

Entornos tecnológicos para el desarrollo del pensamiento computacional y de la competencia en resolución de problemas

Technological environments for the development of computational thinking and problem-solving proficiency

Diago, P. D.^a, del Olmo-Muñoz, J.^b, González-Calero, J. A.^b, Arnau, D.^a

^a *Universitat de València,*

^b *Universidad de Castilla-La Mancha*

Resumen

Las nuevas directrices curriculares de enseñanzas mínimas para los niveles no universitarios, tanto a nivel nacional como internacional, han apuntado al pensamiento computacional como una de las habilidades imprescindibles del siglo XXI. Este trabajo aporta tres experiencias de aprendizaje basadas en entornos tecnológicos que permitirán desarrollar el pensamiento computacional en distintos niveles escolares a través de la robótica educativa, programación en bloques y DragonBox Algebra. Todas ellas, enlazadas con la competencia matemática en resolución de problemas y alineadas con la propuesta curricular de cada nivel educativo.

Palabras clave: Entornos tecnológicos, Pensamiento computacional, Resolución de problemas, Tecnología educativa.

Abstract

The new curricular guidelines for basic education for non-university levels, both nationally and internationally, have pointed to computational thinking as one of the essential skills for the 21st century. This work provides three learning experiences based on technological environments that allow the development of computational thinking at different school levels through educational robotics, block programming, and DragonBox Algebra. All three of these areas are linked to mathematical competence in problem solving and aligned with the curricular proposal of each educational level.

Keywords: Technological environments, Computational thinking, Problem-solving, educational technology.

LA TRANSFORMACIÓN TECNOLÓGICA que viene acaeciendo en los últimos años ha hecho que las instituciones encargadas de la política educativa reconozcan la importancia de las habilidades relacionadas con el pensamiento computacional tanto en su faceta multidisciplinar como en su aplicación tecnológica a futuros empleos (European Schoolnet, 2015; FECYT, Google, y Everis, 2016; INTEF, 2018). Tanto es así, que el pensamiento computacional ha pasado a ser considerado como una competencia básica del siglo XXI (Consejo de la Unión Europea, 2018).

El pensamiento computacional puede ser entendido como un proceso de resolución de problemas, a la manera matemática (Pólya, 1957), donde se introducen restricciones tanto en el conjunto de instrucciones (lenguaje para expresarse - codificación) como en el conjunto de vías de resolución (formas válidas de resolución - reglas del lenguaje de codificación). Desde esta perspectiva, los entornos tecnológicos relacionados con los lenguajes de programación se convierten en escenarios privilegiados para el desarrollo del pensamiento computacional. A nivel educativo, Reino Unido fue el pionero en introducir la programación como parte de su currículum oficial para Educación Primaria en 2014 (Department for Education, 2013) y desde entonces, varios países europeos han incluido el desarrollo del pensamiento computacional entre las enseñanzas mínimas de las etapas de formación básica. En España, con objeto de adaptar el sistema educativo a los retos y desafíos del siglo XXI y, de acuerdo con los objetivos fijados por la Unión Europea y la UNESCO para la década 2020-2030, los nuevos documentos curriculares de todas las etapas no universitarias establecen el desarrollo de destrezas del pensamiento computacional. Además, se hace explícita su relación directa con la resolución de problemas, la generalización y los procesos algorítmicos en entornos tecnológicos.

EL PENSAMIENTO COMPUTACIONAL EN EDUCACIÓN MATEMÁTICA

El pensamiento computacional ha sido uno de los grandes actores del escenario educativo de los últimos años ligado, especialmente, al uso de la tecnología (Ilic et al., 2018). Su definición formal apareció en 2006, en la clásica publicación de Wing (2006). Allí se definía el pensamiento computacional como una aproximación a la resolución de problemas mediante el uso de estrategias de descomposición, diseño de algoritmos, abstracción y razonamiento lógico. Como bien apuntan muchos autores, en la definición del pensamiento computacional se entrelazan destrezas y procesos cognitivos que tradicionalmente han venido ligados a la resolución matemática de problemas (Glass, 2006; Grover y Pea, 2013; Shute et al., 2017). A pesar de la existente miscelánea de definiciones, todas ellas concuerdan en que el pensamiento computacional es “un enfoque para resolver un problema que faculta a la integración de las tecnologías digitales con las ideas humanas” (ISTE y CSTA, 2011, p. 8), favoreciendo así el uso, diseño y desarrollo de soluciones a problemas planteados en un entorno tecnológico como puede ser un robot programable o, de forma general, un ordenador.

Sin embargo, el pensamiento computacional no puede considerarse un recién llegado al campo de la educación en general y de la educación matemática en particular. En el año 1967, Wallace Feurzeig, Seymour Papert y Cynthia Solomon presentaron el lenguaje de programación *LOGO*. Aunque *LOGO* tenía características de un lenguaje de programación de propósito general, su diseño respondía a intenciones claramente educativas. De hecho, en Feurzeig et al. (1970) se presentaron los primeros resultados de experiencias con dicho lenguaje de programación en aulas de matemáticas de primaria y secundaria. En este artículo se apuntaba que una instrucción con un lenguaje de programación adecuado podría contribuir a la educación matemática en aspectos como: (a) facilitar la adquisición de un pensamiento y expresión rigurosos; (b) permitir la introducción de conceptos matemáticos clave, como la idea de variable; (c) proporcionar situaciones donde aplicar recursos heurísticos típicos de la resolución de problemas matemáticos; (d) fomentar la evaluación de las soluciones planteadas a partir de la retroalimentación automática proporcionada por el ordenador mediante el uso de un vocabulario formal desde el que organizar la discusión; (e) permitir la creación de experimentos matemáticos donde poner a prueba ideas; (f) posibilitar el contacto de las matemáticas con otras ciencias. El cambio de paradigma desde el que se aborda el uso de pensamiento computacional se pone de manifiesto en la siguiente reflexión:

Nuestro interés no es enseñar programación como un tema auxiliar, sino explorar formas de usarla como base para un curso integrado de matemáticas. Esta concepción de la programación es distinta a otras ya familiares y valiosas de la enseñanza de la programación informática como habilidad práctica por sí misma o para ser utilizada en cursos especializados en aplicaciones numéricas, en matemáticas aplicadas, métodos computacionales y similares (Feurzeig et al., 1970, p. 17).

Desde el punto de vista de la didáctica de las matemáticas, y más específicamente desde la resolución matemática de problemas (Pólya, 1957), el pensamiento computacional puede ser considerado como un proceso de resolución en el que es necesario idear, generar, desplegar y gestionar estrategias que permitan abordar un problema con la restricción de que la solución debería poder ser implementada en un entorno tecnológico. Es claro, además, que este proceso de resolución exigirá la gestión de decisiones con el objeto de organizar una planificación estratégica y táctica desde el punto de vista meta-cognitivo para poder enfrentarse al problema (Schoenfeld, 1985). Desde esta perspectiva, el pensamiento computacional va más allá de la simple instrumentalización o competencia digital, pues permite tomar consciencia de los procesos que un entorno tecnológico es capaz de llevar a cabo y gestionarlos para elaborar un plan que permita hacer frente al problema planteado. En definitiva, el pensamiento computacional es un campo privilegiado donde desarrollar la enseñanza de la resolución de problemas como contenido.

En lo que sigue, describiremos, para distintos niveles, propuestas para el desarrollo del pensamiento computacional basadas en entornos tecnológicos. En ellas las

soluciones tecnológicas han sido seleccionadas de acuerdo con algunas de las características y habilidades necesarias para desarrollar dicha destreza desde el enfoque de la competencia en resolución de problemas. Así, a pesar de que el desarrollo del pensamiento computacional puede realizarse en situaciones de enseñanza y aprendizaje que no requieran de la presencia física de un entorno tecnológico con los conocidos enfoques desenchufados (de su voz en inglés, *unplugged*; Brackmann et al., 2017), en esta propuesta centraremos la atención en el potencial de las propuestas enchufadas, es decir, aquéllas en la que el alumnado hace uso directamente de dispositivos o entornos tecnológicos. Una característica de los enfoques enchufados es la retroalimentación que el entorno tecnológico produce a la solución propuesta. Como veremos en la descripción de estas propuestas, el uso de este enfoque enchufado tiene una clara intención didáctica, pues ofrece entornos tecnológicos privilegiados para la aparición de habilidades de resolución de problemas, permiten la retroalimentación inmediata a la solución implementada por el usuario y, además, permite integrar contextos matemáticos y científicos con las herramientas tecnológicas desde las primeras edades escolares.

