

Recursos didácticos para el desarrollo del pensamiento computacional en estudiantes de básica
secundaria y media

Monografía

Wilson Alfredo Ortega Ordoñez

Especialización en Pedagogía para el Desarrollo del Aprendizaje Autónomo

Asesor:

Mg. Numar Álvarez Cardona

Universidad Nacional Abierta y a Distancia – UNAD

Escuela Ciencias de la Educación – ECEDU

Popayán, octubre de 2020

Resumen analítico especializado (RAE)	
Título	Recursos didácticos para el desarrollo del pensamiento computacional en estudiantes de básica secundaria y media
Modalidad de Trabajo de grado	Monografía.
Línea de investigación	Educación y desarrollo humano
Autores	Wilson Alfredo Ortega Ordoñez
Institución	Universidad Nacional Abierta y a Distancia
Fecha	23 de octubre de 2020
Palabras claves	Recursos didácticos, pensamiento computacional, herramientas didácticas, programación.
Descripción.	Este documento presenta los resultados del trabajo de grado realizado en la modalidad monografía, inscrito en la línea de investigación “Educación y desarrollo humano” de la Escuela de Ciencias de la Educación (ECEDU) de la Universidad Nacional Abierta y a Distancia (UNAD), el cual se basó en el procedimiento de investigación documental para realizar una compilación de recursos didácticos valiosos para el desarrollo del pensamiento computacional en estudiantes de básica secundaria y media.
Fuentes	<p>Para el desarrollo de la investigación se usaron las siguientes fuentes principales:</p> <p>Adell, J. S., Llopis, M. A. N., Esteve, M. F. M., y Valdeolivas, N. M. G. (2019). El debate sobre el pensamiento computacional en educación. <i>RIED. Revista Iberoamericana de Educación a Distancia</i>, 22(1), pp. 171-186. doi: 10.5944/ried.22.1.22303</p> <p>Coronel-Díaz, E., y Lima-Silvain, G. (2020). El pensamiento computacional. Nuevos retos para la educación del siglo XXI Computational Thinking. New challenges for 21st century education. 20, 115-137. Recuperado de https://revistas.unc.edu.ar/index.php/vesc/article/view/27451/29019</p> <p>CSTA & ISTE (2011). <i>Caja de herramientas para el pensamiento computacional</i>. Recuperado de http://www.eduteka.org/pdfdir/PensamientoComputacional1.pdf</p> <p>Fernández, L. A. G. (2010). <i>Recursos didácticos: Elementos indispensables para facilitar el aprendizaje</i>. México: Limusa.</p>

	<p>Roig-Vila, R., y Moreno-Isac, V. (2020). El pensamiento computacional en Educación. Análisis bibliométrico y temático. <i>Revista de Educación a Distancia</i>, 20(63). doi: 10.6018/red.402621</p> <p>Wing, J. M. (2006). Computational thinking. <i>Communications of the ACM</i>, 49(3), 33-35. doi: 1118178.1118215</p>
Contenidos	<ul style="list-style-type: none"> • Portada • RAE - Resumen analítico del escrito • Índice general • Índice de tablas y figuras • Introducción • Justificación • Definición del problema • Objetivos • Marco teórico • Aspectos metodológicos • Resultados • Conclusiones y recomendaciones • Referencias
Metodología	<p>Esta investigación se realiza bajo un enfoque cualitativo teniendo en cuenta que su propósito principal es identificar los recursos pedagógicos que pueden aportar al desarrollo del pensamiento computacional en los estudiantes de básica secundaria y media. Asimismo, esta investigación se sitúa en una perspectiva metodológica de investigación documental.</p>
Conclusiones	<p>El pensamiento computacional no debe entenderse como el aprendizaje de lenguajes de programación de computadores, sino como un conjunto a habilidades que todas las personas deben desarrollar para resolver problemas. Aunque varios autores han aportado acerca de la definición del pensamiento computacional, aún no existe consenso en cuanto a las habilidades que involucra y sobre las estrategias para fomentarlas y evaluarlas.</p> <p>Para la enseñanza de los conceptos y habilidades asociadas con el pensamiento computacional se debe recurrir tanto a recursos didácticos conectados, los cuales están mediados por dispositivos tecnológicos, como a recursos didácticos desconectados, los cuales no requieren de este tipo de dispositivos, sino que incluyen actividades físicas y mentales que fomentan el desarrollo de habilidades asociadas al pensamiento computacional.</p> <p>El uso de recursos didácticos desconectados para fomentar el desarrollo del pensamiento computacional se enfoca en motivar en los estudiantes la resolución de problemas a partir de conceptos y actividades propias de la programación de computadores. Este tipo de actividades tienen la ventaja de ser</p>

	<p>atractivas para los estudiantes debido a que en su mayoría incluyen actividades físicas que son desarrolladas en equipo. Los ambientes de programación basados en bloques permiten la introducción de conceptos y habilidades relacionadas con la programación de computadores sin tener que profundizar en la sintaxis de los lenguajes. Estos ambientes son visuales, atractivos e intuitivos para los estudiantes debido a que utilizan la metáfora de las piezas de un rompecabezas para crear programas sólo arrastrando bloques que representan conceptos de la computación como las condiciones, las iteraciones, la entrada de datos, las variables, entre otros. El ambiente de este tipo más usado actualmente es Scratch.</p>
<p>Referencias bibliográficas</p>	<p>Adell, J. S., Llopis, M. A. N., Esteve, M. F. M., y Valdeolivas, N. M. G. (2019). El debate sobre el pensamiento computacional en educación. <i>RIED. Revista Iberoamericana de Educación a Distancia</i>, 22(1), pp. 171-186. doi: 10.5944/ried.22.1.22303</p> <p>Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. <i>Education Technology and Society</i>, 19, 47-57. Recuperado de https://hdl.handle.net/11245/1.547418</p> <p>Ávila Baray, H.L. (2006). <i>Introducción a la metodología de la investigación</i>. Edición electrónica. España. Recuperado de http://www.univermedios.com/wp-content/uploads/2018/08/introduccion-a-la-metodologia-de-la-investigacion.pdf</p> <p>Balladares-Burgos, J. A., Avilés-Salvador, M. R., y Pérez-Narváez, H. O. (2016). Del pensamiento complejo al pensamiento computacional: retos para la educación contemporánea. <i>Sophía</i>, 2(21), 143. doi: https://doi.org/10.17163/soph.n21.2016.06</p> <p>Bers, M. U., González-González, C., & Armas-Torres, B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. <i>Computers & Education</i>. doi:10.1016/j.compedu.2019.04.013</p> <p>Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). <i>Developing computational thinking in compulsory education implications for policy</i></p>

and practice. Sevilla: Joint Research Centre. doi:
10.2791/792158

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of Computational Thinking Skills through Unplugged Activities in Primary School. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, 65–72. doi:10.1145/3137065.3137069

Compañ-Rosique, P., Satorre-Cuerda, R., Llorens-Largo, F., y Molina-Carmona, R. (2015). Enseñando a programar: un camino directo para desarrollar el pensamiento computacional. *Revista de Educación a Distancia (RED)*, (46). doi.org/10.6018/red/46/11

Conde, M. Á., Fernández-Llamas, C., Rodríguez-Sedano, F. J., Guerrero-Higueras, Á. M., Matellán-Olivera, V., & García-Peñalvo, F. J. (2017). Promoting Computational Thinking in K-12 Students by Applying Unplugged Methods and Robotics. *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality*. doi:
10.1145/3144826.3145355

