

Desarrollo del pensamiento computacional en educación primaria: una experiencia educativa con scratch

UTE. Revista de Ciències de l'Educació

2017 núm. 2. Pag. 45-64

ISSN 1135-1438. EISSN 2385-4731

<http://revistes.publicacionsurv.cat/index.php/ute>



DOI: <http://dx.doi.org/10.17345/ute.2017.2.1820>

Marian Álvarez Rodríguez

Rebut: 11/09/2017 Acceptat: 04/12/2017

Resumen

Este estudio pretende evaluar la factibilidad del desarrollo del Pensamiento Computacional (PC) de alumnos de 6º de Primaria, que carecen de conocimientos computacionales previos al estudio, a través de proyectos realizados con la herramienta Scratch. Para ello se ha trabajado en el diseño y la elaboración de varias actividades que conforman la Guía de Iniciación a Scratch. Finalmente se ha evaluado el PC mediante el "Test de Pensamiento Computacional" (TPC) diseñado y verificado por Román-González (2016), así como con otros instrumentos de evaluación como son las tablas de reflexión que contiene la propia Guía y la plataforma Dr. Scratch. Los resultados obtenidos son positivos, sobre todo en lo referente a la adquisición de conceptos computacionales que establecen la base del PC, tales como la repetición de programas, las direcciones o las funciones simples, lo que ha permitido obtener conclusiones favorables al estudio.

Palabras clave: Pensamiento Computacional, Programación Computacional, Scratch, Educación Primaria.

Abstract

This research expects to evaluate the development of Computational Thinking with students of 6º grade of Primary Education, who lack of previous knowledge of this concepts, after working with the Scratch tool. To do so, we have worked on the design and elaboration of various activities that form the index Guide of Scratch. Finally, we have evaluated CT through the "Computational Thinking Test" designed and verified by Román-González (2016), in the same way that the tables thoughts that contains the own guide and platform Dr. Scratch. The results obtained have been positive, especially in reference to the acquisition of computational concepts that establish the base of the CT, such as repetition of programmes, directions or simple functions, therefore obtaining positive results about the CT.

Keywords: Computational Thinking, Computer Programming, Scratch, Primary Education.

Introducción

A lo largo de las últimas décadas, la programación computacional ha estado muy presente en la educación básica dependiendo de las necesidades sociales del momento. Actualmente está adquiriendo gran importancia debido a la evolución de las nuevas tecnologías, creándose así, una tendencia mundial que considera la programación en el aula como una actividad fundamental del presente y del futuro para el desarrollo de diferentes competencias fundamentales, relacionadas con la realidad del mundo laboral y personal de los estudiantes (Vázquez-Cano y Ferrer, 2015).

La programación pretende desarrollar el pensamiento computacional, siendo esta una competencia compleja de "alto nivel" relacionada con un modelo de conceptualización específica de los seres

humanos que desarrolla ideas y está vinculada con el pensamiento abstracto-matemático y con el pragmático-ingenieril que se aplica en múltiples aspectos de nuestra vida diaria (Valverde, Fernández y Garrido, 2015).

Es importante que los alumnos de Educación Primaria y Secundaria desarrollen su Pensamiento Computacional, para ello deben entender que los ordenadores automatizan y agilizan los procesos con el fin de solucionar los problemas de forma eficiente (Cearreta-Urbieta, 2015). Pero esto no significa que se deba basar su aprendizaje en escribir código de forma compulsiva, lo importante es saber cómo representar la realidad, el mundo de los objetos y expectativas, pues al fin y al cabo son los que piensan los que tienen éxito en hacer programas potentes. El principio básico del "pensamiento computacional" es conocer el mundo de ideas y de representaciones y cómo operan. Además, si se aprende a programar asociado a un lenguaje es posible que no se produzca la transferencia y en un futuro no se pueda repetir el proceso (Zapato-Ros, 2015).

De cara a un futuro, Wing (2011) afirma que a mediados del siglo XXI el Pensamiento Computacional será una destreza fundamental usada por todo el mundo, así como lo son ahora otras destrezas como leer, escribir y la aritmética.

Como consecuencia a esta necesidad, cada vez son más las herramientas que trabajan la programación informática, aunque en la actualidad la herramienta más destacable es Scratch. Se trata de una herramienta que permite programar objetos a través de bloques con diferentes eventos, evitando así la posibilidad de cometer errores en la escritura (Resnick et al., 2009). Además, Scratch contribuye al desarrollo del PC, tal y como afirman los creadores de la herramienta a través de un estudio realizado en su propio laboratorio de investigación (Brennan y Resnick, 2012).

El proyecto Scratch se inició en el año 2003. Se trata de un entorno de programación informática creado por un grupo de investigadores de Lifelong Kindergarten del Laboratorio de Medios del MIT, (Resnick et al., 2009) bajo la dirección y liderazgo del Dr. Michael Resnick. El objetivo de Scratch es crear productos multimedia como vídeos musicales, presentaciones, juegos de ordenador y otro tipo de animaciones y compartirlos con toda la comunidad que accede a la plataforma. Esto favorece el crecimiento de la creatividad de los niños y niñas además de introducirlos en la programación de forma divertida (Wilson y Moffat, 2010).

Vázquez-Cano y Ferrer (2015) afirman que la idea principal de Scratch usado para fines educativos debe ser que el lenguaje de programación facilite el juego y se puedan probar con facilidad diferentes opciones de creación multimedia. Por ello, el lenguaje de programación Scratch no requiere escribir líneas de programación, evitando así los errores al teclear.