EL PENSAMIENTO COMPUTACIONAL EN LOS NUEVOS CURRÍCULOS

Como describiremos a continuación, uno de los objetivos del currículo para la etapa de Educación Infantil es asentar las bases del pensamiento computacional mediante procesos sencillos y manipulativos que, progresivamente, ganarán en complejidad y requerirán mayor capacidad de abstracción (Real Decreto 95/2022). A partir de la Educación Primaria se propone el establecimiento explícito de relaciones entre el pensamiento computacional y su aplicación en la vida diaria a través de recursos tecnológicos, evidenciando la necesidad de que las soluciones obtenidas puedan ser ejecutadas por un sistema informático (Real Decreto 157/2022). Y, explícitamente, en los documentos que contemplan las enseñanzas mínimas para las etapas de Educación Secundaria Obligatoria y Bachillerato se propone el desarrollo de programas mediante lenguajes informáticos de programación (Real Decreto 217/2022 y Real Decreto 243/2022, respectivamente). En particular, en lo que sigue mostramos los extractos de los decretos de enseñanzas mínimas que recogen competencias específicas relacionadas con el pensamiento computacional. En Educación Infantil, la competencia específica 2 dice:

Desarrollar, de manera progresiva, los procedimientos del método científico y las destrezas del pensamiento computacional, a través de procesos de observación y manipulación de objetos, para iniciarse en la interpretación del entorno y responder de forma creativa a las situaciones y retos que se plantean (Real Decreto 95/2022, p. 22).

En Educación Primaria, la competencia específica 4 del área de Matemáticas dice:

Utilizar el pensamiento computacional, organizando datos, descomponiendo en partes, reconociendo patrones, generalizando e interpretando, modificando y creando algoritmos de forma guiada, para modelizar y automatizar situaciones de la vida cotidiana (Real Decreto 157/2022, p. 95).

En el documento curricular para Educación Primaria se hace hincapié en conceder especial relevancia a la manipulación, en particular en los primeros niveles, por lo que las propuestas basadas en robótica educativa serían válidas y adaptables, al menos, al primer ciclo de Educación Primaria. Si bien es cierto que se promueve una transición progresiva y continua hacia recursos digitales, siendo explícita la recomendación del uso de software de programación en bloques o robótica educativa en este nivel educativo (Real Decreto 157/2022). Además, las destrezas relacionadas con el pensamiento computacional aparecen en otras áreas del decreto de mínimos, como son las áreas de Conocimiento del Medio Natural, Social y Cultural por su carácter multidisciplinar ligado a la búsqueda de soluciones, la descomposición de un problema en partes más sencillas o la creación de algoritmos para automatizar procesos de la vida cotidiana.

En Educación Secundaria Obligatoria, la competencia específica 4 de la materia de Matemáticas dice:

Utilizar los principios del pensamiento computacional organizando datos, descomponiendo en partes, reconociendo patrones, interpretando, modificando y creando algoritmos para modelizar situaciones y resolver problemas de forma eficaz (Real Decreto 217/2022, p. 143).

Aunque también se hace referencia a la habilidad computacional en las materias de Biología y Geología (competencia específica 4), Tecnología (competencia específica 4) y Tecnología y Digitalización (competencia específica 5). En su redacción encontramos respectivamente:

Utilizar el razonamiento y el pensamiento computacional, para resolver problemas o dar explicación a procesos de la vida cotidiana relacionados con la biología y la geología, analizando críticamente las respuestas y soluciones y reformulando el procedimiento, si fuera necesario (Real Decreto 217/2022, p. 35).

Desarrollar soluciones automatizadas a problemas planteados aplicando los conocimientos necesarios e incorporando tecnologías emergentes para diseñar y construir sistemas de control programables y robóticos (Real Decreto 217/2022, p. 168)

Desarrollar algoritmos y aplicaciones informáticas en distintos entornos, aplicando los principios del pensamiento computacional e incorporando las tecnologías emergentes, para crear soluciones a problemas concretos, automatizar procesos y aplicarlos en sistemas de control o en robótica (Real Decreto 217/2022, p. 175).

También se hace mención al pensamiento computacional en la competencia específica 2 del ámbito de Ciencias Aplicadas de Ciclos Formativos de Grado Básico, en la que puede leerse:

Interpretar y modelizar en términos científicos problemas y situaciones de la vida cotidiana y profesional aplicando diferentes estrategias, formas de razonamiento, herramientas tecnológicas y el pensamiento computacional para hallar y analizar soluciones asegurando su validez (Real Decreto 217/2022, p. 182).

Por último, en las enseñanzas mínimas del Bachillerato encontramos la siguiente redacción en la competencia específica 4 de la materia de Matemáticas:

Utilizar el pensamiento computacional de forma eficaz, modificando, creando y generalizando algoritmos que resuelvan problemas mediante el uso de las matemáticas, para modelizar y resolver situaciones de la vida cotidiana y del ámbito de la ciencia y la tecnología (Real Decreto 243/2022, p. 262)

Para la materia de Matemáticas aplicadas a las Ciencias Sociales encontramos una redacción similar (competencia específica 4):

Utilizar el pensamiento computacional de forma eficaz, modificando, creando y generalizando algoritmos que resuelvan problemas mediante el uso de las matemáticas, para modelizar y resolver situaciones de la vida cotidiana y del ámbito de las ciencias sociales (Real Decreto 243/2022, p. 273).

Y finalmente, en la asignatura Matemáticas Generales aparece (competencia específica 4):

Utilizar el pensamiento computacional de forma eficaz, modificando y creando algoritmos que resuelvan problemas mediante el uso de las matemáticas, para modelizar y resolver situaciones de la vida cotidiana y de diversos ámbitos. (Real Decreto 243/2022, p. 283).

ROBÓTICA EDUCATIVA EN EDUCACIÓN INFANTIL

Justificación curricular

En el decreto se destaca la necesidad, en esta etapa inicial, de incentivar el pensamiento científico y la curiosidad por el conocimiento. Se sugiere que a lo largo de la etapa de Educación Infantil se propongan situaciones de aprendizaje en la que el alumnado trate de encontrar soluciones o alternativas originales y creativas, a través de procesos manipulativos graduados en complejidad, que requieran paulatinamente mayor capacidad de abstracción. Como se indica en el decreto, dichos procesos de resolución son propios del pensamiento computacional y, entre otras estrategias, se sugiere explícitamente el uso de la descomposición de dichas tareas en otras más simples, a través de la exploración e investigación (Real Decreto 95/2022).

La aproximación al pensamiento computacional en Educación Infantil mediante la robótica educativa nos posiciona en un escenario privilegiado para abordar distintos aspectos curriculares propios de esta etapa. Por un lado, el propio robot tangible, como elemento físico manipulativo permite introducir situaciones de aprendizaje globales, significativas y estimulantes. Y por otro, la robótica educativa permite cohesionar contenidos de diferentes áreas curriculares desde el punto de vista multidisciplinar y transversal. En particular, el conocido enfoque *STEM* (del inglés *Science, Technology, Engineering and Mathematics*) proporciona escenarios en los que es posible una integración real de la tecnología y de la ingeniería con los contenidos matemáticos y científicos, tan limitados y, a veces escasos, en primeras edades escolares (Sullivan y Bers, 2016); y, a su vez, el enfoque *STEM* ha demostrado despertar el interés y la motivación reduciendo las diferencias de género (Master et al., 2017).

El entorno tecnológico: robots de suelo programables

En cuanto a la elección del robot, dada la amplia variedad de oferta en materia de robótica educativa (Hamilton et al., 2020), proponemos el uso de robots de suelo programables. Como se ha demostrado, con este tipo de robots además de desarrollar el pensamiento computacional se desarrollan habilidades relacionadas con el pensamiento crítico, el pensamiento creativo, la resolución de problemas, el trabajo en equipo, la toma de decisiones o el método científico (Benitti, 2012).

A continuación, mostraremos ejemplos basados en uno de los robots de suelo más conocido y utilizado por profesorado de Educación Infantil y Primaria: *Bee-bot* (Schina et al., 2021). Como se observa en la Figura 1, *Bee-bot* es un robot de suelo programable con una botonera física situada sobre el propio robot. Hacemos notar que los elementos que aquí traemos podrían adaptarse a cualquier robot de suelo cuyo funcionamiento sea similar a *Bee-bot*.

La programación de *Bee-bot* se realiza presionando, de forma secuencial, los siguientes botones: giro a derecha o a izquierda de 90° grados sobre sí mismo, avance de 15cm en línea recta o retroceso de 15cm en línea recta. Además, presenta botones de pausa (pausa el movimiento por 1 segundo), borrado (borra la secuencia de instrucciones almacenada) y GO (ejecuta las instrucciones almacenadas de forma secuencial y en el orden de introducción). Es importante resaltar que todos estos movimientos son relativos al sistema de referencia del propio robot. Como se observa en la Figura 1, las tareas típicas que involucran este tipo de robots de suelo suelen estar configuradas en tableros con cuadrículas que representan el paso de avance del robot (cuadros de 15cm en el caso de *Bee-bot*).



