

# El pensamiento computacional y las nuevas ecologías del aprendizaje

## Computational thinking and new learning ecologies

Jesús Valverde Berrocoso  
Universidad de Extremadura. España  
jevabe@unex.es

María Rosa Fernández Sánchez  
Universidad de Extremadura. España.  
rofersan@unex.es

María del Carmen Garrido Arroyo  
Universidad de Extremadura. España.  
cargarri@unex.es

### Resumen

Los sistemas educativos están incorporando en sus currículos oficiales nuevos conocimientos relacionados con el pensamiento computacional. Las administraciones educativas consideran que existen argumentos económicos, laborales, educativos, sociales y culturales para introducir la programación informática en el diseño curricular de la educación obligatoria. En este artículo se describen y analizan tres diseños curriculares que incluyen el pensamiento computacional. Por una parte, dos currículos prescriptivos (Reino Unido y Comunidad Autónoma de Madrid) organizados en torno a asignaturas, centrados en contenidos obligatorios y resultados de aprendizaje estandarizados. Por otra parte, un currículo innovador y globalizado (Q2L), que incluye el pensamiento computacional como una competencia básica, transversal y contextualizada. El análisis de los diseños permite observar dos enfoques diferentes en la introducción de la programación en las aulas. Se concluye con la necesidad de fundamentar los diseños curriculares en la experiencia acumulada sobre el uso educativo del pensamiento computacional (Papert), los resultados de la investigación educativa y las nuevas ecologías del aprendizaje.

### Palabras clave

Pensamiento computacional; Programación; Alfabetización tecnológica; Tecnología Educativa; Innovación Educativa; Diseño curricular.

### Abstract

Educational systems are incorporating in their official curricula new knowledge related to computational thinking. Education authorities consider that there are economic, labor, educational, social and cultural arguments to introduce computer programming in the curriculum of compulsory education. This article describes and analyzes three curricular

designs that include computational thinking. On the one hand, two prescriptive curricula (UK and Autonomous Community of Madrid) organized around subjects, focusing on mandatory content and standardized assessment. Moreover, an innovative and globalized curriculum (Q2L) including computational thinking as a basic, transversal and contextual skill. Analysis of the designs can observe two different approaches to the introduction of programming in the classroom. It concludes with the need to base curriculum design in the accumulated experience in the educational use of computational thinking (Papert), the results of educational research and new learning ecologies.

### **Keywords**

Computational thinking; Programming; Technological Literacy; Educational Technology; Educational Innovation; Curriculum Design.

## **Introducción**

La competencia digital de los ciudadanos es, además de una necesidad propia de una sociedad post-industrial, un derecho que debe protegerse y fomentarse con el fin de desarrollar las capacidades cognitivas, afectivas y sociales imprescindibles para interactuar, de una manera crítica, en un contexto digital enormemente flexible y cambiante. La alfabetización digital ofrece la oportunidad de adquirir competencias que permiten conocer y comprender los procesos de comunicación a través de dispositivos digitales; valorar reflexivamente qué recursos, contenidos y artefactos ofrecen un servicio de calidad adecuado a las necesidades personales y sociales; reconocer los principios ideológicos y los intereses económicos que están unidos a organizaciones e individuos que forman parte de la sociedad-red y, por último, construir y difundir mensajes en diferentes lenguajes como medio de expresión propia, libre, crítica y responsable con su comunidad.

Ya sea en el contexto político-ideológico, económico-empresarial, científico-técnico o socio-educativo, existe una visión compartida relacionada con la necesidad de una reforma profunda de los sistemas educativos, que se consideran anclados en un modelo de sociedad industrial que no satisface las actuales demandas educativas, laborales o económicas de la sociedad. Las diferencias se sitúan en el enfoque que se adopta para tratar de ofrecer soluciones a este problema. Así nos encontramos, por un lado, con propuestas que incorporan nuevos contenidos a los currículos escolares. Estas nuevas asignaturas se suman a las ya existentes sobrecargando los planes de estudio, o bien se introducen a costa de otros conocimientos que se consideran menos relevantes para la educación básica de los ciudadanos. Aquí las áreas más perjudicadas suelen ser las humanísticas (como la filosofía o las lenguas clásicas) y las artísticas (como la música o la plástica). En la justificación de estas decisiones es común acudir a argumentos de carácter económico o laboral, de modo que se prima la introducción de contenidos considerados necesarios para la cualificación de futuros profesionales en ámbitos de actividad empresarial, que se perciben con un interesante potencial de desarrollo a corto y medio plazo. Para llevar a cabo estas propuestas se modifican los currículos oficiales, por parte de las administraciones competentes (*top-down*), se invierten partidas presupuestarias para la adquisición de dispositivos tecnológicos y se aplican planes intensivos de formación de profesorado en el período temporal que la iniciativa

reformadora pervive. Como las necesidades actuales se renuevan con una gran rapidez, las reformas educativas se suceden en tan poco tiempo que los cambios apenas se producen y, en consecuencia, no se perciben ni se evidencian. Las organizaciones educativas se caracterizan por poseer una cultura escolar, fuertemente arraigada en tradiciones y rutinas docentes, que no es fácilmente modificable. En este enfoque reformador, además de vincular la selección de contenidos a criterios de carácter económico-laboral, también se aboga por trasladar modelos de gestión empresarial a la organización escolar, poniendo el acento en la evaluación de los resultados desde una perspectiva diferenciadora, competitiva, individualista y burocratizadora.