Durante los últimos años, investigadores del ámbito de la educación han tratado de elaborar y verificar el instrumento de evaluación idóneo que consiga determinar el grado de PC de sus sujetos como es el caso de Fairy Assessment in Alice (Wener, Denner, Campe y Kawamoto, 2012). Entre ellos cabe destacar el Test de Pensamiento Computacional (TPC) de Román-González (2016), pues será el instrumento de evaluación principal de este estudio. Este instrumento de evaluación consiste en un Test de Pensamiento Computacional específicamente diseñado para población española de primer ciclo de la Educación Obligatoria (1º ESO – 2º ESO), pero con aplicaciones complementarias en los últimos niveles de Educación Primaria (5º Primaria – 6º Primaria). Se trata de una prueba objetiva de elección múltiple con 4 opciones de respuesta (sólo 1 correcta), el tiempo estimado de realización es de 45 minutos y está formado por 40 ítems.

Se ha procedido a la elaboración de una revisión sistemática sobre el estado de la cuestión en la actualidad, para obtener una visión amplia sobre los resultados obtenidos en diferentes estudios que pretenden investigar el desarrollo del Pensamiento Computacional

1.1. *Método*

En primer lugar, se llevó a cabo una búsqueda de artículos de revistas científicas en las bases de datos Scopus, Web of Science, Dialnet, Teseo y Google Académico, con las palabras clave siguientes: programación, educación primaria, Scratch y pensamiento computacional.

Además, se rastreó la bibliografía de aquellos artículos que trataban directamente sobre el Pensamiento Computacional, así como los que estaba relacionados con la herramienta Scratch, con el fin de localizar nuevos artículos.

De los resultados obtenidos se descartaron aquellos que no cumplieran con alguno de los siguientes criterios:

- Presentar resultados de un trabajo empírico.
- Incluir a un grupo de participantes de edades comprendidas entre los 7 y 18 años de edad.
- Haber sido publicados entre los años 2006-2017.
- Estar redactados en español o inglés.

Tras establecer estos criterios, se obtuvieron en total 27 artículos. Una vez fueron localizados, se procedió a su lectura y a su posterior análisis de los resultados.

1.2. Resultados

La búsqueda en las bases de datos arrojó un total de 27 artículos, de los cuales 5 procedían de la base de datos Scopus, 16 de Web of Science, 1 de Teseo, 2 de Dialnet y 3 de Google Académico. De éstos únicamente se tuvo acceso a 19 y sólo 11 cumplieron con los criterios de inclusión propuestos, descartando 7 de los artículos obtenidos.

A continuación, se muestran los resultados obtenidos, especificando el nombre del autor, la muestra, el método del estudio, instrumentos de evaluación y los principales resultados de los estudios seleccionados.

Autores	Muestra	Método	Instrumentos	Principales resultados
Wilson, A. y Moffat, D. C. (2010).	N= 12. <i>Edad:</i> 8-9 años. <i>Etapas educativas:</i> Ed. Primaria. <i>Procedencia:</i> Glasgow, Escocia.	Durante las 8 semanas del estudio se empleó 1h semanal. Al comienzo de la sesión el maestro explica el trabajo y en parejas lo desarrollan durante la sesión. Cada semana se incrementa la dificultad. Los sujetos cumplimentan un pre-test y un pos-test.	- Pre-test. - Post-test. - Observación.	En cuanto al aprendizaje no se obtuvo una diferencia significativa entre los resultados del test 1 (52%) y del test 3 (64%). Los resultados en cuanto a afectividad son muy positivos.

López García, J. C. (2014).	<p>N= 66. <i>Edad:</i> 7-10 años. <i>Etapa educativa:</i> Ed. Primaria. <i>Procedencia:</i> Cali, Colombia.</p>	<p>El estudio duró un curso académico. Al principio del curso se midió el pensamiento computacional de los alumnos, creando tres grupos (básico, intermedio y avanzado). Se llevó a cabo el mismo procedimiento al final del curso.</p>	<p>- Instrumento de medición del pensamiento computacional. - Grabaciones en video. - Imágenes de las plantillas de análisis.</p>	<p>Se obtuvieron resultados muy positivos. Entre la aplicación inicial y la aplicación final de la prueba se apreció una disminución de 4.10% en el nivel básico, un descenso de 2.46% en el nivel intermedio y un aumento de 6.56% en el nivel avanzado.</p>
Morrás Aranoa, H. (2014).	<p>N= 21 <i>Edad:</i> 8-9 años. <i>Etapa educativa:</i> Ed. Primaria <i>Procedencia:</i> Glasgow, Escocia</p>	<p>Durante la sesión de informática de 8 semanas se trabajó con la herramienta Scratch. Se evaluó el proceso, el funcionamiento, la Interfaz y la programación.</p>	<p>- Rúbrica de evaluación.</p>	<p>Los resultados obtenidos fueron satisfactorios. El alumnado mostró cierta mejora cognitiva respecto a la clase de informática usual, aunque en pocas sesiones es difícil ver una mejoría significativa.</p>
Morrás Aranoa, H. (2014).	<p>N= 21 <i>Edad:</i> 8-9 años. <i>Etapa educativa:</i> Ed. Primaria <i>Procedencia:</i> Glasgow, Escocia</p>	<p>Durante la sesión de informática de 8 semanas se trabajó con la herramienta Scratch. Se evaluó el proceso, el funcionamiento, la Interfaz y la programación.</p>	<p>- Rúbrica de evaluación.</p>	<p>Los resultados obtenidos fueron satisfactorios. El alumnado mostró cierta mejora cognitiva respecto a la clase de informática usual, aunque en pocas sesiones es difícil ver una mejoría significativa.</p>
Cearreta- Urbieto, I. (2015).	<p>N=10 <i>Edad:</i> 16-17. <i>Etapa educativa:</i> Bachillerato: <i>Procedencia:</i> País Vasco, España.</p>	<p>Sigue metodología ex post-facto. Recogida de información de alumnado, profesores e investigadores sobre el pensamiento computacional desde tres dimensiones: conceptos, prácticas y perspectivas computacionales.</p>	<p>- Cuestionarios. - Análisis de documentos.</p>	<p>Resultados satisfactorios, Scratch ayuda a los sujetos a desarrollar el pensamiento computacional, aunque los sujetos tienen dificultades para percibir esta mejora.</p>

