



## *Competencia de Pensamiento Computacional en la educación no formal*

### *Competence of Computational Thinking in non-formal education*

 Manuel Zarza González; [manuel.zarza@estudiants.urv.cat](mailto:manuel.zarza@estudiants.urv.cat)

 Josep Holgado García; [josep.holgado@urv.cat](mailto:josep.holgado@urv.cat)

Universitat Rovira i Virgili (España)

#### Resumen

Aunque la trayectoria de la educación en el tiempo libre es dilatada, frecuentemente se subestima su potencial pedagógico y no es considerada como una opción para el desarrollo de competencias para la vida. De esta manera, tampoco se debe obviar el poder de la educación no formal como ámbito para la adquisición de competencias propias de una sociedad digital. Es por este motivo que el presente artículo aborda el desarrollo de la Competencia de Pensamiento Computacional en los contextos de educación no curricular.

Para lograr los objetivos de la investigación, se plantea un estudio sujeto a una pequeña muestra participante en un casal de verano. La discusión de los datos obtenidos demuestra una mejora en el grado de conocimiento de los contenidos computacionales, tanto conceptuales como procedimentales, además de desarrollar un abanico de conocimientos del ámbito saber ser gracias al seguimiento de una secuencia didáctica basada en el uso del recurso digital Code.org.

**Palabras clave:** Competencia; Pensamiento Computacional; Educación no-formal

#### Abstract

*Although the trajectory of non-formal education is extensive, its pedagogical potential is frequently underestimated and is not considered as an option for the development of life competences. In this way, the power of non-formal education as a field for skills education in a digital society should not be ignored. It is for this reason that this article addresses the development of the Computational Thinking Competency in non-curricular education contexts.*

*To achieve the objectives of the research, a study is proposed subject to a small sample participating in a summer camp. The discussion of the data obtained demonstrates an improvement in the degree of knowledge of the computational contents, both conceptual and procedural, in addition to developing a range of knowledge of the field know how to be thanks to the follow-up of a didactic sequence based on the use of the digital resource Code .org*

**Keywords:** Competence, Computational Thinking; Non-Formal Education



## 1. INTRODUCCIÓN.

Atendiendo al cambio constante de la sociedad del conocimiento, nuestro alumnado se encontrará en un futuro próximo una realidad muy diferente a la actual a la cual deberá dar respuesta. A su vez, los docentes deben prepararlos para atender estos nuevos desafíos promoviendo un cambio metodológico a fin de fomentar competencias que permitan dotarles de nuevas estrategias a para de resolver problemas concretos de manera efectiva (Boix Tormos, 2016).

Un modelo educativo basado en competencias ha de facilitar a los estudiantes el hecho de identificar los mecanismos necesarios para dar respuesta a los problemas a los que se enfrentan con acciones competentes. De esta manera, podremos hablar de instituciones educativas que formen para la vida (Zabala y Arnau, 2011).

No hay que obviar la importancia que tienen los centros reglados como organismos formadores en competencias (Decreto 119/2015, 2015). Sin embargo, también conviene destacar la importancia de la educación en el tiempo libre en la adquisición de competencias dado que este modelo educativo también las promueve fuera del ámbito curricular, aunque frecuentemente, todo el bagaje y potencial de la educación no formal está infrautilizado e infravalorado (Ruiz, 2012). Considerando este aspecto, la educación no formal también debe dar respuesta a la complejidad de la sociedad actual de la misma forma que lo hace la formación curricular, por lo tanto, se debe experimentar un mayor grado de innovación en el desempeño de las acciones educativas de tiempo libre con el uso de las tecnologías digitales (Cañellas, 2005).

De hecho, contemplando la diversidad de actividades extraescolares, dos de las más demandadas son inglés e informática (Cañellas, 2005). En este sentido, se observar mediante un claro ejemplo que la educación no formal puede complementar aquellos aspectos que se trabajan competencialmente en el ámbito curricular y viceversa.

Atendiendo a la patente relación entre estos dos tipos de educación, resulta de especial interés el estudio del impacto de las Tecnologías del Aprendizaje y del Conocimiento (TAC) en el desarrollo de las competencias digitales de los educandos en ambos ámbitos de acción educativa. No obstante, siendo la educación no formal un tipo de educación frecuentemente valorada en menor grado que la educación curricular, en el presente estudio se plantea el Pensamiento Computacional como una competencia que enriquece el ámbito de la educación en el tiempo libre y que fomenta el interés hacia las STEAM (acrónimo en inglés de las disciplinas de Ciencias, Tecnología, Ingeniería, Arte y Matemáticas) fuera del ámbito curricular.

### 1.1. La Competencia de Pensamiento Computacional

El término de Pensamiento Computacional, de aquí en adelante PC, se concibe en Europa pero se desarrolla por primera vez en los estados unidos cuando el Dr. Seymour Papert ideó y desarrolló, juntamente con el psicólogo Jean Piaget, el primer lenguaje de programación diseñado para la



educación (Picas, 2017). No obstante, aunque fue el Dr. Papert quién estipuló los inicios de la programación y la robótica educativa, no es hasta que la doctora Jeannette Wing estipuló una primera definición del concepto de PC en el año 2006 (Zúñiga y Hurtado, 2015).

Wing (2006) defiende que los conceptos computacionales suponen la resolución de problemas, el planteamiento de procedimientos y la comprensión de la conducta humana basándose en las nociones elementales de la informática. Además, el PC utiliza la abstracción y la descomposición de tareas para que un problema sea soluble. Esta definición generó un gran debate a nivel internacional del valor del PC en el ámbito de la educación y por la falta de consenso del término, Wing propuso una redefinición de este en un artículo titulado “Pensamiento computacional: ¿Qué y por qué?” (Wing, 2010). Actualmente, esta misma terminología sigue teniendo distintos significados según el contexto, por lo que no existe un acuerdo en la definición del concepto (Román, 2016).

