

El desarrollo del pensamiento computacional en educación: valoración basada en una experiencia con *Scratch*

The development of computational thinking in education:
Assessment based on an experience with *Scratch*

RECIBIDO 03/03/2021 ACEPTADO 05/05/2021 PUBLICADO 01/06/2022

 Rakel Gamito

Departamento de Didáctica y Organización Escolar, Universidad del País Vasco, España
rakel.gamito@ehu.eus

 Pilar Aristizabal

Departamento de Didáctica y Organización Escolar, Universidad del País Vasco, España
p.aristizabal@ehu.eus

 Maitane Basasoro

Departamento de Didáctica y Organización Escolar, Universidad del País Vasco, España
maitane.basasoro@ehu.eus

 Irati León

Departamento de Didáctica y Organización Escolar, Universidad del País Vasco, España
irati.leon@ehu.eus

RESUMEN

En la actual sociedad digitalizada el pensamiento computacional se ha convertido en una competencia imprescindible para la resolución de problemas cotidianos. Del mismo modo, la programación emerge con un gran potencial para el desarrollo de dicha competencia. Por ello, resulta urgente incluir la código-alfabetización en la formación inicial del futuro profesorado. En este estudio, se pretende valorar la experimentación basada en Scratch llevada a cabo con el alumnado de Grado de Educación Primaria de la Universidad del País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU). Asimismo, se ha recogido la percepción del alumnado en relación con las posibilidades de uso que ofrece Scratch para el desarrollo de habilidades y actitudes del pensamiento computacional y a las expectativas de utilización de este lenguaje de programación en su futuro profesional. Los resultados indican que el alumnado considera evidente la relación de Scratch con la enseñanza de la programación. La experiencia ha sido valorada positivamente y, a su vez, señalan que Scratch puede ser útil para el desarrollo de los procesos del pensamiento computacional en el aula de Educación Primaria, añadiendo que lo incluirán en su futuro profesional docente.

PALABRAS CLAVE pensamiento computacional, lenguajes de programación, Scratch, formación del profesorado, tecnología educativa.

ABSTRACT

In today's digitalized society, computational thinking has become an essential competence for solving everyday problems. In the same way, programming emerges with great potential for the development of such a competence. Therefore, there is an urgent need to include code literacy in the initial training of future teachers. In this study, we intend to assess the Scratch-based experimentation carried out with the students of the Primary Education Degree at the University of the Basque Country (UPV/EHU). Furthermore, the students' perceptions related both to the possibilities of use offered by Scratch for the development of computational thinking skills and attitudes and to use expectations of this programming language in their professional future have been collected. The results indicate that students consider Scratch's relationship with the teaching of programming to be evident. The experience has received positive feedback by the students who, in turn, indicate that Scratch can be useful for the development of computational thinking processes in the Primary Education classroom, adding that they will include it in their professional future as teachers.

KEYWORDS computational thinking, programming languages, Scratch, teacher education, educational technology.

1. INTRODUCCIÓN

En las últimas décadas, la sociedad se ha transformado a todos los niveles: social, económico, cultural y, también, educativo. La influencia de las Tecnologías de la Información y la Comunicación (TIC) en dicha transformación es innegable, así como la incertidumbre que todo ello crea cuando se habla del futuro.

Para responder al escenario actual resulta esencial replantearse las estrategias de la formación en competencias, los procesos de aprendizaje y las metodologías utilizadas (Alsina, & Acosta, 2018; García, & Caballero, 2019). Al fin y al cabo, "la Sociedad del Conocimiento está en continuo cambio, no sólo en relación con los recursos de que se dispone, sino también en relación con cómo aprendemos o qué debemos aprender" (González *et al.*, 2018, p. 29).

En este sentido, la competencia digital se ha convertido en competencia clave para desenvolverse en el siglo XXI (2006/962/CE). DigComp 2.1 (Carretero *et al.*, 2017) es el actual Marco Europeo de Competencias Digitales para la Ciudadanía y recoge 21 competencias, divididas en cinco áreas: alfabetización informacional, comunicación y colaboración, creación de contenidos digitales, seguridad y resolución de problemas.

A pesar de que en muchas ocasiones no se considera parte de la competencia digital (Gamito *et al.*, 2018), programar también tiene cabida en dicho marco. Dentro de la competencia del área de creación de contenidos digitales, DigComp recoge la planificación y desarrollo de secuencias de instrucciones para la resolución de problemas haciendo uso de sistemas informáticos (Carretero *et al.*, 2017). Por ese motivo, se ha reforzado la apuesta por trabajar la programación en las diferentes etapas educativas (Bender *et al.*, 2015; García, & Caballero, 2019; Santos, & Osório, 2019).

Este nuevo reto requiere que las acciones formativas del ámbito digital dirigidas al profesorado se enfoquen en el uso educativo de la tecnología (Prendes *et al.*, 2018). Asimismo, precisa de prácticas innovadoras en la formación inicial para vencer las resistencias hacia el uso de la tecnología en educación (Brazuelo *et al.*, 2017) y ofrecer herramientas al futuro profesorado para que pueda responder a los desafíos del siglo XXI integrando el pensamiento computacional y la programación en el aula de manera natural (Valverde *et al.*, 2015).

1.1. El pensamiento computacional en el ámbito educativo

“El pensamiento computacional implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, basándose en los conceptos fundamentales de la informática, [...] una habilidad fundamental para todos” (Wing, 2006, p. 33). Por ello, actualmente se considera básico para desenvolvernosen nuestras acciones cotidianas (González *et al.*, 2018), independientemente de las características disciplinarias.

Son muchas las voces a favor de su potencial educativo (Moreno *et al.*, 2019) ya que, la resolución de problemas reales y cotidianos requiere de un gran grupo de habilidades y actitudes transversales (González, 2019). Para esta investigación hemos optado por las indicadas por la Sociedad Internacional de Tecnología en Educación (ISTE) y la Asociación de Maestros de Ciencias de la Computación (CSTA), por ser organizaciones de reconocido prestigio internacional al servicio del profesorado interesado en el uso de la tecnología en la educación (ISTE, & CSTA, 2011, p. 1):

Habilidades:

- Formulación de problemas que nos permitan usar un ordenador y otras herramientas para ayudar a resolverlos.
- Organización y análisis lógico de datos.
- Representación de datos a través de abstracciones tales como modelos y simulaciones.
- Automatización de soluciones a través del pensamiento algorítmico (una serie de pasos ordenados).
- Identificación, análisis e implementación de las posibles soluciones con el objetivo de lograr la combinación más eficiente y efectiva de pasos y recursos.
- Generalización y transferencia de este proceso de resolución del problema a una amplia variedad de problemas.

