

# Impacto de una formación intensiva en programación en el desarrollo del Pensamiento Computacional en futuros/as maestros/as

## Impact of intensive programming training on the development of Computational Thinking in prospective teachers



- ID Juan González-Martínez - *Universitat de Girona (España)*
- ID Marta Peracaula-Bosch - *Universitat de Girona (España)*
- ID Rafel Meyerhofer-Parra - *Universitat de Girona (España)*

### RESUMEN

Formar a los profesores en formación en su propio pensamiento computacional (PC) es fundamental para construir con ellos el discurso de la didáctica del PC y su inclusión en el aula con niños de educación infantil y primaria. Esta investigación plantea posibles soluciones a ese reto y analiza los resultados de una intervención llevada a cabo con 71 alumnos de dos cohortes diferentes de 2.º curso de los grados de magisterio. Dicha intervención se fundamenta en la práctica intensiva de programación por bloques visuales en un proyecto de Scratch durante una primera parte de una asignatura semestral. Analizados los niveles previo y posterior de PC (estrategia pre-experimental) por medio de una prueba estandarizada para su medición (Test PC), se confirma que la experiencia formativa intensiva permite a todos/as los/as futuros/as maestros/as alcanzar un nivel suficiente de PC, independientemente de su experiencia previa en programación y de su nivel inicial de PC (todos acaban aprendiendo, sea aumentando su nivel de PC, sea mejorando su eficiencia). Por otro lado, medidos los resultados de aprendizaje en el proyecto de Scratch, vemos una relación clara entre un nivel alto de PC resultante (POST) y un buen desempeño en la tarea de aprendizaje de programación por bloques, lo que evidencia que el proyecto de Scratch ayuda a desarrollar el PC de los/as futuros/as maestros/as.

**Palabras clave:** pensamiento computacional; lenguaje de programación; formación de profesores; evaluación.

### ABSTRACT

Training pre-service teachers in their own Computational Thinking (CT) is essential to build with them the discourse of CT didactics and its inclusion in the classroom with children in early childhood and primary education. This research proposes possible solutions to this challenge and analyses the results of an intervention carried out with 71 students from two different cohorts of 2nd year teacher training students. This intervention is based on the intensive practice of coding using visual blocks through a Scratch project during the first part of a semester course. After analyzing the previous and subsequent levels of CT (pre-experimental strategy) by means of a standardized test for its measurement (CTt), it is confirmed that the intensive training experience allows all future teachers to reach a sufficient level of CT, regardless of their previous experience in programming and their initial level of CT (they all end up learning, either by increasing their level of CT or by improving their efficiency). On the other hand, measuring the learning outcomes in the Scratch project, we see a clear relationship between a high resulting CTt level (POST) and a good performance in the block programming learning task, which is evidence that the Scratch project helps to develop the future teachers' CT.

**Keywords:** computational thinking; programming language; teacher training; evaluation.

## INTRODUCCIÓN

No cabe duda de la popularidad que el concepto de Pensamiento Computacional (PC) ha adquirido en el ámbito de la educación en los últimos años. En vista de ello, es necesario que se profundice en el propio concepto de PC y en el análisis de su práctica educativa desde la perspectiva investigadora.

Al hablar de PC, sin duda debemos prestar atención al trabajo de Seymour Papert (1980), quien, al final de su clásico *Mindstorms*, menciona el pensamiento computacional, y se refiere a los entornos de aprendizaje en los que el ordenador es “un objeto con el que pensar” (p. 182). Tradicionalmente, sin embargo, se considera que el concepto en sí nace con la definición seminal de Wing (2006), un cuarto de siglo después. Aunque se trata solamente de una aproximación conceptual, no cabe duda de que ha funcionado en gran medida como definición que señala elementos importantes del PC: “Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science.” (Wing, 2006, p. 33). Sin embargo, es en Wing (2014) donde encontramos una definición completa: “Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer -human or machine- can effectively carry out”. Se trata de una definición con ideas clave muy importantes: un proceso mental, la formulación de un problema, la expresión de sus soluciones, la necesidad de un computador (humano o máquina); a lo que se añaden interesantes reflexiones sobre la abstracción o la importancia del PC fuera de los contextos concretos de la informática.

A partir de aquí, muchos autores consideran que el PC es una competencia básica para el siglo XXI (Angeli et al., 2016; European Commission/EACEA/Eurydice, 2012; Fluck et al., 2016). Nos permite desarrollar un procedimiento eficaz de resolución y planteamiento de problemas (Fluck et al., 2016) en cualquier dimensión del mundo en el que vivimos, y en consecuencia nos ayuda a comprenderlo y a vivir en él (Furber, 2012). En ese sentido, Grover y Pea (2013) destacan que la computación es una actividad humana; la abstracción ayuda a centrarse en lo esencial y a descuidar lo superfluo; y, en consecuencia, el PC permite la creación de conocimiento, la creatividad y la innovación, en todos los sentidos y a todos los niveles. Se señalan también elementos actitudinales como la seguridad, la perseverancia o la colaboración (Bocconi et al., 2016b). Por estos motivos, muchos sistemas educativos han decidido incluirla en el plan de estudios: Canadá, Reino Unido, Finlandia y Australia son ejemplos de ello (Acevedo Borrega, 2016; Bocconi et al., 2016a).

Con todo, lo que nos interesa ahora es centrarnos en el profesorado que debe acompañar este proceso de aprendizaje. En este sentido, es generalmente aceptado que los profesores necesitan ser formados específicamente en PC; y, como consecuencia de esta asunción común, desde hace más de una década se ha puesto el

foco tanto en la propia formación del profesorado como en los modelos pedagógicos asociados a el PC (Morreale et al., 2012).

Según Butler y Leahy (2021), la falta de investigación sobre los procesos de formación en PC del profesorado futuro justifica que, de forma prioritaria, nos centremos en ella para mejorar la formación de los/as formadores/as y, en consecuencia, la formación de los/as niños/as. Para ello, es imprescindible una primera parte dedicada al desarrollo del PC antes de abordar las cuestiones didácticas; y, con base en lo que ya sabemos, pocos esfuerzos en esta dirección dan muchos resultados: tanto en la mejora del PC como en la mejora de las actitudes hacia ella (Bustillo y Garaizar, 2015; González-Martínez et al., 2018; Peracaula-Bosch et al., 2020). Partimos de la evidencia de una falta de base conceptual sobre PC en los/as maestros/as (Acevedo-Borrega et al., 2022; Morze et al., 2022), así como de la necesidad de focalizar, tanto en la formación inicial como en la permanente, en las destrezas técnicas y conceptuales específicas (Rich, Mason y O'Leary, 2021) para mejorar la propia actuación docente en contextos educativos formales (Collado-Sánchez et al., 2021). La investigación, de hecho, nos dice que la mejora en la formación específica no solo mejora la autopercepción, sino que esta genera una espiral perfecta: la mejora del propio PC permite a los docentes sentirse más seguros, tanto en su práctica del PC como en su práctica docente; y esta seguridad mejora efectivamente su competencia real (Rich, Larsen y Mason, 2021), incluso en experiencias formativas breves (Pala y Türker, 2021), que debe focalizar, para ser efectiva, en elementos como la edad o el diseño correcto de las experiencias (Li, 2021). Este hecho es sustentado en el modelo TPACK (Mishra y Koehler, 2006) que se basa en la necesaria superposición por parte de los formadores de tres tipos de conocimiento: del contenido, pedagógico y tecnológico. En este caso, el nivel adecuado y desarrollo del PC correspondería al conocimiento en contenido (*content knowledge*) y su carencia representaría una barrera en el proceso de enseñanza desplegado por el/la docente. En la revisión de literatura realizada por Mason y Rich (2019) por ejemplo, se analizan 21 estudios (12 relativos a maestros en formación y 9 a maestros en activo, con un total de 802 participantes) en que los resultados indican que factores como la auto-eficacia en PC percibida por los maestros aumentan con la formación que promueve su propio desarrollo y ello repercute en la comprensión del concepto y la capacidad de diseñar y evaluar experiencias de aprendizaje en PC. Estos resultados son corroborados posteriormente en Rich, Mason y O'Leary (2021) en un estudio con 127 maestros, que concuerda con el estudio de Lamprou y Repenning (2018), con 471 maestros en formación inicial. En él, la formación en PC, focalizando, por ejemplo, en los conceptos de abstracción, análisis y automatización, aumenta drásticamente la confianza de los/as profesores/as en su capacidad de educar en PC. Así, la percepción de autoeficacia en PC conlleva también la autoeficacia en la capacidad para formar en este. Por último, cabe destacar un reciente estudio con 245 maestros/as de 47 escuelas de Hong Kong (Kong et al., 2023) que recibieron formación en desarrollo del PC durante un curso escolar y que además fueron