Por otro lado, vinculado con los avances científicos y con la innovación educativa, las reformas se plantean de un modo diferente. Partiendo de la misma visión de unos sistemas educativos que no satisfacen las necesidades de una sociedad digital, las soluciones pasan por incorporar nuevas metodologías en el proceso de enseñanza-aprendizaje y modificar sustancialmente el modelo organizativo de las instituciones educativas. Con relación a las metodologías, la investigación educativa ofrece evidencias que nos permiten comprender mejor los procesos de enseñanza-aprendizaje y, desde la realidad de cada contexto socio-educativo, nos ofrece conocimientos para desarrollar propuestas curriculares que generen aprendizajes profundos (comprensión vs. memorización). De ahí que el diseño instruccional sitúe su eje en la actividad (significativa, contextualizada, realista, crítica) y no en el contenido; en la globalidad del conocimiento (proyectos, casos, solución de problemas) y no en la compartimentalización de las asignaturas, que atomizan los conocimientos dificultando su comprensión y aplicabilidad en la realidad; en la evaluación para el aprendizaje (retroalimentada, auto y hetero-evaluada, formativa) y no en la evaluación de resultados, que orienta todo el proceso educativo hacia la superación de unas pruebas o test. Los centros educativos innovadores forman parte de una nueva ecología del aprendizaje que tiene como particularidad romper los límites espacio-temporales del proceso educativo gracias a las tecnologías digitales. Simultáneamente, se alteran los tradicionales roles educador-educando, puesto que en el contexto de una enseñanza y aprendizaje en red, no se define de manera inamovible quién enseña y quién aprende. Desde este enfoque se entiende que los sistemas educativos no necesitan más contenidos sino, esencialmente, entornos flexibles que permitan desarrollar las capacidades de autoaprendizaje, creatividad, autonomía, iniciativa y expresión multilingüaje.

En este contexto ha resurgido en los últimos años un movimiento educativo a nivel internacional relacionado con la introducción del pensamiento computacional, la programación informática y la robótica en las escuelas. En la historia de la Tecnología Educativa se describe cómo los primeros pasos de la informática en las aulas, en la década de los ochenta del siglo XX, estuvieron ligados al aprendizaje de la geometría mediante el uso de un lenguaje de programación denominado «Logo» y su famosa «tortuga», dentro de un proyecto educativo conformado por las ideas de Dewey, Piaget o Vygotsky, y materializado por Seymour Papert. El desarrollo del hardware y el software durante ese período no ofrecía la enorme variedad de aplicaciones que tenemos en la actualidad, las interfaces estaban prácticamente basadas en texto y el usuario tenía que escribir líneas de código para realizar determinadas operaciones. Era, por

consiguiente, un entorno favorable a la enseñanza y aprendizaje de lenguajes de programación. Pero, con la aparición de sistemas operativos visuales e intuitivos, así como la proliferación de aplicaciones informáticas «empaquetadas» con fines diversos, se desvalorizó en la actividad docente el uso de lenguajes de programación frente al aprendizaje, como usuario, de diferentes herramientas informáticas (especialmente los paquetes ofimáticos).

En la última década, la web 2.0, los dispositivos móviles, la poderosa industria del videojuego y el resurgir del movimiento DIY (Do It Yourself) han puesto en un primer plano las necesidades y carencias en la formación sobre pensamiento computacional y programación. Como respuesta a esta demanda tanto gobiernos, empresas y organizaciones sin ánimo de lucro o instituciones educativas, han desarrollado proyectos y adoptado decisiones relacionadas con el fomento de la «codificación» en la educación.

### **El pensamiento computacional en la educación desde una perspectiva socioconstructivista**

El concepto de pensamiento computacional es una competencia compleja de «alto nivel» relacionada con un modelo de conceptualización específica de los seres humanos que desarrolla ideas y vinculada con el pensamiento abstracto-matemático y con el pragmático-ingenieril que se aplica en múltiples aspectos de nuestra vida diaria. El pensamiento computacional no es sinónimo de capacidad para programar un ordenador, puesto que requiere pensar en diferentes niveles de abstracción y es independiente de los dispositivos. Se puede desarrollar pensamiento computacional sin utilizar ordenadores (basta papel y lápiz), si bien los dispositivos digitales nos permiten abordar problemas que sin ellos no nos atreveríamos a enfrentar. Por otra parte, es una competencia básica que todo ciudadano debería conocer para desenvolverse en la sociedad digital, pero no es una habilidad «rutinaria» o «mecánica», ya que es una forma de resolver problemas de manera inteligente e imaginativa (cualidades humanas que no poseen los ordenadores). Además, posee las características de combinar abstracción y pragmatismo, puesto que se fundamenta en las Matemáticas, un mundo de ideas, y se desarrolla a partir de proyectos de ingeniería que interactúan con el mundo real. Los conceptos computacionales se utilizan para enfocar y resolver problemas reales, comunicarnos con otras personas y gestionar múltiples aspectos de nuestra vida cotidiana (Wing, 2006).

«Logo» fue el primer lenguaje de programación basado en un enfoque pedagógico que permitió el aprendizaje de matemáticas, cibernética y ciencias de una forma innovadora. Fue el precursor de otro entorno de programación con un gran potencialidad educativa que se denomina «Scratch». La principal novedad fue la introducción de un objeto gráfico (la famosa «tortuga») sobre el que se aplicaban las instrucciones que los alumnos escribían en un código de fácil comprensión. Esta idea básica de proporcionar un objeto programable continúa con la mascota de *Scratch*, un gato de color naranja que puede moverse, emitir sonidos o dibujar, entre otras posibles acciones que se diseñan a

través de piezas preconfiguradas, que se encajan a modo de rompecabezas en una secuencia codificada según las intenciones del usuario. *Logo* introdujo la programación orientada a objetos dentro de la educación y, en la actualidad, la mayoría de lenguajes profesionales de programación la utilizan, por ser un método de fácil aprendizaje por su carácter intuitivo.

Al escribir código los alumnos aprenden cómo organizar un proceso, reconocen rutinas o repeticiones y descubren errores en su pensamiento computacional cuando su programa no funciona según la idea o expectativa con la que fue concebido. Todas ellas son características clave del pensamiento computacional. Mediante la codificación se pueden construir aprendizajes significativos desde un punto de vista individual, social, cultural y tangible que conduzcan a una «participación computacional».

La programación fomenta la construcción de conocimientos mediante «apropiación», es decir, de manera que los resultados de las acciones cognitivas son consideradas como propias, personales, algo que he construido yo. Pero, además, los objetos y los artefactos (programas, videojuegos, robots) juegan un papel central en este proceso porque materializan el pensamiento y, simultáneamente, lo modifican y enriquecen. La «tortuga» de *Logo* se convierte en un «objeto para pensar» con el que el usuario fácilmente se identifica. Al convertir el pensamiento en objeto mediante procedimientos, algoritmos y estructura de datos, el conocimiento personal se hace público y puede ser compartido con otros. Así el pensamiento computacional pasa a ser participativo.