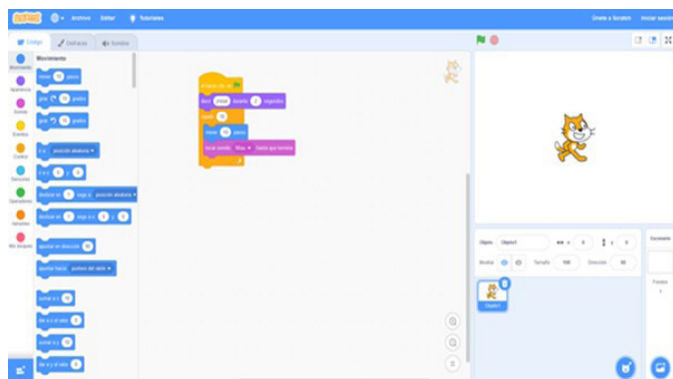
Actitudes:

- Confianza en el manejo de la complejidad.
- Persistencia en el trabajo con problemas difíciles.
- Tolerancia a la ambigüedad.
- Capacidad de lidiar con problemas abiertos.
- Capacidad de comunicarse y trabajar con otros para lograr un objetivo o solución común.

Otros trabajos (Corradini *et al.*, 2017; Csizmadia *et al.*, 2015; Zapata, 2019), también añaden a todas ellas:

- Descomposición del problema.
- Metacognición.
- Reconocimiento de patrones.
- Uso de algoritmos para definir los pasos.
- Creatividad.
- Experimentación.
- Realización de simulaciones.
- Razonamiento lógico.

FIGURA 1. Pantalla de programación de Scratch.
Fuente: elaboración propia en <https://scratch.mit.edu/>



- Técnicas de reflexión.
- Evaluación y corrección.
- Recursividad.

Es evidente que las habilidades propias del pensamiento computacional no son pocas y en ellas se apoyan las voces que destacan el potencial educativo del pensamiento computacional para el empoderamiento de las futuras generaciones (González, 2019). Por ello, la integración del pensamiento computacional en el aprendizaje formal e informal ha emergido con fuerza en los últimos años (García *et al.*, 2019).

Es por ello que “el desarrollo del pensamiento computacional a través de actividades de programación es una temática ampliamente estudiada en contextos educativos” (Monjolat, 2019, p. 5) y actualmente “existe un movimiento global que apuesta por la enseñanza de la programación informática y el pensamiento computacional” (Arranz, & Pérez, 2017, p. 26). Se defiende que ambas áreas son relevantes en entornos escolares y desde edades tempranas (González, 2019) y la programación ya forma parte de planes de estudios de Educación Primaria y Educación Secundaria en muchos países (Román, 2016).

1.2. La programación en el aula

El pensamiento computacional y la programación no son lo mismo, pero están estrechamente relacionados (Bers, 2018; INTEF, 2017) puesto que programar favorece tareas cognitivas implicadas en la formulación y resolución de problemas (Caballero, & García, 2020; Delval, 1986; Durak, & Saritepeci, 2018).

Al resolver desafíos, diseñar, explorar, razonar, crear, interactuar y producir digitalmente, el alumnado se convierte en protagonista de su propio aprendizaje (Moreno, & Robles, 2016) mientras trabaja conceptos de la código-alfabetización (algoritmo, secuencia, eventos, condicionales y repeticiones) y múltiples contenidos de manera transversal (García, & Caballero, 2019; Santos, & Osório, 2019).

1.2.1. Scratch

Scratch es un lenguaje visual de programación por bloques de diferentes colores creado por el grupo *Lifelong Kindergarten* del Instituto Tecnológico de Massachusetts (MIT, s. f.). Permite iniciarse en el mundo de la programación y que personas con diferentes intereses y/o estilos de aprendizaje, puedan crear sus propios proyectos: narraciones interactivas, actividades animadas, juegos y/o simulaciones, de forma atractiva, intuitiva y lúdica (INTEF, 2017; Resnick, & Brennan, 2011).

Se presenta a modo de “piezas de puzle” que se deben arrastrar, soltar y unir de manera lógica para crear el conjunto de instrucciones o *script* encargado de controlar las acciones que deben ejecutar las escenas y/o los objetos. Esto quiere decir que el código está predefinido. La disposición de dichos bloques se divide en diferentes categorías y colores: movimiento (azul oscuro), apariencia (morado), sonido (magenta), eventos (amarillo), control (naranja claro), sensores (azul claro), operadores (verde), variables (naranja fuerte) y las extensiones añadidas posteriormente por la persona usuaria (figura 1).

Scratch fomenta el aprendizaje interdisciplinar, el desarrollo intelectual, la experimentación, la manipulación práctica, la abstracción, la creatividad, la autoestima (Hervás *et al.*, 2018), la autonomía en el aprendizaje, la toma de decisiones y la reflexión (Gómez, & Williamson, 2018). Sin duda, la programación y *Scratch* ofrecen estrategias didácticas para las aulas (Hervás *et al.*, 2018).

1.3. Orientar la formación inicial del futuro profesorado de Educación Primaria hacia las competencias del siglo XXI

Uno de los pilares y, a su vez, uno de los principales desafíos en el desarrollo del pensamiento computacional y las competencias para la resolución de problemas en el ámbito educativo es la formación inicial del futuro profesorado (Castañeda *et al.*, 2018; Monjolat, 2019; Ordoñez *et al.*, 2013). Se debe realizar una reflexión crítica en torno al papel de la tecnología en el desarrollo humano (Castañeda *et al.*, 2018) e interiorizar las posibilidades y los beneficios de trabajar la programación en el aula.

De esta manera, el futuro profesorado podrá asumir su responsabilidad profesional como parte de la sociedad digital y promover procesos de enseñanza de calidad contextualizados y significativos orientados al desarrollo de la competencia digital (Cela *et al.*, 2017; Monjolat, 2019).