seguidos en su implementación en las aulas, con un análisis directo de la relación entre la formación de los/as maestros/as, su actuación docente y el aprendizaje del alumnado. El estudio corrobora la necesidad de formación a maestros/as en PC y su desarrollo, validando los programas de formación basados en entornos de programación por bloques y mixtos y recomendando soporte continuo durante la docencia a través de plataformas, mentorías y repositorios de materiales. Y, en este sentido, recogemos una visión que nos parece muy acertada, propuesta por Estebanell et al. (2018), en la que señalan que, antes de enfrentarse a la cuestión de la didáctica del PC, el profesorado en formación debe consolidarse como usuario/a del PC y como usuario/a reflexivo/a (es decir, desarrollar su propia PC de forma reflexiva) antes de pasar a desarrollar su dimensión como docentes en PC o docentes reflexivos/as en PC, que sería la última etapa. Sabemos también que ofrecer modelos claros que susciten la reflexión puede ser provechoso como estrategia para compensar la falta de formación específica (Dobgenski y Garcia Silva, 2022). Sin embargo, la solución no es sencilla, y hay muchas cuestiones que aún deben aclararse. Por un lado, como en cualquier asignatura de didáctica disciplinar, es importante no detenerse en el desarrollo del conocimiento disciplinar, sino saltar a cuestiones didácticas; por tanto, el tiempo dedicado al conocimiento disciplinar debe ser muy limitado y orientado. Por otro lado, precisamente por esta razón, parece sensato pensar que, en términos de eficiencia, el esfuerzo que dediquemos al desarrollo del PC debería permitirnos aprender el uso de herramientas que luego podamos utilizar también en el enfoque didáctico y para la conceptualización de procesos reflexivos (Pérez-Marín et al., 2020).

Con todo, dedicar una primera parte intensiva de un curso o asignatura al desarrollo del PC en la formación del profesorado no es fácil, ni sus resultados pueden darse por descontados (el PC, por concepto, es complejo y su desarrollo no es inmediato). Es en ese contexto donde nace esta propuesta de investigación; en ella, nos planteamos, como objetivo general, analizar el desarrollo del PC a través de una actividad de programación por bloques intensiva en la formación universitaria de los/as estudiantes de magisterio, previamente a la formación en la didáctica del PC (que es aquello que, en efecto, tenemos como objetivo final de primer orden en la formación de maestros/as).

Por ello, al final de esta reflexión, llegamos a nuestra pregunta de investigación: ¿es posible desarrollar el PC de los/as futuros/as maestros/as por medio de una formación en Scratch, al comienzo de una asignatura, antes de pasar a los aspectos didácticos?

Y, a esta pregunta general, podemos añadir las siguientes, que nos ayudarán a profundizar en nuestro objeto de estudio:

- ¿Cómo influyen en los resultados los diferentes puntos de partida de los/as estudiantes en el desarrollo del PC?

- ¿Es posible relacionar los resultados de PC con algunas de las competencias y habilidades requeridas para la elaboración de un proyecto de programación con Scratch?
- El test de PC que usaremos mide comprensión de los lenguajes y la lógica computacional en retos de configuración de menos a más compleja. A partir de esto, ¿haber realizado un programa complejo, con soluciones ingeniosas y optimizadas y que funcione se relaciona con los resultados de PC?

## MÉTODOS E INSTRUMENTOS

Para esta investigación, decidimos trabajar con dos grupos de estudio compuestos por alumnos/as de segundo año de los grados de Educación Infantil y de Educación Primaria de la Universidad de Girona (algunos/as de ellos/as de la doble titulación de EI y EP). El estudio se realizó en el marco de la asignatura Pensamiento Computacional y Programación de la mención en Tecnologías de la Información y la Comunicación impartida durante el segundo semestre de los años académicos 2020/21 y 2021/22. El total de participantes fue de 71, de los cuales 36 cursaron la asignatura en la primera cohorte y 35 en la segunda.

En cuanto a sus conocimientos previos sobre programación y robótica, los/as estudiantes han recibido durante el primer año de los estudios, en el marco de una asignatura obligatoria, una introducción al PC con algunos ejercicios prácticos (demostraciones de robótica educativa e introducción a la programación con Scratch). Además, en el test se pregunta a los/as estudiantes si tienen o no experiencia previa en actividades de programación, refiriéndonos a la posible formación que hayan podido recibir en las etapas previas a su ingreso en la Universidad, tanto en el ámbito formal como fuera de él.

En relación con el diseño de la investigación y con su encaje con la experiencia de aprendizaje (orientada al desarrollo del PC), se planteó un diseño pre-experimental por lo que respecta a la aplicación de uno de los instrumentos (el Test PC, que detallaremos más adelante) y una recogida de información ex post facto por lo que respecta al análisis de parte de las pruebas de evaluación entregadas por los participantes al final de la experiencia (proyecto de programación con el lenguaje visual por bloques Scratch, <https://scratch.mit.edu/>). En cuanto a la estrategia pre-experimental, el Test PC se aplicó al inicio del curso y al final de la primera parte (febrero a abril), dedicada a la implementación de diferentes actividades para desarrollar el PC de los/as estudiantes, antes de abordar la segunda parte del semestre, dedicada al desarrollo de competencias didácticas para enseñar PC a niños de infantil y primaria. En versión resumida, esta estrategia formativa y de investigación consideraba la siguiente secuencia:

- Semanas 1 a 4: actividades de introducción al PC
  - Actividad de PC desenchufada
  - Test PC (PRE)
  - Análisis teórico y reflexión
  - Taller de programación por bloques (TurtleStitch, <https://www.turtlestitch.org/>) en 2021 y MicroBlocks (<https://microblocks.fun/>) en 2022
- Semanas 5 a 9: tutoriales y ejercicios Scratch partiendo de un nivel básico (programación del movimiento de un elemento o personaje) e incrementando paulatinamente la complejidad: interacción con personajes y entre ellos, sensores, condicionales variables, operadores, funciones y modularización, y uso del editor gráfico para la creación y modificación de personajes y escenarios.
- Semanas 5 a 10: desarrollo en trabajo individual y autónomo de un proyecto Scratch que simule un fenómeno científico de libre elección (se ponen como ejemplos entre otros: ciclo del agua, fases de la luna, desarrollo de una planta, caída de un meteorito en presencia de unos dinosaurios que nada sospechan, etc.). El objetivo es no solo aprender a programar sino también poner de relieve que el hecho de tener que narrar visualmente en un lenguaje preciso propicia la comprensión profunda del fenómeno que se requiere explicar. En cuanto a la programación, se indica a los estudiantes que se valorará el uso de diversos elementos (personajes) programados, cambios de escenarios, y elementos característicos de los lenguajes computacionales, como son condicionales, iteración en bucles, operadores, variables y cierta interactividad con el usuario. Durante el desarrollo del proyecto, los estudiantes reciben los tutoriales indicados en el punto anterior, tutorías y pueden consultar y participar en un foro de dudas, respuestas y descubrimientos para la comunicación y para el intercambio de todo el grupo y el profesorado.
- Semana 11: test PC (POST)
- Semanas 11 a 14: fase de diseño: el curso continúa durante unas semanas centrado en la didáctica del PC (diseño e implementación de actividades que permiten su desarrollo), cuyo análisis escapa al objetivo de esta reflexión.

