



UNIVERSIDAD DE CÓRDOBA

Doctorado en Ciencias Sociales y Jurídicas

**Contribución del Pensamiento Computacional
con Scratch al proceso de enseñanza y aprendizaje de
las Matemáticas**

Contribution of computational thinking with Scratch to the
teaching and learning process of Mathematics

Directores

Natividad Adamuz Povedano Rafael Bracho López

Doctorando:

Álvaro Molina Ayuso

Córdoba, 2022

TITULO: *Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas*

AUTOR: *Álvaro Molina Ayuso*

© Edita: UCOPress. 2022
Campus de Rabanales
Ctra. Nacional IV, Km. 396 A
14071 Córdoba

<https://www.uco.es/ucopress/index.php/es/>
ucopress@uco.es

**TÍTULO DE LA TESIS:**

Contribución del pensamiento computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

DOCTORANDO:

Álvaro Molina Ayuso

INFORME RAZONADO DE LOS DIRECTORES DE LA TESIS

El propósito de este trabajo doctoral ha sido analizar como el desarrollo del Pensamiento Computacional a través de prácticas educativas desarrolladas con el software de programación visual por bloques Scratch puede favorecer al proceso de enseñanza y aprendizaje de las Matemáticas en distintos niveles educativos: en Educación Primaria, Secundaria y en la formación inicial del profesorado de Educación Primaria.

Los estudios llevados a cabo han ayudado a comprobar el efecto positivo que puede tener la inclusión de prácticas computacionales con Scratch y el desarrollo de habilidades de Pensamiento Computacional en el proceso de aprendizaje de las Matemáticas en diferentes contextos. El ver que puede favorecer el desarrollo de habilidades asociadas a la resolución de problemas hace que sea un recurso que se puede integrar perfectamente en el aprendizaje de esta disciplina ya que la resolución de situaciones problemáticas es uno de los ejes principales que permite a nuestro alumnado tener una adecuada alfabetización matemática. Por otro lado, y de manera más específica para su inclusión curricular, los efectos positivos que puede tener para trabajar contenidos y saberes propios de las Matemáticas facilitan la inclusión de este tipo de recursos en el proceso de enseñanza en un contexto formal y adecuado para distintos niveles educativos. Pero para poder llevar esto al aula de manera efectiva es necesario que el profesorado tenga un nivel mínimo de habilidades y destrezas computacionales. Para que esto sea así, en el tercer estudio de la tesis se muestra el efecto positivo que tiene la inclusión de prácticas computacionales en la formación inicial de futuros y futuras docentes a la vez que motiva su interés por este tema y hace que se conciba como un elemento importante para la educación que requiere cualquier estudiante del Siglo XXI

En cuanto al proceso de elaboración de esta tesis, el doctorando ha conseguido hábilmente integrar su experiencia profesional como docente de matemática en Educación Secundaria de largo recorrido con las competencias investigadoras que ha ido desarrollando a lo largo del doctorado. Al mismo tiempo, el doctorando ha realizado una profunda revisión de antecedentes para diseñar las implementaciones presentadas en la tesis, algo que se plasma en el presente trabajo.

Su gran compromiso le ha permitido superar exitosamente todos los problemas surgidos a lo largo del periodo doctoral, incluso los debidos a la pandemia por COVID-19, llevando a cabo con éxito una experimentación en cada una de las tres etapas educativas mencionadas anteriormente.

Por todo ello, se autoriza la presentación de la tesis doctoral.

Córdoba, 7 de noviembre de 2022

Firma de los directores

Fdo.: Rafael Bracho López

Fdo.: Natividad Adamuz Povedano

Resumen

El objetivo de esta tesis es contribuir a la inclusión curricular del Pensamiento Computacional en distintos niveles educativos, desde la Educación Primaria hasta la formación inicial del profesorado. Para ello, se han analizado datos los obtenidos y extraído conclusiones a partir de distintas experiencias cuyo diseño ha permitido trabajar habilidades relacionadas con el Pensamiento Computacional a través de un contexto relacionado con la asignatura de Matemáticas y con la formación inicial del profesorado de Educación Primaria en este ámbito. Esto ha dado la posibilidad de analizar distintos enfoques que han permitido incluir recursos que favorecen el desarrollo del Pensamiento Computacional de manera transversal para abordar contenidos propios de la asignatura de Matemáticas como los procesos de resolución de problemas o distintas estructuras y conceptos de razonamiento.

En las distintas experiencias que forman parte de esta investigación se ha empleado fundamentalmente una metodología cuantitativa, si bien en algunos casos los datos se han acompañado de un análisis cualitativo del trabajo realizado por el alumnado y el profesorado participante. En todos los casos se ha realizado una toma de datos con estudios pre-post para poder analizar el impacto de cada planteamiento empírico. En algunas ocasiones, por limitaciones propias de los estudios, no ha sido posible tomar muestras de grupos control y experimental. Pero esta comparativa sí es la predominante ya que permite analizar de manera más completa el impacto que tiene la

experiencia de aprendizaje diseñada en cada grupo. Esto nos ha permitido saber si el denominado Pensamiento Computacional se corresponde con un conjunto de habilidades y destrezas adecuadas para ser trabajadas en el ámbito de la educación matemática, facilitando el desarrollo de otras habilidades como el proceso de resolución de problemas. Para llevar a cabo los distintos procesos de evaluación en cada una de las experiencias realizadas, se ha optado por utilizar herramientas de evaluación de diferentes características. Por un lado, se han utilizado herramientas de transferencia de habilidades para analizar la forma en la que se han trabajado distintos contenidos de Matemáticas a través de recursos de Pensamiento Computacional de manera transversal. En otros estudios se han utilizado herramientas de carácter sumativo y aptitudinal para evaluar el conocimiento y grado de adquisición de distintas destrezas de Pensamiento Computacional. Los resultados obtenidos en los distintos estudios realizados en esta tesis apuntan a la efectividad de trabajar contenidos de Matemáticas como la Geometría o los proceso de resolución de problemas de manera transversal utilizando recursos de Pensamiento Computacional. Por otra parte, comprobamos que incluir este tipo de prácticas en la formación inicial del profesorado favorece el desarrollo de habilidades asociadas al Pensamiento Computacional y ofrecen la posibilidad de plantear nuevas experiencias de aprendizaje que promueven el desarrollo de la creatividad y el pensamiento lógico matemático.

Abstract

The aim of this thesis is to contribute to the curricular inclusion of computational thinking in different educational levels, from primary education to initial teacher training. For this purpose, we have designed and analyzed data obtained from different experiences that have allowed to develop skills related to computational thinking through a mathematics context. This has made it possible to analyze different approaches that have made it possible to include resources that encourage the development of computational thinking in a transversal way to address content specific to the subject of mathematics, such as problem-solving processes or different reasoning structures and concepts.

In the different experiences that completed this research, a quantitative methodology was used for the most parts, although in many of the cases the data were accompanied by a qualitative analysis of the work carried out by the students and the participating teachers. In all cases, data collection has been conducted with pre-post studies in order to analyze the impact of each empirical approach. In all cases, due to the limitations of the studies themselves, it has not been possible to take samples of control and experimental groups. However, this comparison is the predominant one, since it allows a more complete analysis of the impact of the learning experience designed for each group. This has allowed us to know if the so-called

computational thinking corresponds to a set of skills and abilities that are appropriate to be worked on in the field of mathematics education. In order to implement the different evaluation processes in each of the experiences carried out, we have used different types of evaluation tools. On the one hand, skills transfer tools have been used to analyze the way in which different mathematical contents have been worked on through computational thinking resources in a transversal manner. In other studies, a summative and aptitudinal tool have been used to evaluate the knowledge and degree of acquisition of different computational thinking skills.

The results obtained in the different studies carried out in this thesis indicate the effectiveness of working on mathematics content, such as geometry or problem-solving processes, in a transversal way using computational thinking resources. On the other hand, we found that including this type of practices in initial teacher training encourages the development of skills associated with computational thinking and the possibility of proposing new learning experiences that favor the development of creativity and mathematical logical thinking.

DEDICATORIA

Quiero dedicar todo el esfuerzo empeñado en terminar este trabajo a toda mi familia. Especialmente a mi padre, Francisco. Un ejemplo para todos y todas de cómo debemos adaptarnos y adecuarnos a las distintas situaciones que nos puede plantear la vida para sacarle el máximo partido a nuestras posibilidades. Un ejemplo que siempre he tenido para llegar al final de este camino.

Igualmente, quiero dedicar este trabajo a mis hijos, Marcos y David. Ellos son el motivo más importante para seguir avanzando y mejorar día a día.

AGRADECIMIENTOS

Esta tesis no habría podido realizarse sin la ayuda de tanta gente que siempre me ha apoyado, ayudado y animado a continuar con este trabajo a pesar del esfuerzo añadido que ha supuesto durante todos estos años.

En primer lugar, quiero agradecer a mis directores, Natividad Adamuz y Rafael Bracho, el gran esfuerzo que han realizado para que pueda culminar este trabajo. Desde que les planteé la posibilidad de hacer este estudio, siempre me han guiado con toda su ayuda para llegar al final de este largo camino y gracias a ellos he podido aprender mucho, tanto para completar este trabajo como para mejorar mi práctica diaria como docente.

Quiero agradecerérselo a mi familia, especialmente a Lara ya que sin su compañía ni este trabajo ni ningún otro logro que pueda alcanzar en mi vida personal y profesional puede ser posible. Gracia también a mis padres y a toda mi familia por apoyarme siempre y en todo momento.

Gracias todos mis amigos y amigas que siempre me han animado a seguir con este trabajo y con quienes siempre he compartido el día a día de esta experiencia, muchas gracias por todo lo que habéis aportado. Aunque no se vea, ha sido mucho.

Índice

I. Parte introductoria.....	19
Capítulo 1: Introducción.....	21
1.1 Pertinencia del proyecto de investigación.....	21
1.2 Planteamiento del problema de investigación.....	24
1.3 Objetivos.....	25
1.4 Estructura y contenido de la tesis.....	29
II. Parte teórica.....	31
Capítulo 2: ¿Qué es el Pensamiento Computacional?.....	33
2.1 Origen de la idea de Pensamiento Computacional.....	35
2.2 La búsqueda de una definición consensuada.....	44
2.3 Marco conceptual para el Pensamiento Computacional.....	46
2.4 Definición de Pensamiento Computacional.....	50
2.5 Integración del Pensamiento Computacional en el currículo.....	56
2.6 Herramientas de evaluación del Pensamiento Computacional.....	63
Capítulo 3: Programación informática y Pensamiento Computacional en el sistema educativo.....	69
3.1 El marco de la alfabetización digital y la código-alfabetización.....	76
3.2 Un movimiento social para desarrollar el Pensamiento Computacional...	83
Capítulo 4: Lenguajes de programación visual por bloques. Scratch.....	85
4.1 Programación con Scratch en contextos educativos.....	91
4.2 Programación con Scratch para aprender Matemáticas.....	99

III. Parte empírica.....	105
Capítulo 5: La resolución de problemas basada en el método de Polya usando el Pensamiento Computacional y Scratch con estudiantes de Educación Secundaria.....	107
Capítulo 6: Pensamiento Computacional con Scratch. Recurso para trabajar Geometría en 5º de Educación Primaria.....	133
Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria.....	157
IV. Parte final.....	181
Capítulo 8: Conclusiones.....	183
8.1 Aportaciones del marco teórico al proceso de investigación.....	183
8.2 Contribuciones de la investigación.....	187
8.2.1 Objetivo específico 1.....	191
8.2.2 Objetivo específico 2.....	192
8.2.3 Objetivo específico 3.....	194
8.2.4 Objetivo específico 4.....	196
8.2.5 Objetivo específico 5.....	198
8.2.6 Objetivo específico 6.....	200
8.2.7 Objetivo específico 7.....	201
8.2.8 Objetivo específico 8.....	203
8.3 Limitaciones y futuras líneas de investigación.....	205
Referencias bibliográficas.....	210
V. Anexos.....	238

Anexo 1. Problemas de pretest para estudiantes de Educación Secundaria. .	238
Anexo 2. Problemas del postest para estudiantes de Educación Secundaria.	240
Anexo 4. Prueba de Geometría para estudiantes de Educación Primaria.....	243
Anexo 5. Guión para la discusión con los estudiantes del proyecto de investigación.....	251
Anexo 6. Guión para la discusión con los estudiantes del proyecto de investigación.....	252
Anexo 7. Cuestionario para estudiantes en el Grado de Educación Primaria	253

Índice de tablas

Tabla 1: Parrilla de corrección de problemas.....	122
Tabla 2: Comparación de resultados obtenidos en la encuesta de reflexión..	125
Tabla 3: Contenidos incluidos en las preguntas.....	144
Tabla 4: Categorías definidas en el análisis cualitativo.....	145
Tabla 5: Homogeneidad de grupos.....	147
Tabla 6: Distribución normal de la muestra.....	147
Tabla 7: Parrilla de corrección para el test.....	148
Tabla 8: Distribución de los participantes.....	167
Tabla 9: Datos del diagrama de cajas.....	172
Tabla 10: Puntuación obtenida en cada test.....	173
Tabla 11: Homogeneidad de grupos.....	174
Tabla 12: Distribución normal de los grupos.....	174
Tabla 13: Análisis ANOVA.....	175
Tabla 14: Relación ente hipótesis y objetivos con los estudios realizados....	189

Índice de figuras

Figura 1: Conexión entre el PC y la CD.....	60
Figura 2: Diagrama de Venn del pensamiento matemático y computacional	104
Figura 3: Medida test inicial y test final en %.....	146
Figura 4: Diferencia de los test de cada grupo.....	146
Figura 5: Promedios de puntuaciones obtenidas.....	171
Figura 6: Diagrama de cajas de las puntuaciones obtenidas.....	171

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

I. Parte introductoria

Capítulo 1: Introducción

[1.1 Pertinencia del proyecto de investigación](#)

[1.2 Planteamiento del problema de investigación](#)

[1.3 Objetivos](#)

[1.4 Estructura y contenido de la tesis](#)

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Capítulo 1: Introducción

1.1 Pertinencia del proyecto de investigación

Entendiendo el Pensamiento Computacional como un conjunto de habilidades relacionadas con el razonamiento lógico y con los procesos de resolución de problemas, es una realidad el hecho de que el desarrollo de estas destrezas e ideas han tomado un papel muy importante en los procesos de innovación educativa durante la última década a nivel global (Falloon, 2016). Esto precisa que nuestro alumnado desarrolle estas destrezas y habilidades asociadas a los procesos de resolución de problemas nuevos y no estructurados, empleando recursos tecnológicos que propicien una correcta alfabetización digital en distintas perspectivas, como ciudadanos preparados para un mundo laboral cada vez más cambiante, diverso y transversal.

Obviamente, no todas las profesiones a las que accederá nuestro alumnado en un futuro reciente van a desarrollarse utilizando ordenadores o cualquier tecnología programable (Adell Segura et al., 2019). Pero la conceptualización actual del Pensamiento Computacional va más allá de aprender a programar o a realizar procesos utilizando un ordenador: aborda y desarrolla los procesos y habilidades necesarias para plantear y resolver problemas complejos. Además, tal y como apuntan las directrices de la Comisión Europea (Consejo de la Unión Europea, 2018), el desarrollo de estas habilidades está ligado a un uso

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

seguro, crítico y responsable de las tecnologías digitales, las cuales no solo están presentes en el mundo laboral, sino que también tienen una importante presencia en nuestra sociedad. Por lo que el Pensamiento Computacional se posiciona como un conjunto de destrezas fundamentales para cualquier estudiante del Siglo XXI.

Por tanto, la idea del Pensamiento Computacional como un conjunto de habilidades básicas para nuestro alumnado es algo que requiere una incorporación curricular adecuada que facilite su implantación y desarrollo en los distintos niveles educativos, tanto de carácter obligatorio como postobligatorio. Para esto es necesario partir de un marco conceptual que ayude a establecer una definición que permita diseñar distintas estrategias de enseñanza y aprendizaje, así como disponer de las necesarias herramientas de evaluación para analizar el trabajo desarrollado con estas prácticas junto a la capacidad de resolución de problemas (Ortega-Ruipérez y Asensio, 2021). Igualmente, es necesario realizar procesos empíricos en los que se valoren tanto el grado de adquisición y desarrollo de estas destrezas como la posibilidad de utilizarse recursos educativos que permitan trabajar el Pensamiento Computacional de manera transversal en otras asignaturas y ámbitos. Esta necesidad también está presente en la formación inicial del profesorado, la cual debe estar sustentada en unos cimientos sólidos y estables tales que permitan poder llevar al aula de manera efectiva el desarrollo de este

conjunto de habilidades. Además, la rapidez con la que se desarrollan distintos recursos tecnológicos y su presencia cada vez más numerosa en los procesos de enseñanza-aprendizaje, hacen necesario investigar acerca de nuevos modelos que faciliten toda esta incorporación para capacitar a nuestro alumnado en todas estas nuevas demandas sociales y profesionales (Prendes Espinosa y Cerdán Cartagena, 2020).

En cierto modo, todavía pueden ser insuficientes las investigaciones que se han realizado en el ámbito del Pensamiento Computacional para valorar su impacto en el ámbito educativo (Guamán Gómez et al., 2021). Pero hoy nos encontramos con una corriente muy importante de investigación sobre este tema, tanto para su conceptualización como para analizar su influencia y desarrollo en el ámbito educativo. Una corriente que ayuda a identificar nuevas líneas para el Pensamiento Computacional, nuevos modelos de desarrollo y ejemplos de cómo usar estas ideas (Palts y Pedaste, 2020).

En España son muchas las directrices que apuntan a la necesidad de incluir el desarrollo de habilidades de Pensamiento Computacional en el currículo, tanto a través de la legislación estatal como desde las distintas comunidades autónomas. Como se ha dicho anteriormente, este proceso de inclusión hace necesario que se realicen experiencias de investigación y análisis que permitan obtener conclusiones que ayuden a desarrollar este tipo de prácticas en el

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

currículo oficial para llegar a un adecuado grado de adquisición de estas competencias. Así, la presente investigación busca contribuir al desarrollo de estas experiencias aportando conclusiones que puedan ayudar a los docentes de distintos niveles educativos a incluir estas prácticas en el proceso de enseñanza y aprendizaje de las Matemáticas.

1.2 Planteamiento del problema de investigación

El desarrollo tecnológico que marca muchos de los aspectos cotidianos de nuestra vida, tiene una importante presencia y repercusión en la evolución de la Educación y en las necesidades de nuestro alumnado para adaptarse al desarrollo social y económico que marcan estos factores. Esto hará que muchos de los trabajos que conocemos en la actualidad acaben desapareciendo o siendo suplantados por otras actividades basadas en procesos automatizados. Para Blender et al. (2015) muchos de estos trabajos y las habilidades que se requieren para afrontarlos estarán estrechamente ligados al desarrollo de competencias relacionadas con las propias habilidades del Pensamiento Computacional. Esto puede dejar entrever la necesidad de incluir el desarrollo de este tipo de destrezas en la educación formal, siendo necesario que se realice de una manera transversal e integrada en el currículo. Con esta idea se plantea la necesidad de que una vertiente del proceso de innovación que puede darse dentro del sistema educativo español esté ligado a trabajar

este tipo de competencias y prácticas asociadas al Pensamiento Computacional.

En este contexto, en la investigación realizada en el presente trabajo se han analizado procesos de aprendizaje de las Matemáticas que se han desarrollado utilizando recursos de Pensamiento Computacional. Para abordar los distintos escenarios e intentar arrojar luz sobre cómo puede ser la evolución en el proceso de aprendizaje de nuestro alumnado y el desarrollo de las destrezas asociadas a este concepto, se han planteado y analizado investigaciones en distintos niveles educativos: Educación Primaria, Educación Secundaria y formación inicial del profesorado de Educación Primaria. Todas estas actuaciones se han realizado contextualizadas e incluidas en el aprendizaje de las Matemáticas, ya que el carácter transversal es algo importante en el desarrollo de prácticas de Pensamiento Computacional.

1.3 Objetivos

La finalidad de esta investigación es contribuir al desarrollo de las habilidades y destrezas asociadas al Pensamiento Computacional a través de la inclusión de prácticas que fomenten estas acciones en la enseñanza de las Matemáticas. Por dicho motivo, se han realizado diferentes experiencias empíricas que han permitido analizar tanto el desarrollo de este tipo de destrezas como el impacto que tiene la inclusión de prácticas computacionales en el aprendizaje

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

de las Matemáticas en distintos niveles educativos, incluyendo en la formación inicial del profesorado de Educación Primaria.

El objetivo general de esta investigación ha sido **analizar el impacto sobre el aprendizaje de las Matemáticas en Educación Primaria y Secundaria que tiene la inclusión de prácticas asociadas al Pensamiento Computacional utilizando el lenguaje de programación Scratch, así como el desarrollo de habilidades asociadas a estas prácticas en la formación inicial del profesorado de Educación Primaria.**

Este objetivo general se desglosa en un total de 8 objetivos específicos que derivan directamente de la revisión bibliográfica realizada a lo largo de todo el periodo de trabajo, el cual ha estado marcado por las diferentes experiencias empíricas desarrolladas. A continuación, se enumeran dichos objetivos específicos y la relación con los distintas experiencias realizadas.

1. Estudiar la capacidad de análisis y la comprensión de los enunciados de problemas para la elaboración de estrategias de resolución empleando prácticas de Pensamiento Computacional con Scratch **(OE1).**

2. Analizar la influencia de las prácticas de Pensamiento Computacional en el abordaje de los procesos que se siguen en la resolución de problemas **(OE2)**.
3. Estudiar la motivación del alumnado realizando actividades de Pensamiento Computacional con Scratch **(OE3)**.
4. Potenciar el uso de nuevas metodologías y estrategias de aprendizaje a través de la inclusión curricular del software de programación visual Scratch **(OE4)**.
5. Evaluar el desarrollo de la competencia y el razonamiento matemático en Geometría trabajando actividades de Pensamiento Computacional como recurso transversal de aprendizaje con alumnos de 5º curso de Educación Primaria **(OE5)**.
6. Analizar el grado de desarrollo de diferentes competencias computacionales realizando actividades con el lenguaje de programación Scratch en la formación inicial del Profesorado de Educación Primaria **(OE6)**.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

7. Valorar la percepción de este tipo de actividades y recursos educativos en la formación inicial de futuros docentes de Educación Primaria **(OE7)**.
8. Potenciar el uso de nuevas metodologías y estrategias de aprendizaje trabajando el Pensamiento Computacional de manera transversal en la enseñanza de las Matemáticas **(OE8)**.

Del mismo modo, teniendo como punto de referencia las distintas experiencias que se enmarcan en el contexto de esta investigación, con los objetivos previamente planteados, se determinan las siguientes hipótesis:

1. La inclusión curricular del Pensamiento Computacional favorece la capacidad de resolución de problemas en el aprendizaje de las Matemáticas **(H1)**.
2. El desarrollo de actividades relacionadas con el Pensamiento Computacional en el área de las Matemáticas favorece el desarrollo de los contenidos curriculares y mejora el proceso de aprendizaje **(H2)**.
3. Trabajar actividades de Pensamiento Computacional con Scratch en la formación inicial del profesorado de Educación Primaria mejora sus destrezas y habilidades en este ámbito **(H3)**.

1.4 Estructura y contenido de la tesis

Este trabajo de investigación está dividido en 4 grandes bloques, dentro de los cuales se incluyen un total de 8 capítulos que se describen a continuación.

El primer bloque se corresponde con la introducción del trabajo en la que se plantea la importancia y adecuación a la situación actual de esta investigación, así como la forma en la que se ha planteado el problema y se ha analizado la literatura correspondiente.

El segundo bloque se corresponde con el marco teórico de la investigación, donde se analizan los aspectos más importantes y relevantes para entender el tema que se trata y la importancia e impacto en el ámbito educativo

El tercer bloque se corresponde con la parte empírica y en él se incluyen los distintos estudios realizados para valorar la inclusión del Pensamiento Computacional en diferentes niveles educativos y el impacto que puede tener tanto en procesos de enseñanza como de aprendizaje de las Matemáticas.

En el cuarto bloque se incluyen todas las conclusiones extraídas de la presente investigación y que emanan directamente de las prácticas realizadas y expuestas en el bloque anterior. Igualmente, así como las limitaciones encontradas y los planteamientos para continuar con futuros trabajos académicos.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

II. Parte teórica

Capítulo 2: ¿Qué es el Pensamiento Computacional?

[2.1 Origen de la idea de Pensamiento Computacional](#)

[2.2 La búsqueda de una definición consensuada](#)

[2.3 Marco conceptual para el Pensamiento Computacional](#)

[2.4 Definición de Pensamiento Computacional](#)

[2.5 Integración del Pensamiento Computacional en el currículo](#)

[2.6 Herramientas de evaluación del Pensamiento Computacional](#)

Capítulo 3: Programación informática y Pensamiento Computacional en el sistema educativo

[3.1 El marco de la alfabetización digital y la código-alfabetización](#)

[3.2 Un movimiento social para desarrollar el Pensamiento Computacional](#)

[Capítulo 4: Lenguajes de programación visual por bloques. Scratch](#)

[4.1 Programación con Scratch en contextos educativos](#)

[4.2 Programación con Scratch para aprender Matemáticas](#)

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Capítulo 2: ¿Qué es el Pensamiento Computacional?

Cuando se habla de Pensamiento Computacional (PC), es imprescindible hacer referencia a las aportaciones dadas en el ámbito de las Ciencias de la computación (CC) por la ingeniera estadounidense Jannette Wing. Sus contribuciones presentadas en una comunicación en la ACM (Association for Computing Machinery de Estados Unidos) de 2006 marcaron un punto de partida muy importante para muchas de las investigaciones realizadas en los últimos 15 años. En este trabajo Wing establece el Pensamiento Computacional como un conjunto de habilidades aplicables de manera universal y que todo estudiante debe desarrollar en su etapa educativa, destacando el hecho de que no debe ser considerado como una competencia exclusiva de un informático (Wing, 2006). Como un antecedente a este importante apunte teórico sobre el concepto de Pensamiento Computacional y la importancia de incorporarlo en el ámbito escolar como parte de un currículo activo, cabe destacar el marco de trabajo aportado por diSessa (2000) en su libro *Changing minds*. Aquí el autor establece una base conceptual para abordar la alfabetización computacional.

A partir de aquí se pueden destacar muchos trabajos que han buscado la unificación en un marco conceptual que sirva de referencia para el desarrollo de experiencias empíricas relacionadas con el Pensamiento Computacional. El

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

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 (National Research Council , 2010). Aquí también se aborda el alcance y naturaleza del PC planteando cinco cuestiones abiertas que ayudan a establecer las bases de un marco conceptual:

1. ¿Cuál es la estructura del Pensamiento Computacional?
2. ¿Cómo se puede reconocer a un pensador computacional?
3. ¿Cuál es la conexión entre la tecnología y el Pensamiento Computacional?
4. ¿Cuál es la mejor pedagogía para el Pensamiento Computacional?
5. ¿Cuál es el rol institucional adecuado de la comunidad informática respecto al Pensamiento Computacional?

En la misma línea, la International Society for Technology in Education (ISTE) y la Computer Science Teacher Association (CSTA) definen el Pensamiento Computacional como un proceso de resolución de problemas que incluye características como la organización lógica de la información y el análisis de datos, la representación de la información a través de la abstracción utilizando modelos y simulaciones, la automatización de soluciones o la generalización y transferencia de los procesos de resolución de problemas a una amplia variedad de situaciones o contextos (ISTE, 2011). A su vez, destacan que estas habilidades ayudan a potenciar una serie de actitudes como la confianza ante situaciones complejas, la persistencia en trabajos difíciles, la tolerancia a la ambigüedad o la capacidad de enfrentarse a problemas abiertos.

2.1 Origen de la idea de Pensamiento Computacional

Desde que Jannette Wing estableciera la idea conceptual de lo que hoy conocemos como Pensamiento Computacional, son muchos los trabajos y aportaciones realizados por autores e instituciones para establecer una definición consensuada de qué es el Pensamiento Computacional y qué supone su trabajo y desarrollo en el ámbito escolar. Pero las aportaciones dadas por Wing (2006) que se han mencionado en el capítulo anterior no son el origen o punto de partida en el desarrollo de este concepto. Si queremos hacer referencia a los primeros pasos del desarrollo de esta idea, hay que

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

remontarse a los trabajos iniciales de Seymour Papert. En su artículo *Teching Children Thinking (1972)* no hizo una definición de lo que se podía entender por Pensamiento Computacional, ni utiliza de manera explícita esas palabras, pero sí establece las bases ideológicas sobre cómo el uso de los ordenadores pueden potenciar el proceso de aprendizaje de los estudiantes, empleándolos como un recurso que el propio estudiante puede manipular, modificar y emplear en el desarrollo de proyectos para tener un mejor conocimiento del mundo que le rodea. Una idea basada en el uso de los ordenadores como recurso para aprender construyendo y para reflexionar sobre lo que hacen. Un enfoque pedagógico que hoy está muy ligado a la filosofía del denominado *Movimiento Maker* en el ámbito educativo, cuyas bases parten de potenciar el proceso de aprendizaje de un estudiante ofreciéndole la posibilidad de emplear distintos recursos tecnológicos para aprender haciendo, creando, desarrollando, inventando y reinventado cosas en el ámbito de la programación y la robótica, entre otras áreas (Halverson y Sheridan, 2014).

Igualmente, esta idea también soporta la teoría del construccionismo de Papert: un enfoque del desarrollo intelectual en el que una persona construye su propio conocimiento elaborando un producto con elementos de su entorno; un proceso de aprendizaje basado en la interacción de la persona con el mundo físico, social y cultural en el que vive (Papert y Harel, 1991). Además, el construccionismo de Papert también se centra en el respeto de los intereses,

motivaciones y ritmo de aprendizaje de cada estudiante. Este enfoque ayuda a dar presencia a un elemento muy importante en el proceso de aprendizaje: la creatividad. Pero no solo como capacidad para elaborar un producto, también creatividad para abordar una lectura o resolver un problema (Obaya, 2003). De la mano de esta teoría educativa, Seymour Papert también desarrolló un lenguaje de programación que es el punto de partida para el desarrollo del modelo de software educativo que ha permitido la democratización de este concepto y que ha facilitado la inclusión curricular del Pensamiento Computacional en todos los niveles educativos: el lenguaje de programación LOGO. Este fue el primer lenguaje de programación diseñado específicamente para niños, fundamentalmente enfocado a trabajar en el ámbito de la Geometría y los movimientos en el plano. Originalmente, este lenguaje estaba destinado a trabajar con un robot formado por una semiesfera de plástico creado en el laboratorio de inteligencia artificial del M.I.T. (Instituto Tecnológico de Massachusetts) junto a Marvin-Minski. Posteriormente, este objeto y concepto evolucionó a controlar con comandos una tortuga en la pantalla del ordenador (Bull, 2005). Pero no solo se creó un lenguaje de programación para hacer esta disciplina más accesible en las escuelas, casi una utopía en sus inicios. Papert define LOGO como un lenguaje de programación que va unido a una filosofía de la educación (Papert, 1999).

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Desde la perspectiva del lenguaje de programación, la evidencia es clara en cuanto a la influencia que ha tenido sobre los lenguajes de programación visuales por bloques que se usan hoy, como Scratch, ya que su diseño y funcionalidad parten de un mismo objetivo: ayudar a los usuarios a crear sus propios proyectos de manera personalizada, atractiva y motivadora (Kafai y Resnick, 1996; Papert, 1980). A la hora de trabajar con LOGO, como con cualquier lenguaje de programación de texto, los estudiantes encontraban su mayor limitación en el hecho de que cada línea de código debe adherirse a restricciones sintácticas. Este frágil entorno puede hacer que los estudiantes centren su atención en la sintaxis del código y pierdan la oportunidad de centrarse en el significado semántico (Lewis, 2010). Sin embargo, en los lenguajes de programación para niños actuales como Scratch los bloques solo se pueden encajar en un sitio acorde a una sintaxis adecuada, por lo que los fallos de programación nunca van a ser debidos a un error de escritura. Esto hace que se aprendan de manera fácil e intuitiva conceptos complejos como los elementos del flujo de control, los bucles y los condicionales (Parsons y Haden, 2007).

Como filosofía de la educación, LOGO y su implementación didáctica tenían como premisa el “deja que ocurra”. Un punto de partida que busca un desarrollo de las capacidades de los estudiantes descubriendo los hechos por sí mismos, adquiriendo y aplicando destrezas con las que poder afrontar

Capítulo 2: ¿Qué es el Pensamiento Computacional?

problemas nuevos. Un proceso que busca que los niños reflexionen sobre cómo pueden hacer una tarea o resolver un problema por sí solos y, por tanto, reflexionar sobre cómo piensan ellos mismos (Clements, 1986). Este enfoque es, en ocasiones, difícil de plasmar en una asignatura tradicional o aplicarlo en un proceso de aprendizaje basado en una metodología clásica o expositiva en la que se busca retener información. Las ideas que plantean requieren otro enfoque metodológico y aunque se planteasen hace más de tres décadas, hoy día todavía suponen un reto para conseguir la formación transversal y completa de nuestro alumnado.

Como reflejo de toda esta filosofía y concepción del uso de la tecnología en el ámbito educativo, Papert (1972) expone el objetivo que debe perseguir la educación para conseguir que la tecnología sea un recurso práctico que permita desarrollar habilidades de pensamiento:

Afirmo que la informática es la fuente más rica que contiene estos intereses. Podemos ofrecer a los niños un poder sin precedentes para inventar y desarrollar emocionantes proyectos proporcionándoles el acceso a los ordenadores, con un lenguaje de programación adecuadamente claro e inteligible y con dispositivos periféricos capaces de producir acciones en línea en tiempo real. (pág. 2)

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Recordando hoy estas palabras, es común referirse a ellas como un punto de partida de lo que hoy se persigue en las aulas de Educación Primaria o Secundaria.

En cierto modo, no es difícil entender que el desarrollo curricular y la inclusión en el ámbito educativo de estas tecnologías educativas tuviesen ciertas dificultades, ya que realmente la programación informática conlleva el uso de lenguajes que para los estudiantes de Educación Primaria y Secundaria puede tener complejidad. Presenta ciertas dificultades y limitaciones como una sintaxis estricta y un significado cerrado para los comandos: el movimiento de la tortuga está limitado a unas cuantas direcciones (Pomper, 1990). A pesar de ser una visión utópica en aquel entonces, Papert no era el único que volcaba sus esfuerzos en creer y desarrollar la idea de que los estudiantes pudiesen aprender a través de la programación con un ordenador. También el distinguido investigador Donald Knuth decía, refiriéndose al aprendizaje de las Matemáticas, que una persona no entiende algo realmente hasta que puede enseñárselo a un ordenador (Knuth, 1974).

En definitiva, con todo este desarrollo que tanto ha influenciado lo que conocemos hoy en el ámbito de la tecnología y con todos los matices filosóficos y pedagógicos que impulsó Papert en las últimas décadas del siglo XX, podemos decir que fijó un punto de partida de lo que hoy conocemos

Capítulo 2: ¿Qué es el Pensamiento Computacional?

como Pensamiento Computacional a partir de dos ideas (Bull, Garofalo, y Hguyen, 2020):

- Programar un ordenador puede facilitar el proceso de aprendizaje de materias como Matemáticas o diferentes lenguas.
- El uso de la programación para facilitar y explorar en el aprendizaje de materias como Matemáticas o las diferentes lenguas requiere el desarrollo de un lenguaje de programación específico.