El aprendizaje es un proceso social y, especialmente, en el ámbito de la cultura digital, las comunidades de usuarios conforman una nueva ecología de aprendizaje caracterizada por una alta motivación hacia la participación y la colaboración. La creación de programas no es un proceso que se realice de modo aislado sino en un contexto social. Las experiencias más innovadoras se sitúan fuera del sistema educativo. Una de las más representativas es *The Intel Computer Clubhouse Network* que posee centros en todo el mundo para que los jóvenes puedan, de manera voluntaria, producir vídeos, música y gráficos digitales. Esta iniciativa se inició en el año 1993 como un servicio del Museo de la Ciencia de Boston en colaboración con el Media Lab del Instituto Tecnológico de Massachusetts (MIT) y se ha expandido por 19 países a través de un centenar de centros<sup>1</sup>.

La actividad en estos centros se basa en cuatro principios que constituyen su modelo de aprendizaje: (a) Llevar a cabo tareas que motiven a los jóvenes a trabajar como diseñadores, inventores y creadores; (b) Fomentar que los jóvenes trabajen en proyectos relacionados con sus propios intereses; (c) Crear un sentido de comunidad, donde los jóvenes trabajen juntos con el apoyo y orientación de adultos; (d) Ofrecer recursos y oportunidades para aquellos que no pueden tener accesos a las tecnologías digitales, en un ambiente de confianza y respeto. El entorno de aprendizaje de los *Computer Clubhouse* está diseñado para lograr los siguientes resultados de aprendizaje (Michalchik, Llorente, Lundh, & Remold, 2008):

---

1 Computer Clubhouse: países y centros <http://www.computerclubhouse.org/locations>

- Capacidad para expresarse con tecnologías.
- Habilidad para colaborar, comunicar y trabajar en equipos.
- Capacidad para resolver problemas complejos.
- Habilidad para desarrollar, planificar y ejecutar proyectos complejos.
- Desarrollar la autoestima y la autoconfianza.

Estos centros están estructurados de tal forma que no sólo son inclusivos para todo tipo de jóvenes, sino que también lo son para todo tipo de experiencias. Cada miembro posee diversos contextos sociales (familia, amistades, barrio, colegio, mundos virtuales, entornos personales de aprendizaje) y el centro les ofrece un espacio de práctica para afrontar sus retos, preocupaciones y desarrollar sus oportunidades en cualquiera de sus espacios de relación. Están orientados a la creación de comunidades de aprendizaje de modo que las competencias, habilidades, herramientas y conexiones individuales revierten en el propio *Clubhouse* y en otros pertenecientes a la red. Gracias a la participación se desarrollan entornos de aprendizaje que conducen a resultados de aprendizaje muy positivos, ofreciendo un modelo para la innovación en los centros educativos.

En resumen, los *Computer Clubhouse* son una experiencia de éxito porque basan su actividad en el desarrollo de las competencias del siglo XXI a partir de resolución de problemas auténticos, el desarrollo de la creatividad en la producción de objetos digitales, la potenciación de nuevas formas de comunicación y el trabajo colaborativo. Por otra parte, la tecnología es una oportunidad para crear nuevos contenidos y productos, fomentar la participación y el compromiso con un proyecto significativo, difundir las creaciones mediante el uso de licencias *copyleft*, aportar los propios conocimientos y aprender de otros, construir la propia identidad como persona y colectivo y trabajar en torno a proyectos. El sentimiento de pertenencia al grupo y a sus resultados ofrece la oportunidad de comprometerse con la comunidad y lograr que su impacto social sea más amplio. Se ha evidenciado una influencia positiva de los miembros de los *Computer Clubhouse* en sus expectativas de futuro, su rendimiento académico y la identificación de nuevos tipos de opciones profesionales (Michalchik, Llorente, Lundh, & Remold, 2008).

Otros ejemplos de la importancia de la dimensión social del pensamiento computacional y la programación lo encontramos en otras experiencias iniciales de comunidades virtuales como *MOOSE Crossing* (Bruckman, 1997) o, más recientemente, la comunidad de *Scratch*, conformada por más de 1 millón de usuarios registrados entre los 11 y los 18 años, que comparten más de 10 millones de proyectos. Los usuarios realizan animaciones, videojuegos, historietas, simulaciones científicas y arte interactivo a través de un entorno visual de programación que, además, permite compartir el código y participar activamente en una red social para valorar los proyectos, crear colecciones y colaborar en los proyectos de otros usuarios. El perfil del usuario se define fundamentalmente por la calidad de sus proyectos y los comentarios que recibe. Se ha evidenciado que existe diversas clases de usuarios en *Scratch* y que la

realización de un proyecto y la descarga de proyectos ajenos son una puerta de acceso a todas las actividades sociales de la red. De hecho, los usuarios más activos dan más importancia al diseño de proyectos y al análisis de proyectos de otros usuarios, antes que simplemente hacer clic sobre el botón de «favorito», tan habitual en redes sociales más orientadas a las relaciones que a la creación de contenidos digitales (Fields, Giang, & Kafai, 2013). Sólo un 29% de los usuarios de Scratch comparten proyectos y la mayoría son de sexo masculino (Grimes & Fields, 2012). Aunque el mayor número de usuarios de Scratch son menores de 18 años, existen experiencias educativas con adultos que subrayan el valor social y emancipador de esta tecnología (Bustillo-Bayón, Vizcarra-Morales, & Aristizabal-Llorente, 2014).

Las dimensiones culturales del aprendizaje son especialmente relevantes en el ámbito digital donde se utilizan formas particulares de acceso al conocimiento, diferentes puntos de vista sobre la realidad o formas distintas de llevar a cabo las acciones. Así, por ejemplo, entre los programadores el enfoque de planificación estructurado y jerarquizado no siempre es el mejor método de trabajo ya que hay profesionales que prefieren un modelo de pensamiento concreto frente a la abstracción que domina la ciencia de la computación. El movimiento del software libre se ha beneficiado del entusiasmo entre los jóvenes por participar en la construcción de códigos fuente, cuya finalidad es hacer el desarrollo tecnológico accesible y explícito dentro de un contexto de máxima colaboración y creatividad.

