



## La resolución de problemas basada en el método de Polya usando el pensamiento computacional y Scratch con estudiantes de Educación Secundaria

Álvaro Molina Ayuso, Natividad Adamuz Povedano y Rafael Bracho López

Universidad de Córdoba – España

### RESUMEN

La inclusión de la tecnología como recurso educativo supone un nuevo escenario de trabajo en el aula que requiere de un proceso de investigación empírica para valorar su efectividad. El presente trabajo consiste en un análisis de cómo el uso del pensamiento computacional con Scratch permite abordar el desarrollo de la competencia matemática en resolución de problemas, observando la implementación del método de Polya en la resolución de problemas aritméticos. El trabajo se ha desarrollado bajo un modelo metodológico mixto, realizando un estudio de series cronológicas con medidas de pretest sustitutivas. Esta información se ha completado con un análisis cualitativo sobre la percepción del propio alumnado sobre su proceso de aprendizaje. Para llevar a cabo este trabajo, los estudiantes han realizado unas actividades diseñadas con Scratch y un proyecto para crear un videojuego basado en la resolución de problemas aritméticos. Los resultados evidencian una implementación positiva del método y un uso eficaz del recurso empleado mostrando una mejoría en la competencia en resolución de problemas y un mejor desarrollo de la competencia lingüística: se observa una importante mejoría en el proceso de lectura y comprensión del enunciado del problema, proceso fundamental para llevar a cabo la resolución de un problema.

Palabras clave: Lenguajes de programación, informática educativa, resolución de problemas, Matemáticas en Educación Secundaria, pensamiento computacional.

### **Problem solving with Polya's technique using computational thinking and Scratch with Secondary School students**

### ABSTRACT

This article consists of an analysis of how the mathematical competence in problem solving, using the Polya's technique, can be developed using computational thinking with Scratch as a transversal resource, checking how students use Polya's technique while solving arithmetic problems. To know and internalize learning about this problem-solving heuristic, the student has worked previously this kind of thought process with computational thinking using the educational resource. The work has been put into practice using a mixed methodology, accomplishing a time-series experiment with non-substitutive pretest measures. This information has been completed with a qualitative measurement with the student's own perception about their learning process during the development of the empirical experience. To carry out the implementation of this experimental design, the students have done a series of tasks designed with Scratch and developing a project to create a videogame based on solving arithmetic problems. The outcomes show positive results using this methodology and a useful implementation of the computational thinking as a transversal resource in Mathematics, not only improving the mathematical competence but also the linguistic. An important improvement is observed in the reading and comprehension of the statement of the problem, essential process to carry out good resolution of a problematic situation faced by a student using a problem-solving technique.

Keywords: Programming Languages, Computer Science Education, problem solving, Secondary school Mathematics, computational thinking.

## 1. Introducción

La enseñanza de las matemáticas se ha enriquecido en los últimos años gracias al uso de las tecnologías para el aprendizaje y el conocimiento (TAC) adoptando nuevas metodologías más activas y participativas, permitiendo desarrollar un currículo determinado no solo por los contenidos de la asignatura. La incorporación de las tecnologías educativas abre, sin duda, un campo de nuevas posibilidades de innovación y mejora de los procesos formales de aprendizaje, lo cual debe ir acompañado de un importante proceso de investigación empírica que permita mejorar la calidad de la enseñanza (Onrubia, 2016). En concreto, se puede llevar a la práctica favoreciendo un desarrollo de las competencias matemáticas relacionado con la capacidad de estimación, el análisis y el tratamiento de la información, con la habilidad para modelar situaciones abiertas y para resolver problemas no tipificados (Recio, 2002).

En la actualidad, uno de los ejemplos más destacados de esta realidad es la inclusión del pensamiento computacional como herramienta de aprendizaje. En los primeros trabajos en este campo desarrollados por Seymour Papert (1980), se mostraba un punto de vista utópico de cómo la tecnología podría ayudar en el proceso de aprendizaje, idea que fue desarrollando posteriormente hasta la concepción de una completa alfabetización inicial en la infancia (1996). Hoy son muchos los recursos disponibles para trabajar estas habilidades, siendo Scratch uno de los más populares. Scratch es un lenguaje de programación visual diseñado por el MIT Media Lab para docentes y estudiantes, que ayuda a usar la programación para el desarrollo de estrategias (modulación, procesos iterativos, abstracción o descomposición) mejorando aspectos que no solo influyen en el desarrollo de las habilidades propias de un programador (Resnick et al., 2009). En la última década, el desarrollo del pensamiento computacional ha hecho que muchos autores y entidades dediquen sus esfuerzos a establecer un marco de referencia común para trabajar con este concepto. En 2010, la National Research Council establece un listado de las 20 destrezas y prácticas importantes que deben ser incluidas cuando se trabaja con el pensamiento computacional: abstracción en la resolución de problemas, descomposición, heurística de razonamiento, estrategias, y conocimientos de conceptos informáticos como paralelismo o recursividad, entre otros (NRC, 2010). Igualmente, la International Society for Technology in Education (ISTE) y la Computer Science Teacher Association (CSTA) ofrecen un marco de trabajo y un vocabulario común sobre el pensamiento computacional (ISTE, 2011).

Una definición ampliamente aceptada, es la establecida por Janet Wing (2011): «el pensamiento computacional describe una actividad mental para formular un problema y admitir una solución informática, la cual puede llevar a cabo una persona o una máquina o, de manera más general, la combinación de personas y máquinas» (p.20). El pensamiento computacional tendría que considerarse como una destreza para cualquier persona, no solo para informáticos, motivo por el cual debería incluirse como una parte más de las habilidades de análisis de todo estudiante (Wing, 2006). También es importante la visión del pensamiento computacional aportada por DiSessa (2000) en su libro *Changing Minds*, donde establece la posibilidad de que los ordenadores sean la base para una nueva forma de alfabetización con el poder de introducirse transversalmente en las asignaturas, en los contenidos y en las áreas de conocimiento. Lee y otros autores (2011) sugieren que el pensamiento computacional puede aplicarse en tres prácticas diferentes: modelado/simulación, robótica y diseño de juegos.