Aun cuando no hay un consenso generalizado en la definición del término, existe un mínimo común entre las diferentes dinámicas fundamentadas en los preceptos del PC: Este consiste en la relación del pensamiento matemático-abstracto con el pragmatismo-ingenieril, hecho que acentúa la correlación entre el pensamiento y la acción, idea necesaria para dar respuestas eficaces a situaciones cotidianas con el uso de recursos propios de la computación (Valverde et al., 2015).

Es importante enfatizar que aunque la relación de los saberes matemáticos e ingenieriles están estrechamente relacionados, paralelamente se ha de propiciar el desarrollo de los mecanismos de abstracción y descomposición de problemas; la resolución de los mismos gracias al uso de los conocimientos generados por otras situaciones similares; y conformar soluciones empleando la información de un modo claro, elegante y eficiente (Zúñiga y Hurtado, 2014) con la finalidad de no incurrir en dinámicas de enseñanza y aprendizaje basadas en la rutina y la mecánica de los procesos de resolución de problemas ya que se deben estimular respuestas imaginativas e inteligentes a problemas contextualizados (Wing, 2006).

Si bien el empleo de recursos propios de la computación facilita el desarrollo de la Competencia de Pensamiento Computacional (Valverde et al., 2015), conviene subrayar que los términos de PC y codificación no son sinónimos, pero sí que están muy relacionados: Por un lado, el PC utiliza los conocimientos del ámbito de la informática con el propósito de resolver problemas e involucra un conjunto de habilidades como el análisis de problemas y el pensamiento algorítmico, mientras que, por el otro lado, la programación es un recurso para la educación del PC mediante la generación de código (Sullivan, Bers y Mihm, 2017).

A pesar de que la literatura sobre el PC va *in crescendo*, todavía existe un vacío sobre cómo actuar para promover esta competencia en las acciones educativas de una manera interdisciplinaria y con eficiencia (Román, 2016). Aunque en el ámbito profesional se frecuenta la escritura de líneas de código a fin de generar respuestas a problemas específicos, a la hora de trabajar el PC con niños y niñas de educación primaria, esta opción puede quedar descartada debido a su dificultad (Vidal et



al., 2015). De hecho, si se aprende a programar con un lenguaje de programación específico basado en texto, es posible que aquello trabajado con los educandos no sea transferido a otros contextos (Álvarez, 2017).

Teniendo en cuenta que la escritura de código de manera compulsiva no es sinónima a la adquisición de esta competencia (Álvarez, 2017) pero que a la vez la programación es una habilidad estrechamente relacionada con el PC (Strawhacker y Bers, 2015) deberemos emplear recursos que no estén basados en la redacción de código a la hora de proponerlos en acciones educativas.

Un buen y popular ejemplo es la herramienta SCRATCH: Este programa de libre acceso permite que los usuarios puedan crear sus proyectos sin tener que lidiar con la falta de conocimientos de codificación basada en texto pues las instrucciones se introducen mediante bloques visuales. Así entonces, se centra toda la atención en el desarrollo del PC con el objetivo de fomentar la resolución de problemas evitando los errores de sintaxis que puedan surgir de la escritura de código (Álvarez, 2017). Además, este tipo de codificación visual basado en bloques gráficos es utilizado por infinidad de recursos educativos ya que también permite el trabajo de los procesos de resolución de un problema de la misma forma que la programación tradicional e industrial (Basogain y Olabe, 2015).

## 1.2. La Competencia de Pensamiento Computacional en la educación no formal

Cada vez son más los espacios que promueven la adquisición de esta competencia en horas no curriculares tanto dentro de las propias instalaciones de los centros escolares como en otros espacios físicos. A modo de ejemplo, una clara evidencia este tipo de comunidades son los conocidos *Code Clubs* que fueron creados con el propósito de inspirar las vocaciones STEAM mediante la programación educativa y el voluntariado (Smith, Sutcliffe, y Sandvik, 2014).

En la actualidad, ya se han registrado un total de trece mil clubs en ciento sesenta países. Estos han propiciado que un total de ciento ochenta mil inscritos hayan encontrado una comunidad donde aprender a crear sus propios proyectos de manera creativa, inicialmente con el uso de SCRATCH y posteriormente, con lenguajes de programación escritos como el HTML, CSS y PHYTON (Codeclub.org, 2020).

A nivel iberoamericano y español, también existen instituciones que impulsan prácticas basadas en la programación y las implementan con el uso de la robótica educativa en entornos extraescolares. En este caso, estos proyectos están liderados mayoritariamente por empresas privadas y se desenvuelven en una franja de edad que oscilan entre los 13 y los 17 años (Pittí, Curto, Moreno, y Rodríguez, 2014).



Estas acciones de robótica educativa implican el uso de motores, sensores, conectores y más elementos de maquinaria informática juntamente con la programación. Un ejemplo es el popular torneo de robótica educativa *First Lego League* que emplea como recurso el kit *Lego Mindstorms*. Este tipo de dinámicas y sus actividades derivadas permiten experimentar una mejoría, por parte de los participantes, en el conocimiento conceptual, procedimental y actitudinal hacia el método científico, el trabajo en equipo y la robótica (Karp y Maloney, 2013).

Estas prácticas aprovechan el potencial de los grandes avances tecnológicos para favorecer los procesos de enseñanza y aprendizaje en la educación de tiempo en un entorno lúdico y cordial (Castro, Briegas, González, y González, 2017). No obstante, el principal problema de estas actividades consiste en que no se extienden a todo el alumnado por su elevada implicación económica y por la dificultad que tienen las familias para afrontar los gastos derivados si estos no están subvencionados con fondos públicos (Vega, Cufié, Rueda y Llinás, 2016).

