



## ¿Niños programadores? Ventajas y retos de enseñar código desde pequeños

Jonathan Zacek Alcazar Jurado<sup>1\*</sup>, Morelia Janneth Ruiz Machado<sup>2</sup>, María Magdalena Gutiérrez Constantino<sup>1</sup>

<sup>1</sup>Tecnológico Nacional de México / Instituto Tecnológico Superior de Uruapan

<sup>2</sup>Dirección General de Educación Tecnológica Industrial / Centro de Estudios Tecnológicos Industriales y de Servicio No. 27

\*Contacto: [jonathan.aj@uruapan.tecnm.mx](mailto:jonathan.aj@uruapan.tecnm.mx)



Imagen generada por Adobe Firefly

## Resumen

Hoy en día, aprender a programar ya no es solo cosa de expertos: los niños de todo el mundo están descubriendo el fascinante mundo del código desde muy pequeños. Cada vez más escuelas integran actividades creativas que convierten la programación en un juego: robots que siguen tus instrucciones, historias animadas hechas con Scratch y videojuegos que ellos mismos pueden diseñar.

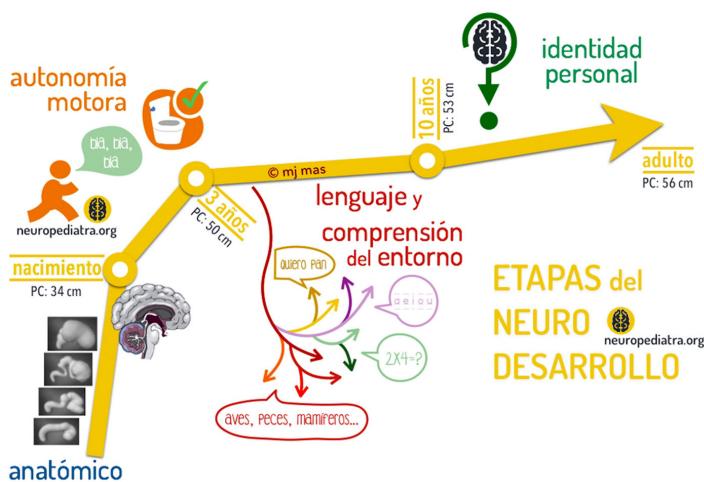
¿Pero realmente es necesario que los niños aprendan a programar desde tan temprano? Más que una moda, enseñar a programar es como darles una nueva forma de leer y escribir en el mundo digital: despierta su creatividad, fortalece su pensamiento lógico y les abre oportunidades para construir el futuro. En este artículo te invitamos a explorar de manera sencilla y divertida cómo la programación temprana está transformando la educación y por qué podría convertirse en una de las habilidades esenciales del siglo XXI.

**Palabras clave:** Programación infantil, Educación digital, Robots educativos, Aprender jugando, Habilidades del futuro.

## ¿Qué es programar y por qué enseñarlo desde pequeños?

Programar significa dar instrucciones a una computadora para que realice tareas específicas. Enseñarlo desde pequeños favorece el pensamiento lógico, la creatividad y la resolución de problemas. No se trata solo de escribir líneas de código, sino de entender cómo estructurar ideas, secuencias y consecuencias. Así como aprender un nuevo idioma desarrolla el cerebro, programar puede convertirse en una forma de alfabetización digital temprana. También estimula el desarrollo del pensamiento crítico, la capacidad de abstracción y la creatividad al resolver problemas [1].

Desde el punto de vista del neurodesarrollo, los primeros años de vida son fundamentales para establecer las bases de habilidades cognitivas complejas. La plasticidad cerebral permite que los estímulos adecuados, como la resolución de problemas y la estructuración lógica, se integren con mayor facilidad durante la infancia. La programación, como ejercicio mental, estimula áreas asociadas al razonamiento abstracto, funciones ejecutivas y memoria de trabajo [1]. Las etapas del neurodesarrollo, base de la adquisición de habilidades cognitivas complejas en contextos estructurados y lúdicos, evidencian esta relación (Fig. 1).



## Ventajas: más que solo saber usar una computadora

Estudios muestran que los niños que aprenden programación desarrollan mayor autonomía, creatividad y tolerancia al error. Además, mejora su comprensión matemática, fomenta el trabajo colaborativo y puede motivarlos a futuras carreras STEM (ciencia, tecnología, ingeniería y matemáticas) [1]. Asimismo, diversos estudios sugieren que, desde los primeros niveles educativos, el pensamiento computacional mejora la capacidad de análisis y la toma de decisiones; sus pilares —descomposición, reconocimiento de patrones, abstracción y diseño de algoritmos— se sintetizan en la Figura 2 [2].

