

Análisis del “pensamiento computacional”: una perspectiva educativa

Analysis of “computational thinking”: An educational approach

J. Ángel Velázquez Iturbide
Universidad Rey Juan Carlos, Madrid, España
angel.velazquez@urjc.es

Mercedes Martín Lope
Universidad Rey Juan Carlos, Madrid, España
mercedes.martin@urjc.es

Resumen

Desde la aparición del término “pensamiento computacional” hace aproximadamente quince años, se ha consolidado internacionalmente como una tendencia en el mundo educativo preuniversitario. Su imprecisión dificulta la celebración de un debate sobre su papel curricular, algo que se echa especialmente de menos en España. El artículo pretende contribuir a aclarar el significado de “pensamiento computacional” y sus implicaciones educativas. Se ha realizado un análisis cualitativo de contenido, utilizando como materiales primarios las definiciones realizadas por una selección de autores e instituciones destacados. Se han analizado tres aspectos: definición, características y elementos, y ámbito de conocimiento. Se han encontrado dos componentes comunes a las definiciones analizadas: la referencia a una “actividad mental” y la inclusión de habilidades de programación. Sin embargo, difieren en la inclusión de otros elementos, que pueden clasificarse como conocimientos, actitudes o competencias. El artículo debate las consecuencias de una definición basada en una “actividad mental” en lugar de constructos educativos. Se concluye que el debate sobre la educación preuniversitaria de la informática debería basarse en un lenguaje educativo más preciso, así como en los actuales conocimientos de los campos de conocimiento de la pedagogía, la informática y la didáctica de la informática.

Palabras clave: análisis de contenido; informática; Educación Primaria; Educación Secundaria; programación de ordenadores.

Abstract

The term “computational thinking” emerged fifteen years ago and has become a world-wide popular tendency in the pre-university educational scope. Its looseness makes difficult to hold a debate on its curricular role, a lack which is especially severe in Spain. The article aims at contributing to clarify the meaning of “computational thinking” and its educational implications. We conducted a qualitative content analysis, using the definitions provided by a selection of outstanding authors and institutions as primary materials. Three aspects were analyzed: definition, features and elements, and scope of knowledge. We have found two common components in the definitions analyzed: the reference to a “mental activity” and the inclusion of programming skills. However, they differ in the inclusion of other elements, which can be classified as knowledge, attitudes or competences. The article discusses the consequences of a definition based on a “mental activity” rather than educational constructs. We conclude that the debate on pre-university computing education should be based on a more precise terminology, as well as on the current body of knowledge in pedagogy, computing and didactics of computing.

Keywords: contents analysis; computing, Primary Education; Secondary Education; computer programming.

1. Introducción

En la última década ha aumentado el interés por la enseñanza de la informática en las etapas educativas preuniversitarias, tanto por razones puramente educativas como por razones económicas (Vogel, Santo & Ching, 2017). También se está fraguando un consenso en que una buena educación en informática no debería limitarse al desarrollo de la competencia digital, sino que los alumnos también deberían adquirir conocimientos básicos de la informática como ciencia y tecnología (Blikstein, 2018; Gander et al., 2013). Las respuestas nacionales a los requerimientos para que los alumnos adquieran competencias digitales e informáticas son muy variadas (Balanskat & Engelhardt, 2015; Bocconi et al., 2016; Vahrenhold et al., 2017), desde asignaturas obligatorias de informática en los currículos nacionales o regionales a iniciativas de carácter informal o no formal, pasando por su enseñanza transversal.

Desafortunadamente, no existe una estrategia nacional en España que garantice a todos los alumnos una formación universal en informática. Se han emprendido numerosas iniciativas de carácter variado, entre otras la creación de asignaturas de libre configuración autonómica (INTEF, 2018). Estaríamos ante un ejemplo de “política rápida” (Adell, Llopis, Esteve & Valdeolivas, 2019), adoptada sin un debate, un consenso y una planificación previos. En consecuencia, estas asignaturas sólo engloban una parte de la materia propia de la informática, sobre todo programación de ordenadores.

Recientemente ha surgido con éxito un nuevo término relacionado: el “pensamiento computacional” (García Peñalvo, 2018; Zapata, 2015), bajo el cual se han desarrollado iniciativas, como la “Escuela de Pensamiento Computacional” del INTEF (<https://intef.es/tecnologia-educativa/pensamiento-computacional/>). También ha sido objeto de estudio de investigación, sobre todo en el ámbito informático (García et al., 2017; Llorens et al., 2017).

Dado que el “pensamiento computacional” no designa ninguna materia, sería necesario aclarar su significado para poder emprender con rigor un debate sobre la educación informática en España (Adell et al., 2019). El primer uso del término “pensamiento computacional” aparece en los trabajos de Seymour Papert sobre la enseñanza del lenguaje de programación LOGO a niños (Papert, 1980), aunque sin definirlo. Su uso actual se debe a Jeannette Wing (2006), siendo directora adjunta para informática en la agencia estadounidense NSF, que proclamó la existencia de dicho tipo de pensamiento, ligado a la resolución de problemas con ayuda del ordenador. Lo calificó de habilidad fundamental, abogando por su enseñanza a los niños al mismo nivel que leer, escribir y hacer operaciones aritméticas. El artículo daba una definición indirecta de pensamiento computacional (Wing, 2006, p. 33): “El pensamiento computacional implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, basándose en los conceptos fundamentales de la informática”. El término englobaría las “herramientas mentales” propias de la informática, que incluso ayudarían a realizar mejor muchas actividades diarias.

La proclama de Wing ha tenido un gran eco (Roig-Vila & Moreno-Isac, 2020). La ACM Digital Library sola acumula más de 400 publicaciones con el término “pensamiento computacional” en el título o en el resumen (Nardelli, 2019). Muchas de estas publicaciones reconocen la falta de una definición precisa, pero eluden este problema. Se han publicado varios estados del arte sobre el tema (Corradini, Lodi & Nardelli, 2017; Lye & Koh, 2014; Mannila et al., 2014; Tikva & Tambouris, 2021), pero sigue sin haber una definición precisa de pensamiento computacional. En esta

situación, diversos autores han advertido de las contradicciones y las consecuencias negativas que se derivan de algunas afirmaciones frecuentes sobre el pensamiento computacional (Armoni, 2016; Denning, 2009, 2017; Hemmendinger, 2010; Nardelli, 2019).