Todo este desarrollo ha impulsado un significativo aumento en la inclusión e integración curricular del pensamiento computacional en algunos países de Europa (Balanskat y Engelhardt, 2015), tanto en Educación Secundaria como en Primaria. En esta línea hay autores como Gander et al. (2013) y Nelson (2009) que nos advierten sobre la importancia de no quedarse atrás en la incorporación curricular del pensamiento computacional, refiriéndose a la necesidad de generar nuevos escenarios de aprendizaje que contribuyan al desarrollo de las competencias propias de una sociedad globalizada en la que el buen uso de las tecnologías de la información y el conocimiento ha de convertirse en un elemento esencial en la formación de los ciudadanos.

Es bien sabido que el verdadero aprendizaje se consigue cuando uno es partícipe y creador de su propia experiencia, por lo que debe tenerse la misma consideración con el uso de las tecnologías para el aprendizaje en el aula. El efecto de esta cultura de aprender haciendo es que la educación se aleja cada vez más del concepto clásico que la hace ser un producto cerrado, previsible, con una evaluación que se asemeja más al control de calidad de un proceso industrial y que no responde a las necesidades de nuestro tiempo (Romero-Frías y Magro Mazo, 2016). Estas estrategias de innovación educativas permiten definir nuevos entornos de aprendizaje personalizados, con el fin de facilitar un proceso autónomo y a la vez cooperativo, que contribuya el desarrollo de la competencia digital (Gutierrez-Esteban y Becerra Traver, 2014).

Gracias a la inclusión de este tipo de recursos educativos, el uso de la programación se convierte en una herramienta con la que los estudiantes pueden crear sus propios contenidos de manera sencilla. Muchos de los considerados como nativos tecnológicos nacidos con Google, Wikipedia y multitud de redes sociales, más que adquirir las destrezas y habilidades tecnológicas esperadas usando los recursos que tienen a su disposición, desarrollan ciertos comportamientos sociales, no siempre beneficiosos, que dan lugar a multitud de debates mostrando la necesidad de buscar nuevas alternativas educativas (Selwyn, 2009); es importante que el estudiante se desarrolle como creador de contenidos y no como mero consumidor tecnológico. La mayoría de los jóvenes saben enviar mensajes, jugar en plataformas online y buscar en Internet, pero estas habilidades no les convierten en buenos usuarios tecnológicos. Ellos interactúan con el mundo digital constantemente, pero pocos son capaces de crear sus propios juegos, animaciones o simulaciones; es como si supieran leer, pero no escribir (Resnick et al., 2009). Tal y como indica el eslogan del propio programa, “imagina, crea, comparte”, el uso de Scratch permite introducir un entorno de aprendizaje personalizado y abierto, ya que se trata de un software educativo de libre uso con una red de usuarios y docentes interconectados que pueden intercambiar recursos, creaciones o experiencias educativas gracias a su plataforma web, característica que lo diferencia de otros recursos también populares como Alice, Snap o Kodu. Así, los usuarios de Scratch tienen la oportunidad de involucrarse en una de las características más importantes que identifican a los movimientos FLOSS (Free/Libre Open Source Software): la contribución de cada usuario a la comunidad (Moreno-León, Robles, y Román-González, 2016). Este tipo de estrategias ayudan a despertar la motivación del alumnado, ofreciendo distintas alternativas de trabajo y dinamismo a las sesiones de aula, permitiendo descubrir y realizar de manera más personal todo el proceso de aprendizaje gracias al avance de las tecnologías para el aprendizaje y el conocimiento (Leris López y Sein-Echaluce Laclea, 2011). Todos estos aspectos, junto al actual acceso libre de hardware y software, hace que esté al alcance de cualquier estudiante la posibilidad de aprender haciendo, creando, desarrollando,

inventando y reinventando cosas en el ámbito de la programación y la robótica, entre otras áreas. Tal experiencia está ligada con la filosofía del Movimiento Maker, el cual tiene su influencia en el ámbito educativo al ser considerado un entorno pedagógico que permite a cualquier persona resolver sus propios problemas a través de un proceso de creación (Halverson y Sheridan, 2014).

El desarrollo de este trabajo parte del hecho de que el pensamiento computacional es un recurso que cada vez más docentes de distintas disciplinas incorporan en su programación de aula para el desarrollo curricular, por lo que es necesario establecer los beneficios de esta herramienta de aprendizaje dentro de un marco formal. Igualmente, las publicaciones científicas en este campo son cada vez más abundantes, siendo una demanda analizar el impacto que tiene el uso del pensamiento computacional en aplicaciones concretas. Del mismo modo que las revisiones bibliográficas evidencian los efectos de la inclusión del pensamiento computacional en el currículum a distintas edades, también dan a entender la necesidad de realizar más investigaciones de este tipo en los centros escolares (Moreno-Leon y Robles, 2016).