## 2. MÉTODOS

La investigación centra su atención en la posible correlación entre la educación no formal y el desarrollo de la Competencia de Pensamiento Computacional. En este sentido, se propone la relación de dos variables: Por un lado, la participación en actividades de programación educativa en un entorno de educación no formal y por el otro, el desarrollo de la Competencia de Pensamiento Computacional.

El presente estudio se realiza desde la Investigación Acción (IA) y sigue un enfoque de tipo mixto (cualitativo y cuantitativo) para la recogida y análisis de datos. En se escoge la IA dado que permite a los docentes el hecho de integrar la reflexión de las prácticas que se realizan en las aulas, en este caso en un contexto de educación no formal, a fin de concebir el cambio educativo fundamentado mediante la investigación (Torrecilla, 2011). Asimismo, se emplean técnicas y métodos de investigación cualitativa y cuantitativa en un único enfoque como alternativa a los métodos únicamente cuantitativos o cualitativos (Pérez, 2011).

Atendiendo a la tipología mixta de los datos que se toman en el presente estudio, los datos cualitativos se recogieron, redujeron, codificaron y categorizaron en las categorías que posteriormente se describen y para los cuantitativos, se utiliza un diseño no experimental descriptivo.

### 2.1. Objetivos y preguntas de investigación

En las siguientes líneas, se muestran el conjunto de objetivos a alcanzar con la investigación. Se expresa un objetivo general y un conjunto de 3 específicos:



- **Objetivo General:** Implementar y precisar el impacto de una secuencia didáctica relacionada con la Competencia de Pensamiento Computacional en un entorno de educación en el tiempo libre.
  - **Objetivo específico 1:** Evaluar el grado de conocimientos de Pensamiento Computacional antes de empezar una secuencia didáctica y al finalizar la misma.
  - **Objetivo específico 2:** Identificar el conjunto de actitudes y capacidades desarrolladas durante la realización de actividades de programación educativa.
  - **Objetivo específico 3:** Conocer la percepción del alumnado sobre el aprendizaje de Competencia de Pensamiento Computacional experimentado tras finalizar la experiencia.

Y surgen las siguientes preguntas de investigación

- **Pregunta 1:** ¿Qué diferencia de nivel de conocimientos de Pensamiento Computacional supone al alumnado al participar en una secuencia didáctica de programación educativa?
- **Pregunta 2:** ¿Qué capacidades y actitudes se fomentan al trabajar la programación educativa en un ámbito de educación no formal?
- **Pregunta 3:** ¿Qué percepción de aprendizaje de Competencia de Pensamiento Computacional han experimentado los alumnos al finalizar una experiencia en el ámbito de la educación no formal?

## 2.2. Participantes

El contexto en el cual se basa esta investigación es el de un casal de verano situado en el municipio de Reus (Tarragona, Cataluña) y se enmarca en la definición de "Casales de verano y de vacaciones" (Decreto 267/2016, 2016).

La muestra para este proyecto de investigación ha sido de un total de 9 educandos que durante el estudio estaban estudiando quinto y sexto de primaria. Es decir, el grupo se ha conformado por los usuarios de ciclo superior, estos representan el 16,67% de la participación total del casal.

De esta población, considerando la variable de género, 6 personas se identifican como chicos y 3 como chicas. Mientras que atendiendo al curso académico que han realizado antes del verano, 5 han cursado quinto mientras que 4 han realizado sexto.



Es importante subrayar que un total de 4 participantes han sido excluidos de la investigación por no poder participar en el total de sesiones de la secuencia didáctica y es por este motivo que sus datos no han sido expresados en el presente estudio. Este hecho es motivado porque sus familias han realizado una inscripción al casal que no les ha permitido asistir a la totalidad de las sesiones.

### 2.3. Instrumentos de recogida de datos

Por un lado, se emplea el Test de Pensamiento Computacional (TPC) del Dr. Marcos Román puesto que permite conocer el grado de desarrollo del Pensamiento Computacional en sujetos de primero y segundo curso de Educación Secundaria Obligatoria (ESO) aunque también es funcional para el último ciclo de Educación Primaria (Quinto y Sexto) y tercero y cuarto de la ESO (Román, 2016).

Se trata de una prueba objetiva en la cual hay 28 ítems a responder mediante la elección múltiple con cuatro opciones de respuesta y solamente una es correcta. El tiempo máximo para realizar-la es de 45 minutos (Román, 2016) y se emplea en dos sesiones distintas: Inicialmente como Pre-Test y posteriormente como Post-Test.

Por el otro lado, se propone el uso del Cuestionario de Satisfacción para estudiantes sobre el Curso K-8 de Code.org del mismo autor (Román, 2016). Este instrumento contiene un total de 12 preguntas que combinan en un único espacio preguntas de elección múltiple, escalas Likert e interrogantes abiertos (Román, 2016).

Con esta herramienta, se pretende recoger información de carácter cuantitativa y cualitativa una vez finalizada la participación en el curso K-8 propuesto por la plataforma Code.org.

### 2.4. Intervención didáctica en el ámbito de educación no formal

Con el propósito de desarrollar una acción formativa en el marco de la educación no formal, se planifica una secuencia didáctica basada en la propuesta del curso “*K-8 Intro to Computer Science*” de la plataforma Code.org. La motivación por el uso del contenido de esta plataforma ha sido basada en los siguientes preceptos:

- Los retos planteados son ordenados por orden de dificultad.
- Las actividades están contextualizadas en películas y videojuegos conocidos por los participantes. Hecho que permite la proximidad entre contenido y educando.



