erjournal.ru/pdf.html?n=1342467174.pdf>

Cómo citar este artículo:

Navas López, E. A. (2024). Relaciones entre la matemática, el pensamiento algorítmico y el pensamiento computacional. *IE Revista de Investigación Educativa de la REDIECH*, 15, e1929. https://doi.org/10.33010/ie_rie_rediech.v15i0.1929



Todos los contenidos de *IE Revista de Investigación Educativa de la REDIECH* se publican bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional, y pueden ser usados gratuitamente para fines no comerciales, dando los créditos a los autores y a la revista, como lo establece la licencia.