

DOI: <http://doi.org/10.15517/revedu.v47i2.53645>

## Pensamiento computacional y programación en la formación de estudiantes desde edades tempranas

*Computational Thinking and Programming in the Training of Students from Early Age*

Efraín Alfredo Barragán Perea  
Universidad Autónoma de Chihuahua  
Chihuahua,  
México  
[ebarragan@uach.mx](mailto:ebarragan@uach.mx)  
<https://orcid.org/0000-0002-9045-5425>

Recepción: 27 de enero 2023  
Aprobación: 28 de febrero 2023

### ¿Cómo citar este artículo?

Barragán-Perea, E. A. (2023). Pensamiento computacional y programación en la formación de estudiantes desde edades tempranas. *Revista Educación*, 47(2). <http://doi.org/10.15517/revedu.v47i2.53645>



**RESUMEN**

El uso de las tecnologías de la información y la comunicación son el camino idóneo para el desarrollo de las competencias que el alumnado necesita para desenvolverse en una sociedad ampliamente digitalizada, con el fin de dejar de ser solo personas que consumen tecnología. De esta manera, uno de los principales retos de la educación actual implica que la población estudiantil aprenda a resolver problemas a través del uso de las tecnologías y la creación de software, para lo cual es necesario el desarrollo de competencias relacionadas con el pensamiento computacional y la programación como parte de los nuevos procesos de aprendizaje. El objetivo de la investigación fue analizar el impacto del pensamiento computacional y la programación en la formación de estudiantes desde edades tempranas, por lo cual se profundizó en los conceptos de pensamiento computacional, lenguajes de programación, competencias de programación, alfabetización digital y aprendizaje de la programación desde edades tempranas. Metodológicamente, se realizó una revisión bibliográfica de la literatura científica acerca del tema, mediante la aplicación de un estudio descriptivo. Se encontró que el cambio en el proceso de enseñanza-aprendizaje tradicional demanda la incorporación del pensamiento computacional y la programación como parte de los nuevos procesos de aprendizaje, además, que el mundo de los lenguajes de programación se ha adaptado a las necesidades de las personas, permitiendo desarrollar habilidades relacionadas con la resolución de problemas, la creatividad, el pensamiento lógico, el razonamiento y el trabajo colaborativo. No obstante, existen razones que dificultan dicho proceso, como son la discapacidad, la inversión necesaria en tiempo y dinero, el escaso manejo de la frustración, así como cuestiones de inclusión y género. Se recomienda realizar nuevas investigaciones que permitan conocer si los cuerpos docentes en los países latinoamericanos cuentan con las competencias necesarias para educar al alumnado en la programación, a fin de transitar hacia el pensamiento computacional.

**PALABRAS CLAVE:** Pensamiento computacional, Lenguajes de programación, Competencias de programación, Alfabetización digital.

**ABSTRACT**

Information and communication technologies represent the ideal path for developing the necessary skills for students to function in a widely digitized society and to stop being simple consumers of technology. In this sense, one of the main challenges of current education implies that students learn to solve problems through the use of technologies and the creation of software, which implies the development of skills related to computational thinking and programming as part of the

new learning processes. Thus, the objective of the research was to understand the impact of computational thinking and programming in the training of students. Therefore, the researcher focused on the concepts of computational thinking, programming languages, programming skills, digital literacy, and programming learning from an early age. Regarding the methodology, through the application of a descriptive study, the researcher carried out a bibliographic review of the scientific literature on the subject. Hence, the investigator concluded that the traditional teaching-learning process demands the incorporation of computational thinking and programming as new learning processes. In addition, programming languages have adapted to the needs of people, which has allowed the development of skills related to problem-solving, creativity, logical thinking, reasoning, and collaborative work. However, certain elements hinder this process; for instance, disability, necessary investment in time and money, and poor management of frustration, as well as inclusion and gender topics. Therefore, the author recommends carrying out new investigations that allow knowing if the professors in Latin American countries have the required skills to educate students in programming and thus move towards computational thinking.

**KEYWORDS:** Computational Thinking, Programming Languages, Programming Skills, Digital Literacy.

## INTRODUCCIÓN

En los últimos años, las instituciones educativas han resaltado la importancia sobre el uso de las tecnologías de la información y la comunicación (TIC) como el camino idóneo para el desarrollo de las competencias que el alumnado de todas las edades necesita para desenvolverse en la sociedad actual (Polanco et al., 2021; Almiron y Porro, 2014). Lo anterior plantea la necesidad de que el estudiantado desarrolle nuevas capacidades a través de la alfabetización digital, lo que les permitirá desenvolverse en una sociedad ampliamente digitalizada (Reddy et al., 2020).

Al respecto, la Real Academia de la Lengua Española define la palabra alfabetizar como la capacidad para enseñar a leer y escribir a las personas (Real Academia Española, s.f.), sin embargo, en una sociedad en la que predominan las TIC esta definición ha quedado obsoleta. En este sentido, es necesario hablar de alfabetización digital, un proceso primordial para adquirir las habilidades necesarias para ser competente en el uso de las nuevas tecnologías (George y Avello, 2021).