Por último, la dimensión tangible del pensamiento computacional y la programación introduce la materialidad y la realidad física. Tanto en *Logo* como en *Scratch* los usuarios manipulan objetos en la pantalla como lo harían si estuvieran en el mundo real. Pero también se producen efectos entre objetos físicos y programación en el mundo de la robótica, que a través de herramientas como los iniciales productos *Lego/Logo*, permiten construir todo tipo de robots con motores y engranajes que son controlados por programas para realizar acciones e interactuar con el entorno.

En consecuencia, la programación no es solo una competencia cognitiva que se utiliza para diseñar códigos. Es también una competencia social y cultural que se usa para participar en grupos. Este «aprendizaje conectado» es algo más que lenguaje de programación porque nos permite comprender cómo funciona la tecnología y cómo el diseño pueden incorporar nuevas posibilidades y soluciones a problemas de la vida cotidiana. La clave de una buena enseñanza es encontrar un equilibrio entre organización y acción, que es el gran desafío de la educación. Por una parte, los educadores necesitan establecer diferencias entre el rol docente y el rol discente. Y, por otro lado, de manera paradójica, esta distinción debe difuminarse de modo que ambos se conviertan en aprendices. De la misma manera, las percepciones, las prácticas, los contextos y la cultura juegan un rol fundamental a la hora de establecer conexiones entre lo que se está aprendiendo, cómo se está aprendiendo y quién debe aprenderlo. La clave de la explicación de las dificultades de introducir el pensamiento computacional en las aulas se encuentra en estas dimensiones del «aprendizaje conectado» y deberían ser consideradas en cualquier iniciativa futura (Kafai & Burke, 2014).

## Diseños curriculares para el pensamiento computacional: 'viejas' y 'nuevas' ecologías para el aprendizaje

El pensamiento computacional y la programación empieza a formar parte del currículo oficial en los sistemas educativos formales (Záhorec, Hašková, & Munk, 2014). A continuación se describen los diseños curriculares elaborados por el Departamento de Educación del Reino Unido y la Consejería de Educación del Gobierno de la Comunidad Autónoma de Madrid, que se enmarcan dentro de propuestas curriculares prescriptivas, organizadas en torno a disciplinas académicas, para grupos homogéneos de estudiantes y con un grado de innovación educativa dependiente, en gran medida, de la opción metodológica del profesorado en cada contexto específico. Como contraste se describe un diseño curricular globalizado y basado en principios pedagógicos y metodologías didácticas coherentes con las competencias del siglo XXI, orientado por el diseño de juegos y los sistemas de pensamiento, desarrollado por un equipo docente con un alto grado de coordinación en la visión y misión del proceso de enseñanza-aprendizaje. Se trata de las escuelas «Quest To Learn» (Q2L) en los Estados Unidos.

### La asignatura «Computing» (Reino Unido)

El currículo oficial del Reino Unido introdujo, en el año 2014, una nueva asignatura denominada «Computing»<sup>2</sup> que sustituyó a la anterior asignatura «Tecnologías de la Información y la Comunicación» para los niveles educativos de Educación Primaria (*Key Stage 1 y 2*) y Educación Secundaria (*Key Stage 3 y 4*). El Departamento de Educación del gobierno británico sostiene que la introducción de la programación en el currículo se fundamenta en la relevancia del pensamiento computacional y la creatividad para comprender y cambiar el mundo. En este tipo de conocimiento computacional están implicadas diferentes disciplinas como las matemáticas, las ciencias experimentales, la tecnología o el diseño. Las tres dimensiones de este conocimiento son las ciencias de la computación que estudian lo que puede ser computado, cómo codificarlo y cómo aplicarlo a la solución de problemas; las tecnologías de la información que se ocupan de los dispositivos digitales y cómo usarlos para el almacenamiento, recuperación, transmisión y análisis de datos y, por último, la alfabetización digital o capacidad para navegar eficaz, responsable, segura y críticamente, así como crear productos digitales usando diversas tecnologías digitales. La administración británica considera que la computación permite que los estudiantes puedan crear programas, sistemas y contenidos multimedia, además desarrolla su competencia digital, es decir, la capacidad para usar, expresar y desarrollar sus ideas a través de las tecnologías de la información y la comunicación, en un nivel adecuado a su futuro profesional y como ciudadano activo en un mundo digital.

---

2 Currículo oficial para Educación Primaria, disponible en [https://www.gov.uk/government/uploads/system/uploads/attachment\\_data/file/239033/PRIMARY\\_national\\_curriculum\\_-\\_Computing.pdf](https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf) . Currículo oficial para Educación Secundaria disponible en [https://www.gov.uk/government/uploads/system/uploads/attachment\\_data/file/239067/SECONDARY\\_national\\_curriculum\\_-\\_Computing.pdf](https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/239067/SECONDARY_national_curriculum_-_Computing.pdf)



Los objetivos que se definen para este currículo específico sobre programación buscan garantizar que todos los estudiantes:

- a) Puedan comprender y aplicar los principios y conceptos fundamentales de la ciencia de la computación, incluyendo la abstracción, la lógica, los algoritmos y la representación de los datos.
- b) Puedan analizar los problemas bajo un enfoque computacional, tengan experiencia práctica en programación para resolver este tipo de problemas.
- c) Puedan evaluar y aplicar las tecnologías de la información, incluidas tecnologías emergentes (nuevas o desconocidas), analíticamente para resolver problemas.
- d) Sean usuarios responsables, competentes, seguros y creativos de las tecnologías de la información y la comunicación.