## 2. La resolución de problemas

Wing (2006) manifiesta cómo el pensamiento computacional puede utilizarse como una herramienta fundamental para desarrollar la competencia en resolución de problemas. Dentro del enfoque por competencias (Niss, 2002), la resolución de problemas se considera como una subcompetencia de la competencia matemática. Con ella se pretende desarrollar la capacidad de plantear y resolver problemas identificando y especificando distintos tipos, con planteamientos propios o dados por otros. El concepto de “problema” es algo muy relativo y no es propio solamente de una tarea matemática. Un problema aparece cuando un individuo tiene una meta pero no sabe cómo puede alcanzarla, es algo particular entre un individuo y la tarea que tiene que realizar (Schoenfeld, 1985). Se puede concebir el proceso de resolución de un problema desde un sentido más amplio que el que aporta la aplicación de las matemáticas. Para Mayer y Wittrock (2006) resolver un problema es un proceso cognitivo para alcanzar una meta cuando el método para la solución no es obvio para quien lo resuelve. Así, se le asocian cuatro características: cognitivo, al ser algo interno al individuo; es un proceso, ya que incluye una representación y manipulación del conocimiento; es dirigido, ya que el proceso cognitivo está guiado por la meta del individuo; y es personal porque todo lo anterior difiere para cada individuo. Por otra parte, Klieme (2004) añade a esta visión el carácter emocional en el proceso de resolución del problema, dándole importancia al contexto social y a la motivación. Sin embargo, la resolución del problema está principalmente determinada por la comprensión de la situación problemática, por el razonamiento usado para acercarse al problema y lo apropiado de la solución.

Profundizando en la importancia de la comprensión, Lesh y Zawojewski (2007) definen la resolución de problemas como un proceso en el que se da una interpretación matemática de una determinada situación, haciendo a su vez una importante distinción entre la perspectiva tradicional para la resolución de problemas y la resolución de problemas aplicados, relacionando esto último con destrezas que se consideran propias del pensamiento computacional: elaboración de patrones, recursividad, abstracción o descomposición.

La aplicación de la resolución de problemas aporta un sentido matemático al proceso de aprendizaje, pero con la inclusión del pensamiento computacional no es necesario ceñirse al desarrollo de esta disciplina; el proceso de razonamiento para la resolución de problemas se aborda, no solo desde otra perspectiva

diferente a la meramente matemática, sino que puede plantearse para trabajar en otro contexto o incluido en el desarrollo de otra área de conocimiento. La resolución de un problema lleva asociada un importante factor cognitivo: la capacidad de planificar, una habilidad fundamental para desarrollar acciones de gran importancia como pensar, ser críticos, tener criterio propio, decidir o valorar (Mayoral, Roca, Timoneda y Serra, 2015). Especialmente, cabe desatacar la importancia de desarrollar en nuestro alumnado un pensamiento crítico que les sirva para enfrentarse a problemas reales, persistentes y complejos desarrollando habilidades genéricas sin dejar a un lado el elemento clave para esta transformación: el conocimiento (Ross y Gautreaux, 2018).

La contextualización de un problema y los métodos de resolución son un campo muy amplio y diverso, por lo que este trabajo se ha focalizado en la implementación de una metodología de resolución de problemas basada en la heurística tradicional de Polya (1945), referente de las metodologías modernas de resolución de problemas. Este método se caracteriza por cuatro pasos, acompañado cada uno de una serie de cuestiones, para llevar a cabo el proceso de resolución de manera estructurada. Estos pasos son:

- Comprender el problema: ¿qué hay que calcular?, ¿cuáles son los datos?, ¿cuál es la condición?, ¿es posible cumplir esta condición? (conviene apoyarse en técnicas como dibujar una figura, usar una notación adecuada, o dividir en varios apartados).
- Concebir un plan: hay que buscar la relación entre los datos y lo que hay que calcular. ¿Has trabajado un problema similar? Si conoces un problema relacionado con este tema que se ha trabajado anteriormente, ¿puedes usarlo?, ¿puedes replantear el problema de manera diferente.
- Ejecutar el plan: llevar a cabo el plan que se ha trazado para resolver el problema, comprobando cada paso: ¿está claro que cada paso es correcto?, ¿puede comprobarse?
- Verificar el resultado: ¿es posible comprobar el resultado?, ¿es posible comprobar el razonamiento?, ¿se pueden obtener los resultados de manera diferente?, ¿se puede emplear el método utilizado en otros problemas?

En ocasiones, estas heurísticas son demasiado generales y no son igualmente aplicables para la resolución de cualquier problema. Cuando un profesor se encuentra con esta característica, recurre a enseñar distintas heurísticas ajustándose cada una al tipo de problema que hay que resolver. Esto hace que se pierda demasiado tiempo en aprender técnicas particulares que no van a usarse con la frecuencia necesaria para que el alumnado las asimile. Conviene evitar metodologías específicas y centrarse más en enseñar formas de afrontar un problema y conceptos que puedan ser necesarios de una forma más general (Schoenfeld, 1985). Posteriormente, Schoenfeld (2012) expone que el motivo del poco éxito que en algunas situaciones presenta la implementación de este tipo de métodos puede ser que resultan algo confusos para el estudiante, pudiendo estar utilizando en el proceso de resolución otras subestrategias de las cuales no es consciente. Por ello, conviene emplear otra denominación a la hora de trabajar esta heurística de resolución de problemas para intentar que los alumnos y alumnas la interioricen de manera más efectiva.

Utilizar una denominación alternativa ha sido un aspecto importante a la hora de diseñar el trabajo empírico con los estudiantes para la presente investigación, cuyo objetivo es analizar los resultados en la resolución de problemas tras incluir una metodología de resolución a través del pensamiento computacional. Pero con esta nueva metodología no parece útil emplear de-

nominaciones clásicas, es necesario abordar con otra perspectiva más completa el desarrollo de la competencia en resolución de problemas. Para poder desarrollar esta idea, se han diseñado una serie de actividades con el software educativo Scratch para trabajar el método de razonamiento para la resolución de problemas basado en la heurística tradicional de Polya, pero usando una denominación alternativa. El pensamiento computacional es la herramienta transversal de aprendizaje que permite abordar este proceso desde otra perspectiva, fuera del contexto meramente matemático.