El objetivo del artículo es aclarar la naturaleza del término “pensamiento computacional”. Obsérvese que nuestro objetivo no es proponer una definición precisa del término, sino solamente caracterizar las propuestas desde un punto de vista educativo, lo cual permitirá juzgar mejor sus implicaciones educativas. Tampoco aborda otras cuestiones relevantes para su implantación, como la didáctica (muy variada, incluso sin ordenadores, véase Zapata, 2019) o la formación del profesorado (García et al., 2016).

2. Material y métodos

La investigación presentada en este artículo pretende responder a la pregunta: ¿en qué consiste el llamado “pensamiento computacional”? mediante un análisis cualitativo de las definiciones realizadas por diversos autores y organizaciones de prestigio.

2.1. Materiales primarios

Los materiales primarios de nuestro estudio son definiciones escritas de pensamiento computacional. La mayor parte de las publicaciones existentes remiten a alguna propuesta ajena. Por tanto, hemos considerado que lo más adecuado es realizar una selección con propósito, consistente en una selección de definiciones realizadas por personas o entidades de prestigio. Se han seleccionado tres autores reconocidos por sus aportaciones a la enseñanza de la informática o al propio pensamiento computacional y tres asociaciones que son referentes por sus recomendaciones o desarrollos curriculares sobre informática o pensamiento computacional.

La selección está formada por:

- Jeannette Wing. Es la autora del artículo seminal sobre pensamiento computacional (Wing, 2006).
- Peter Denning. Es un autor destacado en el mundo académico por sus reflexiones sobre la informática y su enseñanza.
- Mitchel Resnick. Es un destacado representante del construccionismo y coautor del lenguaje de programación Scratch, referente en la enseñanza del pensamiento computacional.
- Computer Science Teacher Association (CSTA). Es una asociación estadounidense de profesores no universitarios de informática. Ha codesarrollado el K-12 Computer Science Framework (ACM et al., 2016) y los K-12 Computer Science Standards.
- International Society for Technology in Education (ISTE). Es una asociación estadounidense para el fomento del uso de la informática en la educación. Destaca su definición de “estándares” para el adecuado desempeño de la labor de distintos agentes educativos.
- Computing at School. Es una asociación de profesionales preocupados por fomentar e investigar el aprendizaje de la informática en Gran Bretaña. Sus actividades principales se han dirigido a la formación del profesorado y al desarrollo curricular.

2.2 Procedimiento

Tras una primera lectura de las fuentes seleccionadas, observamos que no siempre definen explícitamente el pensamiento computacional. Por ello, el análisis se dividió en tres fases:

1. Análisis de la propia definición de pensamiento computacional. Supongamos que la estructura sintáctica de una definición es: “el pensamiento computacional es un A tal que B”, es decir, consta de un atributo A y algunos complementos B. En este primer paso, se analizó el atributo A, solamente en aquellas definiciones que son explícitas.
2. Análisis de las características o los componentes de los que consta la definición. En este segundo aspecto, se analizaron todas las definiciones. En las publicaciones que contienen una definición explícita, se analizaron sus complementos (B en la estructura sintáctica presentada en el párrafo anterior). También se ha analizado la enumeración de componentes que completan varias definiciones.
3. Análisis del ámbito de conocimiento asociado a las dos fases anteriores.

2.3 Análisis

Las definiciones se han analizado cualitativamente mediante técnicas de análisis de contenido. Como se ha comentado anteriormente, el análisis se ha estructurado en tres fases, lo que ha obligado a combinar varias técnicas (Hsieh & Shannon, 2005). Se ha realizado un análisis de contenido latente al buscar el término “pensamiento computacional” para determinar el contexto de su uso y su significado subyacente. Asimismo, se ha realizado un análisis de contenido dirigido cuando se ha necesitado clasificar los componentes presentes en las definiciones, recurriendo a categorías existentes.

El análisis dirigido ha utilizado dos clases de categorías:

- Elementos curriculares. Por ser ampliamente conocidos, hemos tomado los elementos curriculares incluidos en la legislación española (BOE, 2014): currículo, objetivos, competencias, contenidos (conocimientos, habilidades, destrezas y actitudes), criterios de evaluación, estándares de aprendizaje evaluables y metodología didáctica.
- Áreas de la informática. Dada su amplia aceptación en el mundo académico informático, se han adoptado las áreas de conocimiento identificadas en los Computing Curricula (ACM & IEEE Computer Society, 2013). El artículo presenta un etiquetado simplificado de estas áreas de conocimiento, por ejemplo, “algoritmos” como simplificación de “algoritmos y complejidad”.

El análisis se ha realizado en tres iteraciones:

1. Análisis preliminar de una selección de definiciones. Fue realizado por uno de los autores.
2. Análisis revisado de las definiciones tras una ligera ampliación de los materiales seleccionados. Fue realizado por el mismo autor.
3. Revisión por el otro autor de la presentación de las definiciones y de los resultados del análisis. Los desacuerdos se resolvieron mediante un diálogo, terminando la revisión al alcanzar un consenso.

2.4 Materiales

En esta sección, presentamos extractos de las definiciones seleccionadas del pensamiento computacional. Para no alargar excesivamente esta sección, las definiciones se presentan de forma muy simplificada, pero suficiente para apreciar su

estilo y contenido. La terminología que utilizan muchos autores es peculiar, en cuyo caso la entrecorramos la primera vez que aparece.

Jeannette Wing

Ya hemos presentado en la introducción la definición indirecta contenida en su artículo seminal (Wing, 2006). En un artículo posterior, Wing daba una definición explícita (Wing, 2010):

“El pensamiento computacional son los procesos de pensamiento implicados al formular problemas y sus soluciones de forma que las soluciones se representen de una forma que pueda ser realizada eficazmente por un agente de procesamiento de información. Informalmente, el pensamiento computacional describe la actividad mental realizada al formular un problema para que admita una solución computacional.”

La autora avisa de que no usa las palabras “problema” y “solución” para referirse solamente a problemas bien definidos, como los problemas algorítmicos, sino para hablar de problemas reales que pueden resolverse completa o parcialmente con ayuda del ordenador, o incluso de problemas cotidianos (por ejemplo, organizar eficientemente la entrega de diplomas en un acto de graduación).

Peter Denning

Peter Denning considera que el pensamiento computacional es una nueva formulación del “pensamiento algorítmico”. Para él “[...] significa una orientación mental a formular problemas como conversiones de unas entradas en salidas y a buscar algoritmos que realicen las conversiones.” (Denning, 2009, p. 28).