Coronel-Díaz, E., y Lima-Silvain, G. (2020). El pensamiento computacional. Nuevos retos para la educación del siglo XXI Computational Thinking. New challenges for 21st century education. 20, 115-137. Recuperado de <https://revistas.unc.edu.ar/index.php/vesc/article/view/27451/29019>

CSTA & ISTE (2011). *Caja de herramientas para el pensamiento computacional*. Recuperado de <http://www.eduteka.org/pdfdir/PensamientoComputacional1.pdf>

Dwyer, H., Hill, C., Hansen, A., Iveland, A., Franklin, D., & Harlow, D. (2015). Fourth Grade Students Reading Block-Based Programs: Predictions, Visual Cues, and Affordances. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, 111–119. doi:10.1145/2787622.2787729

	<p>Fernández, L. A. G. (2010). <i>Recursos didácticos: Elementos indispensables para facilitar el aprendizaje</i>. México: Limusa.</p> <p>Grover, S., & Basu, S. (2017). Measuring Student Learning in Introductory Block-Based Programming: Examining Misconceptions of Loops, Variables, and Boolean Logic. <i>Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education</i>, 267–272. doi:10.1145/3017680.3017723</p> <p>Hernández-Sampieri, R., Fernández-Collado, C. y Baptista-Lucio, P., (2014). <i>Metodología de la Investigación</i>, 6ta edición, México: McGraw-Hill.</p> <p>Hoyos, C. (2000). Un modelo para una investigación documental. Guía teórico- práctica sobre construcción de estados del arte. Medellín: Señal Editora.</p> <p>ISTE. (2016). <i>ISTE Standards for Students</i>. Washington DC: International Society for Technology in Education. Recuperado de https://www.iste.org/standards-for-students</p> <p>Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. <i>Computers in Human Behavior</i>, 52, 200–210. doi:10.1016/j.chb.2015.05.047</p> <p>Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., & Mendes, A. J. (2018). Learning Computational Thinking and scratch at distance. <i>Computers in Human Behavior</i>, 80, 470–477. doi:10.1016/j.chb.2017.09.025</p> <p>Roig-Vila, R., y Moreno-Isac, V. (2020). El pensamiento computacional en Educación. Análisis bibliométrico y temático. <i>Revista de Educación a Distancia</i>, 20(63). doi: 10.6018/red.402621</p> <p>Scott, C.L. 2015. <i>El futuro del aprendizaje 2 ¿Qué tipo de aprendizaje se necesita en el siglo XXI?</i> Investigación y Prospectiva en Educación UNESCO, París. [Documentos de Trabajo ERF, No. 14]. Recuperado de https://unesdoc.unesco.org/ark:/48223/pf0000242996_spa</p> <p>Valverde-Berrocoso, J., Fernández-Sánchez, M. R., y Garrido-Arroyo, M. C. (2015). El pensamiento computacional y</p>
--	--

	<p>las nuevas ecologías del aprendizaje. <i>Revista de Educación a Distancia</i>, (46). Recuperado de https://revistas.um.es/red/article/view/240311</p> <p>Weinberg, A. (2013). <i>Computational Thinking: An Investigation of the existing scholarship and research (tesis doctoral)</i>. Universidad estatal de Colorado. Fort Collins, Colorado. Recuperado de https://mountainscholar.org/handle/10217/78883</p> <p>Weintrop, D., & Wilensky, U. (2015). To Block or Not to Block, That is the Question: Students' Perceptions of Blocks-Based Programming. <i>Proceedings of the 14th International Conference on Interaction Design and Children</i>, 199–208. doi: 10.1145/2771839.2771860</p> <p>Weintrop, D., & Wilensky, U. (2017). Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms. <i>ACM Trans. Comput. Educ.</i>, 18(1). https://doi.org/10.1145/3089799</p> <p>Wing, J. M. (2006). Computational thinking. <i>Communications of the ACM</i>, 49(3), 33-35. doi: 1118178.1118215</p> <p>Wing, J. M. (2008). Computational thinking and thinking about computing. <i>Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences</i>, 366(1881), 3717-3725. doi: 10.1098/rsta.2008.0118</p> <p>Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. <i>RED. Revista de Educación a Distancia</i>, (46). Recuperado de https://www.um.es/ead/red/46/</p>
--	---

Tabla de contenido

Resumen analítico especializado (RAE).....	2
Introducción	10
Justificación	12
Definición del problema	14
Objetivos	16
Objetivo general.....	16
Objetivos específicos	16
Marco teórico	17
El concepto de pensamiento computacional	17
Habilidades del pensamiento computacional.....	20
La importancia de los recursos didácticos	22
Criterios para seleccionar recursos didácticos	23
Desafíos existentes para el uso de recursos didácticos en el desarrollo del pensamiento computacional	23
Recursos didácticos para el desarrollo del pensamiento computacional en los estudiantes	25
Aspectos metodológicos	28
Resultados	31
Conclusiones y recomendaciones	37
Referencias.....	39

Tabla de figuras

Figura 1. Definición operativa del pensamiento computacional.	19
Figura 2. Fases de la investigación documental.....	28
Figura 3. Ejemplo del ambiente de programación basado en bloques Scratch.....	35
Figura 4. Ejemplo del tutorial “La hora del código” en Code.org.....	36

Introducción

Actualmente, existe un creciente interés en la integración del pensamiento computacional en la educación formal, el cual está motivado por la necesidad de formar personas con las habilidades necesarias para enfrentar los retos del siglo XXI. Entre estas habilidades se destacan: la capacidad de la resolución de problemas, la creatividad, el trabajo en equipo, y el pensamiento algorítmico. En este sentido, el desarrollo del pensamiento computacional en los estudiantes se considera fundamental para resolver problemas de la vida cotidiana aplicando conceptos propios de las ciencias de la computación.

Teniendo en cuenta la importancia del desarrollo del pensamiento computacional en los estudiantes, los gobiernos de numerosos países han lanzado iniciativas para su integración en el currículo de los sistemas educativos formales. Sin embargo, la falta de consenso en cuanto a la definición del pensamiento computacional y las habilidades que abarca ha dificultado la definición de estrategias aceptadas internacionalmente para fomentar su desarrollo. Lo anterior, ha provocado que la didáctica asociada a este tipo de pensamiento quede por fuera de la formación y actualización docente.

Por otra parte, el pensamiento computacional se ha asociado normalmente con el aprendizaje de lenguajes de programación y el desarrollo de software, encasillándolo como un tema a tratar sólo en las asignaturas o cursos de informática. Lo anterior, lo ha alejado de la visión del pensamiento computacional como un conjunto de habilidades que todas las personas deben tener independientemente de su profesión y que pueden ser desarrolladas desde cualquier área del currículo.

Teniendo en cuenta la problemática descrita, mediante esta monografía se pretende identificar los recursos didácticos que pueden ser usados por los docentes de educación básica

secundaria y media que deseen fomentar el desarrollo del pensamiento computacional en sus estudiantes. Para identificar estos recursos se realizó un análisis documental a partir de artículos indexados en bases de datos académicas que presentaran resultados de investigaciones en el tema de pensamiento computacional en la educación.

A partir del análisis de los documentos seleccionados fue posible identificar recursos didácticos que pueden ser utilizados para el desarrollo del pensamiento computacional en estudiantes de educación básica secundaria y media, los cuales se categorizaron en recursos conectados y desconectados teniendo en cuenta si requieren o no del uso de dispositivos tecnológicos, respectivamente.