### 3. Método

#### 3.1. Objetivos e hipótesis

La hipótesis inicial de este trabajo es que la inclusión curricular del pensamiento computacional favorece el desarrollo de la competencia matemáticas en resolución de problemas. Para poder contrastar dicho punto de partida, el desarrollo de esta experiencia empírica se ha realizado con el objetivo principal de analizar la mejora en dicha competencia trabajando la metodología de resolución con Scratch en el primer curso de Educación Secundaria Obligatoria. Como objetivos específicos se plantean aspectos tales que permitan al alumnado desarrollar destrezas asociadas de manera directa a dicha metodología:

1. Mejorar la capacidad de análisis y comprensión de los enunciados de problemas.
2. Desarrollar habilidades necesarias para la elaboración de estrategias de resolución de problemas.
3. Saber ejecutar de manera adecuada un plan de resolución de problemas previamente elaborado.
4. Desarrollar una capacidad crítica que lleve a comprobar las operaciones que realice en el proceso de resolución y promover la necesidad de hacer distintos planteamientos que ayuden a validar todo el proceso.

#### 3.2. Metodología

El proceso empírico de la investigación se ha llevado a cabo con un total de 18 estudiantes de 1º de Educación Secundaria con edades comprendidas entre los 11 y los 12 años. Se ha desarrollado en un Centro Público de Educación Secundaria Obligatoria, ubicado en un entorno socioeconómico y cultural medio, dentro del contexto de una población rural andaluza.

A la hora de analizar un proceso de aprendizaje, un análisis cuantitativo ofrece solo una generalización estadística, mientras que un proceso cualitativo aporta una visión analítica más contextualizada, que permite ampliar las conclusiones obtenidas (Onwuegbuzie y Collins, 2007). Por este motivo, en ciencias sociales en general, y en educación matemática en particular, es necesario emplear una metodología heterogénea para dar mayor significado a los datos obtenidos; la naturaleza de los problemas que se abordan requiere el uso de metodologías mixtas para contextualizar las actividades educativas, pudiendo al mismo tiempo aportar recomendaciones ampliamente generalizables (Castro y Godino, 2011). Los métodos mixtos se consideran como el tercer paradigma de la investigación, junto a la metodología cualitativa y cuantitativa (Johnson, Onwuegbuzie y Turner, 2007). Se pueden encontrar multitud de denominaciones para el uso de varios procesos en una investigación, lo cual sugiere que una metodología mixta no es algo limitado a mezclar varios métodos, sino que habría que entenderlo en un sentido más amplio (Johnson et al., 2007).

Para realizar el presente trabajo se ha diseñado un proceso cuasiexperimental, en el que los sujetos no han sido asignados al grupo de manera aleatoria. Se ha tomado un diseño de series cronológicas con un grupo experimental con medida de pretest sustitutiva, al no haber utilizado el mismo instrumento de medida para el postest (Campbell y Stanley, 1966), el cual permite analizar el efecto de un método didáctico innovador sobre el rendimiento académico (Martínez, 2007). La variable principal de la investigación es la competencia en resolución de problemas, estando esta desglosada en cuatro subvariables, cada una de ellas asociada a uno de los pasos del método de Polya: comprensión del enunciado, elaboración de un plan para la resolución del problema, ejecución de dicho plan y comprobación de la solución del problema analizando nuevas vías de solución.

#### 3.3. Desarrollo de la experiencia empírica

En primer lugar, los estudiantes realizaron un pretest consistente en seis problemas aritméticos acordes a su nivel académico. Este tipo de problemas ocupan la mayor parte del currículo tanto en Educación Primaria como en Educación Secundaria Obligatoria. Este tipo de problemas son muy recurrentes porque responden a situaciones cotidianas de los estudiantes, por lo que facilita la aplicación de habilidades matemáticas en distintos contextos (Roy y Roth, 2016). Una vez que los alumnos y alumnas realizaron la primera prueba de problemas, completaron un cuestionario de reflexión del que se pudo obtener información acerca de cómo consideraban que han trabajado la resolución de problemas. Tanto la encuesta de reflexión como la parrilla de corrección empleada para el análisis de los problemas contemplan cada una de las variables definidas para el estudio desglosada en una serie de indicadores como se muestra en la tabla 1.

A continuación, el alumnado realizó el trabajo con Scratch dividiendo el proceso en dos partes. Primero trabajaron, durante un total de 8 sesiones, con una serie de actividades diseñadas con Scratch para introducir el método de Polya consistentes en la corrección de un código erróneo y una serie de preguntas de reflexión: ¿cuál es el problema?, ¿cuál es el fallo en el programa?, ¿cómo lo has solucionado?, ¿existen otras alternativas para resolver el problema? El estudiante tiene que explicar cuál es el problema, ya que el programa no realiza la misma acción que se detalla en el enunciado. No se pide explícitamente comprender bien el enunciado, pero se le pide explicar cuál es el fallo en el programa respecto al enunciado, estando implícita la necesidad de comprenderlo. Posteriormente, para trabajar la elaboración de un plan de trabajo y su ejecución, se le pide explicar cuál es el fallo en el algoritmo y cómo lo ha solucionado. Por último, el estudiante debe plantear una alternativa para el programa que realice la misma acción que le pide el enunciado. Así, se está llevando a cabo una comprobación y planteamiento de otros planes viables y equivalentes a la resolución del problema.

En la segunda fase del trabajo, el alumnado realizó una serie de actividades para completar la programación de un videojuego de preguntas sobre la asignatura de matemáticas. Cada una de las actividades tenía planteadas una serie de preguntas de reflexión sobre la construcción del programa que el estudiante debe contestar antes de comenzar la programación de cada parte. Para esta fase del trabajo se invirtieron un total de 8 sesiones.