Autores	Muestra	Método	Instrumentos	Principales resultados
Moreno-León, J. Robles, G. y Román-González, M. (2015).	<i>N</i> = 109. <i>Edad</i> : 10-14. <i>Etapa educativa</i> : Ed. Primaria y Secundaria. <i>Procedencia</i> : España.	Se comprobó si Dr. Scratch (herramienta para analizar proyectos de Scratch), era útil para alumnos que previamente había trabajado con la herramienta Scratch. Los sujetos debían analizar sus proyectos con dicha herramienta.	- Cuestionarios.	La mayoría de los sujetos respondió que Scratch es una herramienta atractiva y de fácil manejo. Se obtuvo que la mayoría de los sujetos adquiriría habilidades de programación mediante la realización de proyectos con Scratch. En cuanto a Dr. Scratch, finalmente se concluyó con que la herramienta permitía analizar los proyectos a la mayoría de los sujetos.
Vázquez-Cano, E. y Ferrer, D. (2015).	<i>N</i> = 24. <i>Edad</i> : 16-17. <i>Etapa educativa</i> : Bachillerato. <i>Procedencia</i> : España.	El método de trabajo se desarrolla a través de un aula virtual con apuntes, videos, fotos y recursos de descarga para llevar a cabo la creación de un videojuego con Scratch.	- Observación.	Los sujetos han conseguido elaborar un videojuego mediante Scratch desarrollando diferentes competencias entre las que destacan la expresión y la creatividad.
Vidal, C. L., Cabezas, C., Parra, J. H, y López, L. P. (2015).	<i>N</i> = 27. <i>Edad</i> : 14-16. <i>Etapa educativa</i> : Ed. Secundaria. <i>Procedencia</i> : Chile.	Se inició a los sujetos en el uso de Scratch. Se les planteó casos hipotéticos que debían ser solucionados con Scratch. Esta acción se realizó al principio, durante y al final del estudio.	- Pre-test. - Post- test.	La experiencia muestra que Scratch constituye una herramienta propicia para el desarrollo del pensamiento lógico y algorítmico para niños y estudiantes.
Basogain, X. Olabe, M. A. Olabe, J. C. Ramírez, R. Del Rosario, M.	<i>N</i> = 32 <i>Edad</i> : 6-12 años <i>Etapa educativa</i> : Ed. Primaria.	Durante 6-10 meses se llevó a cabo un curso sobre pensamiento computacional mediante la plataforma "Egelapi".	- Test. - Entrevistas.	Los resultados obtenidos de los test muestran un alto grado de éxito. Las entrevistas indican gran nivel de satisfacción entre profesores y alumnos.

y García, J. (2016).	<i>Procedencia:</i> Santo Domingo, República Dominicana.			
Galindo Suárez, M. (2016).	<i>N=</i> 62. <i>Edad:</i> 10-11. <i>Etapa educativa:</i> Ed. Primaria. <i>Procedencia:</i> Colombia.	Durante una sesión de 2 horas, el grupo control asistió a una clase de matemáticas mediante Scratch y el grupo experimental asistió a una clase de matemáticas con metodología tradicional. Al finalizar la sesión los sujetos realizaron un test.	- Test.	Los resultados indican que el uso Scratch resulta muy atractivo para los estudiantes, por lo que para las matemáticas se obtuvieron mejores resultados de aprendizaje con el software de programación mencionado.
Autores	Muestra	Método	Instrumentos	Principales resultados
Román González, M. (2016).	<i>N=</i> 1251. <i>Edad:</i> 10-16. <i>Etapa educativa:</i> Ed. Primaria y Secundaria. <i>Procedencia:</i> España.	Los estudios se llevaron a cabo durante un trimestre con alguna diferencia temporal entre centros. Se realizó un pre-test y un post-test para conocer el grado de pensamiento computacional de los sujetos.	- Test de pensamiento computacional (TPC)	El TPC presenta un grado de dificultad adecuado-medio. El rendimiento se incrementa en función del curso de manera significativa. Los chicos muestran un rendimiento superior que las chicas. Es una herramienta óptima para evaluar el Pensamiento Computacional en alumnos de Ed. Primaria y Secundaria
Sáez-López, J. M., Román-González, M., y Vázquez-Cano, E. (2016).	<i>N=</i> 107. <i>Edad:</i> 10-12. <i>Nivel educativo:</i> Ed. Primaria. <i>Procedencia:</i> España.	Durante 21 sesiones, se llevó a cabo una actividad con Scratch para evaluar dos dimensiones, creatividad computacional y el proceso de aprendizaje y actitudes, a través de un pre-test y un post-test.	- Pre-test. - Post- test. - Observación.	Se encontraron mejoras significativas en el post-test en cuanto a la creatividad computacional. El uso de la herramienta Scratch mejora capacidades como la lógica, el aprendizaje de conceptos de programación y la práctica computacional.

Diseño

Descripción

A medida que pasan los años, la sociedad progresa y con ello sus necesidades, así como el ámbito de la educación, que acompaña a las necesidades del momento. Los niños del presente deben estar preparados para afrontar todo tipo de problemas del futuro y resolverlos con agilidad. Con todo tipo de problemas nos referimos no sólo a los problemas de la vida cotidiana, si no a los relacionados con la realidad ficticia, la programación.

Desarrollar destrezas computacionales, como es el caso del PC, va más allá de la mera adquisición de conceptos computacionales propios de la programación, también consiste en desarrollar habilidades de pensamiento lógico. Se trata de entender la realidad desde una perspectiva computacional, donde todo tiene un porqué matemático, el cual se resuelve a través del razonamiento

Por ello, cada vez es más frecuente trabajar en las aulas de etapas educativas tempranas con herramientas de programación computacional. En España la herramienta que más se utiliza en los centros educativos es Scratch, destinada a estudiantes mayores de 8 años.