La monografía tiene la siguiente organización: en primer lugar, se presenta la justificación y definición del problema, donde se explica la importancia del desarrollo del pensamiento computacional en el siglo XXI y la dificultad relacionada con la definición de una didáctica para su desarrollo en los ambientes de educación formal. Luego, se presentan los objetivos de la investigación y la metodología empleada. En el capítulo denominado marco teórico se presenta la literatura asociada con la definición del pensamiento computacional, sus habilidades, y las estrategias y recursos que se han empleado para su inclusión en la educación formal. En el capítulo de resultados se presentan los recursos didácticos identificados describiendo su nombre, objetivo y las habilidades relacionadas con el pensamiento computacional que ayudan a desarrollar. Para finalizar, se presentan las conclusiones y recomendaciones para trabajos futuros en la temática.

Justificación

La preparación de los estudiantes para el siglo XXI constituye un gran reto que demanda la adquisición de conocimientos y habilidades que distintos estamentos han denominado competencias del siglo XXI. Aunque existen diversas opiniones acerca de cuáles deben ser estas competencias, varios autores coinciden en que los estudiantes, desde edades tempranas de la educación formal, deben desarrollar habilidades de pensamiento crítico, resolución de problemas, comunicación y colaboración, creatividad e innovación, uso adecuado de las tecnologías de la información y la comunicación (TIC), pensamiento emprendedor, trabajo en equipo, entre otras (Scott, 2015).

Wing (2006) catalogó el pensamiento computacional como una habilidad fundamental para todas las personas en el siglo XXI, inclusive con la misma importancia que representa la lectura, la escritura y la aritmética. Además, Wing (2006) definió el pensamiento computacional como “La resolución de problemas, el diseño de sistemas, y la comprensión de la conducta humana haciendo uso de los conceptos fundamentales de las ciencias de la computación” (p.33). El pensamiento computacional no debe ser entendido como un sinónimo de programación de computadores, sino como una competencia compleja de alto nivel que todas las personas deben desarrollar para resolver problemas de una manera inteligente e imaginativa (Valverde-Berrocoso, Fernández-Sánchez y Garrido-Arroyo, 2015).

En los últimos años, se ha dado una tendencia creciente a nivel internacional hacia la introducción del pensamiento computacional en las escuelas, especialmente mediante la creación de nuevas asignaturas relacionadas con las nuevas tecnologías, la programación de computadores y la robótica. Además, en Estados Unidos, se han implementado experiencias de diseño de entornos de aprendizaje basados en el juego para desarrollar capacidades vinculadas al

pensamiento computacional desde un punto de vista individual, social y cultural. En estos entornos se ha promovido la resolución de problemas complejos de matemáticas, ciencias, lenguaje, sociales y artes. (Valverde-Berrocoso et al., 2015).

Teniendo en cuenta la importancia del pensamiento computacional para la sociedad actual, es necesario que en la educación formal se desarrollen actividades que promuevan su desarrollo en los estudiantes. En este trabajo, se presentan los resultados de una revisión documental que tuvo por objetivo identificar los principales recursos didácticos que pueden ser usados por los docentes de educación básica secundaria y media para promover el desarrollo del pensamiento computacional en sus estudiantes. De esta manera, se pretende aportar al desarrollo de prácticas docentes que estén alineadas con las necesidades de aprendizaje de los estudiantes del siglo XXI y que a su vez puedan ser integradas en las distintas áreas de enseñanza.

Definición del problema

La integración del pensamiento computacional en la enseñanza obligatoria se ha convertido en un tema de creciente interés debido a que se relaciona con un conjunto de habilidades necesarias para tener mejores oportunidades en el sistema educativo, el mundo del trabajo y la vida. Entre estas habilidades, las que se citan con mayor frecuencia son: abstracción, pensamiento algorítmico y descomposición de problemas (Coronel-Díaz y Lima-Silvain, 2020).

Los gobiernos de numerosos países han apostado por la integración del pensamiento computacional en la educación obligatoria desde edades tempranas. Sin embargo, su definición ha estado en construcción en los últimos años, por lo cual ha sido complicado incluir su didáctica en la formación inicial y permanente del profesorado. (Adell, Llopis, Esteve y Valdeolivas, 2019). Además, cada país ha diseñado estrategias propias para la enseñanza de los conceptos asociados al pensamiento computacional de acuerdo con su realidad particular (Coronel-Díaz y Lima-Silvain, 2020).

Zapata-Ros (2015) señala que la forma habitual que han usado los países para incluir el pensamiento computacional en las escuelas ha sido favorecer el aprendizaje de la programación y de sus lenguajes desde las primeras etapas de formación. De esta manera, se ha dado una progresión desde tareas de programación sencillas y lúdicas hacia tareas cada vez más complejas y aburridas. Una propuesta mucho más del lado del construccionismo plantea una alfabetización que ponga a los niños en un entorno de objetos y de acciones que promuevan aprendizajes adecuados para favorecer el pensamiento computacional a través de la observación y de la manipulación.

Cuando el desarrollo del pensamiento computacional se reduce a aprender lenguajes de programación y desarrollar programas desde los primeros años de vida, se pierde la visión

aportada por Wing (2006) acerca de este tipo de pensamiento, el cual se entiende como una habilidad que debe ser desarrollada por todas las personas, independientemente de su formación disciplinar, para solucionar los problemas de la vida cotidiana. El hecho de convertir el desarrollo del pensamiento computacional en clases de lenguajes de programación lo convierte en un área de interés solamente para quienes deseen continuar sus estudios técnicos y profesionales en el campo del desarrollo de software.

Teniendo en cuenta lo anterior, es necesario que se identifiquen los recursos didácticos que pueden ser integrados por los docentes de educación básica secundaria y media para fomentar el desarrollo del pensamiento computacional en sus estudiantes independientemente de su interés en el desarrollo de software como área de desempeño laboral. De esta manera, es posible abordar el pensamiento computacional en las aulas desde la visión de Zapata-Ros (2015), quién señala que la programación de computadores es solamente la parte más visible de este tipo de pensamiento, pero es aún más importante el desarrollo de habilidades como: análisis, pensamiento divergente, creatividad, resolución de problemas, pensamiento abstracto, recursividad, iteración, entre otros, la cuales son útiles en todos los ámbitos de la vida.

Objetivos

Objetivo general

Identificar recursos didácticos que apoyen al estudiante de educación básica secundaria y media en el desarrollo del pensamiento computacional, a partir de un proceso de revisión documental.

Objetivos específicos

1. Seleccionar artículos indexados en bases de datos académicas que presenten información acerca de recursos didácticos usados para el desarrollo del pensamiento computacional.
2. Caracterizar los recursos didácticos que pueden ser usados para apoyar el desarrollo del pensamiento computacional en los estudiantes de educación básica secundaria y media.
3. Describir los aportes de los recursos didácticos identificados para el desarrollo del pensamiento computacional en estudiantes de básica secundaria y media.

Marco teórico

El concepto de pensamiento computacional

Coronel-Díaz y Lima-Silvain (2020) sostienen que el potencial de las computadoras en la educación fue reconocido en los años 60 por Seymour Papert quien es considerado como uno de los precursores del concepto de pensamiento computacional. Papert participó en el diseño del lenguaje de programación Logo, el cual estaba dirigido a los niños y permitía programar los movimientos de una tortuga en la pantalla de un computador. Este ambiente proporcionaba a los niños la posibilidad de asumir el rol de maestros, planificando un método de aprendizaje, dividiendo el conocimiento en temas más sencillos, creando una solución usando el lenguaje de programación y estableciendo el conocimiento adquirido como una base para futuras experiencias de aprendizaje. Aunque en esta época no se usó el término de pensamiento computacional, muchas de las habilidades que lo sustentan estaban presentes en el uso del lenguaje Logo.