2. DISEÑO Y METODOLOGÍA

Este estudio pretende recoger la reflexión y valoración del alumnado de Grado de Educación Primaria (EP) de la Facultad de Educación y Deporte de la Universidad del País Vasco (UPV/EHU) sobre el desarrollo de competencias (habilidades y actitudes del pensamiento computacional) y las posibilidades educativas que ofrecen la programación y *Scratch* con el fin de poner en valor dichos ámbitos en su futura práctica docente. Asimismo, se analizan también los conocimientos previos sobre programación y los sentimientos referentes a la experiencia de aprendizaje llevada a cabo en la facultad.

El trabajo realizado se enmarca en un enfoque de tipo fenomenológico (Salgado, 2007), dado que trata de identificar y describir los fenómenos desde el punto de vista de cada participante respecto a su experiencia de aprendizaje con *Scratch*. Consta de dos fases, una primera fase de experimentación y una segunda fase de reflexión.

2.1. Participantes

El colectivo participante tanto en la fase de experimentación como en el cuestionario de reflexión ha sido el alumnado de Grado de EP de la Facultad de Educación y Deporte de la UPV/EHU y, por tanto, futuro profesorado responsable de llevar estas innovaciones al aula. En total han participado 119 personas, 62 chicas y 57 chicos. Se trata de una muestra intencional, ya que la única condición para participar en el estudio fue estar matriculado durante el curso académico 2019/20 en la asignatura TIC, que se imparte en el segundo curso del Grado.

2.2. Instrumento de recogida de información

El instrumento utilizado para la recogida de información ha sido un cuestionario, creado *ad hoc* para la experiencia y validado según juicio de expertos. Consta de cuatro preguntas, tres de ellas abiertas (preguntas 1, 3 y 4) y una escala Likert cerrada (pregunta 2). Las preguntas abiertas fueron las siguientes:

- ¿Qué es *Scratch*? Defínelo con tus propias palabras, como lo harías a tus amistades.
- ¿Crees que, en un futuro, como docente, vas a utilizar *Scratch* en el aula? ¿Por qué? ¿Para qué?
- ¿Qué tipo de conocimientos previos tenías sobre *Scratch* antes de trabajarlo en las sesiones? ¿Cómo te has sentido?

Por su parte, el objetivo de la escala Likert ha sido recoger las percepciones y/o valoraciones sobre el grado en que *Scratch* desarrolla las habilidades y actitudes del pensamiento computacional detalladas por ISTE y CSTA (2011). Los niveles de la escala fueron: Nada; Suficiente; Bien; Muy bien; Excelente.

2.3. Análisis de datos

El análisis de la información cualitativa se ha realizado a través del programa Nvivo11 Plus, organizando las respuestas a través de un sistema categorial (Tabla 1) de carácter deductivo a partir de la información emergente en las respuestas recogidas.

TABLA 1. Herramienta de análisis: Sistema categorial

Dimensiones	Indicadores nivel I	Indicadores nivel II
Qué es <i>Scratch</i>	Aplicación informática	
	Animación de imágenes	
	Creación de juegos, vídeos...	Creación de videojuegos
		Componentes de <i>Scratch</i>
	Utilización en educación	Trabajar contenidos
		Forma diferente de trabajar
		Trabajar competencias
	Programación	Tipo de contenido que se programa
		Características del programa
		Trabajar la resolución de problemas
Expectativas de aplicación en futuro profesional	Pensamiento computacional	
	SI	Aplicación didáctica en el aula
		Integración TIC en el aula
		Desarrollo de otras capacidades
	NO	Utilidad escasa
		Dificultad
	Tal vez	Inseguridad ante dificultad
	Necesidad de formación	
Conocimientos previos		Dependiendo de características del aula/grupo
	SI	Trayectoria académica
		Otros
Sentimientos ante la experiencia	NO	
	Experiencia positiva	
	Experiencia negativa	
	Experiencia mixta	Mejora ante el conocimiento
		Agobio ante bloqueos

Con los datos cuantitativos recogidos, en cambio, se han realizado operaciones estadísticas básicas (media) en una hoja de cálculo (Excel). Para facilitar la lectura, se han omitido los decimales en los porcentajes de los resultados, pudiendo crear esto un error de entre el 1 y 2 %.

2.4. Procedimiento

El presente estudio fue realizado en otoño de 2019 en la Facultad de Educación y Deporte de la UPV/ EHU y, como se ha adelantado, constó de dos fases: experimentación con *Scratch* y cuestionario sobre percepciones y sensaciones.

En la primera fase, el alumnado participó en un módulo de cuatro sesiones dedicado al pensamiento computacional, la programación y *Scratch*. Primeramente, se explicaron las bases del pensamiento computacional y, a continuación, se presentó *Scratch* como herramienta para trabajar el pensamiento computacional en el aula de EP. En dicha presentación, se indicó que *Scratch* es un lenguaje de programación visual por bloques que se clasifican en diferentes categorías, que deben ser arrastrados y que deben unirse para crear conjuntos de acciones. A partir de ahí, se plantearon una serie de retos (10) a abordar de manera inductiva e individual haciendo uso de *Scratch* 3.0. Los retos se presentaron en orden ascendente, según su nivel de complejidad.

Una vez finalizada la fase de experimentación, el alumnado respondió de manera individual un cuestionario *online* dirigido a favorecer la reflexión personal sobre las posibilidades educativas que ofrecen la programación y *Scratch* y los sentimientos referentes a la experiencia de aprendizaje.

3. RESULTADOS

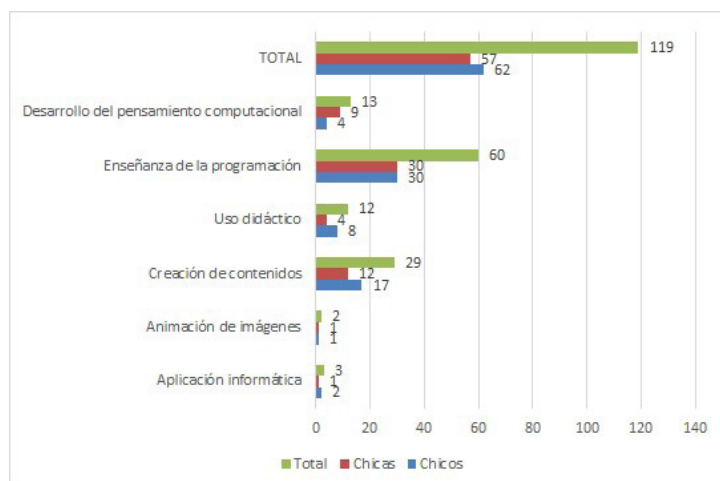
3.1. ¿Qué es *Scratch*?