Objetivo

Lo que se pretende comprobar en este estudio es si la herramienta de programación Scratch, permite desarrollar el Pensamiento Computacional de los sujetos después de haber trabajado con ella en diferentes proyectos.

El objetivo que se plantea como general de este estudio es el siguiente:

- Evaluar el Pensamiento Computacional en sujetos de 11-12 años tras trabajar con Scratch para comprobar si la herramienta permite desarrollar el PC.

Enfoque

Este estudio ha intentado ser un proceso de evaluación sistemático y riguroso, dirigido a estudiar el Pensamiento Computacional de alumnos de edades tempranas que trabajan con la herramienta de programación Scratch.

Para el logro del objetivo definido en este estudio, el tipo de investigación seguida ha sido la investigación evaluativa, pues se ha convertido en los últimos años en una parcela muy relevante en la actividad investigadora en diversos ámbitos como el de la educación (Serrano y Prendes, 2015).

Como método de investigación se ha empleado el estudio de caso pues se ha considerado que implica un proceso de indagación sistemático y en profundidad (Martínez, 2006).

Respecto a la evaluación, el estudio se ha basado en el modelo de evaluación MIT-Harvard (Brennan y Resnick, 2012). Este modelo, propone tres aproximaciones para obtener una evaluación rigurosa del PC, siendo estas: análisis de portfolio-proyecto, entrevistas basadas en artefactos-objetos y escenarios-situaciones de diseño. Los autores recomiendan una combinación de las tres, ya que ninguna engloba todas las dimensiones.

Método

Los participantes con los que se llevará a cabo el estudio son tres alumnos de 11/12 años de edad, que actualmente se encuentran en el 6º nivel de Educación Primaria. Estos alumnos pertenecen a un pequeño pueblo de la provincia de Toledo, Castilla- La Mancha.

Se ha escogido a estos sujetos debido a la gran facilidad de acceso desde este estudio, pues son alumnos de la propia investigadora. Además, ninguno de ellos posee ningún tipo de conocimiento sobre programación, debido a que anteriormente no han trabajado con ninguna herramienta que desarrolle el pensamiento computacional, lo que permite obtener resultados claros y verídicos.

Teniendo en cuenta el objetivo que se pretenden alcanzar con este estudio, su enfoque y el tipo de estudio planteado, la metodología de trabajo se compone por cuatro fases.

- **Primera fase.** Elaboración de una guía de Scratch como recurso de aprendizaje.

- **Segunda fase.** Presentación de la guía al alumnado y actividades con la herramienta Scratch.
- **Tercera fase.** Evaluación del Pensamiento Computacional del alumnado mediante el TPC (Román-González, 2016).
- **Cuarta fase.** Análisis de los resultados obtenidos y elaboración de las conclusiones finales.

Primera fase. Elaboración de una guía de Scratch como herramienta de aprendizaje.

Para trabajar con la herramienta Scratch, se ha diseñado una guía de iniciación como recurso para los sujetos, además, su carácter evaluativo permite que se utilice como instrumento de evaluación complementario al TPC final.

Esta guía se ha inspirado en la creada por ScratchEd, la cual dispone de mayor número de actividades y reflexiones, desde un nivel básico hasta niveles muy avanzados de programación con Scratch.

La guía está compuesta por 5 unidades que a su vez están formadas por diferentes actividades las cuales van incrementando su dificultad, de esta forma la unidad 0 es la base para comenzar a manejar Scratch con actividades como explorar la herramienta y crearse una cuenta, y la unidad 4 corresponde a la creación de juegos que conllevan cierta dificultad

Segunda fase. Presentación de la guía al alumnado y actividades con la herramienta Scratch.

Como se ha referido en apartados anteriores, los sujetos no disponen de ningún tipo de conocimiento de programación, tampoco han trabajado nunca con la herramienta Scratch, por lo que es necesario partir de cero e ir incrementando la dificultad. Para ello utilizaremos la guía explicada en el apartado anterior durante un total de catorce sesiones de 45 minutos cada una.

Tercera fase. Evaluación del pensamiento computacional del alumnado mediante un post-test (TPC).

Tras finalizar la segunda fase, donde los sujetos han trabajado en diferentes experiencias con Scratch, se procede a evaluar su pensamiento computacional para comprobar si la herramienta ha contribuido a su desarrollo.

Para ello se empleará una sesión de 45 minutos, concretamente la sesión número quince. Cada sujeto deberá completar un Test de Pensamiento Computacional. Es importante tener en cuenta que los alumnos no pueden comunicarse entre ellos mientras que completan el test para asegurarnos de una mayor veracidad de los resultados.

Cuarta fase. Análisis de los resultados obtenidos y elaboración de las conclusiones finales.

Durante esta fase se han analizado los resultados obtenidos a través de los diferentes instrumentos de evaluación, a continuación, se especifica el proceso de recogida de datos de cada instrumento.

- *Guía de iniciación a Scratch:* se han recogido las reflexiones realizadas por cada uno de los sujetos tras cada una de las actividades de la guía. El análisis se ha llevado a cabo con una tabla que diferencia las cuestiones realizadas en los siguientes apartados:

Problemas: ¿Ha habido algo que no ha funcionado? ¿El qué?

Mejoras: ¿Qué se podría mejorar?

Dificultades: ¿Ha habido algo difícil de entender?

Favorito: ¿Qué es lo que más te ha gustado del proyecto?

Estos apartados han sido evaluados en función de tres variables: ninguno, programas (respuestas relacionadas con la programación de alguno de los objetos) y disfraces (respuestas relacionadas con los fondos y la selección y edición de objetos).