Como se ha dicho anteriormente, todas estas ideas y planteamientos, todos estos enfoques filosóficos y pedagógicos marcaron un punto de partida muy importante en el desarrollo de las ciencias de la computación en el ámbito educativo y su inclusión curricular, tanto para Educación Primaria como Secundaria. Y en consonancia con esto, en la primera década del siglo XXI, podemos destacar una serie de acontecimientos y trabajos que dieron un paso muy importante en el desarrollo de la idea de Pensamiento Computacional.

En primer lugar podemos destacar el libro publicado por el organismo estadounidense National Academy of Engineering, Committee on Technological Literacy, titulado *Technically Speaking: Why All Americans Need to Know More About Technology* (National Research Council, 2002).

Esta obra parte de una realidad de la sociedad estadounidense que

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

perfectamente puede aplicarse a cualquier otra sociedad occidental: el hecho de que a pesar de haber tenido un importante avance tecnológico en la última década del siglo XX y a principios de XXI en nuestras vidas, la sociedad no está preparada para reflexionar o abordar los retos que plantea este desarrollo tecnológico. Se hace una llamada a la adecuada administración del cambio tecnológico en la sociedad y a una correcta alfabetización digital, término muy ligado al desarrollo del Pensamiento Computacional. Además, no solo lo destacan como una idea importante en el ámbito educativo y como un planteamiento para el desarrollo completo de los estudiantes. También destacan que la alfabetización tecnológica es muy importante para líderes de empresas, gobernantes y medios de comunicación que toman e influyen en decisiones importantes para la sociedad.

En segundo lugar, es importante destacar la publicación de otro libro por parte del National Research Council de Estados Unidos titulado *Computer Science: Reflections on field, reflections from the field* (National Research Council, 2004). En este libro se parte de una situación que marcó el cambio de siglo en nuestra sociedad en cuanto a la tecnología se refiere: lo que hacía una década pertenecía al dominio de expertos y a personas aficionadas con altos conocimientos de sistemas informáticos, pasó a extenderse en un uso cotidiano permitiendo que todo el mundo pudiera realizar prácticamente cualquier acción de su vida diaria utilizando algún recurso tecnológico. Al

Capítulo 2: ¿Qué es el Pensamiento Computacional?

principio, incluso las acciones más habituales de hoy como mandar un correo electrónico o revisar noticias en red estaban destinadas a pequeños sectores de la sociedad. El cambio en esta dirección ha marcado muchos aspectos de nuestra vida y, en ocasiones, nos mantiene sujetos a ciertas conductas prediseñadas para propiciar unos u otros hábitos sociales, de consumo, ocio o algunos incluso relacionados con los procesos de enseñanza y aprendizaje. De ahí la necesidad de establecer un marco de referencia para marcar la evolución y concepción de qué es la ciencia de la computación y cómo abordar los distintos aspectos asociados a ella. Algo muy importante si nos lo planteamos desde la perspectiva y necesidad de desarrollar una correcta educación en el uso de la tecnología de manera práctica, constructiva y que favorezca el desarrollo de nuestro alumnado desde su Educación Primaria. En este sentido, destacan las palabras de Gerald Sussman en las que caracteriza esta revolución como un cambio intelectual con un efecto en la cultura comparable al que marcó en su época el desarrollo de la Geometría o el Cálculo. La revolución informática es una revolución en la forma de pensar y de expresar lo que pensamos (Abelson y Sussman, 1996).

Sin duda, estas publicaciones fueron un precedente fundamental para impulsar el desarrollo de la idea de Pensamiento Computacional originada por Papert de manera más extensa, conforme a un escenario tecnológico que se avecinaba tan propicio como necesario en el ámbito educativo. Por ello, se entiende que

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

en los años posteriores hubiese cierta discusión en cuanto a la definición tanto conceptual como de contexto sobre este término. De ahí que fuesen muchos los trabajos que buscaron establecer un consenso y formalizar la definición de Pensamiento Computacional.

2.2 La búsqueda de una definición consensuada

Como se ha citado al inicio de la sección anterior, referente a los primeros indicios de la idea de Pensamiento Computacional, el origen de una búsqueda consensuada o el principio de la carrera por dar una definición de Pensamiento Computacional por parte de universidades, empresas o investigadores, según el punto de vista que se quiera considerar, está en la definición que dio Jannette Wing en 2006. Su artículo ofreció una descripción muy amplia y contemplaba muchos de los beneficios que ofrece el Pensamiento Computacional. A partir de aquí se dieron muchas definiciones que, en cierto modo, podían considerarse como superficiales, sin mucho detalle ni aportes significativos. Por otro lado, también surgieron definiciones con excesivos matices y de difícil comprensión que no han tenido mucha aceptación dentro de profesionales del ámbito educativo que no son expertos en el campo de la informática.

Desde este punto, podemos citar varios trabajos que han intentado buscar una definición con el objetivo de consensuar las principales ideas de la gran

mayoría de definiciones, desarrollos conceptuales y contextos teóricos publicados en torno a la idea de formalizar una definición para el Pensamiento Computacional.

En primer lugar, se pueden destacar las aportaciones dadas por Cynthia Selby (2013), realizando un trabajo en el que llevó a cabo una revisión bibliográfica analizando publicaciones referentes al Pensamiento Computacional desde 2006, identificando los términos, descripciones y significados más repetidos. Destaca la aparición sistemática de tres términos, lo cual se puede tomar como un indicio necesario para establecer un consenso sobre la definición de Pensamiento Computacional. Concretamente, se trata de la idea de que es un proceso de pensamiento, la aparición del concepto de abstracción y el concepto de descomposición.

Otra labor importante en este aspecto de revisión desde la publicación del trabajo de Jannette Wing en 2006 es el análisis realizado por Grover y Pea (2013). En esta revisión se desglosa la influencia de los términos y conceptos lanzados por Wing años atrás, abordando distintas propuestas de definiciones y tratando temas importantes como la necesidad de formalizar un proceso de evaluación para el Pensamiento Computacional de cara a una inclusión curricular efectiva. Para poder tener una referencia que permita dar una definición adecuada que ayude a caracterizar los principales elementos y

prácticas del Pensamiento Computacional, es importante tener un marco conceptual adecuado. Igualmente, esto permite plantear la inclusión curricular, su proceso de evaluación y establecer una relación adecuada con otros elementos curriculares.

2.3 Marco conceptual para el Pensamiento Computacional

Derivado de todos estos estudios que buscaban dar forma a una definición para el Pensamiento Computacional a partir de las aportaciones de Jannette Wing y el resto de publicaciones en este campo, tanto de investigadores como de entidades dedicadas al mundo de la informática, Karem Brennan y Mitchel Resnick establecen las bases para un marco conceptual que permita dar forma tanto a la definición del Pensamiento Computacional como al diseño de experiencias educativas. Es importante destacar que este marco se ha definido en el contexto de trabajo particular del lenguaje de programación visual por bloques Scratch, al cual se hará referencia de manera extendida en el capítulo 4, y tomando también como referente las prácticas analizadas con este tipo de lenguajes de programación visual. Hoy son muchos los lenguajes de este tipo disponibles para acercar la programación a los estudiantes en edades tempranas, aunque Scratch es uno de los más populares. Como veremos en el capítulo 4 de este trabajo con mayor profundidad, en realidad Scratch es algo más que software educativo. Se trata de un lenguaje de programación

Capítulo 2: ¿Qué es el Pensamiento Computacional?

diseñado por el MIT Media Lab para docentes y estudiantes, que ayuda a usar la programación para el desarrollo de estrategias de razonamiento (Resnick et al., 2009).

Este marco conceptual, conocido como el marco Brennan-Resnick, describe las principales dimensiones para el Pensamiento Computacional: *conceptos computacionales*, *prácticas computacionales* y *perspectivas computacionales* (Brennan y Resnick, 2012).

Los *conceptos computacionales* son esa serie de conocimientos o ideas técnicas básicas para trabajar con cualquier lenguaje de programación y que se pueden extrapolar a otros contextos que no requieran programar con un ordenador: secuencias, bucles, paralelismo, eventos, condicionales, operadores y datos.

Considerar únicamente conceptos computacionales para intentar definir prácticas de Pensamiento Computacional es algo insuficiente. Por lo que en este marco se define la idea de *prácticas computacionales* para hacer también referencia al proceso de construcción y tener una concepción más amplia del proceso de aprendizaje que conllevan este tipo de prácticas educativas. Las prácticas computacionales se centran en el proceso de razonamiento y aprendizaje, no solo contemplando qué se aprende, sino también cómo se aprende. Es un planteamiento en el que el proceso de aprendizaje se desarrolla

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

a través de un “hacer” computacional (Hemmendinger, 2010) para involucrar más al alumnado en el proceso de abstracción y otras prácticas que favorezcan el desarrollo de habilidades de Pensamiento Computacional (Basu et al., 2016). Por prácticas computacionales, este marco contempla el ser gradual e iterativo, probar y corregir, reutilizar y remezclar y la abstracción y la capacidad de modular.

Para completar el enfoque que permite articular de manera más completa este marco para el Pensamiento Computacional, los autores añaden la dimensión denominada *perspectivas del Pensamiento Computacional*. Esta componente se centra en la comprensión por parte del estudiante de su propio proceso de aprendizaje, su relación con los demás y con la propia concepción y uso de la tecnología que ellos y ellas pueden hacer. En esta dimensión se incluye la expresión, conexión con otros usuarios y la acción de preguntar.

Este marco conceptual nace de las prácticas y posibilidades que aporta el uso de un lenguaje de programación visual por bloques para desarrollar habilidades de Pensamiento Computacional. Pero, en cierto modo, para favorecer una integración curricular efectiva es interesante hacer referencia a un marco conceptual que parta de una perspectiva más general como es la de la enseñanza de las ciencias o al enfoque de prácticas STEM (Science, Technology, Engineering and Mathematics), por su acrónimo en inglés, o

CTIM (Ciencia, Tecnología, Ingeniería y Matemáticas) en nuestra lengua. El Pensamiento Computacional es un conjunto de habilidades transversales aplicables en cualquier contexto educativo. Pero más aún es un enfoque STEM, ya que es con el apoyo de estas disciplinas donde su aplicación tiene un efecto más directo pudiendo trabajar desarrollando prácticas educativas transversales que potencien habilidades de resolución de problemas (Li et al., 2020).

Por ello, es importante hacer referencia al marco sugerido por Grover y Pea (2018) en el que se sitúa al Pensamiento Computacional como un elemento mediador en las prácticas de ámbito STEM. Este marco conceptual establece una serie de elementos, divididos en conceptos y prácticas, que ayudan a entender qué es el Pensamiento Computacional y qué proceso de razonamiento supone. Como conceptos de Pensamiento Computacional se incluyen la lógica y el pensamiento lógico, los algoritmos y el pensamiento algorítmico, los patrones y el reconocimiento de patrones, la abstracción y generalización, la evaluación y la automatización. Las prácticas de Pensamiento Computacional a las que se refiere este marco conceptual intentan acercar al trabajo que puede realizar un informático cuando se enfrenta a la resolución de un problema. Se considera dentro de esta categoría la descomposición de un problema, la creación de artefactos computacionales, probar y depurar, la mejora reiterada y la colaboración y creatividad.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Ambos marcos conceptuales hacen referencia a la importancia de la relación del Pensamiento Computacional y la informática con elementos principales de cualquier contexto curricular, más aún si lo planteamos desde el ámbito científico-tecnológico o cualquiera de las disciplinas STEM. Además, a lo largo de la última década, han servido para dar forma a una definición de mayor consenso y a la terminología correspondiente para referirnos al Pensamiento Computacional y para diseñar prácticas educativas en un contexto adecuado.

2.4 Definición de Pensamiento Computacional

Hemos visto que el término de Pensamiento Computacional comenzó a concebirse como tal en la década de los 80 con los trabajos e investigaciones de Seymour Papert y que, en adelante, se dieron muchos esfuerzos para revitalizar este concepto; sobre todo, para alcanzar el objetivo de democratizar los conocimientos informáticos como un conjunto importante de conocimientos que los alumnos necesitan tener para afrontar los retos del Siglo XXI.

Anteriormente se hizo referencia a la definición dada por la investigadora Jannet Wing en el año 2006 y que supuso un punto de inflexión muy importante para que el desarrollo del Pensamiento Computacional fuese concebido como una destreza fundamental para cualquier estudiante del Siglo

XXI y no como algo dedicado a personas que trabajasen en el mundo de la informática. Posteriormente, la propia investigadora destacaba estos aspectos subrayando que si el Pensamiento Computacional se usa en diferentes contextos, incluso marcando cierta influencia en la investigación de muchas disciplinas tanto de ciencias como humanidades (Bundy, 2007), todo el mundo debería conocerlo de manera directa o indirecta, lo cual supuso un reto educativo (Wing, 2008).

En líneas muy generales, Hemmendinger (2010) da una definición para el Pensamiento Computacional en la que lo plantea como la acción de pensar como un economista, como un físico o como un artista y entender cómo utilizar la computación para resolver sus problemas y para crear y descubrir nuevas cuestiones que puedan ser exploradas. Posteriormente, Jan Cuny, Larry Snyder y Jeannette Wing definían el Pensamiento Computacional de manera más completa y acorde a las directrices dadas en los años anteriores como un proceso de pensamiento desarrollado en la formulación de problemas y en sus soluciones de manera que estas son representadas de tal forma que pueden ser desarrolladas por un procesador de información (Wing, 2011); una definición en la que contempla de manera acertada el trabajo de la creación en el ámbito de la informática. En este punto, fue interesante la visión que aportaban distintos profesionales de la educación para sugerir la importancia de comenzar con ejemplos prácticos sobre qué entendemos por Pensamiento

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Computacional y utilizarlos para identificar eso a lo que se hace referencia cuando se habla de abstracción, automatización y análisis, ya que es especialmente útil para entender cómo los estudiantes pueden enfrentarse a estos problemas novedosos (Lee et al., 2011).

En una publicación de la Royal Society para abordar el tema de cómo debería desarrollarse el ámbito de la computación en las escuelas de Reino Unido, se define el Pensamiento Computacional de manera igualmente general, y en cierto modo algo abstracta, como el proceso para reconocer aspectos relacionados con la computación en el mundo que nos rodea, así como utilizar herramientas y técnicas propias de la informática para entender y razonar sistemas y procesos naturales y artificiales (Furber, 2012). En estos años Wing continuaba estableciendo una definición haciendo otro aporte más genérico en el que dice que el Pensamiento Computacional es un proceso de pensamiento utilizado para formular un problema y expresar su solución o soluciones en términos que una computadora pueda aplicar de manera efectiva (Wing, 2014).

Otros autores como Yadav et al. (2014) mantenían la línea de trabajo de Wing e intentan aportar definiciones más concretas, considerando el Pensamiento Computacional como una actividad mental para la abstracción de problemas y la formulación de sus posibles soluciones de manera tal que puedan ser

Capítulo 2: ¿Qué es el Pensamiento Computacional?

automatizadas. Además, acompañan esta definición de una realidad que cada vez va tomando más relevancia en esos años al considerar el Pensamiento Computacional como una habilidad fundamental para el Siglo XXI y de ahí la importancia de incorporarlo en distintas áreas educativas introduciendo los conceptos básicos de manera progresiva desde la Educación Primaria.

En este punto, la definición formal de Pensamiento Computacional estaba todavía sin consensuar de manera definitiva. Se mantenía como un tema abierto y de discusión permanente, pero aún así se daba el acuerdo de que el Pensamiento Computacional incluye un pensamiento algorítmico, un manejo de múltiples niveles de abstracción, la descomposición de problemas en partes más pequeñas y la representación de datos mediante modelos (Jacob y Warschauer, 2018). También se pueden destacar otras contribuciones como la de Mannila et al. (2014) en la que definen el Pensamiento Computacional como un conjunto de conceptos y procesos de pensamiento propios de la informática que ayudan en la formulación de problemas y sus soluciones en diferentes disciplinas.

Desde organismos como la Computer Science Teachers Association (CSTA, 2016), en su marco conceptual de estándares de aprendizaje, se contempla una definición para el Pensamiento Computacional en el que pone el foco principal en su transversalidad y desarrollo en distintos hábitos, manteniendo

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

su núcleo conceptual en que es una metodología para la resolución de problemas. Se establece el Pensamiento Computacional como una metodología de resolución de problemas que amplía el campo de actuación de la informática en todas las disciplinas, proporcionando así una manera distinta de analizar y desarrollar soluciones a problemas que pueden ser resueltos aprovechando el potencial de un ordenador. Con un enfoque centrado en la abstracción, la automatización y el análisis, el Pensamiento Computacional es un elemento principal en la disciplina informática, por lo que debe estar entrelazada en todos los estándares de este área a todos los niveles educativos.

En los últimos años se continúan sucediendo definiciones, algunas en un contexto más informal como la de Wolfram (2016) . Este autor se refiere al Pensamiento Computacional diciendo que su núcleo intelectual consiste en formular las cosas con al suficiente claridad, y de manera suficientemente sistemática, como para poder decirle a un ordenador cómo hacerlas. Estableciendo cierta analogía, y partiendo de que el pensamiento matemático consiste en formular las cosas de manera que uno pueda manejarlas matemáticamente, mantiene que el Pensamiento Computacional es algo más amplio que este proceso ya que hay muchas más cosas o situaciones que pueden manejarse de manera computacional. Más reciente, y haciendo un análisis de los trabajos de investigación en la bibliografía sobre Pensamiento

Capítulo 2: ¿Qué es el Pensamiento Computacional?

Computacional y su definición, Moreno et al. proponen una definición para el Pensamiento Computacional concibiéndolo como la capacidad de formular y representar problemas para resolverlos mediante el uso de herramientas, conceptos y prácticas de la disciplina informática, como la abstracción, la descomposición o el uso de simulaciones (Moreno-León et al., 2019).

Todas estas definiciones que se han citado tienen como elementos principales y mayormente coincidentes el hecho de relacionar el Pensamiento Computacional en un contexto referido a la resolución de problemas, al uso de la tecnología y a un proceso de pensamiento. Dentro de la idea de resolución de problemas, destacan la capacidad de abstracción y de análisis. Esto hace que se pueda decir que el hecho de que la definición de Pensamiento Computacional esté tan relacionada con la capacidad para trabajar la resolución de problemas se debe a que está basado en este mismo proceso. Pero aunque la actual definición de Pensamiento Computacional esté en relación con la resolución de problemas con ayuda de la tecnología, también es necesario introducir otras dimensiones que son igualmente efectivas. En este sentido es necesario examinar el Pensamiento Computacional a partir de factores personales, ambientales, sociales, afectivos, psicológicos y éticos en un entorno multidimensional y más allá de la situación actual con el fin de mejorar el impacto y uso práctico en nuestra sociedad (Haseski et al., 2018).

2.5 Integración del Pensamiento Computacional en el currículo

Todo el desarrollo conceptual del Pensamiento Computacional expuesto en las secciones anteriores va de la mano, como se ha citado en la sección 2.3, de la importancia de enfocar toda esta evolución a una adecuada inclusión e integración curricular del Pensamiento Computacional en algunos países de Europa (Balanskat y Engelhardt, 2015), tanto en Educación Secundaria como en Primaria.

Pero la idea de una inclusión curricular del Pensamiento Computacional no solo debe contemplar la introducción en el currículo de actividades de programación con software educativos adaptados a cada nivel. Hay que pretender ciertos objetivos y mantener una relación estrecha con otros elementos curriculares. No conviene establecer la premisa de que realizar actividades de Pensamiento Computacional ayudará a los estudiantes a desarrollar habilidades de este ámbito que puedan ser transferidas a otras áreas de conocimiento o materias sin más; la transferencia del aprendizaje a través de los contextos no se produce automáticamente. Las ciencias del aprendizaje defienden que la transferencia debe estar mediada por técnicas empíricamente establecidas que exigen, entre otras cosas, introducir conexiones explícitas entre los contextos de aprendizaje original y de transferencia (Grover, Cooper y Pea, 2014).

Capítulo 2: ¿Qué es el Pensamiento Computacional?

A la hora de contemplar esta inclusión curricular, no solo hay que centrarse en la importancia de una adecuada competencia asociada al Pensamiento Computacional. La Comisión Europea destaca la importancia de que los ciudadanos adquieran en general un conjunto de competencias clave necesarias para el desarrollo personal, la inclusión social, la ciudadanía activa y el empleo. Entre otras, se incluyen la competencia digital y la competencia en resolución de problemas (European Commission, 2016). Para promover la adquisición de estas destrezas, incluyendo el ámbito educativo, el programa denominado *The Digital Competence Framework for Citizens*, también conocido como *DigComp*, supone una herramienta para mejorar la competencia digital de los ciudadanos, ayudar a los responsables políticos a formular políticas que apoyen el desarrollo de la competencia digital y planificar iniciativas de educación y formación para mejorar la competencia digital en contextos específicos (Riina, Yves, Gómez y Brade, 2016).

Manteniendo el foco en esta línea, es fácil entender que la inclusión curricular efectiva del Pensamiento Computacional debe estar estrechamente ligada a la adquisición de un conjunto de habilidades asociadas con la competencia digital que necesita todo estudiante para desarrollarse en su vida personal y profesional en el Siglo XXI. Por lo tanto, es conveniente que exista una relación entre la competencia digital y el Pensamiento Computacional tal y como establece Juškevičiene y Dagiene (2018) (Figura 1). Para ello, plantea

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

un análisis de las palabras, términos o conceptos que más aparecen en la bibliografía reciente. Las expresiones o palabras que observa con mayor frecuencia en las definiciones son resolución de problemas, abstracción, ordenador, proceso, ciencia, eficaz, datos, algoritmo, conceptos, habilidad, herramientas y análisis. Partiendo de este análisis crea ocho grupos de conceptos para describir el Pensamiento Computacional y relacionarlo con la competencia digital: análisis y representación de datos, artefactos computacionales, descomposición, abstracción, algoritmos, comunicación y colaboración, informática y sociedad, y evaluación. A continuación, se muestran las principales características con las que plantea cada uno de estos grupos.

El primero, análisis y representación de datos, hace referencia a la recogida y posterior análisis y representación de datos, incluyendo también el concepto de generalización como la forma de obtener información a partir de similitudes entre objetos (Krauss y Protsman, 2017).

Los artefactos computacionales hacen referencia al aspecto creativo de la computación; consiste en el diseño y desarrollo de estos artefactos y de la aplicación de técnicas computacionales para resolver problemas de forma creativa.

Capítulo 2: ¿Qué es el Pensamiento Computacional?

El tercer grupo, la descomposición, hace referencia a la división de una tarea en partes más pequeñas y sencillas.

La abstracción, para la que también toma la referencia de Krauss y Protsman (2017), supone la capacidad de dar una solución a un problema de forma genérica, sin atender a características específicas de una determinada situación.

El grupo de algoritmos puede ser uno de los más importantes dentro del Pensamiento Computacional, ya que puede incluir características de muchos grupos como la automatización. Un algoritmo es un conjunto de pasos debidamente ordenados para llegar a una solución de un problema.

El sexto grupo, la comunicación y colaboración, es fundamental en el desarrollo de la informática, ya que es importante tener la habilidad de comunicarse y trabajar con otras personas para llegar a un fin común. Esto también tiene relación con otros grupos como la creación de artefactos computacionales, lo cual es casi inviable si no se desarrolla con un trabajo en equipo.

El grupo de informática y sociedad hace referencia a la influencia que tiene este campo tanto en la sociedad como en los individuos que la conforman. También supone tener en cuenta el uso responsable que debemos hacer como

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

sociedad de la tecnología, así como de nuestro deber a estar debidamente formados y capacitados para ello.

Por último, el proceso de evaluación es el que lleva a asegurar una solución dada por un algoritmo, obtenida durante un proceso de trabajo o por un sistema analítico.

Con todos estos grupos debidamente definidos, se establece una relación con los distintos elementos que caracterizan la competencia digital según el marco europeo para esta destreza en educación (DigCompEdu) (Redecker, 2017): información y alfabetización mediática, comunicación digital y colaboración, contenido digital, uso responsable y resolución digital de problemas.

Como resultado de esta relación, Juškevičiene y Dagiene plantean el siguiente gráfico:

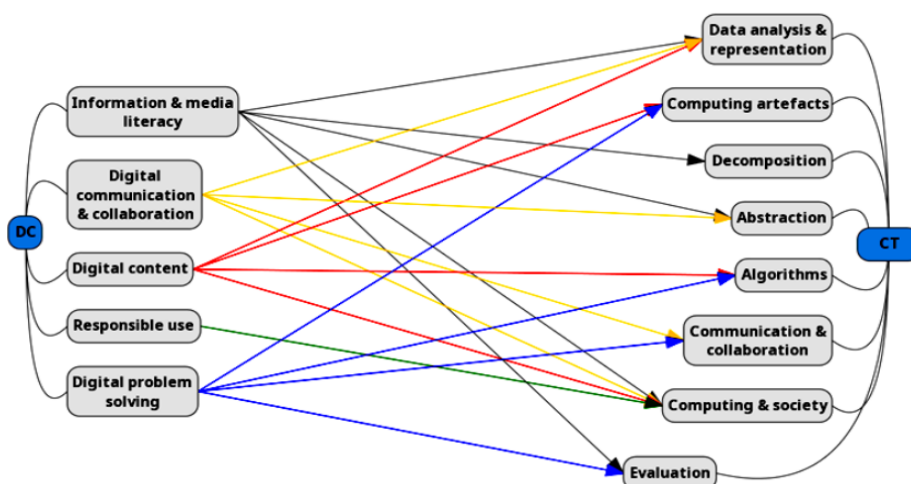


Figura 1: Conexión entre el PC y la CD. Fuente: (Juškevičiene y Dagiene, 2018)

Pero esta conexión no establece por sí sola una transferencia directa entre el desarrollo de la competencia digital y el desarrollo de habilidades asociadas al Pensamiento Computacional. Es importante porque ayuda a relacionar diferentes elementos del currículo, pero hay que tener en cuenta que estas conexiones no justifican una inclusión curricular de cualquier modo. Por lo tanto, conviene concretar qué es importante en la inclusión curricular del Pensamiento Computacional. Angeli y Giannakos (2020) observan que los estudios realizados en este campo apuntan a que, con el fin de hacer que el Pensamiento Computacional sea un elemento curricular de peso en el ámbito educativo, se necesita invertir más esfuerzo en investigaciones que aborden las siguientes cuestiones:

- Definir las competencias de Pensamiento Computacional para cada nivel educativo.
- El uso de metáforas en la enseñanza de conceptos de Pensamiento Computacional de manera eficaz y eficiente.
- El uso de estrategias pedagógicas y tecnológicas para enseñar el Pensamiento Computacional.
- El desarrollo profesional del profesorado en Pensamiento Computacional.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

- La evaluación de competencias y habilidades de Pensamiento Computacional.

Destaca la importancia de la formación y desarrollo del profesorado en este ámbito, ya que es algo fundamental y prioritario el hecho de que el profesorado sepa cómo diseñar actividades y experiencias de aprendizaje adecuadas, cómo enseñar, cómo evaluar y cómo utilizar los recursos disponibles de manera adecuada para enseñar los conceptos del Pensamiento Computacional. Pero esto no solo indica que es necesaria una adecuada formación continua del profesorado en este campo, sino que también supone que el profesorado debe formarse en esta materia desde su formación inicial con el objetivo de ayudarles a tener una concepción más precisa y concreta de su aplicación en el aula (Yadav et al., 2017).

En Europa, desde un punto de partida institucional, los intentos para una inclusión formal en el currículo educativo ha tenido importantes puntos a lo largo de la última década. Uno de los ejemplos más destacados puede ser la implantación que ha tenido en Reino Unido, donde desde 2012 podemos ver directrices específicas y completas (Berry, 2013), así como procesos de análisis y reflexión necesarios (Furber, 2012), para llevar a cabo esta inclusión. Desde el gobierno francés, en 2013 se destacaba la idónea situación que se daba en el ámbito social para trabajar en la escuela la verdadera

enseñanza de la informática debido a diversos factores como el desarrollo de la industria, el desarrollo de la tecnología en el ámbito educativo o la evolución de procesos de enseñanza en línea (Abiteboul et al., 2013). En España, podemos destacar la publicación de un informe por parte de diferentes entidades y Universidades en 2015 en el que se la enseñanza de la informática en nuestro país desde Educación Primaria hasta Bachillerato. En este informe se pone de manifiesto tanto la situación en ese momento de la informática como la importancia de las habilidades y destrezas necesarias para estudiantes de cada nivel educativo, así como la importancia del marco para la competencia digital (Meseguer et al., 2015).

2.6 Herramientas de evaluación del Pensamiento Computacional

A pesar de la sucesión de definiciones, aceptaciones o marcos conceptuales que se fueron sucediendo desde que Jannette Wing abriese el camino para la conceptualización del Pensamiento Computacional tal y como lo tenemos concebido hoy, podemos decir que no hay una única definición o un único marco que recoja todas sus características o que esté universalmente aceptado como para que se derive una aceptación clara de cómo puede evaluarse el Pensamiento Computacional. La evaluación es un aspecto fundamental para su correcta inclusión curricular, ya que poder llevar a cabo un control del

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

progreso de un estudiante a la hora de adquirir una serie de competencias es lo que posibilita una formación continuada en este área.

Si planteamos diferenciar tipos de herramientas para evaluar el Pensamiento Computacional, podemos encontrar una categorización inicial (Román-González et al., 2017) en función de los elementos que aborda y el planteamiento o formato que presenta para su resultado, diferenciándose los siguientes tipos:

- Herramientas sumativas de PC, entre las que se pueden diferenciar pruebas aptitudinales y herramientas para evaluar el conocimiento del contenido.
- Herramientas formativas-iterativas, recursos que devuelven un feedback automático a los usuarios para que mejoren sus prácticas computacionales y suelen ser específicas de un lenguaje o entorno de programación.
- Herramientas de transferencia de habilidades de Pensamiento Computacional. Estas herramientas buscan la forma en la que las habilidades de Pensamiento Computacional adquiridas con una determinada práctica pueden ser aplicadas en diferentes tipos de problemas de la vida real.

Capítulo 2: ¿Qué es el Pensamiento Computacional?

- Escalas de percepción de actitudes de Pensamiento Computacional, la cual se utiliza para determinar los niveles adquiridos en diferentes destrezas.
- Evaluaciones de vocabulario de Pensamiento Computacional, que se utilizan para medir elementos y dimensiones del Pensamiento Computacional expresadas verbalmente.

El hecho de ver diferentes herramientas que atienden a distintos aspectos relacionados con el Pensamiento Computacional, está en consonancia con las diferentes aceptaciones de la definición de este concepto que encontramos en la correspondiente bibliografía.

De entre los distintos tipos de herramientas clasificadas, en las partes empíricas que se incluyen en el presente trabajo de investigación se han incluido herramientas de transferencia de habilidades para analizar la adquisición de contenidos de Matemáticas trabajados con herramientas de Pensamiento Computacional y una herramienta de carácter sumativo y aptitudinal para evaluar el conocimiento en contenidos de Pensamiento Computacional: el test de Pensamiento Computacional o Computational Thinking test (Román-González, 2015). El objetivo de esta herramienta es medir la capacidad de crear y resolver problemas partiendo de los conceptos

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

básicos de programación y el uso de la sintaxis lógica de estos lenguajes como bucles, funciones, variables o condicionales (Chan et al., 2020). Este test se presenta como una de las herramientas actualmente más populares basadas en una evaluación con un lenguaje de programación por bloques, aunque no está ligada a ningún programa en particular (Cutumisu et al., 2019). En su formato básico, el test está compuesto de 28 preguntas de opción múltiple diferenciadas en categorías según se traten conceptos computacionales como el uso de direcciones básicas y secuencias, bucles de repetición con o sin condicionales, condicionales simples y compuestos (si, si-no), condicionales “mientras” y funciones simples. Estos conceptos están asociados con los estándares dados por *Computer Science Teacher Asociation*, los cuales se revisan y adaptan continuamente, y en base al marco de Brennan & Resnick citado en el apartado 2.3. (Brennan & Resnick, 2012). En función del nivel educativo en el que se quiera emplear esta herramienta, existen diferentes versiones. Por ejemplo, para niveles superiores como Bachillerato o primeros cursos de estudios universitarios es conveniente emplear la versión del test de Pensamiento Computacional que incluye problemas Bebras. Estos problemas se basan en el denominado concurso Bebras, un evento internacional que se desarrolla de manera online cada año y cuyo objetivo es la promoción del Pensamiento Computacional. Además, permite analizar si las habilidades de Pensamiento Computacional se aplican o transfieren correctamente a

Capítulo 2: ¿Qué es el Pensamiento Computacional?

situaciones cotidianas (Dagiene y Futschek, 2008). El proceso de validación del test se ha llevado a cabo con un grupo de 20 expertos, 14 hombres y 6 mujeres, procedentes de varios grupos de investigación y asociaciones relacionadas con el fomento y desarrollo del Pensamiento Computacional en el ámbito educativo (Román-González, 2015).

Está claro que es necesario integrar el Pensamiento Computacional en cualquier asignatura de diferentes niveles educativos de manera transversal. Por lo tanto, existe la necesidad de disponer de herramientas de evaluación que permitan al profesorado mantener un seguimiento en el proceso de aprendizaje de su alumnado. Por un lado, es conveniente que el profesorado disponga de recursos para llevar a cabo una evaluación sumativa con herramientas como la expuesta en este capítulo, la cual también puede ser utilizada por investigadores en diseños experimentales con pre-post test (Tang et al., 2020) . Pero estos autores también apuntan a que la evaluación con el alumnado se puede llevar a cabo con otros fines, usando otros recursos como un portfolio que permita conocer la impresión de un estudiante en el desarrollo de su trabajo, así como las entrevistas. Este último es un recurso de gran utilidad en ambos contextos. Para el profesorado siempre es útil conocer la impresión de un grupo de trabajo para el desarrollo de actividades de aula. Para los investigadores es una herramienta indispensable ya que es un complemento para toda la información sumativa recabada con otras

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

herramientas en el estudio de casos que permite conocer más a fondo el proceso de resolución de problemas que han desarrollado los participantes con los diferentes retos que han podido encontrar en el desarrollo de prácticas de Pensamiento Computacional.

Capítulo 3: Programación informática y Pensamiento Computacional en el sistema educativo

Durante la última década, han sido mucho los docentes que en nuestro país han trabajado desarrollando prácticas en distintos niveles educativos para incluir el Pensamiento Computacional como elemento transversal. Esto es un reflejo de la importancia de incluir este recurso en el currículo educativo y como tal, queda reflejado en la normativa educativa de las distintas comunidades autónomas. Por ejemplo, en Andalucía estaba la Orden del 14 de julio de 2016 por la que se desarrolla el currículo de Educación Secundaria Obligatoria (Boletín oficial de la Junta de Andalucía, 2016) y Bachillerato (Boletín oficial de la Junta de Andalucía, 2016) se incluían contenidos relacionados con programación y robótica en las siguientes asignaturas:

- Tecnología aplicada (1º de E.S.O., libre configuración autonómica).
- Tecnología (2º y 3º de E.S.O., materia específica).
- Tecnología (4º de E.S.O., materia troncal)
- Tecnología Industrial I y II, Tecnologías de la Información y Comunicación I y II, Programación y Computación (Bachillerato, libre configuración)

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Además, para Educación Primaria, la Instrucción 12/2019 del 27 (Consejería de Educación y Deporte, 2019) de junio establece la apertura en el horario de asignaturas troncales que permita el desarrollando de la capacidad de resolución de problemas, fortaleciendo habilidades y destrezas de razonamiento matemático a través de la robótica educativa, aspecto que está estrechamente ligado al desarrollo de habilidades propias del Pensamiento Computacional. En 2021, la Orden del 15 de enero de 2021 (Boletín Oficial de la Junta de Andalucía, 2021) actualiza contenidos relacionados con el Pensamiento Computacional y la robótica para la Educación Secundaria en la que, además de definir diferentes materias para trabajar en este campo, como recomendaciones didácticas se especifica que las programaciones didácticas de las distintas materias y ámbitos de Educación Secundaria Obligatoria incluirán actividades que estimulen la motivación por la integración de la robótica y el Pensamiento Computacional, entre otros contenidos transversales. En esta misma orden se define una asignatura de libre configuración denominada Computación y Robótica para la etapa de Educación Secundaria Obligatoria.

