

# INICIACIÓN AL PENSAMIENTO COMPUTACIONAL A TRAVÉS DE TAREAS DESCONECTADAS SOBRE PATRONES

## Initiation to computational thinking through unplugging tasks about patterns

Ayala-Altamirano, C., Arjona-Aranda, G., Moral-Sánchez, S. N. y Sánchez-Cruzado, C.

Universidad de Málaga

### Resumen

*Dada la reciente introducción del pensamiento computacional en el currículo del área de matemáticas en España, es importante plantear propuestas posibles de implementar en el aula. En esta investigación nuestro objetivo es analizar las respuestas de un grupo de estudiantes de cuarto de primaria (9 a 10 años) cuando resuelven una tarea que involucra patrones de repetición, empleando un recurso manipulativo que simula una máquina de Turing. A partir de un análisis microgenético del proceso de solución de diversas tareas, se describe cómo desarrollan destrezas propias del pensamiento computacional en tareas desconectadas. Los resultados contribuyen a la comprensión de destrezas tales como organización de datos, la creación de algoritmos, testeo y depuración y generalización. Los algoritmos que se propusieron tienen distintos grados de eficiencia y se muestra cómo los gestos tienen un rol importante en la comunicación de las soluciones.*

**Palabras clave:** educación primaria, patrones, pensamiento computacional, tareas desconectadas, máquinas de Turing.

### Abstract

*Given the recent introduction of computational thinking in the Spanish mathematics curriculum, it is essential to set out possible proposals which could be brought to classes. Our goal in this investigation is to analyze the answers of a group of students in fourth grade in primary school (9 to 10 years old) when they solve tasks that involve patterns of repetition, using a manipulative resource which simulates a Turing machine. We describe how they develop computational thinking skills in unplugged tasks based on a microgenetic analysis. The results contribute to comprehending abilities such as data organization, algorithm creation, testing, debugging and generalization. The proposed algorithms have various degrees of efficiency and showcase the importance of gestures when communicating the solutions.*

**Keywords:** primary education, patterns, computational thinking, unplugged tasks, Turing machines.

### INTRODUCCIÓN

La legislación educativa española actual (Ministerio de Educación y Formación Profesional [MEFP], 2022), incorpora en el currículo del área de matemáticas el pensamiento computacional, en concreto, dentro de los saberes algebraicos. En la legislación autonómica se contempla la competencia específica “Utilizar el pensamiento computacional, organizando datos, descomponiendo en partes, reconociendo patrones, generalizando e interpretando, modificando y creando algoritmos de forma guiada, para modelizar y automatizar situaciones de la vida cotidiana.” (Consejería de Desarrollo Educativo y Formación Profesional, 2023, p. 139). El desarrollo del pensamiento computacional es clave para el futuro del estudiantado, ya que está directamente vinculado con la resolución de problemas y la formulación de procedimientos (Brennan y Resnick, 2012).

El diseño de propuestas didácticas basadas en el currículo es un aspecto fundamental en la labor docente, y la inclusión del pensamiento computacional en el mismo es un reto para el profesorado

que se auto percibe poco preparado para tal fin (Santaengracia et al., 2023). Resulta necesario contar con propuestas referentes que, mediante indicadores objetivos, garanticen la eficacia didáctica del uso de determinados recursos y materiales para el desarrollo de estas competencias. Se han encontrado numerosas iniciativas en las que se ha trabajado el pensamiento computacional utilizando ordenadores, las cuales incluyen lenguajes básicos de programación o robots (Acosta et al., 2022), pero pocas en las que se desarrollen estas destrezas mediante manipulación de elementos y materiales. Para Boaler (2020) al iniciar al alumnado al pensamiento computacional, concretamente en educación primaria, se deben implementar actividades experienciales manipulativas sin el uso del ordenador, fomentando así la reflexión y el razonamiento. De esta forma, se evitan potenciales inconvenientes como el aprendizaje de un lenguaje de programación o los posibles problemas logísticos, tales como disponibilidad de ordenadores o las dificultades en el acceso a internet, especialmente para el caso del alumnado en los niveles iniciales (Curzon, 2013). Es imprescindible utilizar lo que en inglés se denominan como *pedagogical unplugged experiences*, que traducimos como experiencias pedagógicas *desconectadas* (Kotsopoulos, 2017; Zhan et al., 2022).