Figura 1. El robot de suelo Bee-bot

Geometría sintónica y habilidades espaciales

Al igual que otras áreas, la educación matemática ha estudiado el papel que juegan tanto los gestos como las posturas del propio cuerpo en contextos de enseñanza y aprendizaje, en particular en la resolución de problemas (Nemirovsky y Ferrara, 2009; Radford, 2009). Papert (1980) denominó aprendizaje sintónico a aquel que permitía a los niños usar el conocimiento de su propio cuerpo para explorar la geometría cuando programaban la tortuga de LOGO. Del mismo modo, términos como cognición encarnada (*embodied cognition*) han sido utilizados para referirse a aprendizajes basados en la experiencia física y sensorio-motriz típica de prácticas basadas en la manipulación y la interacción con objetos del mundo real (Lakoff y Núñez, 2000; Tall, 2013). En este sentido, las propuestas basadas en la programación de robots de suelo, como *Bee-bot*, giran claramente alrededor del aprendizaje sintónico, permitiendo que el niño se vea a sí mismo como el robot cuando ha de secuenciar las instrucciones de movimiento para resolver una tarea (Sabena, 2017). Es por ello que el éxito en la resolución de tareas de programación con robots de suelo va a estar estrechamente relacionado con la capacidad del estudiante de situarse en la posición del robot, y de su capacidad para pensar en los movimientos como si fuera el propio robot.

Desde el punto de vista del sentido espacial, es sabido que el trabajo orientado a la programación de robots de suelo potencia habilidades espaciales, como la rotación mental y la visualización, dado que las habilidades puestas en juego al programar el movimiento del robot involucran el conocimiento del entorno y del espacio que les rodea (Diago et al., 2022; Sabena, 2017). Así, desde el punto de vista didáctico, el uso de robots de suelo potencia directamente saberes básicos relacionados con las habilidades espaciales propuestos en el decreto de enseñanzas mínimas, a saber: la exploración del entorno a través de los sentidos, el conocimiento de nociones espaciales básicas relacionadas con el propio cuerpo y con otros objetos, tanto para ubicarse como para moverse (Real Decreto 95/2022).

Programación manipulativa, física y tangible

Es conocido el papel fundamental de los materiales manipulativos en los primeros niveles escolares, especialmente en el caso de las matemáticas, donde se revelan como nexo entre el concepto matemático abstracto y la representación física y significativa para el niño (Baroody, 2017; Bartolini y Martignone, 2020). Continuando este enfoque, desde finales del siglo XX se han diseñado diferentes entornos tecnológicos basados en lenguajes de programación tangibles para ser utilizados en contextos educativos sin mediación de pantallas o teclados (McNerney, 2004). En estos entornos tangibles, los niños usan diferentes tipos de objetos como bloques o cuentas para elaborar secuencias de instrucciones físicas, como, por ejemplo, en el entorno tecnológico *TERN* (Horn et al., 2012) o *KIBO* (Bers, 2018). Por la edad a la que van dirigidos estos entornos de programación manipulativa, los comandos de programación vienen descritos en lenguaje natural (verbal o pictórico) y, generalmente, en formato de bloque de construcción. De este modo se pueden crear programas a partir de las interacciones físicas con los bloques, apilándolos u ordenándolos en secuencia. Esta idea, en la que las instrucciones o comandos necesarios para crear programas vienen representados por bloques que pueden ser manipulados, es la que posteriormente dará soporte al software de programación en bloques, en un entorno digital esta vez.

En el caso de la mayoría de robots de suelo, como es el caso de *Bee-bot*, la creación de un programa suele llevarse a cabo presionando secuencialmente los botones dispuestos en el propio robot. Esta configuración, en la que el entorno tecnológico está conformado exclusivamente por el robot, no permite una auténtica programación tangible, ya que la única forma de pensar en el programa completo estriba en la capacidad del usuario para recordar las instrucciones secuenciadas. Así, en palabras traducidas de Papert (1980), la programación directa sobre *Bee-bot* no nos proporciona “objetos con los que pensar”, en el sentido tangible y sensorial.



Figura 2. Bloques de instrucciones tangibles de creación propia para programar Bee-bot (izquierda) y una versión comercializada para Blue-bot (TacTile, derecha)

En esta propuesta, con el fin de proporcionar un entorno tecnológico que ofrezca una verdadera experiencia de programación tangible, seguiremos un planteamiento similar al de Perlman (1976) con la tortuga de LOGO. Así, pondremos que junto con el robot de suelo se proporcione un juego físico de cartas correspondientes a los movimientos del robot que actúen como bloques físicos de instrucciones con las que programar. Este juego de cartas puede ser fácilmente elaborado con materiales de aula, como se muestra en la Figura 2.a, aunque en algunas versiones más recientes de robots de suelo (como puede ser *Blue-bot*, Figura 2.b) se ha implementado la comunicación entre tarjetas y robot por medio de una conexión Bluetooth. De este modo, el alumnado puede programar el movimiento del robot a mano agregando, reorganizando o quitando tarjetas de la secuencia de movimientos.

Se considera esencial que el alumnado elabore completamente el programa (el plan de resolución) con *Bee-bot* haciendo uso de las tarjetas antes de trasladarlo al robot. Una vez el programa ha sido diseñado y, solo entonces, se introducirán las instrucciones secuenciadas en el robot para que éste las ejecute. Esta configuración, en la que el robot de suelo está acompañado de un conjunto de instrucciones tangibles, se caracteriza por ofrecer un entorno tecnológico en el que el estudiante resuelve problemas a través de un conjunto de reglas precisas que vienen enunciadas en términos de los movimientos que puede ejecutar el robot. A la manera matemática de resolver problemas (Pólya, 1957), en este entorno tecnológico el estudiante puede: (i) realizar un plan previo a la programación por medio de un lenguaje visual de programación por bloques (tarjetas de movimiento); (ii) evaluar el plan ideado a partir de la respuesta proporcionada por el robot de suelo (el propio movimiento de *Bee-bot* sobre el tablero al reproducir la secuencia programada); y (iii) modificar el plan ideado en caso de no tener éxito en la resolución del problema. Desde este punto de vista, este es un contexto adecuado para estimular y desarrollar estrategias heurísticas y procesos de control (Diago et al., 2018; Sabena, 2017), considerados como base fundamental en competencia de resolución de problemas (Schoenfeld, 1985) y, por ende, pilar del pensamiento computacional.

SOFTWARE DE PROGRAMACIÓN VISUAL EN BLOQUES EN EDUCACIÓN PRIMARIA

Justificación curricular

En el documento de enseñanzas mínimas para la Educación Primaria se presenta el pensamiento computacional como una de las habilidades clave en el futuro del alumnado, enlazando la resolución de problemas con el planteamiento de procedimientos y estrategias para llegar a soluciones que puedan ser ejecutadas por un sistema informático, un humano o una combinación de ambos (Real Decreto

157/2022). En particular, el área de Conocimiento del Medio Natural, Social y Cultural establece de forma explícita la iniciación a la programación a través de recursos digitales, con software de programación en bloques, como saber básico. Por otra parte, en el área de Matemáticas, como criterio clave para la evaluación de la adquisición del pensamiento computacional, se introduce la descripción de rutinas y actividades sencillas de la vida cotidiana paso a paso y se aboga por el uso de herramientas tecnológicas orientadas al proceso de resolución de problemas. En concreto, para el primer ciclo se estipula la interpretación de algoritmos sencillos (rutinas o instrucciones con pasos ordenados) como saber básico para la adquisición de la competencia específica 4 del área de Matemáticas. En el caso del segundo ciclo se añade como saber básico la interpretación y modificación de algoritmos sencillos. Se propone explícitamente que estos algoritmos incluyan patrones repetitivos (bucles) y que sean abordados mediante software de programación en bloques o robótica educativa. Finalmente, en tercer ciclo se añade a estos saberes básicos la creación de algoritmos que incluyan instrucciones anidadas y condicionales, en representaciones más cercanas a la ciencia de la computación por medio, también, de la programación en bloques o la robótica educativa (Real Decreto 157/2022).

Así, propuestas basadas en software de programación en bloques, como las que describiremos, permitirán al alumnado de Educación Primaria aprender los rudimentos de los lenguajes de programación a la par que les ayudará a desarrollar competencias multidisciplinares ligadas al pensamiento computacional, la resolución de problemas, la creatividad y al trabajo en equipo (Sáez-López et al., 2016; Weintrop, 2019). Para la etapa de la Educación Secundaria Obligatoria, el desarrollo de la competencia basada en el pensamiento computacional se amplía con una mención explícita a la codificación de situaciones en un lenguaje fácil de interpretar por un sistema informático (Real Decreto 217/2022), por lo que la propuesta que viene a continuación sería apta también para este nivel educativo.