En otras comunidades autónomas también se encuentran otros ejemplos de inclusión curricular para el Pensamiento Computacional. En Islas Baleares, según el Decreto 34/2015 y 35/2015 de 15 de mayo, por el que se establece el currículo de Educación Secundaria Obligatoria (Boletín Oficial las Illes

Capítulo 3: Programación informática y Pensamiento Computacional en el sistema educativo

Balears, 2016) y Bachillerato (Boletín Oficial de las Illes Balears, 2015) respectivamente, se definen las siguientes materias en las que se incluyen contenidos de Pensamiento Computacional:

- Tecnología (4º E.S.O., asignatura troncal)
- Tecnologías de la Información y la Comunicación I y II (1º y 2º de Bachillerato, materia específica)

En Castilla y León, la Orden EDU/589/2016, de 22 de junio (Boletín Oficial de Castilla y León, 2016) recoge las siguientes materias de Educación Secundaria Obligatoria:

- Control y robótica (3º de E.S.O., libre configuración autonómica)
- Programación informática (4º de E.S.O., libre configuración autonómica)

En Castilla-La Mancha, el Decreto 40/2015 del currículo de Secundaria de esta comunidad autónoma (Diario Oficial de Castilla-La Mancha, 2009) recoge la siguiente asignatura:

- Robótica (4º de E.S.O., libre configuración autonómica)

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

En Cataluña, el Decreto 119/2015 del 23 de junio (Generalitat de Catalunya, 2015a) recoge contenidos de programación, robótica educativa y Pensamiento Computacional en las áreas de Matemáticas y conocimiento del medio en Educación Primaria. Para Educación Secundaria, el Decreto 187/2015 del 25 de agosto (Generalitat de Catalunya, 2015b) incluye la programación y robótica dentro de del currículo de las asignaturas de Tecnología para 2º, 3º y 4º de E.S.O. y para T.I.C. de 4º de E.S.O.

En la comunidad Valenciana, en el Decreto 87/2015 del 5 de junio (Diari Oficial de la Generalitat Valenciana, 2015) se incluye el Pensamiento Computacional en las asignaturas de T.I.C. para 4º de E.S.O. y 1º y 2º de Bachillerato y para Tecnología en 1º, 2º y 3º de E.S.O.. Además, la asignatura de informática de libre configuración autonómica también aborda contenidos de Pensamiento Computacional en 1º y 3º de E.S.O. En el caso de esta comunidad autónoma vemos que el alumnado de cualquier curso de Educación Secundaria puede trabajar este tipo de contenidos.

En Galicia, la Orden del 15 de julio de 2015 (Diario Oficial de Galicia, 2015) recoge la asignatura de Programación dentro del ámbito de libre configuración autonómica. Esta asignatura se oferta en 1º y 2º de E.S.O. y aporta conocimiento informáticos necesarios para resolver problemas diseñando algoritmos y programación.

Capítulo 3: Programación informática y Pensamiento Computacional en el sistema educativo

En la comunidad de Madrid, el Decreto 89/2014 del 24 de julio (Boletín Oficial de la Comunidad de Madrid, 2014) define la asignatura Tecnología y recursos digitales para la mejora del aprendizaje dentro de la oferta de libre configuración autonómica en la etapa de Educación Primaria. Por otro lado, el Decreto 48/2015 del 14 de mayo (Boletín Oficial de la Comunidad de Madrid, 2015) recoge la asignatura Tecnología, programación y robótica como obligatoria en 1º, 2º y 3º de E.S.O.

En la región de Murcia, el Decreto 220/2015 del 2 de septiembre (Boletín Oficial de la Región de Murcia, 2015) recoge la asignatura de Robótica dentro del ámbito de libre configuración autonómica para 2º de E.S.O.

En la comunidad foral de Navarra, el Decreto Foral 60/2014 del 16 de julio (Boletín Oficial de Navarra, 2014) integra contenidos de programación y Pensamiento Computacional en la asignatura de Matemáticas para los niveles de 4º y 5º de Educación Primaria.

En La Rioja, el Decreto 19/2015 (Boletín oficial de La Rioja, 2015) del 22 de junio incluye contenidos de programación, robótica y Pensamiento Computacional en las siguientes asignaturas:

- Tecnología (4º de E.S.O., asignatura troncal)

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

- Tecnología Industrial I y II (Bachillerato, materia específica)
- Tecnologías de la Información y la comunicación I y II (Bachillerato, materia específica)

Todos estos ejemplos indican que en torno al año 2015 existía una necesidad de incluir contenidos curriculares relacionados con el Pensamiento Computacional y la robótica en distintos niveles educativos. La Ley Educativa vigente en ese periodo de tiempo, la Ley Orgánica para la Mejora de la Calidad Educativa, no lo recogía de manera específica, por lo que cada comunidad autónoma lo hace de manera independiente. Además, en estas comunidades, dentro de un ámbito de educación no formal se contemplan numerosas iniciativas que promueven el desarrollo de habilidades de Pensamiento Computacional para los estudiantes de todos los niveles educativos con el fin de ampliar la formación y el desarrollo de competencias en este campo de manera transversal e integrada (MECD, 2018).

Posteriormente, encontramos una muestra del crecimiento continuado de la inclusión curricular del Pensamiento Computacional en el ámbito nacional en el proyecto llevado a cabo por el Ministerio de Educación y Formación Profesional denominado Escuela de Pensamiento Computacional (Ministerio de Educación y Formación Profesional, 2019). Esta iniciativa nace para dar a

nuestro alumnado la posibilidad de desarrollar el rol de creadores tecnológicos y alejarse de un perfil meramente de consumo, lo cual ayuda a desarrollar sus competencias en el uso de las herramientas tecnológicas de manera más eficaz y productiva para su aprendizaje y formación. Además, el desarrollo de este programa contempla el análisis empírico para conocer en qué medida este tipo de prácticas educativas tienen el efecto adecuado en nuestro alumnado.

Todo esto no son más que muestras de la necesidad de incluir de manera oficial en el currículo educativo el Pensamiento Computacional. Esto queda plasmado en los Reales Decretos publicados en 2022 que establecen la ordenación y enseñanzas mínimas, tanto en Educación Primaria como en Educación Secundaria (Ministerio de Educación y Formación Profesional, 2022a, 2022b). Esta ley educativa se caracteriza por una apuesta por la enseñanza competencial, dando a entender que se proviene de un currículo cargado de contenidos, en cierto modo prescindibles desde una práctica memorística, y que deben dar paso a lo que es realmente necesario para los estudiantes del Siglo XXI. Esto hace que el Pensamiento Computacional se presente como un elemento curricular importante e integrador para que se desarrollen destrezas clave en el futuro del alumnado, ya que se relaciona directamente con la resolución de problemas y con el planteamiento de procedimientos (Atmatzidou y Demetriadis, 2017)

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Por ejemplo, en Educación Primaria, el Pensamiento Computacional aparece en la descripción general de las competencias específicas como herramienta o recurso para resolver problemas, concretándose luego en las competencias de cada nivel para establecer los distintos criterios de evaluación y la organización de los saberes básicos de cada materia. En el Real Decreto de Educación Secundaria aparece este término de forma similar. Por ejemplo, en Matemáticas aparece relacionado directamente con la resolución de problemas en la descripción general de esta asignatura y posteriormente en la descripción de las competencias específicas y en los criterios de evaluación de los diferentes cursos. Para esta asignatura, los saberes básicos se organizan en varios sentidos, estando el sentido algebraico definido, entre otros elementos, con el Pensamiento Computacional. Además de aparecer de esta forma en la asignatura de Matemáticas, aparece en otras materias como Biología y Geología o Tecnología.

3.1 El marco de la alfabetización digital y la código-alfabetización

Dado que la inclusión curricular del Pensamiento Computacional en los distintos niveles educativos de manera transversal e integrada con el currículo de distintas áreas, asignaturas o ámbitos, es un objetivo relevante en el paradigma educativo actual, se deriva la importancia de un tipo de alfabetización asociada a este conjunto de competencias y saberes.

Concretamente, el desarrollo de estas habilidades y los enfoques metodológicos que pueden asociarse están ligados a la alfabetización digital. Una alfabetización que, como todas, busca paliar diferencias entre los distintos sectores de la sociedad; unos conocimientos básicos que ayudan a mitigar la brecha digital fomentando la capacidad de la ciudadanía para utilizar de manera crítica la información de la que disponemos con acceso libre en Internet (Gurstein, 2011).

La competencia digital está establecida desde hace años como un elemento prioritario en el aprendizaje permanente a lo largo de la vida, tanto a nivel nacional como europeo, así como para otros organismos como la UNESCO, y toma especial importancia para la sociedad nacida en este siglo. Una sociedad en la que emerge una nueva economía global donde la información es valor de cambio (Castells, 1998). En esta era, el modelo comunicativo ha cambiado gracias a Internet. Se favorece un modelo funcionalista de la comunicación que hace que esté jerarquizada y predeterminada por emisor y receptor, se transforma en un esquema abierto, interconectado, descentralizado, dinámico y flexible (Van Djick, 2016). Con todo esto, nace un nuevo tipo de ciudadanía que consume información multimodal e hipertextual, que procesa información muy deprisa, que está continuamente conectado (Reig y Vílchez, 2015) y que precisa de una adecuada alfabetización digital, entendida esta como el proceso de enseñanza-aprendizaje de la lectoescritura de textos digitales.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Dentro de este marco, para diferenciar los conocimientos básicos y habilidades necesarias para desenvolverse en la red de Internet también se da el concepto de alfabetización web. Este hace referencia al conjunto de habilidades y competencias que permiten a un individuo leer, escribir e interactuar en la web. La Red ofrece una ingente cantidad de información y para poder desarrollarse en este sentido, previamente es necesaria una adecuada alfabetización informativa. Esto permitirá saber acceder a ella, interpretarla, analizarla, gestionarla, almacenarla o compartirla (Kiliç-Çakmak, 2010). Este tipo de alfabetización siempre ha estado considerada como un elemento fundamental en la formación continua de una persona. En un sentido amplio, esta alfabetización informativa engloba otros conjuntos de destrezas y habilidades como la alfabetización mediática, la propia de las redes, la alfabetización informática y la alfabetización web, ya que gran parte de la información la obtenemos a través de ordenadores e Internet (Savolainen, 2002). Este concepto derivado directamente de la alfabetización informativa, la alfabetización web, toma especial relevancia para la formación, aprendizaje y desarrollo social de cada persona (Keshavarz, 2020).

En todo este contexto se entiende la necesidad de una adecuada alfabetización digital en todos sus aspectos como la clave para un cambio en el ecosistema de la educación, cuya eficacia dependerá del desarrollo efectivo de esta alfabetización (Pérez-Escoda et al., 2019). Como una respuesta a este nuevo

punto en el paradigma educativo, surge el concepto del modelo holístico de la código-alfabetización. Un modelo que intenta abarcar distintos ámbitos para este tipo de conocimientos básicos al situarse como parte de todo lo que se puede considerar como alfabetización web, que al mismo tiempo la consideramos como parte de lo que hemos empezado analizando como alfabetización digital.

Según Román-González (2016), la código-alfabetización es el proceso de enseñanza-aprendizaje de lectoescritura con los lenguajes informáticos de programación. Para el autor, este concepto se sitúa en un nivel superior sobre los otros dos, ya que permite trabajar con elementos comunes de los otros tipos de alfabetizaciones: el código informático, el lenguaje con el que uno puede comunicarse con un ordenador y, por tanto, un metalenguaje. De esta relación, se obtiene la consideración de la código-alfabetización como un meta-alfabetismo de orden superior. Pero es obvio ver que para que todo el proceso de aprender un lenguaje de programación sea un proceso de aprendizaje productivo, el sistema lingüístico de una persona tiene que ser el adecuado (Prat et al., 2020). En los últimos años, muchos estudios se han centrado en abordar el proceso de aprendizaje de un lenguaje de programación como una actividad de creación con significado propio en lugar de hacerlo como un proceso de resolución de problemas (Bers, 2018, 2019; Bers y Sullivan, 2019). Estos estudios han arrojado resultados positivos, tanto en el

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

proceso de aprendizaje como en el propio proceso de enseñanza de estas destrezas en el aula (Hassenfeld y Bers, 2020). Esto refleja que el sistema lingüístico y el semántico juegan un papel importante en el proceso de alfabetización digital en general y en el proceso de la código-alfabetización en particular, ya que desempeñan un papel muy importante en el aprendizaje del procesamiento del código informático, especialmente en los niños, cuando el sistema lingüístico aún está en desarrollo (Ivanova et al., 2020).

Todos estos tipos de alfabetización pueden quedar recogidos en distintos modelos más o menos específicos, siendo uno de los más influyentes el DigCompEdu de 2017, mencionado en el apartado 2.5 al abordar la relación curricular del Pensamiento Computacional, establecido en el marco de acción de la Unión Europea. Este tipo de alfabetizaciones son fundamentales en el proceso de aprendizaje continuo de una persona, pero esto cobra especial interés al referirnos a un sector de la sociedad al que se le asoció una etiqueta de “nativo digital” con la publicación de un artículo de Marc Prensky (2001) en la revista “On the Horizon”. En este trabajo plantea las características de una generación que ha nacido rodeada de dispositivos electrónicos y la forma en la que procesan la información para aprender, tanto en el ámbito académico como en el personal. Pero con el paso del tiempo, desde que se acuñó este término se ha puesto de manifiesto la necesidad de una adecuada alfabetización digital en todas sus dimensiones. La mayoría de

estas personas presenta cierta apariencia de confort con el uso cotidiano de las tecnologías. Se ven cómodas mandando correos electrónicos, jugando a videojuegos online, teniendo que relacionándose en otro idioma o navegando de manera básica por Internet. Pero como hemos visto antes, esto se puede asociar a una pequeña parte de una alfabetización digital y de la alfabetización web en la que se pretende saber usar los recursos digitales. Muchos de estas personas no son capaces de hacer mucho más, no son capaces de analizar contenidos a los que tienen acceso de manera crítica, resolver problemas o incluso entender cómo funcionan. A pesar de interactuar con los medios tecnológicos continuamente, pocos son capaces de crear un sencillo juego, una animación o una simulación. En cierto sentido, esto es como si alguien supiera leer pero no escribir (Resnick et al., 2009). Para desarrollar destrezas digitales dentro de este campo de alfabetización tan necesario hoy en día, no basta con saber chatear, buscar o interactuar. Es necesario saber diseñar, crear e inventar nuevos contenidos multimedia (Resnick, 2007). En este sentido, Prensky considera la código-alfabetización como una habilidad única que marcará un distintivo claro entre las personas por tener las destrezas necesarias para controlar, en cierto modo, la tecnología digital y poder adaptarla a determinadas necesidades y propósitos (Prensky, 2008). Otros autores como Rushkoff (2012) ven este tipo de alfabetización como algo indispensable, como un requisito para participar en el mundo digital, ya que si no se dispone

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

de una alfabetización mínima de código, aceptaremos los dispositivos tecnológicos y el software que usamos día a día con ciertas limitaciones y sujetos a las condiciones establecidas por sus creadores. Si una persona abre su campo de acción en el uso y capacidad de creación con recursos tecnológicos, también ampliará el horizonte de todo aquello nuevo que puede aprender. En particular, y como se ha hecho referencia anteriormente, aprender a programar desarrolla destrezas de Pensamiento Computacional asociadas a los procesos de resolución de problemas y el diseño de estrategias de razonamiento en otros campos que no necesariamente requieren el uso de ordenadores (Wing, 2006).

Además, no podemos obviar el hecho de que nos enfrentamos a una cuarta revolución industrial, la cual se caracteriza por un fuerte impacto de las tecnologías que fusionan el mundo físico, digital y biológico que influye en todas las disciplinas, tecnologías e industrias (Schwab, 2017). Según Balanskat y Engelhardt (2015) los estudiantes de hoy tendrán que trabajar en empleos relacionados con el desarrollo tecnológico ya que será importante y necesario para la sociedad. Por lo tanto, habilidades relacionadas con el Pensamiento Computacional y la programación serán competencias clave para el Siglo XXI y un factor muy importante en la política educativa para que el sector educativo se adapte a las necesidades y demandas de la sociedad.

3.2 Un movimiento social para desarrollar el Pensamiento Computacional

El desarrollo del Pensamiento Computacional y su inclusión en el currículo educativo, no solo ha tenido en la última década un fuerte impulso y apoyo por parte de administraciones públicas y otros organismos oficiales del ámbito educativo sino que también ha sido muy importante la labor y el papel desempeñado por distintas asociaciones a través de la celebración de eventos de carácter internacional que promueven el desarrollo de este tipo de prácticas educativas en todos los niveles, dentro de cualquier asignatura y que lo hacen accesible para cualquier docente. Entre todos los eventos de este tipo que nos podemos encontrar hoy en día, podemos comenzar destacando el que quizá haya tenido más repercusión desde su inicio, “*The hour of code*” (*La hora del código*). Es un movimiento que busca introducir la programación informática en todos los niveles educativos para eliminar esa sensación de misterio o dificultad que puede presentar este campo para muchos docentes no especializados (Wilson, 2015). El término “hora del código” empezó siendo una referencia a una hora de clase para aprender programación, consistió en la creación de un tutorial de programación de una hora para mostrar a los estudiantes que programar puede ser una actividad divertida y creativa (Du y Wimmer, 2019). Desde su edición de 2013, se celebra anualmente de forma global coincidiendo con la *Computer Science Education Week (CSEW)* de Estados Unidos. Este movimiento está promovido por code.org, una

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

organización sin ánimo de lucro cuyo objetivo es ofrecer acceso a las ciencias de la computación para todos los estudiantes, aumentando la participación de mujeres en este campo. Destaca el hecho de que para la mayoría de los estudiantes, realizar actividades de introducción guiadas de menos de una hora tiene un impacto positivo para interesarse por esta ciencia (Phillips y Brooks, 2014)

Otro ejemplo destacado de evento internacional con tal propósito es *Europe Code Week (semana europea de la programación)*. Del mismo modo que la *hora del código*, la importancia de este evento radica en la necesidad que habrá en muchos trabajos de un futuro próximo de tener conocimientos básicos de programación. De este hecho nace el objetivo de acercar los conceptos de programación de manera sencilla y divertida a estudiantes de todos los niveles organizando en toda Europa eventos de todo tipo como talleres, concursos, charlas, etc. La *Semana Europea de la Programación (EU Code Week)* nace en 2013 como una iniciativa del *Young Advisor Group* perteneciente a la Comisión Europea y que se desarrolla con una red de docentes y profesionales del ámbito educativo que participan de manera voluntaria como embajadores o profesores de referencia (*leading teacher*), con el objetivo de dar difusión a las actividades que se realizan en este marco y hacer que el impacto de esta iniciativa llegue al mayor número posible de estudiantes (Moreno-León y Robles, 2015).

Capítulo 4: Lenguajes de programación visual por bloques. Scratch

En un capítulo 2.1 de este trabajo, haciendo referencia a la evolución de la idea de Pensamiento Computacional y sus inicios, se habló de Seymour Papert y del lenguaje de programación LOGO como el punto de partida en la historia para la inclusión y desarrollo de este tipo de habilidades en el ámbito escolar. Como sabemos, desde entonces han sido muchos los cambios tecnológicos y sociales que han favorecido el desarrollo de este campo. Una de las cuestiones más influyentes de Papert en esta evolución fue su visión de la relación que cabe esperar entre los estudiantes y los ordenadores. En su libro *“Mindstorms: children computers and powerful ideas”*, hace más de 40 años, destacaba que en la mayoría de las situaciones educativas en las que un niño entra en contacto con un ordenador es para poner a prueba al estudiante, para proponer un ejercicio de un nivel de dificultad adecuado, para proporcionarles y dispensarles información. En este sentido, parece que es el ordenador el que programa al niño. Con LOGO se pretendía buscar la relación inversa: el niño, incluso en edades tempranas, debe tener el control y ser él el que programe al ordenador (Papert, 1980).

Desde entonces, uno de los aspectos más importante y que ha facilitado la inclusión de este tipo de destrezas y acercado la programación a todos los niveles educativos, incluso en edades tempranas, ha sido la aparición de los

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

lenguajes visuales de programación y, en concreto, los lenguajes de programación con bloques. Este tipo de entornos tienen como objetivo principal eliminar los errores de sintaxis y los correspondientes a la formación de estructuras lógicas, facilitando así la construcción de algoritmos. En lugar de tener que escribir el código, el usuario utiliza distintos bloques, cada uno de los cuales se corresponde con un elemento de programación: una estructura de control, una variable, una función, un condicional, etc. Estos elementos pueden combinarse entre sí simplemente con arrastrarlos por la pantalla para encajar con otros bloques de manera muy intuitiva atendiendo a la lógica que permita formar el algoritmo de un programa informático (Ouahbi, Kaddari, Darhmaoui, Elachqar, y Lahmine, 2015).

Los bloques de programación suelen tener una determinada forma y color, como en un rompecabezas. Además, esta forma suele limitar el uso que puede tener una pieza a la hora de construir un algoritmo, lo cual simplifica mucho la construcción de programas con una sintaxis correcta (Utreras y Pontelli, 2020). Los lenguajes de programación visual son muy eficaces en el ámbito educativo, existiendo una gran variedad de ellos, incluso siendo específicos para determinados productos o recursos. El principal objetivo de estos entornos de trabajo es ayudar a los estudiantes en los cursos de iniciación a la programación y al desarrollo del Pensamiento Computacional trabajando la programación orientada a objetos (Cooper, Dann y Pausch, 2003).

De manera general, entre las ventajas del uso de un lenguaje de programación visual podemos destacar las siguientes (Curran, 2017):

- Los estudiantes no necesitan aprender sintaxis y no se generan errores derivados de escribir código.
- Los estudiantes pueden ver qué bloques (instrucciones) están disponibles.
- Estos elementos a menudo esconden procesos u operaciones complejas en un solo bloque.

De entre todos los entornos de programación de este tipo que existen hoy en día, el que se va a utilizar en las distintas experiencias empíricas de este trabajo de investigación es Scratch: un lenguaje de programación visual gratuito desarrollado para acercar la programación a los estudiantes en edades tempranas. Sin duda Scratch es uno de los programas de este tipo más utilizados para introducir el Pensamiento Computacional con estudiantes de cualquier nivel (Evangelopoulou y Xinogalos, 2018). Este entorno de programación visual por bloques permite crear de forma rápida, fácil y divertida una gran variedad de proyectos como historias interactivas, tarjetas de felicitación, juegos, recursos musicales, animaciones artísticas, tutoriales, proyectos musicales, etc (Moreno-León et al., 2017; Papadakis et al., 2014;

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Tsur, 2017). El origen de Scratch, su influencia principal y referente para su desarrollo, es el conjunto de ideas, la filosofía de enseñanza de LOGO y la teoría del construccionismo, considerando este concepto como una estrategia educativa inspiradora para el desarrollo de un lenguaje de programación que permita al estudiante establecer nuevas relaciones de conocimiento e igualmente realizar nuevas presentaciones de su aprendizaje (Kafai y Resnick, 1996). Este proyecto, desarrollado por el *Lifelong Kindergarten* del Instituto Tecnológico de Massachusetts, nace en 2003 con un diseño original motivado por las necesidades e intereses analizados en jóvenes de entre 8 y 16 años de edad en centros informáticos de actividades extraescolares (Resnick et al., 2003). Un aspecto revolucionario fue que incluyó la capacidad de poder programar en las actividades que resultaban atractivas e interesantes a los jóvenes, permitiendo aprender a través de la exploración y el intercambio entre compañeros, estando menos centrada en instrucciones directas como ocurría con otros lenguajes de programación. Con Scratch es frecuente desarrollar actividades que capten fácilmente la atención de los estudiantes, intentando activar distintos aspectos y que atraigan al alumnado. Igualmente, este tipo de actividades suelen diseñarse de manera que exista conexión con otras áreas como ciencias, literatura o arte (Curzon et al., 2009).

En un principio, Scratch se utilizó en entornos de aprendizaje informales o no reglados como centros sociales, clubes de programación de actividades

extraescolares o bibliotecas, pero su inclusión en el ámbito escolar es algo que con el tiempo ha ido creciendo de manera muy positiva (Maloney et al., 2010). Hay que tener en cuenta que uno de los objetivos principales de Scratch es introducir la programación a quienes no tienen experiencia previa, lo cual impulsó muchos aspectos relacionados con el diseño del entorno de trabajo. Por lo tanto, atendiendo a las necesidades e inquietudes del público objetivo, Resnick et al. (2003) destacan una serie de aspectos relacionados en su diseño inicial tales como:

- Los jóvenes ven la herramienta/actividades como algo divertido, que coincide con sus intereses y pasiones.
- Los jóvenes ven el valor y el potencial de la herramienta de inmediato.
- Los jóvenes pueden crear un primer proyecto con la herramienta de forma rápida y sencilla.
- Los jóvenes pueden crear "productos" que pueden mostrar a los demás (orgullo de autoría).
- La herramienta admite una amplia gama de actividades diferentes.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

- La herramienta y las actividades son atractivas para jóvenes de diferentes orígenes y culturas.
- Los jóvenes pueden aprender las características de la herramienta de forma gradual y progresiva.
- Los jóvenes pueden seguir utilizando la herramienta de forma cada vez más compleja a lo largo del tiempo.

Con estas consideraciones y criterios, se desarrolló Scratch intentando evitar problemas y dificultades que puedan obstaculizar el introducir a los más jóvenes en el mundo de la programación. Así, los autores diseñaron este lenguaje de programación con la idea de hacer un proceso en el que se realiza una construcción por bloques similar a la que se hace de manera manipulativa con piezas de Lego (Olabe et al., 2011) y que permitiese comenzar programando proyectos multimedia, y no solo empezar a programar manipulando números o gráficos sencillos. Se buscó una perfecta integración con el mundo físico y que se pudieran programar todo tipo de sensores o dispositivos adaptables a entornos escolares, que se pudiese trabajar en muchos idiomas y que además tuviese un importante factor o característica para compartir. De esto último, cabe destacar la importancia de la comunidad

de usuarios de Scratch y su portal web como unos factores que hacen que destaque sobre otros recursos educativos de este tipo.

Desde su lanzamiento en 2007, el sitio web de Scratch se ha convertido en una comunidad online apasionante en la que los usuarios pueden compartir, discutir y reinventar proyectos de otras personas, con el debido reconocimiento (Resnick et al., 2009). Este lugar tiene como objetivo proporcionar un espacio para que los diseñadores de Scratch de todos los niveles estén conectados entre sí compartiendo su trabajo y apoyando el aprendizaje de otros. Cada usuario tiene una página en la que puede almacenar sus proyectos y otros elementos de participación en línea como seguidores o estudios de proyectos. Como en otras redes de usuarios, pueden dejar comentarios sobre el trabajo de otro usuario o expresar su aprecio con el icono de *Love it*. Además de este tipo de interacciones, un usuario puede descargar un proyecto para analizar el código y todos los objetos incluidos (Brennan, 2012).

4.1 Programación con Scratch en contextos educativos

Como sabemos, el desarrollo y aplicación del campo de la informática se ha expandido rápidamente en la sociedad en los últimos 20 años. Si observamos los efectos en el ámbito educativo, esto ha propiciado y mejorado la forma en la que se ha podido trabajar la programación informática con estudiantes de

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

distintos niveles. Esto se ha valido de la necesidad e interés existente por desarrollar el Pensamiento Computacional, entendiendo este como un concepto muy general que incluye una base necesaria para entender el mundo tecnológico y desarrollar habilidades de resolución de problemas en distintas disciplinas (Tedre y Denning, 2016). En la época de los 80 y los 90, cuando se hacía referencia al uso de la informática, generalmente el trabajo se centraba en aprender a utilizar herramientas de ofimática. Cuando se abordaban conceptos de programación, se hacía en contextos propios de la iniciación de los estudiantes en este ámbito de manera formal. Por entonces, cuando se hacía referencia a la alfabetización en materia de programación se denominaba como una *habilidad moderna de supervivencia* (Sheil, 1980). Si bien es cierto que se podían encontrar otras denominaciones como alfabetización computacional (DiSessa, 2000), alfabetización en razonamiento algorítmico (Culbertson, 1986) o “la segunda alfabetización” (Ershov, 1981).

Como se ha descrito en secciones anteriores, a lo largo de las últimas décadas el punto más destacado y que más ha favorecido al desarrollo de la informática en el ámbito educativo, así como la forma en la que se concibe y de desarrolla hoy el Pensamiento Computacional, es el conjunto de ideas y planteamientos dados por Seymour Papert en su libro *Mindstorms*. Este libro supone un aporte a la historia de esta ciencia muy conocido dentro y fuera de este ámbito por introducir el construccionismo y su visión del aprendizaje a

través del descubrimiento, centrado en el estudiante y desarrollando proyectos utilizando recursos tecnológicos. Pero hasta que estas ideas se han podido desarrollar de manera efectiva y generalizada, la informática era en cierto modo una asignatura un poco más compleja, destinada solo a unos pocos y que se justificaba fácilmente por el conocimiento de herramientas como el procesador de texto o la hoja de cálculo. Pero gracias al desarrollo tecnológico, tanto de software como de hardware, hoy disponemos de herramientas que facilitan el uso de la programación y el desarrollo del Pensamiento Computacional para cualquier estudiante. Un ejemplo es el desarrollo de los lenguajes de programación visuales como Scratch. Es con esta tecnología con la que se han podido trabajar de manera fácil y generalizada las ideas que Papert recogía en su libro *Mindstorms*, dando la posibilidad de desarrollar habilidades cognitivas de los estudiantes a través de la programación y favoreciendo el desarrollo de destrezas asociadas a la resolución de problemas (Molina et al., 2020). Gracias al uso de Scratch como lenguaje de programación, así como con otras herramientas similares, se ha dado en nuestras escuelas el paso de “aprender a programar” a “programar para aprender” (Resnick, 2013). Esta idea facilita el desarrollo de prácticas educativas que fomentan un trabajo interdisciplinar con el que se desarrollan habilidades desde distintos enfoques, con diferentes perspectivas y con distintos campos de aplicación.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Por ejemplo, cuando un grupo de estudiantes está trabajando en el desarrollo de proyectos con Scratch, está desarrollando habilidades relacionadas con el uso de la información y la comunicación; los estudiantes tienen que aprender a seleccionar, a crear y a manejar distintos elementos multimedia (textos, imágenes, sonidos, vídeos) para integrarlos en sus propias creaciones de manera original y personalizada (Hassan et al., 2018). Otra habilidad que desarrollan los estudiantes a la hora de trabajar con Scratch es la asociada al razonamiento y resolución de problemas. El hecho de tener que controlar distintos tiempos de ejecución y la interacción entre diferentes objetos, hace que tengan que trabajar a través de la descomposición en problemas más sencillos, que tengan que estar comprobando continuamente los pasos que van realizando y analizando el feedback recibido de cada acción para buscar continuas mejoras o ampliaciones de un programa (Mladenović et al., 2017). Esto supone un proceso de resolución de problemas continuo aplicado a un proceso de diseño y producción creativa, algo muy importante en el mundo en que vivimos y es cual está en continuo cambio.

Por otro lado está la capacidad de trabajo colaborativo o el hecho de trabajar junto a otras personas. Trabajar con objetos visuales y poder programar usando bloques hace que la colaboración entre personas sea más fácil y permite que los estudiantes puedan trabajar conjuntamente en proyectos intercambiando ideas, experiencias o código. Además, como una de las

virtudes de Scratch y su comunidad educativa es el intercambio de ideas y proyectos, esto facilita que los estudiantes puedan inspirarse en los proyectos de otros usuarios ofreciendo comentarios a modo de feedback (Dhariwal, 2018).

Los docentes de cualquier disciplina siempre intentan desarrollar estas y otras habilidades cuando realizan distintos proyectos educativos con su alumnado. Scratch ofrece un escenario de trabajo que facilita, como hemos visto, la adquisición de una serie de destrezas y habilidades que pueden aplicarse en distintos ámbitos y que se caracterizan por su transversalidad y potencial práctico. Aunque particularmente Scratch es un entorno idóneo para el desarrollo de modelos en el ámbito científico, su diseño por bloques y la facilidad de programar los distintos objetos que conforman un proyecto hace que su campo de aplicación sea tan amplio, lo cual ayuda enormemente a captar el interés de nuestro alumnado.

Cuando hablamos del interés de nuestros estudiantes en el ámbito educativo, estamos haciendo referencia a ese gusto y capacidad de participación voluntaria que muestran cuando les atrae una actividad, algo que está estrechamente relacionado con la motivación: cuando una persona hace algo que le interesa de manera voluntaria decimos que está muy motivada con esa tarea (Deci, 1992). El interés siempre está relacionado con temas, tareas o

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

actividades específicas de manera tanto afectiva como cognitiva. Por lo que el placer, el bienestar y la felicidad son aspectos emocionales típicos que nuestros estudiantes muestran cuando realizan actividades que realizan con interés y motivación (Krapp, 2002). Cuando un docente consigue despertar interés en su alumnado con el fin de motivar y propiciar un trabajo comprometido en el aula, consigue que se desarrollen unas experiencias afectivas inmediatas que propician su relación y conexión con su entorno, lo cual es especialmente importante para captar su atención (Mitchell, 1992). Para Mithcell (1992), este interés puede mantenerse si se involucra al estudiante en una práctica que pueda percibir como significativa. Esto es algo que el profesorado de cualquier disciplina busca para el desarrollo de su trabajo diario. Scratch facilita en cierto modo el desencadenamiento de este interés y motivación por una práctica. No solo por trabajar con un recurso tecnológico, también por desarrollar una práctica que está relacionada con aspectos de su vida diaria e intereses de una gran mayoría de jóvenes.

Este tipo de estrategias ayudan a despertar la motivación del alumnado ofreciendo alternativas y dinamismo a las sesiones de aula, permitiendo descubrir y realizar de manera más personal todo el proceso de aprendizaje (Lerís y Sein-Echaluce, 2011). Todos estos aspectos, junto al actual acceso libre al hardware y al 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).

Si bien es cierto que no hay que pensar que es una herramienta infalible con la que todos los estudiantes muestran un interés altamente significativo. Pero es cierto que en muchas ocasiones resulta conveniente, en disciplinas que suelen ser menos atractivas para una mayoría del alumnado, emplear recursos de este tipo que ayuden a despertar este atractivo y motivación para intentar trabajar y desarrollar habilidades y destrezas asociadas con esta materia de manera más interesante (Acuña-Medina et al., 2018).