En cuanto a los instrumentos, como decíamos, en primer lugar, decidimos utilizar un instrumento ya existente, el test de Pensamiento Computacional (CTt, al que nos referiremos como Test PC para clarificar el discurso), creado y validado por Román-González (2016) y Román-González et al. (2018). Para este estudio hemos utilizado la adaptación del Test PC original TPC-RA+B, diseñada específicamente para participantes mayores de 14 años (Moreno-Leon et al., 2022). El Test PC mide, según sus propios creadores, el PC entendido del siguiente modo: “CT involves the ability to formulate and solve problems by relying on the fundamental concepts of computing, and using logic-syntax of programming languages: basic sequences,

loops, iteration, conditionals, functions and variables” (Román-González et al., 2017, p. 681). En consecuencia, mide la comprensión de lenguajes computacionales y lógica computacional en configuraciones de menos a más complejas. Durante su resolución, los estudiantes no elaboran algoritmos ni programas propios, sino que descifran algoritmos propuestos, en general respondiendo cuál de diversas opciones resuelve el reto de la pregunta. Los algoritmos se presentan en diferentes lenguajes y simbologías.

En segundo lugar, se analizaron las tareas entregadas para la evaluación de los aprendizajes al finalizar esta primera parte de esta materia. Una gran parte de la formación de los estudiantes se dedica al aprendizaje de la programación visual por bloques a través de distintas actividades y tutoriales, entre las cuales tiene gran protagonismo, por la complejidad y el tiempo de dedicación, la realización de un proyecto Scratch que simule un fenómeno natural o científico.

Para analizar estas tareas, y en relación con las preguntas de investigación específicas de este tópico, del proyecto Scratch que ha desarrollado cada alumno durante la formación y del informe que ha realizado sobre éste, hemos extraído los factores que pueden estar relacionados con su nivel de PC según los parámetros medidos por el Test PC arriba mencionados, de manera que los hemos clasificado en las categorías mostradas en la Tabla 1. En cada una de estas categorías se podía obtener una puntuación máxima de 4 puntos teniendo en cuenta los elementos de evaluación; por ello, se podía obtener una puntuación total máxima de 12 puntos.

**Tabla 1**  
*Elementos evaluados en los proyectos Scratch*

Categorías	Elementos de evaluación	Puntuación máxima
Funcionamiento del programa y comprensión de éste	Funcionamiento correcto del programa.	4
	Uso de condiciones iniciales.	
	Partes del algoritmo comentadas en que se refleja la comprensión de las acciones.	
Complejidad	Descripción de los problemas surgidos y comprensión de las soluciones encontradas.	4
	Diversos personajes/escenarios programados.	
	Cambios de aspecto.	
	Transiciones.	
	Interactividad.	

Categorías	Elementos de evaluación	Puntuación máxima
Optimización e ingenio	Variedad de elementos de lenguaje computacional que ayuden y simplifiquen la programación: bucles, condicionales, sensores, variables, operadores. Soluciones elegantes e ingeniosas.	4

## RESULTADOS

En este apartado expondremos y analizaremos los resultados obtenidos, abordados en dos bloques: el primero de ellos se centrará en los resultados del Test PC, mientras que el segundo de ellos analizará los resultados obtenidos en el Test PC (pre y post) y su relación con los obtenidos en el proyecto de Scratch.

### Análisis comparativo de los resultados del Test PC previos y posteriores a la formación

En la Tabla 2 se detallan los coeficientes de fiabilidad, que se consideran aceptables para los rangos comúnmente aceptados en el ámbito educativo.

**Tabla 2**  
*Niveles de confiabilidad*

Escala	Cronbach's Alpha
Pre-test	0,781
Post-test	0,835

En la Figura 1a se representan, en la misma escala, los histogramas o diagramas de frecuencias de la puntuación de los participantes en los Test PC PRE y POST formación. La prueba de normalidad de Kolmogorov-Smirnov para una muestra nos confirma que ambos siguen una distribución normal o gaussiana, lo que nos permitirá utilizar las medias y errores estándar como descripción válida de los datos y utilizar la estadística paramétrica para realizar los tests de comparación de los estadísticos (Rubio Hurtado y Berlanga Silvente, 2012; Simpson, 2015). Podemos observar que en la distribución del test POST hay un desplazamiento general de las puntuaciones hacia valores superiores a las del test PRE, de manera que si nos centramos en la puntuación 21 (frontera entre los intervalos que contienen la media PRE y POST, marcada por la línea discontinua azul), podemos ver que antes de la

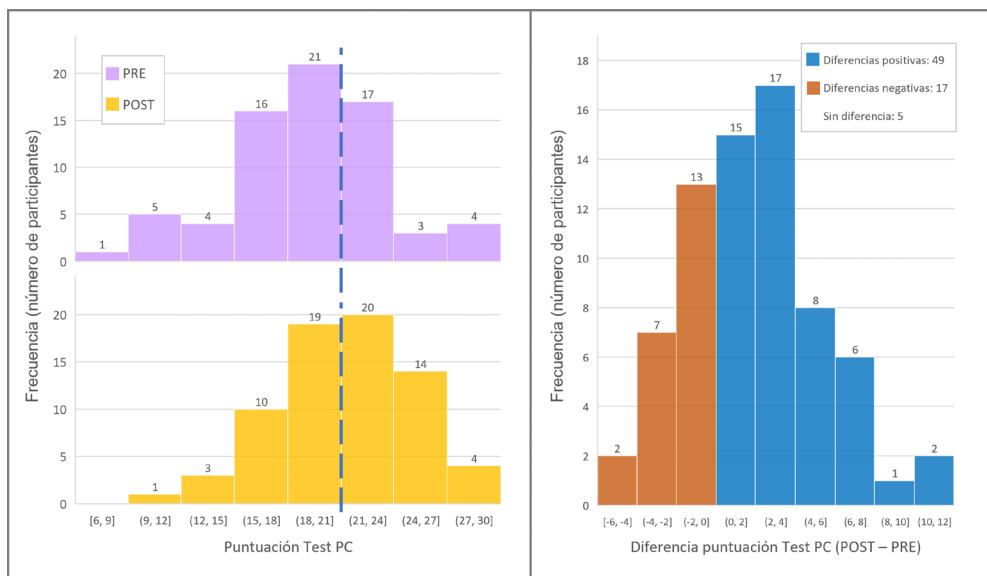


formación tenemos 47 valores por debajo o igual a ella y 24 por encima y después de la formación estos valores pasan a ser 33 y 38 respectivamente. En la Figura 1b se plasma la distribución de la diferencia de puntuación de cada participante entre los test POST y PRE. Además del hecho de que 49 de las diferencias son positivas, 17 negativas y en 5 casos no hay diferencia de puntuación, cabe destacar la asimetría respecto a la diferencia 0 plasmada a lo largo de la distribución, que denota que las diferencias positivas son significativamente mayores. Un ejemplo de ello es el hecho de que la diferencia negativa máxima sea de 5 puntos mientras que la diferencia positiva máxima es de 12 puntos. Así pues, los gráficos de la Figura 1 nos muestran que entre los tests PRE y el POST se ha producido, de media, un desarrollo del PC según los parámetros medidos por el Test PC.

