

Uso de SCRATCH en el aprendizaje de Programación en Educación Superior

Use of SCRATCH in Learning Programming in Higher Education

Hamilton Pérez-Narvaéz¹, 0000-0002-4496-2549

Rosabel Roig-Vila², 0000-0002-7466-7364

Lilian Jaramillo-Naranjo³, 0000-0002-0586-4292

¹ Universidad Central del Ecuador, Quito, Ecuador hperez@uce.edu.ec

² Universidad de Alicante, Alicante, España rosabel.roig@ua.es

³ Universidad Central del Ecuador, Quito, Ecuador lmjaramillon@uce.edu.ec

Resumen

En los últimos años, el término pensamiento computacional y el lenguaje de programación Scratch han irrumpido en el escenario de la enseñanza, primordialmente a nivel de educación básica y bachillerato. Sin embargo, no se evidencian estudios que fundamenten el uso del lenguaje Scratch a nivel superior, por lo que, se constituyó en objetivo de la presente investigación, la verificación del desarrollo del pensamiento computacional entre los estudiantes de Primer Semestre de la Carrera de Informática de la Universidad Central del Ecuador con base a las ocho dimensiones del pensamiento computacional que son: identificación de patrones, uso de instrucciones, variables, secuencia, operadores, reuso, abstracción y funcionamiento y detección de errores. Con esta finalidad se aplicó un diseño cuasiexperimental con preprueba-postprueba y grupos intactos, con enfoque cuantitativo, de campo con apoyo documental y de nivel explicativo. Posteriormente, para los análisis estadísticos se utilizó la herramienta SPSS, tanto para el cálculo de las medidas de tendencia central y la desviación estándar, así como en la comprobación de las hipótesis mediante pruebas T y en la elaboración de figuras estadísticas. Los resultados obtenidos evidenciaron un mayor desarrollo en los niveles del pensamiento computacional con los estudiantes de Primer Semestre de la Carrera de Informática de la Universidad Central del Ecuador que emplearon Scratch.

Palabras clave: Computación, enseñanza, lenguaje, pensamiento, programación, Scratch.

Abstract

In recent years, the term computational thinking and Scratch programming language have broken into the teaching stage, primarily at the level of elementary and high school education. However, there are no studies that support the use of Scratch language at a higher level, thus the aim of this research was the verification of the development of computational thinking among the students of the First Semester of the Computer Science Degree of the Central University of Ecuador based on the eight dimensions of computational thinking that are: identification of patterns, use of instructions, variables, sequence, operators, reuse, abstraction, operation and error detection. For this purpose, a quasi-experimental design with pre-test-post-test and intact groups was applied, with a quantitative, field approach with documentary support and explanatory level. Subsequently, the SPSS tool was used for the statistical analyzes, both for the calculation of the measures of central tendency and the standard deviation, as well as in the verification of the hypotheses by means of T tests and in the elaboration of statistical figures. The results obtained evidenced a greater development in the levels of computational thinking with the students of the First Semester of the Computer Science Career of the Central University of Ecuador who used Scratch.

Keywords: Computing, teaching, language, thinking, programming, Scratch.

Recibido: 19/12/2019

Aceptado: 21/12/2019

1. Introducción

En los últimos años se viene discutiendo por parte de organizaciones relacionadas al ámbito educativo preguntas como las siguientes: ¿cuáles son las capacidades, habilidades, destrezas, competencias que deben alcanzar los estudiantes mediante la educación en el siglo XXI? ¿qué importancia tiene la Informática para el desarrollo de las personas en la sociedad contemporánea? Para responder estas preguntas se han propuesto varios escenarios y múltiples propuestas acerca del tema; sin embargo, existe una que va mostrando mucho interés por sus repercusiones, no solo en el aspecto educativo, sino en todos los ámbitos de la vida denominada pensamiento computacional.

El pensamiento computacional es un término que Zapotecalt (2015) describe como “un proceso de solución de problemas” (p. 12) que conlleva el desarrollo de habilidades tales como: el planteamiento o formulación de problemas, empleando los computadores para el análisis, modelización y resolución. Propone organizar datos de forma lógica para luego representarlos mediante abstracciones llegando a su modelización, algoritmización y finalmente “generalizar y transferir ese proceso de solución de problemas a una gran diversidad de estos” (Zapotecalt, 2015, p. 13).

Este criterio sobre el ilimitado potencial de la computación para la solución de problemas llevó a la International Society for Technology In Education (ISTE) y a la Computer Science Teachers Association (CSTA) a operacionalizar el concepto de pensamiento computacional. De esta manera, los educadores de los diferentes niveles comprendan de que trata el tema y pudieran desarrollarlo con sus estudiantes, expandiendo sus capacidades, su creatividad y aprovechando la transversalidad de la informática en todas las áreas del conocimiento.

También, como antecedente importante para la formulación del pensamiento computacional, Pérez (2017) señala a la Tercera Conferencia Internacional de Informática en las escuelas secundarias, realizada en 2008 en la ciudad de Turon, Polonia:

Donde se analizó por varios expertos las ventajas y dificultades de iniciar el aprendizaje de programación a edades tempranas con la intención de promover el desarrollo de las habilidades concernientes al pensamiento computacional en los estudiantes para desarrollarse en un ambiente donde la presencia de la tecnología computacional es cada vez más común e importante (p. 23).

Se menciona la necesidad de una segunda educación en informática, donde los maestros que dominan esta área del conocimiento sean el puente que relacionen a las demás personas con la comprensión de los principios de la ciencia Informática. Esta intención se evidencia en la formulación de un currículo que sobrepase las limitaciones de los actuales, en los que solamente se enseña el manejo de las TIC.