La novedad de este estudio es aportar pruebas sobre la iniciación al pensamiento computacional a través de actividades sin ordenadores (experiencias desconectadas). El objetivo de esta comunicación es: analizar y describir cómo seis estudiantes, de cuarto de primaria (9-10 años), desarrollan habilidades propias del pensamiento computacional mientras construyen patrones de repetición en tareas desconectadas. Para esto utilizamos un recurso denominado *Turing Tumble*, juego de programación sin ordenadores, que promueve el pensamiento computacional a través de materiales manipulativos. En este estudio indagamos sobre cuáles son las capacidades del pensamiento computacional que se evidencian mientras resuelven las tareas y comunican sus ideas.

## MARCO DE REFERENCIA

No es sencillo dar una definición consensuada de pensamiento computacional (Polanco et al., 2021). Una de las primeras definiciones fue dada por Wing (2006) en la que el pensamiento computacional implicaba resolver problemas, diseñar sistemas y comprender el comportamiento humano, basándose en los conceptos fundamentales de la ciencia de la computación.

Por otro lado, Brennan y Resnick (2012), para estudiar y evaluar el pensamiento computacional, definen tres perspectivas: una relaciona con los conceptos computacionales (secuencias, ciclos o bucles, condicionales, operadores y datos, entre otros), otra perspectiva relacionada con las prácticas computacionales (iteraciones, ensayar (testear) y depurar, reusar y remezclar, abstraer y modularizar), y finalmente perspectivas computacionales (expresar, conectar y cuestionar). Por su parte, Selby y Woolland (2014) describen el pensamiento computacional como un método de resolución de problemas que incorpora procesos de reflexión y utiliza la descomposición, los algoritmos, las abstracciones, la evaluación y el reconocimiento de patrones.

El propósito de iniciar al alumnado al pensamiento computacional sin utilizar ordenadores es introducir conceptos preliminares y fundamentales sin necesidad de conocimientos técnicos, tales como lenguaje de programación o diseño de algoritmos complejos (Kotsopoulos, 2017). También ayuda a que el estudiantado pueda disfrutar trabajando de forma colaborativa y aumente su motivación e interés (Curzon et al. 2014). De igual manera, al no usar ordenadores, se evitaría la posible brecha digital que puede existir, con lo que se atendería a todo el alumnado de manera equitativa.

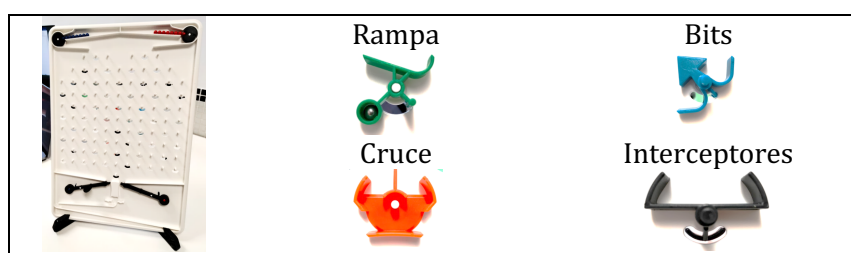
Un recurso para el desarrollo computacional serían las máquinas de Turing. Una máquina de Turing es aquella que es capaz de resolver cualquier problema por complejo que sea estableciendo un algoritmo, es decir una serie de pasos para su resolución (Roncoroni y Bailón, 2020).

Sáez y Delgado (1979) fueron pioneros simulando máquinas de Turing con materiales didácticos para ver el impacto producido en el proceso de enseñanza-aprendizaje. La investigación en este campo

sigue siendo muy activa, Rodríguez et al. (2023) han llevado a cabo una experiencia con máquinas de Turing creando un recurso educativo desconectado, construido en el marco del diseño participativo. En su investigación proporciona indicios alentadores de que este tipo de recurso educativo favorece la construcción efectiva de conocimientos disciplinares de carácter abstracto.