En general, las ideas que tiene el futuro profesorado en formación sobre qué es *Scratch* no son muy precisas y hacen hincapié en aspectos muy diferentes. Las más simples se limitan a indicar que *Scratch* es una aplicación

online. Otras, en cambio, ofrecen una definición mucho más compleja enumerando características del lenguaje de programación y añadiendo que favorece el desarrollo del pensamiento computacional. En la figura 2 se muestra una síntesis de las respuestas recogidas en función del sexo.

Solamente 3 personas se limitan a responder que *Scratch* es una plataforma informática y otras 2 mencionan que es un programa de animación de imágenes. Estas definiciones imprecisas apenas representan un 4 % del total.

FIGURA 2. Qué es *Scratch*: número de respuestas, por categorías



El resto de respuestas (96%) basan la definición de *Scratch* en sus funciones o posibles usos: creación de contenidos (videojuegos, historias o conversaciones, por ejemplo) (24%), uso didáctico (trabajar competencias y contenidos curriculares de manera innovadora y dinámica, por ejemplo) (10%), enseñar a programar (de manera visual, por bloques, de fácil acceso, aprendizaje intuitivo y divertido, por ejemplo) (50%) y trabajar el pensamiento computacional (11%).

En este sentido, 29 personas (17 chicas y 12 chicos) han resaltado la utilidad que tiene *Scratch* para la creación de contenidos como videojuegos, historias o conversaciones de manera sencilla:

“Es una aplicación que se puede utilizar *online*. Se pueden crear proyectos de manera sencilla, estos proyectos son parecidos a los videojuegos y da la opción de crearlos en función de nuestros intereses” (creación de contenido).

Algunas de esas respuestas hacen referencia directa a los componentes de *Scratch* que facilitan la creación de contenidos como escenas, personajes, acciones y sonidos. Destacan también la posibilidad de interacción que ofrece, pero sin tener en cuenta el posible uso educativo:

“Hasta ahora no lo conocía. A través de esta aplicación se pueden elaborar “vídeos” para dar explicaciones o juegos. En estos proyectos los espectadores pueden participar. Se pueden utilizar personajes, escenas y sonidos diferentes” (creación de contenido).

A su vez, otras 12 personas (8 chicas y 4 chicos) van un poco más allá y en sus respuestas, además de la posibilidad de crear contenido, hacen referencia a las diferentes opciones que ofrece *Scratch* para su utilización en educación. Entre otras, destacan la posibilidad de trabajar contenidos académicos relacionados con distintas materias.

“Un recurso o herramienta mediante la que, a través de la programación, se pueden transmitir información, explicaciones y también aprender y adquirir conocimientos o competencias diversas” (uso didáctico).

“Una herramienta que sirve para trabajar cualquier contenido de forma dinámica” (uso didáctico).

Como era de esperar, el alumnado participante relaciona *Scratch* con la programación y es que 60 personas (30 chicas y 30 chicos) han mencionado la enseñanza o aprendizaje de la programación a la hora de describir *Scratch*.

“Una aplicación para trabajar la programación. Se trabaja de modo simple y visual y esto convierte a *Scratch* en una herramienta interesante y divertida” (enseñanza de la programación).

“Es un *software* apropiado para empezar a programar, ya que las órdenes son muy intuitivas y se montan en forma de puzles” (enseñanza de la programación).

Finalmente, 13 personas (4 chicas y 9 chicos) han elaborado definiciones más completas y, además de muchos de los aspectos anteriores, también han recogido la idoneidad de *Scratch* para trabajar la resolución de problemas y, así, el desarrollo del pensamiento computacional:

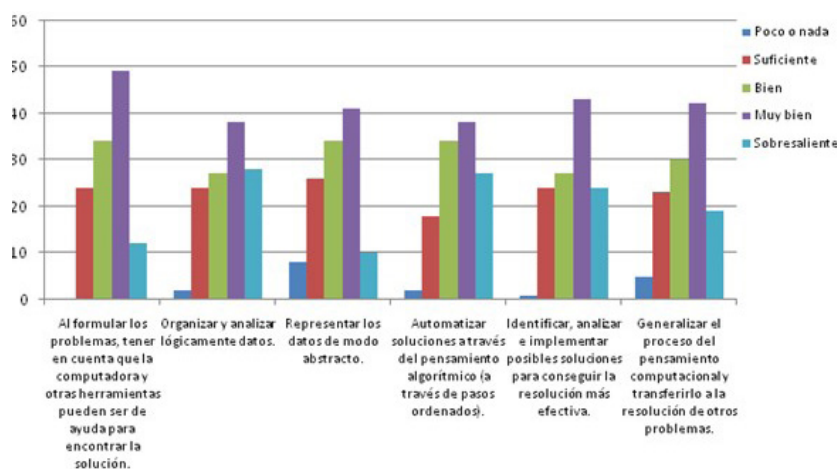
“Puede ser un proyecto para trabajar el pensamiento computacional de niños y niñas. Se pueden utilizar diferentes figuras u personajes y crear pequeñas series y se puede aprovechar la interacción con niñas y niños para que el aprendizaje sea lo más significativo posible” (desarrollo del pensamiento computacional).

“*Scratch* es un recurso tecnológico que nos permite trabajar el Pensamiento Computacional, es decir, la base para resolver los problemas que nos encontramos en la vida cotidiana” (desarrollo del pensamiento computacional).

3.2. ¿En qué grado crees que *Scratch* desarrolla las habilidades del pensamiento computacional?

Como se aprecia en la figura 3, el alumnado participante, basándose en su propia experiencia de aprendizaje, considera que *Scratch* permite desarrollar muy bien la mayoría de las habilidades relacionadas con el pensamiento computacional, destacando la formulación y resolución de problemas. También valoran como sobresaliente el potencial de *Scratch* para desarrollar las habilidades de gestión de datos y automatización de soluciones. Cabe indicar que es insignificante el número de respuestas en las que se considera que no favorece el desarrollo de las diferentes habilidades relacionadas con el pensamiento computacional.