Mittermeir y Syslo (2008) resumen la generalidad de las conferencias señalando “varios autores en estas participaciones responden a estos problemas al dirigir la cuestión de lo que la educación en informática tiene que ofrecer a los jóvenes que vaya más allá de las habilidades de cómo se usan las computadoras.” (p. 3). Frente a esta situación se plantea la necesidad de una investigación que establezca las ventajas de utilizar herramientas mediadas simbólicamente para el aprendizaje de programación como se considera al lenguaje de programación Scratch. Es necesario, además, determinar el aporte de Scratch al desarrollo de habilidades como modelación, abstracción, reconocimiento de patrones, así como mejorar valores como el trabajo cooperativo, tenacidad y perseverancia en el proceso de enseñanza-aprendizaje.

Al hacer una investigación amparada en los principios científicos se asume una posición crítica, misma que reclama Martínez (2001) cuando propone no aceptar por obligación a la tecnología o simplemente oponerse a ella por considerarla deshumanizadora, sino que “la incorporación de la tecnología debe ir precedida de un análisis crítico de las necesidades a cubrir con ella y de las implicaciones que su utilización tiene, estando dispuestos a aceptar sus consecuencias” (p. 197). Tomando en cuenta esa recomendación y conociendo que el aprendizaje está conformado por varios elementos curriculares como contenidos, metodologías, recursos entre otros; es precisamente en este último componente donde la investigación pretende validar el uso de Scratch 2.0 como herramienta que aporte en la formación de los futuros docentes de Informática en el desarrollo de los ámbitos cognitivo, procedimental y emocional de forma englobadora.

En este sentido, el artículo está estructurado de forma que se presentan en un inicio, estudios relacionados y experiencias educativas acerca del pensamiento computacional, posterior a ello, se presentan conceptos y definiciones de respecto a los elementos del pensamiento computacional. Seguidamente, se presenta la metodología utilizada para el diseño de la investigación que fue realizada con enfoque cuantitativo de campo con apoyo documental y de nivel explicativo, seguido de los resultados obtenidos, y, finalmente se presentan las conclusiones obtenidas.

El objetivo de esta investigación es verificar la diferencia significativa en el desarrollo del pensamiento computacional en estudiantes que emplean Scratch para el aprendizaje de Programación comparado con quienes no lo utilizan, es decir, se trata de analizar un antes y un después de la aplicación de un programa de intervención.

2. Conceptos relacionados

2.1. Scratch

Como lo indica Pascual (2015) Scratch “es un lenguaje de programación desarrollado por un grupo de investigación del Instituto de Tecnología de Massachusetts (MIT), su característica principal es su entorno de programación visual y multimedia que permite crear programas de forma fácil e intuitiva mediante una interfaz gráfica” (p. 1).

López (2015) indica que la herramienta Scratch “hace que la programación sea más divertida para todo aquel que se enfrente por primera vez a aprender a programar” (p. 11), y considera que esta herramienta fue diseñada para expresar ideas de forma creativa con el objetivo de incentivar el desarrollo de habilidades de pensamiento lógico y computacional.

Scratch es un nuevo ambiente de programación y como lo menciona Alba (2008) “utiliza la metáfora de piezas encajables para animar objetos que se encuentran en la pantalla y muestra todos los elementos necesarios de uso como son el escenario, los objetos y los elementos del lenguaje” (p. 1).

Isuri Sormenezko Zerbitzuak en Scratch Guía Didáctica para Profesores (2010) menciona: “Scratch ayuda a comprender fácilmente conceptos matemáticos e informáticos que están muy bien integrados en el programa, como son: los procesos interactivos (bucles), los criterios condicionales (sí, entonces, sí-no), las coordenadas en un plano, las variables, etc.” (p. 4-5), de esta forma estos conceptos podrán aprenderse dentro de un contexto significativo y motivador, ya que no será lo mismo entender el significado del uso de las variables en la programación tradicional, que cuando se están utilizando para el control de visualización de alguna animación que se esté creando en ese momento.

Con la herramienta Scratch se trabaja mediante la creación de proyectos, partiendo de una idea principal, junto con la modelación y la experimentación, hasta conseguir un producto final, estos procesos de diseño de proyectos como se menciona en Scratch Guía Didáctica para Profesores (2010), “desarrollan las competencias necesarias para llegar a tener: un pensamiento creativo, un pensamiento lógico, un desarrollo de ideas, desde su concepción inicial hasta el proyecto acabado, una comunicación clara, un análisis sistemático, capacidad de colaboración, una reflexión interactiva., etc.” (p. 6-7).

Scratch permite aprender a programar mediante la experimentación de forma creativa, lo que ayuda al desarrollo del pensamiento lógico y a comprender mejor el funcionamiento computacional. Además, permite el desarrollo de habilidades mentales, favoreciendo la comprensión de los fundamentos de la programación.

2.2. Pensamiento Computacional

El pensamiento computacional es la habilidad que tienen los individuos para resolver problemas mediante el uso de la tecnología, es así que, una de las promotoras del pensamiento computacional, Wing (2006) señala que “el pensamiento computacional implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, haciendo uso de los conceptos fundamentales de la informática” (p. 33), es decir, la esencia del pensamiento computacional es pensar como un informático cuando se trata de resolver un problema. Wing (2006) dice que el pensamiento computacional es una forma de pensar y “que esas son habilidades útiles para todo el mundo, no sólo para los científicos de la computación.” (p. 33).

La definición de pensamiento computacional que se considera la más apropiada, es la que dio Jeannette Wing (2006), Vicepresidente Corporativo de Microsoft Research y profesora de Computer Science Department Carnegie Mellon University, quien popularizó el término en su artículo Computational Thinking señalando

que “el pensamiento computacional, representa una actitud universalmente aplicable y un conjunto de habilidades que todos, no solo los informáticos, estarían ansiosos por aprender y usar.” (p. 33).