- Aunque también se proponen actividades en formato analógico, la mayoría contemplan la programación por bloques con un estilo similar a Scratch como herramienta de aprendizaje.
- Las dinámicas permiten el trabajo colaborativo y entre iguales.
- Existen instrumentos específicos con un gran grado de fiabilidad relacionados con la plataforma que facilitan medir el impacto de estas acciones.

Dadas las condiciones de temporalización del casal de vacaciones de verano, solamente se desarrollan ocho sesiones didácticas aun cuando la plataforma ofrece un abanico más amplio de lecciones. A fin de conocer los objetivos, recursos utilizados, actividades y sus respectivas temporalizaciones, se adjunta en el apartado de enlaces una infografía con el detalle de cada sesión.

### 3. RESULTADOS

Con la finalidad de aclarar los diferentes resultados, se analizan los dos instrumentos de recogida de datos de manera separada y más adelante, se realizan las interrelaciones oportunas atendiendo a los diferentes objetivos y preguntas de investigación. Se inicia la exposición de resultados del TPC.

#### 3.1. Resultados del Test de Pensamiento Computacional (TPC)

Con el propósito de conocer la diferencia de conocimientos de PC de los educandos participantes, se exhiben los resultados obtenidos antes de realizar la acción educativa en un entorno no formal y se comparan con aquellos obtenidos al finalizar la secuencia didáctica.

Una vez recogidas las respuestas del Pre-Test y del Post-Test, se ha enumerado con el valor 0 aquellas preguntas que han sido contestadas incorrectamente. En cambio, las que han recibido una solución acertada han sido puntuadas con el valor 1. Una vez procesados estos datos, se ha evidenciado un porcentaje de respuestas nulas mayor a aquellas que se hayan con la adecuada en el Pre-Test.

Con el objetivo de comprender la tendencia central del conjunto de sumatorios de respuestas correctas del Pre-Test, se desempeña el cálculo de la media aritmética, la cual facilita un resultado de 11,88 periódicos de contestaciones acertadas por participante, un valor no muy lejano la cantidad reflejada por la moda y la mediana, es decir, 12 preguntas sin error.

Gracias a la desviación estándar, obtenida a partir del cálculo de la varianza, se puede observar que puntuaciones están situadas a una distancia mayor o menor a la media obtenida en el Pre-Test.



En este sentido, es observable que la nota mínima conseguida es de 9 puntos y las máximas son de 14 puntos tal y como es observable en la siguiente tabla (Tabla 2).

Tabla 1: Media, mediana, moda, desviación estándar, varianza y valores máximos y mínimos de los resultados del Pre-Test y del Post-Test

	Valores del Pre-Test	Valores del Post-Test	Diferencia valores Pre-Test y Post-Test
Media	11,8888889	15,7777778	1,25
Mediana	12	16	1
Moda	12	15	0
Desviación estándar	1,90029238	1,48136574	2.066
Varianza	3,61111111	2,19444444	4.269
Mínimo	9	13	-3
Máximo	14	18	5

En cambio, el cómputo de la media aritmética del Post-Test proporciona una solución de 15,77 puntos periódicos de contestaciones acertadas por educando, una cifra no muy distante a la cuantía reflejada por la moda (15 puntos) y la mediana (16 puntos).

Obteniendo la desviación estándar, se procede al cálculo de la varianza, con la finalidad de reconocer que puntuaciones están situadas a una distancia mayor o menor a la media obtenida al finalizar la secuencia didáctica del curso “K-8 Intro to Computer Science”. De esta manera, es observable que la nota mínima conseguida en el Post-Test es de 13 puntos y las máximas son de 18 puntos tal y como es observable en la tabla anteriormente expuesta (Tabla 1).

Atendiendo a la diferencia entre las preguntas acertadas en el TPC inicial y final, véase en el siguiente gráfico (Figura 1) que las preguntas 1, 3, 6 y 9 no han supuesto ningún error de contestación en el test inicial. Por el contrario, las cuestiones 8, 14, 16 y 28 no han sido acertadas por ningún educando. Además, un total de 11 interrogantes solo han obtenido respuesta por dos o menos participantes, sin contar en esta cantidad las que no han sido respondidas adecuadamente por nadie.

Mientras que, en el caso del Post-Test, hay un total de 6 preguntas que han sido acertadas por la totalidad de las personas participantes, estas cuestiones corresponden al número 1, 2, 3, 5, 6 y 9, situadas al principio del test. Mientras que un total de 6 preguntas no han sido respondidas correctamente por nadie en el grupo, estas están identificadas por los siguientes números: 8, 15, 22, 23, 25 y 28.



### COMPARATIVA DE RESULTADOS PRE-TEST Y POST-TEST

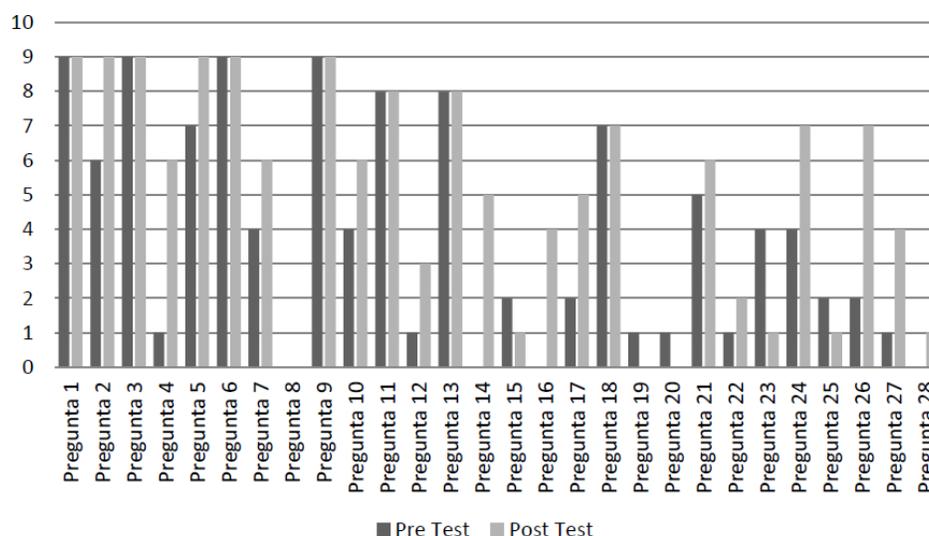


Figura 1: Comparativa de resultados Pre-Test y Post-Test. Fuente: Elaboración propia.