Desde la perspectiva de la didáctica de la matemática, en esta comunicación se proponen actividades donde se desarrollen habilidades del pensamiento computacional relacionadas con la resolución de problemas, ejecutando tareas simples en lenguaje máquina (base binaria) relacionadas con la manipulación de elementos. Esto se llevó a cabo con el juego Turing Tumble (ver figura 1), juego para montar computadoras mecánicas con componentes plásticos (rampas, cruces, interceptores, bits, entre otros). Dichas piezas simulan puertas lógicas básicas y son impulsadas por pequeñas canicas de metal que ruedan por la superficie donde se insertan los componentes. Esta máquina se relaciona con la matemática través del trabajo con patrones y secuencias lógicas, el conteo, la suma, la resta, la multiplicación y la división.

Figura 1. Juego *Turing Tumble*



Alsina (2020) recomienda seguir un itinerario de aprendizaje que incluya distintos contextos que ayuden a visualizar las ideas matemáticas de manera concreta a través de situaciones reales, materiales manipulativos y juegos. Basándonos en estas ideas, en esta experiencia buscamos promover el pensamiento computacional a través de juegos con dicho material manipulativo.

## MARCO METODOLÓGICO

La investigación forma parte de un proyecto más amplio para promover el pensamiento computacional mediante tareas que no requieran el uso de computadoras, y desarrollar destrezas socioafectivas en la enseñanza de la matemática en Educación Primaria. Tiene un enfoque cualitativo y los datos que aquí se analizan corresponden a entrevistas semiestructuradas, que forman parte de un estudio piloto. Este estudio piloto pretende evaluar la viabilidad, claridad, comprensión y funcionalidad del protocolo de investigación. El objetivo de esta comunicación es: analizar y describir cómo seis estudiantes, de cuarto de primaria (9-10 años), desarrollan habilidades propias del pensamiento computacional mientras construyen patrones de repetición en tareas desconectadas.

### Participantes y contexto

Las seis participantes fueron seleccionadas como muestra intencional según disponibilidad. Se recogió el consentimiento informado de sus progenitores para participar en la experiencia. Todas cursaban cuarto de primaria en el mismo colegio público, y aceptaron participar en la experiencia de forma positiva. Solo Alma conocía y había utilizado el juego *Turing Tumble* antes de la implementación, en un nivel muy básico. Mía tenía nociones previas de programación básica realizada en extraescolares. El resto afrontaba la experiencia sin ningún conocimiento previo. Para mantener el anonimato se han cambiado sus nombres.

### Diseño de las tareas y recolección de datos

Las entrevistas semiestructuradas constaban de tres tipos de tareas: 1) tareas con la máquina de *Turing*; 2) Tareas sobre patrones de repetición con mosaicos manipulativos; 3) tareas sobre patrones numéricos en fichas de trabajo. Cada tarea tuvo una duración entre 40 y 50 minutos.

En este trabajo solo analizamos las tareas con el juego *Turing Tumble* las cuales se llevaron a cabo en dos días. El resto de las tareas nos permitió verificar que el grupo de participantes identificaba sin dificultad patrones de repetición y numéricos. Las tareas con el *Turing Tumble* se basaron en la propuesta de Boswell y Boswell (2017), creadores del material manipulativo empleado. Las elegimos para trabajar problemas conectados y de dificultad escalada (ver tabla 1). En la implementación, las seis participantes se agruparon aleatoriamente en duplas, y se les pidió que de forma colaborativa resolvieran las situaciones que algún miembro del equipo de investigación propuso de forma oral.