Como parte de este nuevo escenario donde prácticamente todas las profesiones están relacionadas en mayor o menor medida con aspectos de informática, la realidad apunta a que gran

parte de la población estudiantil y docente desconoce los procesos de creación que existen detrás de las aplicaciones de escritorio, aplicaciones web, videojuegos, sistemas operativos y demás software que utilizan en su vida cotidiana, desconocimiento que les convierte en simples consumidores de tecnología (Motoa, 2019).

Así, aprender a resolver problemas a través del uso de las tecnologías y la creación de software a través de la programación se convierte en uno de los principales retos de la educación actual. Esto implica el desarrollo de competencias relacionadas con la programación, consideradas hoy en día destrezas básicas e instrumentales en la sociedad del conocimiento (Tejera-Martínez et al., 2020; Espino y González, 2015), afirmación que compartía Steve Jobs (cofundador de Apple), cuando mencionaba que todo el mundo debería aprender un lenguaje de programación, ya que esto te enseña a pensar.

Lo anterior conlleva un cambio en el proceso de enseñanza-aprendizaje tradicional, el cual requiere adaptarse para responder a las competencias demandadas, mediante la incorporación del pensamiento computacional y la programación como parte de los nuevos procesos de aprendizaje. Estos son impulsados por la web 2.0, los dispositivos móviles y la industria de los videojuegos, que han puesto de manifiesto la necesidad de formar al alumnado en el ámbito tecnológico para el desarrollo de dichas competencias (Valverde et al., 2015). Por tanto, el objetivo de esta investigación fue analizar el impacto del pensamiento computacional y la programación en la formación de estudiantes desde edades tempranas.

## ESTRATEGIA METODOLÓGICA

Para el logro de esta contribución se realizó una investigación de tipo documental de la literatura científica acerca del tema, mediante la aplicación de un estudio descriptivo.

Se incluyeron los artículos originales más importantes y significativos de acuerdo al criterio del autor, publicados en bases de datos científicas (Scielo, RedAlyC, Dialnet, ScienceDirect, WoS y Scopus), entre los años 2015 y 2022, en idioma español e inglés, que describieran el impacto del pensamiento computacional y la programación como parte de las competencias necesarias para la formación de personas estudiantes. Para lo anterior, se abordaron los temas de: pensamiento computacional, lenguajes de programación, competencias de programación, alfabetización digital, el aprendizaje de la programación desde edades tempranas, así como diversas posturas acerca del aprendizaje de la programación.

## RESULTADOS Y DISCUSIÓN

### Pensamiento computacional

El concepto de pensamiento computacional implica “resolver problemas, diseñar sistemas y comprender el comportamiento humano haciendo uso de los conceptos fundamentales de la informática” (Wing, 2006, p. 33). Se trata de una forma de pensar que no es exclusiva de profesionales de la programación de sistemas o de personas expertas en computación, sino un conjunto de habilidades útiles para cualquier individuo o profesión (Polanco et al., 2021).

Sánchez-Vera y González-Martínez (2019) y Basogain et al. (2015) definieron el pensamiento computacional como una metodología basada en la implementación de conceptos básicos de las ciencias de la computación para resolver problemas de la vida diaria, diseñar sistemas domésticos y realizar tareas rutinarias; por lo tanto, es transferible a multitud de contextos personales y necesario para la ciudadanía.

El pensamiento computacional está centrado, en gran parte, en las habilidades que los niños y niñas desarrollan a partir de la práctica de la programación y los algoritmos, lo cual permite el desarrollo de cualidades como el pensamiento abstracto, la resolución de problemas, el reconocimiento de patrones y el razonamiento lógico (Angeli y Giannakos, 2020).

A decir de Wing (2006), el pensamiento computacional debería ser incluido como una nueva competencia en la formación de estudiantes, ya que, al igual que las matemáticas, es una habilidad fundamental cuyo progreso en la asimilación de un concepto se basa en la comprensión del anterior y se puede desarrollar desde la niñez (Maris, 2018).

Para De Souza (2019), el pensamiento computacional está basado en cuatro ejes:

1. La descomposición de problemas complejos en problemas más pequeños para un mejor manejo.
2. La búsqueda de semejanzas en dichos problemas con el fin de ayudar a resolver el sistema de forma más eficiente.
3. Centrarse en la información importante, es decir, en las características generales que son comunes a cada elemento para crear un modelo del problema que se busca resolver.
4. El diseño de algoritmos (a través de diferentes pasos) para resolver el problema.

De esta manera, a través del pensamiento computacional, es posible mejorar las competencias laborales en todos los sectores, pues ayuda al personal de una empresa a asumir un papel activo en la búsqueda de soluciones a diferentes tipos de problemas con la ayuda del pensamiento algorítmico, mediante técnicas de descomposición y construcción de secuencias de pasos.

### **Lenguajes de programación**

La programación se refiere a la acción de crear programas o aplicaciones a través del desarrollo de un código fuente (escrito en algún lenguaje de programación), que se basa en el conjunto de instrucciones que sigue la computadora para ejecutar un programa (Sáenz, 2019).