Es aparente que no ha existido aumento de puntuación en las preguntas con número 8, 11, 13 y 15. En esta lista se han descontado las cuestiones que no podían experimentar ninguna mejora dado que habían alcanzado la totalidad de personas acertantes y han continuado manteniendo la máxima cifra.

En el caso de las preguntas 15, 19, 20, 23 y 25 se ha experimentado un retroceso, en otras palabras, han obtenido más contestaciones incorrectas al final del proceso que al principio. En su totalidad, en excepción de la pregunta 23, estas solamente contaban en el Pre-Test con una franja que oscilaba entre uno o dos participantes que las habían resuelto con certeza.

Además, es constatable que la pregunta que más personas han fallado en el Post-Test respecto al Pre-Test ha sido la número 23, perdiendo a 3 acertantes. En el caso de las preguntas 4, 14 y 16, han obtenido cinco acertantes más respectivamente. A modo de facilitar el análisis de la información recogida, se expone la mejoría de puntuación que ha experimentado cada pregunta por grupos (Tabla 2).

Tabla 2: Mejoría de puntuación por grupos.

Grado de mejoría	Total preguntas	Preguntas
Mejora de 0 puntos	8	Preguntas 1, 3, 6, 8, 9, 11, 13, 18
Mejora de 1 punto	3	Preguntas 21, 22 y 28
Mejora de 2 puntos	4	Preguntas 5, 7, 10 y 12
Mejora de 3 puntos	3	Preguntas 2, 17 y 27
Mejora de 4 puntos	1	Pregunta 16
Mejora de 5 puntos	3	Pregunta 4, 14 y 26

### 3.2 Datos obtenidos gracias al cuestionario de Satisfacción para estudiantes sobre el Curso K-8 de Code.org

En el caso del TPC, se han obtenido datos de carácter cuantitativo, con este otro instrumento se recogen datos cuantitativos y cualitativos.

En primera instancia se pregunta a los participantes sobre sus notas habituales en diferentes ámbitos de conocimiento en la educación formal. Estos tienen que seleccionar sus calificaciones en un baremo que se inicia en el suspenso y finaliza en sobresaliente. Así entonces, los resultados señalan que los ámbitos curriculares con las puntuaciones más bajas son las de lengua y música, en cambio, las asignaturas con mayor cantidad de sobresalientes son las materias de plástica e informática.

En segundo lugar, se procede a conocer si los participantes tenían conocimientos previos de programación educativa antes de realizar la intervención. De esta manera, se analizan dos preguntas: El segundo interrogante del cuestionario hace referencia a si conocían el concepto de “programación informática” y el tercero, pretende conocer si habían programado con anterioridad. Los resultados muestran que ningún educando conocía el término de “*programación informática*” o “*coding*” aunque prácticamente todos los participantes (8 de los 9 educandos) sí que habían programado previamente.

En tercer lugar, se muestran los resultados de las capacidades que los participantes consideran haber desarrollado en la realización del curso. Los estudiantes han seleccionado una respuesta de una Escala de Likert con un valor mínimo de “nada” (cuantificado con un uno) y un valor máximo de “mucho” (con un total de cinco puntos) para los ítems de pensamiento lógico, creatividad, resolución de problemas, trabajo en equipo, persistencia y capacidad de atención.

Tal y como se puede observar en la siguiente tabla (Tabla 3) la capacidad de atención es el ítem mayor valorado por los educandos seguido del trabajo en equipo, el pensamiento lógico y creatividad. En último lugar, la persistencia es quien es percibida en menor grado, precedida antes por la resolución de problemas.



Tabla 3: Media de las auto valoraciones de las actitudes y habilidades desarrolladas

Pensamiento lógico	Creatividad	Resolución de problemas	Trabajo en equipo	Persistencia	Capacidad de atención
4	4	3.88	4.11	3.66	4.33

En cuarto lugar, se procede a conocer la auto percepción de los educandos sobre sus habilidades de programación informática. Estos señalan este autoconocimiento del nivel adquirido en una pregunta basada en la Escala de Likert que tiene como valor 1 “Se me da muy mal” y como máximo el 5 que tiene como etiqueta “Se me da muy bien”. Un total de 3 estudiantes sitúan su percepción en tres puntos, mientras que dos califican su mejora en 4 puntos y el resto se coloca en la última posición, en otras palabras, consideran que “se les da muy bien”.

En quinto lugar, el instrumento pregunta si se ha superado la totalidad del curso. En este sentido, toda la población ha dado una respuesta negativa, es decir, nadie ha resuelto la totalidad del curso tal y como se esperaba en la programación de la secuencia didáctica planteada en las páginas anteriores.

En sexto lugar, se trabaja la motivación de la muestra por continuar avanzando en los aspectos de programación educativa. En la cuestión no se detalla si desean progresar en un entorno educativo formal, no formal o informal. El 67% de los educandos (6 estudiantes) están “muy motivados” en continuar recibiendo educación en esta materia. Dos alumnos están “bastante” motivados en continuar esta formación y sólo una persona, se sitúa “algo” motivada.