Es común que en muchas ocasiones se utilicen metodologías que aportan resultados positivos al proceso de aprendizaje del alumnado como el aprendizaje basado en proyectos, el aprendizaje basado en problemas, el aprendizaje basado en la indagación o cualquier otra forma de metodología activa (Johnson et al., 2014). Todas estas prácticas educativas permiten que los estudiantes conozcan cómo funciona el mundo y les aporta las herramientas necesarias para afrontar y resolver problemas complejos. Si estas estrategias van ligadas al desarrollo del Pensamiento Computacional, nuestro alumnado

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

podrá tener no solo un adecuado nivel de competencias para desenvolverse en nuestro día a día, también será capaz de aplicar todas estas destrezas adquiridas de manera más completa, integrada y transversal, lo cual es algo fundamental en nuestra sociedad. Además, el desarrollo de los aspectos relacionados con las destrezas de Pensamiento Computacional favorece el trabajo en habilidades de orden superior de los estudiantes a la vez que se desarrollan competencias en resolución de problemas y razonamiento lógico (Fessakis et al., 2013; Kafai y Burke, 2014).

El desarrollo de actividades de Pensamiento Computacional en el ámbito educativo utilizando herramientas como Scratch nos ayuda a desarrollar ese tipo de competencias, habilidades o destrezas asociadas al uso de la tecnología y trabajar las distintas alfabetizaciones asociadas como la alfabetización web o la código alfabetización que se trataron anteriormente, haciendo que nuestros estudiantes pasen a ser creadores y no solo usen la tecnología como meros consumidores.

Esto se consigue con un éxito relativamente rápido gracias al diseño de su entorno de trabajo y a la simplicidad del lenguaje de programación, haciendo que un estudiante sin experiencia previa pueda crear su primer programa o animación en unos 30 minutos. Si bien mantiene la posibilidad de profundizar en función de las necesidades o inquietudes de cada usuario permitiendo que

se pueda usar durante mucho tiempo antes de pasar a otro lenguaje de programación basado en texto (Maloney et al., 2010). Esto hace que los estudiantes pasen más tiempo utilizando Scratch por su propia iniciativa o interés que otro recurso educativo que puedan tener a su disposición (Weintrop et al., 2015).

La importancia de incluir recursos de programación visual como Scratch para trabajar, entre otros aspectos, conceptos de lógica, de Matemáticas, para desarrollar habilidades de creación multimedia o para incluir contenidos artísticos en prácticas STEM favorece el desarrollo de una metodología activa como el aprendizaje basado en proyectos a la vez que aumenta la motivación y el compromiso del alumnado a participar en este tipo de experiencias de aprendizaje (Sáez-López, Román-González y Vázquez-Cano, 2016).

4.2 Programación con Scratch para aprender Matemáticas

La sociedad vive inmersa en la era de la información y la comunicación, y esto hace que esté sujeta a los cambios y al ritmo marcado por este periodo, el cual marcha en una única dirección caracterizada por una evolución tecnológica (Pozo, 2000). Como no puede ser de otra manera, la educación no es ajena a esta realidad y está sujeta a las necesidades que dan respuesta a las nuevas demandas (Téliz, 2015). En ese sentido el uso de recursos tecnológicos para el aprendizaje de las Matemáticas facilitan una inclusión más efectiva y

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

dinámica, facilitando la gestión de la información, el aprendizaje colaborativo, el trabajo en equipo o la argumentación de razonamientos (Podestá, 2011).

En particular, el uso de la programación dentro del ámbito de la enseñanza de las Matemáticas no es algo nuevo, como se ha podido entender en capítulos anteriores en los que se han tratado trabajos iniciales de Seymour Papert (1980), la teoría del construccionismo y el desarrollo de LOGO para hacer que una tortuga se moviese por la pantalla del ordenador, en muchos casos, dibujando figuras geométricas con la intención de ofrecer alternativas para el aprendizaje de las Matemáticas despertando el interés y la motivación del alumnado. Está claro que este tipo de entornos de trabajo pueden ser un potencial recurso para crear un nuevo entorno de aprendizaje de las Matemáticas. Con LOGO en particular, se analizó si existe una relación positiva entre el desarrollo cognitivo y la resolución de problemas con estudiantes de Matemáticas a través de resultados de decenas de artículos de análisis tanto cualitativo como cuantitativo (Yelland, 1995). Pero las conclusiones, en algunos casos, podían resultar contradictorias ya que aunque en muchas ocasiones se mostraba que el uso de LOGO tenía un impacto positivo en el aprendizaje de las Matemáticas, en otras ocasiones no se apreciaba una diferencia en la capacidad de resolución de problemas en estudiantes que habían completado una serie de proyectos o tareas. Sin embargo, independientemente de haber podido obtener resultados en algún

punto contradictorio con un análisis tan variado y completo de trabajos de investigación, recursos como este ofrecían una alternativa útil y práctica en el proceso de aprendizaje de las Matemáticas.

El desarrollo e inclusión de este tipo de recursos educativos y lenguajes de programación ha permitido incorporar el potencial que tiene expresar ideas Matemáticas a través de la creación de programas informáticos en edades tempranas, sin necesidad de aprender complicadas sintaxis más indicadas para cursos superiores.

En la literatura científica correspondiente a este tema se pueden encontrar trabajos importantes con resultados significativos con el uso de diferentes entornos de programación como Scratch o recursos relacionados con la robótica educativa, los cuales también suelen trabajarse a través de un entorno de programación visual por bloques. En relación con esto último, Benitti (2012) observó que en la mayoría de los casos, los beneficios de estas prácticas educativas recaían mayormente sobre conceptos del ámbito STEM en general. Si nos centramos en el ámbito de las Matemáticas, se encuentran numerosos trabajos centrados en distintos temas como Geometría y medidas angulares, fracciones, proporciones o en resolución de problemas. Por ejemplo, Lindh y Holgersson (2007) obtienen resultados positivos a la hora de desarrollar procesos de resolución de problemas lógicos y matemáticos

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

utilizando kit de robótica educativa. Igualmente, Nugent et al. (2009) evidencian un impacto positivo en jóvenes entre 12 y 14 años de edad en el aprendizaje de las Matemáticas y conceptos geoespaciales utilizando kits de robótica y tecnología de geolocalización para desarrollar actividades de programación.

Igualmente, se encuentran resultados positivos en trabajos como el de Song (2017) a la hora de incluir prácticas con Scratch en Educación Secundaria para trabajar aspectos relacionados, por ejemplo, con Trigonometría a través de una metodología basada en el juego. Calder (2010) también evidencia cómo trabajar conceptos e ideas Matemáticas de Geometría y medida a través de proyectos con Scratch. Dentro del mismo bloque de Matemáticas, la Geometría, el trabajo de Iskrenovic-Momcilovic (2020) analiza la influencia que tiene a la hora de aprender conceptos básicos de Geometría la inclusión de Scratch frente a un trabajo tradicional de estos contenidos matemáticos, haciendo que estos sean más interesantes y atractivos para el alumnado.

Chiang (2018), muestra, por su parte, muestra resultados positivos en el proceso de resolución de ecuaciones para estudiantes de Educación Secundaria al trabajar este tema a través de un experiencia basada en la construcción de un juego con Scratch. Taylor et al. (2010) muestran como el uso de Scratch puede fomentar un trabajo cooperativo en el proceso de

resolución de problemas estableciendo objetivos y comprobando ideas trabajadas en el proceso de aprendizaje de las Matemáticas. Lie et al. (2017) también analizan y muestran la relación que se establece al trabajar conceptos de programación con Scratch y el razonamiento matemático y el proceso de resolución de problemas; Vidal-Silva et al. (2020) analiza la forma en la que el uso de Scratch en estudiantes de Educación Primaria permite desarrollar el razonamiento algorítmico y las competencias lógico-Matemáticas de manera efectiva.

En el presente proceso de investigación, muchas de las experiencias desarrolladas se fundamentan en la propuesta de Sneider et al. (2014) para una intersección de habilidades asociadas tanto al razonamiento matemático como al Pensamiento Computacional que pueden resultar de gran importancia para favorecer el desarrollo de ambas destrezas. En su estudio proponen que el razonamiento matemático se produce cuando el estudiante tiene que resolver una nueva situación haciendo uso de recursos matemáticos. Del mismo modo, señalan que el Pensamiento Computacional se da como superposición de las muchas formas en las que una computadora puede ayudar a visualizar una situación problemática y resolverla. Es fácil entender que son cosas diferentes, pero igualmente es sencillo ver que ambas propuestas favorecen el desarrollo de habilidades como la modelización, resolución de problemas o el análisis y la interpretación de datos.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

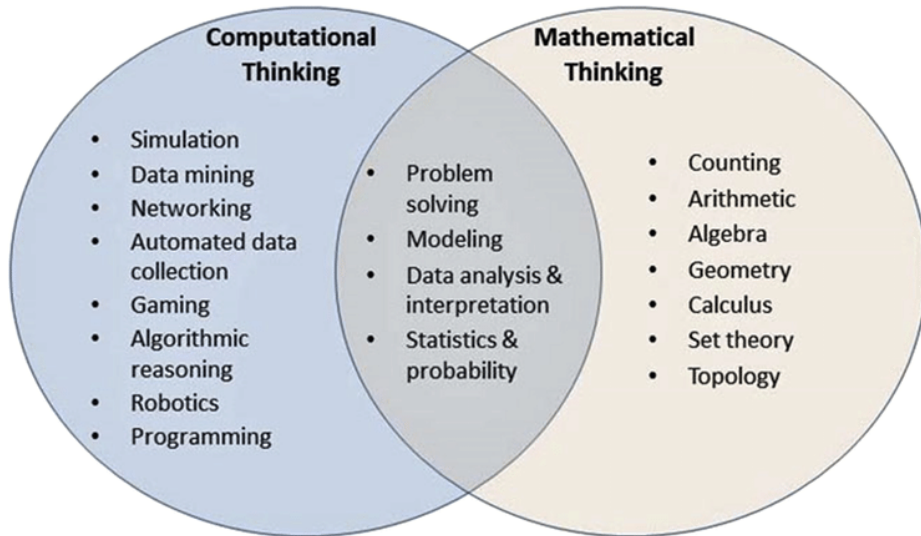


Figura 2: Diagrama de Venn del pensamiento matemático y computacional; Fuente (Sneide et al., 2014)

III. Parte empírica

Capítulo 5: La resolución de problemas basada en el método de Polya usando el Pensamiento Computacional y Scratch con estudiantes de Educación Secundaria

Capítulo 6: Pensamiento Computacional con Scratch. Recurso para trabajar Geometría en 5º de Educación Primaria

Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Capítulo 5: La resolución de problemas basada en el método de Polya usando el Pensamiento Computacional y Scratch con estudiantes de Educación Secundaria¹

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.

1 Molina Ayuso, Álvaro, Adamuz Povedano, N., & Bracho López, R. (2020). La resolución de problemas basada en el método de Polya usando el pensamiento computacional y Scratch con estudiantes de Educación Secundaria. *Aula Abierta*, 49(1), 83–90. <https://doi.org/10.17811/rifie.49.1.2020.83-90>

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Palabras clave: Lenguajes de programación, informática educativa, resolución de problemas, Matemáticas en Educación Secundaria, Pensamiento Computacional, tecnología educativa, movimiento maker.

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. In order to know and internalize learning about this problem solving heuristic, the student have 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 video game 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 in order to carry out good resolution of a problematic situation faced by a student using a problem solving technique.

Capítulo 5: La resolución de problemas basada en el método de Polya usando el Pensamiento Computacional y Scratch con estudiantes de Educación Secundaria

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

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

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

(1980), se mostraba un punto de vista utópico de cómo la tecnología podría ayudar en el proceso de aprendizaje, idea que se 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 (National Research Council, 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).

Capítulo 5: La resolución de problemas basada en el método de Polya usando el Pensamiento Computacional y Scratch con estudiantes de Educación Secundaria

Una definición ampliamente aceptada, es la establecida por Jannet Wing (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 para 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

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

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, E., y Magro, 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 Beccerra, 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 en línea 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

Capítulo 5: La resolución de problemas basada en el método de Polya usando el Pensamiento Computacional y Scratch con estudiantes de Educación Secundaria

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 et al., 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 (Lerís y Sein-Echaluce, 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 reinventado 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 a 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 currículo 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).

La resolución del 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

Capítulo 5: La resolución de problemas basada en el método de Polya usando el Pensamiento Computacional y Scratch con estudiantes de Educación Secundaria

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 et al., 2015). Especialmente, cabe destacar la importancia de desarrollar en nuestro alumnado un pensamiento crítico que le 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:

1. 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).
2. 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.
3. 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?
4. 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 denominaciones 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.

Material y métodos

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. 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.

Metodología

El proceso empírico de la investigación se ha llevado a cabo con un total de 18 estudiantes de primer curso 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 socio-económico 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 con frecuencia conveniente emplear una metodología heterogénea para poder dar mejor significado a los datos obtenidos; la naturaleza de los problemas que se abordan requieren 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 et al., 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.

Para realizar el presente trabajo se ha diseñado un proceso cuasi-experimental, 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 (R. 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.

Desarrollo de la experiencia empírica

En primer lugar, los estudiantes realizaron un pretest consistente en seis problemas aritméticos acordes a su nivel académico (Anexos 1 y 2). Este tipo de problemas ocupan la mayor parte del currículo tanto en Educación Primaria como en Educación Secundaria Obligatoria. Los problemas aritméticos suelen ser 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, 2015). Una vez que los alumnos y alumnas realizaron la primera prueba de problemas, completaron un cuestionario de reflexión (Anexo 3) del que se pudo obtener información acerca de cómo consideran 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

Capítulo 5: La resolución de problemas basada en el método de Polya usando el Pensamiento Computacional y Scratch con estudiantes de Educación Secundaria

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.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Tabla 1: 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.

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 los test 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 ambos test 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 los resultados obtenidos en las encuestas de reflexión (tabla 2).

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 asegura poder explicarlo.

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.

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 ambos test. 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.

Capítulo 5: La resolución de problemas basada en el método de Polya usando el Pensamiento Computacional y Scratch con estudiantes de Educación Secundaria

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

Variable	Valor	Indicador	Resultados pretest		Resultados posttest	
			(porcentaje)		(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

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Ejecución del plan	[0-1]	3.1 Antes de cada operación explica lo que va a realizar.	50	50	50	50
		3.2 Identifica de manera explícita 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	5,6	44,4	55,6

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 el test final. Analizando los datos de la encuesta destaca el aumento del alumnado que redacta la solución del problema.

A la hora de trabajar la resolución de problemas es importante que el estudiante sea consciente de la importancia de comprobar, tanto 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.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Este análisis detallado de las distintas subvariables es debido a las limitaciones propias del diseño cuasi-experimental 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 los test sea estadísticamente significativa, se ha realizado una 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 et al., 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, 5° y 6° de Educación primaria, para reforzar metodologías activas, la motivación y la utilidad del proceso de aprendizaje (Sáez-López et al., 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., 2014a) .

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 a concluir esta apreciación.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

La inclusión del Pensamiento Computacional es una alternativa 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 que para una efectiva transmisión de las habilidades cognitivas (Jaeggi et al., 2014). El diseño de este tipo de estrategias despiertan 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 (Lerís y Sein-Echaluce, 2011).

Limitaciones de la investigación

El propio diseño cuasi-experimental 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 cuasi-experimento 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 obtenido, la evolución natural de este trabajo es un análisis comparativo con un grupo control que siga una metodología tradicional para trabajar

Capítulo 5: La resolución de problemas basada en el método de Polya usando el Pensamiento Computacional y Scratch con estudiantes de Educación Secundaria

contenidos de la asignatura de Matemáticas y un grupo experimental que trabaje estos mismos contenidos con Scratch.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Capítulo 6: Pensamiento Computacional con Scratch. Recurso para trabajar Geometría en 5° de Educación Primaria

Resumen

En este segundo estudio se analizan los resultados obtenidos en una experiencia empírica en la que se trabaja el Pensamiento Computacional con el software educativo de Scratch para abordar contenidos de Geometría en Educación Primaria. En muchas ocasiones, estos contenidos son repetitivos y con poca aplicación práctica; utilizar este recurso como herramienta de aprendizaje nos permite plantear un enfoque más dinámico, motivador y eficaz para los estudiantes. La experiencia se ha desarrollado con un total de 66 estudiantes de 5° curso de Educación Primaria de 3 centros educativos diferentes. Se ha realizado un diseño de grupo control no equivalente con medidas de pretest sustitutivas. Para poder realizar este trabajo, en primer lugar el profesorado implicado ha recibido formación inicial. Posteriormente, los estudiantes han realizado actividades genéricas para familiarizarse con el software educativo y por último unas actividades concretas de Geometría. Los resultados obtenidos evidencian un proceso de aprendizaje más positivo en los estudiantes que han trabajado con Scratch, destacando el aspecto motivador y evidentemente práctico de este recurso.

Introducción

En los últimos años, a nivel europeo ha habido un especial interés en motivar al alumnado a desarrollar su formación en el ámbito STEM (siglas en inglés de Ciencia, Tecnología, Ingeniería y Matemáticas) promoviendo desde las administraciones públicas estrategias educativas de este tipo. Un ejemplo es el proyecto Scientix, que contó con financiación del programa de I+D Horizonte 2020 de la Unión Europea para promover y respaldar la colaboración entre docentes, investigadores del ámbito de la enseñanza, legisladores y otros profesionales de la docencia (Adina Nistor, Agueda Gras-Velazquez, Noelle Billon, 2018). Para Becker y Park (2011) el desarrollo de prácticas educativas en el ámbito STEM tienen un efecto positivo en el proceso de aprendizaje del alumnado, desarrollándose de manera más activa y centrada en el propio sujeto. Además, la inclusión de este tipo de enfoque curricular proporciona un marco constructivista para el desarrollo del conocimiento del alumnado, permitiendo el desarrollo de competencias científicas y contextualizando estas prácticas como una aproximación para la enseñanza de las ciencias, tecnologías, ingenierías y Matemáticas de forma interdisciplinar a través de experiencias didácticas aplicadas al mundo real (García et al., 2017).

En este contexto, el Pensamiento Computacional ha tomado mucho protagonismo en las aulas durante los últimos años, tanto en Educación

Capítulo 6: Pensamiento Computacional con Scratch. Recurso para trabajar Geometría en 5° de Educación Primaria

Primaria como en Secundaria. El desarrollo de este tipo de habilidades de razonamiento, como la abstracción, la resolución de problemas, identificación de patrones o el razonamiento lógico (Grover y Pea, 2018), favorece un aprendizaje en el que el estudiante establece una conexión entre el mundo tal y como es fuera del centro educativo y de sus experiencias en el aula, favoreciendo así el desarrollo de unas competencias que lo prepara para ese mundo real (Suárez Zapata et al., 2018). Asimismo, el uso de recursos educativos asociados al Pensamiento Computacional, como el lenguaje de programación visual Scratch, favorece el desarrollo de procesos matemáticos como el razonamiento o la resolución de problemas a la vez que genera un entorno de aprendizaje motivador (Calao et al., 2015).

La inclusión de las destrezas propias de un programador en el proceso de aprendizaje, no solo quedan en la visión utópica de Seymour Papert (1980), sino que estudios recientes como el realizado por Syslo y Kwiatkowska (2014) abogan por llevar más allá la riqueza tradicional de los contenidos de Matemáticas conectándolos con el Pensamiento Computacional. De un modo más específico, otros estudios sugieren que el desarrollo del Pensamiento Computacional en el aprendizaje de las Matemáticas tiene una importante influencia positiva cuando construyen sus propias aplicaciones (Ke, 2014; Kong, 2019; Molina et al., 2020). Por otro lado, también son muchos los ejemplos que encontramos en la última década en los que se analiza el

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

impacto sobre el proceso de aprendizaje con herramientas de Pensamiento Computacional, lo cual ayuda a afianzar que la inclusión de este recurso facilite el desarrollo transversal de habilidades en cualquier disciplina del ámbito STEAM (Barr y Stephenson, 2011).

El presente trabajo intenta avanzar en el estrecho vínculo entre razonamiento matemático y Pensamiento Computacional, una relación que siempre genera cierta discusión (Kotsopoulos et al., 2019). Si bien, este enfoque permite desarrollar contenidos de Matemáticas a través de una experiencia de aprendizaje integradora y transversal, dando cabida al desarrollo de competencias más allá de la meramente matemática y que promueve una formación de nuestros estudiantes mucho más completa y práctica (Rodríguez-Martínez et al., 2020).

Siguiendo con la contextualización del presente trabajo podemos decir que tiene como marco estructural de referencia una intervención llevada a cabo en Reino Unido para incorporar el Pensamiento Computacional como parte del currículo y dar respuesta a necesidades en materia de educación matemática en la etapa de Educación Primaria (Clark-wilson et al., 2016). Igualmente, en España se ha desarrollado durante el curso académico 2018-2019 una iniciativa similar denominada Escuela de Pensamiento Computacional coordinada por el Instituto Nacional de Tecnologías Educativas (INTEF) y que

está enmarcada en el programa de la Comisión Europea denominado Digital Education Action Plan (Vaikutytė-Paškauskė, Justina Vaičiukynaitė y Donatas, 2018). Una de las conclusiones de este programa (Ministerio de Educación y Formación Profesional, 2019) apunta a que la inclusión de actividades de programación en Educación Primaria en el área de las Matemáticas hace que los estudiantes no solo desarrollen habilidades propias del Pensamiento Computacional, sino que además mejoran su razonamiento matemático en mayor medida que otros estudiantes que han trabajado los mismos contenidos utilizando otros recursos no relacionados con la programación.

El desarrollo de este tipo de experiencias se ha facilitado mucho en la última década gracias a poder incorporar el potencial que tiene el expresar ideas Matemáticas a través de la creación de programas informáticos en edades tempranas, sin necesidad de aprender complicadas sintaxis más indicadas para cursos superiores. La naturaleza de estos recursos ha cambiado, modificando los estereotipos de la idea de programación (Hoyles y Noss, 1996), sin olvidar la importancia del paradigma del construccionismo en el cual el aprendizaje se considera como una construcción de estructuras y es más efectivo cuando el estudiante desarrolla de manera activa la creación de un resultado final tangible y con el que puede interactuar (Papert y Harel, 1991).

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Cuando se habla de enseñanza de la Geometría, ya sea en Educación Primaria como en Secundaria, existe un claro consenso sobre que el aspecto experimental, práctico o manipulativo es esencial para fortalecer el aprendizaje y comprensión de las propiedades geométricas. Pero también es sabido que en muchos casos la enseñanza de la Geometría queda relegada a un plano secundario, situándose en el tramo final del curso escolar y con menos tiempo de trabajo del que sería conveniente. A pesar de que en la antigua Grecia era el centro de las Matemáticas, hoy en las escuelas recibe relativamente poca atención, estando el foco más centrado en la aritmética y álgebra, como ocurre en Educación Primaria y Secundaria respectivamente (Sinclair & Bruce, 2015).

Además, en muchas ocasiones, las clases de Geometría se basan en resolver problemas con demasiadas reglas y procedimientos, estando mayormente apoyados en el uso de expresiones algebraicas o en unos básicos y simples dibujos o esbozos que en muchas ocasiones pueden incluso llegar a confundir o distorsionar la percepción espacial del estudiante. La posibilidad de utilizar recursos tecnológicos permite a todo estudiante dibujar, manipular y analizar elementos geométricos y propiedades con facilidad favoreciendo su atención y desarrollando habilidades cognitivas de mayor nivel (Kesan y Caliscan, 2013). Sin la necesidad de sustituir a metodologías tradicionales de carácter manipulativo y que fomenten el razonamiento y percepción geométrica de

nuestro entorno, el desarrollo y la innovación didáctica de la Geometría debe estar acompañada de la inclusión de recursos tecnológicos. Es necesaria una estrecha relación entre aprendizaje, generación de conocimiento, innovación continua y uso de nuevas tecnologías (Cobo, 2008).

Material y métodos

La hipótesis de partida de este trabajo se centra en que el desarrollo de actividades relacionadas con el Pensamiento Computacional en el área de las Matemáticas favorece el desarrollo de los contenidos curriculares y mejora el proceso de aprendizaje. Para poder contrastar dicho punto de partida, se ha diseñado y desarrollado esta experiencia empírica con el objetivo de analizar el grado de mejoría del proceso de aprendizaje cuando los contenidos del bloque de Geometría, en la asignatura de Matemáticas, son abordados de manera transversal con el software educativo Scratch en el 5° curso de Educación Primaria. Como objetivos específicos se plantean los siguientes puntos:

1. Analizar la motivación del alumnado realizando actividades de Pensamiento Computacional con Scratch.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

2. Potenciar el uso de nuevas metodologías y estrategias de aprendizaje a través de la inclusión curricular del software de programación visual Scratch.
3. Evaluar el desarrollo de la competencia y el razonamiento matemático en Geometría trabajando actividades de Pensamiento Computacional como recurso transversal de aprendizaje con alumnos de 5º curso de Educación Primaria.

Para este trabajo se ha diseñado un proceso de metodología mixta (B. Johnson et al., 2007). Un análisis cuantitativo permitirá analizar el progreso y evolución de la competencia matemática en el alumnado, mientras que un análisis cualitativo complementario ayudará a conocer la realidad desde el punto de vista de sus protagonistas, alumnado y profesorado, a fin de comprender nuestro objeto de estudio analizando las percepciones e interpretaciones de las personas que intervienen en la realidad investigada (Colás y Buendía, 1992). La apuesta por esta combinación de técnicas cualitativas y cuantitativas proporciona unos resultados caracterizados por una perspectiva complementaria para el análisis de los diferentes objetos de estudio caracterizados en este trabajo (Morse y Niehaus, 2009) .

El proceso empírico se desarrolla mediante un diseño cuasi-experimental, con un grupo de control no equivalente (Campbell y Fiske, 1959) y medidas de

pretest y posttest (Campbell y Stanley, 1966). El proceso cualitativo se ha llevado a cabo con la ayuda del diseño de una serie de entrevistas semiestructuradas para recoger información tanto del alumnado como del profesorado implicado en la experiencia, pudiendo así tener más detalle de todo el proceso de trabajo realizado por ambos tipos de participantes.

El estudio se ha realizado con estudiantes de edades comprendidas entre 9 y 11 años, de 5º curso de Educación Primaria. Se han involucrado tres centros públicos de Educación Primaria de características similares, del mismo entorno urbano y con el mismo nivel sociocultural.

Desarrollo de la experiencia

El punto de partida de la experiencia empírica fue la formación del profesorado involucrado en la práctica y la sensibilización de los participantes. Esta primera fase del trabajo ha constituido un pilar fundamental, tanto para el resultado de la experiencia como para la implicación del profesorado, ya que la formación docente es un aspecto fundamental para la mejora del sistema educativo en general (Barber y Mourshed, 2007) y, en particular, la formación enfocada al desarrollo de la competencia digital docente se considera imprescindible para desarrollar habilidades tecnológicas y cognitivas que les permitan hacer frente a diferentes retos de aprendizaje que se planteen desde distintos ámbitos a lo

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

largo de la vida (Selwyn, 2013). Esta formación se llevó a cabo durante 4 sesiones presenciales y un acompañamiento no presencial para hacer el seguimiento del trabajo.

Los grupos experimentales institucionalizaron este proyecto desde el Equipo Directivo y, en contacto con las familias a través del Consejo Escolar, fijaron una hora de trabajo a la semana en el horario del grupo para completar todas las actividades planteadas. Este tipo de estrategias organizativas favorecen el desarrollo para adoptar nuevas estrategias de aprendizaje.

Una segunda fase, consistió en la implementación de la experiencia. Pero antes de comenzar con las actividades específicas del proyecto, los estudiantes realizaron el test inicial (Anexo 4), compuesto por un total de 12 preguntas sobre contenidos de Geometría trabajados en Educación Primaria, y llevaron a cabo el trabajo de aula con Scratch durante tres meses.

Para el desarrollo del trabajo con Scratch, el alumnado comenzó con las denominadas actividades de introducción para conocer a utilizar el programa, así como estructuras de control e instrucciones básicas para abordar con cierta autonomía las siguientes actividades referentes a contenidos de Geometría. En primer lugar, realizaron una actividad para representar un cuadrado a partir de las coordenadas de los vértices. A continuación se programó el movimiento de unos personajes a partir medidas angulares. Para la tercera actividad se

representaron rectas paralelas y perpendiculares programando el movimiento de traslación y giro correspondiente. En cuarto lugar, se realizó una representación de diferentes tipos de ángulos. La quinta actividad consistió en representar figuras planas a partir de sus ángulos. Por último, programaron una traslación y un giro de un objeto para obtener una imagen simétrica respecto de un determinado eje dado previamente. Las actividades han estado dirigidas por el profesorado, si bien los estudiantes tenían libertad para completarlas de manera personal y creativa, algo que no puede dejarse de lado cuando queremos trabajar con este tipo de recursos.

Por último, se desarrolló una tercera fase de recogida de información en la que se realizó el test final (Anexo 4) que nos ha permitido realizar el análisis cuantitativo. Los grupos de control realizaron el test inicial y final en las mismas fechas que los grupos experimentales, habiendo trabajado los contenidos de Geometría de manera tradicional. Ambos test, tenían las mismas actividades, solo que en distinto orden (Tabla 3). En él realizaron actividades para representar puntos en el plano, identificar figuras planas, así como ángulos y sus propiedades, caracterizar rectas paralelas y perpendiculares y realizar representaciones simétricas de una figura dada.

Del mismo modo, para realizar el análisis cualitativo se utilizó como instrumento de recogida de información la entrevista semiestructurada en

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

grupo, entendida como aquella que es guiada por un conjunto de preguntas y cuestiones básicas a explorar, pero ni la redacción exacta, ni el orden de las preguntas están predeterminadas (Valles, 2002). En primer lugar, todo el profesorado realizó la entrevista (Anexo 5) con preguntas enfocadas a recoger sus impresiones acerca de este tipo de prácticas educativas considerando las categorías que se detallan en la tabla 4. Posteriormente, se realizó una entrevista con el alumnado (Anexo 6) participante enfocada a obtener información cualitativa sobre cómo habían trabajado en clase durante el desarrollo de las sesiones con Scratch. Estas entrevistas se realizaron por parejas y las cuestiones a explorar estaban establecidas a partir de las mismas categorías trabajadas en la entrevista realizada al profesorado.

Tabla 3: Contenidos incluidos en las preguntas

Pregunta	Contenidos en cada pregunta.
1	Identificación ejes de coordenadas
2	Identificación de las coordenadas de un punto
3 y 4	Representación de puntos en el plano
5	Identificación de distintos tipos ángulos
6 y 7	Ángulos complementarios y suplementarios
8 y 9	Representación de rectas paralelas y perpendiculares
10	Representación figuras geométricas
11 y 12	Caracterización y representación de figuras simétricas

Tabla 4: Categorías definidas en el análisis cualitativo

Categoría	Descripción
Inclusión curricular	Posibilidad de incluir de manera transversal actividades de Pensamiento Computacional.
Motivación	Satisfacción de los participantes trabajando este tipo de actividades.
Metodología	Influencia en el cambio metodológico.
Formación inicial	Importancia de formación inicial para el desarrollo de la experiencia.
Atención a la diversidad	Posibilidad de adaptación a los diferentes ritmos de aprendizaje.

Análisis y resultados

Los resultados obtenidos tanto para los grupos de control (CEIP C1 y C2) como para los grupos experimentales (CEIP E1 y E2) en ambos test, inicial y final, pueden verse en la figura 3. En la figura 4 se muestra la diferencia obtenida entre ambos test para cada uno de los grupos.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

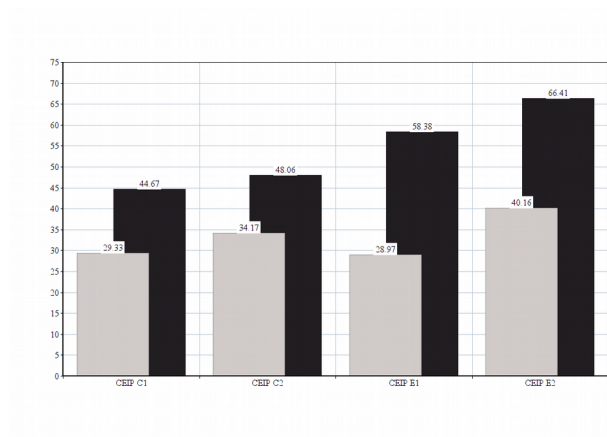


Figura 3: Medida test inicial y test final en %

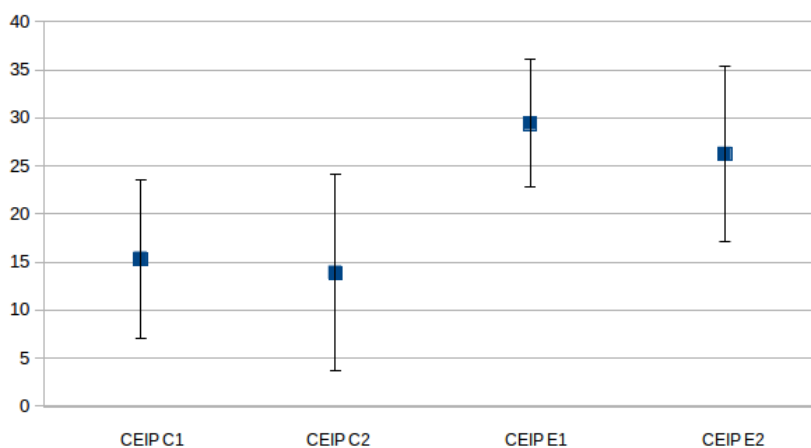


Figura 4: Diferencia de los test de cada grupo

Estos resultados inicialmente muestran una diferencia positiva en la realización de las pruebas en todos los grupos, si bien en los grupos experimentales esta diferencia es mayor. Para dar validez a estos resultados y concluir si estas diferencias son significativas o no, observamos los resultados del análisis estadístico realizado. En primer lugar, el agrupamiento cumple el

supuesto de homocedasticidad para la aleatoriedad del grupo, obteniendo los resultados expuestos en la tablas 5 y 6.

Tabla 5: Homogeneidad de grupos

	Estadístico de Levene	Df1	Df2	Significación
Valor	0,69	3	62	0,562

Tabla 6: Distribución normal de la muestra

	Valor
N	66
Media	21,21
Desviación estándar	18,28
Diferencias más extremas-absoluto	0,08
Diferencias más extremas-positivo	0,04
Diferencias más extremas-negativo	-0,08
Z de Kolmogorov-Smirnov	0,65
Sig. Asint. (2-colas)	0,797

A la vista de estos resultados, se acepta la asignación aleatoria del grupo, pudiéndose considerar la hipótesis nula de que las varianzas son semejantes gracias a la significancia obtenida en la prueba de Levene ($p = 0.562$) (Montgomery, 1991). Igualmente, para la distribución normal de la muestra, obteniéndose una significancia asintótica a dos colas de 0,797.

Por último, para concluir que estas diferencias son significativas conviene hacer una comparación con el análisis de la varianza ANOVA de un factor

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

(Castañeda et al., 2010). A la hora de realizar la comparación entre grupos se obtiene un valor de significancia de 0'024, lo cual indica que la diferencia entre los grupos es estadísticamente significativa al ser menor de 0'05; se ha tomado un intervalo de confianza del 95% para descartar la hipótesis nula de que los grupos son iguales.

Para obtener la valoración cuantitativa y poder realizar el análisis expuesto anteriormente de las diferencias entre grupos, tanto las pruebas iniciales como finales fueron analizadas con la parrilla de corrección que se muestra en la tabla 7.