El entorno tecnológico: software de programación visual en bloques

Esta propuesta para Educación Primaria se plantea como extensión natural a la programación tangible con robots de suelo presentada anteriormente para niveles escolares inferiores, pero esta vez, en soporte digital. De esta forma, los bloques de instrucciones o comandos estarán disponibles en un entorno tecnológico basado en software, bien en aplicaciones para dispositivos móviles o en programas y sitios web accesibles desde el ordenador. A pesar de la naturaleza digital del entorno, estos bloques de programación mantendrán su carácter manipulativo gracias a su componente visual, pues la interfaz se presenta lista para ser explorada y manipulada de forma directa por el usuario (Weintrop y Wilensky, 2015). Además, a diferencia de otros lenguajes de programación más sofisticada-

dos, los bloques aparecen descritos en un lenguaje natural (verbal o pictórico) fácilmente interpretable. Estos bloques podrán ser arrastrados o apilados para conformar las secuencias de instrucciones necesarias para crear un programa al estilo computacional.

Nuevamente, el origen de este entorno de programación en bloques hemos de buscarlo en el desarrollo del lenguaje de programación *LOGO* por Seymour Papert en la década de los 70 (Feurzeig et al., 1970), quien planteó que el diseño de programas informáticos con este lenguaje podría facilitar el aprendizaje de conceptos y procedimientos matemáticos (Papert, 1980). Desde entonces, han sido numerosas las investigaciones con enfoque pedagógico basadas en el lenguaje *LOGO* centradas en procesos de abstracción, razonamiento algebraico y resolución de problemas (Clements y Sarama, 1997; Hoyles y Noss, 1992; entre otros), aspectos que hoy en día quedan incluidos dentro de lo que hemos denominado pensamiento computacional. Recogiendo el testigo de estos primeros esfuerzos, en el año 2004 el MIT creó un nuevo lenguaje de programación visual, llamado *Scratch* (<https://scratch.mit.edu/>), sucesor de *LOGO* y que utiliza bloques ya programados que pueden ser arrastrados y enlazados, como piezas de un puzle, para crear otros programas más complejos, juegos o simulaciones interactivas (Figura 3.a).

En la actualidad existen multitud de propuestas basadas en programación visual por bloques, todas ellas inspiradas en la apariencia y funcionamiento de *Scratch*. En particular, para niños de entre 5 a 7 años es habitual encontrar estos entornos tecnológicos en formato de aplicación para dispositivos móviles, como son las aplicaciones *ScratchJr* (Figura 3.c; Bers y Resnick, 2015) *Daisy the Dinosaur*, *Kodable* o *Light-Bot* (Gouws et al., 2013; Pila et al., 2019). Otro entorno muy popular es el sitio web *Code.org*, en el que se pueden encontrar multitud de secuencias de instrucción basadas en programación visual en bloques protagonizadas por personajes populares del ideario infantil y juvenil. En la Figura 3.b mostramos un ejemplo extraído de una secuencia de instrucción para pre-lectores y en la Figura 3.d otra para lectores competentes. Además, desde hace años se celebran anualmente iniciativas para acercar la programación y el alfabetismo digital al público general, siendo los eventos *La Hora del Código* (organizada por *Code.org*) y *La semana de la programación de la UE* (Consejo de la Unión Europea, 2018) los más populares. En los sitios web de ambas iniciativas podremos encontrar actividades, recursos y secuencias de enseñanza orientadas a la comprensión de la programación y al desarrollo de competencias fundamentales relacionadas con el pensamiento computacional.

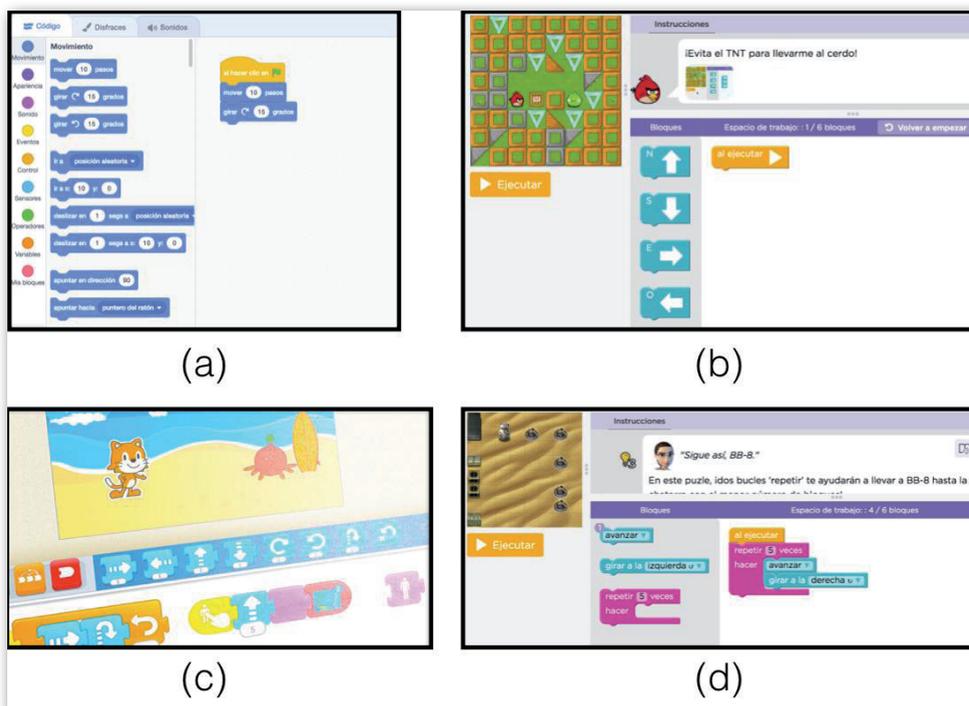


Figura 3. Entornos tecnológicos populares basados en programación visual por bloques: Scratch (a), ScratchJr (c) y Code.org (b y d)

Ventajas de la programación en bloques en educación

Por su naturaleza, el formalismo del lenguaje de programación y la limitación de las reglas posibles configuran un entorno tecnológico donde se puede realizar una discusión formal de la resolución de problemas. Aplicado a contextos educativos, es claro que el uso de entornos tecnológicos basados en programación en bloques influye en la adquisición de destrezas relacionadas con el pensamiento crítico, con la resolución de problemas, con las habilidades sociales y de autogestión, o con otros conocimientos académicos (Popat y Starkey, 2019). Por un lado, la simplicidad del interfaz hace que tanto las habilidades de inhibición como de control de decisiones se vean favorecidas en estos entornos tecnológicos de programación visual en bloques, potenciando así el trabajo cooperativo y la gestión de planes o estrategias de resolución (Arfé et al., 2020; Fessakis et al., 2013). Además, dada la intersección de los dominios computacional y matemático (Benton et al., 2017), el uso de entornos de programación en bloques produce una transferencia (de conocimientos, destrezas y/o habilidades adquiridas) tanto en contenidos como en procedimientos en tareas con rol puramente matemático (Çiftci y Bildiren, 2020; Nordby et al., 2022).

Por otro lado, la mayor complejidad a la hora de generar reglas y combinaciones de bloques ofrece un contexto mucho más cercano a los lenguajes de programación informáticos propios de las ciencias de la computación. En los últimos años, a partir de los diferentes enfoques del pensamiento computacional, se han diseñado instrumentos específicos orientados a cuantificar su grado de adquisición (Poulakis y Politis, 2021; Tang et al., 2020). Gracias a ellos, se ha podido determinar cómo la programación visual en bloques mejora la adquisición de conceptos computacionales, como por ejemplo las repeticiones (bucles), las expresiones condicionales o la gestión de eventos (Sáez-López et al., 2016). Estos elementos, cercanos a la programación informática como lenguaje, se han evidenciado de difícil comprensión para los niños, incluso en este tipo de entornos digitales (Simões et al., 2018). Tal y como ocurre en el dominio matemático, en el dominio de la programación informática el alumnado participa a través de sus actividades de aprendizaje en la construcción activa de un sistema de conocimiento basado en conceptos y habilidades procedimentales (Pea et al., 1987) que consisten en la utilización dinámica de reglas, algoritmos o procedimientos particulares para resolver un problema (Haapasalo y Kadujevich, 2000). Las experiencias tempranas con esta forma de resolución de problemas computacionales, a través de la programación en bloques, no solo deberían aliviar las mencionadas dificultades, sino que también generarán interés y prepararán al alumnado para un mundo de computación ubicua (Grover y Pea, 2013).

DRAGONBOX ALGEBRA DESDE LA EDUCACIÓN PRIMARIA A LA EDUCACIÓN SECUNDARIA

Justificación curricular

Como se pondrá de manifiesto en la descripción de sus características, *DragonBox Algebra* puede considerarse un ejemplo de cómo es posible convertir un conjunto de reglas y de símbolos propio de un dominio matemático, a otras reglas y símbolos que permiten plantear actividades típicas del pensamiento computacional en una situación de juego. En esta transformación, la intención matemática inicial de las reglas –también matemáticas– queda relegada a un segundo plano. Por esta razón, a nivel curricular, *DragonBox Algebra* podría ubicarse dentro del contenido propio del pensamiento computacional (si la intención es únicamente el desarrollo de habilidades de la resolución de problemas y del pensamiento lógico), pero también podría considerarse una herramienta para el desarrollo de contenido propio del álgebra (si la intención es utilizarlo como un soporte para el aprendizaje de las transformaciones algebraicas).

