

ÍNDICE

1. CONCEPTOS DE PROGRAMACIÓN, ROBÓTICA Y PENSAMIENTO COMPUTACIONAL	7
1. Pensamiento computacional, programación y robótica.	8
2. PROGRAMACIÓN, ROBÓTICA EN EDUCACIÓN INFANTIL	19
1. Bee-bot. Un robot en Educación Infantil	20
2. El robot Kibo.	25
3. Cubetto	28
4. Osmo	30
5. Aplicaciones: Kodable, Light bot, The foos y Scratch Jr	32
6. Propuestas didácticas	38
3. PROGRAMACIÓN, ROBÓTICA EN EDUCACIÓN PRIMARIA.	43
1. El robot m-bot. Experimentando en Educación Primaria.	44
2. Dash and Dot en Educación Primaria.	52
3. Ozobot	55
4. Aplicaciones en programación visual por bloques: Blockly y Blockly Games	56
5. Code.org	57
6. Scratch.	58
7. Ejemplo de sesión práctica con Scratch. Mostrando nuestro arte	62
4. PROGRAMACIÓN Y ROBÓTICA EN EDUCACIÓN SECUNDARIA.	69
1. El robot M Bot	70
2. Sphero	75
3. Sensores con Pico board y Raspberry Pi.	78
4. Lego Mindstorms	83

ANEXO I. VISUAL BLOCK CREATIVE COMPUTING TEST (VBCCT). PRUEBA DE COMPUTACIÓN CREATIVA VISUAL POR BLOQUES.	87
ANEXO II. SESIONES CON SCRATCH.	101
Sesión 1: Loro y perro. Primeros bloques y conceptos en computación.	101
Sesión 2: Dibujar en Scratch. Dot day	105
Sesión 3: Fichas Scratch I. Primeros bloques y conceptos en computación.	109
Sesión 4: Fichas Scratch II. Primeros bloques y conceptos en computación.	112
Sesión 5: Aprender jugando. Crear entornos atractivos y lúdicos	115
Sesión 6: Aleatoriedad y clones.	119
Sesión 7: Coordenadas y lápiz.	122
Sesión 8: Giros, interfaz y coordinación: Sistema solar	125
Sesión 9: Aprendizaje basado en juego. Condicionales y más conceptos	128
Sesión 10: Soluciones y opciones múltiples.	132
Sesión 11: Conceptos matemáticos. Coordenadas	135
Sesión 12: Pequeños proyectos. Tarjeta de Navidad	138
Sesión 13: Aprendiendo geografía. Proyectos interactivos	141
Sesión 14: Interactuando con objetos y personajes	145
Sesión 15: Aprendiendo ciencias y jugando	148
Sesión 16: Aprendiendo historia y jugando	151
Sesión 17: Mostrando nuestro arte	153
Sesión 18: Entendiendo nuestro arte	157
Sesión 19: Compartiendo arte y culturas	160
Sesión 20: Trabajando en proyectos colaborativos internacionales	163
EVALUACIÓN.	167
GLOSARIO	167
BIBLIOGRAFÍA Y REFERENCIAS	169

CONCEPTOS DE PROGRAMACIÓN, ROBÓTICA Y PENSAMIENTO COMPUTACIONAL

RESUMEN

Se detallan en este capítulo conceptos esenciales relativos al pensamiento computacional, la programación y la robótica. Se aportan investigaciones que destacan beneficios pedagógicos y ventajas de la programación y la robótica en contextos educativos. Se destaca una sustentación teórica, aportando definiciones a diferentes términos, a la vez que de detalla la situación e iniciativas en política educativa.

OBJETIVOS

- Entender el concepto de pensamiento computacional en contextos pedagógicos.
- Comprobar los resultados de investigaciones relativas a la programación y la robótica.
- Abordar elementos teóricos relativos a la programación y la robótica.
- Conocer diferentes informes y el contexto nacional e internacional en función de iniciativas educativas.
- Conocer los fundamentos científicos, matemáticos y tecnológicos del currículo de esta etapa.
- Propiciar experiencias de iniciación a las tecnologías de la información y la comunicación.
- Analizar los lenguajes audiovisuales y sus implicaciones educativas.