Denning ha analizado si el pensamiento computacional proporciona una caracterización adecuada de la informática. Para valorarlo mejor, lo ha analizado dentro del marco Great Principles of Computer Science, que recoge los principios científicos fundamentales de la informática. En este marco, se distinguen “principios básicos” y “prácticas básicas”. Las prácticas básicas son “áreas de habilidad en las que los informáticos pueden mostrar su grado de rendimiento, por ejemplo, principiante, competente y experto”. Identifica cuatro prácticas básicas: la programación, la ingeniería de sistemas, el modelado y la aplicación (en otros campos). Denning interpreta que el pensamiento computacional es bien un “estilo de pensamiento que aparece en todas las prácticas” bien una quinta práctica básica.

Mitchel Resnick

Mitchel Resnick, junto a Karen Brennan, ha dado una definición de pensamiento computacional basada exclusivamente en el lenguaje de programación Scratch. Aunque inicialmente remiten al pensamiento computacional de Wing (2010), después aportan una definición implícita formada por tres “dimensiones” (Brennan & Resnick, 2012):

- “Conceptos computacionales”. Son los conceptos que se utilizan al programar con Scratch: secuencias, bucles, paralelismo, eventos, condicionales, operadores y datos.
- “Prácticas computacionales”. Son las actividades que se realizan al programar en Scratch: ser incremental e iterativo, probar y depurar, reutilizar y mezclar, y abstraer y modularizar.
- “Perspectivas computacionales”. Describen los cambios de perspectiva de los jóvenes sobre el mundo y sobre ellos mismos: expresarse, conectarse y preguntarse.

CSTA e ISTE

Ambas asociaciones organizaron un taller en 2010 sobre pensamiento computacional, del que resultó la siguiente definición (Barr & Stephenson, 2011, p. 51):

“El pensamiento computacional (PC) es una aproximación a la resolución de problemas de forma tal que pueda implementarse con un ordenador. [...] El PC es una metodología de resolución de problemas que se puede automatizar, transferir y aplicar en cualquier materia”.

Esta definición genérica se ha desarrollado u “operacionalizado” (CSTA & ISTE, 2011a, p. 2):

“El pensamiento computacional (PC) es un proceso de resolución de problemas que incluye (pero no está limitado a) las siguientes características:

- Formular problemas de forma que nos permita usar un ordenador y otras herramientas para ayudar a resolverlos.
- Organizar y analizar datos de forma lógica.
- [...]
- Generalizar y transferir este proceso de resolución de problemas a una gran variedad de problemas.

Estas habilidades se basan y se mejoran mediante diversas disposiciones o actitudes que son dimensiones esenciales del PC. Estas disposiciones o actitudes incluyen:

- Confianza en el manejo de la complejidad.
- [...]
- Habilidad para tratar problemas abiertos.
- [...]

Ambas organizaciones han desarrollado otro documento dirigido a profesores (CSTA & ISTE, 2011b), donde acompañan la definición anterior de un vocabulario de conceptos y de un diagrama del progreso esperado a lo largo de cuatro etapas educativas. Por brevedad, nos limitamos a analizar CSTA e ISTE (2011a).

ISTE

Aparte de sus trabajos conjuntos con la CSTA, la ISTE ha elaborado estándares para distintos agentes del mundo educativo, entre ellos, los alumnos. Se identifican siete áreas, de las cuales una es el pensamiento computacional (ISTE, 2016, p. 2):

“Los alumnos desarrollan y emplean estrategias para comprender y resolver problemas en formas que aprovechan la potencia de los métodos tecnológicos para desarrollar y probar soluciones. Los alumnos:

- al explorar y encontrar soluciones, formulan definiciones de problemas adecuadas para métodos basados en la tecnología como el análisis de datos, los modelos abstractos y el pensamiento algorítmico.
- [...]

Computing at School

Computing at School ha elaborado varios informes destacados. Berry et al. (2012) contiene una definición muy detallada de la informática que incluye, entre otros, “conceptos clave” y “procesos clave”. Los conceptos clave son los contenidos que todo alumno debería conocer y los procesos clave, lo que todo alumno debería ser capaz de hacer. El término “pensamiento computacional” se utiliza para indicar que los procesos clave se centran en él (Berry et al., 2012, p. 9):

“El pensamiento computacional es el proceso de reconocer aspectos de la computación en el mundo que nos rodea y de aplicar herramientas y técnicas de la

informática para comprender y razonar sobre sistemas y procesos tanto naturales como artificiales. [...] Es, sin embargo, un término bastante amplio.”

Computing at School también ha publicado una guía dedicada al pensamiento computacional (Csizmadia, 2015). Asume las definiciones de Wing (2006, 2010) y de Berry et al. (2012), afirmando que “el pensamiento computacional es un proceso cognitivo o de pensamiento que conlleva el pensamiento lógico con el que se resuelven problemas y se comprenden mejor artefactos, procedimientos y sistemas” (Csizmadia, 2015, p. 6). Además, enumera cinco habilidades “de pensamiento”: algorítmicamente, mediante descomposición, mediante generalización, mediante abstracción y mediante evaluación.

3. Resultados

En esta sección presentamos los resultados de analizar las definiciones anteriores.

3.1. Definiciones de pensamiento computacional

La Tabla 1 recoge el atributo propuesto en las definiciones explícitas de pensamiento computacional. Escribimos entre corchetes el comienzo del complemento del atributo para comprender mejor el contexto en el que se utiliza dicho atributo.

Tabla 1.

Definiciones del pensamiento computacional

Propuesta	Atributo
Wing (2010)	<ul style="list-style-type: none">– Procesos de pensamiento [implicados al formular problemas y sus soluciones...]– Actividad mental [realizada al formular un problema...]
Denning (2009)	<ul style="list-style-type: none">– Orientación mental [a formular problemas... y a buscar algoritmos...]– Estilo de pensamiento [que aparece en todas las prácticas]
Brennan y Resnick (2012)	<ul style="list-style-type: none">– Remiten a Wing (2010)
CSTA & ISTE (Barr & Stephenson, 2011)	<ul style="list-style-type: none">– Aproximación [a la resolución de problemas...]– Metodología [de resolución de problemas...]
CSTA & ISTE (2011a)	<ul style="list-style-type: none">– Proceso [de resolución de problemas...]
ISTE (2016)	<ul style="list-style-type: none">– Estrategias [para comprender y resolver problemas...]
Computing at School (Berry et al., 2012)	<ul style="list-style-type: none">– Proceso [de reconocer aspectos de la computación... y de aplicar herramientas y técnicas de la informática...]
Computing at School (Csizmadia et al., 2015)	<ul style="list-style-type: none">– Proceso cognitivo o de pensamiento [... con el que se resuelven problemas y se comprenden...]