En cuanto a la redacción de las competencias y saberes relacionados con el pensamiento computacional, que encontramos en el Real Decreto 217/2022 que establece las enseñanzas mínimas para la Educación Secundaria Obligatoria, este

sigue la línea de los niveles educativos anteriores, haciendo hincapié en su relación con la resolución de problemas y el planteamiento de procedimientos por medio de la identificación de aspectos clave y la descomposición en tareas más simples. Así, como saberes básicos, se incluyen la generalización y transferencia de procesos de resolución de problemas a otras situaciones, la identificación de estrategias para la interpretación, modificación de algoritmos y la formulación de cuestiones susceptibles de ser analizadas utilizando programas y otras herramientas. Además, desde la perspectiva del desarrollo del sentido algebraico, esta propuesta podría proporcionar al estudiante conocimientos sobre las transformaciones propias del conocimiento algebraico simbólico de la Educación Secundaria Obligatoria y del Bachillerato (Real Decreto 217/2022 y Real Decreto 243/2022, respectivamente).

El entorno tecnológico: la aplicación *DragonBox Algebra*

La versión inicial de la aplicación *DragonBox Algebra* apareció en 2012 para los sistemas operativos *Android* e *iOS* con la intención de que se convirtiera en un entorno tecnológico con el que aprender álgebra a través del juego. Actualmente podemos encontrar dos versiones de la misma, *DragonBox Algebra 5+* orientada a alumnado de entre 5 y 8 años, y *DragonBox Algebra 12+* orientada a alumnado a partir de 9 años. No obstante, tras su integración en la compañía noruega *Kaboot!* en 2019, las aplicaciones están comercializadas bajo el nombre de *Kaboot! Algebra by DragonBox* y *Kaboot! Algebra 2 by DragonBox*. Ambas versiones del juego contienen 10 mundos cada una y, a su vez, cada mundo está compuesto por 20 niveles. Cada mundo cuenta con un dragón diferente contenido en una caja (de ahí el nombre del juego). La superación de cada nivel ayudará a que el dragón crezca y, al terminar el mundo, el dragón alcanzará su forma adulta.

Entre otros, *DragonBox Algebra* aborda los siguientes contenidos relacionados con el álgebra y la resolución de ecuaciones lineales:

- Transformaciones para resolver ecuaciones básicas, orientadas a compensar los miembros de la ecuación con las cuatro operaciones básicas
- Factorización y operaciones entre números con signo
- Uso de paréntesis y de la propiedad distributiva
- Factorización y simplificación de fracciones
- Adición de términos semejantes
- Sustitución algebraica

Mediante puzzles se aplican reglas algebraicas propias de la resolución de ecuaciones de primer grado como el neutro de la suma, el neutro de la multiplicación, simplificación de fracciones, obtención de fracciones equivalentes, reducción de fracciones a común denominador o transposición de términos.

El potencial de *DragonBox Algebra* para el aprendizaje del álgebra

La década de los 80 del siglo XX supuso la llegada a las aulas de los primeros sistemas algebraicos computacionales (*CAS*, del inglés *computer algebra systems*). Los *CAS* permitían realizar transformaciones algebraicas y resolver ecuaciones de manera simbólica, lo que abría la posibilidad para desarrollar secuencias de enseñanza del álgebra centradas en procesos matemáticos de nivel superior. La llegada de los entornos gráficos a los ordenadores en la década de los 90 del siglo pasado permitió el desarrollo de soluciones tecnológicas basadas en la interacción y en la representación de las situaciones matemáticas mediante sistemas de representación simultáneos, como *SimCalc* (Hegedus y Roschelle, 2013). Estos desarrollos posibilitaron reorientar la enseñanza del álgebra desde un aprendizaje memorístico de reglas a una enseñanza basada en dotar de sentido a las reglas. En aquel momento, dentro de la comunidad investigadora se introdujo la división entre modelos semánticos y sintácticos para la enseñanza del álgebra (Chaiklin, 1989; Filloy, 1987). Según esta clasificación, los modelos sintácticos ponen el énfasis en las reglas de uso, mientras que los semánticos ponen el énfasis en dar significado a las reglas. Los defensores de los modelos semánticos contribuyeron al desarrollo de aplicaciones informáticas (en este punto cabe destacar la proliferación de *applets*) donde se utilizaban modelos como la balanza para dar sentido a las reglas de transformación cuando se resolvían ecuaciones y a atacar los errores que típicamente se observan en el alumnado recientemente instruido en dicha resolución, como la dificultad para interpretar el signo igual como equivalencia. Desde este punto de vista, las transformaciones que se realizan en una ecuación serían válidas si su representación en el modelo de la balanza, por ejemplo, mantiene el equilibrio de los platos.

Con el tiempo, surgieron voces críticas con la reorientación hacia modelos puramente semánticos. Como señalaba Kirshner (2001), los fallos de los currículos no podían achacarse a un acceso al álgebra basado en la enseñanza de reglas, pues realmente nunca se había puesto en marcha un currículo exclusivamente estructural. En contraposición, este autor planteaba una propuesta curricular que suponía una combinación del componente de análisis sintáctico del álgebra, con las reglas de transformación algebraicas habituales, y que se basaba en una gramática descriptiva del álgebra que intentaba modelar la práctica de los usuarios competentes. Evidentemente, si el lenguaje algebraico se considera desde un punto de vista de la mera aplicación de reglas, su aprendizaje puede considerarse como un caso particular de pensamiento computacional. La modificación del sistema de representación del álgebra por el propio de un micromundo desde el que se introducen elementos típicos de la gamificación permite el diseño de aplicaciones como *DragonBox Algebra*. En este caso, las reglas algebraicas se convierten en reglas de un juego y la resolución de las situaciones problemáticas exigen una secuencia ordenada de un conjunto de reglas. Llegados a este punto, nada impide que el micromundo se presente con anterioridad a la enseñanza del álgebra y tenga utilidad por sí mismo sin ser necesariamente el predecesor de un nuevo contenido.

En un principio, las situaciones problemáticas de *DragonBox Algebra* plantean una superficie dividida en dos parcelas separadas por una frontera (Figura 4). La resolución de cada nivel implica dejar sola una única caja en una de las dos parcelas aplicando un conjunto de reglas (“poderes”) que se van introduciendo a medida que se avanza de nivel. La frontera representa al signo igual de la ecuación; la caja, la incógnita; y las parcelas, los términos de la ecuación. Sin embargo, en los primeros niveles no se establece ninguna relación con el formalismo matemático. En los primeros niveles no encontraremos números ni signos de operación, pero a medida que se avance de nivel la caja pasará a representarse mediante una letra equis y se introducirán los signos de las operaciones matemáticas básicas. Todo lo anterior supone que la interfaz de la aplicación no pretende poner de manifiesto, desde un inicio, la idea de equivalencia de términos de una ecuación ni pretende establecer una conexión con las matemáticas. Serán las propias reglas del juego las que de manera implícita lleven en un principio a suponer que cualquier nuevo elemento que se introduzca en una de las partes, debe tener contrapartida en el otro. De igual manera, el juego tampoco sugiere que la idea subyacente sea determinar la incógnita, sino que la situación de juego se resuelve cuando se consigue dejar la caja sola. Esto sugiere un abordaje a la idea de ecuación de mayor nivel de abstracción, pues desde esta idea se deriva tanto la resolución de ecuaciones de una incógnita como la sustitución formal. De hecho, los primeros niveles del juego plantean situaciones en las que, junto al icono de la caja, se presentan otros iconos que en ningún caso se relacionan de manera explícita con una cantidad, ya sea conocida o desconocida. Es decir, al contrario de lo que ocurre cuando se recurre a modelos semánticos, donde se pierde generalidad, en este caso se plantea una situación más abstracta.



Figura 4. Paneles de juego en *DragonBox Algebra* representando la ecuación $Ax + B = C$

A medida que se va avanzado de nivel, se introducen nuevas reglas que permiten la solución de situaciones más complejas a través de un sistema de cartas, que se ofrecen en versiones de noche y día (con fondo colorido o negro, resaltando su carácter opuesto, ver Figura 5).