Una vez finalizado el trabajo con Scratch, cada estudiante realizó un postest consistente nuevamente en seis problemas aritméticos, distintos a los del pretest, y el mismo cuestionario de reflexión para ver cómo ha cambiado su concepción sobre el proceso de resolución de problemas.

Tabla 1  
Cuestionario de reflexión y parrilla de corrección de problemas

Variable	Valor	Indicador
1. Comprensión del enunciado	[0-1]	1.1 Identifica y expone los datos del problema correctamente.
		1.2 Explica la situación problemática.
		1.3 Expone el objetivo del problema explícitamente.
		1.4 Identifica y explica qué hay que calcular antes de empezar.
2. Elaboración del plan	[0-1]	2.1 Expone algún plan.
		2.2 La explicación del plan es adecuada.
		2.3 El plan que plantea es acertado.
3. Ejecución del plan	[0-1]	3.1 Antes de cada operación explica lo que va a realizar.
		3.2 Identifica explícitamente las distintas operaciones que ha realizado.
		3.3 El proceso seguido se corresponde con la explicación inicial de qué hay que calcular.
4. Comprobación de resultados	[0-1]	4.1 Redacta la solución del problema.
		4.2 Comprueba cada uno de los valores obtenidos en el proceso de resolución.
		4.3 Ha planteado otras alternativas para resolver el problema.
		4.4 Ha verificado la solución del problema con otro planteamiento.

#### 4. Análisis y resultados

El análisis de la variable principal de la investigación, esto es, la competencia en resolución de problemas, se ha realizado observando la puntuación total obtenida en cada uno de las pruebas y la diferencia entre ambas. Se ha obtenido que un 72.2 % del alumnado consigue una mejor puntuación en el postest, un 11.1 % obtiene una puntuación igual en ambas pruebas y un 16.7 % obtiene una puntuación menor en la prueba final. A continuación, se detallan los datos obtenidos para cada una de las subvariables en las que se ha dividido la variable principal para poder analizar a qué se deben los resultados obtenidos y la comparación de resultados obtenidos en las encuestas de reflexión (tabla 2).

##### 4.1. Comprensión del enunciado

A la hora de llevar a cabo la resolución de problemas, la comprensión del enunciado es en muchas ocasiones una complicación añadida por la dificultad que conlleva la creación de un modelo mental de lo que significa para poder elaborar una representación correcta para la elaboración del plan de resolución y su posterior ejecución (Gaulin, 2001). Tras la intervención práctica, un 66.7% de los estudiantes presentaron una mejora en cuanto a la comprensión del enunciado. A pesar de este resultado, el análisis de esta variable muestra que los alumnos tienen especial dificultad a la hora de expresar claramente la situación problemática planteada y el objetivo del problema. Aunque analizando los datos de la encuesta un 66.6% de los estudiantes aseguran poder explicarlo.

##### 4.2. Elaboración de un plan

En este apartado se ha analizado cómo el estudiante expone el plan para resolver el problema y si es acertado. Cabe destacar la importancia de esta etapa a la hora de resolver un problema, ya que es el momento en el que el estudiante determina cómo va a enfocar el proceso de resolución. Se ha obtenido que solo un

33.3 % del alumnado ha mejorado en esta variable, mientras que un 66.7 % mantiene la misma puntuación.

##### 4.3. Ejecución de dicho plan

Para analizar esta variable los indicadores hacen referencia a cómo el alumnado identifica las distintas operaciones que realiza y si el proceso seguido se corresponde con el plan inicial previamente planteado. Así, un 38.8 % del alumnado ha obtenido mejor resultado en estos aspectos, un 55.6 % presenta el mismo resultado y un 5.6 % ha obtenido un resultado peor en el postest.

Analizando los datos correspondientes a estas dos últimas variables de manera conjunta, se observa que el alumnado ha debido tener dificultades para entender cómo plasmar el trabajo en esta fase, ya que en ambos casos un alto porcentaje del alumnado ha obtenido un mismo resultado en ambas pruebas. Este hecho evidencia que no ha considerado la importancia de plasmar en el proceso de resolución el procedimiento correspondiente a la elaboración y ejecución del plan. A pesar de no recoger el proceso completamente, parte del alumnado considera estas partes del proceso de resolución ya que un 66.7% asegura que el procedimiento realizado coincide con su planteamiento inicial y un 61.2% dice que identifica las distintas operaciones realizadas.

##### 4.4. Comprobación de la solución del problema analizando nuevas vías de solución

En el análisis de esta variable no solo se ha tenido en cuenta si el estudiante redacta la solución obtenida, sino que también se ha considerado si hace alguna comprobación de las operaciones realizadas, así como del resultado final, y si propone alguna otra alternativa al proceso de resolución. Un 39 % del alumnado ha obtenido en el postest una calificación mayor, mientras que un 33.3 % ha mantenido la misma calificación y un 27.7 % ha obtenido una calificación menor en la prueba final. Analizando los datos de la encuesta destaca el aumento del alumnado que redacta la solución del problema.