En séptimo lugar, los participantes indican si les gustaría continuar aprendiendo programación educativa de manera online a su propio ritmo de aprendizaje y de manera interactiva. Atendiendo a las respuestas facilitadas, un total de 7 de los 9 participantes, el 78%, ofrecen una contestación positiva.

En octavo lugar, se pregunta el grado de agrado mediante otra escala de valores Likert: el 67% de los participantes (6 personas) indican que les ha gustado “mucho” el curso, un 22% de ellos (2 educandos) les ha gustado “bastante” y la persona restante (11%) le ha gustado “algo”.

En noveno lugar, siendo la última cuestión de carácter cuantitativo, se valora el aprendizaje realizado con el curso que se ha hecho mención anteriormente. En este sentido, se vuelve a utilizar la Escala de Likert con 5 valores, siendo el mínimo “No he aprendido nada” y el máximo “He aprendido mucho”. Un total de 5 educandos (56%) indican “haber aprendido mucho” marcando la quinta opción, 2 más (22%) señala “haber aprendiendo bastante” seleccionando la cuarta opción, mientras que 2 usuarios (22%) tienen la percepción de “haber aprendido algo” marcando la tercera opción.



En décimo lugar, acorde a las etapas que más han gustado y las que más han desagradado de la plataforma Code.org, se han categorizado las aportaciones de los educandos en cuatro categorías: Actividades que más han gustado, actividades que menos han gustado, contenidos que más han gustado y contenidos que menos han gustado.

En la categoría de actividades que más han interesado, se han agrupado los términos de “Abeja”, “Ice Age”, “Angy Birds” y “Zombies”. En cuanto a los contenidos que más han satisfecho a los participantes han sido “Los iniciales” y los “bucles”.

Acorde con esta categorización de los datos, se ha elaborado una nube de palabras (Figura 2) con los motes más empleados según Atlas Ti 8.0 sobre los aspectos que han apreciado más los estudiantes según la frecuencia en que estos se expresan. Se observa que los términos “bucle”, “zombi” y sus respectivos plurales se repiten sucesivamente. Asimismo, también se aprecia palabras como “abeja”, “miel”, “polen” y “flor” que hacen referencia a una misma tipología de pantalla. Además, el contenido del bucle también muestra una alta frecuencia. Algunas otras palabras con menor repetición son: “Angry Birds”, “Pintor” y “Ice Age”.



Figura 2: Nube de palabras de los términos valorados en mayor grado

En cambio, en la categoría de actividades que menos han interesado se han agrupado los términos “Artista”, “Lápiz” y “Pintar” que hacen referencia a la “Etapa 3: Artista” del curso, que representan a las últimas actividades de Code.org realizadas. En cuanto a los contenidos que menos han satisfecho a los participantes han sido “Dibujar”, “Loops”, y “Bucles”.

A continuación, concorde a categorización presentada, se detallan los vocablos utilizados en las proposiciones que aluden a las etapas y contenidos peor valorados por los participantes (Figura 3).



Figura 3: Nube de palabras de los términos valorados en menor grado

En undécimo y último lugar, se han categorizado las aportaciones de los educandos de esta pregunta final en dos categorías: valores y acciones. Como valores se han extraído los siguientes: “Importante”, “Futuro”, “Necesidad” y “Conocimiento” mientras que las acciones que se recogen son: “Aprender”, “Crear”, “Conducir”, “Cocinar”, “Enviar”, “Jugar” y “Limpiar”. Se observa una apreciación de la codificación como una destreza clave para el futuro y en distintas ocasiones, se establecen relaciones entre la programación y las tareas domésticas. A modo de ejemplo se reproducen las siguientes respuestas:

- *“Sí, porque los robots nos ayudarán en las cosas de casa como cocinar o limpiar.”*
- *“Sí, porque los robots nos pueden ayudar a limpiar la casa y nos pueden hacer la comida y la cena.”*
- *“Sí, porque si un robot se estropea lo podremos reprogramar. Seguramente los robots conducirán, harán el trabajo de casa y otras muchas cosas más.”*

Las palabras surgidas de esta técnica están en relación con la consideración que tienen los educandos sobre si entienden la programación informática como un aspecto importante para la formación de la ciudadanía actual. Estos términos se reúnen en una tercera nube de palabras según la frecuencia en las que se repiten (Figura 4).



principio de la prueba y hacen referencia a los diferentes laberintos que se deben resolver de manera visual con el uso de bloques de programación, en estos interrogantes iniciales se trabaja el concepto computacional de la dirección, y en el caso de las preguntas 6 y 9 también se utilizan los conocimientos de bloques de repetición (Román, 2016).

En cuanto a los conocimientos sobre condicionales simples, en la pregunta 4 se ha experimentado una evolución de cinco puntos sobre nueve cuando en sus inicios no consiguió ninguno. Un hecho similar sucede con el interrogante 16 que aborda el mismo contenido, el condicionante, que comienza con ningún acertante y finaliza con un total de 4. En el caso de las otras dos preguntas sobre condicionales, en la 13 se mantiene el mismo nivel que en el inicio (8 acertantes) y la pregunta 15 pierde un punto. Por lo que se concluye que hay una leve mejora en el aprendizaje de los condicionales simples.

El resto de los contenidos no muestran una evolución significativa. Este hecho seguramente es debido a que, tal y como se expone en la planificación de la intervención, solamente se trabajan en el aula los contenidos de direcciones, bucles y condicionales simples.

Este grado de mejora en la respuesta de las preguntas iniciales del test coincide con la relación entre el pensamiento matemático-abstracto que suponen las preguntas del primer instrumento con el conocimiento pragmático-ingenieril que representan los bloques de programación (Valverde et al., 2015) y la relación de los conocimientos generados por otras situaciones similares experimentadas en el desarrollo de la secuencia didáctica (Zúñiga y Hurtado, 2014).