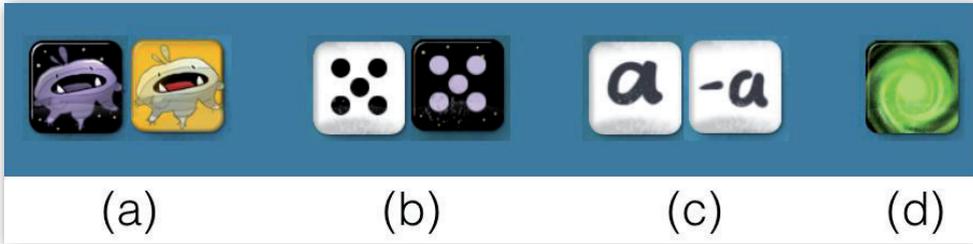


Figura 5. Cartas opuestas disponibles en DragonBox Algebra (a, b, c) y vértice verde (d)

Por ejemplo, en el nivel presentado en la Figura 4, se presenta una situación que sería equivalente a la resolución de la ecuación $Ax + B = C$ (ver secuencia de pasos en la Figura 6). Además, se ofrecen dos cartas en la parte inferior (el pez en versión noche y el insecto en versión día) que pueden usarse para ejecutar las transformaciones que permitirán aislar la caja. El pez en versión noche se utilizará para eliminar el pez versión día opuesto (la B), en un proceso que implicaría realizar tres pasos:

$$Ax + B - B = C - B \text{ (Figura 6.b)}$$

$$Ax + 0 = C - B \text{ (Figura 6.c)}$$

$$Ax = C - B \text{ (Figura 6.d)}$$

En este sentido, el juego apoya que dos cartas aisladas con igual icono, pero en versiones opuestas (noche y día, ver Figura 5.a) se convierten, al juntarse, en una carta de tipo *vértice verde* (equivalente a un 0, Figura 5.d). Los paneles izquierdos de la secuencia de la Figura 6 muestran la transformación $Ax + B - B = Ax + 0$, donde la carta *vértice verde* aislada (Figura 6.c) desaparece al pulsar sobre ella (Figura 6.d, $Ax + 0 = Ax$).

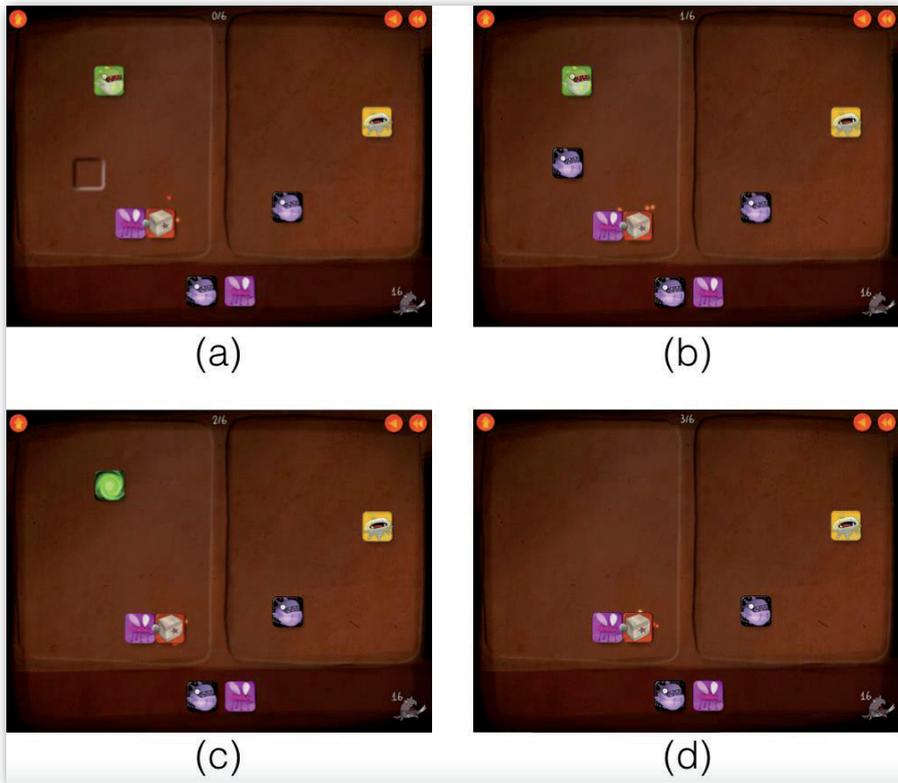


Figura 6. Secuencia de transformación de la ecuación $Ax + B = C$ a la ecuación $Ax = C - B$

Una vez se alcanza $Ax = C - B$, se aplica otra regla utilizando el insecto en versión día. En este caso, la regla establece que es posible ubicar una carta bajo otra carta o grupo de cartas, como se muestra en la secuencia de la Figura 7. El programa genera unos espacios vacíos bajo el resto de cartas aisladas para señalar que la aplicación de este poder exige aplicarla sobre el resto de cartas aisladas o grupos de cartas aisladas (Figura 7.a). Este poder sería equivalente a la transformación (Figura 7.c):

$$\frac{Ax}{A} = \frac{C}{A} - \frac{B}{A}$$

Por último, se aplican los poderes:

$$\frac{Ax}{A} = 1x = x$$

Que permiten definitivamente aislar la caja y resolver el puzzle (Figura 7.d):

$$A = \frac{C}{x} - \frac{B}{x}$$

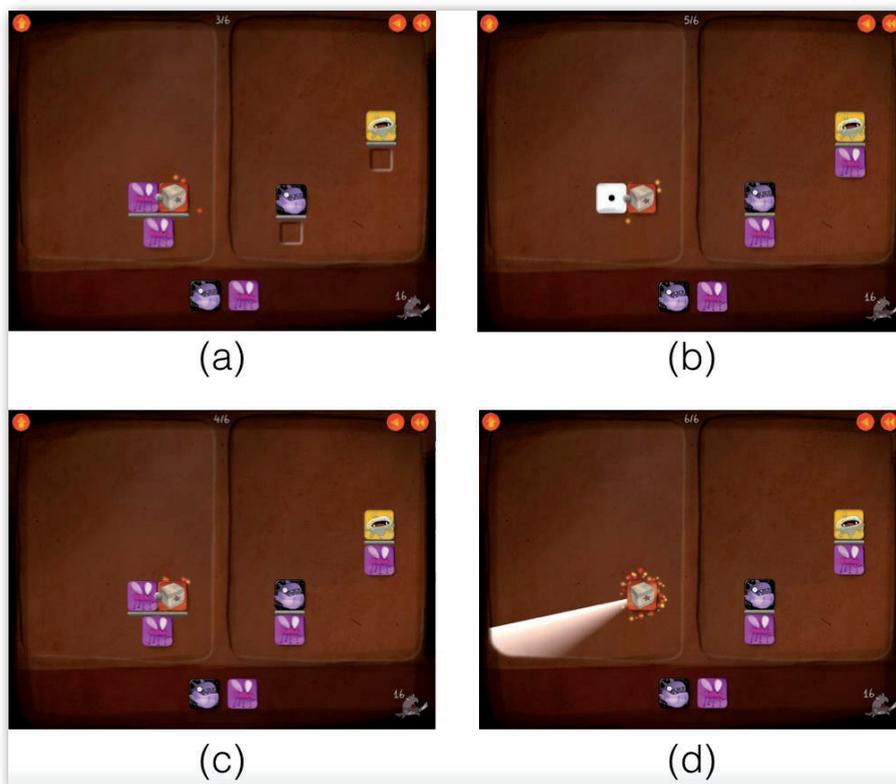


Figura 7. Secuencia de transformación de la ecuación $Ax = C - B$ a $x = C/A - B/A$

De igual manera, a medida que se van superando niveles, el sistema de signos va evolucionando desde una representación icónica (animales y bichos en versión día y noche, Figura 5.a) hasta una representación próxima al lenguaje del álgebra (a y $-a$, en los que el carácter opuesto se desliga de la representación pictórica y viene marcado por el símbolo negativo, Figura 5.c, pasando por una representación pictórica intermedia (un conjunto de puntos en versión día y noche, Figura 5.b).

En relación con el aprendizaje del álgebra, estudios recientes han mostrado cómo el uso de *DragonBox Algebra* permite al alumnado una mejor comprensión sintáctica de las reglas algebraicas a través de la verbalización de las acciones en la aplicación (Katirci, 2017), a la vez que incrementa su motivación por el aprendizaje (Gibbs, 2020). Estudios cuasi-experimentales realizados en alumnado de secundaria han mostrado resultados significativamente relevantes tanto en pensamiento algebraico, competencia en resolución algebraica, como en actitudes hacia las matemáticas tras el uso de *DragonBox Algebra* (Gutiérrez-Soto et al., 2015; Siew et al., 2016). Por último, desde el punto de vista del profesorado de primaria en formación, estos también la perciben como de utilidad para la enseñanza del álgebra (Cates, 2018).