**Figura 1**

a) *Histograma de frecuencias de la puntuación obtenida por los participantes en el test pre-formación (superior) y en el test post-formación (inferior)*

b) *Histograma de frecuencias de la diferencia en puntuación entre los dos tests obtenida por cada participante*

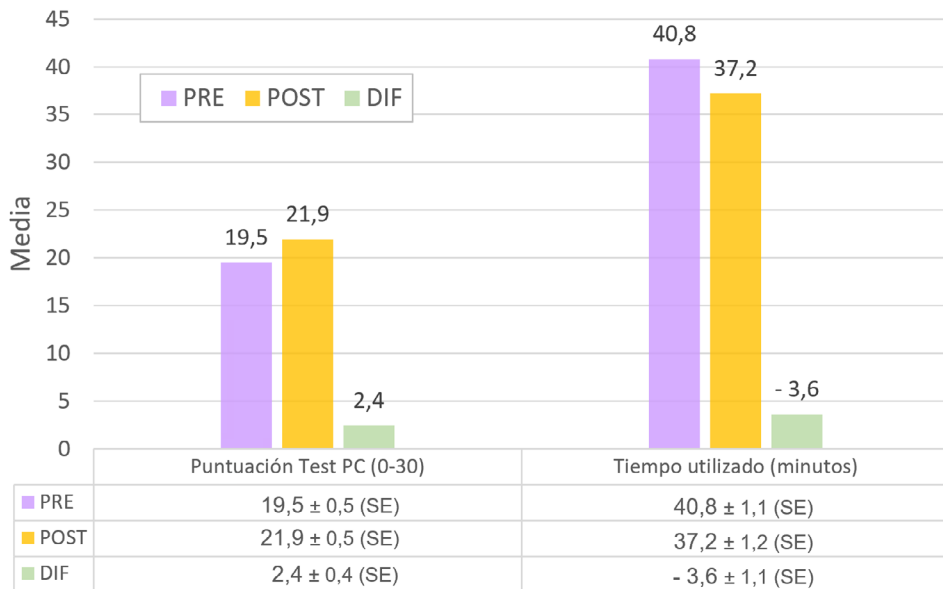


Este desarrollo del PC se muestra cuantificado en la Figura 2, donde se indican las medias de la puntuación del Test PC y del tiempo que necesitaron los/as estudiantes para completarlo en las pruebas PRE y POST con sus errores estándar (SE) o error típico de la media. Como podemos ver, la media del resultado del Test PC ha pasado

de 19,5 a 21,9 en una escala de 30 puntos y se ha producido un decremento en el tiempo necesario en su resolución de 40,8 a 37,2 minutos. Las diferencias en ambos casos son estadísticamente significativas con un valor de  $p < 0,001$ : es decir, podemos afirmar que la probabilidad de que la diferencia no sea debida al azar es mayor que 99,9 %. La significancia práctica medida según el efecto del tamaño de la muestra de la  $d$  de Cohen es 0,6 en el caso de las diferencias en puntuación y 0,4 en el caso de las diferencias en tiempo.

**Figura 2**

Valores medios del resultado del test PC y del tiempo necesario para resolverlo en la población total (71 participantes entre 2021 y 2022) aplicado con anterioridad a la formación (PRE) y con posterioridad a ésta (POST) y su comparativa (DIF). Se indican también sus errores estándar (SE)



A continuación, analizamos los datos agrupando los/as participantes según si habían tenido experiencia previa en actividades de programación o robótica con anterioridad a su ingreso a la universidad, en cualquier etapa educativa tanto en el ámbito formal como informal. De los/as 71 participantes, 33 declararon no tener experiencia previa y 38 tenerla. En la Tabla 3 se exponen, para cada uno de estos dos grupos, las medias en la puntuación obtenida en los Test PC (PRE y POST) y los tiempos utilizados para resolverlos. Asimismo, constan las diferencias obtenidas

comparando los resultados de los dos Tests PC (PRE y POST) en un mismo grupo o los resultados de un solo Test PC (PRE o POST) entre grupos. Para comprobar la significación estadística de estas diferencias se ha utilizado, en el primer caso, la prueba no paramétrica para muestras emparejadas Wilcoxon Signed Rank, y, en el segundo caso, la prueba no paramétrica para muestras independientes Mann-Whitney U. Hemos considerado que las diferencias son significativas (no debidas al azar) para valores de  $p < 0,05$ .

**Tabla 3**

*Valores medios del resultado del test PC (PRE y POST) y del tiempo necesario para resolverlo según experiencia previa con sus errores estándar (SE). Constan también las diferencias entre tests de un mismo grupo y entre grupos de un mismo test, su significación estadística (p) y su significación práctica según el tamaño de la muestra (d de Cohen)*

	Puntuación media Test PC			Tiempo medio utilizado (minutos)		
	PRE	POST	DIF (POST-PRE)	PRE	POST	DIF (POST-PRE)
Sin experiencia (33 participantes)	18,0 ±0,8(SE)	21,1 ±0,6(SE)	3,1 ±0,6(SE) $p < 0,001$ , $d=0,8$	42,0 ±1,6(SE)	39,6 ±1,8(SE)	-2,4 ±1,8(SE) $p = 0,1$ , $d=0,2$
Con experiencia (38 participantes)	20,8 ±0,6(SE)	22,7 ±0,7(SE)	1,9 ±0,5(SE) $p < 0,001$ , $d=0,6$	39,8 ±1,5(SE)	35,2 ±1,6(SE)	-4,6 ±1,5(SE) $p = 0,001$ , $d=0,5$
Diferencia (con exp. – sin exp.)	2,8 ±1,0(SE) $p = 0,02$ , $d=0,6$	1,6 ±1,0(SE) $p = 0,08$ $d=0,4$		-2,2 ± 2(SE) $p = 0,3$ $d=0,2$	-4,4 ±1,5(SE) $p = 0,04$ $d=0,4$	

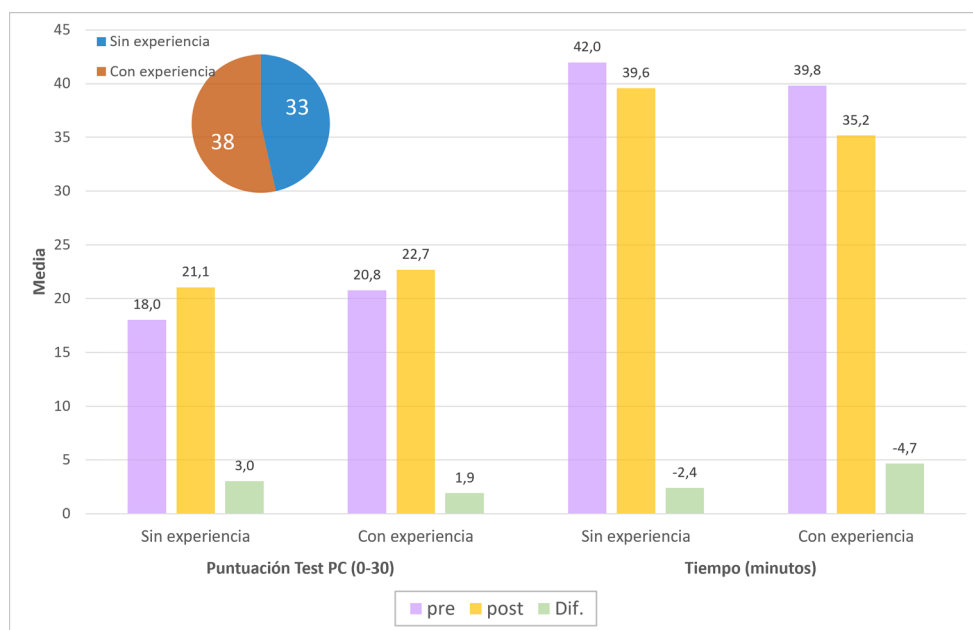
Si nos centramos en la puntuación media del Test PC, y comparamos entre los dos grupos, podemos ver que previamente a la formación los/as estudiantes con experiencia previa obtienen un resultado de 20,8 puntos y este es 2,8 puntos mayor a la puntuación media de los/as estudiantes sin experiencia. Esta diferencia es significativa, con  $p=0,02$ . Después de realizar la formación, la diferencia en la puntuación media del Test PC disminuye y no se puede considerar significativa ( $p=0,08$ ). Cuando comparamos los resultados PRE y POST de cada grupo podemos ver (Figura 3) que ambos grupos han incrementado la puntuación media del Test PC, y es más relevante este incremento en el grupo que no tenía experiencia previa.