Un concepto que proviene del mundo computacional es el señalado por Raja (2014) que indica “el enfoque computacional se basa en ver el mundo como una serie de puzzles, a los que se puede romper en trozos más pequeños y resolver poco a poco a través de la lógica y el razonamiento deductivo” (p. 1), es así que, el enfoque computacional es una forma intuitiva de abordar varios de los métodos existentes de la psicología del aprendizaje.

Es así que se puede señalar a las ocho dimensiones que definen al pensamiento computacional que son:

1. Identificación de patrones. - Consiste en extraer información de objetos que permita establecer propiedades de entre conjuntos de dichos objetos.
2. Uso de instrucciones. - Conjunto de datos insertados en una secuencia estructurada para que el procesador interprete y ejecute.
3. Variables. - Corresponde al área reservada en la memoria principal de un computador.
4. Secuencia. - Es una serie o sucesiones de elementos.
5. Operadores. - Son símbolos que indican cómo se deben manipular los operandos.
6. Reúso. - Corresponde a la reutilización.
7. Abstracción. - Consiste en aislar un elemento del resto de los elementos que lo acompañan.
8. Funcionamiento y detección de errores. - Consiste en detectar y controlar errores.

2.3. Tecnología Ubicua

La idea de las tecnologías ubicuas fue introducida por Weiser (1991), quien describió a las tecnologías ubicuas como “entornos rodeados de computadoras, y redes de comunicación, en conjunto con la interacción de los seres humanos.” (p. 1-10). Pérez y Addati (2013) señala que “las tecnologías ubicuas nos dan una nueva visión de la sociedad, vista a través de las mejoras que se producen en la calidad de vida de los ciudadanos. Se puede decir, entonces, que la ubicuidad de las tecnologías está dada por la disponibilidad de servicios, procesos e información vinculada a ellas en cualquier lugar y en todo momento” (p. 2), es así que este tipo de tecnología ayuda a mejorar la calidad de vida de las personas mediante el uso de herramientas que permitan gestionar la información de una manera más fácil y eficiente, y además que se pueden utilizar en todo momento y lugar.

La tecnología ubicua permitirá la aparición de nuevas aplicaciones donde todos los objetos se encuentren funcionando integrada, lo que conllevará una enorme apertura comercial, económica y social.

3. Estudios relacionados

Entre las experiencias educativas importantes en Latinoamérica acerca del pensamiento computacional está México, con el Instituto Nacional de Astrofísica, Óptica y Electrónica, con sede en Puebla. Es uno de los referentes en el trabajo sobre el tema de desarrollo de cursos online, textos, recolección de experiencias como simulaciones, proyectos, eventos entre otros, que se han propuesto como objetivo modificar la realidad del país en el uso de la Informática.

En el año 2015, una investigación realizada por Hitschfeld, Pérez y Simmonds documentan una experiencia en Chile realizada con estudiantes de nivel escolar, y expresan una preocupación al referirse sobre la necesidad de un salto educativo en los estudiantes, de consumidor de tecnología a generador de tecnología, los autores señalan a la programación como una fuente de conocimientos que posibilita generar estos cambios, principalmente si se trabaja con los más pequeños sin esperar que lleguen a la universidad (Hitschfeld, Pérez y Simmonds, 2015). Los autores consideran que la mayor parte de las personas pueden usar las TIC frente a situaciones cotidianas, pero si el problema es distinto a lo acostumbrado, su capacidad de respuesta disminuye; porque el trabajo continuo en tareas usuales o acciones mecanizadas no aporta al desarrollo de habilidades como abstracción, organización lógica de datos, reconocimiento de patrones, algoritmización, modelación y generalización por no mencionar la escasa importancia que adquieren la práctica de valores. A decir de, Hitschfeld, Pérez y Simmonds (2015) se deja de lado la “confianza en el manejo de la complejidad, persistencia

al trabajar con problemas difíciles, tolerancia a la ambigüedad, habilidad para lidiar con problemas abiertos, y habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común.” (p. 31-32).

En Ecuador, las experiencias educativas en el pensamiento computacional son escasas y aisladas, principalmente relacionadas con el uso de Scratch, lenguaje y herramienta que Pérez y Roig (2015) describen por su interacción con el usuario como “entornos de programación mediados simbólicamente” (p. 6), por sus características de interfaz gráfica, programación orientada a objetos mediante bloques prediseñados y entorno intuitivo diseñado por el Instituto de Tecnología de Massachusetts (MIT) y que en la actualidad ha alcanzado una alta difusión a nivel mundial.

La Escuela Politécnica del Litoral (ESPOL) y la Universidad Yachay, cada una por su cuenta, han desarrollado proyectos relacionados con el uso de Scratch. Tanto los proyectos desarrollados por ESPOL como los de Yachay están orientados fundamentalmente a que los niños aprendan Scratch como lenguaje de programación para elaborar proyectos, pero no evidencian con claridad la intención de desarrollar el pensamiento computacional, lo que conduce a pensar que siguen prestando mayor atención a la enseñanza de la herramienta informática que a las posibilidades consideradas dentro del pensamiento computacional para el estudiante, es decir, no alcanzan a la propuesta integral de las TIC en la educación de la que habla Cabero et. al (2003), que expone la importancia de observar el proceso educativo desde todos sus componentes y relaciones, para el autor el problema radica en:

la falta de formación del profesorado para su incorporación educativa de concederle más significación que las que poseen como instrumentos curriculares, de reproducir con ellas modelos tradicionales educativos y no pensados para las posibilidades comunicativas que tienen las TICs, y de utilizar sobre ellas principios organizativos propios de una escuela analógica y no digital (p. 18).