Fuente: elaboración propia.

Puede verse que las definiciones son bastante heterogéneas. Las definiciones de CSTA e ISTE (Barr & Stephenson, 2011; ISTE, 2016) parecen expresar una habilidad. Sin

embargo, estas mismas fuentes declaran su uso en cualquier materia, lo que sugiere algo más general que una habilidad concreta.

Las definiciones restantes parecen dar a entender la actividad mental que realiza alguien ante una tarea, ya que hablan de “proceso de pensamiento”, “orientación mental” o términos similares. Por tanto, formulan un constructo psicológico, más que educativo. Se sabe que esta actividad mental varía según el grado de maestría que tiene una persona al realizar la tarea, ya que en toda materia existe gran diferencia entre principiantes y expertos (Dreyfus & Dreyfus, 1986; Sweller, Ayres & Kalyuga, 2011). Los principiantes recurren a estrategias generales de resolución de problemas, que son eficaces pero poco eficientes. Según se van ejercitando en la disciplina, desarrollan estrategias de resolución específicas y más eficientes. Estas estrategias se automatizan de forma tal que los expertos las aplican de forma inconsciente. En este caso, la materia en la cual debería desarrollarse el pensamiento computacional está relacionada con los problemas a resolver.

En el tercer paso del análisis volvemos sobre el ámbito del pensamiento computacional.

3.2. Componentes del pensamiento computacional

Esta parte del análisis se centra en la parte restante de cada definición. El resultado aparece resumido en la Tabla 2. La segunda columna contiene un resumen literal de la parte de la definición analizada, sean complementos o componentes enumerativos. En la columna tercera se clasifica dicho resumen en elementos curriculares.

Si examinamos la segunda columna, de nuevo comprobamos que se utiliza una terminología y categorías variadas. Solamente se repiten las palabras “problemas” (5 veces) y “prácticas” (2 veces). Sin embargo, todas las definiciones coinciden en asociar el pensamiento computacional con habilidades. Algunas definiciones aluden a conocimientos (Brennan y Resnick, 2009) o competencias transversales (CSTA e ISTE, 2011a); en ambos casos también incluyen actitudes. Aunque Berry et al. (2012) limita el uso del término “pensamiento computacional” a habilidades, su informe se completa con otros componentes, destacando conocimientos (“conceptos clave”) de informática.

Tabla 2.
 Partes del pensamiento computacional

Propuesta	Complementos/Componentes	Elementos curriculares
Wing (2010)	– Formular problemas y sus soluciones	– Habilidades
Denning (2009)	– Formular problemas y buscar algoritmos – “Práctica básica”	– Habilidades
Brennan & Resnick (2009)	– “Conceptos” – “Prácticas” – “Perspectivas”	– Conocimientos – Habilidades – Actitudes
CSTA & ISTE (Barr & Stephenson, 2011)	– Resolución de problemas	– Habilidades
CSTA & ISTE (2011a)	– “Características” – “Disposiciones” y actitudes	– Habilidades – Actitudes y competencias transversales
ISTE (2016)	– Comprender y resolver problemas	– Habilidades
Computing at School (Berry <i>et al.</i> , 2012)	– “Procesos clave” – Reconocer aspectos de la computación – Aplicar herramientas y técnicas informáticas	– Habilidades
Computing at School (Csizmadia <i>et al.</i> , 2015)	– Resolver problemas y comprender artefactos – Habilidades “para pensar”	– Habilidades

Fuente: elaboración propia

3.3. Ámbito de conocimiento del pensamiento computacional

La Tabla 3 presenta el ámbito de conocimiento del pensamiento computacional según cada definición.

Para delimitar el ámbito de conocimiento, nos hemos centrado en los problemas a abordar y en el ámbito de los componentes presentados en la subsección anterior. Todas las definiciones incluyen la programación o la algoritmia como campo destacado para el desarrollo y aplicación del pensamiento computacional. Las definiciones de Brennan y Resnick (2012) y de Denning (2009) son las más restrictivas. En el primer caso, les interesan actividades de aprendizaje basadas en programación, mientras que Denning remite a los algoritmos.

ISTE (2016) y Computing at School (Berry *et al.*, 2012) presentan mayor ambigüedad. En el caso de ISTE (2016), se debe a la concisión de su definición. Berry *et al.* (2012) ponen el énfasis en la enseñanza de la informática, pero incluyen afirmaciones y ponen ejemplos ajenos al ámbito informático.

Wing (2010) y CSTA e ISTE (2011a, 2001b) son más claros en su transversalidad. Ambos defienden que los conceptos y métodos del pensamiento computacional pueden usarse en otros ámbitos, por ejemplo, en la vida diaria. CSTA e ISTE (2011a) también relacionan el pensamiento computacional con competencias transversales.

Tabla 3.

Ámbito de conocimiento del pensamiento computacional

Propuesta	Ámbito de conocimiento
Wing (2010)	– Informática – Carácter transversal
Denning (2009)	– Algoritmia – Informática
Brennan y Resnick (2012)	– Programación
CSTA & ISTE (2011a)	– Informática: algoritmos, programas, datos – Carácter transversal
ISTE (2016)	– Informática: algoritmos, programas, datos – Carácter transversal
Computing at School (Berry <i>et al.</i> , 2012; Csizmadia <i>et al.</i> , 2015)	– Informática: algoritmos, programas, lenguajes de programación, datos, computadores, redes, cuestiones sociales

Fuente: elaboración propia

4. Debate

En este apartado comentamos los hallazgos anteriores, en algunos casos en relación con los comentarios de otros autores.

4.1. Características formales de las definiciones

Resulta llamativo que todas las definiciones analizadas son deficientes desde un punto de vista formal, independientemente de su contenido. Esta deficiencia se manifiesta de varias formas. En primer lugar, algunas definiciones son circulares: “El pensamiento computacional son los procesos de pensamiento [...]” (Wing, 2010).

En segundo lugar, algunas definiciones son incompletas. Destacan especialmente las siguientes afirmaciones: “El pensamiento computacional [...] incluye (pero no está limitado a) las siguientes características [...]” (CSTA & ISTE, 2011a), “Es, sin embargo, un término bastante amplio” (Berry *et al.*, 2013).