Tabla 2.  
Comparación de resultados obtenidos en la encuesta de reflexión

	Valor	Indicador	Resultados pretest (porcentaje)		Resultado postest (porcentaje)	
			Sí	No	Sí	No
Comprensión del enunciado	[0-1]	1.1 Identifica y expone los datos del problema correctamente	83,4	16,6	83,4	16,6
		1.2 Explica la situación problemática	55,6	44,4	66,7	33,3
		1.3 Expone el objetivo del problema explícitamente	50	50	55,6	44,4
		1.4 Identifica y explica qué hay que calcular antes de empezar	38,9	61,1	55,6	44,4
Elaboración del plan	[0-1]	2.1 Expone algún plan	55,6	44,4	55,6	44,4
		2.2 La explicación del plan es adecuada	16,7	83,3	27,8	72,2
		2.3 El plan que plantea es acertado	27,8	72,2	27,8	72,2
Ejecución del plan	[0-1]	3.1 Antes de cada operación explica lo que va a realizar	50	50	50	50
		3.2 Identifica explícitamente las distintas operaciones que ha realizado	55,6	44,4	61,2	38,8
		3.3 El proceso seguido se corresponde con la explicación inicial de qué hay que calcular	77,8	22,2	66,7	33,3
Comprobación de resultados	[0-1]	4.1 Redacta la solución del problema	22,2	77,8	50	50
		4.2 Comprueba cada uno de los valores obtenidos en el proceso de resolución	50	50	33,4	66,6
		4.3 Ha planteado otras alternativas para resolver el problema	38,9	61,1	22,2	77,8
		4.4 Ha verificado la solución del problema con otro planteamiento	44,4	55,6	44,4	55,6

A la hora de trabajar la resolución de problemas es importante que el estudiante sea consciente de la importancia de comprobar, tantos los resultados obtenidos como la solución final, no solo viendo si las operaciones que ha realizado están bien hechas. En la mayoría de los casos, los estudiantes se quedan en este punto a la hora de comprobar resultados. Los resultados del cuestionario en este aspecto no son muy claros por las diferencias observables entre el pretest y el postest y porque ninguno lo ha mostrado en la prueba escrita. De modo que se presenta el mismo hecho que en las variables anteriores: el alumnado no ha sabido plasmar ciertos aspectos relevantes a la resolución del problema.

Haciendo una valoración más general de todo el proceso de resolución de problemas, los resultados analizados muestran que un 61.1% del alumnado ha obtenido una mejoría en dos o más de las variables analizadas, mientras que un 22.2% y un 16.7% muestra mejoría en una o en ninguna de ellas respectivamente.

Este análisis detallado de las distintas subvariables es debido a las limitaciones propias del diseño cuasiexperimental realizado. Si bien, para poder dar validez a este análisis y rechazar la hipótesis de que estas diferencias sean debidas al azar, esto es que la diferencia entre los resultados de las pruebas sea estadísticamente significativa, se ha realizado un análisis *t* de Student para muestras relacionadas con un nivel de confianza del 95 %. Previamente, y aceptada la asignación aleatoria del grupo, se ha comprobado la homocedasticidad para los resultados obtenidos en las pruebas, pudiendo aceptar la hipótesis nula de que las varianzas son semejantes gracias a la significancia obtenida en la prueba de Levene ( $p = 0.146$ ). A la hora de analizar la normalidad de las muestras, se obtiene una significancia asintótica algo menor que 0.05, pero podemos suponer el carácter normal de la muestra para realizar el estadístico *t* de Student. Para este análisis se ha obtenido una significancia de 0.003. Así, se puede rechazar la hipótesis nula y afirmar que hubo un incremento en el resultado de las pruebas ( $t(17) = -3.42, p < 0.05$ ) entre las

mediciones efectuadas antes ( $X_1 = 1.4$ ) y después ( $X_2 = 2.8$ ) de la intervención práctica. Para el análisis estadístico se ha tomado la aplicación de software libre para el análisis de datos PSPP.

Estos resultados quedan estrechamente ligados con los obtenidos en otros estudios similares que sugieren que la inclusión de esta perspectiva computacional en la resolución de problemas mejora de manera general la comprensión de las propias matemáticas y distintas habilidades de codificación, así como la comprensión de conceptos lógicos básicos de programación (Sung, Ahn y Black, 2017). De manera más general, la influencia positiva de este recurso también queda reflejada en otro estudio que recomienda la inclusión del pensamiento computacional en niveles anteriores, 5º y 6º de Educación primaria, para reforzar metodologías activas, la motivación y la utilidad del proceso de aprendizaje (Sáez-López, Román-González y Vázquez-Cano, 2016). En general, el pensamiento computacional se está convirtiendo en una habilidad fundamental para el Siglo XXI, por lo que es importante incorporarlo en distintas áreas educativas introduciendo los conceptos básicos progresivamente desde la Educación Primaria (Yadav et al., 2014).

## 5. Discusión y conclusiones

A la hora de trabajar la resolución de problemas, el estudiante tiene que aprender no solo a utilizar representaciones e interpretar la solución, también debe saber comunicar los resultados y reflexionar sobre el proceso (Santos-Trigo, 2014). Con esta idea y observando los resultados obtenidos, se puede concluir que el uso del pensamiento computacional para trabajar la resolución de problemas a través de Scratch resulta efectivo. Más de un 70% del alumnado participante ha mejorado, en general, en el proceso de resolución de problemas. Aunque al analizar cada una de las subvariables se ha apreciado la dificultad del estudiante para plasmar, de manera detallada, partes del proceso de trabajo. Es

lógico entender que al expresar la resolución de un problema es difícil que el resolutor muestre todo lo que está pensando, ya que este proceso puede interrumpir el trabajo que está realizando; más aún si tenemos en cuenta la edad del alumnado con el que hemos trabajado. Así, a la vista de los resultados obtenidos, no hay que pensar que el alumnado no haya sabido trabajar de manera adecuada las variables sujetas a estudio, sino más bien que no han adquirido la capacidad necesaria para poder recoger estos aspectos en una prueba escrita. Este hecho queda constatado con las respuestas analizadas del cuestionario de reflexión del alumnado, aspecto que no puede ser observado con un análisis meramente cuantitativo; es la valoración cualitativa del proceso de resolución la que ayuda concluir este resultado.