FIGURA 3. Percepción en torno al desarrollo de las habilidades del pensamiento computacional a través de *Scratch*



Concretamente, el 41% del alumnado encuestado (49 personas) considera que, tras aprender a programar con *Scratch*, será muy probable, (grado muy bien), que el alumnado de primaria, al formular los problemas, tenga en cuenta que el ordenador y otras herramientas pueden ser de ayuda para encontrar la solución. Un 10% (12 personas) ha respondido que el alumnado lo hará de forma “sobresaliente”. Es de reseñar que ninguna persona encuestada ha respondido en grado de “poco o nada”.

En cuanto a la habilidad de organizar y analizar datos de forma lógica, el 32% (38 personas) entiende que el alumnado de EP puede desarrollar esta habilidad “muy bien” haciendo uso de *Scratch* y el 23% (28 personas) prevé su desarrollo de “sobresaliente”. En cuanto a la habilidad de representar datos de modo abstracto, el 34% del total (41 personas) ha respondido que el alumnado puede desarrollar dicha habilidad “muy bien”, mientras aprende a programar con *Scratch* y apenas algo más del 8% (10 personas) opina que lo pueden hacer de manera “sobresaliente”.

En referencia a la habilidad de automatizar soluciones a través del pensamiento algorítmico (a través de pasos ordenados), el 32% del total (38 personas encuestadas) entiende que es posible desarrollar “muy bien” esta habilidad gracias a la aplicación de *Scratch*; y más de un 22% (27 personas) considera que se puede desarrollar de manera “sobresaliente”.

Por lo que respecta a la quinta habilidad, identificar, analizar e implementar posibles soluciones para conseguir la resolución más efectiva, la mayoría de las personas encuestadas (el 36% del total, es decir, 43 personas) entiende que el alumnado puede desarrollar esta habilidad “muy bien” o de manera “sobresaliente” tras el uso de *Scratch* (20%, 24 personas).

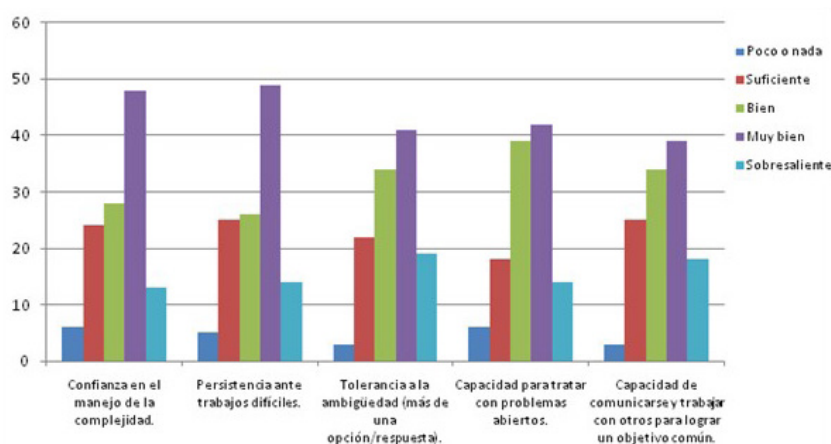
En cuanto a la habilidad de generalizar el proceso del pensamiento computacional y transferirlo a la resolución de otros problemas, la mayoría de las personas encuestadas, concretamente el 35% del total (42 personas), entiende que, tras el manejo de *Scratch*, el alumnado habrá desarrollado “muy bien” esta habilidad y el 16% (19 personas) de manera “sobresaliente”.

Como se puede observar, salvo en las habilidades de representar datos de modo abstracto y de automatizar soluciones, el porcentaje de alumnado que considera que las habilidades del pensamiento computacional se pueden desarrollar muy bien o de manera sobresaliente, supera el 50%.

3.3. ¿En qué grado crees que Scratch desarrolla las actitudes del pensamiento computacional?

Como se refleja en la figura 4, el alumnado participante considera que *Scratch* es un muy buen medio para desarrollar todas las actitudes del pensamiento computacional, especialmente la confianza ante la complejidad y la persistencia ante trabajos difíciles. Creen, además, que la tolerancia a la ambigüedad y la capacidad comunicativa y colaborativa pueden ser desarrolladas de manera sobresaliente mientras se aprende a programar con *Scratch*. Son muy pocas personas las que opinan que *Scratch* desarrolla poco o nada las actitudes del pensamiento computacional.

FIGURA 4. Percepción en torno al desarrollo de las actitudes del pensamiento computacional a través de Scratch



La actitud de tener confianza en el manejo de la complejidad ha sido la que mayor porcentaje de unanimidad ha conseguido entre las personas encuestadas. Así, el 43% (51 personas) estima que la respuesta del alumnado al desarrollo de esta actitud puede ser “muy buena” y el 11% (13 personas) la ha estimado como “sobresaliente”.

Persistencia ante trabajos difíciles ha sido la segunda actitud más destacada. El 41% (49 personas), estima que el nivel de desarrollo de esta actitud en el alumnado tras el manejo de *Scratch* puede ser “muy bueno” y el 12% (14 personas) “sobresaliente”.

Siguiendo con la tercera de las actitudes planteadas, tolerancia hacia la ambigüedad (más de una opción/respuesta), algo más del 34% (41 personas), entiende que el grado de desarrollo de esta actitud por parte del alumnado de EP tras el uso de *Scratch* puede ser “muy bueno”, el 16% (19 personas) “sobresaliente”.

Con respecto a la capacidad para tratar con problemas abiertos, el mayor porcentaje de personas encuestadas, un 35% (42 personas) valora que el desarrollo por parte del alumnado de esta actitud puede ser “muy bueno” gracias al uso de *Scratch* en el aula y el 12% (14 personas) lo estima como “sobresaliente”.