Según el currículo oficial, a los alumnos de Secundaria se les debe enseñar a (Key stage 3):

- Diseñar, usar y evaluar abstracciones computacionales que modelen el estado y comportamiento de problemas del mundo real y sistemas físicos.
- Comprender diversos algoritmos clave que reflejen un pensamiento computacional (por ejemplo, un algoritmo para clasificar y buscar); usar razonamiento lógico y comparar la utilidad de algoritmos alternativos para el mismo problema.
- Usar dos o más lenguajes de programación, al menos uno de los cuales es textual, para resolver una variedad de problemas computacionales; hacer estructuras apropiadas para usar datos (por ejemplo, listas, tablas, secuencias); diseñar y desarrollar programas modulares que usen procedimientos o funciones.
- Comprender la lógica Booleana (por ejemplo los conectores 'Y', 'O' y 'NO') y algunos de sus usos en circuitos y programación; comprender cómo los números pueden ser representados en código binario y ser capaz de llevar a cabo operaciones simples sobre números binarios (por ejemplo, suma binaria y conversión entre binario y decimal).
- Comprender los componentes de hardware y software que constituyen los sistemas informáticos, y cómo se comunican entre ellos y con otros sistemas.
- Comprender cómo las instrucciones se almacenan y ejecutan dentro de un sistema informático; comprender cómo datos de diverso tipo (incluyendo texto, sonidos e imágenes) pueden ser representados y manipulados digitalmente, en forma de dígitos binarios.
- Empezar proyectos creativos que impliquen selección, uso y combinación de

múltiples aplicaciones, preferiblemente a través de un conjunto de dispositivos, para alcanzar metas desafiantes, que incluyan recolección y análisis de datos y satisfagan necesidades de usuarios conocidos.

- Crear, reutilizar y revisar artefactos digitales para una audiencia dada, con atención a la integridad, diseño y usabilidad.
- Comprender un conjunto de formas de uso seguro, respetuoso y responsable de la tecnología, que incluya la protección de su identidad y privacidad online; reconocimiento de contenidos, conductas y contactos inapropiados y saber cómo informar de problemas.

En el ciclo superior de Educación Secundaria, todos los alumnos deben tener la oportunidad de estudiar aspectos de las TIC y la ciencia de la computación con suficiente profundidad para permitirles progresar a los más altos niveles de estudio o a una carrera profesional. Se debe enseñar a los estudiantes a (Key stage 4):

- Desarrollar su capacidad, creatividad y conocimiento en ciencia de la computación, medios digitales y tecnologías de la información.
- Desarrollar y aplicar sus habilidades analíticas, de resolución de problemas, diseño y pensamiento computacional.
- Comprender cómo los cambios en la tecnología afectan a la seguridad, incluyendo nuevas formas para proteger su privacidad e identidad online, y cómo informar de problemas.

El eje central del currículo es el pensamiento computacional entendido como el proceso de reconocer la computación en el mundo que nos rodea y aplicar herramientas y técnicas desde la programación a la comprensión y razonamiento sobre sistemas y procesos naturales o artificiales. El pensamiento computacional proporciona una estructura imprescindible para el estudio de la programación, que va más allá de la codificación en sí misma. Permite al estudiante enfrentar problemas, descomponerlos en elementos y encontrar algoritmos que los resuelvan. En consecuencia, el pensamiento computacional implica: descomposición, reconocimiento de patrones, abstracción, generalización de patrones y diseño algorítmico (Kemp, 2014).

Para la evaluación se adopta el enfoque de «evaluación para el aprendizaje» y se recomienda usar técnicas como la autoevaluación, la heteroevaluación (evaluación entre pares), el establecimiento de objetivos como medio para fomentar el aprendizaje independiente, el uso de preguntas abiertas («¿Por qué has elegido esta solución y no otra?»; «¿Puedes explicar cómo funciona?»), «KWL» (Know/Want/Learn) es decir, preguntar a los estudiantes lo que ya conocen, lo que quieren aprender y lo que han aprendido, como plataforma para la autoevaluación y el establecimiento de objetivos. Además, también se recomienda usar para la evaluación tecnologías como blogs,

comunidades online o cursos virtuales<sup>3</sup>. Con relación a la evaluación final se han eliminado los estándares oficiales de la asignatura precedente y ahora la evaluación es responsabilidad de los centros educativos que son quienes establecen los fines y los procedimientos.

### **La asignatura «Tecnología, Programación y Robótica» de la Comunidad Autónoma de Madrid (España)**

Siguiendo el modelo británico, la Consejería de Educación del Gobierno de la Comunidad Autónoma de Madrid incluyó la asignatura de libre configuración autonómica<sup>4</sup> denominada «Tecnología, Programación y Robótica», que deben cursar todos los alumnos durante el primer ciclo de la Educación Secundaria Obligatoria (tres cursos). Para cada uno de los cursos se establecen 2 horas semanales (de un total de 30 horas de clase a la semana). La materia se articula en torno a cinco ejes:

1. Programación y pensamiento computacional.
2. Robótica y la conexión con el mundo real.
3. Tecnología y el desarrollo del aprendizaje basado en proyectos.
4. Internet y su uso seguro y responsable.
5. Técnicas de diseño e impresión 3D.

En la tabla 1 se recogen los contenidos establecidos por la Administración educativa para cada uno de los tres cursos del ciclo. Como se puede observar, los conocimientos no son exclusivos de programación y pensamiento computacional, sino que se mantienen contenidos de la anterior asignatura de «Tecnología» (v.gr. electricidad, electrónica, proyecto tecnológico). Se espera que como resultados de aprendizaje los alumnos, al acabar el ciclo, sepan crear una página web y una aplicación para móviles, diseñar un videojuego, manejar la impresión en 3D y tener conocimientos de robótica. Se utiliza software open source en la programación por bloques (*Scratch*) y para el desarrollo de aplicaciones móviles (*AppInventor*). Para la robótica se usa hardware libre (*Arduino*).

Tabla 1. Contenidos de la asignatura «Tecnología, Programación y Robótica» por cursos<sup>5</sup>.

1º ESO	2º ESO	3º ESO
1. Internet: páginas Web, aplicaciones que	1. Análisis y resolución de problemas mediante	1. Formulación de un proyecto tecnológico. Identificación

3 Por ejemplo, el MOOC titulado «Cambridge GCSE Computing Online» <http://www.cambridgegcseccomputing.org/>

4 Un tipo de asignatura establecida en la LOMCE - Ley Orgánica para la Mejora de la Calidad Educativa (2013), que permite a las Administraciones educativas establecer sus contenidos, fijar horarios y definir los estándares de aprendizaje evaluables.