**Figura 2.** Pilares esenciales del pensamiento computacional. Recuperado:  
<https://smartteamdigital.com/2022/11/25/ideas-pa-ra-ensenar-pensamiento-computacional/>



**Figura 1.** Mapa visual de las etapas del neurodesarrollo humano. Recuperado de:  
<https://neuropediatra.org/2015/12/16/etapas-del-nurodesarrollo/>

## Desventajas y límites necesarios

Aunque los beneficios son claros, también existen riesgos si se abusa del tiempo frente a pantallas o se enseña de forma rígida. La programación no debe reemplazar el juego libre ni las interacciones físicas. Además, los niños necesitan mediación adulta para comprender el porqué de lo que hacen y evitar frustraciones innecesarias. Por ejemplo, el exceso de tiempo frente a pantallas, sin una adecuada guía, puede afectar la calidad del sueño y la interacción social en la niñez [3]. Las recomendaciones internacionales sobre el uso de pantallas según la edad refuerzan la importancia de un acompañamiento adulto y de equilibrar las actividades digitales con experiencias reales de socialización y juego (Fig. 3). Desde la psicología infantil, se recomienda equilibrar los momentos frente a pantalla con actividades de socialización, juego libre y movimiento físico, que son igualmente esenciales para el desarrollo integral.

### TIEMPO RECOMENDADO PARA EL USO DE PANTALLAS



**Figura 3.** Recomendaciones de tiempo frente a pantallas según la edad.  
Recuperado de:  
<https://www.mspbs.gov.py/portal/27094/oms-recomienda-evitar-expo-ner-a-nintildeos-menores-de-dos-antilideos-de-vida-a-pantallas.html>.

## ¿Cómo enseñar? Herramientas para aprender jugando

Existen múltiples recursos para introducir a niños y jóvenes a la programación de forma lúdica. Plataformas como Scratch y Blockly permiten programar con bloques visuales, sin necesidad de escribir código. Code.org ofrece cursos con enfoque lúdico desde preescolar. Otros recursos como LEGO Mindstorms, Micro:bit o Arduino combinan robótica y programación [4, 5]. Estos kits de construcción, (Fig. 4), permiten a los estudiantes crear robots funcionales y personalizables mientras aprenden programación visual y basada en texto. Estos kits de robótica presentan una amplia variedad en precio, determinada por los materiales, la complejidad de sus componentes y el nivel de funcionalidad que ofrecen, desde modelos básicos de iniciación hasta plataformas avanzadas para proyectos complejos. Existen opciones para diversas edades y niveles de complejidad, lo que fomenta un aprendizaje progresivo, adaptado y accesible a diferentes contextos educativos.

Entre los prototipos que pueden armar los chicos



**Figura 4.** Kit LEGO Mindstorms Robot Inventor: Robótica y programación. Fuente: Propia

se incluyen un robot LEGO Mindstorms tipo humanoide, cuadrúpedos tipo "perro", caminadores de seis patas, seguidores de línea y mecanismos simples en madera o acrílico, algunos programables y otros basados en electrónica analógica o mecánica básica (Fig. 5). Este tipo de recursos posibilita el desarrollo de prototipos que van desde vehículos autónomos, brazos robóticos y sistemas con sensores, hasta mecanismos de exploración. Además, mediante la construcción y programación



**Figura 5.** Robot LEGO armado y otros prototipos educativos, programables y no programables. Fuente: Elaboración propia.

de estos prototipos, los niños y jóvenes reciben estímulos acordes a sus edades, favoreciendo la motricidad fina, el pensamiento lógico y la creatividad. Estas herramientas no solo desarrollan habilidades técnicas, sino que también promueven el aprendizaje activo, la perseverancia, el trabajo en equipo y el aprendizaje basado en proyectos [6].

Desde un enfoque de estimulación temprana, estos recursos permiten introducir el pensamiento computacional de forma gradual, respetando los procesos cognitivos y los tiempos individuales del aprendizaje. Además, es importante considerar la edad al seleccionar recursos: ScratchJr está diseñado para niños de 5 a 7 años, Scratch a partir de los 8, Blockly desde los 7 u 8 con acompañamiento docente, y kits como Micro:bit o Arduino a partir de los 10 u 11 años, cuando se han consolidado habilidades lógicas. La robótica educativa también integra conceptos de electrónica, mecánica y programación, ofreciendo experiencias completas y motivadoras para todas las edades [7, 8]. Se recomienda, además, mantener un uso equilibrado de las pantallas y

combinar la programación con actividades físicas, artísticas y de exploración, logrando así un desarrollo integral y estimulante para cada etapa evolutiva.