El origen del término pensamiento computacional normalmente se sitúa en una publicación de Wing (2006), en la cual lo definió como:

Una habilidad fundamental para todos los individuos, no sólo para los científicos en computación. (...) involucra la resolución de problemas, el diseño de sistemas, y la comprensión de la conducta humana haciendo uso de los conceptos fundamentales de las ciencias de la computación.” (p.33).

De acuerdo con lo anterior, cuando se habla de pensamiento computacional se hace a referencia a una habilidad que todas las personas de la sociedad debemos desarrollar, por lo cual, el fomento de su desarrollo debería formar parte de la educación formal.

Teniendo en cuenta el eco de los aportes de Wing, la *Computer Science Teachers Association (CSTA)* y la *International Society for Technology in Education (ISTE)* elaboraron una propuesta de definición operacional del pensamiento computacional para facilitar su inclusión en el sector educativo formal de los Estados Unidos. Según CSTA & ISTE (2011):

El pensamiento computacional es un enfoque para resolver un determinado problema que empodera la integración de tecnologías digitales con ideas humanas. No reemplaza el énfasis en creatividad, razonamiento o pensamiento crítico, pero refuerza esas habilidades al tiempo que realza formas de organizar el problema de manera que el computador pueda ayudar. (...) refuerza los estándares educativos en todas las asignaturas para acrecentar la habilidad del aprendiz para solucionar problemas y comprometerse con pensamientos de orden superior (p.8).

En este sentido, el desarrollo del pensamiento computacional debe ser transversal al currículo debido a que en todas las disciplinas se deben analizar y resolver problemas. Mediante el desarrollo de las habilidades propias de este tipo de pensamiento, los estudiantes podrán mejorar sus desempeños dentro y fuera de las aulas. En la Figura 1 se pueden observar las características del pensamiento computacional según la definición operativa proporcionada por CSTA & ISTE (2011).

Unos años más tarde, ISTE (2016) definió el pensamiento computacional en el ámbito educativo como el desarrollo y la utilización por parte de los estudiantes de estrategias que les permitan comprender y resolver problemas aprovechando el poder de los métodos tecnológicos para desarrollar y probar soluciones. Esta definición, destaca la habilidad de comprensión y

resolución de problemas usando conceptos de la computación como eje central del pensamiento computacional.

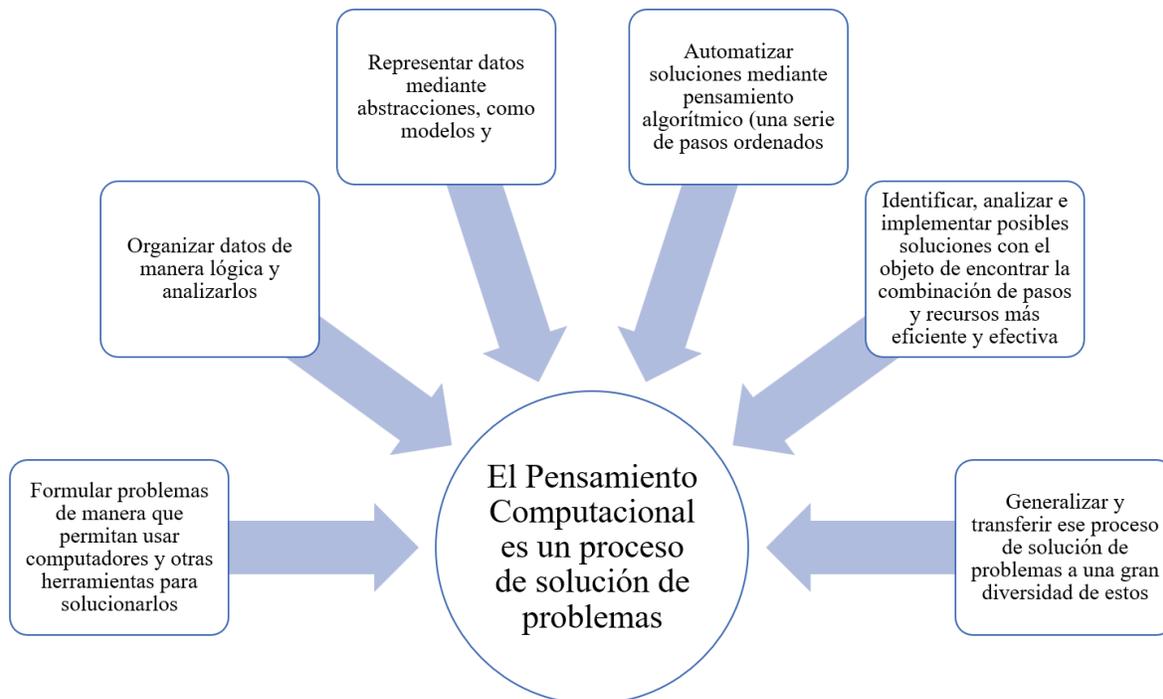


Figura 1. Definición operativa del pensamiento computacional.

Es habitual que se asocie el pensamiento computacional solamente con el ámbito de la ingeniería informática y la computación. Sin embargo, es una competencia compleja que se aplica en múltiples aspectos de la vida diaria. Por lo tanto, no se debe considerar como la capacidad de programar un computador y no es necesario que esté asociado al uso de dispositivos electrónicos (Roig-Vila y Moreno-Isac, 2020).

A pesar de la gran cantidad de estudios acerca del pensamiento computacional, su definición sigue siendo emergente (Weinberg, 2013). Sin embargo, la mayoría de los enfoques están en la línea propuesta por Wing (Coronel-Díaz y Lima-Silvain, 2020). Teniendo en cuenta

lo anterior, se debe considerar el pensamiento computacional como un componente importante para preparar a los estudiantes de todos los niveles para su desempeño en el siglo XXI.

Habilidades del pensamiento computacional

Existen varias propuestas acerca de las habilidades relacionadas con el pensamiento computacional. A continuación, se presentarán las propuestas de algunos autores.

ISTE (2016) define las siguientes habilidades relacionadas con el pensamiento computacional:

- Formulación de problemas soportada por análisis de datos, modelos abstractos, y pensamiento algorítmico.
- Recolección de datos e identificación de datos relevantes usando herramientas digitales para su representación y análisis.
- Descomposición de problemas para extraer información clave y desarrollo de modelos descriptivos para entender sistemas complejos.
- Comprensión de la automatización y uso del pensamiento algorítmico para crear y probar soluciones automatizadas.

Para Bocconi et al. (2016), las habilidades esenciales del pensamiento computacional son la abstracción para reducir los detalles innecesarios de los artefactos y problemas, el pensamiento algorítmico para encontrar una serie de pasos para solucionar un problema, la automatización para ejecutar tareas repetitivas de manera eficiente usando un computador, la descomposición de problemas para hacerlos más fáciles de resolver, la depuración para analizar y predecir salidas, y la generalización para resolver problemas basándose en soluciones encontradas previamente.