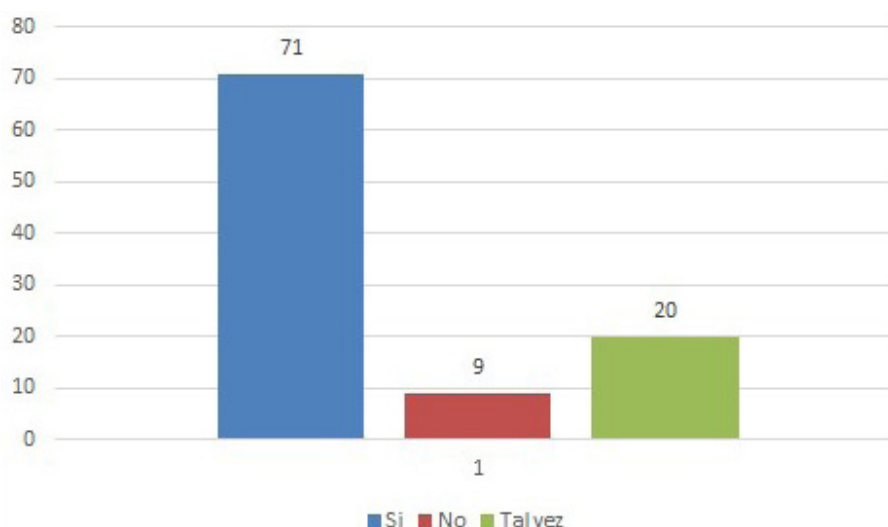
La última actitud planteada ha sido la capacidad de comunicarse y trabajar con otros para lograr un objetivo común. La mayoría de las personas encuestadas, el 32% (39 personas) entiende que, tras utilizar *Scratch*, el grado de desarrollo de esta actitud puede ser “muy bueno” y el 15% (18 personas) “sobresaliente”.

En resumen, más del 50% del alumnado considera que, a través de la utilización de *Scratch* se pueden desarrollar de manera sobresaliente o muy bien las actitudes relacionadas con el pensamiento computacional, a excepción de la capacidad para trabajar con problemas abiertos y la capacidad de comunicarse y trabajar con otras personas.

3.4. ¿Crees que, en un futuro, como docente, vas a utilizar *Scratch* en el aula? ¿Por qué? ¿Para qué?

En términos generales, la percepción sobre la utilidad pedagógica y/o didáctica de *Scratch* en el aula de EP es buena. La gran mayoría del alumnado encuestado indica tener la intención de incluir la programación en sus prácticas de enseñanza-aprendizaje como futuro docente. Aquellos y aquellas que no lo tienen claro o que expresan su opinión a modo de negativa, son minoría (figura 5).

FIGURA 5. Intención de incluir la programación en sus prácticas de enseñanza-aprendizaje como futuro docente



El 71% del alumnado participante confirma con certeza que utilizará *Scratch* en el aula como docente, integrando su uso en diversos procesos educativos. Entre el alumnado universitario que sí ve *Scratch* como una opción, se identifican diferentes motivos y motivaciones.

En el 66% de los casos, el alumnado considera que *Scratch* puede ser útil para el proceso de aprendizaje, tanto para trabajar nuevos contenidos como para profundizar y evaluar lo que ya se ha trabajado.

“Sí, creo que lo utilizaré. Puede ser útil para interiorizar conceptos de verdad, ya que para crear un *Scratch* sobre un concepto primero hay que dominarlo”.

Cabe destacar que, en más de la mitad de los casos, al igual que en la descripción, añaden que no sólo sirve para trabajar los contenidos del aula, sino que también es una manera más divertida, motivadora, dinámica y atrayente de hacerlo.

“Sí, porque me parece una propuesta interesante y atrayente para trabajar los conceptos, y una oportunidad excepcional para que la motivación del alumnado aumente”.

“Sí, porque mediante *Scratch* puedes trabajar muchos objetivos y conceptos de manera atractiva. Puedes relacionarlo con cualquier asignatura y aprender el contenido creando un juego”.

En el 32% de los casos también destacan el potencial de *Scratch* como herramienta para acercarse a la tecnología y al pensamiento computacional. En algunas respuestas (13 personas) comentan la oportunidad que ofrece de estrechar lazos con las TIC, mejorar en la competencia digital y, así, incluirlas en los procesos de aprendizaje. Unas pocas respuestas (5 personas) hacen mención a la oportunidad que ofrece *Scratch* para aprender a programar. Y 10 personas las incluyen directamente en el concepto de pensamiento computacional, haciendo referencia a cómo éste puede trabajarse mediante este lenguaje de programación.

Asimismo, un 13% de las respuestas obtenidas también destaca *Scratch* como un recurso útil para trabajar otros aspectos como el pensamiento crítico (2 personas), resolución de problemas (4 personas), creatividad (4 personas), participación (2 personas) y/o trabajo en equipo (2 personas).

En dirección contraria, el 9% del alumnado no considera una opción utilizar *Scratch* en su futuro como docente ya que consideran muy difícil su uso (3 personas) o no lo consideran útil o conveniente para el alumnado de EP (4 personas).

El otro 20% no sabe si lo utilizará. En el caso del alumnado que no lo tiene claro, algunos y algunas comentan (7 personas) qué les parece una herramienta interesante y útil, pero que, a su vez, es difícil y compleja. Otras menciones (8 personas) van en la misma línea, y, a pesar de valorar positivamente la herramienta, dicen que no saben si la utilizarán. En 5 de los casos relacionan su utilidad a las características y edad del grupo o aula. Y un último grupo (3 personas), aclara que antes necesitaría formarse más.

3.5. ¿Qué tipo de conocimientos previos tenías sobre *Scratch* antes de trabajarlo en las sesiones? ¿Cómo te has sentido?

El 45% del alumnado ha oído hablar o, incluso, ha llegado a utilizar *Scratch* con anterioridad a la etapa universitaria. Sin duda, los conocimientos previos ayudan a que la experiencia sea mejor y, por ello, el 66% del alumnado que ha trabajado la programación en etapas anteriores ha valorado positivamente el proceso llevado a cabo, resaltando emociones positivas como la motivación por poder explorar recursos útiles para su futura profesión.

“Me ha parecido interesante, entretenido y útil. En las sesiones he estado a gusto y la verdad es que las clases se me pasaban volando haciendo los retos”.

Por lo contrario, el 55% del alumnado dice no conocer ni haber utilizado *Scratch* anteriormente. En estos casos, las emociones y valoraciones son muy variadas y ambiguas. En el lado positivo se subrayan la satisfacción de ir adquiriendo conocimientos y mejora personal. Al lado negativo, en cambio, se le atribuyen emociones como frustración o inseguridad referidas a las dificultades para entender el funcionamiento de *Scratch* y resolver los retos.

“Me he sentido contento, a pesar de que a veces he estado perdido y no sabía cómo seguir. Cuando consigues avanzar la sensación es increíblemente buena”.

“He sentido falta de confianza al trabajar con *Scratch*”.