Sin embargo, la estancamiento en el grado de acierto del resto de los contenidos concuerda con la imposibilidad de relacionar estos conocimientos con actividades previas puesto que no se completó la totalidad del curso de Code.org tal y como queda reflejado en las respuestas obtenidas en la sexta pregunta en la que los educandos señalan que no han completado el 100% de las etapas de la plataforma.

Cuanto al segundo y tercer objetivo, los educandos han valorado positivamente el trabajo de las diferentes capacidades y actitudes. Siendo estas -clasificadas según valoración ascendente- la persistencia, la resolución de problemas, la creatividad, el pensamiento lógico, el trabajo en equipo y la capacidad de atención como valores promovidos en esta secuencia de actividades.

Además, a través del análisis cualitativo de los datos, se interpreta una fuerte relación entre lo aprendido durante la realización del curso y el contexto tecnológico-digital cotidiano de los niños y de las niñas ya que en distintas ocasiones, se establecen relaciones entre las dinámicas de codificación realizadas en el casal de verano y las tareas domésticas, aspecto muy relacionado con el precepto propio de la Competencia de Pensamiento Computacional como herramienta a fin de dar respuestas eficaces a situaciones cotidianas con el uso de recursos propios de la computación (Valverde et al., 2015).



Considerando las apreciaciones de las capacidades, actitudes y la contextualización de estos aprendizajes en el ámbito cotidiano por parte del alumnado, con el presente estudio se añade un pequeño grano de arena en propiciar la idea de que este tipo de acciones educativas pueden colaborar en el estímulo de actitudes positivas hacia la ciencia y la tecnología e incluso, implicar una evolución de ciertas actitudes personales como es en el caso del trabajo en equipo con el uso de este tipo de recursos (Pittí et al., 2014).

Simultáneamente, la autopercepción de los educandos en sus capacidades en PC tras la realización de las dinámicas es positiva ya que la mayoría se consideran muy buenos y se muestran muy motivados por continuar formándose en este ámbito. Esta valoración puede contribuir al uso de los avances de la codificación educativa en las propuestas de educación no formal y estimular la creación de espacios de aprendizaje lúdicos y cordiales que promuevan la Competencia de Pensamiento Computacional como eje transversal (Castro et al., 2017) y así enriquecer el desempeño de las actividades de tiempo libre con el uso de las tecnologías digitales (Cañellas, 2005).

En este estudio, se han presentado una serie de limitaciones significativas. Una de las más relevantes ha sido la medida de esta dado que durante toda la intervención sólo han tomado parte de la experiencia 9 participantes. No obstante, los datos recogidos pueden ser de interés para futuros proyectos de investigación aunque en la actualidad no puedan generalizarse debido a la magnitud de la participación.

Esta muestra ha sido inferior a lo esperado porque se ha perdido una parte significativa de la muestra por motivos de logística en la inscripción de los niños y de las niñas en las diferentes semanas del casal. Este aspecto debería ser considerado con anterioridad en futuras acciones en las cuales se deba desarrollar una secuencia didáctica en un casal de vacaciones de verano a “largo plazo”, considerando como “largo plazo” más de ocho sesiones repartidas en diferentes semanas.

Otros proyectos enmarcados en casales de verano también han experimentado la misma dificultad a la hora de mantener un grupo con una participación constante (López, 2012), sea por la infravaloración de este tipo de educación (Ruiz, 2012) y que no se perciba la participación constante como un valor intrínseco en la calidad de este tipo de educación, o bien, las posibles dificultades económicas de los entornos familiares a fin de costear el gasto de las actividades y los recursos si no están subvencionados con fondos públicos (Vega-Moreno et al., 2016). En este sentido, la muestra podría verse menos alterada en otras actividades de educación no formal como es el caso de las actividades extraescolares, los *Code Clubs* y los festivales de programación y de robótica educativa.

Asimismo, se ha de especificar que una segunda limitación ha consistido en la existencia de un gran vacío sobre cómo evaluar el grado de Competencia de Pensamiento Computacional del alumnado (Román, 2016). De esta manera, aunque la muestra es muy reducida, se emplea TPC del Dr. Román (2016) y el cuestionario asociado al recurso de Code.org por su alto grado de fiabilidad. Además,



se apuesta por complementar la información de estos datos cuantitativos con otros de carácter cualitativo.

Finalmente, atendiendo a estas dificultades, algunas propuestas para futuras investigaciones pueden consistir en el diseño de nuevos instrumentos y validación de estos para evaluar el grado de Competencia de Pensamiento Computacional, el uso de otra clase de recursos computacionales en los casales de vacaciones de verano y el estudio del impacto de las acciones de programación y de robótica educativa en otros entornos propios de la educación no formal.

## 5. ENLACE

El enlace que se muestra a continuación hace referencia a la propuesta de secuencia didáctica desarrollada y evaluada durante las diferentes sesiones del casal de vacaciones de verano. Estas actividades didácticas se basan en el recurso educativo Code.org y emplean recursos analógicos y digitales de programación educativa.

<https://view.genial.ly/5e5cd874c1a7970fc7cf9e98/vertical-infographic-pc-ed-no-formal>

## 6. REFERENCIAS

Álvarez, M. (2017). Desarrollo del pensamiento computacional en educación primaria: una experiencia educativa con Scratch.. *Universitas Tarraconensis. Revista de Ciències de l'Educació*, 1(2), 45-64. doi: <https://doi.org/10.17345/ute.2017.2.1820>

Basogain, X., Olabe, M.A. y Olabe, J.C., (2015). Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje. RED. Revista de Educación a Distancia. Número 46. 30 de Septiembre de 2015. Consultado el (16/02/2020) en <http://www.um.es/ead/red/46>

Boix-Tormos, J. J. (2016). Estudio de la influencia del aprendizaje del pensamiento computacional en las materias de ciencias en alumnos de secundaria. Tesis Doctoral. Universitat Oberta de Catalunya.