Un lenguaje de programación es un idioma artificial prediseñado formado por signos, palabras y símbolos que le permite a una persona (programador) escribir instrucciones con las que será posible comunicarse con el hardware (computadora) para la realización de un determinado proceso (Veeraraghavan, 2022). Las instrucciones que sigue la computadora para la ejecución de programas y aplicaciones están escritas en algún lenguaje de programación, luego estas son traducidas a un lenguaje de máquina, el cual es interpretado y ejecutado por el hardware del equipo (Mathieu, 2014).

El código fuente consiste en líneas de texto que expresan en algún lenguaje de programación las instrucciones que debe llevar a cabo una computadora (De Miguel et al., 2021). Dicho código es sujeto de creación, diseño, codificación, mantenimiento y depuración a través de la programación misma.

Actualmente, existen tres tipos de programación:

1. Programación estructurada, orientada a mejorar la claridad, calidad y tiempo de desarrollo, utilizando subrutinas o funciones (López, 2004).
2. Programación modular, donde los programas se dividen en módulos para trabajar con estos y resolver los problemas de manera más simple (Vidal et al., 2019).
3. Programación orientada a objetos, donde se utilizan entidades llamadas objetos (con características, estado y comportamiento) como elementos fundamentales para la búsqueda de soluciones (López, 2018).

El mundo de los lenguajes de programación ha ido adaptándose a las necesidades de las personas interesadas, por lo que también ha estado en constante evolución. En la actualidad, los principales lenguajes de programación son:

- Python. Uno de los lenguajes más sencillos de código abierto utilizado por jóvenes para iniciarse en la programación, de hecho, debería ser el primer lenguaje en ser aprendido por las personas, sin importar las características iniciales del estudiantado (Mendoza, 2017). Sus principales usos son en inteligencia artificial, *big data*, *blockchain* y desarrollo web (Nagpal y Gabrani, 2019).
- Lenguaje C. Un lenguaje de programación de propósito general orientado a la implementación de sistemas operativos (Nengsih et al., 2022).
- JavaScript. Lenguaje orientado a objetos que no necesita compilación. Es ideal para crear sitios web y videojuegos por su gran variedad de elementos visuales (Luna, 2019).
- PHP. Lenguaje que posee una excelente capacidad de comunicación entre el servidor y la web, permitiendo el desarrollo de sitios web muy estables y con gran rendimiento. Este lenguaje ha permitido crear plataformas como Facebook, Wikipedia y Gmail (Bankov, 2019).

Estos lenguajes son los más populares y prometedores para iniciarse en el mundo de la programación y, aunque la curva de aprendizaje de cada uno puede variar dependiendo de la edad y experiencia de cada persona, es posible ver los primeros resultados de manera rápida.

### Competencias clave

Las competencias clave se definen como una combinación de conocimientos, capacidades y actitudes (Rodríguez et al., 2020). Los conocimientos se componen de hechos, conceptos y teorías. Las capacidades son la habilidad para aplicar los conocimientos y obtener resultados, mientras que las actitudes son la disposición de la persona para actuar ante ciertas situaciones (Consejo de la Unión Europea, 2018).

Desde el punto de vista de los planes de estudios, el aprendizaje por competencias implica romper con la estructura tradicional de los campos de estudio, apostando por un enfoque multidisciplinario en el proceso educativo.

El Consejo de la Unión Europea (2018) destaca ocho competencias clave (Tabla 1), las cuales son necesarias para la realización y desarrollo de las personas, así como para la ciudadanía activa, la inclusión social y el empleo (Diego-Mantecón et al., 2021).

TABLA 1.

#### Competencias clave recomendadas por el Consejo de la Unión Europea

Competencia	Capacidad
-------------	-----------

1. Lectoescritura	Comprender, expresar e interpretar información, tanto de forma oral como escrita.
2. Multilingüe	Manejar una o más lenguas extranjeras.
3. Matemática, ciencia, tecnología e ingeniería	Emplear las matemáticas para resolver problemas de la vida real, así como desarrollar la capacidad para explicar el mundo que le rodea, plantear preguntas y sacar conclusiones.
4. Digital	Habilidad para utilizar las tecnologías de forma segura y responsable.
5. Personal y social	Administrar el tiempo y el aprendizaje de forma autónoma y colaborativa
6. Ciudadana	Participar plenamente en la vida social y ciudadana.
7. Emprendedora	Habilidad para transformar oportunidades e ideas en hechos.
8. Conciencia y expresiones culturales	Capacidad para comprender y respetar diferentes contextos culturales y artísticos.

Fuente: Diego-Mantecón et al. (2021); Consejo de la Unión Europea (2018).

Por otra parte, sobre la base del proyecto de Definición y Selección de Competencias clave (DeSeCo) (Organización para la Cooperación y el Desarrollo Económicos [OCDE], 2003) y Leicht et al. (2018), se han podido identificar un conjunto de competencias, divididas en tres categorías (Tabla 2), que involucran el desarrollo de habilidades y capacidades que sirven de base para la adaptación ciudadana a un mundo complejo y en constante cambio. Dichas competencias deben fomentarse de la mano del pensamiento computacional y la programación.

El proyecto DeSeCo es considerado como el inicio de un movimiento que ha servido de soporte e inspiración para un enfoque renovado de los currículos escolares (López-Rupérez, 2022).