Tabla 1. Descripción tareas máquina de *Turing Tumble*

<b>Tarea</b>	<b>Descripción de la tarea</b>
Introducción a la máquina	Indagación libre para reconocer cómo funciona la máquina y cada una de las piezas que la componen.
Tarea 1	A) Se presenta un camino incompleto y se pide que coloquen tantas rampas como sea necesario para que solo caigan canicas azules. B) Tras solucionar lo antes planteado se les pregunta qué camino pueden hacer para que solo caigan canicas rojas.
Tarea 2	Modificando la solución anterior o creando una nueva, se les pide obtener un camino para que la primera canica sea roja y el resto azules.
Tarea 3	Empleando el cruce se le pide construir un camino que permita crear el patrón: rojo, azul, rojo, azul; hasta que no queden más canicas en la máquina.
Tarea 4	A) Variando el camino anterior colocando un bit al empezar, se pide modificarlo para que creen el patrón: azul, azul, rojo, azul, azul, rojo. B) Se pregunta sobre cómo puede variar su solución para que, en vez de tener el patrón anterior obtengamos un patrón de azul, azul, rojo, rojo.
Tarea 5	A) Se introduce el interceptor y se pregunta cómo completar el camino para que la primera canica azul caiga en el interceptor y no abajo del todo. B) Al completar el camino, se les pide añadir piezas para que la canica que caiga en el interceptor sea roja.

## **Análisis de datos**

Las fuentes de información fueron las grabaciones en vídeo y sus transcripciones. Con base en estas, realizamos un análisis microgenético de la actividad de las participantes. Este análisis permite estudiar una habilidad, concepto o estrategia en un breve periodo. Además, permite detectar la variabilidad del comportamiento ante una misma tarea (Wertsch y Stone, 1978).

En primer lugar, basándonos en los trabajos de Brennan y Resnick (2012) y de Selvy y Woolard (2014), se identifican las destrezas del pensamiento computacional que las participantes evidenciaban en las tareas 1 a la 5. Luego, se seleccionan las destrezas mostradas y se describen según lo observado en los vídeos (ver tabla 2). Es importante tener en cuenta que, a la hora de categorizar, en este tipo de tareas las destrezas computacionales se evidencian en manifestaciones muy elementales.

Tabla 2. Categorías de análisis de tareas máquina de Turing

<b>Destrezas</b>	<b>Descripción</b>
Organizar o analizar datos	Entender los elementos que conforman la situación: a) número y el color de canicas iniciales, b) patrón de la secuencia final que se quiere lograr y c) tipo y número de piezas de las que disponen.
Crear algoritmos	Conjunto de instrucciones resultantes de la ubicación de las piezas en el tablero creando una secuencia o camino que permite obtener la respuesta a lo solicitado en la tarea (patrón repetición u otro), a partir de una disposición previa de piezas o un tablero en blanco.
Testear y depurar	A través del ensayo y error observar si se logra el objetivo de la tarea y en caso de errores, los resuelven cambiando el algoritmo testeado.
Generalizar	Transferir soluciones a otros desafíos. Extender patrones sobre el funcionamiento de las piezas del tablero.

Para analizar los algoritmos en cuanto a su eficiencia y eficacia establecimos los siguientes grados: (a) alto, se propone una solución correcta con el menor número de pasos o elementos; (b) medio, se propone una solución correcta con elementos redundantes; (c) bajo, se propone una solución, pero no es la solución del problema.

## RESULTADOS

Los resultados los dividimos en dos apartados: (a) proceso de creación del algoritmo y (b) algoritmo como respuesta a la tarea planteada. En el primer apartado mostramos las destrezas evidenciadas por las participantes. En el segundo analizaremos el grado de eficiencia de los algoritmos creados.

### El proceso de crear algoritmos desconectados

A las participantes se les presentó diversas tareas en las que debían crear distintos patrones de repetición con dificultad creciente. En la figura 2 se muestran en orden cronológico las destrezas evidenciadas por las duplas en cada una de las tareas. A modo general, las destrezas de organizar o analizar datos, crear algoritmos y, testear y depurar, se evidenciaron casi en igual medida en las tres duplas. Además, se observan siguiendo el orden mencionado. La destreza de generalizar se observa en menor medida, principalmente en aquellas tareas en las que en el diseño buscaba de forma intencionada transferir los resultados obtenidos en las tareas previas (tareas 1B, 4B y 5B).