Castro, F. V., Briegas, J. J. M., Ballester, S. G., y González, D. V. (2017). Actividad extraescolar para aprender a aprender: la robótica como herramienta educativa. *Revista de estudios e investigación en psicología y educación*, (13), 124-128.

Cañellas, A. (2005). Continuidad y complementariedad entre la educación formal y no formal. *Revista de educación*, (338), 9-22

Codeclub.org. (2020). Página rincipal. Recuperado el 25 de mayo 2020, de <https://codeclub.org>



- Decreto 119/2015, de 23 de junio, de ordenación de las enseñanzas de la educación primaria. Diari Oficial de la Generalitat de Catalunya (23 de junio de 2015). núm. 6900. Recuperado de <http://portaldogc.gencat.cat/utillsEADOP/PDF/6900/1431927.pdf>
- Decreto 267/2016, de 5 de julio, de las actividades de educación en tiempo libre en las que participan menores de 18 años. Diari Oficial de la Generalitat de Catalunya (23 de juny de 2015). núm. 7157. Recuperado de <http://portaldogc.gencat.cat/utillsEADOP/PDF/7157/1513060.pdf>
- Karp, T., y Maloney, P. (2013). Exciting Young Students in Grades K-8 about STEM through an Afterschool Robotics Challenge. *American Journal of Engineering Education*, 4(1), 39-54.
- López Martínez, U. (2012). Los nuevos Guernicas, reflexiones en torno al trabajo con jóvenes productores de cultura visual. Edarte, Grupo de investigación (ed.)(2013). Investigar con jóvenes:¿ Qué sabemos de los jóvenes como productores de cultura visual? Pamplona: Pamiela–Edarte (UPNA/NUP).
- Pérez, Z. P. (2011). Los diseños de método mixto en la investigación en educación: Una experiencia concreta. *Revista electrónica educare*, 15(1), 15-29.
- Picas, N. (2017). Introducció de la robòtica a l'ESO. Universitat Politècnica de Catalunya. Recuperado de <http://hdl.handle.net/2117/110878>
- Pittí, K., Curto, B., Moreno, V. y Rodríguez, M. J. (2014). Uso de la Robótica como herramienta de aprendizaje en Iberoamérica y España. *VAEP-RITA*, 2(1), 41-48.
- Román, M. (2016). Código alfabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de programas. Tesis doctoral. UNED.
- Ruiz, R. (2012). L'educació en el lleure a l'inici del segle XXI a Catalunya. *Educació social. Revista d'intervenció socioeducativa*, (50), 69-79
- Smith, N., Sutcliffe, C. y Sandvik, L. 2014. Code club: Bringing programming to UK primary schools through scratch. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE'14)*. Nueva York: ACM, 517–522
- Strawhacker, A., y Bers, M. U. (2015). "I want my robot to look for food": Comparing kindergartner's programming comprehension using tangible, graphic, and hybrid user interfaces. *International Journal of Technology and Design Education*, 25(3), 293–319.
- Sullivan, A., Bers, M.U.yMihm, C. (2017). Imagining , Playing , and Coding with KIBO : Using Robotics to Foster Computational Thinking in Young Children. En S. C. Kong, J. Sheldony K. Y. Li (Eds.), *Conference Proceedings of International Conference on Computational Thinking Education 2017*(pp. 110–115). Hong Kong: The Education University of Hong Kong.



- Torrecilla, F. (2011). Investigación acción. Métodos de investigación en Educación Especial. 3ª Educación Especial. Curso, 2011. Recuperado de: [https://www.uam.es/personal\\_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Curso\\_10/Inv\\_accion\\_trabajo.pdf](https://www.uam.es/personal_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Curso_10/Inv_accion_trabajo.pdf)
- Valverde, J., Fernández, M. R., y Garrido, M. del C. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. Revista de Educación a Distancia (RED). <https://doi.org/10.6018/red/46/3>
- Vega-Moreno, D., Cufí, X., Rueda, M<sup>a</sup>. J, y Llinás, D. (2016). Integración de robótica educativa de bajo coste en el ámbito de la educación secundaria para fomentar el aprendizaje por proyectos. International Journal of Educational Research and Innovation (IJERI), 6, 162-175. Recuperado de <https://www.upo.es/revistas/index.php/IJERI/article/view/1653/1564>
- Vidal, C. L., Cabezas, C., Parra, J. H., y López, L. P. (2015). Experiencias prácticas con el uso del lenguaje de programación scratch para desarrollar el pensamiento algorítmico de estudiantes en Chile. Formación Universitaria, 8(4), 23-32. <https://doi.org/10.4067/S0718-50062015000400004>
- Wing, J.M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J.M. (2010). Computational Thinking: What and Why? Disponible el 16 de febrero de 2020 en <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
- Zabala, A. y Arnau, L. (2007). La enseñanza de las competencias. Aula de Innovación Educativa, 161, 40-46.
- Zuñiga, R. y Hurtado, J.(2014). Comprendiendo procesos de abstracción computacional en los niños. Un estudio de caso. 10.13140/RG.2.1.2992.3363.

#### Para citar este artículo:

Zarza González, M., y Holgado García, J. (2020). Competencia de pensamiento computacional en la educación no formal. Edutec. Revista Electrónica De Tecnología Educativa, (72), 68-87. <https://doi.org/10.21556/edutec.2020.72.1575>