1. PENSAMIENTO COMPUTACIONAL, PROGRAMACIÓN Y ROBÓTICA

Papert (1980) desarrolló la teoría construccionista del aprendizaje que demuestra que el aprendizaje se produce cuando los alumnos se dedican a la creación de artefactos significativos que pueden ser explorados y compartidos. Esta teoría sostiene que cuando los niños trabajan con materiales que les permiten diseñar y construir artefactos significativos, aprenden mejor (Rogers & Portsmore, 2004). Piaget describió el constructivismo como el proceso mediante el cual los estudiantes construyen sus propios sistemas únicos de conocimiento, por lo que el docente debe centrarse en este proceso individual de construcción interna. Seymour Papert, que era estudiante de Piaget, amplió esto para describir el construccionismo en términos de ayudar al estudiante a producir construcciones que otros puedan ver y criticar.

Hoy en día hay tendencias emergentes que afectan a la integración de nuevos modelos y métodos en contextos educativos (Freeman, Adams Becker, Cummins, Davis, & Hall, 2017). Desde una perspectiva centrada en la diversión, compromiso y motivación del estudiante, se proponen la aplicación de una serie de herramientas que permiten manipular conceptos computacionales y prácticas computacionales.

Varias investigaciones destacan la importancia de la informática en la educación (Clark, Rogers, Spradling y Pais, 2013; Gardner y Feng, 2010; Lambert y Guiffre, 2009) y la necesidad de enseñar habilidades de pensamiento computacional a una temprana (Fletcher y Lu, 2009).

El **pensamiento computacional** es «la solución de problemas, el diseño de sistemas y la comprensión de la conducta humana, haciendo uso de los conceptos fundamentales de la informática» (Wing, 2006: 33). Los métodos y modelos computacionales permiten resolver problemas y diseñar sistemas que los humanos no podríamos hacer solos.

El pensamiento computacional hace referencia a procesos mentales y estrategias que incluyen: descomposición, coincidencia de patrones, abstracción, algoritmos. Se busca la descomposición de problemas en problemas más pequeños, creación de algoritmos paso a paso, la búsqueda de patrones repetitivos y la abstracción de diferen-

cias específicas para hacer que una solución funcione para múltiples problemas.

En este proceso se busca la solución a un problema o situaciones cotidianas, con la posibilidad de diferentes soluciones. En la resolución de los mencionados problemas se puede dar: descomposición, un problema complejo se desarticula y estructura en partes más abordables; reconocer patrones; abstracción, omitiendo información irrelevante del problema; algoritmos, a través de unos pasos para la resolución.

La importancia de la enseñanza de habilidades de pensamiento computacional desde una edad temprana es un elemento clave que ha captado la atención de muchos investigadores (Fletcher & Lu, 2009; Guzdial, 2008). Implica el análisis lógico y la organización de los datos; modelado, abstracciones, y simulaciones; y la identificación, las pruebas y la aplicación de posibles soluciones. Estos enfoques en el mundo de la educación ayudan a los alumnos a adquirir habilidades que se consideran esenciales en la solución de problemas complejos (Johnson, Adams Becker, Estrada, Freeman, 2014). Estos autores ya comentaban en el 2014 que estos conceptos son: «Pensamiento complejo y desafío difícil de comunicación: que comprendemos pero cuyas soluciones son difíciles de alcanzar».

Las evaluaciones deben explorar múltiples formas de conocimiento. «La intersección de los conceptos de pensamiento computacional y prácticas computacionales conduce a múltiples formas de conocimiento» (Brennan y Resnick, 2012, p. 23).

La **programación** o *coding* va adquiriendo progresivamente más presencia en la educación formal y no formal. Teniendo en cuenta que la informática utiliza las computadoras para resolver problemas, programar se refiere al algoritmo que ha sido codificado en algo que puede ser ejecutado por una máquina, por tanto se trata de producir o crear un programa. La programación es un conjunto de instrucciones que el ordenador ejecuta para alcanzar un objetivo particular. Es el arte de analizar problemas para diseñar, escribir y probar programas.