La inclusión del pensamiento computacional es una herramienta efectiva para trabajar metodologías estructuradas para la resolución de problemas, ya que permite utilizar habilidades como el uso de patrones, la abstracción, la descomposición o el pensamiento algorítmico (Resnick et al., 2009), destrezas fundamentales para este proceso de razonamiento. Pero también conviene destacar la importancia de que aportar un nuevo escenario para desarrollar este tipo de habilidades incluye un importante aspecto motivador, siendo este un factor clave para una efectiva transmisión de las habilidades cognitivas (Jaeggi, Buschkuhl, Shah, y Jonides, 2014). El diseño de este tipo de estrategias despierta la motivación del alumnado, ofreciendo distintas alternativas de trabajo y dinamismo a las sesiones de aula, permitiendo descubrir y realizar de manera más personal todo el proceso de aprendizaje gracias al avance de las tecnologías para el aprendizaje y el conocimiento (Leris López y Sein-Echaluce Lacleta, 2011).

### Limitaciones de la investigación

El propio diseño cuasiexperimental de medidas repetidas que se ha realizado conlleva una serie de limitaciones, principalmente el trabajar con un solo grupo experimental dada la disponibilidad del alumnado participante. Esto limita igualmente el tamaño de la muestra, ya que al trabajar con un solo grupo de Educación Secundaria no es posible tener un elevado número de participantes. A pesar de esta situación, la correcta ejecución y planificación del cuasiexperimento ha aportado información adecuada para probar que trabajar contenidos curriculares a través del pensamiento computacional y Scratch es efectivo y se puede hacer un análisis de cómo este recurso mejora el proceso de aprendizaje de los estudiantes. En base a los resultados que se han obtenidos, la evolución natural de este trabajo es un análisis comparativo con un grupo control que siga una metodología tradicional para trabajar contenidos de la asignatura de matemáticas y un grupo experimental que trabaje estos mismos contenidos con Scratch.

### Referencias bibliográficas

Balanskat, A., y Engelhardt, K. (2015). Computing our future. Computer programming and coding. Priorities, school curricula and initiatives across Europe. Brussels: European Schoolnet, Belgium. Recuperado de: (<https://goo.gl/j0PSxq>) (2015-11-23).

Campbell D.T., y Stanley J.C. (1966/1963). Experimental and quasi-experimental designs for research. (R. McNally, Ed.) Chicago: Houghton Mifflin Company (<https://goo.gl/ZzzSid>) (2016-03-21).

Castro, W. F. y Godino, J. D. (2011). Métodos mixtos de investigación en las contribuciones a los simposios de la SEIEM (1997-2010). En M. Marín, G. Fernández, L. Blanco y M. Pa-

larea (Eds.), Investigación en Educación Matemática XV (pp. 99-116). Ciudad Real: SEIEM. (<https://goo.gl/QzKnU4>) (2017-02-10).

DiSessa, A. (2000). Changing minds: computers, learning, and literacy. Cambridge: MIT Press.

Gander, W., Petit, A., Berry, G., Demo, B., Vahrenhold, J., McGettrick, A., ... Meyer, B. (2013). Informatics Education: Europe Cannot Afford to Miss the Boat. Joint Informatics Europe y ACM Europe working group on informatics Education. Recuperado de: (<https://goo.gl/nrddkS>) (2017-01-22)

Gaulin, D. C. (2001). Tendencias actuales en la resolución de problemas. Sigma, (19), 51-63. (<https://goo.gl/T1dUMh>) (2017-02-13).

Gutiérrez-Esteban, P., y Beccerra Traver, M. T. (2014). Los entornos personales de aprendizaje (PLE). Una experiencia de aprendizaje informal en la formación inicial del profesorado. Revista Latinoamericana de Tecnología Educativa, 13(2), 49-60. <https://doi.org/10.17398/1695>

Halverson, E. R., & Sheridan, K. (2014). The Maker Movement in Education. Harvard Educational Review, 84(4), 495-504. <https://doi.org/10.17763/haer.84.4.34j1g681403820>

ISTE. (2011). Operational Definition of Computational Thinking. Recuperado de: (<https://goo.gl/ij4xfI>) (2016-02-20).

Jaeggi, S. M., Buschkuhl, M., Shah, P., y Jonides, J. (2014). The role of individual differences in cognitive training and transfer. Memory y Cognition, 42(3), 464-480. <https://doi.org/10.3758/s13421-013-0364-z>

Johnson, B., Onwuegbuzie, A. J., y Turner, L. A. (2007). Toward a Definition of Mixed Methods Research. Journal of Mixed Methods Research, 1(2), 112-133. <https://doi.org/10.1177/1558689806298224>

Klieme, E. (2004). Assessment of Cross-Curricular problem-solving competencies, Comparing Learning Outcomes. In M. S. J. H. Moskowitz (Ed.), Comparing Learning Outcomes: International Assessment and Educational Policy (pp. 81-107). London: Routledge Falmer, London. (<https://goo.gl/P69mfm>) (2015-10-15).

Lee, F. Martin, J. Denner, B. Coulter, W. Allan, J. Erickson, J. Malyn-Smith, J., and L. Werner (2011)., "Computational Thinking for Youth in Practice". Journal of Computational Science Education. 2(1), 1-10. <https://doi.org/10.1016/j.promfg.2015.07.761>

Leris López, M. D., y Sein-Echaluce Lacleta, M. L. (2011). La personalización del aprendizaje: un objetivo del paradigma educativo centrado en el aprendizaje. Arbor: Ciencia, Pensamiento Y Cultura, 187(3), 123-134. <https://doi.org/10.3989/arbor.2011.Extra-3n3135>

Lesh, R., y Zawojewski, J. (2007). Problem solving and modeling. Second handbook of research on mathematics teaching and learning. In F. K. Lester (Ed.), Second handbook of research on mathematics teaching and learning (pp. 763-804). Charlotte, NC: NCTM.