Por ello, en la medida en que en los dos grupos se ha producido un desarrollo del PC, aunque desigual, la formación ha podido en parte compensar la falta de experiencia previa.

Cabe destacar que parte de la habilidad en la resolución del Test PC se plasma también en el tiempo de su resolución, y en el caso de los/as estudiantes con experiencia previa la disminución del tiempo es relevante (4,6 minutos,  $p=0,001$ ). Por tanto, mientras que para los/as estudiantes sin experiencia el incremento en el desarrollo de PC se muestra de forma más evidente en los resultados del Test PC (que aumentan más), los/as estudiantes con experiencia previa lo muestran en el tiempo empleado (que disminuye más). Esto puede indicar que, según el estadio de desarrollo del PC, la imposibilidad de superar ciertas puntuaciones en el test, que ya inicialmente fueron altas, revierte en el tiempo empleado en ello.

### Figura 3

Valores medios del resultado del test PC y del tiempo necesario para resolverlo según experiencia previa aplicado con anterioridad a la formación (PRE) y con posterioridad a ésta (POST) y su comparativa (DIF)

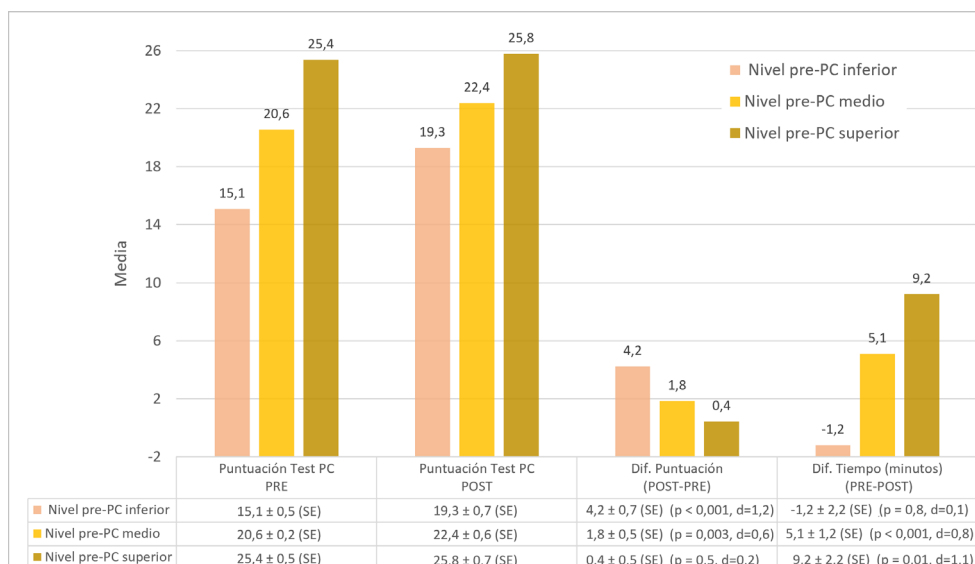


Para seguir analizando este aspecto, hemos agrupado las variables de toda la población en tres muestras, tomando como referencia la puntuación obtenida en el test PRE, previo a la formación. Así, a partir de los terciles calculados en la

distribución de la puntuación PRE, agrupamos los/as participantes en los niveles “pre-PC inferior” (26 participantes con una puntuación media de 15,1), “pre-PC medio” (31 participantes con una puntuación media de 20,6) y “pre-PC superior” (14 participantes con una puntuación media de 25,4) y calculamos para cada uno de ellos los resultados obtenidos en el test POST (posterior a la formación). En la Figura 4 podemos ver, para cada grupo, las medias de las puntuaciones, las diferencias en puntuación media entre el test POST y PRE y las diferencias de tiempo medio utilizado para realizar el test.

**Figura 4**

*Valores medios del resultado del test PC y del tiempo necesario para resolverlo (PRE y POST) agrupando la población según resultado inicial*



Observamos que, en el grupo con nivel inferior, se ha producido un incremento significativo en la puntuación del Test PC (4,2 puntos; p=0,001); en el grupo con nivel medio también se ha producido un incremento significativo, aunque este ha sido menor (1,8 puntos; p=0,003); y en el grupo con nivel superior el incremento en la puntuación del Test PC no es significativo. Sin embargo, si nos centramos en el tiempo medio necesario para la resolución de la prueba, el efecto es diferente: en el nivel inferior se ha mantenido sin cambios significativos y en los niveles medio y superior este ha disminuido significativamente, de manera que el grupo con nivel medio ha necesitado una media de 5,1 minutos menos que en el test PRE para resolver el test POST y el grupo con nivel superior lo ha hecho en una media de tiempo de 9,2

minutos menos. Por tanto, los resultados muestran que la formación ha ayudado a los tres grupos en el desarrollo del PC, y ha tenido cierto efecto igualatorio (el grupo con menor puntuación inicial ha mostrado mayor incremento), y ha mejorado la agilidad en su uso en aquellos/as estudiantes que ya partían de un nivel elevado, que han necesitado menos tiempo para realizarlo. En el análisis de la correlación entre las puntuaciones en Test PC PRE y POST de toda la muestra se obtiene una correlación positiva, moderadamente alta y estadísticamente significativa de valor  $r=0,67$ . Ello se puede interpretar en términos de fiabilidad y estabilidad del test. El hecho de que no se alcance la convergencia total se puede explicar por la diferencia en el incremento del aprendizaje según el nivel de partida hallado en el análisis de las muestras por niveles.

### Relación entre la programación con Scratch y los resultados del Test PC

Queremos analizar ahora si hay relación entre los resultados de la Programación Scratch según las categorías expuestas en la Tabla 1 y los resultados del Test PC, atendiendo fundamentalmente a dos focos de interés: en primer lugar, cómo influye el punto de partida del Test PC en los resultados de la Programación Scratch; en segundo, cómo se plasma en el Test PC final el nivel de desarrollo de la Programación Scratch

Como marco de referencia, en cuanto a la puntuación total en la Programación Scratch según las categorías de la Tabla 1 obtenida por los/as 71 participantes, la puntuación mínima es de 4 puntos y la máxima de 12, el valor medio es de  $9,3 \pm 0,2$  (SE) puntos, la mediana se encuentra en los 10 puntos y la moda en 11. La evaluación ha sido realizada por un único investigador experto en programación Scratch, en un doble ciclo (ponderación y revisión), para garantizar la homogeneidad del criterio aplicado.