- *Dr. Scratch*: los proyectos elaborados por los sujetos han sido evaluadas por la plataforma <http://www.drscratch.org> la cual evalúa al instante el nivel de cada proyecto, pudiendo conseguir una puntuación numérica de 1 a 21 y un nivel básico, medio o avanzado.
- *Test de Pensamiento Computacional (TPC)*: el Test de Román-González (2016) corresponde al instrumento de evaluación principal de este estudio, por ello se han seguido los criterios de evaluación que el mismo autor especifica siendo estos: conceptos computacionales, entorno-interfaz del reactivo, estilo de la alternativa de respuesta, existencia de anidamiento y tarea requerida.
 - Conceptos computacionales: para la recogida de datos de los conceptos computacionales se han dividido las preguntas en función del concepto que trabajan: direcciones, bucles (repetir veces, repetir hasta), condicionales (condicional simple, condicional compuesto, mientras que), y funciones simples. En cuanto a las variables, se han tomado dos: conseguido (respuesta correcta) o no conseguido (respuesta incorrecta o no contestada).
 - Entorno-interfaz del reactivo: al igual que en los conceptos computacionales, la recogida de datos en función del entorno se ha dividido dependiendo de si la pregunta estaba basada en un laberinto o en un lienzo. Las variables que se han tomado han sido conseguido (respuesta correcta) o no conseguido (respuesta incorrecta o no contestada).
 - Estilo de las alternativas de respuesta: en este caso se han dividido las preguntas en función de si contenían una alternativa visual por flechas o visual por bloques. Las variables han sido conseguido (respuesta correcta) o no conseguido (respuesta incorrecta o no contestada).
 - Existencia de anidamiento: la recogida de datos respecto al anidamiento se ha basado en función de si la pregunta contenía anidamiento o no. Al igual que en el resto de indicadores, las variables que se han tomado han sido conseguido (respuesta correcta) o no conseguido (respuesta incorrecta o no contestada).
 - Tarea requerida: este indicador se ha analizado en función de si la pregunta requería secuenciación, completamiento o depuración. Las variables que se han tomado han sido conseguido (respuesta correcta) o no conseguido (respuesta incorrecta o no contestada).

Instrumentos de evaluación.

Desde este estudio se ha utilizado diferentes instrumentos de evaluación para responder a cada una de las aproximaciones propuestas por Brennan y Resnick (2012), considerándose herramientas complementarias al Test de Pensamiento Computacional de Román-González (2016).

- Dr. Scratch <http://www.drscratch.org/> (Moreno-León y Robles, 2014, 2015).

Los proyectos realizados por los sujetos serán analizados a través de la plataforma Dr. Scratch para conocer su nivel de programación y detectar posibles errores cometidos.

- Guía de iniciación a Scratch (Anexo I).

Esta guía está formada por proyectos de diferentes niveles y cuestiones que permiten reflexionar acerca de cada uno de los proyectos. La guía se ha diseñado basándose en la elaborada por Brennan, Balch y Chung (2014), miembros del equipo de investigación de ScratchEd de la Universidad de Harvard. Su carácter evaluativo proporciona información sobre la evaluación del PC de los sujetos a través de las respuestas a las cuestiones de reflexión.

La guía ha sido proporcionada a los sujetos en formato papel y formato digital a través de la dirección: <https://marialvarezrodri.wixsite.com/guiascratch>.

- Test de Pensamiento Computacional (Román-González, 2016) (Anexo II).

El TPC ha sido el instrumento de evaluación principal del estudio, aportando datos cuantitativos al análisis del desarrollo del Pensamiento Computacional del alumnado. Ha sido elegida como pilar de esta investigación ya que se trata de una herramienta validada que permite obtener datos con rigor y sistematicidad. El test se ha presentado a los sujetos desde el siguiente link: <http://goo.gl/IYEKMB> y estos han tenido una sesión de 45 minutos para completarlo.

Para la validación de este test han participado 20 expertos, para ello se les envió tres cuestionarios sobre el TPC y este en formato PDF. Durante dos semanas, los expertos completaron los cuestionarios obteniendo así resultados positivos en cuanto a la veracidad de este.

Resultados

Se han registrado las respuestas del TPC de los tres sujetos, diferenciando los ítems según el concepto computacional que implica. De esta forma se ha recogido que el concepto de funciones simples es el más desarrollado por los sujetos con un 58,33% de media de respuestas correctas. Por otro lado, los ítems sobre direcciones y repeticiones han obtenido un 50% de aciertos, siendo los ítems sobre condicionales los que menos respuestas correctas tienen con un porcentaje del 29,16%.

Dentro del concepto de repetición se ha encontrado una pequeña diferencia entre el concepto de repetir veces y el de repetir hasta siendo este último el que menor porcentaje de aciertos se ha recogido, un 41,66%. Los ítems de repetir veces han sido respondidos correctamente un 58,33%.

Por otro lado, se ha recogido información sobre tres conceptos condicionales: condicionales simples (if) con un porcentaje de aciertos del 25%, condicional compuesto (if/else) siendo los ítems de este los que menos respuestas correctas se han recogido, con un 16,66% y, por último, el concepto while, el cual ha obtenido el porcentaje más alto, un 33,33% de aciertos.

En la gráfica 1 se muestra una relación de las respuestas correctas de los ítems que trabajan cada uno de los conceptos.



Gráfica 1. Porcentajes de respuestas correctas del TPC por conceptos computacionales

a. *Direcciones.*

En cuanto a las respuestas de los ítems que trabajan las direcciones (ítem 1-4), el sujeto nº1 ha obtenido un 75% de aciertos, mientras que el sujeto nº2 ha respondido correctamente un 25% y el sujeto nº3 un 50%.