La Carrera de Informática de la Facultad de Filosofía de la Universidad Central del Ecuador, formas profesionales para ser docentes en esta área del conocimiento desde el año 1996, en los últimos tiempos presenta dificultades en el desarrollo de las habilidades referentes al pensamiento computacional con sus estudiantes. Actualmente, la carrera continúa utilizando herramientas tradicionales para el aprendizaje de programación y el diseño de algoritmos siendo las más aplicadas y actuales, por así llamarlo, Diagrama de Flujo de Datos (DFD) y PSeint con las dificultades que presenta en su empleo. Además, se observa errores de simbologías, pues están orientados a un paradigma poco usado en programación.

4. Metodología

El diseño de la investigación corresponde a cuasiexperimental con preprueba-postprueba y grupos intactos, con enfoque cuantitativo, de campo con apoyo documental y de nivel explicativo.

La población seleccionada fueron los estudiantes de Primer Semestre de la Carrera de Informática del semestre abril-septiembre 2015, conformada por dos cursos el uno correspondiente a la jornada matutina con la denominación de Primer Semestre A y el segundo de la jornada vespertina denominado Primer Semestre B. Se procedió a seleccionar los grupos de manera aleatoria, el paralelo A fue escogido como grupo experimental mientras que el paralelo B se convirtió en grupo control, ambos compuestos por un total de 40 estudiantes.

A continuación, se muestra el cuadro 1 donde se resume el total general de población y los clasifica por género, estableciendo el número y porcentaje de participantes masculinos y femeninos.

Cuadro 1: Población					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	masculino	64	80.0	80.0	80.0
	femenino	16	20.0	20.0	100.0
Total		80	100.0	100.0	

En relación con las edades de los participantes en el grupo experimental se establece como muestra la Cuadro 2 que el promedio es de 19,9 siendo la edad más frecuente entre los participantes 19 y 20 años.

Cuadro 2: Edades de la población del grupo experimental

		Frecuencia	Porcentaje	Porcentaj e válido	Porcentaje acumulado
Válidos	18	7	17.5	17.5	17.5
	19	12	30.0	30.0	47.5
	20	12	30.0	30.0	77.5
	21	3	7.5	7.5	85.0
	22	3	7.5	7.5	92.5
	23	1	2.5	2.5	95.0
	24	1	2.5	2.5	97.5
	26	1	2.5	2.5	100.0
	Total	40	100.0	100.0	

De acuerdo con el Cuadro 2, las edades de los participantes del grupo control tienen un promedio de 20,28 y la edad más frecuente entre el grupo es de 19 años

Cuadro 3: Edades de la población del grupo control

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	18	10	25.0	25.0	25.0
	19	11	27.5	27.5	52.5
	20	3	7.5	7.5	60.0
	21	5	12.5	12.5	72.5
	22	5	12.5	12.5	85.0
	23	4	10.0	10.0	95.0
	25	1	2.5	2.5	97.5
	30	1	2.5	2.5	100.0
	Total	40	100.0	100.0	

En relación con el objetivo, referente a la verificación del desarrollo del pensamiento computacional entre el grupo control y experimental, se estableció la hipótesis: existe diferencias en el desarrollo de las habilidades del pensamiento computacional de los estudiantes de Primer Semestre de la Carrera de Informática de la Universidad Central del Ecuador.

4.1. Instrumentos para la recolección de información

Se diseñaron tres instrumentos para recoger información con diferentes intenciones: un cuestionario de escala de satisfacción acerca de las habilidades del pensamiento computacional que consideran poseer los estudiantes, y dos cuestionarios de problemas relacionados al pensamiento computacional. Estos instrumentos sirvieron para establecer los avances en las habilidades antes mencionadas, y fueron aplicados en diferentes momentos del proceso investigativo.

Las pruebas de pretest y posttest se organizaron por el modelo de pruebas de base estructurada con preguntas o reactivos de cuestionamiento directo, cada cuestionario está constituido de dieciséis problemas. Andrade

(2013) menciona que la diferencia entre las pruebas objetivas y de base estructurada es que la segunda “privilegia procesos cognitivos de mayor nivel que el simple conocimiento por memorización” (p. 4), lo que permite desarrollar conocimientos y mejorar las habilidades para asimilar y procesar información.

El primer instrumento fue aplicado al inicio de la investigación y otro de similares características fue aplicado al final, para de esta manera comparar los resultados respecto del grupo control y experimental.

Los criterios que guiaron la estructura para la elaboración de los problemas en ambos instrumentos fueron las ocho dimensiones del pensamiento computacional que son: identificación de patrones, uso de instrucciones, variables, secuencia, operadores, reuso, abstracción y funcionamiento y detección de errores sobre las que se formularon dos problemas por cada dimensión seleccionada. Es necesario señalar que, el nivel de dificultad de los problemas es básico o inicial, por tratarse de estudiantes de Primer Semestre muy pocos de ellos conocen de programación, por lo que, aplicar un instrumento de mayor dificultad en los problemas no hubiese sido acorde con el grado de conocimiento de los participantes y su comprensión, esto en ningún caso afectó el carácter científico y su validez.

Para los análisis estadísticos se utilizó la herramienta SPSS 20 tanto para el cálculo de las medidas de tendencia central y la desviación estándar, así como en la comprobación de las hipótesis y la elaboración de figuras estadísticas donde se aplicó la comparación de medias mediante pruebas T.

Recogida la información de la encuesta, del cuestionario pretest y postest, se procedió a su ingreso en tres matrices diferentes, asignando a cada estudiante un código numérico que lo identifique y otro para el género, de manera que sea posible posteriormente agruparlos con base en dicha variable.