Por otra parte, Coronel-Díaz y Lima-Silvain (2020) proponen un conjunto de habilidades esenciales del pensamiento computacional a partir del análisis de las definiciones provistas por varios autores. A continuación, se describen estas habilidades:

- Capacidad de abstracción: Reducir los datos innecesarios para mejorar la comprensión.
- Capacidad de análisis de información: Análisis y organización de datos primarios, detectar relaciones significativas y representarlos para que se transformen en conocimiento útil.
- Pensamiento algorítmico: Definir una secuencia ordenada de pasos para resolver un problema.
- Automatización: Usar un programa ejecutado por un computador para que un algoritmo se ejecute automáticamente.
- Capacidad de análisis: División de un problema mediante su división en sus componentes más pequeños con el fin de estudiarlo en profundidad.
- Pensamiento recursivo: Se asocia con la capacidad de análisis. Permite analizar a cada uno de los componentes que hacen parte de un problema para solucionarlo por partes.
- Simulación: Representar datos mediante abstracciones usando modelos y simulaciones. Además, formular los problemas de manera que se puedan usar computadoras para su solución mediante la programación.
- Evaluación y depuración: Evaluar y validar los productos generados con el fin de mejorar su diseño.
- Generalización: Identificar patrones, similitudes y conexiones con el fin de resolver problemas asociados a otros que fueron resueltos anteriormente.

La importancia de los recursos didácticos

Según Fernández (2010), los recursos didácticos se pueden definir como “Los elementos (técnicas, procedimientos, actividades y también materiales) que permiten apoyar el logro de aprendizajes significativos” (p.9). En este sentido, los recursos pueden ser considerados didácticos cuando se usan de manera intencional para propiciar el aprendizaje (Fernández, 2010). Con base en lo anterior, en el proceso de enseñanza-aprendizaje, se puede usar una extensa lista de recursos didácticos que propicien aprendizajes específicos en los estudiantes y que se adapten a sus gustos y necesidades. Estos aprendizajes no se limitan sólo a conceptos, sino que pueden ser extendidos al desarrollo de destrezas intelectuales, psicomotrices, afectivas y sociales.

A continuación, se explican algunas de las funciones de los recursos didácticos cuando se usan de manera adecuada, pertinente y oportuna (Fernández, 2010):

- **Innovación:** Mediante el uso de nuevos materiales y actividades en el proceso de enseñanza-aprendizaje que impliquen cambios en la manera en que interactúan y participan los estudiantes.
- **Motivación:** Captar el interés y la atención de los estudiantes se constituye en un proceso cognitivo fundamental para el aprendizaje.
- **Estructuración de la realidad:** Se puede hacer uso de modelos que permitan representar la realidad y hacerla más cercana.
- **Mediación de procesos cognitivos:** El docente puede intervenir con los recursos de manera intencionada para estimular el desarrollo de destrezas mentales en los estudiantes.
- **Orientación del aprendizaje:** Los recursos se seleccionan para facilitar la construcción del conocimiento por parte de los estudiantes.

- **Función formativa:** La selección, uso y configuración de los recursos permite estructurar los saberes y generar nuevas visiones del mundo.

Criterios para seleccionar recursos didácticos

A continuación, se presentan algunos criterios que se pueden tener en cuenta al momento de seleccionar los recursos didácticos que se integrarán en el proceso de enseñanza-aprendizaje según la propuesta de Fernández (2010):

- **Coherencia con el plan didáctico:** Los recursos seleccionados deben estar alineados con el modelo educativo y curricular institucional.
- **Adecuación al contexto y disponibilidad:** Se debe tener en cuenta la disponibilidad de recursos en el entorno del estudiante sin caer en gastos excesivos.
- **Diversidad:** Los recursos deben ser diversos teniendo en cuenta la diversidad existente en los estudiantes en cuanto a los aspectos demográficos, estilos de aprendizaje, motivaciones e intereses.
- **Rigor académico:** Se deben revisar los recursos para evitar errores conceptuales y ofrecer un mensaje válido y confiable al estudiante.
- **Congruencia con los valores:** Los recursos se deben alinear con los valores que se fomentan en la institución.

Desafíos existentes para el uso de recursos didácticos en el desarrollo del pensamiento computacional

Para Wing (2008), el pensamiento computacional representa un desafío para el sector educativo debido a que al ser considerado como un conjunto de habilidades que todas las

personas deben desarrollar, se debe establecer cómo y cuándo las personas deben aprender esta clase de pensamiento, así como la forma de enseñarlo. En esta línea, según Wing (2008), los sistemas educativos formales se enfrentan a distintos desafíos entre los que se encuentran:

- ¿Cuáles son las maneras más efectivas para enseñar el pensamiento computacional a los niños?
- ¿Cuáles son los conceptos fundamentales del pensamiento computacional?
- ¿Cuál sería el orden correcto de los conceptos relacionados para enseñar el pensamiento computacional a los niños?
- ¿Cómo se debe integrar el uso de los computadores con la enseñanza de los conceptos del pensamiento computacional?

La falta de consenso en cuanto a las habilidades esenciales del pensamiento computacional agrega complejidad a los desafíos descritos anteriormente. En este sentido, es importante tener claridad acerca de cuáles son las habilidades que se deben desarrollar con el fin de analizar sus relaciones, priorizarlas y diseñar las estrategias de enseñanza adecuadas.

Otro desafío importante para los docentes radica en su formación y actualización constante debido a la complejidad y rápida evolución de las ciencias de la computación y las tecnologías relacionadas. Además, la literatura de contenido pedagógico que sirva de apoyo a los docentes para la enseñanza de los conceptos de ciencias de la computación es escasa (Coronel-Díaz y Lima-Silvain, 2020). En el campo de las ciencias de la computación, el cambio de tecnologías y de paradigmas y lenguajes de programación es constante, por lo cual es necesaria una constante actualización de los docentes en plataformas, herramientas y lenguajes de programación.

Recursos didácticos para el desarrollo del pensamiento computacional en los estudiantes

Teniendo en cuenta los desafíos expuestos anteriormente, la inclusión del pensamiento computacional en el currículo ha sido un tema de debate. Algunos países han incluido una asignatura independiente en la educación secundaria que puede ser optativa u obligatoria. Otros países han seguido la visión de Wing (2016) de integrarla en varias áreas y asignaturas, entendiendo que el objetivo no es enseñar a pensar como un programador, sino a usar los conceptos asociados a la computación para solucionar problemas de distintas disciplinas.

Según Balladares-Burgos, Avilés-Salvador y Pérez-Narváez (2016), una de las iniciativas con mayor impacto para la incorporación del pensamiento computacional en el sistema educativo ha sido la llevada a cabo por Estados Unidos, la cual se denominó CODE y surgió con el objetivo de promover la enseñanza de la programación para motivar a los estudiantes a continuar sus estudios de educación superior en áreas relacionadas con las ciencias de la computación. Por otro lado, en el Reino Unido se diseñó una iniciativa denominada Code Club donde una red de voluntarios realiza actividades relacionadas con la informática para incentivar la creatividad, el pensamiento lógico y la solución de problemas en niños de nueve a once años.

Al analizar la literatura relacionada con la incorporación del pensamiento computacional en la educación, se puede observar que las iniciativas se han centrado principalmente en la enseñanza y aprendizaje de lenguajes de programación visuales. (Adell et al., 2019). En este sentido, la “codificación” se considera una nueva alfabetización del siglo XXI. Esta actividad consiste en escribir el código de los programas de computador, el cual es una representación simbólica en un lenguaje de programación. La codificación puede ser considerada como una herramienta para enseñar pensamiento computacional (Bers, González-González & Armas-Torres, 2019). Por otro lado, a menudo se usan los términos codificación y programación

indistintamente, sin embargo, se debe tener claro que la programación hace referencia al conjunto amplio de actividades de análisis de un problema, diseño e implementación de la solución. Mientras que la codificación hace alusión solamente a la fase de implementación de la solución en un lenguaje particular de programación (Bocconi et al., 2016).