## Programar desde temprana edad, con responsabilidad y visión de futuro

La programación, cuando se introduce de manera planificada y con acompañamiento adulto, puede convertirse en una herramienta educativa poderosa para niños, niñas y jóvenes. No se trata de imponer conocimientos técnicos prematuros, sino de ofrecer oportunidades para desarrollar pensamiento crítico, creatividad y autonomía en un mundo donde la tecnología es parte cotidiana de la vida [9]. Integrar el pensamiento computacional en la infancia es una apuesta por una educación integral que forma ciudadanos digitales más conscientes y preparados para enfrentar los retos del siglo XXI. Sin embargo, este avance debe realizarse con responsabilidad, respetando los ritmos del neurodesarrollo y evitando la sobreexposición a pantallas. Diversas investigaciones en

psicología del desarrollo resaltan que los primeros años son cruciales para establecer conexiones neuronales esenciales, y que una estimulación inadecuada puede afectar áreas como la atención, el control de impulsos y la empatía [10]. Por ello, la inclusión de recursos como ScratchJr, Scratch, Blockly o kits de robótica debe adaptarse a cada etapa, siempre guiada por adultos que contextualicen la experiencia, fomenten el juego y favorezcan un desarrollo equilibrado.

Estamos a favor de introducir la programación desde edades tempranas, entendida no solo como una habilidad técnica, sino como una vía para potenciar el aprendizaje activo, la curiosidad y el trabajo colaborativo. Con una implementación responsable, estas experiencias tecnológicas pueden convertirse en herramientas de desarrollo emocional, cognitivo y social, contribuyendo a formar generaciones creativas, resilientes y preparadas para un futuro en constante transformación.

## Referencias bibliográficas

1. Resnick M, Maloney J, Monroy-Hernández A, Rusk N, Eastmond E, Brennan K, et al. Scratch: Programming for all. *Communications of the ACM*. 2009;52(11):60-67. Disponible en:  
<https://doi.org/10.1145/1592761.1592779>.
2. Bers MU. Coding as a playground: Programming and computational thinking in the early childhood classroom [Internet]. New York: Routledge; 2018. Disponible en:  
[https://api.pageplace.de/preview/DT0400.9781000194500\\_A39675062/preview-9781000194500\\_A39675062.pdf](https://api.pageplace.de/preview/DT0400.9781000194500_A39675062/preview-9781000194500_A39675062.pdf).
3. Livingstone S, Franklin B. Families and screen time: Current advice and emerging research [Internet]. LSE Media Policy Brief. 2018;(20). London: London School of Economics and Political Science. Disponible en:  
<https://eprints.lse.ac.uk/66927/1/Policy%20Brief%202017-%20Families%20%20Screen%20Time.pdf>.
4. Grover S, Pea R. Computational thinking in K-12: A review of the state of the field. *Educational Researcher*. 2013;42(1):38-43. Disponible en:  
[https://multimedia.uoc.edu/carlos/chipro/wp-content/uploads/2013/10/38.full\\_.pdf](https://multimedia.uoc.edu/carlos/chipro/wp-content/uploads/2013/10/38.full_.pdf)
5. Kafai YB, Burke Q. Connected code: Why children need to learn programming [Internet]. Cambridge (MA): MIT Press; 2014. Disponible en:  
[https://static1.squarespace.com/static/61148be5890d0754e5fe8c5b/t/61215c3b4135f6579e3aee82/1629576251725/Connected+Code\\_Rich\\_2015.pdf](https://static1.squarespace.com/static/61148be5890d0754e5fe8c5b/t/61215c3b4135f6579e3aee82/1629576251725/Connected+Code_Rich_2015.pdf).
6. Sáez-López JM, Román-González M, Vázquez-Cano E. Visual programming languages integrated across the curriculum in elementary school: A two-year case study using Scratch in five schools. *Computers & Education*. 2016;97:129-141. Disponible en:  
<https://www.sciencedirect.com/science/article/abs/pii/S0360131516300549>.
7. National Association for the Education of Young Children. Technology and young children. 2022. Disponible en: <https://www.naeyc.org/resources/topics/technology-and-media>.
8. Christakis DA. The role of media in child development. *Pediatrics*. 2019;143(6):e20193008. Disponible en:  
<https://doi.org/10.1542/peds.2019-3008>.
9. Wing JM. Computational thinking. *Communications of the ACM*. 2006;49(3):33-35. Disponible en:  
<https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>.
10. American Academy of Pediatrics. Media and young minds. *Pediatrics*. 2016;138(5):e20162591. Disponible en:  
<https://doi.org/10.1542/peds.2016-2591>.