**TABLA 2.**

**Competencias clave de la OCDE sacadas del proyecto DeSeCo**

Competencia clave	Capacidad
<b>Categoría 1: Usar las herramientas de forma interactiva</b>	
1. Uso interactivo del lenguaje, los símbolos y los textos	Usar efectivamente las destrezas lingüísticas orales y escritas, las habilidades de computación y otras capacidades matemáticas.
2. Uso interactivo del conocimiento y la Información	Reconocer lo que no se sabe, identificar y acceder a fuentes de información apropiadas, evaluar la calidad, propiedad y valor de dicha información y organizar dicho conocimiento.
3. Uso interactivo de la tecnología	Relacionar las posibilidades que yacen en las herramientas tecnológicas con sus propias circunstancias y metas e incorporar la tecnología en prácticas comunes.

---

**Categoría 2: Interactuar en grupos heterogéneos**

---

- |   |   |
|---|---|
| 1. Relacionarse de manera adecuada con otras personas | Desarrollar la inteligencia emocional, respetar valores, creencias, culturas e historias de otros, desarrollar la empatía y autorreflexión y lograr el conocimiento sobre uno mismo y los demás.        |
| 2. Cooperar y trabajar en equipo                      | Presentar ideas y escuchar las de otras personas, entender las dinámicas del debate, construir alianzas y la capacidad para negociar y tomar decisiones.  |
| 3. Manejar y resolver conflictos                      | Analizar los intereses en juego, los orígenes del conflicto y el razonamiento de todas las partes, identificar áreas de acuerdo y desacuerdo, recontextualizar el problema y priorizar las necesidades. |

---

**Categoría 3: Actuar de forma autónoma**

---

- |   |   |
|---|---|
| 1. Actuar dentro del contexto del gran panorama                   | Comprender patrones, conocer el sistema en el que existe (estructuras, cultura, prácticas, reglas, roles y normas sociales), identificar consecuencias de sus acciones y reflexionar sobre las mismas.              |
| 2. Formar y conducir planes de vida y proyectos personales        | Definir proyectos y metas, identificar y evaluar recursos, priorizar metas, aprender de acciones pasadas y monitorear el progreso haciendo los ajustes necesarios.  |
| 3. Defender y asegurar derechos, intereses, límites y necesidades | Comprender intereses propios y colectivos, conocer las reglas y principios para basar un caso, argumentar para que los derechos y necesidades propios sean reconocidos, sugerir arreglos o soluciones alternativas. |

---

Fuente: OCDE (2003); Leicht et al. (2018).

Como se puede apreciar, las competencias desarrolladas por distintas instituciones internacionales como la OCDE y el Consejo de la Unión Europea, con el fin de alcanzar un mejor desarrollo del estudiantado para el futuro laboral, tienen lugares comunes, así como ligeras variaciones. Sin embargo, los focos están centrados, en gran medida, en los dominios cognitivo, intrapersonal e interpersonal.

Las autoridades educativas deben sumar esfuerzos junto con la sociedad para incluir el pensamiento computacional en los planes educativos como una habilidad imprescindible en la formación del estudiantado, la cual debe iniciar en la educación básica y continuar hasta la formación superior.

No obstante, el pensamiento computacional no debería limitarse a la inclusión de alguna asignatura como parte de alguna competencia dentro de los planes de estudio, sino verse como un eje

transversal que coadyuve a la construcción de una mejor sociedad, permitiendo la resolución de problemas sobre la base de la comprensión y el razonamiento.

### **El aprendizaje de la programación desde edades tempranas**

Según datos de la Encuesta Nacional sobre Disponibilidad y Uso de Tecnologías de la Información en los Hogares (ENDUTIH) (Instituto Nacional de Estadística y Geografía [INEGI], 2021), en México el 44.8 % de los hogares cuenta con al menos una computadora, es decir, 43 844 751 usuarios, de los cuales 10.3 % tiene entre 6 y 11 años de edad, 16 % entre 12 y 17 años y 19.3 % entre 18 y 24. De lo anterior, se desprende que la alfabetización digital desde edades tempranas se establece no solo como una posibilidad, sino como una prioridad en un mundo en el que la tecnología está cambiando rápidamente el panorama educativo y laboral.

Aprender a programar ayuda a los jóvenes de todas las edades a desarrollar habilidades de resolución de problemas (Gökçe y Yenmez, 2022), promueve la creatividad (Noh y Lee, 2020), el pensamiento lógico (Cerón, 2022), el razonamiento (Macías et al., 2021) y el trabajo colaborativo (Polanco et al., 2021). Al respecto, Tim Cook (CEO de Apple) considera que aprender a programar es más importante, incluso, que aprender el idioma inglés (Schneidermann, 2017).

También, es importante resaltar que, en la mayoría de los países, las etapas educativas en las que los lenguajes de programación están más presentes son la universitaria, seguida de la educación secundaria, quedando la educación primaria e infantil menos representadas; aunque se observa un creciente interés (Tejera-Martínez et al., 2020).