Para analizar si el punto de partida indicado por el Test PC previo a la formación puede tener relación con el resultado del proyecto Scratch, hemos calculado los valores medios de la puntuación agrupando la población según las 3 muestras “Nivel pre-PC inferior”, “Nivel pre-PC medio” y “Nivel pre-PC superior” que se habían utilizado en la comparativa entre los Test PC PRE y POST, y hemos obtenido los resultados mostrados en la Tabla 4. Como podemos ver, no hay variación significativa en los resultados relacionados con el PC en el proyecto Scratch entre los/as participantes que partían del nivel medio respecto de los/as que partían del nivel inferior. Sí hay cierta diferencia respecto a los demás niveles en los resultados obtenidos por aquellos/as participantes que partían de un nivel superior, que obtienen también mejor resultado en la programación. Este resultado puede apoyar, por un lado, el aspecto compensatorio de la formación respecto al nivel de PC según los parámetros del Test PC, especialmente entre los grupos de nivel PRE PC inferior y medio; y, por el otro, el hecho de que los elementos de programación identificados como

característicos de los parámetros que mide el Test PC son aplicados a sus proyectos Scratch por los/as estudiantes que partían de un nivel superior.

**Tabla 4**

*Relación entre los valores del Test PC (pre) y el proyecto Scratch*

Nivel Test PC PRE	PRE-PC Inferior	PRE-PC medio	PRE-PC superior
Puntuación PC en la Programación Scratch	8,8 ± 0,4(SE)	9,1 ± 0,4(SE)	10,9 ± 0,2(SE)
Diferencias	Medio-Inferior: 0,3 ± 0,5 (SE) p = 0,8	Superior-Medio: 1,8 ± 0,6 (SE) p=0,02	Superior-Inferior: 2,1 ± 0,6 (SE) p = 0,006

En este punto, podemos preguntarnos si haber creado un programa complejo (utilizando un lenguaje computacional, usando ingenio y optimización y que funcione) guarda relación con los resultados del test PC. Para analizar esta relación entre los resultados de la programación Scratch y los resultados del Test PC posterior a la creación del proyecto, hemos agrupado los/as participantes según los niveles de programación inferior, programación media, y programación superior tomando como referencia los terciles de la distribución de la puntuación de la programación. De esta manera, en el grupo del nivel de programación Inferior (rango de puntuación de 4 a 8 puntos) hay 20 participantes; en el nivel Medio (rango de puntuación de 9 y 10 puntos) hay 25 participantes; y en el nivel Superior (rango de puntuación de 11 y 12 puntos) hay 26 participantes. En la Tabla 5 se exponen la puntuación media del total de la programación para cada uno de estos grupos y sus puntuaciones en el Test PC POST. Se muestran también las diferencias en los resultados del Test PC entre los tres niveles con sus significancias estadísticas.

**Tabla 5**

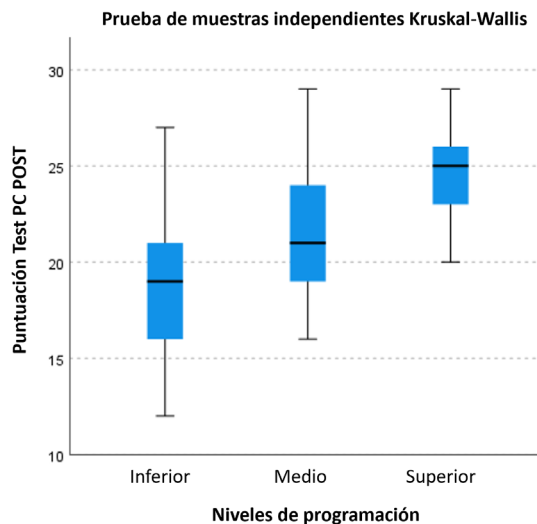
*Relación entre los valores del proyecto Scratch y el Test PC (post)*

Nivel de programación	Inferior [4,8] 20 participantes	Medio [9,10] 25 participantes	Superior [11,12] 26 participantes
Total programación	6,6 ± 0,3 (SE)	9,6 ± 0,1 (SE)	11,27 ± 0,09 (SE)
Puntuación Test PC POST	18,9 ± 0,8 (SE)	21,6 ± 0,7 (SE)	24,6 ± 0,6 (SE)
Diferencias	Medio-Inferior: 2,7 ± 1,0 (SE) p = 0,03	Superior-Medio: 3,1 ± 0,9 (SE) p=0,007	Superior-Inferior: 5,7 ± 1,0 (SE) p < 0,001

Podemos observar que hay una gradación en la puntuación del Test PC que corresponde también a la gradación de los tres niveles de programación, de manera que los/as estudiantes pertenecientes al grupo de nivel de programación Inferior obtienen una puntuación media en el Test PC POST de 18,9 y ésta es superada por 2,7 puntos por los/as estudiantes del nivel de programación Medio y por 5,7 puntos por los/as estudiantes del nivel de programación Superior. Para comprobar la significación estadística de las diferencias entre los grupos se ha utilizado la prueba no paramétrica Kruskal-Wallis de comparación para más de 2 grupos y se ha considerado que estas no son debidas al azar para  $p < 0,05$ . Como puede verse también en la Tabla 5, las diferencias en los resultados del Test PC POST son significativas en las comparaciones por parejas de los 3 grupos. La Figura 5 muestra la gradación de la puntuación del Test PC POST según los grupos de los niveles de programación.

### Figura 5

*Distribución de los resultados del Test PC (post) en función de los resultados del proyecto Scratch*



El resultado refleja que el trabajo en Scratch que suponga cuidado y dedicación puede ayudar en el desarrollo del PC independientemente del nivel de desarrollo del PC del que se partía. Por tanto, obtener un buen resultado programando Scratch no depende tanto del nivel de PC inicial (como dicen sus creadores, Scratch tiene un suelo bajo desde el cual se puede evolucionar a un techo tan alto como se quiera (Resnick, 2018)); al contrario, haber realizado un trabajo planificado y consciente



(no incidental) en Scratch explorando y utilizando sus distintas posibilidades y dimensiones repercute en el desarrollo del nivel de PC posterior.

Para completar el análisis de la relación entre los resultados de Programación Scratch y los resultados del Test PC, exponemos los resultados de las correlaciones entre las dos pruebas, ya que se pueden interpretar en términos de validez predictiva del Test PC (en el caso de la correlación “Test PC-pre \* proyectos Scratch” y en términos de validez concurrente/convergente del test, en el caso de la correlación “Test PC-post \* proyectos Scratch” en una aproximación similar a Román-González et al. (2019). El valor de la correlación “Test PC-pre \* proyectos Scratch” es de  $r=0,40$  y es estadísticamente significativa. La convergencia es parcial, y concuerda con los resultados obtenidos en el análisis por terciles según nivel de partida en la puntuación del Test PC, que indican que los/as participantes que partían del nivel inferior y del nivel medio han obtenido unos resultados similares en programación, superados sensiblemente por los/as participantes del nivel superior (Tabla 4), por lo cual la formación en Scratch ha tenido el efecto compensatorio que se va observando a lo largo del análisis. El valor de la correlación “Test PC-post \* proyectos Scratch” es  $r=0,59$ . Este valor más alto que en el caso del Test PRE concuerda con los resultados del análisis por terciles según los resultados de la programación (Tabla 5), lo que confirma que los conceptos y habilidades adquiridos durante la formación en Scratch y el empeño en la realización del proyecto no depende tanto del nivel de PC inicial, pero sí suponen un desarrollo en este como se plasma en el nivel de PC final según el Test.

## DISCUSIÓN Y CONCLUSIONES

A partir de todo lo que hemos venido refiriendo en el apartado de resultados, corresponde ahora tratar de sintetizar los hallazgos principales e interpretarlos a la luz de lo que ya conocíamos por la literatura previa. Por claridad del discurso, intentaremos ordenar esta reflexión en dos grandes partes: la primera de ellas, centrada en la validación de la propia formación (en qué medida una experiencia formativa intensiva como la que presentamos permite desarrollar el PC, al estilo de lo que veíamos en Pala y Türker (2021), por ejemplo). En segundo lugar, nos centraremos en las relaciones entre este desarrollo del PC y la creación de un proyecto Scratch.