Figura 2. Destrezas evidencias por cada dupla

Duplas	Destrezas	F	Tarea 1A	Tarea 1B	Tarea 2	Tarea 3	Tarea 4A	Tarea 4B	Tarea 5A	Tarea 5B
Alma y Mía	Organizar o analizar datos	11	■	■	■	■	■	■	■	■
	Crear algoritmos	16	■	■	■	■	■	■	■	■
	Testear y depurar	14	■	■	■	■	■	■	■	■
	Generalizar	7								
Eva y Ana	Organizar o analizar datos	17	■	■	■	■	■	■	■	■
	Crear algoritmos	14	■	■	■	■	■	■	■	■
	Testear y depurar	15	■	■	■	■	■	■	■	■
	Generalizar	4								
Sara y María	Organizar o analizar datos	12	■	■	■	■	■	■	■	■
	Crear algoritmos	15	■	■	■	■	■	■	■	■
	Testear y depurar	18	■	■	■	■	■	■	■	■
	Generalizar	5								

Al *organizar o analizar los datos*, las participantes requerían información que les permitían aclarar el número y el color de canicas; el patrón de la secuencia que se pretendía lograr o el tipo y número de piezas disponible. En las tareas 4 y 5 fue necesario organizar o analizar los datos un mayor número de veces, quizás por la dificultad del patrón a construir y porque el funcionamiento de los elementos denominados bits implica una lógica más avanzada.

Al *crear algoritmos*, las participantes ponían las piezas directamente en el tablero, o planificaban el camino sin construirlo, verbalizando y dibujando con los dedos dónde debían ir las piezas y cuál sería la trayectoria de las canicas, para ubicar las piezas en el camino acordado como solución.

El *testear y depurar* se evidenció cuando las participantes creaban un camino y querían saber si lograban el patrón pedido. En las tareas de la figura 3, las participantes creaban un camino completo y luego probaban, pero en la tarea de introducción a la máquina a veces construían partes de caminos y probaban, continuaban el camino y volvían a testear. En caso de error, dependiendo de cual fuera, lo depuraban agregando más piezas o cambiando posición alguna que ya fuera parte de la solución.

Finalmente, la *generalización* la observamos cuando transferían las soluciones anteriores a los problemas nuevos. Por ejemplo, Eva y Ana resolvieron la tarea 4A, tal como se muestra en la figura 3A. El investigador les comentó que el siguiente desafío (4B) consistía en modificar el camino para que el patrón fuera azul, azul, rojo, rojo. Tras escuchar esto, Eva apuntó a su derecha y bajo las canicas rojas comentando: “Ah, entonces hacemos aquí el mismo camino” (figura 3B). Eva vuelve a

preguntar cuál era el patrón y la dupla completa el camino comparando el lado derecho y el izquierdo (figura 3C), así lograr un camino parcialmente simétrico (figura 3D).

Figura 3. Ejemplo de generalización en la tarea 4B



### Grado de eficiencia y eficacia de los algoritmos creados

La eficiencia de los algoritmos la entendemos como la capacidad que tiene este para crear el patrón pedido en la tarea utilizando la menor cantidad de piezas posibles y de manera rápida. Por otro lado, la eficacia se refiere a la capacidad del algoritmo para resolver la tarea de manera correcta. En la figura 4 clasificamos los algoritmos creados en cada una de las tareas.

Figura 4. Tipos de algoritmos según su eficiencia y eficacia

Dupla	Tarea 1A	Tarea 1B	Tarea 2	Tarea 3	Tarea 4A	Tarea 4B	Tarea 5A	Tarea 5B
Alma y Mia	Alto	Alto	Alto	Alto	Medio	Alto	Alto	Alto
Eva y Ana	Alto	Alto	Alto	Alto	Alto	Alto	Alto	Alto
Sara y María	Alto	Medio	Alto	Medio	Medio	Medio	Alto	Alto

En la figura 4 se observa que todas las respuestas presentadas fueron eficaces, observándose solo algoritmos de grado alto y medio. Es importante señalar que lo han logrado de forma espontánea y sin realizar procesos de optimización.

Con respecto a los algoritmos con un grado medio de eficiencia, podemos observar detalles como: un uso mayor de piezas del necesario, o incluso cambios de dirección evitables. Lo que da lugar a la duplicidad de caminos y a caminos simétricos. En la figura 5 mostramos ejemplos de los algoritmos creados para la tarea 4A. En el caso de Alma y Mía a la izquierda tienen piezas innecesarias y en el caso de Sara y María, estas se observan en la parte inferior del tablero. Eva y Ana logran un algoritmo de grado alto, similar a la solución esperada.