Tabla 7: Parrilla de corrección para el test

Puntos en el plano				
Pregunta	Descriptor		Valoración	
1	Identifica los ejes de coordenadas correctamente como <i>eje X</i> y <i>eje Y</i>		1/0	
2	Identifica las coordenadas de un punto en el plano que no presenta confusión (4, 2)		1/0	
3	Representa correctamente puntos en el plano conociendo sus coordenadas	(0, 0)	1/0	
		Y=1	1/0	
		Misma altura	Y=2	1/0
			Y=3	1/0
		X negativa (-1, 2)	1/0	
4	Representa puntos en el plano a partir de sus coordenadas para formar un cuadrado		1/0	

Capítulo 6: Pensamiento Computacional con Scratch. Recurso para trabajar Geometría en 5° de Educación Primaria

Ángulos			
5	Identifica ángulos a partir de una representación gráfica	Identifica un ángulo < 90°	1/0
		Identifica un ángulo > 90°	1/0
		Identifica un ángulo de 90°	1/0
		Identifica un ángulo de 180°	1/0
6	Conoce la definición de ángulos complementarios y suplementarios	Sabe cuánto suman los dos ángulos	1/0
		Sabe cuánto suman los dos ángulos y los relaciona con su representación gráfica	1/0
7	Interpreta gráficamente la suma de ángulos complementarios y suplementarios		1/0
Rectas			
8	Conoce el concepto de recta perpendicular	Sabe dibujar una recta perpendicular a otra y conoce el ángulo que se forma entre ellas	1/0
		Sabe dibujar una recta perpendicular a otra, pero no conoce el ángulo que se forma entre	0,5/0

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

		ellas	
9	Conoce el concepto de recta paralela	Sabe dibujar una recta paralela y sabe que puede dibujar más	1/0
		Solo dibuja una recta paralela	0,5/0
Figuras planas			
10	Conoce las figuras planas de 3, 4 y 6 lados	Conoce las figuras por el número de lados y por su representación gráfica	1/0
		Solo conoce las figuras por el número de lados o por su representación gráfica	0,5/0
Simetría			
11	A partir de una figura compuesta, hace una representación simétrica		1/0
12	Identifica la relación de simetría como un desplazamiento y un giro		1/0

Todo el análisis cuantitativo se completa con la información obtenida del proceso cualitativo. Como se ha dicho anteriormente, este proceso se ha desarrollado con una serie de entrevistas tanto al profesorado como al

alumnado que han participado en la experiencia. De la información recogida cabe destacar que los participantes ven muy positiva la inclusión curricular de este tipo de recursos que facilitan el desarrollo de habilidades de Pensamiento Computacional de manera transversal, lo cual promueve un cambio metodológico en el aula:

P2: “Es un recurso fácil de manejar. No requiere mucho tiempo para aprender a usarlo y poder aplicarlo en Educación Primaria. Facilita mucho el cambio metodológico”.

A la hora de realizar un proceso de innovación educativa, el profesorado ha manifestado la importancia de recibir una formación inicial adecuada, tanto para el manejo de la herramienta como en un enfoque metodológico. Igualmente, es importante su visión en cuanto al papel que han desempeñado los estudiantes en el desarrollo de las sesiones, destacando la implicación y participación activa en la colaboración con sus compañeros y compañeras. Ha sido frecuente que en el desarrollo de las sesiones cada estudiante trabaje a un ritmo diferente, lo cual ha sido muy positivo ya que les ha permitido colaborar para que los distintos ritmos de aprendizaje queden atendidos adecuadamente; se mejora la atención entre alumnado y profesor y la relación entre el propio alumnado.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

De la información recabada en las entrevistas a los estudiantes, uno de los aspectos más destacados es que todos coinciden, al igual que el profesorado, en la motivación de trabajar los contenidos de Geometría a través de herramientas digitales. Además, destacan el hecho de que no sea un simple recurso que disponga de manera interactiva actividades tradicionales, sino que el poder hacer sus propias construcciones, Scratch les permite realizar el trabajo de manera más personal y creativa.

E8: “El libro y el cuaderno está bien, pero es mejor usar Scratch. Es más divertido. Lo otro parece un poco anticuado”.

E10: “Aprender Matemáticas así es más divertido porque puedes poner tus propios personajes, escenarios y acciones”.

El alumnado ha valorado positivamente trabajar con un recurso digital que ha facilitado que su aprendizaje parta de una experiencia activa que les permita ser creadores de su propio conocimiento.

Discusión y conclusiones

Los resultados mostrados sugieren que el diseño de las actividades y la metodología empleada favorecen el desarrollo de los contenidos de Matemáticas realizando actividades que desarrollan competencias relacionadas con el Pensamiento Computacional gracias al software educativo

Capítulo 6: Pensamiento Computacional con Scratch. Recurso para trabajar Geometría en 5° de Educación Primaria

Scratch. Igualmente, se ha favorecido el desarrollo de otras competencias importantes para el desarrollo de un razonamiento matemático como la capacidad de crear contenidos multimedia con elementos de programación. Aunque estos resultados sean positivos y considerando las características del propio diseño cuasi-experimental, no siempre se podrá asegurar que incluir este tipo de metodologías y recursos pueda ser efectivo para cualquier aula en cualquier contexto o asignatura. Este proceso empírico se ha realizado para reforzar y dar importancia al desarrollo de prácticas de Pensamiento Computacional que ayuden a trabajar contenidos en Educación Primaria, así como para dar solidez y realidad a la posibilidad de un cambio metodológico real que favorezca el desarrollo de habilidades asociadas al ámbito STEM en esta etapa educativa. Por lo tanto, estos resultados sirven para tomar como ejemplo positivo el desarrollo de contenidos de Geometría, como son las medidas angulares, las figuras geométricas en el plano o las propiedades de simetría empleando recursos y planteamientos que faciliten el desarrollo de habilidades de Pensamiento Computacional de manera interdisciplinar en el ámbito STEM en Educación Primaria. Cuando se desarrollan prácticas de Pensamiento Computacional en un ámbito STEM de carácter transversal, se potencia el desarrollo de este tipo de habilidades como una competencia general, haciendo que los estudiantes adquieran las habilidades necesarias para resolver problemas futuros (Li et al., 2020).

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

De los resultados obtenidos, no solo hay que destacar el proceso cuantitativo. También es importante destacar la motivación que ha tenido el profesorado participante, lo cual ha facilitado que el alumnado participante se haya sentido cómodo, incentivado y con ganas de ser partícipe en un aprendizaje más activo, mejorando su comportamiento e interacción con el docente. Se ha conseguido que se mejoren los resultados académicos a la vez que se mejora la implicación en la clase del alumnado, pasando de un rol pasivo y tradicional a un rol activo capaz de crear su propio aprendizaje. Quizá, el simple hecho de llevar al aula un cambio metodológico, independientemente del recurso empleado, hace que el resultado de un análisis cuantitativo sea favorable. En este caso, puede considerarse como un aspecto positivo ya que plantea la posibilidad de ver la inclusión curricular de prácticas educativas asociadas al Pensamiento Computacional como una posibilidad de cambiar la metodología de trabajo tradicional, aprovechando la motivación del alumnado dada por la utilización de enfoques educativos innovadores con el uso de estas herramientas (Stoitsov y Stoitsova, 2019).

El trabajo realizado está enfocado al desarrollo de contenidos de Geometría para el estudio de las Matemáticas. Pero como otros ejemplos e investigaciones similares, ayuda a reforzar con evidencias positivas la importancia de realizar trabajos en los que se desarrollen actividades de Pensamiento Computacional como herramienta transversal que permita

abordar contenidos de Matemáticas (Weintrop et al., 2016; Gadanidis et al., 2016; Orton et al., 2016; Hickmott et al., 2017; Benton et al., 2017; Holmes et al., 2018).

Pero con este trabajo no solo se trata la inclusión positiva de actividades para trabajar el Pensamiento Computacional en el proceso de aprendizaje de nuestros estudiantes. También se pone de manifiesto una realidad muy importante para que nuestros estudiantes desarrollen las competencias propias de cualquier ciudadano del siglo XXI, como es la necesidad de emplear recursos tecnológicos en el aprendizaje de disciplinas STEM. El avance de esta tecnología ha permitido que todo el alumnado tengan un acceso a ella mucho más sencillo y eficaz, permitiendo que se pueda aplicar a cualquier campo, desarrollando nuevas representaciones de distintos fenómenos y situaciones, favoreciendo el hecho de que contenidos que de manera tradicional puedan resultar más complejos se puedan abordar de una forma más accesible y práctica para nuestros estudiantes a la vez que se desarrolla su competencia digital.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria.

El Pensamiento Computacional en el ámbito educativo

La importancia de desarrollar el Pensamiento Computacional como un elemento curricular es un reflejo de unas necesidades que la sociedad está demandando, tanto a nuestro alumnado como al profesorado, y que ha estado ligado al desarrollo tecnológico que se se manifiesta en todos los sectores de nuestra vida cotidiana (Del Álamo Venegas et al., 2021). Obviamente, no todas las profesiones a las que accederá nuestro alumnado en un futuro reciente van a desarrollarse utilizando ordenadores o cualquier tecnología programable (Adell Segura et al., 2019), pero la conceptualización actual del Pensamiento Computacional va más allá de aprender a programar o a realizar procesos utilizando un ordenador: aborda y desarrolla los procesos y habilidades necesarias para plantear y resolver problemas complejos (Wing, 2008).

Cuando se habla de Pensamiento Computacional, es imprescindible hacer referencia a las aportaciones dadas en el ámbito de las Ciencias de la Computación por la ingeniera estadounidense Jannette Wing. Sus contribuciones dadas en la comunicación presentada en la ACM (Association

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

for Computing Machinery de Estados Unidos) de 2006 marcaron un punto de partida muy importante para muchas de las investigaciones realizadas en los últimos 15 años. A partir de aquí, se han ido sucediendo muchas definiciones y contextos para intentar llegar a un consenso tanto en la definición como en el marco conceptual del Pensamiento Computacional. La propia Jannette Wing continuó estableciendo una definición haciendo otro aporte más genérico diciendo que el Pensamiento Computacional es un proceso de pensamiento utilizado para formular un problema y expresar su solución o soluciones en términos que una computadora pueda aplicar de manera efectiva (Wing, 2014). Más reciente, y haciendo un análisis de los trabajos de investigación en la bibliografía sobre Pensamiento Computacional y su definición, Moreno y colaboradores (2019) proponen una definición para el Pensamiento Computacional concibiéndolo como la capacidad de formular y representar problemas para resolverlos mediante el uso de herramientas, conceptos y prácticas de la disciplina informática, como la abstracción, la descomposición o el uso de simulaciones.

En referencia a un marco conceptual, entre otros, destaca el aportado por Grover y Pea (2018). En este, se sitúa al Pensamiento Computacional como un elemento mediador en las prácticas de ámbito STEM (Science, Technology, Engineering and Mathematics). Este marco conceptual establece una serie de elementos, divididos en conceptos y prácticas, que ayudan a entender qué es el

Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria.

Pensamiento Computacional y qué proceso de razonamiento supone. Como conceptos de Pensamiento Computacional se incluyen la lógica y el pensamiento lógico, los algoritmos y el pensamiento algorítmico, los patrones y el reconocimiento de patrones, la abstracción y generalización, la evaluación y la automatización. Las prácticas de Pensamiento Computacional a las que refiere este marco conceptual intentan acercar al trabajo que puede realizar un informático cuando se enfrenta a la resolución de un problema. Se considera dentro de esta categoría la descomposición de un problema, la creación de artefactos computacionales, probar y depurar, la mejora reiterada, la colaboración y creatividad.

Además de las distintas sucesiones y concepciones del Pensamiento Computacional, para poder trabajar de manera efectiva en el aula y realizar una adecuada inclusión curricular es importante abordar la evaluación de este tipo de destrezas en las investigaciones empíricas, utilizando distintas herramientas de medición validadas por expertos que permitan al profesorado analizar el proceso de aprendizaje y adquisición de habilidades de un estudiante (Grover & Pea, 2013). Autores como Román-González y colaboradores (2017) plantean la posibilidad de diferenciar distintos tipos de herramientas para evaluar el Pensamiento Computacional. Aunque en muchas ocasiones, será interesante combinar distintas herramientas de evaluación para

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

tener una visión más completa de la adquisición de habilidades y de la evolución en el aprendizaje de nuestro alumnado (Grover, 2015).

Formación inicial del profesorado en Pensamiento Computacional

Para que el alumnado pueda desarrollar de manera adecuada las habilidades propias del Pensamiento Computacional, es necesaria su inclusión académica en distintos niveles tanto de Educación Primaria como Secundaria (Barr y Stephenson, 2011; Grover y Pea, 2013; Yadav et al., 2016). En la actualidad podemos encontrar muchos estudios que analizan distintos procesos de trabajo en los que se desarrollan este tipo de habilidades de manera transversal e integrada en el currículo. Aunque cabe destacar el informe elaborado por la Comisión Europea en el que se analizan los efectos de la inclusión formal del Pensamiento Computacional en distintos países de la Unión Europea entre 2016 y 2021 en el cual se apela a que la inclusión curricular debe hacerse en todos los niveles educativos de manera global, con el fin de articular una estrategia integradora desde Educación Primaria hasta concluir la educación obligatoria de manera progresiva y acorde a cada edad (Bocconi et al., 2022). Igualmente, se plantea la importancia de la formación del profesorado, sobre todo para aquel sector menos familiarizado con el uso de recursos de Pensamiento Computacional, con el fin de asegurar un desarrollo profesional en este campo. De manera concreta, en el sistema educativo español, este

Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria.

aspecto cobra especial interés al ver la importancia que tiene el Pensamiento Computacional en la actual legislación educativa que regula las enseñanzas mínimas en Educación Primaria (Ministerio de Educación y Formación Profesional, 2022a). En lo que se describe como las habilidades imprescindibles para cualquier estudiante, que son las competencias clave que derivan del marco de referencia europeo (López Rupérez, 2022), se encuentra la competencia digital. Esta competencia hace referencia a que el alumnado adquiera una correcta alfabetización la resolución de problemas y el Pensamiento Computacional y crítico. Además, de manera más concreta, esto se refleja en la descripción de diferentes competencias específicas al definir el desempeño de cada estudiante en un área, ámbito o materia al trabajar los saberes básicos. Dentro de las especificaciones para la enseñanza de las Matemáticas, en la LOMLOE (2022) uno de los ejes principales de organización es la resolución de problemas, proceso que se relaciona directamente con el Pensamiento Computacional. Igualmente que en la descripción general, se incluye en las competencias específicas para establecer la relación con el perfil de salida. Por otro lado, este concepto de Pensamiento Computacional aparece para definir distintos criterios de evaluación y saberes básicos, en los distintos ciclos de Educación Primaria. Por tanto, se entiende que hay una necesidad formativa, tanto inicial como continua, para el profesorado de Educación Primaria que le ayude a adquirir adecuadamente los

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

conceptos necesarios para desarrollar de manera integrada en el currículo prácticas de Pensamiento Computacional.

La formación del profesorado en este campo presenta ciertas limitaciones (Demartini et al., 2020), por lo que se plantea el hecho de actualizar las destrezas de los futuros docentes en este tema (Esteve-Mon et al., 2020) para adecuarse a las necesidades que hoy demanda la educación, siendo en general un reto el desarrollo profesional de docentes cuyas competencias permitan llevar el Pensamiento Computacional al aula de Educación Primaria y Secundaria (Prendes Espinosa y Cerdán Cartagena, 2020). Existen indicios de los beneficios que supone la inclusión de prácticas de Pensamiento Computacional en la formación inicial del profesorado para adquirir los conocimientos necesarios en este campo y poder aplicarlo al aula, pero se necesita la implicación no solo del profesorado de informática, sino de aquellos docentes que trabajan en distintas áreas y que adquieran los conocimientos necesarios en este campo (Yadav et al., 2014). En este sentido, es importante que los profesores en formación analicen y comprendan qué es el Pensamiento Computacional en el contexto de la asignatura en la que quieran abordar su aprendizaje, siendo necesario que conozcan a fondo tanto su propia disciplina como los distintos conceptos de Pensamiento Computacional que pueden aprender los estudiantes en el aula (Yadav et al., 2017). Por lo tanto, la cuestión de cómo desarrollar habilidades de

Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria.

Pensamiento Computacional en la formación inicial del profesorado es un tema importante cuya investigación es oportuna y necesaria (Gretter & Yadav, 2016).

En este sentido, podemos encontrar investigaciones como la de Yadav et al. (2011) que se basan en cuestionarios actitudinales para analizar la percepción y conocimientos que pueden tener los futuros docentes respecto al Pensamiento Computacional utilizando cuestiones y actividades que muestran los principales conceptos computacionales asociados. Otros estudios también analizan estos aspectos además de la forma en la que se puede integrar el Pensamiento Computacional en diferentes disciplinas (Ateşkan y Hart, 2021). Pero el presente trabajo de investigación, además de este tipo de cuestiones, analiza el grado de desarrollo de diferentes destrezas de Pensamiento Computacional a través de actividades integradas en el contexto de la enseñanza de las Matemáticas en la formación inicial de los docentes. Para ello se han realizado una serie de prácticas con el lenguaje de programación por bloques Scratch, el cual permite trabajar fácilmente conceptos computacionales (Resnick et al., 2009). Para medir el grado de desarrollo de dichas habilidades de Pensamiento Computacional, se ha utilizado la herramienta validada denominada Test de Pensamiento Computacional (TPC) (Román-González, 2015). El objetivo de esta herramienta es medir la capacidad de crear y resolver problemas partiendo de los conceptos básicos de

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

programación y el uso de la sintaxis lógica de estos lenguajes como bucles, funciones, variables o condicionales (Chan et al., 2020).

Metodología

El objetivo principal de este trabajo es analizar el grado de adquisición de las competencias asociadas al Pensamiento Computacional con estudiantes de segundo curso del Grado en Educación Primaria así como analizar la concepción que tienen los futuros docentes de la aplicación de este tipo de prácticas con estudiantes de la etapa de Primaria. La experiencia se ha desarrollado dentro de la asignatura Didáctica de las Operaciones Numéricas y la Medida. . Igualmente, como parte de la formación inicial del profesorado, con este tipo de prácticas se busca promover nuevas formas de llevar a cabo en el aula la enseñanza de las Matemáticas a través de prácticas educativas con un enfoque transversal e integrador.

Se analiza así la perspectiva que aporta a los futuros docentes el uso de este tipo de recursos y el desarrollo de las destrezas asociadas a la hora de buscar nuevas formas de llevar a cabo una enseñanza de las Matemáticas acorde a las necesidades de la sociedad actual. Para poder analizar estos aspectos, se ha diseñado una experiencia empírica en la que los participantes han tenido que realizar una serie de prácticas con Scratch y unos retos de corrección de errores.

Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria.

Los objetivos específicos de este trabajo son los siguientes:

O1. Analizar el grado de desarrollo de diferentes competencias computacionales realizando una serie de prácticas con Scratch.

O2. Valorar la percepción de este tipo de prácticas y recursos educativos en la formación inicial de futuros docentes.

O3. Potenciar el uso de nuevas metodologías y estrategias de aprendizaje trabajando el Pensamiento Computacional de manera transversal en la enseñanza de las Matemáticas.

Para llevar a cabo este proceso empírico se ha trabajado con una metodología mixta (Johnson et al., 2007) en la que el análisis cuantitativo se ha realizado con el TPC. Este test se presenta como una de las herramientas actualmente más utilizadas, basadas en una evaluación con lenguajes de programación por bloques, aunque no está ligada a ningún programa en particular (Cutumisu et al., 2019). En su formato básico, el test está compuesto de 28 preguntas de opción múltiple diferenciadas en categorías según se traten conceptos computacionales como el uso de direcciones básicas y secuencias, bucles de repetición con o sin condicionales, condicionales simples y compuestos (si, si-no), condicionales “mientras” y funciones simples. El test es una prueba de respuesta individual en la que cada pregunta tiene 4 opciones posibles con una

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

sola correcta. Para esta experiencia se ha utilizado la modalidad del TPC que se desarrolla a través de tareas Bebras, compuesto por un total de 30 preguntas. Para crear este test a partir del original TPC, se han seleccionado los 20 ítems más discriminativos (Román-González, 2016), se han añadido 4 de dificultad extra y una selección de 6 tareas Bebras (Dagiene y Futschek, 2008). Estas tareas no son un instrumento de medida propiamente dicho; no miden una aptitud, pero sí son válidas para medir la “competencia” en un sentido similar al que se miden distintas habilidades en unas pruebas PISA (Hubwieser y Mühling, 2014). Para Román-González (2016) estas tareas no se orientan hacia la evaluación del Pensamiento Computacional en un estado aptitudinal puro, sino más bien a la percepción de cómo el sujeto es capaz de transferir dicho Pensamiento Computacional a problemas cotidianos y significativos.

Estos datos se apoyan en los obtenidos a partir de un proceso cualitativo, utilizando un cuestionario para conocer las impresiones de los participantes, lo cual ayudará a conocer la realidad desde el punto de vista de sus protagonistas, a fin de comprender nuestro objeto de estudio analizando las percepciones e interpretaciones de las personas que intervienen en este trabajo (Colás y Buendía, 1992). La apuesta por esta combinación de técnicas cualitativas y cuantitativas proporciona unos resultados caracterizados por una

Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria.

perspectiva complementaria para el análisis de los diferentes objetos de estudio caracterizados en este trabajo (Morse y Niehaus, 2009).

El proceso empírico se desarrolla mediante un diseño cuasi-experimental, con un grupo de control no equivalente (Campbell y Fiske, 1959) y medidas de pretest y postest (Campbell y Stanley, 1966). El proceso cualitativo se ha realizado con un cuestionario (Anexo 7) en el que se recogen aspectos referentes al desarrollo de las prácticas, las características de cada una de las tareas realizadas y aspectos generales sobre el uso y aplicación de prácticas de Pensamiento Computacional para Educación Primaria con el software educativo Scratch.

El estudio se ha realizado con un total de 149 estudiantes del Grado de Educación Primaria que cursan la asignatura de Didáctica de las Operaciones Numéricas y la Medida. La muestra se distribuye como aparece en la tabla 8.

Tabla 8: Distribución de los participantes

	Hombre	Mujer	Total
Grupo control	17	53	70
Grupo experimental	20	59	79
Total	37	112	149

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Es importante destacar que los participantes que han intervenido en esta experiencia tienen un nivel de conocimientos tecnológicos básicos en aspectos generales, pero no relativos a Pensamiento Computacional o prácticas relacionadas en este campo.

Desarrollo de la experiencia

Para el grupo experimental, el desarrollo de la experiencia se ha realizado en un total de 5 sesiones: dos de 90 minutos y otras tres de una hora de duración cada una. En la primera sesión se realizó una introducción para explicar qué es el Pensamiento Computacional y su importancia como recurso educativo para Educación Primaria. Posteriormente se realizó el pretest con la herramienta comentada anteriormente. Durante las siguientes tres sesiones se realizaron las prácticas de trabajo con Scratch. En primer lugar se realizó una actividad denominada *Anima tu nombre* cuya finalidad no es otra que familiarizarse con el entorno de trabajo a la vez que se introducen elementos de programación sencillos como los bucles de repetición. Desde el enfoque matemático, también se utilizaron elementos como medidas angulares o las coordenadas del plano cartesiano para definir ciertas animaciones. En la segunda práctica los participantes aprendieron a definir elementos interactivos y a trabajar conceptos como el paralelismo o los bucles condicionales “si, si no”. La tercera práctica consistió en crear una historia animada, aunque estaba

Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria.

enfocada a dibujar un cuadrado en la pantalla a partir de las coordenadas de los cuatro vértices. Para esto, los participantes emplearon distintas funciones del programa a la vez que trabajaban un concepto muy importante en el ámbito de la programación y la resolución de problemas: la descomposición. Por último, realizaron una práctica para crear un juego para cazar objetos que aparecen de manera aleatoria en la pantalla. Esta sencilla idea permite trabajar con el uso de variables para almacenar la puntuación obtenida, paralelismo, bucles condicionales y coordenadas del plano, entre otras cosas. Todas las prácticas estaban enfocadas a abordar los distintos conceptos que se trabajan en el Test de Pensamiento Computacional de manera integrada y con actividades que se pueden contextualizar en el ámbito educativo, ya que uno de los objetivos de la experiencia empírica es potenciar el uso de nuevas metodologías y estrategias de aprendizaje en las Matemáticas desarrollando el Pensamiento Computacional.

A partir de la segunda sesión práctica, los estudiantes tuvieron que resolver unas actividades planteadas para corregir errores en un código similar a las actividades planteadas en clase. El objetivo era buscar cierta aplicación más autónoma de lo aprendido en las sesiones prácticas con el objetivo de que interiorizaran más los conceptos trabajados.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Por último, la sesión final sirvió para plantear dudas, corregir errores de algunos participantes y completar el test final. Por último, tuvieron que completar el cuestionario con el que se ha recopilado la información cualitativa acerca de cómo consideran este recurso para utilizarlo en el aula de Educación Primaria

Los grupos de control realizaron el test inicial y final en las mismas fechas que los grupos experimentales gracias a la coordinación entre los docentes de ambos grupos.

Análisis y resultados

Los resultados obtenidos, tanto por el grupo experimental como por el de control, en el test realizado al inicio y al final de la experiencia pueden verse en las figuras 5 y 6. Se observa que tanto en el grupo experimental como en el de control hay una mejora en la puntuación obtenida en el test final respecto al inicial, si bien la del grupo experimental es más significativa. Pero también cabe destacar que ambos grupos parten de una situación muy similar, ya que presentan muy poca diferencia en la puntuación del test inicial. Se entiende que el grupo usado como control siempre va a presentar cierta mejora por el simple hecho de realizar una segunda vez la prueba, ya que los participantes tienen la posibilidad de plantearse nuevamente algunas dudas que hayan tenido en la realización del test y mejorar alguna respuesta.

Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria.

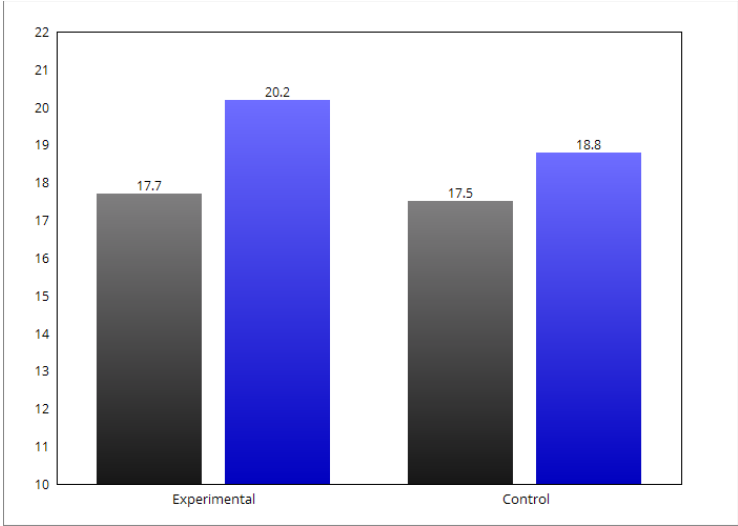


Figura 5: Promedios de puntuaciones obtenidas

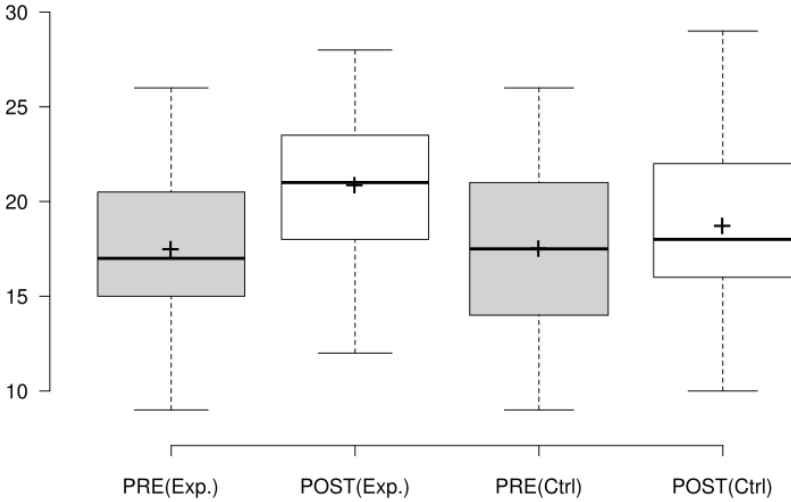


Figura 6: Diagrama de cajas de las puntuaciones obtenidas

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Los datos numéricos correspondientes al diagrama de cajas se detallan en la tabla 9. En estos datos podemos ver lo que refleja el diagrama, que el grupo experimental ha tenido una mejora más equilibrada en todo el grupo. El grupo control, ha presentado mayor dispersión al tener resultados más alejados de la mediana, sin variar mucho este valor. Mientras que el grupo experimental ha presentado un aumento casi constante en los distintos valores que nos ayudan a ver la distribución y comparativa de los datos obtenidos antes y después de la experiencia.

En la tabla 10 se muestran los valores numéricos de la puntuación media de cada prueba. El promedio de la diferencia entre test inicial y final para el grupo de control es de 1.3, mientras que para el experimental es de 2.5, por lo que la intervención práctica tiene claramente unos efectos destacados en el desarrollo de las habilidades analizadas con esta prueba.

Tabla 9: Datos del diagrama de cajas

	PRE (Exp.)	POST (Exp.)	PRE (Ctrl.)	POST (Ctrl.)
Extremo superior	26	28	26	29
Extremo inferior	9	12	9	10
Cuartil 1º	15	18	14	16
Cartil 3º	20,5	23,5	21	22
Mediana	17	21	17,5	18

Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria.

Tabla 10: Puntuación obtenida en cada test

	Grupo experimental	Grupo control
Media pre test	17,7	17,5
Media pos test	20,2	18,8
Diferencia	2,5	1,3

Para considerar la validez de estos datos, a continuación se muestran los valores obtenidos en el correspondiente análisis estadístico. Empleando el software estadístico PSPP, se ha analizado la diferencia de los datos obtenidos en el test inicial y final para poder obtener información respecto a la homogeneidad de varianzas (tabla 11) y su distribución normal (tabla 12).

Con estos datos, se acepta que la distribución de los grupos control y experimental se ajusta a un modelo normal y de que hay una semejanza entre las varianzas gracias al valor de significación obtenido en la prueba de Levene ($p = 0.072$), el cual es mayor que 0.05 (p valor de la significación estadística) (Montgomery, 1991). Igualmente, se acepta una distribución normal de las diferencias obtenidas entre ambos grupos gracias al valor obtenido en la prueba de Kolmogorov-Smirnov con una significancia asintótica de 0.026, el cual es menor a 0.05. Por lo tanto se puede aceptar una distribución normal de esta muestra al haber tomado un intervalo de confianza del 95%.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Tabla 11: Homogeneidad de grupos

	Estadístico de Levene	Df1	Df2	Significación
Valor	3.29	1	147	.072

Tabla 12: Distribución normal de los grupos

	Valor
N	149
Media	2.36
Desviación estándar	3.39
Diferencias más extremas-absoluto	.12
Diferencias más extremas-positivo	.12
Diferencias más extremas-negativo	-.08
Z de Kolmogorov-Smirnov	1.42
Sig. Asint. (2-colas)	.026

Por último, para concluir que estas diferencias son significativas conviene hacer una comparación con el análisis de la varianza ANOVA de un factor (Castañeda et al., 2010). A la hora de realizar la comparación entre grupos se obtiene un valor de significancia inferior a 0.05 (tabla 13), lo cual indica que la diferencia entre ambos grupos es estadísticamente significativa ya que se ha tomado un intervalo de confianza del 95% para descartar la hipótesis nula de que los grupos son iguales.

Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria.

Tabla 13: Análisis ANOVA

	Suma de cuadrados	Df	Cuadrado medio	F	Sign.
Entre grupos	176.34	1	176.34	16.97	.000
Intra grupos	1527.81	147	10.39		
Total	1704.15	148			

Como se ha citado anteriormente, en el proceso de trabajo, los datos cuantitativos y los resultados obtenidos se apoyan en unas evidencias cualitativas obtenidas con un cuestionario en el que los participantes han podido aportar sus impresiones sobre distintos aspectos relacionados con el desarrollo de la experiencia. Esta combinación es el pilar de un método mixto de investigación (Verd Pericás y Lozares Colina, 2014) y ayuda a entender y analizar una situación desde una perspectiva multidimensional con la necesaria fiabilidad y validez (Martínez et al., 2014). Por lo tanto, gracias a las respuestas recopiladas entre los participantes, podemos dar consistencia a los resultados positivos obtenidos en el proceso cuantitativo. En la gran mayoría de las preguntas se refleja una acogida muy buena de las prácticas realizadas por parte de los participantes, ya que lo ven como un recurso muy interesante para incluirlo de manera transversal en los últimos cursos de Educación Primaria. A continuación, podemos destacar algunos de los comentarios recogidos:

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

(Estudiante 27) *“Permitirá que los niños desarrollen estrategias de pensamiento crítico y resolución de problemas que puedan encontrarse a lo largo de su etapa escolar y su vida”*

(Estudiante 39) *“También se ve favorecido por el aumento de la creatividad, la comunicación, la participación y la motivación”.*

Además, esta visión es extensible a incluirlo como recurso para trabajar contenidos de Matemáticas, pudiendo destacar de entre las respuestas recogidas la siguiente:

(Estudiante 22) *“Con él podemos crear juegos para aprender Matemáticas, y así motivar al alumnado en esta asignatura, además, con los controles básicos de Scratch como avanzar hacia x número, los números negativos y positivos para los comandos, entre otros, ya estamos trabajando Matemáticas”.*

Se puede entender que los futuros docentes ven el potencial que tiene este recurso que ayuda a desarrollar habilidades de Pensamiento Computacional con el aprendizaje de las Matemáticas desde una perspectiva integrada. Un aspecto importante que ha ayudado a comprender el desarrollo de las prácticas realizadas ha sido contemplar si ha resultado fácil o difícil trabajar con esta herramienta. Se han encontrado con opiniones diversas como la siguiente:

Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria.

(Estudiante 6) “En un principio me resultó difícil manejar la aplicación porque no la conocía ni la había utilizado nunca, pero una vez que empiezas a crear proyectos, vas conociendo un poco más su funcionamiento y no es tan complicado”.

Lo cual nos da a entender que ven este tipo de recursos como algo importante en su formación docente.

(Estudiante 43) “Considero que es fundamental, al proporcionarnos una serie de conocimientos y recursos tecnológicos aptos para la creación de una serie de situaciones que permitan conectar con el alumnado y brindarle una educación adaptada al cambio”.

Por último, es importante saber si aprender y desarrollar este tipo de habilidades durante la formación inicial del profesorado tendrá un impacto en su trabajo futuro. La gran mayoría de los participantes considera que sí, que quiere utilizarlo en los cursos superiores de Educación Primaria cuando estén ejerciendo su labor docente.

(Estudiante 8) “Sin duda. De hecho, en mi opinión, es un recurso que puede ser utilizado en todos los cursos, aunque sea con supervisión de un docente o familiar cercano, debido a su alto porcentaje enriquecedor del aprendizaje”.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

(Estudiante 28) “Sí, pues podrán estar más cerca de las TICs, que están tan presentes en su día a día, y abordar cuestiones Matemáticas a través de estas”.