Los países líderes en tecnología han hecho grandes esfuerzos en la materia, incluyendo en sus currículos el pensamiento computacional (Motoa, 2019). Así, Estados Unidos, China, Gran Bretaña y España han incorporado la enseñanza de la programación desde la educación básica, con el fin de convertirse en potencias de innovación (Mackey, 2016). Lo anterior se debe a que se estima que la población infantil que hoy inicia su vida escolar ejercerá en un futuro profesiones que aún no existen, utilizará herramientas que todavía no se han desarrollado y logrará dar respuesta a problemas que aún no están planteados (García, 2019).

Los esfuerzos antes mencionados deberían ser emulados por otros países, especialmente los latinoamericanos. No obstante, el escenario planteado demanda el compromiso y disposición del cuerpo docente para familiarizarse con las diversas herramientas y lenguajes disponibles, que permiten el desarrollo del pensamiento computacional, lo cual supone un reto adicional.

En la actualidad, existen diversas plataformas de programación con diferentes grados de dificultad, adaptadas a las diferentes etapas educativas de la infancia con el fin de iniciarles en la programación. Algunos ejemplos de ellos son:

- Scratch. Es el lenguaje de programación más utilizado en todas las etapas del ámbito educativo debido a su fácil adaptación a las características específicas de cada situación docente. Se compone de un sencillo sistema de bloques con el cual es posible crear historias interactivas, juegos y animaciones (Fagerlund et al., 2021; Tejera-Martínez et al., 2020).
- Code.org. Un sitio web que tiene como objetivo alentar al estudiantado a aprender informática. El sitio incluye lecciones de codificación gratuitas, sonidos y otros elementos que ayudan a codificar con fluidez (Sáenz, 2019; Kalelioğlu, 2015).
- Sphero Edu. Una aplicación orientada a la robótica que permite a la población infantil programar sus robots Sphero desde una tableta o teléfono. Además, permite aprender nociones básicas que se aplican en la mayoría de los lenguajes de programación de una forma lúdica (Berberian, 2019).
- Alice. Un lenguaje de programación educativo basado en objetos con un entorno de desarrollo integrado. Este dispone de un entorno que consiste en arrastrar y soltar para crear animaciones por computadora usando modelos 3D (Costa y Miranda, 2019).
- Arduino. Una plataforma de creación de electrónica (creación de robots) de código abierto, flexible y fácil de utilizar, que combina la programación con la electrónica (DesPortes y DiSalvo, 2019).

### **Posturas acerca del aprendizaje de la programación**

Cuando se analiza la posibilidad de que cualquier persona pueda aprender a programar, pueden surgir posiciones encontradas al respecto. Tal es el caso de Bañeres et al. (2018), quienes diseñaron un MOOC (curso online masivo abierto), con el objetivo de acercar la programación a personas sin ningún conocimiento previo en el tema, mediante el pensamiento computacional combinado con la metodología *learning-by-doing* (aprender haciendo) y el lenguaje de programación Scratch. Los resultados fueron muy satisfactorios y con un alto grado de motivación para seguir aprendiendo conceptos relacionados con la ciencia y la tecnología.

En otro estudio, Monjelat et al. (2018) evidenciaron que pese al gran potencial que representa la creación de proyectos de programación con Scratch, su utilización es poco accesible para personas

con discapacidad (visual, auditiva, física, del habla, cognitiva, de aprendizaje y neurológica), lo que representa un área de oportunidad en la evolución antes mencionada de los lenguajes de programación.

Por su parte, Linus Torvalds, creador del sistema operativo Linux, (Love, 2014), considera que la programación informática no es para todas las personas, por lo que expone algunas razones que dificultan su aprendizaje, como son: la falta de pasión para resolver problemas, la inversión necesaria en tiempo y dinero que puede representar la formación de personal especializado en programación, un escaso manejo de la frustración, o bien, la dificultad de la persona para pensar de forma abstracta.

Otras razones que desalientan el aprendizaje de la programación van en relación a cuestiones de inclusión y género, debido, por ejemplo, al alejamiento de las mujeres de la informática desde las primeras etapas de la infancia, fenómeno que se va intensificando a lo largo de la vida y que representa una amenaza a la igualdad de oportunidades (Torres-Torres et al., 2022; Yansen, 2020). En palabras de Sotaminga y Apolo (2021), esto podría atenderse con la participación activa tanto de los padres y madres desde los hogares, como del profesorado desde las instituciones educativas.

## CONCLUSIONES

Con base en la investigación documental realizada, a fin de describir el impacto del pensamiento computacional y la programación como parte de las competencias necesarias para la formación de personas estudiantes, se concluye lo siguiente:

- El empleo de las TIC es el camino propicio para el desarrollo de las competencias necesarias para que el alumnado se desenvuelva en una sociedad cada vez más digitalizada, con el fin de dejar de ser solo personas que consumen tecnología.
- Uno de los principales retos de la educación actual implica que la población estudiantil aprenda a resolver problemas a través del uso de las tecnologías y la creación de software, lo cual implica el desarrollo de competencias relacionadas con la programación.
- Las autoridades educativas y la sociedad deben sumar esfuerzos para incluir el pensamiento computacional y la programación en los planes educativos como una habilidad imprescindible en la formación del estudiantado, la cual debe iniciar en la educación básica y continuar hasta la formación superior.