Una asignatura introductoria a la programación es una buena alternativa para desarrollar habilidades de pensamiento computacional, pero se debe entender la enseñanza de la programación como fomentar la capacidad de análisis de problemas y de diseño de soluciones adecuadas dejando a un lado el lenguaje de programación que se usará para realizar la implementación de la solución. Esta actividad es compleja y requiere de un alto esfuerzo de abstracción para expresar una solución en forma de un algoritmo (Compañ-Rosique, Satorre-Cuerda, Llorens-Largo, y Molina-Carmona, 2015).

Con respecto a los lenguajes de programación usados en la enseñanza de la programación, Scratch es el más utilizado, sin embargo, se ha encontrado que muchos investigadores han diseñado materiales de enseñanza de conceptos relacionados con el pensamiento computacional usando sólo papel y lápiz (Roig-Vila y Moreno-Isac, 2020). En este sentido, se puede afirmar que la falta de dispositivos tecnológicos no debe representar un limitante a la hora de fomentar el desarrollo del pensamiento computacional en las instituciones educativas.

Con relación a la didáctica del pensamiento computacional, Adell et al. (2019) afirman:

El PC ha irrumpido con fuerza en los sistemas educativos de un creciente número de países (...). Sin embargo, por un lado, existe una evidente falta de consenso entre los expertos a nivel mundial sobre un marco conceptual que lo defina con precisión

y que describa sus principales componentes. La consecuencia es que no disponemos de criterios sobre cómo enseñarlo en las distintas etapas educativas, cómo integrarlo dentro de las disciplinas actuales o sobre si es preferible que sea una actividad extraescolar, ni en cómo evaluarlo o formar al profesorado de los distintos niveles (p.181).

De acuerdo con lo anterior, existen brechas en el área del pensamiento computacional en educación que pueden ser abordadas desde proyectos de investigación que estudien aspectos importantes para la enseñanza-aprendizaje del pensamiento computacional como didácticas, formación docente, herramientas, estrategias pedagógicas, entre otros.

Aspectos metodológicos

Esta investigación se realiza bajo un enfoque cualitativo teniendo en cuenta que su propósito principal es identificar los recursos pedagógicos que pueden aportar al desarrollo del pensamiento computacional en los estudiantes de básica secundaria y media, a partir de un análisis documental. En este sentido, Hernández-Sampieri, Fernández-Collado y Baptista-Lucio (2014) afirman que: "el enfoque cualitativo se selecciona cuando el propósito es examinar la forma en que los individuos perciben y experimentan los fenómenos que los rodean, profundizando en sus puntos de vista, interpretaciones y significados" (p.358).

Asimismo, esta investigación se sitúa en una perspectiva metodológica de investigación documental, la cual es definida por definida por Ávila (2006) como: "la investigación documental es una técnica que permite obtener documentos nuevos en los que es posible describir, explicar, analizar, comparar, criticar entre otras actividades intelectuales, un tema o asunto mediante el análisis de fuentes de información" (p.50).

El desarrollo de la investigación documental se llevó a cabo en cuatro fases según los lineamientos propuestos por Hoyos (2000) (Ver Figura 2).



Figura 2. Fases de la investigación documental.

A continuación, se describen las fases del proceso de investigación documental:

- **Fase preparatoria:** En esta fase se realizó una revisión y apropiación de los elementos teóricos que sustentan la investigación y se definió el cronograma de

actividades. Además, se seleccionaron las bases de datos académicas a consultar para obtener los artículos relacionados con el tema de investigación. A continuación, se listan las bases de datos seleccionadas:

- Dialnet
- Redalyc
- SciELO
- Scopus

En cada una de las bases de datos se aplicaron los siguientes criterios de selección de artículos:

- Idioma: Español o Inglés.
 - Acceso: Libre.
 - Tipo de publicación: Artículos publicados en revistas. De esta manera se verifica la calidad de la publicación teniendo en cuenta que han sido objeto de revisión por parte de editores y comités de evaluación.
 - Fecha de publicación: Artículos publicados en los últimos 5 años.
 - Contenido: Artículos que presenten información acerca de recursos didácticos usados para el desarrollo del pensamiento computacional en estudiantes de educación básica secundaria y media.
-
- **Fase descriptiva:** Esta fase comprende el trabajo de campo realizado para encontrar los diferentes estudios que se han efectuado sobre el tema de investigación. Durante esta fase, se inició la búsqueda de artículos en cada una de las bases de datos

académicas seleccionadas aplicando los criterios de selección definidos. Para cumplir este propósito, se realizó una lectura exploratoria de cada artículo obtenido como resultado de la búsqueda en las bases de datos. Esta lectura estuvo centrada en el título, resumen, palabras clave y conclusiones. Para los artículos que cumplieron con los criterios de selección, se realizó una lectura detallada para extraer toda la información relacionada con los recursos didácticos propuestos para el desarrollo del pensamiento computacional en estudiantes de educación básica secundaria y media.

- **Fase interpretativa:** Durante esta fase se realizó un análisis detallado de los artículos seleccionados con el fin de identificar y categorizar los recursos didácticos propuestos para el desarrollo del pensamiento computacional en estudiantes de educación básica secundaria y media.
- **Fase de reporte de resultados:** Esta última fase permitió organizar el reporte de resultados de la investigación describiendo los recursos didácticos propuestos para el desarrollo del pensamiento computacional en estudiantes de educación básica secundaria y media. Además, se identificaron las conclusiones de la investigación y las recomendaciones para trabajos futuros.

Resultados

En este capítulo se presentan los recursos didácticos identificados en el análisis documental realizado, los cuales han sido agrupados en dos categorías: Actividades desconectadas y actividades conectadas.

Actividades desconectadas

Un enfoque para el desarrollo del pensamiento computacional bastante usado en muchos países es el de los métodos desconectados, en los cuales se enseña computación mediante actividades que no requieren el uso de tecnología. Este tipo de actividades involucra resolver problemas para conseguir un objetivo, mientras que durante el proceso se trabaja con conceptos propios de la computación. Además, al involucrar actividades físicas, se tornan dinámicas y atractivas para los estudiantes (Bocconi et al., 2016).

Los métodos desconectados han sido usados con éxito para enseñar computación a estudiantes de todas las edades desde primaria hasta niveles universitarios. Un ejemplo de una actividad desconectada que puede ser usada en clase es proporcionar un conjunto de fichas de colores a los estudiantes y un papel con un gráfico que representa un camino con un inicio y un fin, en el cual se deberán ir ubicando las fichas en secuencia de acuerdo con unos colores específicos. Los estudiantes tendrán que escribir los pasos para ir ubicando las fichas en el gráfico. Esta actividad puede ser diseñada con un mayor nivel de complejidad si se disminuye el número de fichas disponibles de cada color y se introducen los conceptos de condicionales y repetitivas (Conde et al., 2017).