4. DISCUSIÓN Y CONCLUSIONES

Al finalizar el proceso de experimentación inductiva, hemos encontrado diferentes niveles de conocimiento en cuanto a lo que es *Scratch*. Se puede hablar de un *continuum* que va desde las respuestas más simples que se limitan a definir *Scratch* como una mera aplicación informática, a las respuestas más complejas que lo relacionan con el pensamiento computacional.

En ese *continuum*, la mayor parte del alumnado participante subraya especialmente las características del mismo: que es un lenguaje de programación visual, su principal función, es decir la creación de contenidos, y su potencial educativo para trabajar contenidos y competencias de las distintas áreas del currículum.

Así, las definiciones de los futuros docentes recogen las principales características técnicas del lenguaje de programación (INTEF, 2017) y, a su vez, coinciden con la sencillez y el atractivo para iniciarse en la programación descritos por Resnick y Brennan (2011). También señalan el aprendizaje interdisciplinar que detallan Hervás *et al.* (2018) e, incluso, el desarrollo del pensamiento computacional (Bers, 2018).

Al reflexionar más en profundidad sobre el potencial de *Scratch*, después de la fase de experimentación con la herramienta, el alumnado participante considera que la programación con *Scratch* contribuye a desarrollar las habilidades y actitudes relacionadas con el pensamiento computacional en el alumnado de EP.

Destacan dos habilidades que el alumnado de EP puede desarrollar de manera sobresaliente gracias a la enseñanza de la programación: organizar y analizar lógicamente los datos (23%) y automatizar soluciones a través del pensamiento algorítmico (a través de pasos ordenados) (22%). Asimismo, valoran que las actitudes que más puede desarrollar el alumnado de EP son la confianza en el manejo de la complejidad (43%) y la persistencia ante trabajos difíciles (41%).

En definitiva, las respuestas recogidas en el segundo apartado reflexivo de la investigación indican una valoración positiva de *Scratch* como herramienta para desarrollar las habilidades y actitudes atribuidas al pensamiento computacional en la etapa de EP por parte del futuro profesorado (ISTE, & CSTA, 2011).

Por ello, el 71% confirma que, en un futuro, utilizará *Scratch* para el desarrollo de la competencia digital y el pensamiento computacional para responder a los objetivos didácticos de una manera dinámica y atractiva. Se ha constatado que previamente a la experiencia con *Scratch*, predominaba el desconocimiento del lenguaje de programación ante las personas que ya habían trabajado con el mismo. En los casos de conocimiento previo, la mayoría hace referencia al itinerario académico personal. Por lo tanto, se puede entender que, poco a poco, las instituciones educativas están asumiendo su responsabilidad profesional en la actual sociedad digital (Cela *et al.*, 2017). Sin embargo, no debemos olvidar que aún existe un gran porcentaje que no ha trabajado la programación en su trayectoria académica (Bustillo, 2015). Es decir, aunque haya evidencias de avances, la programación educativa sigue siendo un campo en construcción (Monjelat, 2019).

En lo referente a la experiencia de aprendizaje y los sentimientos que ésta ha generado en el alumnado participante, la valoración general es positiva. Una parte hace referencia directa al proceso de mejora en habilidades técnicas (Adell *et al.*, 2017) y a la resolución de problemas (Durak, & Saritepeci, 2018), lo que puede incidir en la actitud positiva hacia el aprendizaje de las tecnologías (Hall *et al.*, 2014; Krumsvik, 2012). Sin embargo, en las valoraciones negativas, destacan sentimientos como la frustración o la inseguridad, refiriéndose, especialmente, a las dificultades que han tenido para entender el funcionamiento de *Scratch* y resolver los retos.

En síntesis, se concluye que la experimentación basada en *Scratch* ha permitido al profesorado en formación entender su obligación social de incrementar las prácticas dirigidas a potenciar el desarrollo del pensamiento computacional en el aula y sus competencias (Prieto, & Berretta, 2014). Por lo tanto, es importante seguir construyendo, en las aulas universitarias, modelos de integración de la enseñanza de la programación, desde una mirada educativa, para garantizar el papel activo, creativo y reflexivo de las personas en el actual mundo digitalizado del siglo XXI.

5. REFERENCIAS

- Adell, J., Esteve-Mon, F. M., Llopis, M. Á., & Valdeolivas, M. G. (2017). El Pensamiento Computacional en la formación inicial del profesorado de Infantil y Primaria. En V. Abella, V. Ausin, & V. Delgado (eds.), *XXV Jornadas Universitarias de Tecnología Educativa: Aulas y Tecnología educativa en evolución* (pp. 1-7). Red Universitaria de Tecnología Educativa.
- Alsina, A., & Acosta, Y. (2018). Iniciación al álgebra en Educación Infantil a través del pensamiento computacional: una experiencia sobre patrones con robots educativos programables. UNIÓN. *Revista iberoamericana de educación matemática*, 52, 218-235.
- Arranz, H., & Pérez, A. (2017). Evaluación del Pensamiento Computacional en Educación. *RiiTE, Revista Interuniversitaria de Investigación en Tecnología Educativa*, 3, 25-39. <http://dx.doi.org/10.6018/riite/2017/267411>
- Bender, W., Urrea, C., & Zapata, M. (2015). Presentación. *RED*, 46, 1-3.
- Bers, M. U. (2018). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge.
- Brazuelo, F., Gallego, D.J., & Cacheiro, M.L. (2017). Los docentes ante la integración educativa del teléfono móvil en el aula. *RED*, 52, 1-22. <http://dx.doi.org/10.6018/red/52/6>
- Bustillo, J. (2015). Formación del profesorado con scratch: análisis de la escasa incidencia en el aula. *Opción*, 31(1), 164-182.
- Caballero-González, Y. A., & García-Valcárcel, A. (2020). Fortaleciendo el pensamiento computacional y habilidades sociales mediante actividad de aprendizaje con robótica educativa en niveles escolares iniciales. *Píxel-BIT*, (58), 117-142. <https://doi.org/10.12795/pixelbit.75059>
- Carretero, S., Vuorikari, R., & Punie, Y. (2017). *DigComp 2.1: The Digital Competence Framework for Citizens with eight proficiency levels and examples of use*. Publications Office of the European Union.
- Castañeda, L., Esteve, F., & Adell, J. (2018). ¿Por qué es necesario repensar la competencia docente para el mundo digital? *RED*, 56(6), 1-20. <http://dx.doi.org/10.6018/red/56/6>
- Cela-Ranilla, J. M., Esteve, V., Esteve, F., González, J., & Gisbert-Cervera, M. (2017). El docente en la sociedad digital: una propuesta basada en la pedagogía transformativa y en la tecnología avanzada. *Profesorado*, 21(1), 403-422.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational Thinking. A guide for Teachers*. Computing At School. <https://community.computingschool.org.uk/resources/2324/single>
- Corradini, I., Lodi, M., & Nardelli, E. (2017). Conceptions and misconceptions about computational thinking among Italian primary school teachers. En J. Tenenber, D. Chinn, Judy Sheard y L. Malmi, *Proceedings of the 2017 ACM Conference on International Computing Education Research* (pp. 136-144). Association for Computing Machinery. <https://dl.acm.org/doi/proceedings/10.1145/3105726>
- Delval, J. (1986). *Niños y máquinas: Los ordenadores y la educación*. Alianza.
- Durak, H. Y., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*, 116, 191-202. <https://doi.org/10.1016/j.compedu.2017.09.004>
- García, A., & Caballero, Y. (2019). Robótica para desarrollar el pensamiento computacional en Educación Infantil. *Comunicar*, 59(2), 63-72. <https://doi.org/10.3916/C59-2019-06>