Figura 5. Ejemplos algoritmos tarea 4A

Solución esperada	Alma y Mia	Eva y Ana	Sara y María

## DISCUSIÓN Y CONCLUSIONES

El objetivo de este trabajo es describir cómo, seis estudiantes de cuarto de primaria (9-10 años), construyen patrones de repetición en tareas desconectadas que promueven el pensamiento computacional.

Los resultados muestran que las participantes logran desarrollar destrezas computacionales, creando algoritmos sin usar ordenadores y sin recurrir a un lenguaje de programación específico (por. ej. Programación por bloques, por código o por etiqueta). Estos hallazgos están en consonancia con los

estudios de Curzon (2013) sobre los beneficios de las actividades desconectadas. Es importante considerar que en las tareas desconectadas las destrezas computacionales desarrolladas se manifiestan a un nivel elemental, no están al mismo nivel de las especificadas para programación con ordenadores. No podemos olvidar que esta experiencia es solo una iniciación al pensamiento computacional.

Las participantes interactúan con el material manipulativo, crean algoritmos con distintos grados de eficiencia y expresan con gestos las soluciones que permiten crear los patrones de repetición solicitados. La importancia de los gestos ha sido investigada desde la perspectiva del pensamiento algebraico (Radford, 2018), destacando su importancia en la expresión de generalizaciones e ideas más sofisticadas, sin tener que recurrir a la simbología alfanumérica. En nuestro caso, con gestos se comunicaron las secuencias, los bucles, condicionales y se testearon las soluciones. Estas ideas que podrían ser abstractas se favorecieron a través de las actividades desconectadas, tal como lo mencionaba Rodríguez et al. (2023), quien encontró indicios que muestran que este tipo de recursos educativos tienen amplias posibilidades para favorecer la construcción de estos conceptos abstractos.

Esta experiencia nos ha servido para evaluar la viabilidad, claridad, comprensión y funcionalidad del protocolo de investigación, nos ha permitido rediseñar determinadas tareas y procedimientos para la intervención en otros contextos. Se ha contribuido en la descripción de una propuesta didáctica basada en el currículo español, en las que las participantes pudieron desarrollar el pensamiento computacional organizando datos, reconociendo patrones y creando algoritmos. Esto puede ser un referente para el profesorado, quienes como señalan Santaengracia et al. (2023) se perciben poco preparados para este fin. Una limitación del estudio es la muestra pequeña, pero esto permitió comparar en detalle las acciones de las participantes. Dado que se trata de un estudio piloto, buscamos que la generalización de nuestros resultados se logre al contrastar estos hallazgos con otros contextos, para esto presentamos en detalle las actividades realizadas, así seguimos los criterios de calidad de las investigaciones cualitativas.

### Agradecimientos

Proyecto B1-2023\_029 financiado por Plan Propio de Investigación, Transferencia y Divulgación Científica de la Universidad de Málaga. Proyecto PID2020-113601GB-I00 financiado por MCIN/AEI/10.13039/501100011033. El equipo de investigación forma parte de los grupos de investigación: FQM-193, HUM-324 y SEJ-462.

### Referencias

- Acosta, Y., Alsina, Á. y Pincheira, N. (2023). Patrones en educación infantil: vinculando pensamiento computacional y algebraico. En C. Jiménez-Gestal, Á. A. Magreñán, E. Badillo y P. Ivars (Eds.), *Investigación en Educación Matemática XXVI* (pp. 115–122). SEIEM.
- Alsina, A. (2020). Itinerario de enseñanza para el álgebra temprana. *Revista Chilena de Educación Matemática*, 12(1), 5-20. <https://doi.org/10.46219/rechiem.v12i1.16>
- Boswell, P. y Boswell, A. (2021) *Turing Tumble: Libro de Problemas*. Upper Story.
- Boaler, J. (2020). *Mentalidades matemáticas: cómo liberar el potencial de los estudiantes mediante las matemáticas creativas, mensajes inspiradores y una enseñanza innovadora*. Editorial Sirio.
- Brennan, K. y Resnick, M. (2012, abril). New frameworks for studying and assessing the development of computational thinking. En AERA 2012 - *annual meeting of the American educational research association* (Vol. 1, p. 25). AERA
- Consejería de Desarrollo Educativo y Formación Profesional (2023). Orden de 30 de mayo de 2023, por la que se desarrolla el currículo correspondiente a la etapa de Educación Primaria en la Comunidad Autónoma de Andalucía. *Boletín nº 104*, de 2 de junio de 2023.
- Curzon, P. (2013). cs4fn and computational thinking unplugged. *WiPSE '13 Proceedings of the 8th Workshop in Primary and Secondary Computing Education*, 47–50.