CONSIDERACIONES FINALES

La aparición de nuevas tecnologías plantea, de manera recurrente, desafíos en el área de la educación matemática. A lo largo de las últimas décadas, el crecimiento de la tecnología y su (no siempre efectiva) aplicación en la educación matemática han evolucionado de forma paralela, algo sobre lo que la investigación ha dado cuenta desde diferentes perspectivas como el aprendizaje del alumnado (Li y Ma, 2010; Verbruggen et al., 2021), su uso en el aula (Bray y Tangney, 2017) o sus efectos en la motivación y las actitudes del alumnado (Higgins et al., 2019). Una de las irrupciones más recientes (aunque con orígenes en la década de los 70 del siglo pasado) en esta materia viene protagonizada por el pensamiento computacional, una habilidad directamente relacionada con la resolución de problemas que, tratándose de una de las competencias básicas para los años venideros, ha encontrado respuesta en forma de políticas educativas que ya lo contemplan en los diferentes documentos curriculares de enseñanzas no universitarias.

El papel vital que la tecnología tiene en los procesos de enseñanza y aprendizaje de las matemáticas, al permitir al alumnado concentrarse en la toma de decisiones, la reflexión, el razonamiento y la resolución de problemas, nos lleva a la cuestión central de este capítulo. Dicha cuestión gira alrededor de cómo los entornos tecnológicos permitirían desarrollar la "nueva" habilidad del pensamiento computacional en cada uno de los niveles escolares y a enlazarla con competencias centrales de las matemáticas como la resolución de problemas. Para dar respuesta a esto, hemos partido de propuestas en las que la forma de interactuar con dichos entornos respondería a la aplicación de un conjunto de reglas y al uso de un lenguaje concreto para abordar la resolución de problemas. Concretamente, se han presentado tres experiencias distintas de aprendizaje directamente alineadas con la propuesta curricular de cada nivel educativo.

En primer lugar, hemos visto cómo los robots de suelo programables pueden brindar, ya desde la etapa de Educación Infantil, una primera aproximación al pensamiento computacional a través de procesos de resolución que incorporen un conjunto de instrucciones secuenciales e impliquen la puesta en práctica de habilidades espaciales, de geometría sintónica y de heurística. A continuación, el uso de lenguajes de programación basados en bloques ofrece un excelente contexto educativo en el que poner en juego habilidades relacionadas con el pensamiento computacional y la resolución de problemas, a la manera matemática. Las numerosas versiones de los entornos de programación visual en bloques representarían ese subsiguiente paso natural en el que se incorporan nuevas herramientas y conceptos computacionales con los que abordar soluciones a problemas más complejos. Por último, hemos explorado el potencial dual que la aplicación *DragonBox Algebra* tiene tanto para el desarrollo de los contenidos del álgebra como para los propios del pensamiento computacional. En definitiva, en las tres propuestas presentadas, la validez de las soluciones supone razonar acerca de su construcción y su corrección matemática,

siendo el propio entorno tecnológico el que se encarga de su evaluación, ofreciendo retroalimentación inmediata.

Una vez más, los cambiantes escenarios sociales y los galopantes avances tecnológicos nos conducen a nuevos escenarios educativos en los que la tecnología debe usarse de manera cada vez más efectiva y responsable para enriquecer el desarrollo de habilidades como el pensamiento computacional, es decir la resolución de problemas embebida en contextos tecnológicos, y, en definitiva, la educación matemática.

REFERENCIAS

- Arfè, B., Vardanega, T., y Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. *Computers and Education*, 148, 103807. <https://doi.org/10.1016/j.compedu.2020.103807>
- Baroody, A. J. (2017). The Use of Concrete Experiences in Early Childhood Mathematics Instruction. En *Advances in Child Development and Behavior (Vol. 53)*. Elsevier. <https://doi.org/10.1016/bs.acdb.2017.03.001>
- Bartolini, M. G. y Martignone, F. (2020). Manipulatives in Mathematics Education. En S. Lerman (Ed.), *Encyclopedia of Mathematics Education. Second Edition*. Springer. https://doi.org/10.1007/978-94-007-4978-8_93
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A. y 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 - WiPSCE '17*, 65–72. <https://doi.org/10.1145/3137065.3137069>
- Bray, A. y Tangney, B. (2017). Technology usage in mathematics education research – A systematic review of recent trends. *Computers y Education*, 114, 255–273. <https://doi.org/10.1016/j.compedu.2017.07.004>
- Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers and Education*, 58(3), 978–988. <https://doi.org/10.1016/j.compedu.2011.10.006>
- Benton, L., Hoyles, C., Kalas, I. y Noss, R. (2017). Bridging Primary Programming and Mathematics: Some Findings of Design Research in England. *Digital Experiences in Mathematics Education*, 3, 115–138. <https://doi.org/10.1007/s40751-017-0028-x>
- Bers, M. U. (2018). Coding, playgrounds and literacy in early childhood education: The development of KIBO robotics and ScratchJr. *IEEE Global Engineering Education Conference, EDUCON*, 2100–2108. <https://doi.org/10.1109/EDUCON.2018.8363498>
- Bers, M. U., y Resnick, M. (2015). *Official ScratchJr Book*. No Starch Press.
- Cates, M. (2018). *The Effect of Using DragonBox on The Mathematics Teaching Efficacy of Preservice Middle Grade Teacher* (Tesis doctoral). Georgia State University.
- Chaiklin, S. (1989). Cognitive studies of algebra problem solving and learning. En S. Wagner y C. Kieran (Eds.), *Research issues in the learning and teaching of algebra* (pp. 93–114). Lawrence Erlbaum Associates and National Council of Teachers of Mathematics.
- Clements, D. H. y Sarama, J. (1997). Research on Logo: a decade of progress. *Computers in the Schools*, 14(1), 9–46. https://doi.org/10.1300/J025v14n01_02

- Consejo de la Unión Europea. (2018). Recomendación del Consejo, de 22 de mayo de 2018, relativa a las competencias clave para el aprendizaje permanente. *Diario Oficial de La Unión Europea*, C189/1, 1–13.
- Department for Education. (2013). Computing programmes of study. *National Curriculum in England*.
- Diago, P. D., Arnau, D. y González-Calero, J. A. (2018). Elementos de resolución de problemas en primeras edades escolares con Bee-bot. *Edma 0-6: Educación Matemática En La Infancia*, 7(1), 12–41. <https://doi.org/https://doi.org/10.24197/edmain.1.2018.12-41>
- Diago, P. D., González-Calero, J. A. y Yáñez, D. F. (2022). Exploring the development of mental rotation and computational skills in elementary students through educational robotics. *International Journal of Child-Computer Interaction*, 32, 100388. <https://doi.org/10.1016/j.ijcci.2021.100388>
- European Schoolnet. (2015). *Computing our future. Computer programming and coding: priorities, school curricula and initiatives across Europe*.
- FECYT, Google y Everis. (2016). *Educación en ciencias de la computación en España 2015*. Ministerio de Economía y Competitividad.
- Fessakis, G., Gouli, E. y Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers and Education*, 63, 87–97. <https://doi.org/http://dx.doi.org/10.1016/j.compedu.2012.11.016>
- Feurzeig, W., Papert, S., Bloom, M., Grant, R., y Solomon, C. (1970). Programming-languages as a conceptual framework for teaching mathematics. *ACM SIGCUE Outlook*, 4(2), 13–17. <https://doi.org/10.1145/965754.965757>
- Fillooy, E. (1987). Modelling and the Teaching of Algebra. En J. C. Bergeron, N. Herscovics, y C. Kieran (Eds.), *Proceedings of the 11th Conference of the International Group for Psychology of Mathematics Education*, 1 (pp. 298–300).
- Gibbs, P. S. (2020). *Game Based Learning: The Effects of DragonBox 12+ on Algebraic Performance of Middle School Students* (Tesis doctoral). University of Baltimore.
- Glass, R. L. (2006). Call it problem solving, not computational thinking. *Communications of the ACM*, 49(9), 13–13.
- Gouws, L. A., Bradshaw, K. y Wentworth, P. (2013). Computational Thinking in Educational Activities: An Evaluation of the Educational Game Light-Bot. *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, 10–15. <https://doi.org/10.1145/2462476.2466518>
- Grover, S. y Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43. <https://doi.org/10.3102/0013189X12463051>
- Gutiérrez-Soto, J., Arnau, D., y González-Calero, J. A. (2015). Un estudio exploratorio sobre el uso de DragonBox Algebra como una herramienta para la enseñanza de la resolución de ecuaciones. *ENSAYOS. Revista De La Facultad De Educación De Albacete*, 30(1), 33–44. <https://doi.org/10.18239/ensayos.v30i1.738>
- Haapasalo, L. y Kadijevich, D. (2000). Two types of mathematical knowledge and their relation. *Journal für Mathematik-Didaktik*, 21(2), 139–157.
- Hamilton, M., Clarke-Midura, J., Shumway, J. F. y Lee, V. R. (2020). An Emerging Technology Report on Computational Toys in Early Childhood. *Technology, Knowledge and Learning*, 25(1), 213–224. <https://doi.org/10.1007/s10758-019-09423-8>
- Hegedus, S. y Roschelle, J. (2013). *The SimCalc vision and contributions: Democratizing access to important mathematics*. Springer.