Discusión y conclusiones

Con todo lo expuesto en los apartados anteriores, queda de manifiesto el efecto positivo que tiene la inclusión de prácticas con Scratch para favorecer habilidades de Pensamiento Computacional en estudiantes de Grado de Educación Primaria. Pero además de ver este resultado que refiere al primer objetivo del presente estudio empírico, también está en consonancia la percepción que tienen los futuros docentes de este tipo de recursos educativos y la importancia de su inclusión transversal para el aprendizaje de las Matemáticas. Ver que desarrollan positivamente las competencias necesarias para implantar estas estrategias educativas a la vez que lo identifican como un factor importante para una educación completa e integrada en el contexto social para nuestro alumnado, da a entender que este tipo de contenidos son importantes para la formación inicial del profesorado.

Coincidiendo con otros estudios que indican que las prácticas tecnológicas para profesores en formación son beneficiosas para implantarlo posteriormente en el ámbito escolar (Vaca Cárdenas et al., 2016), estos resultados apuntan hacia una necesidad de incluir el desarrollo del

Capítulo 7: Introducción y desarrollo del Pensamiento Computacional en la formación inicial del profesorado de Educación Primaria.

Pensamiento Computacional y conocer recursos educativos que lo facilitan dentro de los planes de estudios de los futuros docentes en Educación Primaria. Este enfoque está en relación con las palabras de McComas (2013), quien apunta que es una necesidad imperiosa de nuestra sociedad preparar a los docentes para que desarrollen eficazmente las competencias y se aborden los contenidos, a través de un plan de estudios adecuado. Esta idea se completa con las conclusiones de Gabriele et al. (2019) que hacen esto extensible a la formación inicial del profesorado con el fin de prepararlos para desarrollar en nuestro alumnado las competencias y habilidades del S. XXI. Además, los participantes han destacado el aspecto beneficioso de este tipo de prácticas ya que favorecen el desarrollo e implementación de metodologías y estrategias de aprendizaje innovadoras, siendo la formación del profesorado un contexto muy importante para este punto (Duo-Terron, 2022; Moreno-Morilla et al., 2021).

Para futuras investigaciones, es interesante replicar este tipo de intervención con estudiantes del Grado de Educación Primaria pero ampliando el enfoque metodológico para analizar cómo pueden llevar a cabo la implementación en el aula de este tipo de recursos en el periodo de prácticas docentes que realizan. Esto puede ayudar a dar más solidez a las conclusiones obtenidas respecto a las habilidades desarrolladas en la formación inicial del profesorado sobre Pensamiento Computacional y la mejor forma de aplicarlo en el aula.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Además, esto establece un punto de partida para relacionar la formación inicial del profesorado con el desarrollo de su trabajo en el aula y la necesidad de una adecuada formación continua por parte de las distintas administraciones educativas.

IV. Parte final

Capítulo 8: Conclusiones

8.1 Contribuciones de la investigación

8.2 Limitaciones y propuestas de futuro

8.3 Futuras líneas de investigación

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Capítulo 8: Conclusiones

En esta última sección de la presente tesis doctoral, se exponen las diferentes discusiones y conclusiones relativas tanto al planteamiento general de este trabajo de investigación como a las conclusiones que se derivan de cada una de las experiencias empíricas incluidas en el mismo. Con esto, se establece el correspondiente aporte a la Sociedad del Conocimiento que seguro será de utilidad para otros docentes e investigadores afines al desarrollo de habilidades de Pensamiento Computacional en el aula y a la inclusión de este tipo de prácticas educativas en Educación Primaria y Secundaria.

Por otro lado, también se presentan en este capítulo las limitaciones propias que ha tenido el estudio y el proceso de investigación en su totalidad, así como las que se han encontrado en cada una de las experiencias. Con ello se pretende contribuir al planteamiento y determinación de futuras líneas de investigación que permitan seguir avanzando en este campo.

8.1 Aportaciones del marco teórico al proceso de investigación

Dada la trascendencia que ha tenido el marco teórico presentado de cara al planteamiento y desarrollo de las tres experiencias empíricas abordadas y, en definitiva en la propia tesis, pensamos que es pertinente resaltar algunos de los aspectos más importantes de dicha fundamentación teórica que han ayudado a

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

definir y relacionar los principales apartados de este trabajo, así como a identificar los hallazgos importantes en el campo objeto de nuestro estudio y, en consecuencia, a perfilar nuestros intereses y enfocar el trabajo.

El punto de partida en la estructura del marco teórico ha sido analizar el concepto de Pensamiento Computacional desde el trabajo de los investigadores y asociaciones más destacadas. De este modo, para dar una definición y contextualización de este término, hay que destacar las primeras aportaciones de Jannette Wing (2006) a las que siguieron las directrices dadas por la National Research Council (2010) y la la International Society for Technology in Education (ISTE, 2011). Estas contribuciones permitieron establecer un punto de partida muy importante para muchos trabajos en la literatura científica sobre este tema que, sin duda, han ayudado a desarrollar la idea, concepto e inclusión curricular del Pensamiento Computacional en los distintos niveles educativos. Pero como ocurre en el desarrollo del conocimiento, hemos visto que esta idea no surge en el momento en que cobra especial importancia. Los primeros trabajos que soportan las bases de este concepto provienen de las investigaciones de Seymour Papert (1972), quien a pesar de no hacer una definición formal del este término, constituyó las bases ideológicas para todo lo que se ha desarrollado hasta la fecha. Papert no solo vio la posibilidad de utilizar la programación como un recurso de aprendizaje, también fue pionero en el desarrollo de un lenguaje de programación que

permitiese a estudiantes de edades tempranas introducirse en este ámbito y al que iba unido un enfoque filosófico de la educación: la teoría del construccionismo (Papert, 1999). Como se ha citado anteriormente, de aquí han surgido muchas investigaciones, definiciones y marcos conceptuales que han intentado dar forma al concepto de Pensamiento Computacional con el objetivo de unificar el desarrollo e inclusión en las prácticas educativas. De entre todos los marcos conceptuales que encontramos en la literatura científica, destacan el trabajo de Brennan-Resnick (2012) y el aportado por Suchi Grover y Roy Pea (2018). Ambos coinciden en identificar el Pensamiento Computacional como un elemento integrador en el ámbito STEM, incorporando en este marco conceptual elementos o conceptos y prácticas computacionales; i bien, el marco de Brennan-Resnick añade las denominadas perspectivas computacionales, las cuales intentan promover en el estudiante la comprensión de su propio proceso de aprendizaje, incluyendo la expresión e interacción con otros estudiantes.

Establecer un marco conceptual ha facilitado la aparición de muchas definiciones de Pensamiento Computacional en los últimos años, pero de entre todas las que se han podido incluir en el marco teórico de este trabajo, quizá convenga destacar la aportada por Moreno-León y colaboradores (Moreno-León et al., 2019) , ya que reúne muchos de los elementos característicos de las aportaciones de otros investigadores

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

“El PC sería la capacidad de formular y representar problemas para resolverlos haciendo uso de herramientas, conceptos y prácticas de la disciplina informática, como la abstracción, la descomposición o el uso de simulaciones” (p. 33).

El hecho de que se hayan realizado tantas contribuciones en este tema durante los últimos años, tanto para llegar a una definición como para establecer un marco de referencia, ha facilitado que muchos gobiernos europeos incluyan de manera oficial el Pensamiento Computacional en la legislación educativa. Esto permite que la inclusión curricular en el trabajo diario con el alumnado sea más efectiva, ya que parte de una base formal y facilita que se haga de manera transversal e integrada, no solo en asignaturas de ámbito Tecnológico como se podía estar haciendo hasta hace poco. Esto hace posible que las habilidades básicas de Pensamiento Computacional, consideradas destrezas esenciales para cualquier ciudadano (Wing, 2008), las pueda desarrollar cualquier estudiante. Además, esto no es algo que se refiera simplemente a destrezas, habilidades o competencias académicas. Está en relación con la denominada alfabetización digital, la cual es esencial en nuestra vida diaria y se ve consolidada con el desarrollo del Pensamiento Computacional (Mantilla y Negre, 2021).

Pero para que este planteamiento se pueda llevar al aula de manera efectiva por cualquier docente de cualquier ámbito, es necesario disponer de la tecnología adecuada. Cuando Seymour Papert (1980) desarrolló el que se puede considerar como el primer lenguaje de programación para escolares, LOGO, se postuló una visión utópica en ese momento para integrar la programación de un ordenador en el aprendizaje. En ese momento, no se disponía de un acceso universal a la tecnología ni las características de ese lenguaje de programación eran las que presentan actualmente este tipo de recursos educativos. Fue un momento muy importante para la innovación educativa y el desarrollo de la tecnología pero, por ejemplo, difícil de incluir en la enseñanza de la Biología. Si embargo, hoy día nos encontramos con una amplia gama de lenguajes de programación, como Scratch, que permiten implementar actividades que involucren destrezas y habilidades de Pensamiento Computacional en cualquier asignatura de manera fácil y atractiva (Moreno-León et al., 2017) para dar un importante paso de “aprender a programar” a “programar para aprender” (Resnick, 2013).

8.2 Contribuciones de la investigación

Una vez completado este análisis de los puntos más importantes del marco teórico, estamos en condiciones de destacar las conclusiones más importantes en relación con el objetivo principal de esta tesis doctoral, que es analizar el impacto que puede tener en los procesos de aprendizaje la inclusión de

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

prácticas asociadas al Pensamiento Computacional utilizando el lenguaje de programación Scratch. Concretamente, se ha trabajado en el aprendizaje de las Matemáticas en Educación Primaria, en Educación Secundaria y en el proceso de formación inicial para el profesorado de Educación Primaria.

En la tabla 14 se muestra la conexión de los objetivos específicos con los estudios realizados. A continuación describimos las principales conclusiones asociadas a cada uno de los objetivos de investigación.

Tabla 14: Relación ente hipótesis y objetivos con los estudios realizados

Hipótesis de la tesis	Objetivos de la tesis	Estudios		
		1	2	3
H1. La inclusión curricular del Pensamiento Computacional favorece la capacidad de de resolución de problemas en el aprendizaje de las Matemáticas	OE1. Estudiar la capacidad de análisis y la comprensión de los enunciados de problemas para la elaboración de estrategias de resolución empleando prácticas de Pensamiento Computacional con Scratch	x		
	OE2. Analizar la influencia de las prácticas de Pensamiento Computacional en el abordaje de los procesos que se siguen en la resolución de problemas	x		
H2. El desarrollo de actividades relacionadas con el Pensamiento Computacional en el área de las Matemáticas favorece el desarrollo de los	OE3. Estudiar la motivación del alumnado realizando actividades de Pensamiento Computacional con Scratch		x	
	OE4. Potenciar el uso de nuevas metodologías y estrategias de aprendizaje a través de la inclusión curricular del software de programación visual Scratch		x	

<p>contenidos curriculares y mejora el proceso de aprendizaje</p>	<p>OE5. Evaluar el desarrollo de la competencia y el razonamiento matemático en Geometría trabajando actividades de Pensamiento Computacional como recurso transversal de aprendizaje con alumnos de 5° curso de Educación Primaria</p>		<p>x</p>	
<p>H3. Trabajar actividades de Pensamiento Computacional con Scratch en la formación inicial del profesorado de Educación Primaria mejora sus destrezas y habilidades en este ámbito</p>	<p>OE6. Analizar el grado de desarrollo de diferentes competencias computacionales realizando actividades con el lenguaje de programación Scratch</p>			<p>x</p>
	<p>OE7. Valorar la percepción de este tipo de actividades y recursos educativos en la formación inicial de futuros docentes de Educación Primaria</p>			<p>x</p>
	<p>OE8. Potenciar el uso de nuevas metodologías y estrategias de aprendizaje trabajando el Pensamiento Computacional de manera transversal en la enseñanza de las Matemáticas</p>			<p>x</p>

8.2.1 Objetivo específico 1

OE1. Estudiar la capacidad de análisis y la comprensión de los enunciados de problemas para la elaboración de estrategias de resolución empleando prácticas de Pensamiento Computacional con Scratch

Este objetivo planteado en el primer estudio se toma como un punto de partida para los distintos procesos de análisis expuestos, ya que se centra en uno de los elementos imprescindibles para el aprendizaje de las Matemáticas: la resolución de problemas. Concretamente, el análisis y la comprensión de un enunciado es fundamental para comenzar y completar cualquier razonamiento lógico que se plantee ante un problema de cualquier tipo. Poder desarrollar esta capacidad en un contexto de prácticas computacionales hace que la viabilidad para incluir estos elementos transversales en el aula de Matemáticas cobre especial interés y adquiera más relevancia en el proceso de aprendizaje de todo el alumnado, ya que todas las habilidades del Pensamiento Computacional están estrechamente ligadas con los procesos de representación, formulación y resolución de problemas (Wing, 2014) y se sustenta igualmente en la intersección existente entre el Pensamiento Computacional y el razonamiento matemático (Sneider et al., 2014). Esto se analizó en el primer estudio de los tres que componen este proceso de investigación, obteniendo un resultado general de que tras la implementación

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

práctica un 70% de los participantes habían mejorado el proceso general de resolución de un problema trabajando las destrezas asociadas a través de prácticas computacionales con Scratch. Pero para concretar más en este primer objetivo, el análisis se hizo diferenciando las distintas partes en las que se puede dividir el proceso de resolución de un problema, observando que había una significativa mejora en la comprensión del enunciado, concretamente un 66'7 % de los participantes, mejora que también se traduce en una adecuada elaboración de una estrategia de resolución según el análisis cualitativo realizado, pudiéndose establecer este primer objetivo como alcanzado.

8.2.2 Objetivo específico 2

OE2. Analizar la influencia de las prácticas de Pensamiento Computacional en el abordaje de los procesos que se siguen en la resolución de problemas

Alcanzar este objetivo está estrechamente ligado al desarrollo y consecución del anterior, ya que complementa el análisis y la comprensión de un enunciado con el resto de procesos que pueden involucrarse en la resolución de un problema. Así, plantear estas dos metas en el primer estudio empírico ha permitido observar la influencia de la inclusión de prácticas computacionales en todo el proceso de resolución de problemas que puede realizar cualquier estudiante. Aunque esto se ha analizado sobre una colección de problemas aritméticos, estas habilidades se pueden extrapolar a problemas de cualquier

ámbito ya que se han trabajado los procesos que ayudan a entender y organizar la información planteada para idear y desarrollar una estrategia de resolución, un proceso que tiene mucho en común con la forma en la que un informático puede realizar cualquier proceso de programación para resolver un problema, aspecto que se presenta en la base de cualquier definición o aproximación al concepto de Pensamiento Computacional. A la hora de analizar el grado de consecución de este objetivo, se dio una situación similar a la del anterior; así en el proceso de reflexión y valoración cualitativa del trabajo realizado se apreció que más de un 60% aseguró que la planificación inicial incidió positivamente de cara al abordaje del problema. En definitiva, gracias al análisis combinado de los datos cuantitativos y cualitativos se pudo dar este objetivo como alcanzado y cerrar este análisis sobre el proceso de resolución de un problema al completo ya que en el proceso de reflexión y de intercambio de información con los participantes, explicaban cómo había llevado a la práctica este proceso. Desde una perspectiva más amplia, considerando todas las partes en las que se desarrolla el proceso de resolución de un problema y que se han analizado en nuestro estudio, se vio que un 61.1% del alumnado ha obtenido una mejoría en dos o más de estas partes. Por lo que se pueden establecer estos objetivos igualmente como alcanzados. Esta consecución ha permitido poder dar validez a la primera hipótesis de la investigación:

H1. La inclusión curricular del Pensamiento Computacional favorece la capacidad de de resolución de problemas en el aprendizaje de las Matemáticas

El análisis y la observación de cómo las prácticas computacionales favorecen la capacidad de resolución de problemas se ha basado en la implementación del método de Polya (1965) en el caso concreto en el que nos hemos centrado ; sin embargo, cabe pensar que dicha influencia favorable se puede extrapolar a otras situaciones problemáticas que se pueden plantear en los procesos de enseñanza y aprendizaje de las Matemáticas.

8.2.3 Objetivo específico 3

OE3. Estudiar la motivación del alumnado realizando actividades de Pensamiento Computacional con Scratch.

Este objetivo planteado en el segundo estudio se refiere a una cuestión muy importante y que siempre ha de estar presente en la inclusión de cualquier recurso educativo: la motivación que puede despertar en el alumnado. En algunas ocasiones, el desarrollo de prácticas computacionales o el uso de recursos educativos del ámbito tecnológico puede resultar tedioso o repetitivo para el alumnado, por lo que hay que prestar especial atención e intentar diseñar la inclusión de estos recursos y el desarrollo de las prácticas asociadas de manera que despierte motivación en el alumnado mejorando así su propio

proceso de aprendizaje. El planteamiento que se ha dado al uso de Scratch en este estudio para incluirlo en el proceso de aprendizaje de las Matemáticas ha ayudado a motivar al alumnado y a despertar el interés para trabajar en el aula de manera diferente y atractiva, lo cual ayuda a postular que las actividades de Pensamiento Computacional favorecen los procesos de enseñanza y aprendizaje de las Matemáticas de manera que los hace más atractivos para el alumnado. Este objetivo es difícilmente observable con un proceso cuantitativo, pero su consecución quedó reflejada en el análisis cualitativo realizado durante la correspondiente experiencia y en el que se recogieron ideas e impresiones de los participantes, tanto docentes como estudiantes, a la hora de realizar las prácticas propuestas. De las impresiones del profesorado participante se puede destacar que tras la formación recibida y el correspondiente desarrollo de la experiencia lo ven como un recurso educativo fácil de incluir en su planificación y que ayuda a desarrollar actividades de carácter más competencial. Entre los comentarios del alumnado, dado que este recurso no se usa como una herramienta digital que solo ofrece actividades interactivas, destaca las aportaciones hechas valorando que les permite hacer sus propias construcciones de manera más personal y creativa, lo cual facilitó un planteamiento de un trabajo diferente al convencional. El alumnado lo ve como un recurso más divertido, pero no solo eso. También expresa que con el

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

uso de esta herramienta el aprendizaje de las Matemáticas se hace de manera más divertida.

8.2.4 Objetivo específico 4

OE4. Potenciar el uso de nuevas metodologías y estrategias de aprendizaje a través de la inclusión curricular del software de programación visual Scratch

El uso de las nuevas tecnologías en los procesos de aprendizaje muchas veces se realiza para sustituir tareas que perfectamente pueden completarse de manera tradicional, por lo que puede suponer una distracción en el proceso cognitivo que esté realizando un estudiante en esa tarea concreta. Por este motivo, es conveniente analizar la inclusión de los recursos tecnológicos para que estos estén verdaderamente enfocados al desarrollo de habilidades específicas que puedan mejorar los procesos de aprendizaje en los que se incluyen. Es lo que se pretende con este objetivo específico planteado para su tratamiento fundamental en el segundo estudio. El desarrollo de las habilidades de Pensamiento Computacional en muchas ocasiones está ligado al uso de recursos tecnológicos, por lo que es importante hacer una correcta inclusión curricular de todo esto para promover su buen uso para que formen parte de un aprendizaje activo en nuestro alumnado y que no solo sirva para mostrar o dispensar contenidos; además, el uso de estos recursos y su inclusión en el currículo educativo hace que nuestro alumnado pase de ser un

consumidor pasivo de la tecnología, como ocurre cuando se usa solo para presentar contenidos, a ser un agente activo que puede crear, reinventar, remezclar y compartir elaboraciones propias construidas durante el proceso de aprendizaje, en este caso, de las Matemáticas. La consecución de este objetivo se puede ver plasmada en cómo el profesorado participante puede valorar el desarrollo de esta experiencia y la valoración de su inclusión curricular. De manera similar a la descrita para el objetivo anterior, con la información cualitativa recogida en el proceso de trabajo, cabe destacar que esta inclusión se ha valorado muy positivamente. De los comentarios recogidos en el proceso cualitativo se puede destacar que se ha percibido como “un recurso fácil de manejar, que no requiere de mucho tiempo para aprender a implementarlo en el aula y con el que se potencia un cambio metodológico real en comparación con otros recursos tecnológicos”.

El haber podido alcanzar los objetivos de la investigación referentes al análisis de la motivación del alumnado y al efecto positivo que tiene incluir recursos tecnológicos basados en conceptos de programación y habilidades de Pensamiento Computacional para trabajar con las Matemáticas, va en consonancia con una necesidad que se encuentra en la bibliografía científica: la importancia de realizar investigaciones sobre la integración del Pensamiento Computacional en la enseñanza de las Matemáticas en Educación Primaria (Leung, 2021). En consonancia con los trabajos de

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Rodríguez-Martínez et al. (2020), también hay que destacar la necesidad de realizar este tipo de prácticas empíricas en una línea de investigación como la que se ha tomado para los trabajos presentados en esta tesis: con un enfoque interdisciplinar que permita abordar simultáneamente conceptos computacionales y matemáticos.

8.2.5 Objetivo específico 5

OE5. Evaluar el desarrollo de la competencia y el razonamiento matemático en Geometría trabajando actividades de Pensamiento Computacional como recurso transversal de aprendizaje con alumnos de 5º curso de Educación Primaria

Además de lo mencionado anteriormente, y para concretar un poco más la validez de la hipótesis aceptada en el primer estudio, completando el planteamiento establecido con los dos objetivos específicos anteriores se ha analizado cómo se han desarrollado habilidades Matemáticas asociadas al razonamiento geométrico tras abordar en clase estos contenidos a través de prácticas computacionales con Scratch. Con este enfoque, se puede entender que este objetivo se establece para hacer más extensible la validez de la primera hipótesis con la que se acepta que las prácticas computacionales favorecen procesos generales en el aprendizaje de las Matemáticas como es la resolución de problemas. Se hace una extensión que permite focalizar en un

punto más concreto y analizar el beneficio que hay en el proceso de aprendizaje de un bloque en particular como es el de la Geometría y de los saberes que pueden asociarse a este. Esto quedó reflejado en los resultados obtenidos en el análisis cuantitativo realizado con los participantes. El alumnado de los grupos experimentales obtuvo de media una diferencia entre el test final y el inicial de más de 25 puntos frente a un promedio de 15 puntos que obtuvieron los grupos de control. Una diferencia que quedó respaldada por la valoración cualitativa del alumnado participante al valorar el recurso utilizado muy positivamente y concluir en términos generales que las prácticas han facilitado un aprendizaje más activo. Esta relación entre la consecución de estos tres objetivos específicos a través de una combinación del análisis cualitativo y cuantitativo, permiten definir el escenario adecuado para realizar la experiencia empírica que ha permitido dar validez a la segunda hipótesis plantada en el estudio:

H2. El desarrollo de actividades relacionadas con el Pensamiento Computacional en el área de las Matemáticas favorece el desarrollo de los contenidos curriculares y mejora el proceso de aprendizaje

Una afirmación que se caracteriza por tener su centro en un punto más concreto que el proceso de resolución de problemas y que igualmente requiere que el alumnado utilice los recursos adecuados con una motivación tal que le

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

ayude a llevar a cabo el proceso de aprendizaje de manera efectiva. De esta forma, en el estudio 2 se vio que la inclusión de los recursos tecnológicos se presenta como una herramienta útil y que ayuda a plantear nuevas e innovadoras situaciones de aprendizaje con las que todo el alumnado pueda sentirse como el principal agente en torno al cual gira el proceso de enseñanza y aprendizaje. Esto hace posible una inclusión curricular que favorezca el trabajo y el aprendizaje propio de un bloque específico de las Matemáticas, en este caso el de Geometría, a través de prácticas computacionales ofreciendo un escenario más adecuado, motivador e innovador y que ayude al desarrollo competencial de todo el alumnado. Como se ha mencionado anteriormente, esto ha quedado de manifiesto y respaldado no solo por los datos que se han podido obtener de las pruebas cuantitativas. De las reflexiones e intervenciones recogidas del alumnado y profesorado participante en la experiencia, se establecen unas conclusiones que avalan esta afirmación destacando la importancia de incluir estos recursos en el proceso de enseñanza y aprendizaje de las Matemáticas para definir un escenario más competencial.

8.2.6 Objetivo específico 6

OE6. Analizar el grado de desarrollo de diferentes competencias computacionales realizando actividades con el lenguaje de programación Scratch en la formación inicial del Profesorado de Educación Primaria.

Para que los planteamientos establecidos en los anteriores objetivos específicos y en las correspondientes hipótesis tengan una aplicación real y efectiva en el aula, es imprescindible contar con una adecuada formación del profesorado. Si los y las docentes no tienen unas habilidades computacionales adecuadas, no sería posible diseñar experiencias de aprendizaje ni desarrollar destrezas de Pensamiento Computacional con un grupo de estudiantes. Por lo tanto, es conveniente introducir este tipo de prácticas y desarrollar el Pensamiento Computacional en la formación del profesorado, siendo la formación inicial el punto de partida para que la alfabetización en este campo sea la adecuada. La consecución de este objetivo se ha visto plasmada en la comparación de los resultados obtenidos con los grupos experimentales y de control tras realizar el Test de Pensamiento Computacional. La diferencia en el grupo experimental casi duplica a la obtenida en el grupo de control, lo cual refleja el efecto y la importancia que tiene la realización de unas prácticas adecuadas para que el profesorado adquiera las competencias computacionales necesarias.

8.2.7 Objetivo específico 7

OE7. Valorar la percepción de este tipo de actividades y recursos educativos en la formación inicial de futuros docentes de Educación Primaria

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Igualmente es importante que los y las futuras docentes vean este tipo de recursos y habilidades de Pensamiento Computacional como algo necesario para la formación de cualquier estudiante del Siglo XXI, así como para su competencia y alfabetización digital como ciudadanos activos. Si el profesorado tiene esta percepción, verá necesario continuar su formación en este ámbito, aprender a diseñar situaciones de aprendizaje adecuadas a distintos contextos, niveles y ámbitos y en definitiva podrá llevar a cabo una inclusión del Pensamiento Computacional de manera transversal e integrada en el currículo. Para alcanzar este objetivo y completar parte del tercer estudio realizado durante este trabajo de investigación, se realizó un análisis cualitativo de las impresiones y reflexiones aportadas por los participantes del grupo experimental sobre cómo había sido el desarrollo de las prácticas y la viabilidad o utilidad que podían tener este tipo de recursos tanto en su formación como en el trabajo del aula. Una valoración y enfoque que se manifestó como muy positivo tal y como se recogen en las conclusiones del tercer estudio realizado y de las que se puede destacar que lo ven como un recurso práctico, creativo y fundamental para darle al alumnado nuevos conocimientos.

8.2.8 Objetivo específico 8

OE8. Potenciar el uso de nuevas metodologías y estrategias de aprendizaje trabajando el Pensamiento Computacional de manera transversal en la enseñanza de las Matemáticas

Así, con el planteamiento y consecución de estos dos objetivos específicos se puede orientar en una dirección correcta el uso de recursos de Pensamiento Computacional y acompañarlo en un contexto metodológico dinámico y que ofrezca un papel activo para el alumnado. El profesorado no solo debe saber usar herramientas y recursos tecnológicos. También está obligado a incluirlos en su práctica diaria de manera efectiva, con una metodología adecuada y que ayude a trabajar con el alumnado de manera reflexiva, participativa, creativa y adaptándose a las necesidades de todo los y las estudiantes para poder desarrollar, en este caso, habilidades de Pensamiento Computacional de manera integrada en el aprendizaje de las Matemáticas. Este objetivo se ve alcanzado, y complementa a la consecución del objetivo anterior, a través del análisis cualitativo que se llevó a cabo. De manera específica para este objetivo y planteando una extensión a la consecución de objetivos anteriores, se puede destacar que los y las futuras docentes ven el uso de este recurso educativo como una posibilidad de abordar temas de Matemáticas, como números enteros o direcciones, de una manera diferente a la convencional, con

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

un enfoque metodológico distinto y favoreciendo la comunicación y la participación en clase.

Con el escenario dibujado por estos tres objetivos específicos, se da la posibilidad de validar la tercera y última hipótesis de este trabajo de investigación referente a cómo los y las futuras docentes adquieren las habilidades y competencias necesarias para llevar a cabo una inclusión curricular, en este caso en el contexto del aprendizaje de las Matemáticas, del Pensamiento Computacional y sus prácticas asociadas:

H3. Trabajar actividades de Pensamiento Computacional con Scratch en la formación inicial del profesorado de Educación Primaria mejora sus destrezas y habilidades en este ámbito

La inclusión de este trabajo en su formación inicial ha permitido que desarrollen positivamente las habilidades asociadas al razonamiento lógico y al uso de distintas estructuras y conceptos computacionales como la descomposición, el reconocimiento de patrones, la abstracción o el diseño e implementación de algoritmos sencillos. Esta hipótesis sustenta su veracidad en la metodología mixta que ha permitido dar validez a los objetivos específicos de este estudio, ya que se ha comprobado que la mejora significativa en unas destrezas específicas está asociada a una necesaria y muy importante percepción del recurso educativo como elemento que facilita un

cambio metodológico hacia una práctica educativa más abierta, activa y competencial.

Esto ayuda a que entiendan que la importancia de desarrollar estas destrezas no es para hacer que el alumnado aprenda a programar en un futuro. Lo importante es que el profesorado tenga los conocimientos y habilidades necesarias para facilitar la alfabetización digital de cualquier estudiante desde una perspectiva global e integrada con los necesarios conocimientos de tecnología (Polly, 2015). Además, al igual que destacan autores como Gabriele et al. (2019), con este tipo de experiencias se intenta mitigar una barrera presente en la formación inicial del profesorado en cuanto al uso de recursos tecnológicos y su aplicación en el aula, lo cual es muy importante para obtener unos resultados adecuados en la formación de cualquier estudiante de Educación Primaria.

8.3 Limitaciones y futuras líneas de investigación

El desarrollo de este trabajo de investigación ha estado limitado, en cierto modo, por los condicionamientos propios de cada uno de los estudios empíricos que se han realizado. Por un lado, y esto atañe a los tres estudios, habría que tener en consideración el tamaño de las muestras tomadas para cada uno, ya que aunque habría sido interesante ampliar los estudios a más

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

centros, ello no ha sido posible dadas las limitaciones lógicas en recursos humanos y en tiempo.

En el caso del primer estudio, la limitación principal de la muestra ha sido no poder realizar una comparativa con un grupo de control. Esto está relacionado directamente con la limitación del tamaño de la muestra con la que hemos podido trabajar. Por lo tanto, una futura línea de investigación desde este estudio podría ser la realización de un análisis de cómo desarrollar el Pensamiento Computacional trabajando con Scratch mejora la estrategia de resolución de problemas en Matemáticas con un grupo control y otro experimental. Además, sería necesario analizar este aspecto con grupos de estudiantes de varios niveles educativos y no de uno solo como se pudo abordar en esta experiencia empírica.

En el segundo estudio, la limitación principal surgió del tiempo del que se dispuso para trabajar con los grupos de estudiantes. Esto afectó directamente al tamaño de la muestra inicial, ya que al no poder estar todas las horas previstas con los grupos se tuvo que delegar el trabajo en los profesores responsables de cada centro. Los profesores recibieron la formación adecuada y en este sentido no hubo ningún problema, pero tuvimos que descartar el trabajo en algún centro escolar porque no se completaron todas las prácticas requeridas con Scratch. Este aspecto se presenta ahora en una situación mucho

más favorable. Al estar el Pensamiento Computacional contemplado en la legislación educativa de manera explícita, se pueden plantear prácticas en este tema con mayor facilidad. Por lo que, además de replicar un estudio similar con una muestra mayor, una posible línea de investigación futura que derive de este estudio es analizar la adquisición de contenidos de Matemáticas a través de prácticas computacionales con Scratch abordando diferentes bloques de contenidos.

Contemplando los dos primeros estudios, una posible línea de investigación futura complementaria a ambas situaciones es analizar el desarrollo de habilidades de Pensamiento Computacional a la vez que se analizan distintas destrezas como la resolución de problemas o la forma en la que se trabajan los distintos bloques contenidos o saberes básicos.

Igualmente, y dado el carácter transversal e integrador con el que se contextualiza el Pensamiento Computacional en la educación formal, otra posible línea de investigación futura se podría centrar en analizar el desarrollo de habilidades computacionales con Scratch u otros recursos educativos, realizando prácticas dentro de una asignatura como Matemáticas y en una asignatura que no sea específica a unos contenidos. Esto puede afectar a la forma en la que los estudiantes presentan más o menos motivación por trabajar con estos recursos y puede ofrecer resultados más fiables en cuanto a

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

la aplicación de ciertas habilidades y destrezas de la resolución de problemas al trabajarse fuera de un contexto tradicional tal y como se hace en clase de Matemáticas.

Derivado del tercer estudio, se puede destacar una limitación muy importante dentro de la formación que reciben los estudiantes de Grado en Educación Primaria: la aplicación en el aula con estudiantes. Los participantes que han trabajado en esta experiencia han presentado unos resultados positivos en cuanto a la adquisición de habilidades computacionales, al conocimiento de un recurso educativo como Scratch y al interés en utilizarlo con sus futuros estudiantes. Igualmente, han tomado conciencia de la importancia de trabajar con este tipo de recursos en el aula para el desarrollo de las competencias de manera integral y completa para el alumnado de Educación Primaria. Pero sería interesante conocer cómo aplican posteriormente este trabajo en el aula, por ejemplo, durante el periodo de prácticas docentes, y completar el estudio con una propuesta de clase real en la que se analice el impacto sobre el aprendizaje o destrezas de Pensamiento Computacional o su adecuación al trabajo en el aula y el interés y motivación que los estudiantes de Primaria . Este análisis nos ayudaría a conocer si se está dando un enfoque realmente práctico y con aplicación real en la formación inicial del profesorado y evitar que solo nos quedemos en una formación teórica de lo que pueden o deben trabajar en el aula. Otra posible línea de investigación futura resultante de este

estudio es hacer este análisis en la formación inicial de docentes de otro nivel educativo. Por ejemplo, para el profesorado de Educación Secundaria. Independientemente de la especialidad en la que se formen estos futuros docentes, dentro del ámbito de Ciencia y Tecnología todos los estudiantes deberían tener unas nociones básicas de qué es el Pensamiento Computacional y cómo desarrollarlo a través de prácticas computacionales de manera transversal en cualquier asignatura de este ámbito. Igualmente, tal y como se ha dicho en el párrafo anterior, el impacto de incluir estas prácticas en la formación inicial debería ir ligado a la posibilidad de ponerlo en práctica una vez completado este proceso de formación inicial.