Gráfica 2. Respuestas correctas de los ítems 1-4 sobre direcciones

b. *Bucles (loops).*

- Repetir veces (repeat times).

Respecto a los ítems que incluyen repeticiones (5-8) el sujeto nº1 ha respondido correctamente a un 75% de ellos, a diferencia del sujeto nº2 y nº3 que han obtenido un 50%.



Gráfica 3. Respuestas correctas de los ítems 5-8 sobre repetir veces.

- Repetir hasta (repeat until).

Por otro lado, las respuestas sobre los ítems referidos a repeticiones hasta son muy diferentes entre el sujeto nº 1 y el nº2 obteniendo el primero un 75% de aciertos, y el segundo un 0%. El sujeto nº3 respondió correctamente a un 50% de las cuestiones.



Gráfica 4. Respuestas correctas ítems 9-12 sobre repetir hasta.

c. *Condicionales (conditionals).*

- Condicional simple (if).

El porcentaje de respuestas correctas de los ítems (13-16) que incluyen condicionales simples (if) de todos los sujetos es de un 25%.



Gráfica 5. Respuestas correctas de los ítems 13-16 sobre condicionales simples.

- Condicional compuesto (until).

Respecto a los ítems que contienen condicionales compuestos (until), el sujeto nº2 ha registrado un 50% de respuestas correctas, a diferencia de el sujeto nº1 y nº3 los cuales no han acertado correctamente a ninguno de los ítems.



Gráfica 6. Respuestas correctas de los ítems 17-20 sobre condicionales compuestos.

- Mientras que (while).

En cuanto a los ítems referidos al concepto computacional mientras que (while) los sujetos nº1 y nº2 han contestado correctamente a un 25% de los ítems, mientras que el sujeto nº3 ha obtenido un 50% de aciertos.



Gráfica 7. Respuestas correctas de los ítems 21-24 sobre condicionales mientras que.

d. *Funciones simples (functions).*

Los ítems sobre funciones simples (25-28) han sido respondido correctamente por los sujetos nº1 y nº3 en un 50%. Por otro lado, el sujeto nº2 ha obtenido un 75% de las respuestas correctas.



Gráfica 8. Respuestas correctas de los ítems 25-28 sobre funciones simples.

Las respuestas recogidas del TPC en función del entorno a nivel general son muy semejantes, pues los ítems que contenían laberintos (1-6, 8-11, 13-24, 27-28) han obtenido un 42,05% de respuestas correctas. Las respuestas correctas recogidas de las cuestiones que contenían lienzos (4, 7, 12, 25, 26) son un 33,33%.



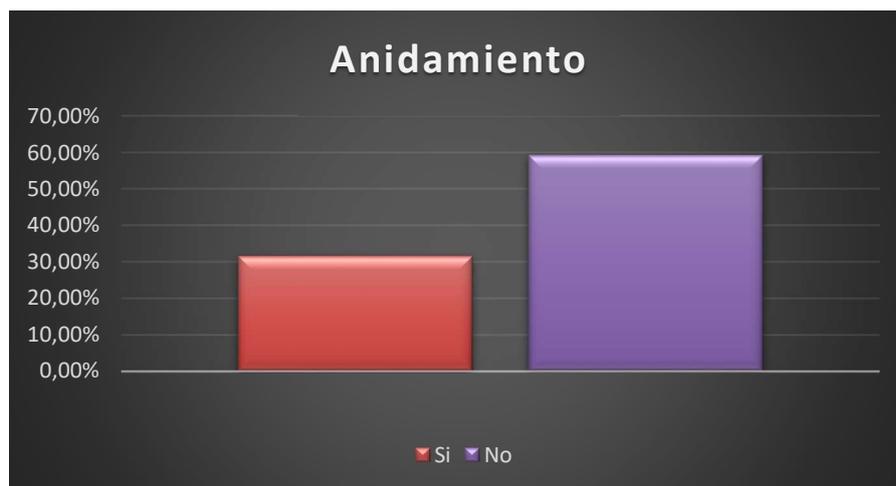
Gráfica 9. Porcentaje respuestas correctas en función de la interfaz del reactivo.

Respecto al estilo de alternativas, se ha recogido que los ítems que contenían un estilo visual por flechas tienen un mayor porcentaje de respuestas correctas 66,66% que los ítems con estilo visual por bloques, con un 30%



Gráfica 10. Porcentaje de respuestas correctas en función del estilo de las alternativas.

Los resultados en cuanto a los ítems que conteniendo anidamiento o no, no son positivos, pues los sujetos tan solo han contestado correctamente a un 31,57% de las cuestiones que incluía anidamiento, a diferencia de las preguntas donde no existía anidamiento, en las que se han registrado un 59,25%.



Gráfica 11. Porcentaje de respuestas correctas en función de la existencia de anidamiento.

Los datos registrados en general han sido respecto a la secuenciación un 26,19% de las respuestas han sido correctas, por otro lado, las tareas sobre completamiento han registrado un 62,96% de aciertos. En el caso de la depuración un 53,33%.



Gráfica 12. Porcentaje de respuestas correctas en función de la tarea requerida.

Conclusiones

Las conclusiones del estudio pretenden responder al objetivo principal del mismo, el cual desemboca en el desarrollo del Pensamiento Computacional de los sujetos a través de la herramienta Scratch.

A nivel general, los resultados no muestran un desarrollo claro del Pensamiento Computacional de todos los sujetos, ya que la media del TPC de Román-González (2016) giraba en torno a las 13 respuestas correctas, y este estudio ha estado en torno a las 11.

En cuanto a los sujetos, el sujeto número 1 destaca de forma significativa del resto. En el TPC ha obtenido un total de 13 respuestas correctas, destacando en la mayoría de conceptos computacionales tales como direcciones, bucles, condicional simple y funciones simples.