- Higgins, K., Huscroft-D'Angelo, J. y Crawford, L. (2019). Effects of Technology in Mathematics on Achievement, Motivation, and Attitude: A Meta-Analysis. *Journal of Educational Computing Research*, 57(2), 283–319.
<https://doi.org/10.1177/0735633117748416>
- Horn, M. S., Crouser, R. J. y Bers, M. U. (2012). Tangible interaction and learning: The case for a hybrid approach. *Personal and Ubiquitous Computing*, 16(4), 379–389.
<https://doi.org/10.1007/s00779-011-0404-2>
- Hoyles, C. y Noss, R. (1992). *Learning Mathematics and Logo*. MIT Press.
- Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF). (2018). *Programación, robótica y pensamiento computacional en el aula. Situación en España, enero 2018*. Ministerio de Educación, Cultura y Deporte.
- Ilic, U., Haseski, H. İ. y Tugtekin, U. (2018). Publication trends over 10 years of computational thinking research. *Contemporary Educational Technology*, 9(2), 131–153.
<https://doi.org/10.30935/cet.414798>
- ISTE y CSTA. (2011). *Computational Thinking, teacher resources*. Technical report.
- Katirci, N. (2017). *The Influence of DragonBox on Student Attitudes and Understanding in 7th Grade Mathematics Classroom* (Tesis doctoral). State University of New York at Albany.
- Kieran, C. (2006). Research on the teaching and learning of algebra. En Á. Gutiérrez y P. Boero (Eds.), *Handbook of research on the psychology of mathematics education: Past, present, and future* (pp. 11–49). Sense Publishers.
- Kirshner, D. (2001). The structural algebra option revisited. En R. Sutherland, T. Rojano, A. Bell, y R. Lins (Eds.), *Perspectives on School Algebra* (pp. 83–98). Kluwer Academic Publishers. https://doi.org/10.1007/0-306-47223-6_5
- Lakoff, G. y Núñez, R. E. (2000). *Where mathematics comes from: How the embodied mind brings mathematics into being*. Basic Books.
- Lí, Q. y Ma, X. (2010). A Meta-analysis of the Effects of Computer Technology on School Students' Mathematics Learning. *Educational Psychology Review*, 22(3), 215–243.
<https://doi.org/10.1007/s10648-010-9125-8>
- Master, A., Cheryan, S., Moscatelli, A. y Meltzoff, A. N. (2017). Programming experience promotes higher STEM motivation among first-grade girls. *Journal of Experimental Child Psychology*, 160, 92–106. <https://doi.org/10.1016/j.jecp.2017.03.013>
- McNerney, T. S. (2004). From turtles to Tangible Programming Bricks: Explorations in physical language design. *Personal and Ubiquitous Computing*, 8(5), 326–337.
<https://doi.org/10.1007/s00779-004-0295-6>
- Nemirovsky, R. y Ferrara, F. (2009). Mathematical imagination and embodied cognition. *Educational Studies in Mathematics*, 70(2), 159–174. <https://doi.org/10.1007/s10649-008-9150-4>
- Nordby, S. K., Bjerke, A. H. y Mifsud, L. (2022). Computational Thinking in the Primary Mathematics Classroom: a Systematic Review. *Digital Experiences in Mathematics Education*, 1, 27–49. <https://doi.org/10.1007/s40751-022-00102-5>
- Papert, S. (1980). *Mindstorms: Children, Computers and Powerful Ideas*. Basic Books.
- Pea, R. D., Soloway E. y Spohrer, J. C. (1987). The buggy path to the development of programming expertise. *Focus on Learning Problems in Mathematics*, 9, 5–30.
- Perlman, R. (1976). Using Computer Technology to Provide a Creative Learning Environment for Preschool Children. En *MIT AI Lab Memo No. 360/Logo Memo, n. 24* (pp. 1–13). MIT AI Lab. <http://18.7.29.232/handle/1721.1/5784>

- Pila, S., Aladé, F., Sheehan, K. J., Lauricella, A. R. y Wartella, E. A. (2019). Learning to code via tablet applications: An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. *Computers and Education*, 128, 52–62. <https://doi.org/10.1016/j.compedu.2018.09.006>
- Pólya, G. (1957). *How to Solve It (2nd ed.)*. Princeton University Press.
- Popat, S. y Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers and Education*, 128, 365–376. <https://doi.org/10.1016/j.compedu.2018.10.005>
- Poulakis, E. y Politis, P. (2021). Computational Thinking Assessment: Literature Review. En T. Tsiatsos, S. Demetriadis, A. Mikropoulos, y V. Dagdilelis (Eds.), *Research on E-Learning and ICT in Education: Technological, Pedagogical and Instructional Perspectives* (pp. 111–128). Springer International Publishing. https://doi.org/10.1007/978-3-030-64363-8_7
- Radford, L. (2009). Why do gestures matter? Sensuous cognition and the palpability of mathematical meanings. *Educational Studies in Mathematics*, 70(2), 111–126. <https://doi.org/10.1007/s10649-008-9127-3>
- Real Decreto 95/2022, de 1 de febrero, por el que se establece la ordenación y las enseñanzas mínimas de la Educación Infantil. *Boletín Oficial del Estado*, 28, de 2 de febrero de 2022. <https://www.boe.es/eli/es/rd/2022/02/01/95/con>
- Real Decreto 157/2022, de 1 de marzo, por el que se establece la ordenación y las enseñanzas mínimas de la Educación Primaria. *Boletín Oficial del Estado*, 52, de 2 de marzo de 2022. <https://www.boe.es/eli/es/rd/2022/03/01/157/con>
- Real Decreto 217/2022, de 29 de marzo, por el que se establece la ordenación y las enseñanzas mínimas de la Educación Secundaria Obligatoria. *Boletín Oficial del Estado*, 76, de 30 de marzo de 2022. <https://www.boe.es/eli/es/rd/2022/03/29/217/con>
- Real Decreto 243/2022, de 5 de abril, por el que se establece la ordenación y las enseñanzas mínimas del Bachillerato. *Boletín Oficial del Estado*, 82, de 6 de abril de 2022. <https://www.boe.es/eli/es/rd/2022/04/05/243/con>
- Sabena, C. (2017). Early Child Spatial Development: A Teaching Experiment with Programmable Robots. En G. Aldon, F. Hitt, L. Bazzini, y U. Gellert (Eds.), *Mathematics and Technology, Advances in Mathematics Education* (pp. 13–30). Springer International Publishing. <https://doi.org/10.1007/978-3-319-51380-5>
- Sáez-López, J. M., Román-González, M. y Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers and Education*, 97, 129–141. <https://doi.org/10.1016/j.compedu.2016.03.003>
- Schina, D., Esteve-González, V. y Usart, M. (2021). Teachers’ Perceptions of Bee-Bot Robotic Toy and Their Ability to Integrate It in Their Teaching. En Lepuschitz W., Merdan M., Koppensteiner G., Balogh R., Obdržálek D. (Eds.), *Robotics in Education. RiE 2020. Advances in Intelligent Systems and Computing* (pp. 121–132). Springer. https://doi.org/10.1007/978-3-030-67411-3_12
- Schoenfeld, A. H. (1985). *Mathematical Problem Solving*. Academic Press: Orlando.
- Shute, V. J. Sun, C., y Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Siew, N. M., Geoffrey, J. y Lee, B. N. (2016). Students’ Algebraic Thinking and Attitudes towards Algebra: The Effects of Game-Based Learning using Dragonbox12+ App. *The Research Journal of Mathematics and Technology*, 5(1), 66–79.
- Sullivan, A. y Bers, M. U. (2016). Robotics in the early childhood classroom: learning outco-

- mes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, 26(1), 3–20. <https://doi.org/10.1007/s10798-015-9304-5>
- Tall, D. (2013). *How Humans Learn to Think Mathematically*. In *How Humans Learn to Think Mathematically*. Cambridge University Press. <https://doi.org/10.1017/cbo9781139565202>
- Tang, X., Yin, Y., Lin, Q., Hadad, R. y Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers and Education*, 148, 103798. <https://doi.org/10.1016/j.compedu.2019.103798>
- Verbruggen, S., Depaepe, F. y Torbeyns, J. (2021). Effectiveness of educational technology in early mathematics education: A systematic literature review. *International Journal of Child-Computer Interaction*, 27, 100220. <https://doi.org/10.1016/j.ijcci.2020.100220>
- Weintrop, D. (2019). Block-based programming in computer science education. *Communications of the ACM*, 62(8), 22–25. <https://doi.org/10.1145/3341221>
- Weintrop, D. y Wilensky, U. (2015). To Block or not to Block, That is the Question: Students' Perceptions of Blocks-based Programming. *Proc. IDC '15. ACM*, 199–208. <https://doi.org/10.1145/2771839.2771860>
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>