- Curzon, P., McOwan, P., Plant, N. y Meagher, L. (2014). Introducing teachers to computational thinking using unplugged storytelling. *WiPSCE'14 Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, 89–92.
- Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J. y Yiu, C. (2017). A pedagogical framework for computational thinking. *Digital experiences in mathematics education*, 3, 154–171. <https://doi.org/10.1007/s40751-017-0031-2>
- Ministerio de Educación y Formación Profesional [MEFP]. (2022). Real Decreto 157/2022, de 1 de marzo, por el que se establecen la ordenación y las enseñanzas mínimas de la Educación Primaria. *Boletín Oficial del Estado*, 52, de 2 de marzo de 2022, 24386–24504.
- Polanco Padrón, N., Ferrer Planchart, S. y Fernández Reina, M. (2021). Aproximación a una definición de pensamiento computacional. *RIED. Revista Iberoamericana de Educación a Distancia*, 24(1), 55-76. <https://doi.org/10.5944/ried.24.1.27419>
- Radford, L. (2018). The emergence of symbolic algebraic thinking in primary school. En C. Kieran (Ed.), *Teaching and Learning Algebraic Thinking with 5- to 12-Year-Olds: The global evolution of an emerging field of research and practice* (pp. 3-25). Springer.
- Rodríguez, J., Parra, G., Gili, G., Parra, S., Dolz, D. y Roumec, H. (2023). A Turing Machine at Secondary School. En P. Pesado (eds) *Computer Science – CACIC 2022. CACIC 2022. Communications in Computer and Information Science*, 1778(1), 295-306. Springer, [https://doi.org/10.1007/978-3-031-34147-2\\_19](https://doi.org/10.1007/978-3-031-34147-2_19)
- Roncoroni, U. y Bailón Maxi, J. (2020). Pensamiento computacional. Alfabetización digital sin computadoras. *Revista Icono 14. Revista científica de comunicación y tecnologías emergentes*, 18(2), 379–405. <https://doi.org/10.7195/ri14.v18i2.1570>
- Román-González, M., Pérez-González, J. y Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? criterion validity of the computational thinking test. *Computers in Human Behavior*, 72, 678-691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Sáez Vacas, F. y Delgado Kloos, C. (1979). *Simulador de máquinas de Turing para fines didácticos*. [Comunicación]. 4º Congreso Informática y Automática. Madrid. Archivo digital Universidad Complutense de Madrid.
- Santaengracia, J. J., Palop, B. y Rodríguez-Muñiz, L. J. (2023). Percepciones del profesorado sobre pensamiento computacional. Estudio de una formación. En C. Jiménez-Gestal, Á. A. Magreñán, E. Badillo, E. y P. Ivars (Eds.), *Investigación en Educación Matemática XXVI* (pp. 491–498). SEIEM.
- Selby, C. y Woollard, J. (2014). *Refining an understanding of computational thinking* (pp. 1–23). University of Southampton. <https://eprints.soton.ac.uk/372410/>
- Wertsch, J. y Stone, C. A. (1978). Microgenesis as a tool for developmental analysis. *Quarterly Newsletter Laboratory of Comparative Human Cognition*, 1(1) 8-10.
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Zhan, Z., He, W., Yi, X. y Ma, S. (2022). Effect of unplugged programming teaching aids on Children’s computational thinking and classroom interaction: *With respect to Piaget’s four stages theory*. *Journal of Educational Computing Research*, 60(5), 1277-1300. <https://doi.org/10.1177/07356331211057143>