En tercer lugar, cada definición utiliza terminología propia en lugar de terminología consolidada en el área de la educación. Es algo que sorprende especialmente en el caso de las asociaciones de profesores CSTA e ISTE.

La principal consecuencia de estas deficiencias es que las definiciones son largas e imprecisas.

4.2. Carácter del pensamiento computacional

En la primera fase de nuestro análisis del pensamiento computacional, se ha visto que todas las definiciones expresan un constructo psicológico (p.ej. “proceso de pensamiento”). Dicha formulación es problemática porque contrasta con la formulación de la educación en otros campos. Por ejemplo, la educación de la física o la lengua no se plantea en términos de “pensamiento físico” (ni siquiera “pensamiento científico”) o “pensamiento lingüístico”, sino en términos de competencias. Un planteamiento en términos psicológicos es una fuente de desconcierto, confusión o rechazo en otros colectivos, incluyendo los profesores (Corradini, Lodi & Nardelli, 2017; Denning, 2017) o incluso los padres o los especialistas de otras áreas (Denning, 2017). En definitiva, este planteamiento psicológico sólo serviría para describir la evolución previsible de la maestría del alumno, no como base para la planificación educativa.

Denning (2017) llama la atención sobre otro inconveniente de este tipo de planteamientos. La maestría en cualquier campo es un proceso gradual, que se desarrolla a lo largo de años de aprendizaje y práctica (Dreyfus & Dreyfus, 1986). En el caso de la informática, la práctica de la programación y otras materias permiten desarrollar una “maestría informática”. Sin embargo, algunas propuestas de pensamiento computacional se plantean a la inversa: el desarrollo del pensamiento computacional favorecerá el posterior aprendizaje de la programación (Csizmadia, 2015). Este orden inverso del desarrollo de capacidades contradice los postulados de algunas teorías del aprendizaje y carece de evidencia experimental. En efecto, según el constructivismo (Ben-Ari, 2001), el alumno desarrolla sus propios modelos mentales, que inicialmente suelen ser incompletos, imprecisos e inviables (Norman, 1983). Según avanza un alumno en su aprendizaje, sus modelos mentales evolucionan hasta modelos viables y coherentes con los modelos conceptuales presentados por los profesores. Es decir, la evidencia científica nos presenta un orden inverso al invocado por algunos defensores del pensamiento computacional: es el aprendizaje de la programación y de otras áreas informáticas el que producirá un desarrollo del pensamiento computacional del alumno.

4.3. Elementos del pensamiento computacional

En la segunda fase de nuestro análisis, se vio que todas las definiciones incluyen habilidades, en su mayor parte de “resolución de problemas” y de programación. Como se comenta más adelante, es un acierto incluir habilidades de programación, que han ido adquiriendo importancia creciente en la enseñanza universitaria de la programación. Sin embargo, las propuestas de elementos del pensamiento computacional adolecen de incompletitud.

Un primer problema es el énfasis exclusivo de algunas definiciones en las habilidades, sin mención alguna a los conocimientos. Esta carencia presenta el riesgo claro de que el conocimiento sólo se adquiriera de forma implícita o intuitiva. Evidentemente, los alumnos tendrán una formación deficiente, al desconocer conceptos clave de la programación, incluso el propio nombre de los conceptos. Asimismo, se corre el riesgo de que la programación se perciba como una mera habilidad, sin ninguna base, un “arte” en lugar de una ciencia. La programación tiene fundamentos conceptuales propios, así como fundamentos matemáticos e ingenieriles, que no pueden ignorarse en una educación de calidad.

Últimamente se ha debatido sobre la existencia y posible identificación de ciertos conceptos, llamados conceptos umbral (Meyer & Land, 2006), que son clave en el proceso de aprendizaje de cada materia hasta el punto de cambiar la percepción que el alumno tiene de la propia materia. Por ejemplo, un firme candidato a la categoría de concepto umbral en programación es el de estado de la ejecución (Shinners-Kennedy, 2008). Se trata de un concepto “invisible” en el texto de un programa pero fundamental para entender su dinámica, por lo que debe enseñarse explícitamente. Un concepto relacionado es el de la “máquina nocional” de todo lenguaje de programación (Sorva, 2103). Sin la enseñanza explícita de este concepto, los alumnos tendrán más dificultades para aprender a programar (du Boulay et al., 1981).

Estas cuestiones son especialmente graves porque aprender a programar es una tarea difícil para los alumnos de informática (Lister et al., 2004), que exige un gran esfuerzo cognitivo (Oliver et al., 2004). Si los alumnos de etapas preuniversitarias aprenden a programar con gran facilidad, se debe a ciertos factores que simplifican enormemente la tarea de programar con lenguajes basados en bloques (Velázquez, 2018). Sin embargo,

no hay que olvidar dichas dificultades si se quiere extender la programación a lenguajes basados en texto.

Un segundo problema es la falta de precisión de algunos conceptos o habilidades. Así, no se distingue la resolución de problemas de programación y de otras clases de problemas (como los matemáticos), cuando una diferencia clave entre ambas es la existencia de un ordenador que debe ejecutar automáticamente nuestras instrucciones de resolución del problema (Aho, 2011). Lo mismo puede decirse sobre la falta de diferenciación entre algoritmo y procedimiento (o secuencia de operaciones). Un algoritmo resuelve un problema bien definido en términos de datos de entrada y de salida, mientras que un procedimiento simplemente permite realizar una tarea. Michal Armoni (2016) ha llamado la atención sobre esta falta de rigor en el uso de conceptos de programación.

4.4. Ámbito de conocimiento del pensamiento computacional

En la tercera fase de nuestro estudio, se analizó el ámbito de conocimiento. Se mezclan elementos propios de la informática (principalmente de programación) con habilidades transversales. Podemos señalar dos problemas principales.

Por un lado, encontramos la llamativa restricción del ámbito de la informática a la programación y la algoritmia, salvo en Computing at School (Berry *et al.*, 2012). Desde hace varias décadas, existe un consenso en la comunidad universitaria informática sobre el hecho de que la programación juega un papel central en la educación informática, pero también en que la educación en informática debe incluir otras áreas (ACM & IEEE Computer Society, 2013). Aunque no tenga sentido incluir todas las áreas de la informática en la educación preuniversitaria, tampoco lo tiene limitarse exclusivamente a la programación.