5 DECRETO 48/2015, de 14 de mayo, del Consejo de Gobierno, por el que se establece para la Comunidad de Madrid el currículo de la Educación Secundaria Obligatoria. Recuperado a partir de [http://www.bocm.es/boletin/CM\\_Orden\\_BOCM/2015/05/20/BOCM-20150520-1.PDF](http://www.bocm.es/boletin/CM_Orden_BOCM/2015/05/20/BOCM-20150520-1.PDF)

1º ESO	2º ESO	3º ESO
intercambian datos. ▶ Uso seguro de Internet.	algoritmos.	del problema. Análisis de su naturaleza.
2. Privacidad y responsabilidad digital.	2. Internet: arquitectura y protocolos.	2. Innovación y creatividad para la búsqueda de soluciones tecnológicas.
3. Herramientas de programación por bloques.	3. Seguridad en Internet.	3. Diseño y representación gráfica de los elementos de un proyecto tecnológico
4. Aplicaciones para dispositivos móviles.	4. Aplicaciones y servicios para internet y nuevas tendencias en la red.	4. Documentación de un proyecto para la elaboración de un prototipo tecnológico.
5. Proyectos tecnológicos. ▶ Fases del proyecto tecnológico y su documentación. ▶ Representación gráfica en proyectos tecnológicos. ▶ Innovación y creatividad tecnológica. ▶ Proyectos de desarrollo de aplicaciones informáticas.	5. Páginas Web. Gestores de contenidos (CMS) y herramientas de publicación.	5. Divulgación de la evolución de un proyecto tecnológico a través de la Web.
6. Materiales de uso tecnológico.	6. Estructuras y mecanismos.	6. Diseño y fabricación de los elementos mecánicos de un proyecto tecnológico mediante impresión 3D.
7. Electricidad y circuitos eléctricos en continua. ▶ Análisis, simulación, montaje y medida de circuitos eléctricos.	7. Diseño e impresión 3D.	7. Diseño, montaje y medida de los circuitos electrónicos de un proyecto tecnológico.
	8. Conceptos básicos de señales y sistemas de comunicaciones.	8. Programación de los circuitos electrónicos de un proyecto tecnológico.
	9. Sistemas electrónicos analógicos y digitales. ▶ Componentes eléctricos y electrónicos. ▶ Análisis, simulación, montaje y medida en circuitos electrónicos.	9. Documentación de un prototipo desarrollado a través de un proyecto tecnológico.
	10. Programación de sistemas electrónicos (robótica)	

El currículo oficial también determina los criterios de evaluación y los estándares de aprendizaje evaluables. Los indicadores del Bloque de Programación son los siguientes:

- Mantener y optimizar las funciones principales de un ordenador, tableta o teléfono móvil en los aspectos referidos a su uso, su seguridad y a las funciones del sistema operativo.
- Analizar los diferentes niveles de lenguajes de programación.
- Utilizar con destreza un entorno de programación gráfica por bloques.
- Desarrollar y programar aplicaciones móviles sencillas en entornos de programación por bloques.
- Desarrollar una página Web sobre un gestor de contenidos (CMS).
- Analizar el proceso de programación de páginas Web en un lenguaje estándar.
- Desarrollar programas en un lenguaje de programación textual (Lenguajes de programación textuales pueden ser, por ejemplo, Python, PHP, Processing, Alice, JavaScript, etc.).

Como se puede observar existe una descripción muy pormenorizada del currículo, una característica frecuente en la legislación educativa española, que deja un margen muy estrecho a docentes y centros educativos para contextualizar y adaptar los diseños

curriculares. Especialmente, los criterios de evaluación poseen un nivel de concreción tan elevado y son tan numerosos, que condicionan la metodología docente y sus posibilidades de crear entornos de aprendizaje flexibles.

### «Quest to Learn» (Q2L) (Estados Unidos de América)

Las escuelas Q2L (Nueva York y Chicago) surgen de la necesidad de diseñar entornos de aprendizaje que respondan a las necesidades de niños y niñas que crecen en un era digital, rica en información y globalmente compleja. Las instituciones que participaron en los años 2006 y 2007, en el diseño educativo de estas escuelas, que usan el «juego como aprendizaje», han sido la Fundación *MacArthur, New Visions for Public Schools, Center for Transformative Media, Parsons the New School for Design* y el *Institute of Play*. La innovación de estos centros radica en el uso de un aprendizaje situado y basado en juegos. Los juegos están diseñados bajo principios pedagógicos, todos ellos conducen a un aprendizaje significativo, son sistemas dinámicos e inmersivos, interactivos y que exigen la participación activa del jugador, ejemplifican «mundos» en los que los jugadores crecen, reciben constante feedback y desarrollan formas de pensamiento y puntos de vista diferentes sobre la realidad.

Las diez prácticas que definen «Quest to Learn» son las siguientes (Salen, Torres, Wolozin, Rufo-Tepper, & Shapiro, 2011):

- Identidad: la identidad como aprendiz es compleja e implica la pertenencia a una comunidad de práctica y el ejercicio de roles tales como escritor, diseñador, lector, productor, profesor, alumno y jugador.
- Uso del diseño de juegos y sistemas de pensamiento: todo lo que se hace en el centro educativo se conecta con la vida fuera de la escuela a través del juego y el conocimiento.
- Prácticas contextualizadas: la escuela es un espacio de práctica donde los sistemas en los que vivimos son modelados, diseñados, examinados, re-inventados y jugados, como formas de conocimiento.
- Juego y reflexión: usar el juego y reflexionar sobre mi aprendizaje con ellos.
- Teorizar y evaluar: el aprendizaje me permite validar teorías sobre el mundo.
- Respuesta a la necesidad de conocer: la motivación conduce a hacerse preguntas difíciles, buscar respuestas complejas e imaginar soluciones con otros.
- Interactuar con otros: los juegos no solo son un modelo para ayudar a pensar como funciona el mundo, sino también un medio dinámico a través del cual comprometerse socialmente y desarrollar una comprensión más profunda de uno mismo en el mundo.
- Experimentar e imaginar posibilidades: tomar riesgos, generar significados y actuar de modo creativo.
- Dar y recibir feedback: hacer visible el aprendizaje, saber como anticipar lo que

necesitaremos aprender después.

- Inventar soluciones: resolver problemas usando el diseño de juegos (identificar reglas, inventar un proceso, ejecutar y evaluar).