Para llevar a cabo la etapa de implementación se realizan las siguientes actividades:

1. Temporalización de la actividad en relación con la planificación de la asignatura durante el semestre. En concreto Programación I contiene una unidad llamada Técnica de resolución de problemas, donde se aborda el trabajo con algoritmos, escenario apropiado para investigación de las variables pensamiento computacional y Scratch, que tiene una duración de aproximadamente un mes y medio, tiempo con el que contó la investigación para realizar la experiencia con los estudiantes.
2. Diseño de la planificación microcurricular para cada sesión aplicando la metodología constructivista conocida como el ciclo del aprendizaje o ERCA.
3. Selección de actividades y problemas a trabajar en cada sesión.
4. Realización de actividades planificadas con la intervención de la herramienta Scratch, se desarrollaron durante siete sesiones que empezaron el 27 de abril de 2015 y concluyeron el 8 de junio de 2015. Cada sesión tuvo la duración de tres horas presenciales y horas de trabajo autónomo que efectuaron los estudiantes al resolver problemas propuestos tanto en forma individual como colaborativa.

Los contenidos y problemas propuestos en cada sesión se muestran en la Cuadro 4

Cuadro 4: Organización del proceso de intervención

FECHA	CONTENIDOS	PROBLEMAS PROPUESTOS
27-04-2015	- Proceso de resolución de problemas. - Definición de algoritmos. - Características - Fases del Método de Polya. - Ingreso y salida de información.	Tres algoritmos con el modelo de entrada de datos, procesos y salidas.

04-05-2015	- Operaciones matemáticas. - Graficación de formas con desplazamiento de objetos Scratch.	Cinco problemas que incluyen graficación, desplazamientos y operaciones matemáticas
11-05-2015	- Condiciones simples - Condiciones compuestas - Condiciones anidadas.	Cuatro problemas que con toma de decisiones, cálculos matemáticos y desplazamientos por pantalla con cambio de escenarios y objetos.
18-05-2015	- Condiciones compuestas - Condiciones anidadas.	Seis problemas que incluyen toma de decisiones, cálculos matemáticos, simulaciones, cambios de escenarios y objetos.
25-05-2015	- Condiciones compuestas - Condiciones anidadas.	Cinco problemas que incluyen toma de decisiones, cálculos matemáticos, simulaciones.
01-06-2015	Lazos de repetición definidos	Cuatro problemas que incluyen estructuras de repetición, operaciones matemáticas y desplazamiento de objetos.
08-06-2015	Lazos de repetición definidos e indefinidos	Seis problemas con lazos de repetición indefinidos que simulan situaciones reales y juegos.

5. Resultados

En búsqueda de similitudes o diferencias entre el grupo control y experimental se realizó una comparación de los resultados por dimensiones del pensamiento computacional utilizando medidas de tendencia central como la media y desviación típica, asumiendo como similares cuando exista una diferencia de ± 0.1 . Así se obtuvo el Cuadro 5 que se muestra a continuación:

Cuadro 5: Comparación de medias grupos de control y experiencia

[illegible]

CONTRO L	Media	.43	.66	.76	.64	.60	.45	.67	.59
	N	80	80	80	80	80	80	80	80
	Des v. típ.	.497	.476	.428	.484	.493	.50 1	.471	.49 5
EXPERI ME NTAL	Media	.75	.67	.79	.61	.60	.53	.55	.64
	N	80	80	80	80	80	80	80	80
	Des v. típ.	.436	.471	.412	.490	.493	.50 3	.501	.48 4
Total	Media	.59	.67	.78	.63	.60	.49	.61	.61
	N	160	160	160	160	160	160	160	160
	Des v. típ.	.494	.472	.419	.486	.491	.50 1	.489	.48 9

El Cuadro 5 muestra las medias de cada dimensión del pensamiento computacional consideradas en la investigación para el grupo control y experimental obtenidas al final de la investigación, de acuerdo al siguiente detalle:

-En la dimensión identificación de patrones, la media del grupo control corresponde a $.43 \pm .497$ y del grupo experimental corresponde a $.75 \pm .436$, de los resultados se observa que el grupo experimental alcanza una media mayor que la del grupo control.

-En la dimensión uso de instrucciones, la media del grupo control corresponde a $.66 \pm .476$ mientras que la media del grupo experimental corresponde a $.67 \pm .471$, por lo que, los valores permiten considerar ligeramente superior al grupo experimental.

-En la dimensión variable, la media del grupo control corresponde a $.76 \pm .428$ mientras en el grupo experimental corresponde a $.79 \pm .412$, valores que permiten considerar ligeramente superior al grupo experimental.

-En la dimensión secuencia, la media del grupo control corresponde a $.64 \pm .484$ mientras en el grupo experimental corresponde a $.61 \pm .490$, valores que muestra un nivel ligeramente superior del grupo experimental.

-En la dimensión operadores, la media del grupo control corresponde a $.60 \pm .493$ mientras en el grupo experimental corresponde a $.60 \pm .493$, valores que son similares.

-En la dimensión reuso, la media del grupo control corresponde a $.45 \pm .501$ mientras en el grupo experimental corresponde a $.53 \pm .503$, se muestra un puntaje superior del grupo experimental

-En la dimensión detección de errores, la media del grupo control corresponde a $.67 \pm .471$ mientras en el grupo experimental corresponde a $.55 \pm .501$, se muestra un puntaje superior del grupo control.

-En la dimensión abstracción, la media del grupo control corresponde a $.59 \pm .495$ mientras en el grupo experimental corresponde a $.64 \pm .484$, se muestra un puntaje superior del grupo experimental.

En resumen, los valores de la media de cada dimensión del pensamiento computacional correspondiente al grupo experimental son iguales o mayores a las medias que obtiene el grupo control, excepto en los casos de las dimensiones secuencia y detección de errores, donde el grupo control muestra un valor superior.

También se recurrió para verificación de resultados a la estadística inferencial, donde se trabajó con la hipótesis ya mencionada

H1: Existe una diferencia significativa en el desarrollo del pensamiento computacional entre los estudiantes que emplean Scratch en el aprendizaje de Programación en relación con quienes no lo utilizan.