Martínez, R.-A. (2007). La investigación en la práctica educativa: Guía metodológica de investigación para el diagnóstico y evaluación en los centros docentes. Madrid: Ministerio de Educación y Ciencia. (<https://goo.gl/L4IQhV>) (2016-04-19).

Mayer, R., y Wittrock, M. (2006). Problem solving, Handbook of Educational Psychology. In P. A. y P. H. W. Alexander y P.H. Winne (Ed.), Handbook of Educational Psychology. (pp. 287-303). Mahwah, NJ: Routledge.

Mayoral, S., Roca, M., Timoneda, C. y Serra, M. (2015). Mejora de la capacidad cognitiva del alumnado de primer curso de Educación Secundaria Obligatoria. Aula Abierta, 43(1), 9-17. <https://doi.org/10.1016/j.aula.2014.10.001>



- Moreno-León, J., Robles, G., y Román-González, M. (2016). How social are Scratch learners? A comprehensive analysis of the Scratch platform for social interactions. 12th International conference on Open Source Systems - Workshop on FLOSS in education and computational thinking, Gothenburg, Suecia.
- Moreno-León, J., y Robles, G. (2016). Code to learn with Scratch? A systematic literature reviews. IEEE Global Engineering Education Conference, EDUCON, 10-13, 150-156. <https://doi.org/10.1109/EDUCON.2016.7474546>
- National Research Council. (2010). Report of a workshop on the scope and nature of computational thinking. Washington, DC. <https://doi.org/https://doi.org/10.17226/12840>
- Nelson, J. (2009, May). Celebrating Scratch in Libraries. Creation software helps young people develop 21st-century literacy skills. *School Library Journal*, 20-21. Recuperado de: (<https://goo.gl/19ucfP>) (2017-01-28)
- Niss, M. (2002). Mathematical Competencies and the Learning of Mathematics: the Danish Kom Project. Proceedings of the 3rd Mediterranean Conference on Mathematical Education. Roskilde, Denmark. (<https://goo.gl/gngHY9>) (2016-03-15).
- Onrubia, J. (2016). Aprender y enseñar en entornos virtuales: actividad conjunta, ayuda pedagógica y construcción del conocimiento. *RED: Revista de Educación a Distancia*, 50(3), 1-14. <http://dx.doi.org/10.6018/red/50/3>
- Onwuegbuzie, A. J., y Collins, K. M. (2007). A typology of mixed methods sampling designs in social science research. *The Qualitative Report*, 12(2), 474-498. (<https://goo.gl/3ubQHo>) (2017-03-25).
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas* (Second 2nd Edition). New York: Basic Books.
- Papert, S. (1996). An exploration in the Space of Mathematics Education. *International Journal of Computers for Mathematical Learning*, 1(1), 95-123. (<https://goo.gl/CFzpf5>) (2016-12-04).
- Polya, G. (1945). *How to Solve It* (2nd ed.). New York: Princeton University Press, Doubleday Anchor Books.
- Puig, L. (1996). EL Modelo de Competencia. In L. Puig (Ed.), *Elementos de la resolución de problemas* (pp. 33-49). Granada: Comares.
- Recio, T. (2002). Situación de la Enseñanza de las Matemáticas en la Educación Secundaria Española. Santander. (<https://goo.gl/zZ0pNO>) (2016-05-14).
- Resnick, M., Maloney, J., Rusk, N., Eastmond, E., Brennan, K., Millner, A., ... Kafai, Y. (2009). Scratch: Programming for all. *Communication of the ACM*, 52(11), 62-67. <https://doi.org/10.1145/1592761.1592779>
- Romero-Frías, E., y Magro Mazo, C. (2016). La emancipación intelectual en la sociedad digital: El maestro ignorante de Rancière en nuestros días. *Revista Letral*, 16, 89-105. (<https://goo.gl/K7wkTg>) (2017-03-21).
- Ross, E. y Gautreaux, M. (2018). Pensando de manera crítica sobre el pensamiento crítico. *Aula Abierta*, 47(4), 383-386. <https://doi.org/10.17811/rifie.47.4.2018.383-386>.
- Roy, S., & Roth, D. (2016). Solving general arithmetic word problems. arXiv preprint. Recuperado de: (<https://arxiv.org/abs/1608.01413>).
- Sáez-López, J.-M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two-year case study using "Scratch" in five schools. *Computers & Education*, 97, 129-141. <https://doi.org/10.1016/j.compedu.2016.03.003>
- Santos-Trigo, M. (2014). Problem solving in mathematics education. En In S. Lerman (ed.), *Encyclopedia of Mathematics Education* (pp. 496-501). NY: Springer.
- Schoenfeld, A. H. (1985). *Mathematical Problem Solving*. San Diego: Orlando Academic Press.
- Schoenfeld, A. H. (2012). How we think: A theory of human decision making, with a focus on teaching. In C. S. (Ed.), *The Proceedings of the 12th International Congress on Mathematical Education* (pp. 229-243). Springer, Cham. [https://doi.org/10.1007/978-3-319-12688-3\\_16](https://doi.org/10.1007/978-3-319-12688-3_16)
- Selwyn, N. (2009). The digital native - myth and reality. *Aslib Proceedings*, 61(4), 364-379.
- Wing, J. (2006). Computational Thinking. *Communication of the ACM*, 49(3), 33-35. (<https://goo.gl/WnL3va>) (2016-04-12).
- Wing, J. (2011). Research notebook: Computational thinking - What and why? (Carnegie Mellon University, Ed.), *The Link Magazine*, Spring (6), 20-23. (<https://goo.gl/UFwLRB>) (2016-04-20).
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational Thinking in Elementary and Secondary Teacher Education. *ACM Transactions on Computing Education*, 14(1), 1-16. <https://doi.org/10.1145/2576872>