Revista Educación, 2023, 47(2), julio-diciembre, ISSN: 0379-7082 / e-ISSN 2215-2644

- El mundo de los lenguajes de programación ha ido adaptándose a las necesidades de las personas, permitiendo desarrollar habilidades relacionadas con la resolución de problemas, la creatividad, el pensamiento lógico, el razonamiento y el trabajo colaborativo.
- Los países líderes en tecnología han hecho esfuerzos para la inclusión del pensamiento computacional en sus currículos, valiéndose de las diversas plataformas de programación existentes, las cuales son fácilmente adaptables a las diferentes etapas educativas, con el fin de iniciar a la población infantil y los grupos adolescentes en la programación. Dichos esfuerzos deberían ser emulados por otros países, especialmente los latinoamericanos.
- No obstante, existen razones que dificultan el aprendizaje de la programación, como son la discapacidad, la falta de pasión para resolver problemas, la inversión necesaria en tiempo y dinero, el escaso manejo de la frustración, la dificultad para pensar de forma abstracta, así como cuestiones de inclusión y género.

## RECOMENDACIONES

Con base en la investigación documental realizada, se recomienda lo siguiente:

- La realización de un estudio que permita conocer si los cuerpos docentes en los países latinoamericanos cuentan con las competencias necesarias para educar al alumnado en la programación, a fin de transitar hacia el pensamiento computacional.
- Un estudio de las universidades del mundo que han incorporado la enseñanza del pensamiento computacional en los planes de estudios para la formación de cuerpos docentes.
- Un estudio de género que muestre la participación de las mujeres en el mundo de la programación.

## REFERENCIAS

Almiron, M. E. y Porro, S. (2014). Las TIC en la enseñanza: un análisis de casos. *Revista electrónica de investigación educativa*, 16(2), 152-160. <https://redie.uabc.mx/redie/article/view/341/937>

- Angeli, C. y Giannakos, M. (2020). Computational thinking education: Issues and challenges [Educación del pensamiento computacional: problemas y desafíos]. *Computers in Human Behavior*, 105. <https://doi.org/10.1016/j.chb.2019.106185>
- Bankov, B. (2019). Software evaluation of PHP MVC web applications [Evaluación de software de aplicaciones web PHP MVC]. *International Multidisciplinary Scientific GeoConference: SGEM*, 19(2.1), 603-610. <https://bit.ly/3GTSvpG>
- Bañeres, D., Casado, C., Ornellas, A., Planas, E., Prieto, J. y Serra, M. (2018). ¿Realmente la programación puede ser para todos? Análisis de la experiencia dentro de un MOOC. *Actas de las Jenui*, 3, 263-270. [https://aenui.org/actas/pdf/JENUI\\_2018\\_017.pdf](https://aenui.org/actas/pdf/JENUI_2018_017.pdf)
- Basogain, X., Olabe, M. A. y Olabe, J. C. (2015). Pensamiento computacional a través de la programación: paradigma de aprendizaje. *Revista de Educación a Distancia*, 46(6). <https://doi.org/10.6018/red/46/6>
- Berberian, P. (2019). Sphero brings stem-based learning to the living room with the release of new sphero mini activity kit [Sphero lleva el aprendizaje basado en stem a la sala de estar con el lanzamiento del nuevo kit de actividades sphero mini]. *Access*, 33(4), 33. <https://search.informit.org/doi/10.3316/informit.836607480872847>
- Cerón, J. A. (2022). La programación para niños: perspectivas de abordaje desde el pensamiento lógico matemático. *Revista Internacional de Pedagogía e Innovación Educativa*, 2(1), 101-122. <https://doi.org/10.51660/ripie.v2i1.70>
- Consejo de la Unión Europea (Ed.). (2018). *Recomendación del consejo de 22 de mayo de 2018 relativa a las competencias clave para el aprendizaje permanente*. <https://bit.ly/3FvYPU2>
- Costa, J. M. y Miranda, G. L. (2019). Using Alice Software with 4C-ID Model: Effects in Programming Knowledge and Logical Reasoning [Uso del software Alice con el modelo 4C-ID: efectos en el conocimiento de la programación y el razonamiento lógico]. *Informatics in Education*, 18(1), 1-15. <https://files.eric.ed.gov/fulltext/EJ1212940.pdf>
- De Miguel, G., Júlvez Bueno, J. y Gracia de Río, J. (2021). *Introducción a la Programación C++ para Ingenieros*. (Vol. 313). Prensas de la Universidad de Zaragoza.
- De Souza, I. (2019, 21 de septiembre). Descubre qué es el pensamiento computacional y sus beneficios desde la niñez hasta la profesión. *Rock Content*. <https://bit.ly/3GT6E6Z>
- DesPortes, K. y DiSalvo, B. (2019, 30 de julio). *Trials and tribulations of Novices working with the Arduino* [Pruebas y tribulaciones de los novatos trabajando con el Arduino] [Conferencia].