Referencias bibliográficas

- Abelson, H., & Sussman, G. J. (1996). *Structure and Interpretation of Computer Programs*. The MIT Press. <https://library.oapen.org/viewer/web/viewer.html?file=/bitstream/handle/20.500.12657/26092/sicp.pdf?sequence=1&isAllowed=y>
- Abiteboul, S., Archambaul, J., Balagué, C., Baron, G., Berry, G., Doweiket, G., de la Higuera, C., Nivat, N., Tort, F., & Viéville, T. (2013). *Rapport de l'académiedes sciences. L'enseignement de l'informatique en France. Il est urgent de ne plus attendre*. Institut de France. Académies des sciences. https://www.academie-sciences.fr/pdf/rapport/rads_0513.pdf
- Acuña-Medina, N., León-Arias, M., López-Palomino, L., Villar-Navarro, C., & Mulford-León, R. (2018). Aprendizajes de las Matemáticas Mediados Por Juegos Interactivos En Scratch En La IEDGVCS. *Cultura Educación y Sociedad*, 9(2), 32–42. <https://doi.org/10.17981/cultedusoc.9.2.2018.03>
- Adell Segura, J., Llopis Nebot, M. Á., Esteve Mon, F., & Valdeolivas Novella, M. G. (2019). El debate sobre el pensamiento computacional en educación. *RIED. Revista Iberoamericana de Educación a Distancia*, 22(1), 171. <https://doi.org/10.5944/ried.22.1.22303>
- Nistor, A., Gras-Velazquez, A., Billon, N. & Mihai, G. (2018). *Science, Technology, Engineering and Mathematics Education Practices in Europe. Scientix Observatory report*. December 2018, European Schoolnet. http://www.scientix.eu/documents/10137/782005/STEM-Edu-Practices_DEF_WEB.pdf/b4847c2d-2fa8-438c-b080-3793fe26d0c8
- Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, 105, 106185. <https://doi.org/10.1016/j.chb.2019.106185>
- Ateşkan, A., & Hart, D. O. (2021). Demystifying computational thinking for teacher candidates: A case study on Turkish secondary school pre-service

- teachers. *Education and Information Technologies*, 26(5), 6383–6399. <https://doi.org/10.1007/s10639-021-10626-9>
- Atmatzidou, S., & Demetriadis, S. N. (2017). A didactical model for educational robotics activities: a study on improving skills through strong or minimal guidance. In A. Dimitris Alimisis, Michele Moro & E. Menegatti (Eds.), *Educational robotics in the makers era* (Vol. 560, pp. 58–72). Springer. <https://doi.org/10.1007/978-3-319-55553-9>
- Balanskat, A., & Engelhardt, K. (2015). Computing our future. In *Computer programing and coding. Priorities, school curricula and initiatives across Europe*. European Schoolnet.
- Barber, M., & Mourshed, M. (2007). *How the world's best-performing school systems come out on top*. McKinsey & Company.
- Barr, B. V., & Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads*, 2(1), 48–54. <https://doi.org/10.1145/1929887.1929905>
- Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. *Research and Practice in Technology Enhanced Learning*, 11(13). <https://doi.org/10.1186/s41039-016-0036-2>
- Becker, K., & Park, K. (2011). Effects of integrative approaches among science, technology, engineering, and mathematics (STEM) subjects on students' learning: A preliminary. *Journal of STEM Education*, 12(5), 23–38.
- Bender, W., Urrea, C., & Zapata-Ros, M. (2015). Presentación del número especial sobre Pensamiento Computacional. *Revista de Educación a Distancia (RED)*, 46(46), 1–3. <https://doi.org/10.6018/red/46/1>
- Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers and Education*, 58(3), 978–988. <https://doi.org/10.1016/j.compedu.2011.10.006>
- Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging Primary Programming and Mathematics: Some Findings of Design Research in

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

- England. *Digital Experiences in Mathematics Education*, 3(2), 115–138. <https://doi.org/10.1007/s40751-017-0028-x>
- Berry, M. (2013). *Computing in the national curriculum A guide for primary teachers*. Computing at School. <http://www.computingatschool.org.uk/data/uploads/CASPrimaryComputing.pdf>
- Bers, M. U. (2018). Coding, Playgrounds and Literacy in Early Childhood Education: the Development of KIBO Robotics and ScratchJr. *2018 IEEE Global Engineering Education Conference (EDUCON)*, 2100–2108.
- Bers, M. U. (2019). Coding as another language: a pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, 6(4), 499–528. <https://doi.org/10.1007/s40692-019-00147-3>
- Bers, M. U., & Sullivan, A. (2019). Computer science education in early childhood: The case of scratchjr. *Journal of Information Technology Education: Innovations in Practice*, 18, 113–138. <https://doi.org/10.28945/443710.28945/4437>
- Bocconi, S., Chiocciariello, A., Kampylis, P., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M., Malagoli, C., Cachia, R., Giannoutsou, N., & Punie, Y. (2022). *Reviewing Computational Thinking in Compulsory Education. State of play and practices from computing education*. Publications Office of the European Union, Luxembourg, 2022. <https://doi.org/10.2760/126955>
- Boletín Oficial de Castilla y León. (2016). Orden EDU/589/2016, de 22 de junio, por la que se regula la oferta de materias del bloque de asignaturas de libre configuración autonómica en tercer y cuarto curso de Educación Secundaria Obligatoria. *Boletín Oficial de Castilla y León*, 122, 29175–29343. <http://bocyl.jcyl.es/boletines/2010/06/23/pdf/BOCYL-D-23062010-1.pdf>
- Boletín Oficial de la Comunidad de Madrid. (2014). Decreto 89/2014, de 24 de julio, del Consejo de Gobierno, por el que se establece para la Comunidad de Madrid el Currículo de la Educación Primaria. *Boletín*

- Oficial de La Comunidad de Madrid*, 175, 10–89.
<https://doi.org/10.2307/j.ctv153k4jj.4>
- Boletín Oficial de la Comunidad de Madrid. (2015). Decreto 48/2015, de 14 de mayo, del Consejo de Gobierno, por el que se establece para la Comunidad de Madrid el currículo de la Educación Secundaria Obligatoria. *Boletín Oficial de La Comunidad de Madrid*, 118, 10–319.
- Boletín oficial de la Junta de Andalucía. (2016). Orden de 14 de julio de 2016, por la que se desarrolla el currículo correspondiente a la Educación Secundaria Obligatoria en la Comunidad Autónoma de Andalucía. *Boletín Oficial de La Junta de Andalucía (B.O.J.A.)*, 108–396.
http://juntadeandalucia.es/eboja/2016/144/BOJA16-144-00289-13500-01_00095875.pdf
- Boletín Oficial de la Junta de Andalucía. (2021). Orden de 15 de enero de 2021, por la que se desarrolla el currículo correspondiente a la etapa de Educación Secundaria Obligatoria en la Comunidad Autónoma de Andalucía. *Boletín Oficial de La Junta de Andalucía*, 7, 656–793.
- Boletín Oficial de la Región de Murcia. (2015). Decreto 220/2015, de 2 de septiembre de 2015, por el que se establece el currículo de la Educación Secundaria Obligatoria en la Comunidad Autónoma de la Región de Murcia. *Boletín Oficial de La Región de Murcia*, 203.
- Boletín oficial de La Rioja. (2015). Decreto 19/2015, de 12 de junio, por el que se establece el currículo de la Educación Secundaria Obligatoria y se regulan determinados aspectos sobre su organización así como la evaluación, promoción y titulación del alumnado de la Comunidad Autónoma de L. *Boletín Oficial de La Rioja*, 79, 12368–12730.
http://ias1.larioja.org/boletin/Bor_Boletin_visor_Servlet?referencia=2386883-1-PDF-493946
- Boletín Oficial de las Illes Balears. (2015). Decreto 35/2015, de 15 de mayo, por el que se establece el currículo del Bachillerato en las Islas Baleares. *Boletín Oficial Las Illes Balears*, 2005, 25225–25264.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

- Boletín Oficial de Navarra. (2014). Decreto Foral 60/2014, de 16 de julio, por el que se establece el currículo de las enseñanzas de Educación primaria en la Comunidad Foral de Navarra. *Boletín Oficial de Navarra*, 174.
- Boletín Oficial las Illes Balears. (2016). Decreto 29/2016 de 20 de mayo, por el que se modifica el Decreto 34/2015, de 15 de mayo, por el que se establece el currículo de la Educación Secundaria Obligatoria en las Illes Balears. *Boletín Oficial Las Illes Balears*, 64, 15365–15394.
- Brennan, K. (2012). *Best of both worlds: issues of structure and agency in computational creation, in and out of school*. Tesis Doctoral. MIT Media Lab.
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Annual American Educational Research Association Meeting, Vancouver, Canada*, 1–25.
- Bull, G. (2005). Children , Computers , and Powerful Ideas. *Contemporary Issues in Technology and Teacher Education*, 5, 349–352.
- Bull, G., Garofalo, J., & Hguyen, N. R. (2020). Thinking about computational thinking: Origins of computational thinking in educational computing. *Journal of Digital Learning in Teacher Education*, 36(1), 6–18. <https://doi.org/10.1080/21532974.2019.1694381>
- Bundy, A. (2007). Computational Thinking is Pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67–69.
- Calao, L.A., Moreno-León, J., Correa, H.E., Robles, G. (2015). Developing Mathematical Thinking with Scratch. In: Conole, G., Klobučar, T., Rensing, C., Konert, J., Lavoué, E. (eds) *Design for Teaching and Learning in a Networked World. EC-TEL 2015*. Lecture Notes in Computer Science(), vol 9307. Springer, Cham. https://doi.org/10.1007/978-3-319-24258-3_2
- Calder, N. (2010). Using Scratch: An integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*, 15(4), 9–14.

- Campbell, D., & Fiske, D. (1959). Convergent and discriminant validation by the multitrait-multimethod matrix.. *Psychological Bulletin*, 56(2), 81–105. <https://doi.org/10.1037/h0046016>
- Campbell, D., & Stanley, J. (1966). *Experimental and quasi-experimental designs for research*. Houghton Mifflin Company. .
- Castañeda, M., Cabrera, A., Navarro, Y., & Vries, W. (2010). *Procesamiento de datos y análisis estadísticos utilizando SPSS*. Edipucrs.
- Castells, M. (2006). *La era de la información: Economía, sociedad y cultura*. Alianza.
- Castro, W., & Godino, J. (2011). Métodos mixtos de investigación en las contribuciones a los simposios de la SEIEM (1997-2010). In M.Marín et al (Ed.), *Investigación en Educación Matemática XV* (pp. 99–116).
- Chan, S. W., Looi, C. K., & Sumintono, B. (2020). Assessing computational thinking abilities among Singapore secondary students: a Rasch model measurement analysis. *Journal of Computers in Education*, 8, 213–236. <https://doi.org/10.1007/s40692-020-00177-2>
- Chiang, F. kuang, & Qin, L. (2018). A Pilot study to assess the impacts of game-based construction learning, using scratch, on students’ multi-step equation-solving performance. *Interactive Learning Environments*, 26(6), 803–814. <https://doi.org/10.1080/10494820.2017.1412990>
- Clark-wilson, A., Noss, R., Hoyles, C., Saunders, P., & Benton, L. (2016, February 6-10). *Key Factors for Successfully Embedding a Programming Approach to the Primary Maths Curriculum at Scale*. In 11th Congress of the European Society for Research in Mathematics Education, Utrecht.
- Clements, D. H. (1986). Effects of Logo and CAI Environments on Cognition and Creativity. *Journal of Educational Psychology*, 78(4), 309–318. <https://doi.org/10.1037/0022-0663.78.4.309>
- Cobo, J. C. (2008). El concepto de tecnologías de la información. Benchmarking sobre las definiciones de las TIC en la sociedad del conocimiento Informazio-teknologien. *Zer*, 14(27), 295–318.
- Colás, P., & Buendía, L. (1992). *Investigación Educativa*. Alfar.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

- Consejería de Educación y Deporte. (2019). Instrucción 12/2019, de 27 de junio, por la que se establecen aspectos de organización y funcionamiento para los centros que imparten Educación Primaria para el curso 2019/2020. *Consejería de Educación y Deporte*, 16.
- Consejo de la Unión Europea. (2018). Recomendación del Consejo, de 22 de mayo de 2018, relativa a las competencias clave para el aprendizaje permanente. *Diario Oficial de La Unión Europea*, C189/1(2), 1–13.
- Cooper, S., Dann, W., & Pausch, R. (2003). Using animated 3d graphics to prepare novices for cs1. *Computer Science Education*, 13(1), 3–30. <https://doi.org/10.1076/csed.13.1.3.13540>
- CSTA. (2016). *CSTA K–12 Computer Science Standards*. CSTA. <https://www.csteachers.org/page/standards>
- Culbertson, J. A. (1986). Whither computer literacy? In Culbertson and L. L. Cunningham (Ed.), *Microcomputers and Education* (Vol. 1, pp. 109–131). The University of Chicago Press.
- Curran, J. (2017). *A guide to programming languages for coding in class*. <https://medium.com/groklearning/a-guide-to-programming-languages-for-coding-in-class-4e6de6c01343>
- Curzon, P., Peckman, J., Settle, A., & Robert, E. (2009). Computational Thinking (CT): On Weaving *ACM SIGCSE Bulletin*, 41(3), 201–202. <https://doi.org/10.1145/1595496.1562941>
- Cutumisu, M., Adams, C., & Lu, C. (2019). A Scoping Review of Empirical Research on Recent Computational Thinking Assessments. *Journal of Science Education and Technology*, 28(6), 651–676. <https://doi.org/10.1007/s10956-019-09799-3>
- Dagienė, V., Futschek, G. (2008). Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. In: Mittermeir, R.T., Sysło, M.M. (eds) *Informatics Education - Supporting Computational Thinking. ISSEP 2008*. Lecture Notes in Computer Science, vol 5090. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-69924-8_2

- Deci, E. E. (1992). The Relation of Interest to the Motivation of Behavior: A Self-Determination Theory Perspective. In K. A. Renninger, S. Hidi, & A. Krapp (Eds.), *The Role of interest in Learning and Development* (pp. 43–70). Lawrence Erlbaum Associates, Inc. <https://doi.org/10.4324/9781315807430-12>
- Del Álamo Venegas, J. J., Alonso Díaz, L., Yuste Tosina, R., & López Ramos, V. (2021). La Dimensión Educativa De La Robótica: Del Desarrollo Del Pensamiento Al Pensamiento Computacional En El Aula. *Campo Abierto*, 40(2), 221–233. <https://doi.org/10.17398/0213-9529.40.2.221>
- Demartini, C. G., Benussi, L., Gatteschi, V., & Renga, F. (2020). Education and digital transformation: The “Riconnessioni” project. *IEEE Access*, 8, 1–34. <https://doi.org/10.1109/ACCESS.2020.3018189>
- Dhariwal, S. (2018). Scratch memories: A visualization tool for children to celebrate and reflect on their creative trajectories. *IDC 2018 - Proceedings of the 2018 ACM Conference on Interaction Design and Children*, 449–455. <https://doi.org/10.1145/3202185.3202770>
- Diari Oficial de la Generalitat Valenciana. (2015). DECRETO 87/2015, de 5 de junio, del Consell, por el que establece el currículo y desarrolla la ordenación general de la Educación Secundaria Obligatoria y del Bachillerato. *Diari Oficial de La Generalitat Valenciana*, 7544. https://dogv.gva.es/portal/ficha_disposicion_pc.jsp?sig=005254/2015&L=1
- Diario Oficial de Castilla-La Mancha. (2009). Decreto 40/2015, de 15/06/2015, por el que se establece el currículo de Educación Secundaria Obligatoria y Bachillerato. *Diario Oficial de Castilla-La Mancha*, 18872–20324.
- Diario Oficial de Galicia. (2015). ORDEN de 15 de julio de 2015 por la que se establece la relación de materias de libre configuración autonómica. *Diario Oficial de Galicia*, 136, 30316–30431.
- DiSessa, A. (2000). *Changing minds: computers, learning, and literacy*. MIT Press.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

- Du, J., & Wimmer, H. (2019). Information Systems Education Journal. *Information Systems Education Journal (ISEDJ)*, 17(4), 91–100.
- Duo-terron, P. (2022). *STEAM in Primary Education . Impact on Linguistic and Mathematical Competences in a Disadvantaged Context*. 7(June), 1–14. <https://doi.org/10.3389/feduc.2022.792656>
- Ershov, A. P. (1981). Programming, the second literacy. *Microprocessing and Microprogramming*, 8(1), 1–9. [https://doi.org/10.1016/0165-6074\(81\)90002-8](https://doi.org/10.1016/0165-6074(81)90002-8)
- Esteve-Mon, F., Llopis, M. Á., & Adell Segura, J. (2020). Digital competence and computational thinking of student teachers. *International Journal of Emerging Technologies in Learning*, 15(2), 29–41. [doi:10.3991/ijet.v15i02.11588](https://doi.org/10.3991/ijet.v15i02.11588)
- European Commission. (2016). *Communication on new skills agenda, human capital, employability and competitiveness 2016*. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52016DC0381>
- Evangelopoulou, O., & Xinogalos, S. (2018). MYTH TROUBLES: An Open-Source Educational Game in Scratch for Greek Mythology. *Simulation and Gaming*, 49(1), 71–91. <https://doi.org/10.1177/1046878117748175>
- Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jr. On the iPad. *Journal of Computer Assisted Learning*, 32(6), 576–593. <https://doi.org/10.1111/jcal.12155>
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers and Education*, 63, 87–97. <https://doi.org/10.1016/j.compedu.2012.11.016>
- Furber, S. (2012). Shut down or restart? The way forward on computing in UK schools. In *British Journal of Educational Technology* (Issue January, pp. 789–801).
- Gabriele, L., Bertacchini, F., Tavernise, A., Vaca-Cárdenas, L., Pantano, P., & Bilotta, E. (2019). Lesson planning by computational thinking skills in Italian pre-service teachers. *Informatics in Education*, 18(1), 69–104. <https://doi.org/10.15388/infedu.2019.04>

- Gadanidis, G., Hughes, J. M., Minniti, L., & White, B. J. G. (2016). Computational Thinking, Grade 1 Students and the Binomial Theorem. *Digital Experiences in Mathematics Education*, 3(2), 77–96. <https://doi.org/10.1007/s40751-016-0019-3>
- Gander, W., Petit, A., Berry, G., Demo, B., Vahrenhold, J., McGettrick, A., Boyle, R., Drechsler, M., Mendelson, A., Stephenson, C., Ghezzi, C., & Meyer, B. (2013). *Informatics Education: Europe Cannot Afford to Miss the Boat*. Informatics Europe & ACM Europe working group on informatics Education. <https://www.informatics-europe.org/images/documents/informatics-education-acm-ie.pdf>
- García, Y., Reyes, D., & Burgos, F. (2017). Actividades STEM en la formación inicial de profesores: nuevos enfoques didácticos para los desafíos del siglo XXI. *Diálogos Educativos*, 18(33), 35–46.
- Gaulin, D. C. (2001). Tendencias actuales en la resolución de problemas. *Sigma*, 19, 51–63.
- Generalitat de Catalunya. (2015a). Decreto 119/2015, de 23 de junio, de ordenación de las enseñanzas de la Educación Primaria. *Diari Oficial de La Generalitat de Catalunya*, 120, 18872–20324. <https://portaldogc.gencat.cat/utillsEADOP/PDF/6900/1431927.pdf>
- Generalitat de Catalunya. (2015b). DECRETO 187/2015, de 25 de agosto, de ordenación de las enseñanzas de la Educación Secundaria Obligatoria. *Diari Oficial de La Generalitat de Catalunya*, 6945, 1–314.
- Gretter, S., & Yadav, A. (2016). Computational Thinking and Media & Information Literacy: An Integrated Approach to Teaching Twenty-First Century Skills. *TechTrends*, 60(5), 510–516. <https://doi.org/10.1007/s11528-016-0098-4>
- Grover, S. (2015). “Systems of Assessments” for Deeper Learning of Computational Thinking in K-12. In *Proceedings of the 2015 Annual Meeting of the American Educational Research Association* (pp. 15–20).
- Grover, S., Cooper, S., & Pea, R. (2014). Assessing computational learning in K-12. *ITICSE 2014 - Proceedings of the 2014 Innovation and*

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Technology in Computer Science Education Conference, 57–62.
<https://doi.org/10.1145/2591708.2591713>

Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43.
<https://doi.org/10.3102/0013189X12463051>

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. In S. Sentence, E. Barendsen, & C. Schulte (Eds.), *Computer science education: Perspectives on teaching and learning in school* (pp. 20–35).

Guamán Gómez, V. J., Daquilema Cuásquer, B. A., & Espinoza Guamán, E. E. (2021). El pensamiento computacional en el ámbito educativo. *Sociedad & Tecnología*, 2(1), 59–67. <https://doi.org/10.51247/st.v2i1.69>

Gurstein, M. (2011). Open Data: Empowering the empowered of effective data use for everyone. *First Monday*, 16(2), 2–7.
<http://firstmonday.org/article/view/3316/2764>

Gutierrez-Esteban, P., & Beccerra, 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., & Sheridan, K. (2014). The Maker Movement in Education. *Harvard Educational Review*, 84(4), 495–504.
<https://doi.org/10.17763/haer.84.4.34j1g681403820>

Haseski, H. I., Ilic, U., & Tugtekin, U. (2018). Defining a New 21st Century Skill-Computational Thinking: Concepts and Trends. *International Education Studies*, 11(4), 29. <https://doi.org/10.5539/ies.v11n4p29>

Hassan, H., Adnan, A., Poh, L. G., & Hashim, M. S. M. (2018). Using Scratch Programming To Engage Primary School Penggunaan Scratch Programming Untuk Meningkatkan Penglibatan Murid Sekolah Rendah. *Jurnal Penyelidikan Tempawan*, XXXV, 49–57.

Hassenfeld, Z. R., & Bers, M. U. (2020). Debugging the Writing Process: Lessons From a Comparison of Students' Coding and Writing Practices. *Reading Teacher*, 73(6), 735–746. <https://doi.org/10.1002/trtr.1885>

- Hemendinger, D. (2010). A plea for modesty. *ACM Inroads*, 1(2), 4–7. <https://doi.org/10.1145/1805724.1805725>
- Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2017). A Scoping Review of Studies on Computational Thinking in K–12 Mathematics Classrooms. *Digital Experiences in Mathematics Education*, 1–22. <https://doi.org/10.1007/s40751-017-0038-8>
- Holmes, K., Prieto-Rodriguez, E., Berger, & Holmes, K., Prieto-Rodriguez, E., Hickmott, D., & Berger, N. (2018). *Using coding to teach mathematics: results of a pilot project*. In *Integrated Education for the Real World: 5th International STEM in Education Conference: Post-conference Proceedings*. November, 152–158.
- Hoyles, R., & Noss, C. (1996). Windows on mathematical meanings: learning cultures and computers. In *Choice Reviews Online* (Board, Vol. 17). Kluwer. <https://doi.org/10.5860/choice.35-0959>
- Hubwieser, P., & Mühling, A. (2014). *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*. 128–129. <http://dx.doi.org/10.1145/2670757.2670759>
- Iskrenovic-Momcilovic, O. (2020). Improving Geometry Teaching with Scratch. *International Electronic Journal of Mathematics Education*, 15(2). <https://doi.org/10.29333/iejme/7807>
- ISTE & CSTA. (2011). *Operational Definition of Computational Thinking for K-12 Education*.
- Ivanova, A. A., Srikant, S., Sueoka, Y., Kean, H. H., Dhamala, R., O'Reilly, U. M., Bers, M. U., & Fedorenko, E. (2020). Comprehension of computer code relies primarily on domain-general executive resources. *BioRxiv*, 1–24. <https://doi.org/10.1101/2020.04.16.045732>
- Jacob, S. R., & Warschauer, M. (2018). Computational Thinking and Literacy. *Journal of Computer Science Integration*, 1(1), p.1. <https://doi.org/10.26716/jcsi.2018.01.1.1>
- Jaeggi, S. M., Buschkuhl, M., Shah, P., & Jonides, J. (2014). The role of individual differences in cognitive training and transfer. *Memory & Cognition*, 42(3), 464–480. <https://doi.org/10.3758/s13421-013-0364-z>

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

- Johnson, B., Onwuegbuzie, A., & Turner, L. (2007). Toward a Definition of Mixed Methods Research. *Journal of Mixed Methods Research*, 1(2), 112–133. <https://doi.org/10.1177/1558689806298224>
- Johnson, L., Adams, B., Estrada, V., & Freeman, A. (2014). The NMC Horizon Report: 2014 K-12. The New Media Consortium. <http://www.nmc.org/publications/2014-horizon-report-k12>
- Juškevičiene, A., & Dagiene, V. (2018). Computational thinking relationship with digital competence. *Informatics in Education*, 17(2), 265–284. <https://doi.org/10.15388/infedu.2018.14>
- Kafai, Y., & Burke, Q. (2014). *Connected code: why children need to learn programming*. MIT Press.
- Kafai, Y., & Resnick, M. (1996). *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World*. Lawrence Erlbaum Associates.
- Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers and Education*, 73, 26–39. <https://doi.org/10.1016/j.compedu.2013.12.010>
- Kesan, C., & Caliscan, S. (2013). The Effect Of Learning Geometry Topics Of 7 Th Grade In Primary Education With Dynamic Geometer`s Sketchpad Geometry. *The Turkish Online Journal of Educational Technology*, 12(1), 131–139.
- Keshavarz, H. (2020). Web self-efficacy: A psychological prerequisite for web literacy. *Webology*, 17(1), 81–98. <https://doi.org/10.14704/WEB/V17I1/A209>
- Kiliç-Çakmak, E. (2010). Learning strategies and motivational factors predicting information literacy self-efficacy of e-learners. *Australasian Journal of Educational Technology*, 26(2), 192–208. <https://doi.org/10.14742/ajet.1090>
- 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).
- Knuth, D. E. (1974). Computer Science and its Relation to Mathematics. *The American Mathematical Monthly*, 81(4), 323–343. <https://doi.org/10.1080/00029890.1974.11993556>
- Kong, S. (2019). Computational Thinking Education. In *Computational Thinking Education*. Springer Singapore. <https://doi.org/10.1007/978-981-13-6528-7>
- Kotsopoulos, D., Floyd, L., Nelson, V., & Makosz, S. (2019). Mathematical or Computational Thinking? An Early Years Perspective. *Mathematical Learning and Cognition in Early Childhood*, 79–90. https://doi.org/10.1007/978-3-030-12895-1_6
- Krapp, A. (2002). Structural and dynamic aspects of interest development: Theoretical considerations from an ontogenetic perspective. *Learning and Instruction*, 12(4), 383–409. [https://doi.org/10.1016/S0959-4752\(01\)00011-1](https://doi.org/10.1016/S0959-4752(01)00011-1)
- Krauss, J., & Protsman, K. (2017). *Computational Thinking and Coding for Every Student. The Teacher's Getting-Started Guide*. Crowin Press Inc.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32–37. <https://doi.org/10.1145/1929887.1929902>
- Lerís, M. D., & Sein-Echaluce, 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., & Zawojeuski, 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).
- Leung, H. Y.-H. (2021). Integrating computational thinking into mathematics education. In et al. M. Chang (Ed.), *Doctoral student forum proceedings* 223

of the 25th Global Chinese Conference on Computers in Education (GCCCE 2021) (pp. 7–12).

- Lewis, C. M. (2010). How programming environment shapes perception, learning and goals: Logo vs. Scratch. *SIGCSE'10 - Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, 346–350. <https://doi.org/10.1145/1734263.1734383>
- Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). On Computational Thinking and STEM Education. *Journal for STEM Education Research*, 3(2), 147–166. <https://doi.org/10.1007/s41979-020-00044-w>
- Lie, J., Hauge, I., & Meaney, T. (2017). Computer Programming in the Lower Secondary Classroom: Mathematics Learning. *Italian Journal of Educational Technology*, 25(2), 27–35. <https://doi.org/10.17471/2499-4324/911>
- Lindh, J., & Holgersson, T. (2007). Does lego training stimulate pupils' ability to solve logical problems? *Computers and Education*, 49(4), 1097–1111. <https://doi.org/10.1016/j.compedu.2005.12.008>
- López Rupérez, F. (2022). El enfoque del currículo por competencias. Un análisis de la LOMLOE. *Revista Española de Pedagogía*, 80(281), 55–68. <https://doi.org/https://doi.org/10.22550/REP80-1-2022-05>
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 1–15. <https://doi.org/10.1145/1868358.1868363>
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. *ITiCSE-WGR 2014 - Working Group Reports of the 2014 Innovation and Technology in Computer Science Education Conference, January 2016*, 1–29. <https://doi.org/10.1145/2713609.2713610>
- Mantilla, R., & Negre, F. (2021). Pensamiento computacional, una estrategia educativa en épocas de pandemia. *INNOEDUCA Revista*, 7(1), 89–106. <https://doi.org/10.24310/innoeduca.2021.v7i1.10593>

- Martínez, M., Castellanos López, M. Á., & Chacón López, J. C. (2014). *Métodos de investigación en psicología*. Madrid: EOS Universitaria.
- Martínez, R. (2007). *La investigación en la práctica educativa: Guía metodológica de investigación para el diagnóstico y evaluación en los centros docentes*. Ministerio de Educación y Ciencia.
- Mayer, R. E., & Wittrock, M. C. (2006). Problem Solving. In P. A. Alexander & P. H. Winne (Eds.), *Handbook of Educational Psychology* (pp. 287-303). Routledge. <https://doi.org/10.4324/9780203874790.ch13>
- Mayoral Rodríguez, S., Roca Tena, M., Timoneda i Gallart, C., & Serra Sala, M. (2015). Mejora de la capacidad de planificación 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>
- McComas, W. F. (2013). *The Language of Science Education: an Expanded Glossary of Key Terms and Concepts in Science Teaching and Learning*. Rotterdam: Sense Publishers.
- MECD. (2018). *Programación, robótica y pensamiento computacional en España*. <https://code.intef.es/wp-content/uploads/2017/09/Pensamiento-Computacional-Fase-1-Informe-sobre-la-situación-en-España.pdf>
- Meseguer, P., Moreno, J., Moreno, J. J., Olcoz, K., Pimentel, E., Toro, M., Velázquez, Á., & Vendrell, E. (2015). *Enseñanza de la informática en primaria, secundaria y bachillerato: estado español*.
- Ministerio de Educación y Formación Profesional. (2019). *Escuela de Pensamiento Computacional*. https://intef.es/wp-content/uploads/2019/12/Impacto_EscueladePensamientoComputacional_Curso2018-2019.pdf
- Ministerio de Educación y Formación Profesional. (2022a). *Real Decreto 157 / 2022 , de 1 de marzo , por el que se establecen la ordenación y las enseñanzas mínimas de la Educación Primaria .* (pp. 1–109).

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

- Ministerio de Educación y Formación Profesional. (2022b). *Real Decreto 217/2022, de 29 de marzo, por el que se establece la ordenación y las enseñanzas mínimas de la Educación Secundaria Obligatoria* (pp. 41571–41789).
- Mitchell, M. (1992). Situational Interest: Its Multifaceted Structure in the Secondary Mathematics Classroom. *Journal of Educational Psychology*, 85(3), 424–436.
- Mladenović, M., Krpan, D., & Mladenovi, S. (2017). Learning programming from scratch. *Turkish Online Journal of Educational Technology*, 2017(November Special Issue INTE), 419–427.
- Molina, A., Adamuz, N., & Bracho, R. (2020). La resolución de problemas basada en el método de Polya usando el pensamiento computacional y Scratch con estudiantes de Educación Secundaria. *Aula Abierta*, 5(1), 83–90. <https://doi.org/10.17811/rifie.49.1.2020.83-90>
- Montgomery, D. C. (2019). *Design and Analysis of Experiments* (10th ed.). WILEY .
- Moreno-Leon, J., & Robles, G. (2016). Code to learn with Scratch? A systematic literature review. *IEEE Global Engineering Education Conference, EDUCON*, 10–13, 150–156. <https://doi.org/10.1109/EDUCON.2016.7474546>
- Moreno-León, J., & Robles, G. (2015). The Europe Code Week (CodeEU) initiative. *2015 IEEE Global Engineering Education Conference (EDUCON), March*, 554–559.
- Moreno-León, J., Robles, G., & Román-González, M. (2016). Code to Learn: Where Does It Belong in the K-12 Curriculum? *Jite.Org*, 15, 283–303.
- Moreno-León, J., Robles, G., & Román-González, M. (2017). Towards Data-Driven Learning Paths to Develop Computational Thinking with Scratch. *IEEE Transactions on Emerging Topics in Computing*, 8(1), 193–205. <https://doi.org/10.1109/TETC.2017.2734818>
- Moreno-León, 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*, 26–35.
<https://doi.org/10.6018/riite.397151>
- Moreno-Morilla, C., Guzmán-Simón, F., & García-Jiménez, E. (2021). Digital and information literacy inside and outside Spanish primary education schools. *Learning, Culture and Social Interaction*, 28, 100455.
<https://doi.org/10.1016/j.lcsi.2020.100455>
- Morse, J. M., & Niehaus, L. (2009). *Mixed Method Design: Principles and Procedures*. Left Coast Press, Inc.
- National Research Council. (2002). *Technically Speaking: Why All Americans Need to Know More About Technology*. National Academy of Engineering. <https://doi.org/https://doi.org/10.17226/10250>
- National Research Council. (2004). *Computer Science: reflections on fields, reflections on the field*. The National Academy Press.
- National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. National Research Council of the National Academies. <https://doi.org/https://doi.org/10.17226/12840>
- Nelson, J. (2009). Celebrating Scratch in Libraries. Creation software helps young people develop 21st-century literacy skills. *School Library Journal*, 20–21.
- Niss, M. (2002). Mathematical Competencies and the Learning of Mathematics: the Danish Kom Project. In *Proceedings of the 3rd Mediterranean Conference on Mathematical Education*.
- Nugent, G., Barker, B., Grandgenett, N., & Adamchuk, V. (2009). The use of digital manipulatives in K-12: Robotics, GPS/GIS and programming. *Proceedings - Frontiers in Education Conference, FIE*, 1–6.
<https://doi.org/10.1109/FIE.2009.5350828>
- Obaya, A. (2003). El construccionismo y sus repercusiones en el aprendizaje asistido por computadora. *ContactoS*, 48, 61–64.
- Olabe, J. C., Olabe, M. A., Basogain, X., Maiz, I., & Castaño, C. (2011). Programming and Robotics with Scratch in Primary Education.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Education in a Technological World: Communicating Current and Emerging Research and Technological Efforts, July 2016, 356–363.