Un programador es una persona que escribe el código, que es el lenguaje que da las instrucciones al ordenador. El código son algo-

ritmos o comandos en un ordenador, un conjunto de instrucciones expresadas en un lenguaje de programación, por tanto, el código es el lenguaje que los programadores usan para decirles a los ordenadores qué hacer. La secuencia es uno de los elementos básicos para controlar las estructuras en la programación. Es el primer concepto que un estudiante debe entender al aprender a programar, pues se trata de un orden de los eventos, de este modo, el ordenador ejecutará los comandos en el orden que están insertados.

El lenguaje de programación basado en bloques es cualquier lenguaje de programación que permite a los usuarios crear programas mediante la manipulación de «bloques» o elementos gráficos de programación, en lugar de escribir código usando texto. También se le llama programación de arrastrar y soltar, codificación visual, o bloques de programación gráfica. Algunos ejemplos son Scratch y Blockly que veremos en posteriores capítulos.

Sáez-López, Román-González y Vázquez-Cano (2016) destacan la importancia de la programación visual por bloques en contextos educativos. En su investigación aportan un diseño pedagógico con este tipo de programación (Scratch 2.0) en el que los estudiantes interactúan y crean su propio contenido relacionado con las áreas curriculares; destacando varias ventajas, tales como motivación, diversión, compromiso y entusiasmo.

Se destacan mejoras relacionadas con pensamiento computacional y prácticas computacionales, así como la comprensión de conceptos computacionales a través de un enfoque activo, aprendizaje basado en proyectos, utilidad, motivación y compromiso, que pone de relieve la importancia y la eficacia de la implementación del lenguaje de programación visual por bloques y de metodologías activas en educación. Desde los resultados positivos de este estudio se recomienda implementar un Lenguaje de programación visual por bloques en entornos educativos en educación primaria a través de una implementación intercurricular.

Mitchel Resnick hace la siguiente reflexión (fuente: <https://www.edsurge.com/n/2013-05-08-learn-to-code-code-to-learn>):

¿Es importante que todos los niños aprendan a escribir? Después de todo, muy pocos de ellos en su vida adulta serán periodistas, novelistas o escritores profesionales. Cabe entonces la pregunta ¿por qué todo el mundo debe aprender a escribir? Por supuesto, estas parecen preguntas tontas. Las personas usan la escritura en todas las facetas de su vida; para enviar felicitaciones de cumpleaños a sus amigos, para hacer la lista del mercado, para consignar en sus diarios sentimientos personales. El acto de escribir también compromete a las personas en nuevas formas de pensar. A medida que escriben, aprenden a organizar, refinar y reflexionar sobre sus ideas. Claramente, existen razones poderosas para que todas las personas aprendan a escribir. Yo veo la programación (programar computadores), como extensión de la escritura. La habilidad para programar permite a las personas «escribir» nuevos tipos de cosas, tales como: historias interactivas, juegos, animaciones y simulaciones. Además, como sucede con la escritura tradicional, hay razones poderosas para que todos aprendan a programar.

Para Mitchel Resnick, el mayor reto para el futuro no es tecnológico sino cultural y educativo. Realmente lo que se necesita es un cambio de mentalidad, de manera que las personas comiencen a ver la programación no solo como un camino hacia un buen empleo, sino como una nueva forma de expresión y un nuevo contexto para el aprendizaje.

Los entornos de programación visual de bloques nos permiten experimentar, compartir y crear productos particulares con código. Los modelos y métodos computacionales nos permiten resolver problemas y diseñar sistemas que los humanos no podrían hacer solos. La alfabetización de codificación en su nivel básico también puede ayudar a los estudiantes a desarrollar habilidades transferibles (Freeman, Adams Becker, Cummins, Davis y Hall, 2017). Por lo tanto, la computadora se convierte en la herramienta esencial para la resolución de problemas (Sáez-López, Román y Vázquez-Cano, 2016). La programación y la robótica se utilizan para apoyar el aprendizaje en las escuelas, especialmente en las asignaturas de ciencias, tecnología, ingeniería, matemáticas e informática (Kim, Kim, Yuan, Hill, Doshi y Thai, 2015; Kucuk & Sisman, 2017). La programación destaca las ventajas del pensamiento lógico, las matemáticas y la creatividad.