La expresión matemática para la hipótesis es: $H1: \mu_e \neq \mu_c$

Donde μ_e corresponde al valor de la diferencia del grupo experimental, μ_c corresponde al valor de la diferencia del grupo control.

Hipótesis nula formulada correspondió a:

H0: No existe una diferencia significativa en el desarrollo del pensamiento computacional entre los estudiantes que emplean Scratch en el aprendizaje de Programación en relación con quienes no lo utilizan.

La expresión matemática para la hipótesis es: $H_0: \mu_e = \mu_c$

Para la comprobación estadística se procedió primero a comprobar la normalidad mediante la prueba de Kolmogorov y Smirnov, relacionando las variables grupo y la diferencia obtenida de la resta entre la nota del postest y la nota del pretest, cuyo resultado se muestra en el siguiente cuadro:

Cuadro 6: Pruebas de normalidad

GRUPO		Kolmogorov-Smirnova ^a			Shapiro-Wilk		
		Estadístico	gl	Sig.	Estadístico	gl	Sig.
DIFERENCIA	CONTROL	.154	4	.01	.948	4	.06
	A		0	7		0	6
	EXPERIMENTAL	.140	4	.04	.982	4	.75
	L		0	5		0	7

a. Corrección de la significación de Lilliefors

El Cuadro 6 muestra los resultados comprobatorios sobre la normalidad entre los grupos control y estadístico obteniendo una significancia de .017 para el control y .045 para la experimental, razón por la cual, se acepta la hipótesis que señala que los datos provienen de grupos con diferente normalidad.

Como siguiente paso previo a la selección del estadístico para comprobación de hipótesis se realizó la prueba de varianzas entre los grupos, los resultados se muestran en el cuadro a continuación:

Cuadro 7: Pruebas de muestras independientes

Prueba de Levene para la igualdad de varianzas					Prueba T para la igualdad de medias				
	F	Sig.	t	Gl	Sig. (bilateral)	Diferencia de medias	Error típ. de la diferencia	Inferior	95% Intervalo de confianza para la diferencia Superior
Se han asumido varianzas iguales	13.855	.000	-1.030	78	.306	-1.0938	1.0618	-3.2077	1.0202
No se han asumido varianzas iguales			-1.030	65.227	.307	-1.0938	1.0618	-3.2142	1.0267

En el Cuadro 7 se muestran los resultados para la prueba de varianzas de Levene que alcanza una significancia de .000, razón por la cual, no se descarta la hipótesis que señala que ambos grupos tienen varianzas distintas. Con estos resultados se puede establecer que los grupos no son paramétricos, por lo que, se seleccionó la prueba de comprobación de hipótesis llamada U de Mann Whitney por el número de estudiantes, también por

considerarse a los datos muestras independientes, además, la variable organiza el tipo numérico y correspondió a un estudio longitudinal.

Apoyado en el software estadístico SPSS se obtuvo el resultado expresado en el Cuadro 8 para la hipótesis de trabajo planteada:

Cuadro 8: Prueba de hipótesis

Hipótesis nula	Test	Sig.	Decisión
La distribución de DIFERENCIA es la misma entre las categorías de GRUPO	Prueba U de Mann- Whitney de muestras independientes	.728	Retener la hipótesis nula

De acuerdo con el Cuadro 8 se obtiene el valor de significancia de .728, lo que se interpreta que no existen las suficientes evidencias para descartar la hipótesis nula

Por lo tanto, se considera que entre las notas del grupo control y las notas del grupo experimental no existe una diferencia significativa en el desarrollo del pensamiento computacional empleando Scratch en el aprendizaje de Programación

6. Discusión y conclusiones

En el Cuadro 5 se presentó la comparación de medias en las respuestas del grupo control y experimental, con base a estos resultados se evidencian mejoras en las dimensiones identificación de patrones, uso de instrucciones, variables, uso de operadores y detección de errores en el grupo que utilizó Scratch en el aprendizaje de Programación. El grupo que no utilizó Scratch muestra mantener el mismo desarrollo o mejorar en las otras cuatro dimensiones: identificación de patrones, uso de instrucciones, variables y detección de errores por lo que se podría asumir que existe una diferencia, aunque no significativa entre los grupos. Sin embargo, al realizar la prueba de comprobación de hipótesis que muestra el Cuadro 8, se establece que las evidencias son escasas para rechazar la hipótesis nula. Por lo tanto, se asume que no existen suficientes evidencias para afirmar que Scratch ayude de manera directa al desarrollo del pensamiento computacional en todos los niveles considerados en la investigación.

En comparación de lo que informan Sáez, Miyata y Domínguez (2016) en su investigación sobre la creación de códigos multimedia mediante Scratch con estudiantes de educación superior donde señalan “A partir de los resultados de la prueba t de Student efectuada, se puede afirmar que existe mejoras significativas en los resultados de la prueba administrada, por lo que el programa mejora la capacidad de los estudiantes para comprender la gestión de la Programación de contenidos multimedia con Scratch.” (p. 154), no se observan iguales resultados. Probablemente algunas de las causas que originan los resultados distintos se deban al origen de los participantes, en el caso de las investigaciones que se mencionó la población seleccionada corresponde a 113 estudiantes, entre hombres y mujeres, que participan voluntariamente del programa EUROMIME de formación a nivel de posgrado que fortalece los conocimientos en Informática de profesionales relacionados con la rama tecnológica, mientras que los estudiantes que formaron parte de la investigación en la Universidad Central del Ecuador corresponden al nivel inicial en la Carrera de Informática donde muchos de ellos no han tenido experiencias previas en relación con la creación de programas.