Por otro lado, las definiciones incluyen y, con frecuencia, enfatizan componentes heterogéneos y propuestos *ad hoc*, de distintos ámbitos. Por ejemplo, se proponen componentes propios de programación junto con componentes que no son específicos de la misma, como la abstracción o el trabajo colaborativo. La abstracción es incluso más característica de las matemáticas, mientras que el trabajo colaborativo es una habilidad transversal. La competencia digital es reclamada por Computing at School (Berry *et al.*, 2012), dentro de su visión más completa de una enseñanza informática.

La mezcla de elementos propios de la informática con elementos compartidos con otras disciplinas o con elementos transversales se realiza sin una justificación clara. Además, parece mezclarse con otra cuestión distinta, que es el uso transversal de la informática en otras materias (no la mera competencia digital), incluso la transferencia de las competencias desarrolladas a otras materias e incluso a la vida diaria para la resolución de problemas. Aunque sean afirmaciones bienintencionadas, no existe evidencia científica de dicha transferencia (Denning, 2017). Sería deseable sustituir estas afirmaciones por otras verificables y más modestas (Hemmendinger, 2010), de forma que no se levanten expectativas excesivas que posteriormente resulten defraudadas. En definitiva, sería deseable situar el pensamiento computacional al mismo nivel que otros pensamientos (pensamiento científico, pensamiento crítico, etc.), reconociendo sus virtudes y sus limitaciones.

4.5. Trabajos relacionados

Como se comentó en la introducción, el número de publicaciones realizadas sobre el pensamiento computacional en los últimos quince años es ingente, incluyendo recopilaciones o revisiones sistemáticas sobre distintos aspectos del mismo. No conocemos ningún otro trabajo que realice un análisis educativo como el nuestro, pero

obviamente existen publicaciones relacionadas. Presentamos aquellas que tienen una relación más directa con este trabajo, comentando sus semejanzas y diferencias.

Polanco, Ferrer y Fernández (2021) analizan una selección de definiciones similar a la nuestra, aunque algo más amplia. Su análisis señala el “énfasis” de cada definición, que coincide con nuestra definición explícita (véase sección 3.1), pero no profundiza en otras características o elementos.

Corradini et al. (2017) realizan un análisis del pensamiento computacional a partir de cinco fuentes, cuatro en común con nuestro análisis. Sin embargo, su estudio es instrumental para el objetivo de su investigación, que es el estudio de las concepciones y malas concepciones desarrolladas por profesores italianos sobre el término. Su análisis clasifica los elementos de las definiciones en cuatro categorías, sin un análisis crítico posterior: procesos mentales, métodos, prácticas y habilidades transversales.

Kalelioğlu, Gülbahar y Kukul (2016) analizan las definiciones presentes en 125 publicaciones, identificando los términos más frecuentes. Sin embargo, no los categorizan según su papel educativo.

Bocconi et al. (2016) reúne los elementos presentes en ocho publicaciones y los clasifica en conceptos, habilidades y actitudes. Sin embargo, el análisis se queda en una mera presentación descriptiva, sin analizar potenciales consecuencias de su análisis

5. Conclusiones

Diversos países están potenciando la enseñanza de la informática en Educación Primaria y Secundaria. Se trata de una tendencia emergente que busca que los alumnos estén mejor formados para su vida futura en una sociedad de la información. Sin embargo, el uso del término “pensamiento computacional” dificulta el debate sobre la introducción de la informática como materia en la educación preuniversitaria. En el artículo hemos presentado un análisis, desde un punto de vista educativo, de las definiciones del pensamiento computacional con el objetivo de aclarar su significado y valorar sus implicaciones educativas. Se han analizado con métodos cualitativos las definiciones realizadas por destacados autores y organizaciones. Todas las definiciones coinciden en referirse a “actividades mentales” y a habilidades de programación, pero difieren en otros componentes e incluso en su ámbito de aplicación.

Esta disparidad confirma la conveniencia de evitar el uso de este término para la planificación educativa porque introduce imprecisión en el debate. Sin embargo, tampoco puede ignorarse el éxito del término. No parece que el éxito del término pensamiento computacional se deba a su definición, sino a que ha satisfecho una necesidad que estaba latente. Como sugiere Enrico Nardelli (2019), el término puede utilizarse para resumir la necesidad percibida de proporcionar una mejor formación en informática a niños y jóvenes.

Un aspecto clave es que el pensamiento computacional no es una materia que pueda impartirse, sino que la materia de referencia es la informática. Por tanto, el debate debería centrarse en la educación informática. Obviamente, hay muchas cuestiones sin respuesta clara, que deberían abordarse con claridad conceptual y teniendo en cuenta los resultados de las investigaciones sobre el aprendizaje de la informática a nivel universitario (Fincher & Robins, 2019) y las experiencias de países pioneros (Hubwieser, Armoni, Giannakos & Mittermeir, 2014).

Cabe destacar que las asociaciones profesionales y científicas internacionales de informática han tomado partido por la opción de una asignatura obligatoria (ACM, Code.org, CSTA, Cyber Innovation Center & National Math and Science Initiative, 2016; Caspersen, Gal-Ezer, McGettrick & Nardelli, 2018; ECDL, 2015; Furber, 2012),

a los que puede añadirse un reciente informe de la Sociedad Científica Informática de España (Velázquez, 2018). De forma resumida, su planteamiento consiste en que los alumnos reciban una educación integral en informática mediante una asignatura obligatoria que incluya competencia digital y fundamentos de informática. Dicha educación se implantaría preferiblemente en Educación Primaria y Secundaria (aunque hay experiencias incluso en Educación Infantil, véase González, 2019). Es un planteamiento similar al de Pérez, Castro y Fandos (2015), pero más ambicioso al no limitarse a la competencia digital y considerar que la única forma de garantizar dicha educación integral en España es mediante una asignatura obligatoria.

Terminamos mostrando el contraste entre estas recomendaciones, bastante precisas, con la definición imprecisa del pensamiento computacional. Veamos las cuatro recomendaciones emitidas por ACM Europe e Informatics Europe (Gander et al., 2013) sobre la educación informática: (a) todos los alumnos deberían beneficiarse de una educación en competencia digital desde edades tempranas, que incida en un uso eficaz, seguro y ético de las tecnologías informáticas; (b) todos los alumnos deberían beneficiarse de una educación en informática, como una disciplina con sus propios conocimientos y habilidades; (c) debería emprenderse urgentemente un programa de formación de profesorado a gran escala; y (d) la definición del currículo debería basarse en los conocimientos generados por la comunidad investigadora en educación informática. Son cuatro recomendaciones sencillas en su planteamiento, que inciden sobre las principales dificultades de este reto, señalando caminos claros para avanzar.