A continuación, se describen otras actividades desconectadas que han sido aplicadas exitosamente en España con niños de edades comprendidas entre los 10 y 12 años por Brackmann et al. (2017):

- **Descomposición:** Se presenta un problema a los estudiantes y ellos tienen que descomponerlo en los pasos necesarios para resolverlo. Por ejemplo: lavarse las manos, preparar el desayuno, plantar un árbol, etc. Con esta actividad se trabaja el concepto de diseño de algoritmos y la habilidad de descomposición.
- **El mapa de Mónica:** Se presenta un mapa en una hoja dividida en cuadros donde se encuentran varios personajes. Los estudiantes tienen que encontrar la ruta más corta entre ellos usando solamente recorridos de un paso hacia abajo, arriba, derecha e izquierda. En un siguiente nivel, los estudiantes pueden usar multiplicadores y avanzar un número determinado de pasos en una dirección mediante una sola instrucción. Con esta actividad se trabajan habilidades de reconocimiento de patrones y diseño de algoritmos.
- **Elefantes:** Se usa una canción popular como ejemplo de la transformación de una canción en un algoritmo. Se presentan varias estrofas de la canción de los elefantes que se balancean en una tela de araña. Además, se plantea una variable X que se actualiza con el número de elefantes que se balancean en cada momento de la canción. La repetición de estrofas se simboliza con una estructura repetitiva y a medida que se avanza en la canción, los estudiantes van actualizando el valor de la variable hasta llegar a la última estrofa. Mediante esta actividad se trabaja la abstracción, el reconocimiento de patrones y el diseño de algoritmos.

- **Tetris:** Es una actividad para trabajar en parejas con dibujos de algunas de las piezas del juego de Tetris. Uno de los estudiantes puede ver el dibujo de las piezas y el otro estudiante tiene una hoja cuadriculada pero no puede ver las imágenes de las piezas. El segundo estudiante debe dibujar en su hoja cada una de las piezas siguiendo solamente las indicaciones de su compañero. Estas indicaciones se limitan a “iniciar”, “arriba”, “abajo”, “izquierda”, “derecha” y “parar”. Esta actividad permite trabajar las habilidades de reconocimiento de patrones y diseño de algoritmos.

Angeli et al. (2016) proponen un marco de referencia que puede ser usado en las escuelas para definir actividades que fomenten el desarrollo del pensamiento computacional de niños con edades entre los 2 y 12 años. A continuación, se describen las actividades macro que pueden ser desarrolladas con niños de 11 y 12 años y las habilidades que permiten desarrollar:

- **Abstracción:** Creación de modelos o representaciones para resolver un problema.
- **Generalización:** Mezclar y reusar recursos que fueron creados en actividades anteriores.
- **Descomposición:** Dividir una tarea compleja en tareas más simples. También se puede desarrollar una solución integrando pequeñas partes.
- **Pensamiento algorítmico:** Pueden desarrollarse las siguientes actividades: i) Definir los pasos para encontrar una solución, ii) organizar instrucciones en una secuencia correcta, iii) repetir una secuencia varias veces, iv) tomar decisiones basándose en condiciones, v) almacenar, recuperar y actualizar variables, y vi) formular expresiones lógicas y matemáticas.

- **Depuración:** Reconocer cuando las instrucciones no concuerdan con las acciones.
También se pueden detectar y eliminar errores en una secuencia de instrucciones.

Actividades conectadas

Bocconi et al. (2016) sostienen que la introducción de los conceptos y habilidades relacionadas con el pensamiento computacional requieren de herramientas de aprendizaje que permitan a los estudiantes desarrollar actividades de programación desde temprana edad. En este sentido, ambientes de programación basados en bloques permiten introducir los conceptos de programación sin tratar con aspectos relacionados con la sintaxis de los lenguajes de programación. Como ejemplos populares de estos ambientes se tienen: Scratch, Alice, App Inventor, Snap! y Blockly (Grover & Basu, 2017).

Weintrop & Wilensky (2015) definen los ambientes de programación basados en bloques como: “una clase de lenguaje de programación visual que aprovechan la metáfora de las piezas de un rompecabezas” (p.200). En estos ambientes, los estudiantes pueden ensamblar programas a partir de bloques predefinidos que representan los conceptos básicos de la programación usando sólo el ratón del computador. Cuando dos bloques no pueden ensamblarse para formar una sentencia válida en el programa, el ambiente previene el error. Además, se usan colores para identificar tipos de bloques y el alcance de las instrucciones (Weintrop & Wilensky, 2017). En la Figura 3 se puede observar un fragmento del popular ambiente de programación basado en bloques Scratch.

Los ambientes como Scratch y Alice permiten a los estudiantes el desarrollo de actividades de aprendizaje de la programación a partir de la creación de animaciones y

videojuegos. Asimismo, existen ambientes de este tipo que permiten desarrollar aplicaciones móviles como App Inventor y Pocket Code (Weintrop & Wilensky ,2015).

Scratch es un ambiente y lenguaje de programación desarrollado especialmente para niños, pero puede ser usado por personas de todas las edades. El diseño de Scratch fue pensado para facilitar su uso independientemente de la edad y del área de aplicación. Además, permite a los usuarios crear sus proyectos, los cuales pueden ser historias interactivas, juegos, animaciones, simulaciones, entre otros. Los proyectos de Scratch se componen de actores que se comportan según las instrucciones que el usuario indica mediante los bloques de programación (Marcelino et al., 2018).



Figura 3. Ejemplo del ambiente de programación basado en bloques Scratch.

Algunos ambientes de programación basados en bloques se han convertido en comunidades online que animan a sus usuarios a ver y experimentar con los códigos que publican los miembros. Adicionalmente, los estudiantes pueden seguir tutoriales de programación online que usan la programación por bloques como “La hora del código” en Code.org (Ver Figura 4) (Dwyer et al., 2015).

Code.org es una plataforma de programación en bloques lanzada en 2013 y ofrece juegos, tutoriales y videos donde personalidades importantes del mundo inspiran a los estudiantes a aprender acerca de las ciencias de la computación. Siguiendo tutoriales basados en juegos, los estudiantes aprenden los conceptos básicos de la programación como los condicionales, variables, repeticiones y funciones. Los cursos que proporciona la plataforma pueden ser tomados por estudiantes desde los cuatro años. Una característica importante de code.org es que incluye actividades conectadas y actividades desconectadas para aprender los conceptos de las ciencias de la computación (Kalelioğlu, 2015).

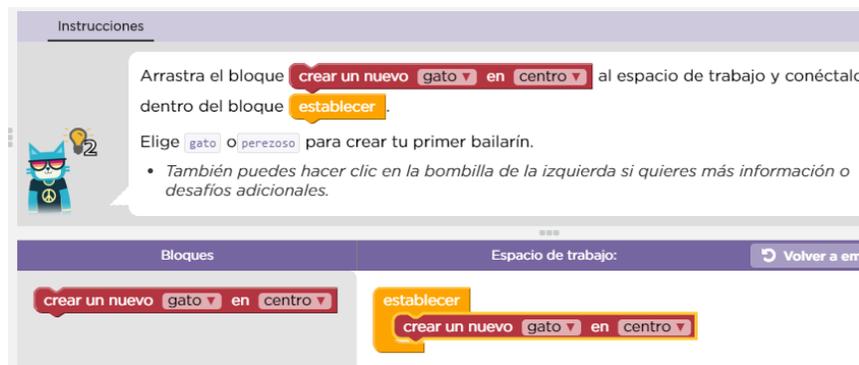


Figura 4. Ejemplo del tutorial “La hora del código” en Code.org.

Existen múltiples iniciativas para la enseñanza de la programación en las instituciones educativas y hay una tendencia creciente de creación de herramientas de libre acceso para promover el aprendizaje de la programación y el desarrollo de las habilidades del pensamiento computacional. Sin embargo, se debe trabajar en el desarrollo de entornos de aprendizaje mucho más potentes, crear metodologías de aprendizaje del pensamiento computacional, y evaluar el verdadero impacto del desarrollo del pensamiento computacional en la capacidad de resolver problemas en cualquier campo (Coronel-Díaz y Lima-Silvain, 2020).