- Gamito, R., Aristizabal, P., & Vizcarra, M. T. (2018). Pre-school Education Degree students' prior knowledge and perception of digital competence. En J. Domenech, P. Merello, E. de la Poza, & D. Blazquez (eds.), *4th International Conference on Higher Education Advances* (pp. 1421-1428). Editorial Universitat Politècnica de València.
- Gómez, E., & Williamson, G. (2018). Autonomía y TIC en el aprendizaje de jóvenes y adultos. Pedagogía socio-crítica a través de talleres de Scratch. *Praxis Educativa*, 22(3) 71-82.
- González, C. S. (2019). Estado del arte en la enseñanza del pensamiento computacional y la programación en la etapa infantil. *Education in the Knowledge Society*, 20, 1-15. http://dx.doi.org/10.14201/eks2019_20_a17
- González, J., Estebanell, M., & Peracaula, M. (2018). ¿Robots o programación? El concepto de Pensamiento Computacional y los futuros maestros. *Education in the Knowledge Society*, 2(18), 29-45. <https://doi.org/10.14201/eks20181922945>
- Hall, R., Atkins, L., & Fraser, J. (2014). Defining a self-evaluation digital literacy framework for secondary educators: the DigiLit Leicester project. *Research in Learning Technology*, 22(1), 1-17. <https://doi.org/10.3402/rlt.v22.21440>
- Hervás, C., Ballesteros, C., & Corujo M. T. (2018). La robótica como estrategia didáctica para las aulas de educación primaria. *Revista Educativa Hekademos*, 24, 30-40.
- INTEF (2017). *El Pensamiento Computacional en la Enseñanza Obligatoria (Computhink). Implicaciones para la política y la práctica*. Ministerio de Educación, Cultura y Deporte.
- ISTE, & CSTA (2011). *Operational Definition of Computational Thinking for K-12 Education*. National Science Foundation. <https://bit.ly/31D0fbH>
- Krumsvik, R. J. (2012). Teacher educators' digital competence. *Scandinavian Journal of Educational Research*, 58(3), 269-280. <https://doi.org/10.1080/00313831.2012.726273>
- MIT (s. f.). *Media Lab*. <https://bit.ly/2ZvhyJc>
- Monjelat, N. (2019). Programación de tecnologías para la inclusión social con Scratch: Prácticas sobre el pensamiento computacional en la formación docente. *Educare*, 23(3), 1-25. <http://dx.doi.org/10.15359/ree.23-3.9>
- Moreno, J., & Robles, G. (2016). Code to learn with Scratch? A systematic literature review. En Institute of Electrical and Electronics Engineers, *Global Engineering Education Conference (EDUCON)* (pp. 150-156). IEE <http://dx.doi.org/10.1109/EDUCON.2016.7474546>
- Moreno, J., Robles, G., Román, M., & Rodríguez, J. D. (2019). Not the same: a text network analysis on computational thinking definitions to study its relationship with computer programming. *Revista Interuniversitaria de Investigación en Tecnología Educativa*, (7). <https://doi.org/10.6018/riite.397151EE>
- Ordoñez, M. A., González, M., & Ballesteros, M. Á. (2013). Alfabetización digital desde la práctica: una propuesta innovadora para el desarrollo de competencias digitales en adultos. En Jornadas de Innovación Docente, *III Jornadas de Innovación Docente. Innovación Educativa: respuesta en tiempos de incertidumbre*. Universidad de Sevilla, Facultad de Ciencias de la Educación.
- Prendes, M. P., Gutiérrez, I., & Martínez, F. (2018). Competencia digital: una necesidad del profesorado universitario en el siglo XXI. *RED*, (56), 1-22. <http://dx.doi.org/10.6018/red/56/7>
- Prieto, E., & Berretta, R. (2014). Digital Technology Teachers' Perceptions of Computer Science: It is not all about programming. En IEEE, *Frontiers in Education Conference* (pp. 1-5). IEEE.
- Recomendación del Parlamento Europeo y del Consejo, de 18 de diciembre de 2006, sobre las competencias clave para el aprendizaje permanente (2006/962/CE). *Diario Oficial de la Unión Europea*, L394, del 30 de diciembre del 2006, pp. 10-18. <https://cutt.ly/zFs055r>
- Resnick, M., & Brennan, K. (2011). *Computational Thinking Practices: April 2011 Webinar*. ScratchedED, Harvard University. <https://bit.ly/2NPhGxF>
- Román, M. (2016). *Códigoalfabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de programas* [Tesis doctoral, UNED]. e-spacio. <http://e-spacio.uned.es/fez/view/tesisuned:Educacion-Mroman>
- Santos, M., & Osório, A. (2019). Aprender a programar en educación infantil: análisis con la escala de participación. *Píxel.BIT*, (55), 133-156. <https://doi.org/10.12795/pixelbit.2019.i55.08>

- Valverde, J., Fernández, M. R., & Garrido, M. C. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *RED*, 46(3), 1-18. <https://doi.org/10.6018/red/46/3>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Zapata, M. (2019). Pensamiento computacional desenchufado. *Education in the Knowledge Society*, 20, 1-29. http://dx.doi.org/10.14201/eks2019_20_a18