Presentación del artículo: 27 de junio de 2021

Fecha de aprobación: 7 de septiembre de 2021

Fecha de publicación: 30 de noviembre de 2021

Velázquez Iturbide, J. Á. y Martín Lope, M. (2021). Análisis del “pensamiento computacional”: una perspectiva educativa. <i>RED. Revista de Educación a Distancia</i> , 21(68). http://dx.doi.org/10.6018/red.485321

Financiación

Este trabajo se ha financiado con el proyecto de investigación e-Madrid-CM (S2018/TCS-4307) de la Comunidad Autónoma de Madrid. El proyecto e-Madrid-CM también está financiado con los fondos estructurales FSE y FEDER.

Referencias

- ACM, Code.org, CSTA, Cyber Innovation Center y National Math and Science Initiative. (2016). *K-12 Computer Science Framework*. Obtenido de <https://bit.ly/2HDJcMC>.
- ACM & IEEE Computer Society. (2013). *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. Obtenido de <https://bit.ly/2E6dDGR>.
- Adell Segura, A., Llopis Nebot, M.Á., Esteve Mon, F.M. & Valdeolivas Novella, M.G. (2019). El debate sobre el pensamiento computacional en educación. *RIED. Revista Iberoamericana de Educación a Distancia*, 22(1), 171-186. DOI 10.5944/ried.22.1.22303.

- Aho, A. V. (2011). Computation and computational thinking. *ACM Ubiquity*, 2011(enero), artículo 1. DOI 10.1145/1922681.1922682.
- Armoni, M. (2016). Computing in schools: Computer science, computational thinking, programming, coding: The anomalies of transitivity in K-12 computer science education. *ACM Inroads*, 7(4), 24-27. DOI 10.1145/3011071.
- Balanskat, A. & Engelhardt, K. (2015). *Computing our Future: Computer Programming and Coding – Priorities, school curricula and initiatives across Europe*, European Schoolnet. Obtenido de <https://bit.ly/2IgUx2N>.
- Barr, V. & Stephenson, C. (2011). What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54. DOI 10.1145/1929887.1929905.
- Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*, 20(1), 45–73.
- Berry, M., et al. (2012). *Computer science: A curriculum for schools*. Computing at School. Obtenido de <https://bit.ly/2TgpdG3>.
- Blikstein, P. (2018). *Pre-College Computer Science Education: A Survey of the Field*. Google LLC. Obtenido de <https://bit.ly/2Fc0PQW>.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A. & Engelhardt, K. (2016). *Developing computational thinking in compulsory education – Implications for policy and practice*. EUR 28295 EN, Joint Research Centre. Obtenido de <https://doi.org/10.2791/792158>.
- BOE (2014). Real Decreto 126/2014, de 18 de febrero, por el que se establece el currículo básico de la Educación Primaria, 1 marzo 2014, nº 52, sec. I, pp. 19.349-19.420. Obtenido de <https://bit.ly/2CTe4qt>.
- Brennan, K. & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the Annual American Educational Research Association Meeting, AERA 2012*. Vancouver: Canadá. Obtenido de <https://bit.ly/39YYqrS>.
- Caspersen, M.E., Gal-Ezer, J., Gettrick, A. & Nardelli, E. (2018). *Informatics for All – The strategy*. ACM Europe e Informatics Europe. Obtenido de <https://bit.ly/2Y5Wllk>.
- Corradini, I., Lodi, M. & Nardelli, E. (2017). Conceptions and misconceptions about computational thinking among Italian primary school teachers. *Proceedings of the Fourteenth Annual ACM International Computing Education Research Conference, ICER 2017* (pp. 136-144). Nueva York: ACM Press. DOI 10.1145/3105726.3106194.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. & Woollard, J. (2015). *Computational Thinking. A Guide for Teachers*. Computing at School. Obtenido de <https://bit.ly/1QukAU9>.
- CSTA & ISTE. (2011a). *Operational Definition of Computational Thinking for K–12 Education*. Obtenido de <https://bit.ly/2UFCERf>.
- CSTA & ISTE. (2011b). *Computational Thinking Teacher Resources*. Obtenido de <https://bit.ly/2FdiesE>.
- Denning, P.J. (2009). Beyond computational thinking. *Communications of the ACM*, 52(6), 28-30. DOI 10.1145/1516046.1516054.
- Denning, P.J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39. DOI 10.1145/2998438.
- Dreyfus, H. & Dreyfus, S. (1986). *Mind Over Machine*. Nueva York: Free Press.