En relación con el primer bloque, constatamos las siguientes tres ideas importantes. (1) Hay un desarrollo generalizado del PC según los parámetros medidos por el Test PC, lo que evidencia que la propuesta formativa permite a todo el alumnado alcanzar sus objetivos de aprendizaje. (2) Los/as participantes sin experiencia previa parten de un nivel PC inferior que los/as participantes con experiencia, en efecto; pero los/as primeros/as experimentan un incremento medio en la puntuación del Test PC mayor a los/as segundos/as. Sin embargo, en el grupo con mayor experiencia, después de la formación, el tiempo necesario para realizar

el test disminuye. Y (3) el análisis de la población analizada en tres niveles iniciales según la puntuación PRE en el Test PC y el análisis de los resultados del Test PC POST para cada uno de estos grupos muestra que los/as estudiantes que parten de niveles inferiores experimentan un incremento medio en la puntuación del Test PC mayor a los/as estudiantes que partían de niveles superiores. En los niveles superiores el desarrollo del PC se muestra en la disminución del tiempo necesario en la resolución.

De estos tres puntos concluimos que la formación cumple con el objetivo de desarrollar el PC según los parámetros medidos por el Test PC, como ya se documentaba, con una sola cohorte, en Peracaula-Bosch y González-Martínez (2022); pero no solo eso, sino que también tiene un efecto igualatorio en la capacidad de resolver los problemas planteados por el test y agiliza su resolución en aquellos/as estudiantes que ya partían de un nivel tan elevado que difícilmente permitía mejoras en la puntuación. De algún modo, esto va en sintonía con lo que señala Wong (2023) con la población escolar: “The results of the study show that children, regardless of their prior problem-solving skills, significantly improve their understanding of basic CT through programming, particularly for low-performing students” (p. 17). En su sencillez y economía, pues, la estrategia formativa resulta efectiva en general, y sobre todo para aquellos con puntos de partida menos ventajosos (Morze et al., 2022). Y es especialmente importante si tenemos presente la necesidad de garantizar un conjunto de destrezas técnicas de usuario suficiente para afrontar la segunda parte de la asignatura, dedicada ya sí a la didáctica del PC (y por lo que avanza para esta segunda parte que el alumnado no solo desarrolle su propio PC, sino que conozca Scratch lo suficiente para poderlo transferir a su planeación didáctica), en línea con lo que señalan Rich, Larsen y Mason (2021) y Collado-Sánchez et al. (2021).

En relación con el segundo bloque, dedicado a analizar la relación entre el nivel de programación y los resultados del Test PC, hemos desglosado el estudio en dos pasos. Y hemos encontrado que, por un lado, los resultados del Test PC previo a la formación se relacionan con las competencias y habilidades demostradas por los/as estudiantes en la elaboración de un proyecto complejo de programación con Scratch en el caso de los sujetos que partían de una puntuación alta en el Test PC. Para los/as estudiantes que puntuaron en los niveles inferior e intermedio esta relación no es significativa, de modo que priman en los resultados los aprendizajes de la formación. Por el otro, la puntuación en competencias y habilidades mostradas en el uso de elementos y conceptos de programación en la realización del proyecto Scratch sí muestra una relación significativa con los resultados del Test PC POST, lo que indica que la formación realizada mediante este recurso y con el diseño utilizado ha repercutido en el desarrollo del PC (Pala y Türker, 2021; Rich, Mason y O’Leary, 2021), a pesar del poco tiempo disponible para ello, que obliga a un planteamiento intensivo (Ung et al., 2022).

Finalmente, a pesar de estos buenos resultados (tanto por la confirmación de experiencias previas como por la profundización en el análisis específico de proyecto Scratch), debemos reconocer limitaciones en la investigación que, al mismo

tiempo, se convierten en posibles futuras líneas indagatorias. En la evaluación final del trabajo de los/as estudiantes se tenía también en cuenta otros factores como creatividad, uso de elementos artísticos, y calidad de la redacción del documento final donde explicaban el proceso, los aprendizajes obtenidos y la transferencia y su uso didáctico en la escuela. Sin embargo, para nuestro análisis nos limitamos al análisis de los elementos que tienen relación directa con el desarrollo del PC según los parámetros medidos por el Test PC. No cabe duda de que la complejidad del propio PC (Wing, 2014) nos debe llevar a plantear ulteriores análisis. Y es que, como señalan Acevedo-Borrega et al. (2022) o Morze et al. (2022) debemos profundizar en la conceptualización del PC en el ámbito educativo, en un camino de ida y vuelta entre la investigación y la práctica.

## REFERENCIAS

- Acevedo-Borrega, J. (2016). *El pensamiento computacional en la educación obligatoria. Una revisión sistemática de la literatura* [Tesis de Maestría, Universidad de Extremadura]. DEHESA. Repositorio institucional Universidad de Extremadura. <http://hdl.handle.net/10662/5356>
- Acevedo-Borrega, J., Valverde-Berrocoso, J. y Garrido-Arroyo, M. D. C. (2022). Computational Thinking and Educational Technology: A Scoping Review of the Literature. En *Education Sciences*, 12(1), Artículo 39. <https://doi.org/10.3390/educsci12010039>
- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J. y Zagami, J. (2016). A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. *Educational Technology & Society*, 19(3), 47-57. <https://www.jstor.org/stable/jeductechsoci.19.3.47>
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A. y Engelhardt, K. (2016a). *Developing Computational Thinking in Compulsory Education. Implications for policy and practice*. <https://doi.org/10.2791/792158>
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P. y Punie, Y. (2016b). Developing Computational Thinking: Approaches and Orientations in K-12 Education. En *Proceedings of the EdMedia 2016 Conference* (pp. 13-18). Association for the Advancement of Computing in Education (AACE). <http://www.learntechlib.org/p/172925/>
- Bustillo, J. y Garaizar, P. (2015). Scratching the surface of digital literacy... but we need to go deeper. *Proceedings of the Frontiers in Education Conference (FIE)*, 1-4. <https://doi.org/10.1109/FIE.2014.7044224>
- Butler, D. y Leahy, M. (2021). Developing preservice teachers' understanding of computational thinking: A constructionist approach. *British Journal of Educational Technology*, 52(3), 1060-1077. <https://doi.org/10.1111/bjet.13090>
- Collado-Sánchez, M., García-Peñalvo, F. J. y Pinto-Llorente, A. M. (2021). Computational thinking competences training for primary education teachers. En *Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21)*, 758-762. <https://doi.org/10.1145/3486011.3486544>
- Dobganski, J. y Garcia Silva, A da F. (2022). *A practical experience in pre-service teacher education focusing on*