- 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), 1–147. <http://dx.doi.org/10.6018/red/50/3>
- Onwuegbuzie, A. J., & Collins, K. M. . (2007). A typology of mixed methods sampling designs in social science research. *The Qualitative Report*, 12(2), 474–498.
- Ortega-Ruipérez, B., & Asensio, M. (2021). Evaluar el Pensamiento Computacional mediante Resolución de Problemas: Validación de un Instrumento de Evaluación. *Revista Iberoamericana de Evaluación Educativa*, 14(1), 153–171. <https://doi.org/10.15366/riee2021.14.1.009>
- Orton, K., Weintrop, D., Beheshti, E., Horn, M., Jona, K., & Wilensky, U. (2016). Bringing computational thinking into high school mathematics and science classrooms. *Transforming Learning, Empowering Learners: The International Conference of the Learning Sciences (ICLS)*, 705–712. <https://doi.org/10.3102/0013189X12463051>
- Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., & Lahmine, S. (2015). Learning Basic Programming Concepts by Creating Games with Scratch Programming Environment. *Procedia - Social and Behavioral Sciences*, 191, 1479–1482. <https://doi.org/10.1016/j.sbspro.2015.04.224>
- Palts, T., & Pedaste, M. (2020). A model for developing computational thinking skills. *Informatics in Education*, 19(1), 113–128. <https://doi.org/10.15388/INFEDU.2020.06>
- Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2014). Novice programming environments. Scratch & app inventor: A first comparison. *ACM International Conference Proceeding Series*, 09-June-20, 1–7. <https://doi.org/10.1145/2643604.2643613>
- Papert, S. (1972). Teaching Children Thinking. *Programmed Learning and Educational Technology*, 9(5), 245–255. <https://doi.org/10.1080/1355800720090503>

- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas* (2nd Ed.). Basic Books.
- Papert, S. (1996). An exploration in the Space of Mathematics Education. *International Journal of Computers for Mathematical Learning*, 1(1), 95–123.
- Papert, S. (1999). What is Logo and who needs it? In *Logo philosophy and implementation*. <https://doi.org/10.1057/s41599-018-0114-8>
- Papert, S., & Harel, I. (1991a). Situating constructionism. *Constructionism*, 36, 1–14.
- Parsons, D., & Haden, P. (2007). Programming osmosis: Knowledge transfer from imperative to visual programming environments. *Conference of the National Advisory Committee on Computing Qualifications, May*, 209–215. <http://www.citrenz.ac.nz/conferences/2007/209.pdf>
- Pérez-Escoda, A., García-Ruiz, R., & Agüaded, I. (2019). Dimensions of digital literacy based on five models of development. *Cultura y Educación*, 31(2), 232–266. <https://doi.org/10.1080/11356405.2019.1603274>
- Phillips, R. S., & Brooks, B. P. (2014). The Hour of Code: Impact on Attitudes Towards and Self-Efficacy with Computer Science. *Code*, 66(4), 0–6.
- Podestá, P. (2011). *Geometría* (1ª Edición). Ministerio de Educación de la Nación.
- Polly, D. (2015). Elementary Education Pre-service Teachers' Development of Mathematics Technology Integration Skills in a Technology Integration Course. *Journal of Computers in Mathematics and Science Teaching*, 34(4), 431–453.
- Polya, G. (1945). *How To Solve It* (2nd ed.). Princeton University Press, Doubleday Anchor Books.
- Polya, G. (1965). *Mathematical Discovery*. John Wiley and Sons.
- Pomper, M. A. (1990). LOGO in Practice. *Journal of Visual Literacy*, 10(2), 84–141. <https://doi.org/10.1080/23796529.1990.11674456>

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

- Pozo, J. (2000). *Aprendices y maestros*(. Alianza Editorial.
- Prat, C. S., Madhyastha, T. M., Mottarella, M. J., & Kuo, C. H. (2020). Relating Natural Language Aptitude to Individual Differences in Learning Programming Languages. *Scientific Reports*, 10(1), 1–10. <https://doi.org/10.1038/s41598-020-60661-8>
- Prendes Espinosa, M. P., & Cerdán Cartagena, F. (2020). Tecnologías avanzadas para afrontar el reto de la innovación educativa. *RIED. Revista Iberoamericana de Educación a Distancia*, 24(1), 35. <https://doi.org/10.5944/ried.24.1.28415>
- Prensky, M. (2001). Digital natives, digital immigrants. *From On the Horizon. MCB University Press*, 9(5), 1–6.
- Prensky, M. (2008). *Programming Is the New Literacy*. <https://www.edutopia.org/literacy-computer-programming>
- Recio, T. (2002). *Situación de la Enseñanza de las Matemáticas en la Educación Secundaria Española*.
- Redecker, C. (2017). *Digital Competence of Educators (DigCompEdu)*. Publications Office of the European Union, Luxembourg. <https://publications.jrc.ec.europa.eu/repository/handle/JRC107466>
- Reig, D., & Vílchez, L. (2015). Los jóvenes en la era de la hiperconectividad: tendencias, claves y miradas. *Revista Fuentes*, 17, 145–147. <https://doi.org/10.12795/revistafuentes.2015.i17.08>
- Resnick, M. (2007). Sowing the Seeds for a more Creative Society. *Learning & Leading with Technology*, 35(4), 18–22. <https://doi.org/10.1145/1518701.2167142>
- Resnick, M. (2013). Learn to code, code to learn. *EdSurge*, Mayo. <http://web.media.mit.edu/~mres/papers/L2CC2L-handout.pdf>
- Resnick, M., Kafai, Y., & Maeda, J. (2003). A Networked, Media-Rich Programming Environment to Enhance Technological Fluency at After-School Centers in Economically-Disadvantaged Communities. *Proposal to the National Science Foundation*, 1–14.

- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communication of the ACM*, 52(11), 62–67. <https://doi.org/10.1145/1592761.1592779>
- Riina, V., Yves, P., Gómez, S. C., & Brade, G. Van den. (2016). *DigComp 2.0: The Digital Competence Framework for Citizens. Update Phase 1: the Conceptual Reference Model*.
- Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: an experiment with sixth-grade students. *Interactive Learning Environments*, 28(3), 316–327. <https://doi.org/10.1080/10494820.2019.1612448>
- Román-González, M. (2015). Computational Thinking Test: Design Guidelines and Content Validation. *Proceedings of the 7th Annual International Conference on Education and New Learning Technologies (EDULEARN 2015)*, January, 2436–2444. <https://doi.org/10.13140/RG.2.1.4203.4329>
- Román-González, M. (2016). *Codigoalfabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de programas*. UNED
- Román-González, M., Moreno-León, J., & Robles, G. (2017). Complementary Tools for Computational Thinking Assessment. *Proceedings of International Conference on Computational Thinking Education (CTE 2017)*, 154–159.
- Romero-Frías, E., y Magro, C. (2016). La emancipación intelectual en la sociedad digital: El maestro ignorante de Rancière en nuestros días. *Revista Letral*, 16, 89–105.
- Ross, E. W., & Gautreaux, M. (2018). Pensando de Manera Crítica sobre el Pensamiento Crítico. *Aula Abierta*, 47(4), 383. <https://doi.org/10.17811/rifie.47.4.2018.383-386>
- Roy, S., & Roth, D. (2015). Solving general arithmetic word problems. *Conference Proceedings - EMNLP 2015: Conference on Empirical*

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

Methods in Natural Language Processing, 1743–1752.
<https://doi.org/10.18653/v1/d15-1202>

Rushkoff, D. (2012, November 13). *Code Literacy: A 21st-Century Requirement*. <https://www.edutopia.org/blog/code-literacy-21st-century-requirement-douglas-rushkoff>

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 and Education*, 97, 129–141.
<https://doi.org/10.1016/j.compedu.2016.03.003>

Santos-Trigo, M. (2014). Problem solving in mathematics education. In S. Lerman (Ed.), *Encyclopedia of Mathematics Education* (pp. 496–501). NY: Springer.

Savolainen, R. (2002). Network competence and information seeking on the Internet: From definitions towards a social cognitive model. *Journal of Documentation*, 58(2), 211–226.
<https://doi.org/10.1108/00220410210425467>

Schoenfeld, A. H. (1985). *Mathematical Problem Solving*. 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).
https://doi.org/10.1007/978-3-319-12688-3_16

Schwab, K. (2017). *The fourth industrial revolution*. Crown Business.

Selby, C. (2013). Computational Thinking : The Developing Definition. *ITiCSE Conference 2013*, 5–8.

Selwyn, N. (2009). The digital native: myth and reality. *Aslib Proceedings*, 6(4), 364–379. <https://doi.org/10.1108/00012530910973776>

Selwyn, N. (2013). *Educational in a digital world. Global Perspectives on Technology and Education*. Routledge.

- Sheil, B. A. (1980). Teaching procedural literacy. *Proceedings of the ACM 1980 Annual Conference, ACM 1980*, 125–126. <https://doi.org/10.1145/800176.809944>
- Sinclair, N., & Bruce, C. D. (2015). New opportunities in geometry education at the primary school. *ZDM Mathematics Education*, 47(3), 319–329. <https://doi.org/10.1007/s11858-015-0693-4>
- Sneider, C., Stephenson, C., Schafer, B., & Flick, L. (2014). Teacher’s Toolkit: Exploring the Science Framework and NGSS: Computational Thinking in the Science Classroom. *Science Scope*, 038(03), 10–15. https://doi.org/10.2505/4/ss14_038_03_10
- Song, J. (2017). Effects of Learning through Scratch-Based Game Programming on Students’ Interest in and Perceived Value of Mathematics Curriculum. *Journal of The Korean Association of Information Education*, 21(2), 199–208. <https://doi.org/10.14352/jkaie.21.2.199>
- Stoitsov, G., & Stoitsova, G. (2019). Increasing the Motivation of Primary School Pupils Through the Use of Ict in the Educational Process. *International Journal of Research -GRANTHAALAYAH*, 7(2), 207–213. <https://doi.org/10.29121/granthaalayah.v7.i2.2019.1025>
- Suárez Zapata, A., García Costa, D., Martínez Delgado, P. A., & Martos Torres, J. (2018). Contribución de la robótica educativa en la adquisición de conocimientos de matemáticas en la Educación Primaria. *Magister*, 30(1), 43–54.
- Sung, W., Ahn, J., & Black, J. B. (2017). Introducing Computational Thinking to Young Learners: Practicing Computational Perspectives Through Embodiment in Mathematics Education. *Technology, Knowledge and Learning*, 22(3), 443–463. <https://doi.org/10.1007/s10758-017-9328-x>
- Syśło, M. M., & Kwiatkowska, A. B. (2014). Learning Mathematics Supported By Computational Thinking. In *Constructionism and Creativity* (pp. 258–268). G. Futschek, G., Kynigos, C.
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies.

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

- Computers and Education*, 148(April), 103798.
<https://doi.org/10.1016/j.compedu.2019.103798>
- Taylor, M., Harlow, A., & Forret, M. (2010). *Using a Computer Programming Environment and an Interactive Whiteboard to Investigate Some Mathematical Thinking*. 8, 561–570.
<https://doi.org/10.1016/j.sbspro.2010.12.078>
- Tedre, M., & Denning, P. J. (2016). The long quest for computational thinking. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (pp. 120–129). Koli Calling '16, New York. ACM. <https://doi.org/10.1145/2999541.2999542>
- Téliz, F. (2015). Uso didáctico de las TIC en las buenas prácticas de enseñanza de las matemáticas. Estudio de las opiniones y concepciones de docentes de Educación Secundaria en el departamento de Artigas. *Cuadernos de Investigación Educativa*, 6(2).
- Tsur, M. (2017). *Scratch Microworlds : introducing novices to scratch using an interest-based, open-ended, scaffolded experience* [Graduate dissertation, Massachusetts Institute of Technology]. MIT Libraries. <https://dspace.mit.edu/handle/1721.1/112561>
- Utreras, E., Pontelli, E. (2020). Accessibility of Block-Based Introductory Programming Languages and a Tangible Programming Tool Prototype. In: Miesenberger, K., Manduchi, R., Covarrubias Rodriguez, M., Peñáz, P. (eds) *Computers Helping People with Special Needs. ICCHP 2020. Lecture Notes in Computer Science()*, vol 12376. Springer, Cham. https://doi.org/10.1007/978-3-030-58796-3_4
- Vaca Cárdenas, L. A., Tavernise, A., Bertacchini, F., Gabriele, L., Valenti, A., Pantano, P., & Bilotta, E. (2016). An Educational Coding Laboratory for Elementary Pre-service Teachers: A Qualitative Approach. *International Journal of Engineering Pedagogy (IJEP)*, 6(1), 11–17.
<https://doi.org/10.3991/ijep.v6i1.5364>
- Vaikutyté-Paškauskė, Justina Vaičiukynaitė, J., & Donatas, P. (2018). *Research for CULT Committee - Digital Skills in the 21st century*. European Parliament

- Valles, M. (2002). *Entrevistas cualitativas*. Centro de Investigaciones Sociológicas.
- Van Djick, J. (2016). *La cultura de la conectividad*. Siglo XXI.
- Verd Pericás, J., & Lozares Colina, C. (2016). *Introducción a la investigación cualitativa*. Síntesis.
- Vidal-Silva, C., Jiménez, C., Yupanqui, M. T., Sepúlveda, S., & Madariaga, E. (2020). Development of algorithmic and mathematical-logic competences of children in Chile with scratch. *International Journal of Scientific and Technology Research*, 9(4), 381–385.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. <https://doi.org/10.1007/s10956-015-9581-5>
- Wilson, C. (2015). Hour of code - A record year for computer science. *ACM Inroads*, 6(1), 22. <https://doi.org/10.1145/2723168>
- Wing, J. (2006). Computational Thinking. *Communication of the ACM*, 49(3), 33–35.
- Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725. <https://doi.org/10.1098/rsta.2008.0118>
- Wing, J. (2011). Research notebook: Computational thinking - What and why? *The Link Magazine, Spring*(6), 20–23.
- Wing, J. (2014, January 10). *Computational Thinking Benefits Society*. 40th Anniversary Blog of Social Issues in Computing, 2014. <http://socialissues.cs.toronto.edu/index.html%3Fp=279.html>
- Wolfram, S. (2016, September 7). *How to Teach Computational Thinking*. Stephen Wolfran Writings <https://writings.stephenwolfram.com/2016/09/how-to-teach-computational-thinking/>

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

- Yadav, A., Mayfield, C., Zhou, N., Hambruch, S., & Korb, J. T. (2014a). Computational Thinking in Elementary and Secondary Teacher Education. *ACM Transactions on Computing Education*, 14(1), 1–16. <https://doi.org/10.1145/2576872>
- Yadav, A., Mayfield, C., Zhou, N., Hambruch, S., & Korb, J. T. (2014b). Computational Thinking in Elementary and Secondary Teacher Education. *ACM Trans. Comput. Educ.*, 14(1).
- Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, 60(4), 55–62. <https://doi.org/10.1145/2994591>
- Yadav, A., Zhou, N., Mayfield, C., Hambruch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. *SIGCSE'11 - Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, 465–470. <https://doi.org/10.1145/1953163.1953297>
- Yelland, N. (1995). Mindstorms or a storm in a teacup? A review of research with Logo. *International Journal of Mathematical Education in Science and Technology*, 26(6), 853–869. <https://doi.org/10.1080/0020739950260607>

Referencias bibliográficas

V. Anexos

Anexo 1. Problemas de pretest para estudiantes de Educación Secundaria

1. El cartel de una cafetería nos muestra su oferta para el desayuno:

Zumo 1,5 € Tortilla 1,7 €

Tostada 1 € Café 0,9 €

Al comprar dos de los productos que aparecen en el anuncio, Antonio pagó 3 € y le sobraron 0,4 €. ¿Qué dos productos compró?

2. Alicia quiere hacer 300 km del Camino de Santiago. Planea hacer 18 km diarios y llegar a Santiago el día del Santo (25 de julio). ¿Qué día de julio debe ponerse en marcha?

3. Si le doy a mi primo dos tabletas de chocolate, me presta su bicicleta durante 3 horas. Si le doy 12 bombones, me la presta 2 horas. Mañana le daré una tableta de chocolate y tres bombones ¿cuánto tiempo me dejará la bicicleta?

a) Media hora b) 1 hora c) 2 horas d) 3 horas e) 4 horas

4. El equipo de fútbol del instituto decide celebrar su victoria de liga yendo de viaje con su entrenador. Sabiendo que el equipo lo componen 20

alumnos, que el viaje les cuesta a cada uno 150 €, la noche en habitación individual 50 € y que han pagado 7350 € en total, ¿cuántos días han estado de viaje?

5. Un autobús comienza el viaje con 45 pasajeros. En la primera parada se bajan 7 y se suben 12. En la segunda se bajan 10 y se suben 8, y en la tercera se bajan 4. ¿Cuántos pasajeros hay en el autobús?

6. José fue a la papelería para comprar material escolar. Con 3,5 € compró dos bolígrafos y un rotulador. Ana se gastó en la misma papelería 2€ por el mismo rotulador y un bolígrafo igual que el de José. ¿Cuánto cuesta un bolígrafo?
¿Cuánto cuesta el rotulador?

Anexo 2. Problemas del postest para estudiantes de Educación

Secundaria

1. Dos impresoras tienen que imprimir 440 carteles. En un minuto, la rápida imprime siete carteles y la lenta solo seis. Comienzan juntas a funcionar y al cabo de un rato la rápida se estropea por lo que la lenta deberá seguir trabajando 17 minutos más. ¿Cuántos carteles imprimió la rápida?
2. Lucía tiene 600 € y cada semana recibe 25 €. Julián tiene 1200 € y gasta 25 € por semana. ¿Dentro de cuántas semanas, a partir de ahora, los dos tendrán la misma cantidad de dinero?
3. Fernando entró en un ascensor de unos grandes almacenes. El ascensor, primero bajó tres plantas, a continuación bajó dos, después subió siete plantas, bajó cuatro y subió seis. Si, tras todo esto, Fernando se bajó en la 3ª planta, ¿en qué planta tomó el ascensor?
4. En un quiosco de prensa al final de la mañana se ha vendido la mitad de los periódicos. Por la tarde se vendieron la mitad de los que quedaban y se quedaron 40 periódicos sin vender. ¿Cuántos periódicos había en el quiosco al comenzar el día?
5. En una caja hay más de 40 monedas pero menos de 70. Si las repartimos a partes iguales entre 6 personas sobran 4, pero si lo hacemos entre 5 sobran 3

monedas. ¿Cuántas sobrarían si las repartiéramos equitativamente entre 7 personas?

6. José fue a la papelería para comprar material escolar. Con 3,5 € compró dos bolígrafos y un rotulador. Ana se gastó en la misma papelería 2€ por el mismo rotulador y un bolígrafo igual que el de José. ¿Cuánto cuesta un bolígrafo?
¿Cuánto cuesta el rotulador?

Anexo 3. Cuestionario de reflexión para el alumnado

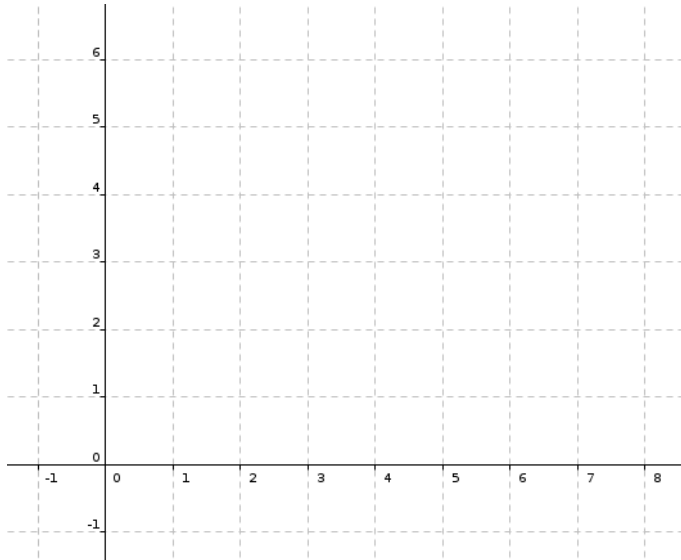
1. ¿Has leído varias veces el enunciado?
2. ¿Has comprendido el enunciado?
3. ¿Puedes explicar el problema con tus propias palabras?
4. ¿Has tenido claro el objetivo final del problema?
5. ¿Has identificado todos los datos previamente?
6. ¿Has descompuesto el problema en problemas más pequeños.
7. ¿Has explicado las operaciones matemáticas?
8. ¿Has diferenciado las operaciones?
9. ¿Coinciden los pasos realizados con el planteamiento inicial?
10. ¿Has redactado la solución?
11. ¿Has verificado cada una de las operaciones?
12. ¿Has analizado si existen varias alternativas? ¿Las has considerado?
13. ¿Has comprobado la solución?

Anexo 4. Prueba de Geometría para estudiantes de Educación Primaria

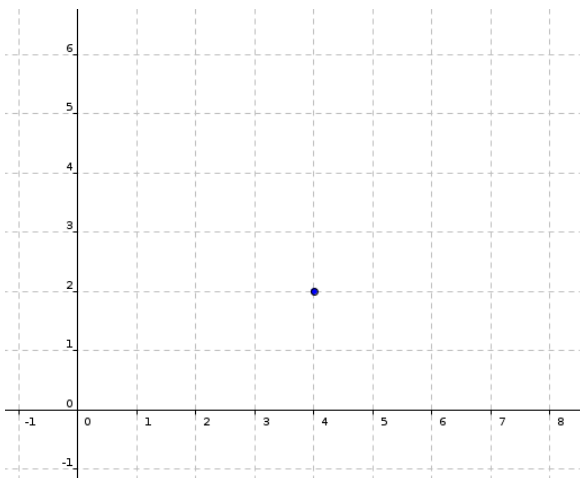
Nombre:

CEIP:

1. Indica cuál es el eje X y cuál es el eje Y:



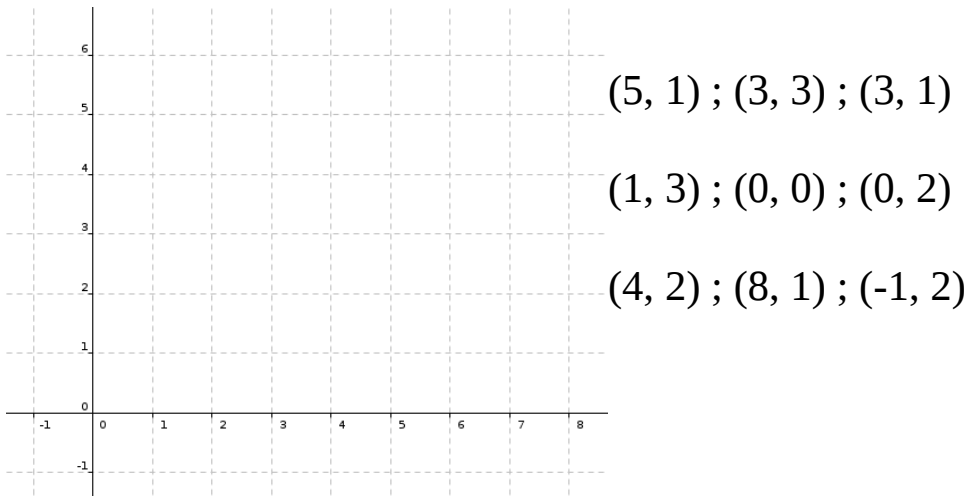
2. ¿Cuál es el punto señalado?



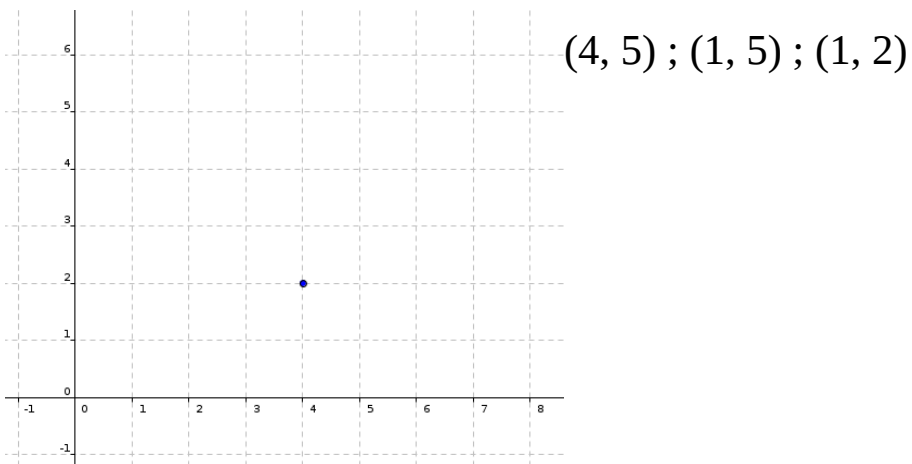
Punto = (,)

Contribución del Pensamiento Computacional con Scratch al proceso de enseñanza y aprendizaje de las Matemáticas

3. Dibuja en el plano los siguientes puntos:

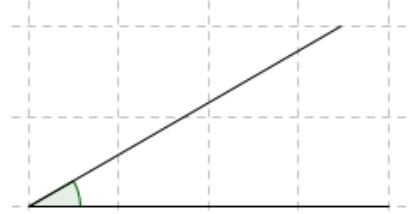


4. Dibuja los siguientes puntos y une los 4 con una línea.

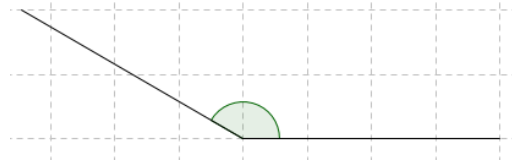


5. Indica cuál de los siguientes ángulos corresponde con cada una de los siguientes valores:

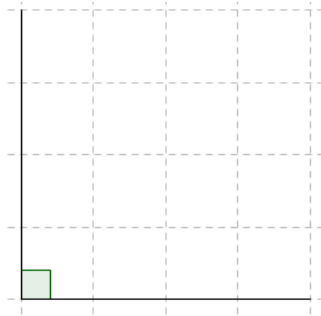
Ángulo recto



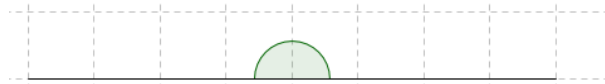
Ángulo 30°



Ángulo 180°



Ángulo 150°

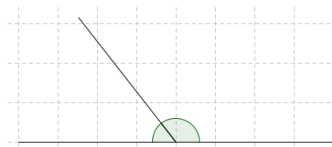


6. Completa las siguientes definiciones uniendo los recuadros:

SUMAN

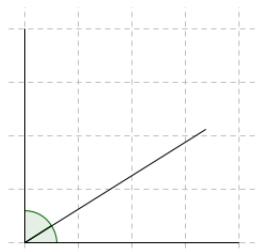
Ángulos
Complementarios

180°

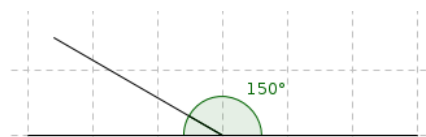
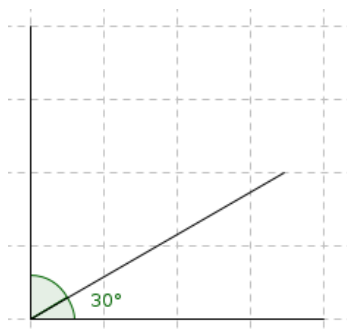


Ángulos
Suplementarios

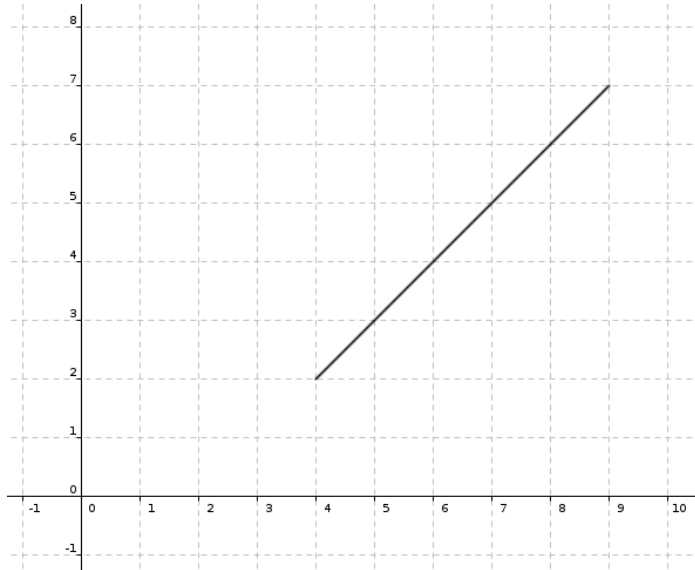
90°



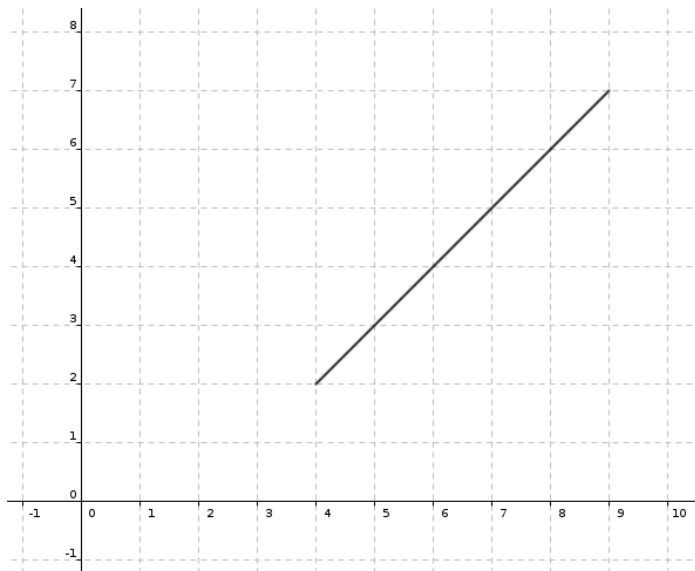
7. Indica el valor del ángulo que falta en cada una de las siguientes figuras:






8. Dibuja una recta perpendicular a la que aparece representada. ¿Qué ángulo forman?



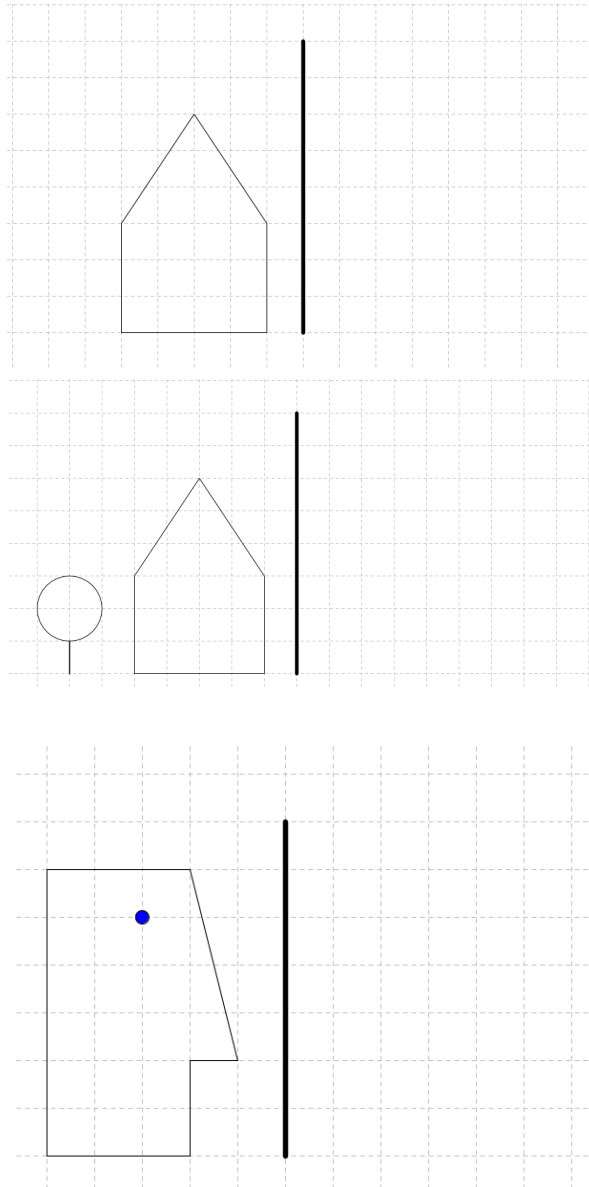
9. Dibuja una recta paralela a la que aparece representada. ¿Cuántas más puedes dibujar?



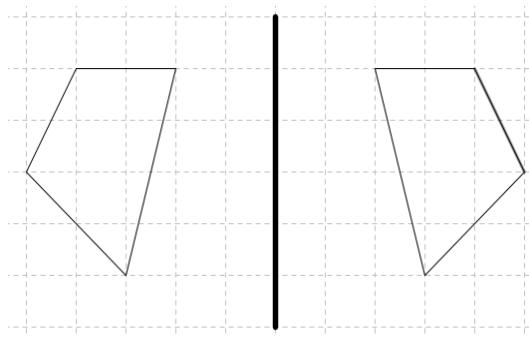
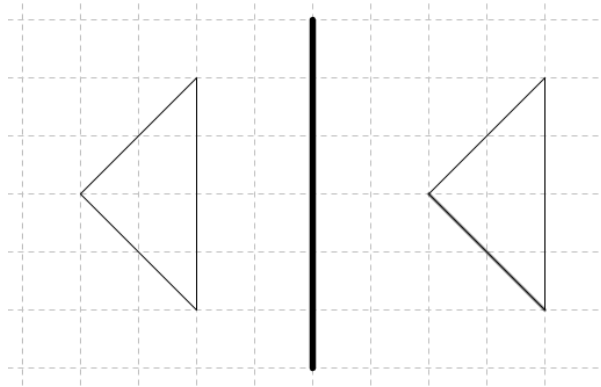
10. Indica el número de lados y dibuja las siguientes figuras:

		<u>Dibujo</u>
Cuadrado lados	
Triángulo lados	
Hexágono lados	

11. Dibuja una figura simétrica utilizando el eje



12. Indica si son simétricas las siguientes figuras:



Anexo 5. Guión para la discusión con los estudiantes del proyecto de investigación

-¿Qué te ha parecido Scratch?

-¿Cuáles fueron los primeros trabajos? ¿Te gustó poder animarlo para que se moviese y cambiase de tamaño, color, forma?

-¿Te resulta divertido para usarlo en clase? ¿crees que más profesores podría usarlo contigo? Por ejemplo, en inglés o ciencias naturales

-¿Cómo han sido las clases con Scratch? ¿Cómo habéis trabajado? Grupo o individual

-¿Os habéis ayudado entre compañeros? ¿Quién ha hecho los proyectos más divertidos?

-¿Se incluye algún cambio en el trabajo de aula? Cambio metodológico, cambio en la participación del alumnado, cambio en el papel del profesorado participantes

-Hay dos bloques de actividades, las de introducción y las de geometría. ¿Te han gustado las segundas? ¿Crees que son útiles para el tema?

-Has visitado la página web de Scratch? Hay millones de proyectos de estudiantes como tú y puedes usarlos para reinventarlos o simplemente coger partes que te gusten.

-¿Es accesible para todo el alumnado?

-¿Conocías Scratch antes de usarlo en clase?

-¿Le has contado a tus papás lo que haces con Scratch?

Anexo 6. Guión para la discusión con los estudiantes del proyecto de investigación

-¿Qué expectativas tenéis de este proyecto?

-¿Cuáles son los posibles inconvenientes que podéis tener en el desarrollo del trabajo?

-¿Algo que os haga falta algo más para desarrollar el trabajo?

-Pero a pesar de ello, ¿creéis que es interesante realizar este tipo de proyectos de innovación?

¿Creéis que existen algún tipo de condicionantes clave para el correcto desarrollo?

-¿Se incluye algún cambio en el trabajo de aula? Cambio metodológico, cambio en la participación del alumnado, cambio en el papel del profesorado participantes.

-¿Los aspectos que cambian son positivos? Creéis que es algo necesario para los estudiantes? Es importante trabajarlo desde una asignatura en concreto como matemáticas o de manera transversal en cualquier área.

-¿Se puede implementar adecuadamente una atención a la diversidad?

-¿Es accesible para todo el alumnado?

-Es importante para el profesorado este tipo de prácticas?

-¿Es necesario informar al claustro y a las familias?

Anexo 7. Cuestionario para estudiantes en el Grado de Educación

Primaria

1. En general, ¿crees que es positivo trabajar el pensamiento computacional (PC) con el alumnado de Educación Primaria?
2. ¿Trabajar el PC con Scratch es positivo para el aprendizaje de las Matemáticas?
3. ¿Trabajar el PC es interesante para desarrollar el pensamiento lógico?
4. ¿Crees que utilizar recursos como Scratch es positivo para tu formación inicial como docente?
5. Aprender a crear proyectos con Scratch, ¿es fácil o te ha resultado complicado?
6. ¿Crees que utilizar recursos como Scratch es motivador para estudiantes de último ciclo de Ed. Primaria.
7. El entorno de trabajo de Scratch es fácil de utilizar para estudiantes de Ed. Primaria.
8. Las actividades desarrolladas en el curso han sido adecuadas para conocer los aspectos básicos del programa.
9. Las actividades desarrolladas en el curso son adecuadas para incluirlas en la asignatura de Matemáticas.
10. Las actividades desarrolladas en el curso son adecuadas para trabajar conceptos computacionales (direcciones, bucles, condicionales, funciones).
11. ¿Cómo valoras el trabajo realizado con Scratch?
12. ¿Te resultaría interesante aprender más sobre pensamiento computacional para incluirlo en tu futura práctica docente?
13. Deja algún comentario general sobre el trabajo realizado