- ACM Conference on International Computing Education Research, Nueva York, Estados Unidos. <https://doi.org/10.1145/3291279.3339427>
- Diego-Mantecón, J. M., Fernández-Blanco, T., Ortiz-Laso, Z. y Lavicza, Z. (2021). Proyectos STEAM con formato KIKS para el desarrollo de competencias clave. *Comunicar: Revista científica iberoamericana de comunicación y educación*, (66), 33-43. <https://dialnet.unirioja.es/servlet/articulo?codigo=7696990>
- Espino E. y González C. (2015). Estudio sobre diferencias de género en las competencias y las estrategias educativas para el desarrollo del pensamiento computacional. *Revista de Educación a Distancia*, 46(12). <https://doi.org/10.6018/red/46/12>
- Fagerlund, J., Häkkinen, P., Vesisenaho, M. y Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review [Pensamiento computacional en programación con Scratch en escuelas primarias: una revisión sistemática]. *Computer Applications in Engineering Education*, 29(1), 12-28. <https://doi.org/10.1002/cae.22255>
- García, L. (2019). Necesidad de una educación digital en un mundo digital. *RIED. Revista Iberoamericana de Educación a Distancia*, 22(2), 09-22. <https://doi.org/10.5944/ried.22.2.23911>
- George, C. E. y Avello, R. (2021). Alfabetización digital en la educación. Revisión sistemática de la producción científica en Scopus. *Revista de Educación a Distancia*, 21(66), 1-21. <https://doi.org/10.6018/red.444751>
- Gökçe, S. y Yenmez, A. A. (2022). Ingenuity of scratch programming on reflective thinking towards problem solving and computational thinking [Ingenio de la programación desde cero en el pensamiento reflexivo hacia la resolución de problemas y el pensamiento computacional]. *Education and Information Technologies*, 21(9), 1-25. <https://doi.org/10.1007/s10639-022-11385-x>
- Instituto Nacional de Estadística y Geografía [INEGI]. (2021). *Encuesta Nacional sobre Disponibilidad de Tecnologías de Información en los hogares (ENDUTIH-2021)*. <https://www.inegi.org.mx/programas/dutih/2021/>
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. [Una nueva forma de enseñar habilidades de programación a estudiantes de K-12: Code.org]. *Computers in Human Behavior*, 52, 200-210. <https://doi.org/10.1016/j.chb.2015.05.047>

- Leicht, A., Heiss, J. y Byun, W. J. (Eds.) (2018). *Issues and trends in education for sustainable development* [Problemas y tendencias en la educación para el desarrollo sostenible]. (Vol. 5). UNESCO Publishing. <https://bit.ly/3k0ZMeU>
- López, D. A. (2018). *Programación orientada a objetos I*. Servicio de Publicaciones de la Universidad Católica de Ávila.
- López, L. (2004). *Programación estructurada. Un enfoque algorítmico (2a. Edición)*. AlfaOmega.
- López-Rupérez, F. (2022). El enfoque del currículo por competencias. Un análisis de la LOMLOE [The competency-based curriculum approach. An analysis of the LOMLOE]. *Revista Española de Pedagogía*, 80(281), 55-68. <https://www.jstor.org/stable/48645738>
- Love, D. (2014, 7 de junio). A conversation with Linus Torvalds, who built the world's most robust operating system and gave it away for free [Una conversación con Linus Torvalds, quien construyó el sistema operativo más robusto del mundo y lo regaló gratis]. *Insider*. <https://bit.ly/3XIEqSA>
- Luna, F. (2019). *JavaScript: Aprende a programar en el lenguaje de la web*. RedUsers. <https://bit.ly/3Zg5zx0>
- Macías, R., Cedeño, E., Zambrano, W., Zambrano, D. y Ramírez, G. C. (2021). Scratch, tecnología utilizada como herramienta para mejorar las habilidades de razonamiento lógico y algorítmico en niños de edad escolar. *Revista Ibérica de Sistemas e Tecnologías de Informação*, (E43), 619-632. <https://dialnet.unirioja.es/servlet/articulo?codigo=8649755>
- Mackey, J. (2016, 21 de septiembre). China Pushes Coding for Kids in Effort to Tackle Innovation Gap [China impulsa la codificación para niños en un esfuerzo por abordar la brecha de innovación]. *NBC News*. <https://nbcnews.to/3B38iPU>
- Maris, S. (2018, 11 de diciembre). *Pensamiento computacional: por qué incluirlo en el proceso de aprendizaje*. Net-Learning. <https://bit.ly/3CzE5c9>
- Mathieu, M. J. (2014). *Introducción a la programación*. Grupo Editorial Patria.
- Mendoza, I. J. (2017). Python, nuevo paradigma en la educación universitaria. *Journal Boliviano de Ciencias*, 13(39), 68-75. [http://www.revistasbolivianas.ciencia.bo/scielo.php?pid=S2075-89362017000100007&script=sci\\_arttext&tlng=es](http://www.revistasbolivianas.ciencia.bo/scielo.php?pid=S2075-89362017000100007&script=sci_arttext&tlng=es)
- Monjelat, N. G., Cenacchi, M. A. y San Martín, P. S. (2018). ¿Programación para todos? Herramientas y accesibilidad: un estudio de caso. *Revista latinoamericana de educación inclusiva*, 12(1), 213-227. <https://bit.ly/3X5i7pa>