Conclusiones y recomendaciones

El pensamiento computacional no debe entenderse como el aprendizaje de lenguajes de programación de computadores, sino como un conjunto a habilidades que todas las personas deben desarrollar para resolver problemas. Aunque varios autores han aportado acerca de la definición del pensamiento computacional, aún no existe consenso en cuanto a las habilidades que involucra y sobre las estrategias para fomentarlas y evaluarlas.

Para la enseñanza de los conceptos y habilidades asociadas con el pensamiento computacional se debe recurrir tanto a recursos didácticos conectados, los cuales están mediados por dispositivos tecnológicos, como a recursos didácticos desconectados, los cuales no requieren de este tipo de dispositivos, sino que incluyen actividades físicas y mentales que fomentan el desarrollo de habilidades asociadas al pensamiento computacional.

El uso de recursos didácticos desconectados para fomentar el desarrollo del pensamiento computacional se enfoca en motivar en los estudiantes la resolución de problemas a partir de conceptos y actividades propias de la programación de computadores. Este tipo de actividades tienen la ventaja de ser atractivas para los estudiantes debido a que en su mayoría incluyen actividades físicas que son desarrolladas en equipo.

Los ambientes de programación basados en bloques permiten la introducción de conceptos y habilidades relacionadas con la programación de computadores sin tener que profundizar en la sintaxis de los lenguajes. Estos ambientes son visuales, atractivos e intuitivos para los estudiantes debido a que utilizan la metáfora de las piezas de un rompecabezas para crear programas sólo arrastrando bloques que representan conceptos de la computación como las condiciones, las iteraciones, la entrada de datos, las variables, entre otros. El ambiente de este tipo más usado actualmente es Scratch.

Como trabajo futuro se recomienda realizar investigaciones sobre aspectos que han sido poco estudiados hasta la fecha acerca del pensamiento computacional como su evaluación y la formación y actualización docente requerida para abordar su enseñanza. Igualmente, se propone la creación de un repositorio público de recursos didácticos desconectados que puedan ser usados por docentes de educación básica secundaria y media.

Referencias

- Adell, J. S., Llopis, M. A. N., Esteve, M. F. M., y Valdeolivas, N. M. G. (2019). El debate sobre el pensamiento computacional en educación. *RIED. Revista Iberoamericana de Educación a Distancia*, 22(1), pp. 171-186. doi: 10.5944/ried.22.1.22303
- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. *Education Technology and Society*, 19, 47-57. Recuperado de <https://hdl.handle.net/11245/1.547418>
- Ávila Baray, H.L. (2006). *Introducción a la metodología de la investigación*. Edición electrónica. España. Recuperado de <http://www.univermedios.com/wp-content/uploads/2018/08/introduccion-a-la-metodologia-de-la-investigacion.pdf>
- Balladares-Burgos, J. A., Avilés-Salvador, M. R., y Pérez-Narváez, H. O. (2016). Del pensamiento complejo al pensamiento computacional: retos para la educación contemporánea. *Sophía*, 2(21), 143. doi: <https://doi.org/10.17163/soph.n21.2016.06>
- Bers, M. U., González-González, C., & Armas-Torres, B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*. doi:10.1016/j.compedu.2019.04.013

- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education implications for policy and practice*. Sevilla: Joint Research Centre. doi: 10.2791/792158
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of Computational Thinking Skills through Unplugged Activities in Primary School. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, 65–72. doi:10.1145/3137065.3137069
- Compañ-Rosique, P., Satorre-Cuerda, R., Llorens-Largo, F., y Molina-Carmona, R. (2015). Enseñando a programar: un camino directo para desarrollar el pensamiento computacional. *Revista de Educación a Distancia (RED)*, (46). doi.org/10.6018/red/46/11
- Conde, M. Á., Fernández-Llamas, C., Rodríguez-Sedano, F. J., Guerrero-Higueras, Á. M., Matellán-Olivera, V., & García-Peñalvo, F. J. (2017). Promoting Computational Thinking in K-12 Students by Applying Unplugged Methods and Robotics. *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality*. doi: 10.1145/3144826.3145355
- Coronel-Díaz, E., y Lima-Silvain, G. (2020). El pensamiento computacional. Nuevos retos para la educación del siglo XXI Computational Thinking. New challenges for 21st century education. 20, 115-137. Recuperado de <https://revistas.unc.edu.ar/index.php/vesc/article/view/27451/29019>

CSTA & ISTE (2011). *Caja de herramientas para el pensamiento computacional*. Recuperado de <http://www.eduteka.org/pdfdir/PensamientoComputacional1.pdf>

Dwyer, H., Hill, C., Hansen, A., Iveland, A., Franklin, D., & Harlow, D. (2015). Fourth Grade Students Reading Block-Based Programs: Predictions, Visual Cues, and Affordances. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, 111–119. doi:10.1145/2787622.2787729

Fernández, L. A. G. (2010). *Recursos didácticos: Elementos indispensables para facilitar el aprendizaje*. México: Limusa.

Grover, S., & Basu, S. (2017). Measuring Student Learning in Introductory Block-Based Programming: Examining Misconceptions of Loops, Variables, and Boolean Logic. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 267–272. doi:10.1145/3017680.3017723

Hernández-Sampieri, R., Fernández-Collado, C. y Baptista-Lucio, P., (2014). *Metodología de la Investigación*, 6ta edición, México: McGraw-Hill.

Hoyos, C. (2000). Un modelo para una investigación documental. Guía teórico- práctica sobre construcción de estados del arte. Medellín: Señal Editora.

- ISTE. (2016). *ISTE Standards for Students*. Washington DC: International Society for Technology in Education. Recuperado de <https://www.iste.org/standards/for-students>
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200–210. doi:10.1016/j.chb.2015.05.047
- Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., & Mendes, A. J. (2018). Learning Computational Thinking and scratch at distance. *Computers in Human Behavior*, 80, 470–477. doi:10.1016/j.chb.2017.09.025
- Roig-Vila, R., y Moreno-Isac, V. (2020). El pensamiento computacional en Educación. Análisis bibliométrico y temático. *Revista de Educación a Distancia*, 20(63). doi: 10.6018/red.402621
- Scott, C.L. 2015. *El futuro del aprendizaje 2 ¿Qué tipo de aprendizaje se necesita en el siglo XXI?* Investigación y Prospectiva en Educación UNESCO, París. [Documentos de Trabajo ERF, No. 14]. Recuperado de https://unesdoc.unesco.org/ark:/48223/pf0000242996_spa
- Valverde-Berrocoso, J., Fernández-Sánchez, M. R., y Garrido-Arroyo, M. C. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia*, (46). Recuperado de <https://revistas.um.es/red/article/view/240311>

Weinberg, A. (2013). *Computational Thinking: An Investigation of the existing scholarship and research (tesis doctoral)*. Universidad estatal de Colorado. Fort Collins, Colorado.

Recuperado de <https://mountainscholar.org/handle/10217/78883>

Weintrop, D., & Wilensky, U. (2015). To Block or Not to Block, That is the Question: Students' Perceptions of Blocks-Based Programming. *Proceedings of the 14th International Conference on Interaction Design and Children*, 199–208. doi: 10.1145/2771839.2771860

Weintrop, D., & Wilensky, U. (2017). Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms. *ACM Trans. Comput. Educ.*, 18(1). <https://doi.org/10.1145/3089799>

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. doi: 1118178.1118215

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725. doi: 10.1098/rsta.2008.0118

Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. RED. *Revista de Educación a Distancia*, (46). Recuperado de <https://www.um.es/ead/red/46/>