No existen muchas investigaciones de la herramienta Scratch aplicada a nivel superior con las cuales comparar los resultados, al respecto Taborda y Medina (2012) también coinciden en esta dificultad y expresan que “no obstante, y a pesar de la amplia aceptación que ha tenido en la comunidad de educadores, no hemos encontrado investigaciones publicadas que hayan informado acerca del impacto que el uso del SCRATCH genera en el aula de clases.” (p. 6). Otra posición en cuanto a la formulación de problemas que permitan observar de mejor manera otros aspectos del pensamiento computacional a ser evaluados la proponen Nancovzka, Ternik, Koron

y Koron (2017) que expresan “existen varias formas de mejorar el andamiaje del desarrollo cognitivo para jóvenes programadores de Scratch. Una es introducir ejercicios enfocados en el paralelismo y sincronización, como pedirles a los alumnos que expliquen partes del código. Otra forma es preguntar a los estudiantes para encontrar errores en un código” (p. 10).

Como se mencionó existen varios aportes, pero principalmente a nivel de educación básica e inicial, otras en relación con el interés y motivación como el trabajo de Pérez y Roig (2015) sobre entornos de programación no mediados simbólicamente donde se recoge la opinión de los estudiantes del nivel inicial de la Carrera de Informática, acerca del uso de Scratch que consideran que “facilita el trabajo por su interfaz, además desarrolla la creatividad y el trabajo colaborativo” (p. 16). Esto lleva a considerar que la validez y el potencial que posee la herramienta Scratch no sea tan evidente en todas las dimensiones del pensamiento computacional que forman parte de la investigación, pero si en ciertos aspectos como mejorar la motivación, interés, creatividad e imaginación y facilitar el manejo de las sentencias al no concentrar la atención de los estudiantes en la sintaxis como en otros lenguajes de programación.

Referencias bibliográficas

- Alba, R. (2008). Iniciándose en la programación con Scratch. Observatorio Tecnológico del Ministerio de Educación, Cultura y Deporte de España. Retrieved from <http://recursostic.educacion.es/observatorio/web/fr/software/programacion/619-iniciandose-en-la-programacion-con-scratch?format=pdf>
- Andrade, X. (2013). Guía para la elaboración de pruebas de base estructurada. Quito: Universidad Central del Ecuador. Retrieved from <http://es.slideshare.net/italito/guia-para-la-elaboracin-de-pruebas-de-base-estructurada>
- Cabero, J., Castaño, C., Cebreiro, B., Gisbert, M., Martínez, F., Morales, J. y Prendes, M. (2003). Las nuevas tecnologías en la actividad universitaria. Pixel Bit. Revista de Medios Y Educación, 20, 81–100. Retrieved from <http://acdc.sav.us.es/ojs/index.php/pixelbit/issue/view/37>
- Hitschfeld, N., Pérez, J. y Simmonds, J. (2015). Pensamiento computacional y programación a nivel escolar en Chile: el valor de formar a los innovadores tecnológicos del futuro. Bits de Ciencia, 12, 28–33.
- Isuri, S. (2010). Scratch Guía Didáctica para Profesores. Isuri Sormenezko Zerbitzuak Servicios Creativos. 4-7. Retrieved from <https://bit.ly/2S3whsK>
- López, J. (2015). Programación con scratch cuaderno de trabajo para estudiantes. Fundación Gabriel Piedrahita Uribe. Cuarta edición. 11. Retrieved from <http://eduteka.icesi.edu.co/pdfdir/AlgoritmosProgramacionCuaderno1.pdf>
- Martínez, F. (2001). El profesorado ante las nuevas tecnologías. In C. y T. Junta de Extremadura. Consejería de Educación (Ed.), Sociedad de la información y Educación (pp. 195–218). Mérida. Retrieved from <https://bit.ly/2s1fjk5>
- Mittermeir, R. y Syslo, M. (2008). Informatics Education-supporting Computational Thinking. (M. Mittermeir, R y Syslo, Ed.) (1st ed.). Berlín: Springer. Retrieved from <https://bit.ly/2Zawcoz>
- Nančovská, I., Šerbec Ternik, Z., Koron, A. y Koron, T. (2017). Learning Programming Concepts through Maze Game in Scratch. Retrieved from https://www.researchgate.net/publication/320170373_Learning_Programming_Concepts_through_Maze_Game_in_Scratch
- Pascual, J. (2015). Scratch, programación sencilla y gratis para niños y mayores. Retrieved from <https://computerhoy.com/noticias/software/scratch-programacion-sencilla-gratis-ninos-mayores-37925>
- Pérez, G. y Addati, G. (2013). Documentos de trabajo: Tecnologías Ubicuas. Editor Jorge M. Streb. Nro. 531. 1-2. Retrieved from <https://ucema.edu.ar/publicaciones/download/documentos/531.pdf>
- Pérez, H. y Roig, R. (2015). Entornos de programación no mediados simbólicamente para el desarrollo del pensamiento computacional. Una experiencia en la formación de profesores de Informática de la Universidad Central del Ecuador. RED. Revista de Educación a Distancia, 14(46), 1–22. Retrieved from <http://www.um.es/ead/red/46/rosabel.pdf>
- Pérez, H. (2017). Uso de Scratch como herramienta para el desarrollo del pensamiento computacional en programación I de la carrera de informática. Universidad Central del Ecuador. (23). Retrieved from https://rua.ua.es/dspace/bitstream/10045/82731/1/tesis_hamilton_omar_perez_narvaez.pdf
- Raja, T. (2014). We can code it!. Retrieved from <http://www.motherjones.com/media/2014/06/computer-science-programming-code-diversity-sexism-education>