Desde la perspectiva de diversos autores (Resnick, M.; Maloney, J.; Hernández, A. M.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B.; & Kafai, Y., 2009) la programación es una extensión de la escritura y como sucede con la escritura tradicional hay razones poderosas para que todos aprendan a programar. En Educación Primaria la programación se puede aplicar transversalmente en muchas asignaturas (matemáticas, informática, lenguaje, arte, ciencias sociales...) y, además, simultáneamente están aprendiendo estrategias para solucionar problemas, diseñar proyectos y comunicar ideas (Sáez-López y Cózar, 2017).

Por otra parte, ya hay evidencia empírica que sugiere la efectividad de la **Robótica Educativa** como una herramienta complementaria para el aprendizaje (Benitti, 2012) con muchas posibilidades y plataformas de codificación y robótica para el niveles elementales; y una necesidad de herramientas de aplicación pedagógica y evaluación de las mencionadas aplicaciones y plataformas.

Básicamente los estudiantes pueden escribir programas y probarlos en un robot virtual primero, y luego turnarse para ejecutar programas probados en los robots. Las actividades pueden ser sencillas: movimientos con ruedas, palabras, agitar las manos, sentarse, sensores táctiles, movimiento de caminar y animación. En primaria y secundaria se puede avanzar en conceptos computacionales como algoritmos, variables, condicionales, bucles, ejecución paralela o eventos.

¿Cómo ayuda entonces la robótica educativa al aprendizaje de los estudiantes? Se destacan varios estudios: *Does lego training stimulate pupils' ability to solve logical problems?* (Lindh & Holgersson, 2007) con un análisis estadístico que sugiere que no hay nada obvio en el efecto general de lego, aunque hay efectos positivos significativos de lego para subgrupos de alumnos (ANOVA)

Otro estudio: *Exploring the educational potential of robotics in schools: A systematic review* de Benitti (2012) detalla un metaestudio, 10 artículos del 2006-2009. Los artículos revisados sugieren que la robótica educativa generalmente actúa como un elemento que mejora el aprendizaje, sin embargo, este no es siempre el caso, ya que hay

estudios que informaron situaciones en las que no hubo mejoría en el aprendizaje.

Por otra parte, la investigación: *Collaborative robotic instruction: A graph teaching experience* (Mitnik, Recabarren, Nussbaum, & Soto, 2009) presenta resultados relativos a que los estudiantes que usan la actividad robótica logran un aumento significativo en sus habilidades de interpretación gráfica. Otro estudio de Spolaôr, y Vavassori-Benitti (2017) *Robotics applications grounded in learning theories on tertiary education: A systematic review* muestra evidencia empírica que sugiere la efectividad de la robótica como herramienta complementaria de aprendizaje

Por otra parte, Mazzoni y Benvenuti (2015) *A Robot-Partner for Preschool Children Learning English Using Socio-Cognitive Conflict*. Aporta un análisis de las dos condiciones experimentales (niño-niño y niño-robot) y demuestra la efectividad del conflicto Socio-Cognitivo para mejorar el aprendizaje de inglés de los niños. Otro estudio, *Assessing elementary students' computational thinking in everyday reasoning and robotics programming* (Chen, Shen, Barth-Cohen, Jiang, Huan y Eltoukhy, 2017) aporta un instrumento que se administró como una medida previa y posterior en una escuela primaria donde se adoptó un nuevo plan de estudios de robótica humanoide para el quinto grado. Los resultados muestran que el instrumento tiene buenas propiedades psicométricas y tiene el potencial de revelar los desafíos y la mejora del aprendizaje de los estudiantes y su competencia tecnológica.

Los profesores que quieren empezar a trabajar con robótica en educación deben tener en cuentas los recursos y su adaptación a las etapas y niveles educativos. Es importante que conozcan los recursos y los adapten a sus necesidades. Se aporta una clasificación en la obra, «Entornos humanos digitalizados», coordinada por Ramón Cózar (2017) en la que se destaca el capítulo: Robots educativos y programación por bloques en educación infantil y primaria: propuestas con Bee bot y M bot (Sáez-López, 2017).