- Motoa, S. P. (2019). Pensamiento computacional. *Revista Educación y Pensamiento*, 26(26), 107-111.
- Nagpal, A. y Gabrani, G. (2019, 4-6 de febrero). *Python for data analytics, scientific and technical applications* [Python para análisis de datos, aplicaciones científicas y técnicas] [Conferencia]. 2019 Amity international conference on artificial intelligence, Dubai, United Arab Emirates. <https://doi.org/10.1109/AICAI.2019.8701341>
- Nengsih, Y. K., Handrianto, C., Pernantah, P. S., Kenedi, A. K. y Tannoubi, A. (2022). The implementation of interactive learning strategy to formulating learning objectives in package C program [La implementación de la estrategia de aprendizaje interactivo para formular objetivos de aprendizaje en el programa Paquete C]. *Spektrum: Jurnal Pendidikan Luar Sekolah (PLS)*, 10(2). <https://doi.org/10.24036/spektrumpls.v10i2.117215>
- Noh, J. y Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students [Efectos de la programación robótica en el pensamiento computacional y la creatividad de los estudiantes de primaria]. *Educational technology research and development*, 68(1), 463-484. <https://doi.org/10.1007/s11423-019-09708-w>
- Organización para la Cooperación y el Desarrollo Económicos [OCDE]. (2003). *The definition and selection of key competencies. Executive summary* [La definición y selección de competencias clave. Resumen ejecutivo]. <https://bit.ly/3Cq3XHn>
- Polanco, N., Ferrer, S. y Fernández, 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>
- Real Academia Española. (s.f.). *Alfabetizar*. Diccionario de la lengua española. <https://dle.rae.es/alfabetizar>
- Reddy, P., Sharma, B. y Chaudhary, K. (2020). Digital literacy: A review of literature [Alfabetización digital: una revisión de la literatura]. *International Journal of Technoethics (IJT)*, 11(2), 65-94. <https://doi.org/10.4018/IJT.20200701.oa1>
- Rodríguez, F., Macía, X. C. y Armas, F. X. (2020). De los contenidos a las competencias: aprender transversalmente a través del conocimiento social. *Innovación educativa*, (30), 41-56. <https://doi.org/10.15304/ie.30.6944>
- Sáenz, J. M. (2019). *Programación y robótica en educación infantil, primaria y secundaria*. Editorial UNED. <https://bit.ly/3vPOFYx>

- Sánchez-Vera, M. M. y González-Martínez, J. (2019). Pensamiento computacional, Robótica y Programación en educación. *RiiTE Revista Interuniversitaria de Investigación en Tecnología Educativa*, (7), 8-11. <https://doi.org/10.6018/riite.407731>
- Schneidermann, P. (2017, 12 de octubre). Pour Tim Cook, le patron d'Apple, un écolier français devrait apprendre le code plutôt que l'anglais [Para Tim Cook, el jefe de Apple, un escolar francés debería aprender código en lugar de inglés]. *Konbini*. <https://bit.ly/3H2SiS6>
- Sotaminga, M. y Apolo, D. (2021). Pensamiento computacional, Scratch y educación: una experiencia colaborativa en Ecuador. En F. Silva-Garcés y I. Terceros (Coords). *Pensamiento computacional, programación creativa y ciencias de la computación para la educación: Reflexiones y experiencias desde América Latina* (pp. 43-59). CIESPAL. <https://bit.ly/3ZgrkN8>
- Tejera-Martínez, F., Aguilera, D. y Vílchez-González, J. M. (2020). Lenguajes de programación y desarrollo de competencias clave. Revisión sistemática. *Revista electrónica de investigación educativa*, 22, e27. <https://bit.ly/3jVWmtO>
- Torres-Torres, Y. D., Román-González, M. y Pérez-González, J. C. (2022). Brechas de Género en la iniciación a la Programación Informática en Educación Secundaria en España. *Revista complutense de educación*, 33(4), 701-712. <https://doi.org/10.5209/rced.76564>
- Valverde, J., Fernández, M. R. y Garrido, M. C. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia*, (46). <https://revistas.um.es/red/article/view/240311>
- Veeraraghavan, S. (2022, 23 de diciembre). *20 Most Popular Programming Languages to Learn in 2023* [Lenguajes de programación más populares para aprender en 2023]. Simpli Learn. <https://bit.ly/3GYiUCH>
- Vidal, C., Pham, T., Sepúlveda, S. y Carter, L. (2019). En búsqueda de un procedimiento de desarrollo de software modular. Simbiosis entre programación orientada a la característica y programación orientada a aspectos JPI. *Información tecnológica*, 30(3), 95-104. <http://dx.doi.org/10.4067/S0718-07642019000300095>
- Wing, J. (2006). Computational thinking [Pensamiento computacional]. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Yansen, G. (2020). Género y tecnologías digitales: ¿qué factores alejan a las mujeres de la programación y los servicios informáticos? *Teknokultura*, 17(2), 239-249. <https://dialnet.unirioja.es/servlet/articulo?codigo=7568084>