- computational thinking*. 12th Congress of the European Society for Research in Mathematics Education (CERME12), Feb 2022, Bozen-Bolzano, Italia. <https://hal.archives-ouvertes.fr/hal-03748506>
- Estebanell, M., López, V., Peracaula-Bosch, M., Simarro, C., Cornellà, P., Couso, D., González-Martínez, J., Alsina, À., Badillo, E. y Heras, R. (2018). *Teacher training in Computational Thinking*. *Teaching Guide*. <https://pecofim.wixsite.com/pecofim/guia-didactica>
- European Commission / EACEA / Eurydice. (2012). *Developing Key Competences at School in Europe: Challenges and Opportunities for Policy*. Eurydice Report. Luxembourg: Publications Office of the European Union. <https://doi.org/10.2797/93204>
- Fluck, A., Webb, M., Cox, M., Angeli, C., Malyn-smith, J., Voogt, J. y Zagami, J. (2016). Arguing for Computer Science in the School Curriculum. *Educational Technology & Society*, 19(3), 38-46. <https://www.jstor.org/stable/jeductechsoci.19.3.38>
- Furber, S. (2012). *Shut down or restart? The way forward for computing in UK schools*. The Royal Society. <https://royalsociety.org/-/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf>
- González-Martínez, J., Estebanell, M. y Peracaula-Bosch, M. (2018). ¿Robots o programación? El concepto de Pensamiento Computacional y los futuros maestros. *Education in the Knowledge Society (EKS)*, 19(2), 29-45. <https://doi.org/10.14201/eks20181922945>
- Grover, S. y Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>
- Kong, S. C., Lai, M. y Li, Y. (2023). Scaling up a teacher development program for sustainable computational thinking education: TPACK surveys, concept tests and primary school visits. *Computers & Education*, 194, Artículo 104707. <https://doi.org/10.1016/j.compedu.2022.104707>
- Lamprou, A. y Repenning, A. (2018). Teaching How to Teach Computational Thinking. En *Proceedings of 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'18)*. ACM. <https://doi.org/10.1145/3197091.3197120>
- Li, Q. (2021). Computational thinking and teacher education: An expert interview study. *Human Behavior and Emerging Technologies*, 3, 324-338. <https://doi.org/10.1002/hbe2.224>
- Mason, S. L. y Rich, P. J. (2019). Preparing Elementary School Teachers to Teach Computing, Coding, and Computational Thinking. *Contemporary Issues in Technology and Teacher Education*, 19(4), 790-824. Waynesville, NC USA: Society for Information Technology & Teacher Education <https://citejournal.org/volume-19/issue-4-19/general/preparing-elementary-school-teachers-to-teach-computing-coding-and-computational-thinking>
- Mishra, P. y Koehler, M. J. (2006). Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge. *Teachers College Record*, 108(6), 1017-1054. <https://doi.org/10.1111/j.1467-9620.2006.00684.x>
- Moreno León, J., Román-González, M., Robles, G. (2022). *Escuela de Pensamiento Computacional e Inteligencia Artificial 20/21: Enfoques y propuestas para su aplicación en el aula. Resultados de la investigación*. Publicaciones Ministerio de Educación y Formación Profesional. <https://sede.educacion.gob.es/publivena/d/25861/19/0>
- Morreale, P., Jimenez, L., Goski, C. y Stewart-Gardiner, C. (2012). Measuring the impact of computational thinking workshops on high school teachers.

- Journal of Computing Sciences in Colleges*, 27(6), 151-157. <https://dl.acm.org/doi/abs/10.5555/2184451.2184486>
- Morze, N., Barna, O. y Boiko, M. (2022). The relevance of training primary school teachers computational thinking. En *Proceedings of the 17th International Conference on ICT in Education Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer (ICTERI 2021)*, 3104, 141-153. <https://ceur-ws.org/Vol-3104/paper218.pdf>
- Pala, F. K. y Türker, P. M. (2021). The effects of different programming trainings on the computational thinking skills. *Interactive Learning Environments*, 29(7), 1090-1100. <https://doi.org/10.1080/10494820.2019.1635495>
- Papert, S. (1980). *Mindstorms. Children, computers and powerful ideas*. Basic Books.
- Peracaula-Bosch, M., Estebanell-Minguell, M., Couso, D. y González-Martínez, J. (2020). What do pre-service teachers know about computational thinking? *Aloma. Revista de Psicologia, Ciències de l'Educació i de l'Esport*, 38(1), 75-86. <https://doi.org/10.51698/aloma.2020.38.1.75-86>
- Peracaula-Bosch, M. y González-Martínez, J. (2022). Developing computational thinking among pre-service teachers. *QWERTY. Open and Interdisciplinary Journal of Technology, Culture and Education*, 17(1), 28-44. <https://doi.org/10.30557/QW000049>
- Pérez-Marín, D., Hijón-Neira, R., Bacelo, A. y Pizarro, C. (2020). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Computers in Human Behavior*, 105(August 2018), Artículo 105849. <https://doi.org/10.1016/j.chb.2018.12.027>
- Resnick, M. (2018). *Lifelong Kindergarten. Cultivating Creativity through Projects, Passion, Peers and Play*. The MIT Press. <https://doi.org/10.7551/mitpress/11017.001.0001>
- Rich, P. J., Larsen, R. A. y Mason, S. L. (2021). Measuring teacher beliefs about coding and computational thinking. *Journal of Research on Technology in Education*, 53(3), 296-316. <https://doi.org/10.1080/15391523.2020.1771232>
- Rich, P. J., Mason, S. L. y O'Leary, J. (2021). Measuring the effect of continuous professional development on elementary teachers' self-efficacy to teach coding and computational thinking. *Computers & Education*, 168, Artículo 104196. <https://doi.org/10.1016/j.compedu.2021.104196>
- Román-González, M. (2016). *Codigoalfabetización y Pensamiento Computacional en Educación Primaria y Secundaria: validación de un instrumento y evaluación de programas* [Tesis Doctoral, Universidad Nacional de Educación a Distancia]. Repositorio Institucional de la Universidad Nacional de Educación a Distancia. [https://doi.org/10.1007/978-981-13-6528-7\\_6](https://doi.org/10.1007/978-981-13-6528-7_6)
- Román-González, M., Moreno-León, J. y Robles, G. (2019). Combining assessment tools for a comprehensive evaluation of computational thinking interventions. En S. Kong y H. Abelson (Eds.), *Computational thinking education* (pp. 79-98). Springer. <https://doi.org/10.1016/j.chb.2016.08.047>
- Román-González, M., Pérez González, J. C. y Jiménez Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior* 72(July), 678-691. <https://doi.org/10.1016/j.chb.2018.06.004>
- Román-González, M., Pérez-González, J. C., Moreno-León, J. y Robles, G. (2018). Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal*

- of *Child-Computer Interaction*, 18(November), 47-58. <https://doi.org/10.1016/j.ijcci.2018.06.004>
- Rubio Hurtado, M. J. y Berlanga Silvente, V. (2012). Cómo aplicar las pruebas paramétricas bivariadas t de Student y ANOVA en SPSS. Caso práctico. *REIRE Revista d'Innovació i Recerca en Educació*, 5(2), 83-100. <https://doi.org/10.1344/reire2012.5.2527>
- Simpson, S. H. (2015). Creating a Data Analysis Plan: What to Consider When Choosing Statistics for a Study. *The Canadian Journal of Hospital Pharmacy*, 68(4), 311-317. <https://doi.org/10.4212/cjhp.v68i4.1471>
- Ung, L. L., Labadin, J. y Mohamad, F. S. (2022). Computational thinking for teachers: Development of a localised E-learning system. *Computers & Education*, 177(February). Artículo 104379. <https://doi.org/10.1016/j.compedu.2021.104379>
- Wing, J. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. (10 de enero 2014). Computational Thinking benefits society. *Social Issues in Computing*. <http://socialissues.cs.toronto.edu/index.html%3Fp=279.html>
- Wong, G. K. W. (2023). Amplifying children's computational problem-solving skills: A hybrid-based design for programming education. *Education and Information Technologies*. <https://doi.org/10.1007/s10639-023-11880-9>

**Fecha de recepción del artículo:** 1 de junio de 2023

**Fecha de aceptación del artículo:** 30 de agosto de 2023

**Fecha de aprobación para maquetación:** 6 de octubre de 2023

**Fecha de publicación en OnlineFirst:** 19 de octubre de 2023

**Fecha de publicación:** 1 de enero de 2024