Las preguntas de la guía muestran que la mayoría de sus reflexiones están vinculadas a los bloques de programación, refiriendo apenas los disfraces, lo cual indica que el sujeto está comenzando a desarrollar su PC, pues las reflexiones que aporta son propias del proceso de programación.

El sujeto nº 2 consiguió 11 respuestas correctas, lo cual no permite confirmar con certeza si el sujeto ha desarrollado su PC, aunque si se puede afirmar que se ha observado una mejora en cuanto al principio del estudio. Este sujeto muestra los mismos resultados en cuanto a la adquisición de conceptos computacionales, destacando en las funciones simples y los bucles (repetir veces), dato significativo que permite percibir que el sujeto ha desarrollado su PC respecto al principio, aunque de una forma muy básica.

Los resultados obtenidos de la Guía de iniciación a Scratch muestran que dicho sujeto apenas muestra interés por la programación con bloques ni por los disfraces, lo que probablemente signifique una falta de entendimiento de las actividades, siendo esto respuesta a la carencia de PC que la guía requiere para su ejecución y desarrollo.

El sujeto nº3 tan solo consiguió 10 respuestas correctas de las 28 que contenía el TPC, por lo que se concluye con la misma afirmación que el sujeto número 2, existe una mejora en el desarrollo del PC puesto que anteriormente se carecía de este, pero a un nivel básico. En cuanto a los conceptos computacionales, apenas desataca en ninguno, tan solo en los condicionales, lo cual puede ser fruto de la repetición de programas.

En cuanto a las reflexiones, plantea problemas relacionados con la programación y con los disfraces, mostrando cierta reflexión por el proceso de programación. Aunque en este estudio el sujeto no muestra un claro desarrollo del PC, las inquietudes a cerca de la programación son el punto clave para seguir trabajando en su desarrollo del PC.

Como ya se ha referido anteriormente, el principal objetivo de este estudio pretendía verificar si gracias a la elaboración de proyectos con la herramienta Scratch los alumnos podrían desarrollar su Pensamiento Computacional. Dado que finalmente este estudio corrobora dicha afirmación, aunque teniendo en cuenta que la muestra es limitada y se trata de un estudio de caso, la conclusión general del estudio es positiva, los resultados han sido favorables, mostrando un cierto desarrollo del Pensamiento Computacional en algunos de los sujetos.

Se debe tener en cuenta, que se ha recogido un número escaso de respuestas correctas por parte de alguno de los sujetos, lo que puede deberse a que previamente carecían por completo de PC, lo que supone un proceso largo de trabajo con herramientas de programación computacional como Scratch.

Anteriormente ya se conocía que el Pensamiento Computacional podía ser desarrollado desde edades tempranas y que muchas son las herramientas que pretenden conseguirlo, al igual que se conocía que la herramienta Scratch era una de las herramientas más relevantes de la última década en la educación de todo el mundo, este estudio se acerca un poco más a esta realidad, teniendo en cuenta la limitación temporal y de muestra que este presenta.

Debido a los resultados positivos obtenidos, se puede concluir que la herramienta Scratch es propicia para desarrollar el Pensamiento Computacional de alumnos de Educación Primaria, concretamente de alumnos entre 11 y 12 años, lo que significa que es una buena herramienta para trabajar la base de la programación computacional desde las aulas de la educación básica. Esto implica cierto avance teniendo en cuenta todas las ventajas de dicha destreza, pues un alumno que desde la educación básica comienza a desarrollar su Pensamiento Computacional se convierte poco a poco en una persona resolutiva, capaz de razonar y resolver problemas con agilidad.

Por todo esto, incluir una herramienta que facilite desarrollar conceptos computacionales como es el caso de Scratch, es fundamental en la educación actual. Como se hacía referencia en el bloque teórico, muchos son los países que están incluyendo la programación computacional en sus currículos educativos de enseñanzas básicas, por lo que el segundo paso es encontrar las herramientas y recursos más óptimos para desarrollar dichas destrezas en el aula. Con este estudio nos acercamos un poco más en esa búsqueda por encontrar la herramienta adecuada, pues los resultados positivos de este permiten confirmar que la Scratch favorece la adquisición de conceptos computacionales, lo que implica un desarrollo de los pilares del Pensamiento Computacional.

Tras haber comprobado que los resultados son favorables, analizando diferentes ítems relacionados no solo con los conceptos computacionales si no también con la interfaz del programa, las dificultades encontradas, aportaciones de sugerencias de mejora, etc., se concluye con la afirmación de que utilizar la herramienta Scratch no solo implica mejorar las habilidades propias de la programación computacional, como es el Pensamiento Computacional, si no que se trata de una herramienta que potencia contenidos transversales tales como el razonamiento lógico-matemático, la autonomía personal, la creatividad y el trabajo en grupo.

Las limitaciones del estudio que se podrían mejorar de cara a futuras líneas de investigación están especialmente relacionadas con el escaso número de sujetos y de las actividades propuestas en la Guía de iniciación a Scratch.

La media de respuestas correctas obtenida no es especialmente significativa puesto que se trata de un estudio de caso y tan solo se ha contado con tres sujetos. Tan solo uno de ellos, el sujeto nº1 alcanzó la media propuesta por Román-González (2016) por lo que sería necesario evaluar a un mayor número de sujetos para verificar que el proceso a través de la Guía de Iniciación a Scratch es el adecuado para constituir la base del Pensamiento Computacional en alumnos de edades entre 11 y 12 años.

Por otro lado, sería conveniente ampliar el número de actividades y, por ende, el tiempo con el que el sujeto trabaja con la herramienta en cuestión. La Guía contaba con un total de ocho actividades adecuadas a la temporalización del estudio, por lo que disponer de un mayor número de proyectos a través de los cuales se trabaje desde la plataforma y se razone sobre posibles problemas y soluciones sería positivo para obtener una mayor veracidad de los resultados del análisis.