- du Boulay, B., O'Shea, T. & Monk, J. (1981). The black box inside the glass box: Presenting computing concepts to novices. *International Journal of Man-Machine Studies*, 14(3), 237-249, 1981. DOI 10.1016/S0020-7373(81)80056-9.
- ECDL Foundation (2015). *Computing and Digital Literacy: Call for a Holistic Approach*. Council of European Professional Informatics Societies. Obtenido de <https://bit.ly/2WcDLbQ>.
- Fincher, S. & Robins, A.V., Eds. (2004). *The Cambridge Handbook of Computing Education Research*. Cambridge: Cambridge University Press.
- Furber, S. (2012). *Shutdown or Restart? The Way Forward for Computing in UK Schools*. The Royal Society. Obtenido de <https://bit.ly/1RjhCi7>.
- Gander, W., Petit, A., Berry, G., Demo, B., Vahrenhold, J., McGettrick, A., Boyle, R., Drechsler, M., Mendelson, A., Stephenson, C., Ghezzi, C. & Meyer, B. (2013). *Informatics Education: Europe Cannot Afford to Miss the Boat*. Joint Informatics Europe & ACM Europe Working Group on Informatics Education. Obtenido de <https://bit.ly/2UHSo6w>.
- García Peñalvo, F.J. (2018). Computational thinking. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 13(1), 17-19. DOI 10.1109/RITA.2018.280993.
- García Peñalvo, F.J., Llorens Largo, F., Molero Prieto, X. & Vendrell Vidal, E. (2017). Educación en Informática sub 18 (EI<18). *ReVisión*, 10(2), 13-18.
- García Peñalvo, F.J., Reimann, D., Tuul, M., Rees, A. & Jormanainen, I. (2016). *An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers*. TACCLE3 Consortium. DOI 10.5281/zenodo.165123.
- González-González, C. S. (2019). State of the art in the teaching of computational thinking and programming in childhood education. *Education in the Knowledge Society*, 20, artículo 17. DOI 10.14201/eks2019_20_a17.
- Hemmendinger, D. (2010). A plea for modesty. *ACM Inroads*, 1(2), 4-7. DOI 10.1145/1805724.1805725.
- Hsieh, H.-F. & Shannon, S.E. (2005). Three approaches to qualitative content analysis. *Qualitative Health Research*, 15(9), 1.277-1.288. DOI 10.1177/1049732305276687.
- Hubwieser, P., Armoni, M., Giannakos, M.N. & Mittermeir, R.T. (2014). Perspectives and visions of computer science education in primary and secondary (K-12) schools. *ACM Transactions on Computing Education*, 14(2), artículo 7. DOI 10.1145/2602482.
- INTEF. (2018). *Programación, robótica y pensamiento computacional en el aula: Situación en España y propuesta normativa*. Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado. Obtenido de <https://bit.ly/2LCSqcf>.
- ISTE. (2016). *ISTE Standards for Students*. International Society for Technology in Education. Obtenido de <https://bit.ly/29aqQky>.
- Kalelioğlu, F., Gülbahar, Y. & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, 4(3), 583-596.
- Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, Sanders, K., Seppälä, O., Simon, B. & Thomas, L. (2004). A multinational study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin*, 36(4), 119-150. DOI 10.1145/1044550.1041673.
- Llorens Largo, F., García Peñalvo, F J., Molero Prieto, X. & Vendrell Vidal, E. (2017). La enseñanza de la informática, la programación y el pensamiento computacional en los estudios preuniversitarios. *Education in the Knowledge Society*, 18(2), 7-17. <https://doi.org/10.14201/eks2017182717>.

- Lye, S.Y. & Koh, J.H.L. (2014). Review on teaching and learning of computational thinking through programming. What is next for K-12? *Computers in Human Behavior*, 41, 51-61. DOI 10.1016/j.chb.2014.09.012.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, M., Mirolo, C., Rolandsson, L. & Settle, A. (2014). Computational thinking in K-9. *ITiCSE-WGR'14: Working Group Reports of the 2014 Conference on Innovation & Technology in Computer Science Education, ITiCSE 2014* (pp. 1-29). Nueva York: ACM Press. DOI 10.1145/2713609.2713610.
- Meyer, J.H.F. & Land, R., Eds. (2006). *Overcoming Barriers to Student Understanding: Threshold Concepts and Troublesome Knowledge*, Londres: Routledge.
- Nardelli, E. (2019). Do we really need computational thinking? *Communications of the ACM*, 62(2), 32-35. DOI 10.1145/3231587.
- Norman, D. A. (1983). Some observations on mental models. En D. Gentner and A. L. Stevens (eds.), *Mental Models*, Lawrence Erlbaum, pp. 7–14.
- Oliver, D., Dobeles, T., Greber, M. & Roberts, T. (2004). Comparing course assessments: When lower is higher and higher, lower. *Computer Science Education*, 14(4), 321-341, DOI 10.1080/0899340042000303465.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Nueva York: Basic Books.
- Pérez Escoda, A., Castro Zubizarreta, A. & Fandos Igado, M. (2016). La competencia digital de la Generación Z: claves para su introducción curricular en la Educación Primaria. *Comunicar*, XXIV(49), 71-80. DOI 10.3916/C49-2016-07.
- Polanco Padrón, N., Ferrer Planchart, S. & Fernández Reina, M. (2021). Aproximación a una definición de pensamiento computacional. *RIED. Revista Iberoamericana de Educación a Distancia*, 24(1), 55-76. DOI 10.5944/ried.24.1.27419.
- Roig-Vila, R. & Moreno-Isac, V. (2020). El pensamiento computacional en educación. Análisis bibliométrico y temático. *RED Revista de Educación a Distancia*, 20(63), artículo 5, DOI 10.6018/red.402621.
- Shinners-Kennedy, D. (2008). The everydayness of threshold concepts: State as an example from computer science. En R. Land & J. H. F. Meyer (eds.), *Threshold Concepts within the Disciplines*, Sense Publishers, pp. 119–128.
- Sorva, J. (2013). Notional machines and introductory programming education. *ACM Transactions on Computing Education*, 13(2), article 8, DOI 10.1145/2483710.2483713.
- Sweller, J., Ayres, P. y Kalyuga, S. (eds.). (2011). *Cognitive Load Theory*. Nueva York: Springer Science+Business Media.
- Tikva, C. & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature review. *Computers & Education*, 162, 104083. DOI 10.1016/j.compedu.2020.104083.
- Vahrenhold, J., Caspersen, M.E., Nardelli, E., Perreira, C., Berry, G., Gal-Ezer, J., Kölling, M., McGettrick, A. & Westermeier, M. (2017). *Informatics Education in Europe: Are We All in the same Boat?* The Committee on European Computing Education, Informatics Europe y ACM Europe. DOI 10.1145/3106077.
- Velázquez Iturbide, J.Á., Ed. (2018). *Informe del grupo de trabajo SCIE/CODDII sobre la enseñanza preuniversitaria de la informática*. Sociedad Científica Informática de España. Obtenido de <https://www.scie.es/actividades/educacion/>.
- Vogel, S., Santo, R. & Ching, D. (2017). Visions of computer science education: Unpacking arguments for and projected impacts of CS4All initiatives. *Proceedings*

- of the 2017 SIGCSE Technical Symposium on Computer Science Education, SIGCSE 2017* (pp. 609-614). Nueva York: ACM Press. DOI 10.1145/3017680.3017755.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. DOI 10.1145/1118178.1118215.
- Wing, J. (2010). *Computational thinking—What and why?* *The Link Magazine*. School of Computer Science, Carnegie Mellon University. Obtenido de <https://bit.ly/2jUgWs1>.
- Zapata Ros, M. (2015). Pensamiento computacional: una nueva alfabetización digital. *RED Revista de Educación a Distancia*, 46, DOI 10.6018/red/46/4.
- Zapata Ros, M. (2019). Computational thinking unplugged. *Education in the Knowledge Society*, 20, artículo 18. DOI 10.14201/eks2019_20_a18.