La arquitectura interna de los juegos (reglas, componentes, mecánicas, objetivos, conflictos, opciones y espacio) guían el diseño de las experiencias de aprendizaje. A través del currículo Q2L el diseño de juegos, que requiere un alto nivel de pensamiento complejo para asegurar que todos los elementos interactúan de modo significativo, es una estrategia de aprendizaje para los estudiantes. El objetivo de Q2L es crear entornos de aprendizaje en los que los alumnos resuelvan problemas complejos de matemáticas, ciencias, lenguaje, estudios sociales, plástica y música a través de procesos basados en juegos.

El currículo Q2L activa cinco condiciones para el aprendizaje del estudiante: una necesidad de conocer, una necesidad de compartir y reflexionar, una ocasión para compartir, un contexto para el feedback y la evaluación continua y canales para comunicarse tanto interna como externamente (Salen, Torres, Wolozin, Rufo-Tepper, & Shapiro, 2011). Las bases teóricas de este diseño curricular están en el aprendizaje situado (Brown, Collins, & Duguid, 1989), la investigación sobre el videojuego y el aprendizaje (Squire, 2006), los sistemas de pensamiento (Assaraf & Orion, 2005), el pensamiento crítico (Gee, 2003) o las nuevas alfabetizaciones (The New London Group, 1996).

Las competencias que se espera desarrollar en los alumnos conforman una serie de capacidades vinculadas con el pensamiento computacional, desde un punto de vista individual, social y cultural (Salen, Torres, Wolozin, Rufo-Tepper, & Shapiro, 2011, p. 55): distinguir lo importante de lo accesorio; identificar relaciones causales entre cosas e ideas; secuenciar causas y efectos para actuar y pensar eficazmente; establecer patrones y relaciones en el tiempo y el espacio; clarificar datos dispares y reorganizarlos en categorías; resolver tensiones y discrepancias dentro de estructuras existentes; explicar el conocimiento desde el propio punto de vista; proporcionar ejemplos relevantes desde otros conocimientos básicos que ayuden a demostrar la eficacia de conceptos fundamentales; aplicar el conocimiento a nuevas circunstancias y situaciones; justificar una teoría o idea ofreciendo evidencias en su defensa o diseñando y desarrollando un experimento para examinar la hipótesis; comparar y contrastar el conocimiento actual el conocimiento actual con otro conocimiento de una clase similar para establecer limitaciones; sintetizar la información de modo que la suma del conocimiento sea más grande que sus partes; usar la iteración para la solución de problemas.

En los centros Q2L se utiliza, entre otros, el software «*Gamestar Mechanic*»<sup>6</sup> que está diseñado bajo los principios del diseño de juegos y sistemas de pensamiento. Permite a los estudiantes construir competencias técnicas, tecnológicas, artísticas, cognitivas, sociales y lingüísticas, mediante el diseño y elaboración de videojuegos. Es una herramienta que usa la programación, pero que se orienta especialmente al diseño del

---

6 <https://gamestarmechanic.com/>

juego. El proceso integra sistemas de pensamiento, solución creativa de problemas, colaboración, diseño artístico, escritura y narrativa, diseño interactivo y alfabetización digital. El diseñador del juego debe ser también un ingeniero social, pensando cómo las personas interactúan con el juego y cómo el juego puede conformar una interacción social competitiva y colaborativa.

Los videojuegos son recursos que, tanto en su análisis como en su diseño y elaboración, permiten comprender dinámicas de retroalimentación y cómo los cambios en un nivel «micro» puede afectar a niveles «macro». También contribuyen a entender cómo funcionan los sistemas dinámicos, es decir, las múltiples relaciones que se dan dentro de un sistema. Por otra parte, explicitan las dimensiones «ocultas» de un sistema y permiten comprender cómo la modificación de determinados elementos conducen a cambios que no son fácilmente reconocibles dentro de un sistema. Los videojuegos también permiten comprender la calidad de las relaciones dentro de un sistema, es decir, si el sistema funciona o no trabajando a un nivel óptimo. Y, por último, contribuyen a entender las homologías o similitudes que pueden existir entre las dinámicas de sistemas diferentes (Salen, Torres, Wolozin, Rufo-Tepper, & Shapiro, 2011).

Los estudiantes al diseñar videojuegos aprenden a pensar de manera analítica y global, a experimentar y validar sus teorías, así como a considerar a otras personas como parte de un sistema que ellos han creado. Les permite aprender a construir significados (para ellos mismos y para sus potenciales usuarios) poniendo su creatividad, su capacidad de expresión y de innovación en la codificación y decodificación de sistemas numéricos, lingüísticos, sociales y culturales. Un enfoque que supera la dicotomía entre productor y consumidor, permitiendo que los estudiantes desarrollen competencias válidas para una ciudadanía en un mundo conectado y participativo.

## Conclusiones

La historia de la Tecnología Educativa se nos presenta con frecuencia como un proceso cíclico de avance-retroceso sobre la incorporación de dispositivos tecnológicos que prometen cambios sustanciales en el aprendizaje. La actual tendencia de las políticas educativas hacia la incorporación de la programación y del pensamiento computacional en los diseños curriculares oficiales tiene argumentos sólidos basados en demandas y necesidades de los ciudadanos de sociedades digitales que precisan conocer cómo funcionan los dispositivos que los comunican con el mundo y les permite acceder a la información como en ningún otro período de la historia.

Sin embargo, se puede observar que algunas propuestas curriculares incluyen conocimientos novedosos y oportunos bajo estructuras epistemológicas y organizativas que contradicen su propia naturaleza. El pensamiento computacional, como cualquier otra competencia básica en el siglo XXI, no debería estar limitada dentro de una asignatura de un currículo (Goode, Margolis, & Chapman, 2014). Esta fragmentación del conocimiento es contradictoria con la actual concepción del aprendizaje y con los requisitos del mundo laboral y el desenvolvimiento ciudadano en sociedades complejas e hiperconectadas. De ahí que los diseños curriculares aquí descritos del Reino Unido y

la Comunidad de Madrid, siendo loables, nacen con importantes limitaciones porque, además, su introducción no va acompañada de una reforma sustancial de la organización escolar, ni tampoco de los principios metodológicos que debería asumir el aula del siglo XXI. Es destacable cómo en el Reino Unido la evaluación de la asignatura ya no se realiza bajo estándares oficiales, mientras que en la Comunidad Autónoma de Madrid se opta por una evaluación final basada en indicadores precisos, extensos y homogéneos. Un diseño curricular como el descrito en las escuelas Q2L son un ejemplo pertinente de una propuesta y una práctica innovadora, que redefine el papel de la escuela, el docente y el alumnado bajo unos fundamentos acordes con las competencias requeridas en nuestra era digital. Demuestra que es posible cambiar las estructuras y establecer nuevos principios pedagógicos dentro del propio sistema educativo. Y, lo que es más relevante en el tema que nos ocupa, integra de manera natural el pensamiento computacional y la programación dentro del currículo en coherencia con sus principios y competencias, sin la necesidad de establecer disciplinas cerradas.