En cuanto a la evaluación con Dr. Scratch, los resultados obtenidos han sido muy similares en el caso de los tres sujetos, el motivo de esto puede ser la gran similitud que todas las actividades tenían entre ellas, ya que se trataba de fomentar la reflexión sobre los programas desde una estructura de programación básica, propia de los inicios del desarrollo del PC, formada por bloques ya estipulados. Sería favorable para el estudio, mejorar lo referente a este tema, con propuestas de proyectos más abiertos, donde cada sujeto trabaje en actividades singulares, que puedan ser evaluadas a través de la herramienta Dr. Scratch obteniendo diferentes niveles a las de sus compañeros. Para poder llevar a cabo esta mejora, es necesario ampliar la temporización, pues disponer de un mayor margen de tiempo es fundamental para el desarrollo positivo del estudio.

Referencias bibliográficas

- Basogain, X., Olabe, M. Á., y Olabe, J. C. (2015). Pensamiento Computacional a través de la *Programación: Paradigma de Aprendizaje*. *Revista de Educación a Distancia*, (46). Recuperado de: <http://revistas.um.es/red/article/view/240011/182851>
- Brennan, K., Balch, C., y Chung, M. (2014). Creative computing. Harvard Graduate School of Education. Recuperado de: http://scratched.gse.harvard.edu/guide/files/CreativeComputing20140820_LearnerWorkbook.pdf
- Brennan, K., y Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada (pp. 1-25). Recuperado de: <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>.
- Cearreta-Urbieta, I. (2015). Scratch como recurso didáctico para el desarrollo del Pensamiento Computacional de los alumnos de Secundaria y Bachillerato en la asignatura de Informática y como recurso transversal en el resto de asignaturas. Recuperado de: <http://reunir.unir.net/handle/123456789/3150>

- Galindo Suárez, M. (2016). Efectos del Proceso de Aprender a Programar con "Scratch" en el Aprendizaje Significativo de las Matemáticas en los Estudiantes de Grado Quinto de Educación Básica Primaria. *Repositorio Institucional del Tecnológico de Monterrey*. Recuperado de: <https://repositorio.itesm.mx/ortec/handle/11285/619592>
- López García, J. C. (2014) Actividades de aula con Scratch que favorecen el pensamiento algorítmico. Recuperado de: http://repository.icesi.edu.co/biblioteca_digital/bitstream/10906/76942/1/aula_scratch_insa.pdf
- López García, J. C. (2014) Actividades de aula con Scratch que favorecen el pensamiento algorítmico. Recuperado de: http://repository.icesi.edu.co/biblioteca_digital/bitstream/10906/76942/1/aula_scratch_insa.pdf
- Martínez Carazo, P. C. (2006). El método de estudio de caso: estrategia metodológica de la investigación científica. *Pensamiento y gestión*, (20). <http://www.redalyc.org/html/646/64602005/>
- Moreno, J., y Robles, G. (2014) Automatic detection of bad programming habits in scratch: A preliminary study. In *Frontiers in Education Conference (FIE), 2014 IEEE (pp. 1-4)*. Recuperado de: <http://ieeexplore.ieee.org/document/7044055/#full-text-section>
- Moreno-León, J., Robles, G., y Román-González, M. (2015). Dr. Scratch: análisis automático de proyectos Scratch para evaluar y fomentar el Pensamiento Computacional. *RED, Revista de Educación a Distancia*, 46. Recuperado de http://www.um.es/ead/red/46/moreno_robles.pdf
- Morrás Aranoa, H. (2014). Iniciación a la programación informática en Educación Primaria con Scratch. *Academia-e*. Recuperado de: <http://academica-e.unavarra.es/handle/2454/14363>
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., y Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67. Recuperado de: <http://dl.acm.org/citation.cfm?id=1592779>
- Román González, M. (2016). Codigoalfabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de programas. Recuperado de: <http://e-spacio.uned.es/fez/view/tesisuned:Educacion-Mroman>
- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, 97, 129-141. Recuperado de: <http://www.sciencedirect.com/science/article/pii/S0360131516300549>
- Serrano, J. L., & Prendes Espinosa, M. P. (2015). Integración de TIC en aulas hospitalarias como recursos para la mejora de los procesos educativos. Recuperado de: <http://dadun.unav.edu/handle/10171/38947>
- Valverde, J., Fernández, M. R. y Garrido, M. C. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia*, (46). Recuperado de: <http://revistas.um.es/red/article/view/240311/182991>
- 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. Recuperado de: http://www.scielo.cl/scielo.php?pid=S0718-50062015000400004&script=sci_arttext
- Vázquez- Cano, E., y Ferrer, F. (2015). La creación de videojuegos con Scratch en Educación Secundaria. *Communication papers: media literacy and gender studies*, 4(6), 63-73.
- Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: Measuring computational thinking in middle school. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 215-220. Recuperado de: <http://dl.acm.org/citation.cfm?id=2157200>

Wilson, A., & Moffat, D. C. (2010). Evaluating Scratch to introduce younger schoolchildren to programming. *Proceedings of the 22nd Annual Psychology of Programming Interest Group (Universidad Carlos III de Madrid, Leganés, Spain)*. Recuperado de: <https://pdfs.semanticscholar.org/a05f/71c355da3c534908f4451211382517dd84f1.pdf>

Wing, J. M. (2006). Computational thinking. Recuperado de: https://www.microsoft.com/en-us/research/wp-content/uploads/2012/08/Jeannette_Wing.pdf

Wing, J. M. (2011, March). Computational thinking. Recuperado de: <https://pdfs.semanticscholar.org/f0fe/32388ff4472e92c17753e8689ac56ff85bc1.pdf>

Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. *Revista de Educación a Distancia*. Recuperado de: <http://revistas.um.es/red/article/view/240321/1830>