De todo ello se deriva que a la hora de elaborar propuestas curriculares es imprescindible hacer uso de las experiencias previas en educación formal y no formal, de las evidencias de la investigación educativa y la reflexión sobre la historia de la Tecnología Educativa para sustentar los diseños en las prácticas de éxito y en unas bases pedagógicas sólidas.

## Agradecimientos

Este artículo es resultado del proyecto de investigación «*Ecologías del aprendizaje en contextos múltiples: análisis de proyectos de educación expandida y conformación de ciudadanía*», financiado por el Ministerio de Economía y Competitividad – Proyectos I+D, del Programa Estatal de Fomento de la Investigación científica y técnica de excelencia, Subprograma Estatal de Generación del Conocimiento. Referencia: EDU2014-51961-P.

Presentación del artículo: 31 de Julio de 2015  
Fecha de aprobación: 2 de septiembre de 2015  
Fecha de publicación: 15 de septiembre de 2015

Valverde-Berrocoso, J., Fernández-Sánchez, M.R., Garrido-Arroyo, M.C. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *RED, Revista de Educación a Distancia*. 46(3). Septiembre de 2015. Consultado el (dd/mm/aa) en <http://www.um.es/ead/red/46>

## Referencias

Assaraf, O. B.-Z., & Orion, N. (2005). Development of system thinking skills in the context of earth system education. *Journal of Research in Science Teaching*, 42(5), 518-560. <http://doi.org/10.1002/tea.20061>



- Brown, J. S., Collins, A., & Duguid, P. (1989). Situated Cognition and the Culture of Learning. *Educational Researcher*, 18(1), 32-42.
- Bruckman, A. (1997). *MOOSE Crossing: Construction, Community, and Learning in a Networked Virtual World for Kids*. MIT Media Lab, Boston, MA. Recuperado a partir de <http://www.cc.gatech.edu/~asb/thesis/moose-crossing-entire.pdf>
- Bustillo-Bayón, J., Vizcarra-Morales, M. T., & Aristizabal-Llorente, P. (2014). Análisis del proceso formativo de un grupo de reclusos en un taller de Scratch. *Revista Latinoamericana de Tecnología Educativa - RELATEC*, 13(1), 37-49. Recuperado a partir de <http://relatec.unex.es/article/view/1279>
- Fields, D. A., Giang, M., & Kafai, Y. B. (2013). Understanding collaborative practices in the Scratch online community: Patterns of participation among youth designers. En N. Rummel, M. Kapur, M. Nathan, & S. Puntambekar (Eds.), *CSCL 2013 Conference Proceedings, Volume 1* (pp. 200-207). Madison, WI: International Society of the Learning Sciences: Recuperado a partir de [http://cdn.scratch.mit.edu/scratchr2/static/\\_ff878b6c9b6c4d18fab154a707dc253c\\_/pdfs/research/Fields\\_etal\\_CSCL\\_2013.pdf](http://cdn.scratch.mit.edu/scratchr2/static/_ff878b6c9b6c4d18fab154a707dc253c_/pdfs/research/Fields_etal_CSCL_2013.pdf)
- Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York: Palgrave Macmillan.
- Goode, J., Margolis, J., & Chapman, G. (2014). Curriculum is not enough: the educational theory and research foundation of the exploring computer science professional development model. En J. D. Dougherty & Association for Computing Machinery (Eds.), *SIGCSE '14 Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 493-498). New York: ACM. Recuperado a partir de <http://dl.acm.org/citation.cfm?doid=2538862.2538948>
- Grimes, S., & Fields, D. (2012). *Kids online: A new research agenda for understanding social networking forums*. New York: The Joan Ganz Cooney Center at Sesame Workshop. Recuperado a partir de [http://www.joanganzcooneycenter.org/wp-content/uploads/2012/11/jgcc\\_kidsonline.pdf](http://www.joanganzcooneycenter.org/wp-content/uploads/2012/11/jgcc_kidsonline.pdf)
- Kafai, Y. B., & Burke, Q. (2014). *Connected code: why children need to learn programming*. Cambridge, Massachusetts: The MIT Press.
- Kemp, P. (2014). *Computing in the national curriculum - A guide for secondary teachers. Computing at School*. London: NAACE. Recuperado a partir de <http://community.computingsatschool.org.uk/files/3383/original.pdf>
- Michalchik, V., Llorente, C., Lundh, P., & Remold, J. (2008). *A Place to Be Your Best: Youth Outcomes in the Computer Clubhouse*. Boston, MA: Center for Technology in Learning - SRI International. Recuperado a partir de <http://www.computerclubhouse.org/sites/default/files/A%20Place%20to%20Be%20Your%20Best%20FINAL%207-25.pdf>

Salen, K. S., Torres, R., Wolozin, L., Rufo-Tepper, R., & Shapiro, A. (2011). *Quest to learn: developing the school for digital kids*. Cambridge, Mass: MIT Press.

Squire, K. (2006). From Content to Context: Videogames as Designed Experience. *Educational Researcher*, 35(8), 19-29.

The New London Group. (1996). A Pedagogy of Multiliteracies: Designing Social Futures. *Harvard Educational Review*, 66(1), 60-92.

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35. Recuperado a partir de <http://exploringcs.org/wp-content/uploads/2010/09/Wing06.pdf>

Záhorec, J., Hašková, A., & Munk, M. (2014). Assessment of Selected Aspects of Teaching Programming in SK and CZ. *Informatics in Education*, 13(1), 157-178.