- Sáez, J., Miyata, Y., y Domínguez, M. (2016). Creative Coding and Intercultural Projects in Higher Education: a Case Study in Three Universities / Codificación creativa y proyectos interculturales en Educación Superior: Un estudio de caso en tres universidades. RIED. Revista Iberoamericana de Educación a Distancia, 19(2), 145– 165.
- Taborda, H. y Medina, D. (2012). Programación de computadores y desarrollo de habilidades de pensamiento en niños escolares: fase exploratoria. Cali: Universidad ICESI.
- Weiser, M. (1991). The Computer for the 21st Century. Scientific American Ubicomp Paper. Retrieved from <https://www3.nd.edu/~cpoellab/teaching/cse40463/weiser.pdf>
- Wing, J. M. (2006). Computational Thinking. Communications of the acm /Vol. 49, No. 3. Retrieved from <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>
- Zapotecalt, J. (2015). Pensamiento computacional. Curso Pensamiento Computacional. Puebla: Instituto Nacional de Astrofísica, Óptica y Electrónica de Puebla. Retrieved from <http://www.pensamientocomputacional.org/index.php/home/menu-definicion>

Autores

HAMILTON PEREZ- NARVÁEZ obtuvo su título de Licenciado en Ciencias de la Educación especialización Informática, Especialista en Entornos Virtuales OEI. Magíster en Educación Superior y Doctor en Investigación Educativa en la Universidad de Alicante. Ha escrito varios artículos relacionados con la Tecnología Educativa y el pensamiento computacional. Profesor de Bachillerato Técnico en Informática, Profesor de la Universidad Central del Ecuador en las cátedras de: Programación, Didáctica de la Informática y Pedagogía, profesor de la Pontificia Universidad Católica del Ecuador en cursos de TIC.

Actualmente es director de la carrera de Pedagogía de las Ciencias Experimentales en Informática y coordinador de la Red ecuatoriana de tecnología e innovación educativa. Ha sido tutor de tesis de grado y posgrado en la Facultad de Filosofía, coordinador de Maestría en Educación mención Gestión del aprendizaje mediado por TIC.

ROSABEL ROIG -VILA Rosabel Roig Vila es Doctora en Pedagogía (premio extraordinario) y Profesora Titular de la Universidad de Alicante [UA] en el Departamento de Didáctica General y Didácticas Específicas. Actualmente es directora del Instituto de Ciencias de la Educación (ICE) de la Universidad de Alicante. Ha sido Decana de la Facultad de Educación de la Universidad de Alicante (2005-2009). Tiene reconocidos dos tramos de investigación y 7 trienios de docencia. Es editora de la revista electrónica científica Journal of New Approaches in Educational Research (NAER) (<http://www.naerjournal.ua.es>), indexada en SCOPUS y ESCI. Dirige desde hace años el grupo de investigación EDUTIC-ADEI de la Universidad de Alicante (VIGROB-039) con el compromiso de investigar en los procesos de enseñanza-aprendizaje como medio para ampliar el conocimiento científico y la innovación educativa. Ha coordinado proyectos de investigación I+D+i en España y la Unión Europea. Su línea de investigación se centra en el campo de las Tecnologías de la Información y la Comunicación aplicadas a la educación. Dirige desde el año 2003 el Máster de la UA: “Educación y Tecnologías de la Información y Comunicación” y participa en otros Másteres interuniversitarios (UOC, Universidad de Huelva, Universidad de Cádiz y Universitat de les Illes Balears). Ha impartido docencia, por invitación, en el Dottorato di ricerca in Pedagogia en la Università degli Studi Suor Orsola Benincasa. Es coordinadora de diversos cursos organizados de formación del profesorado universitario. Ha impartido cursos y ponencias por invitación en la Universitat Politècnica de Catalunya, Universidad de Bilbao, Universidad de Valencia, etc. Es coordinadora de las listas de distribución EDUTIC y ARTICUA (RedIRIS-CSIC). Ha participado en más de 30 tribunales de Tesis Doctorales como vocal, secretaria y presidenta. Ha escrito más de 100 publicaciones, tales como: Evaluación de la calidad pedagógica de los MOOC, Assessment of Web Content Accessibility Levels in Spanish Official Online Education Environments, Investigar el Cambio Curricular en el Espacio Europeo de Educación Superior, La articulación de las Tecnologías de la Información y la Comunicación en la educación, El uso de las TIC en el aula de Educación Especial: percepción de los maestros. Ha participado en diversos comités científicos internacionales tales como Comitato Scientifico Internazionale della Collana editoriale Pedagogie e didattiche (Casa Editrice Pensa Multimedia), Advisory Board 2012 Global Education Conference (Global Education Conference Network) y E-Learning Conference Committee (IADIS-International Association for Development of the Information Society is a non-profit association).

LILIAN JARAMILLO - NARANJO obtuvo su título de Magíster en Tecnologías para la Gestión y Práctica Docente por la Facultad de Educación de la Pontificia Universidad Católica del Ecuador (Ecuador) en 2015. Obtuvo el título de Magíster en Educación y Desarrollo Social en la Facultad de Comunicación Artes y

Humanidades de la Universidad UTE (Ecuador) en 2008. Obtuvo el título de Licenciada en Biología por la Facultad de Filosofía, Letras y Ciencias de la Educación de la Universidad Central del Ecuador en 2007. Obtuvo el título de Doctora en Ciencias de la Educación.

Actualmente es profesora auxiliar de la Facultad de Filosofía, Letras y Ciencias de la Educación de la Universidad Central Ecuador en la Carrera Pedagogía Informática. Fue docente titular de la Universidad UTE del Ecuador. Es articulista y miembro del Consejo Internacional de Revisores de Revista Sophia de la Universidad Politécnica Salesiana (Ecuador). También es editor asociado de la *Revista Cátedra* de la Facultad de Filosofía, Letras y Ciencias de la Educación de la Universidad Central del Ecuador. Sus principales temas de investigación incluyen educación, TIC, biología, ciencias naturales, didáctica de ciencias naturales, currículo, pedagogía informática, desarrollo de estrategias metacognitivas para la práctica docente desde las TIC. Ha participado en el Instituto de Posgrados como tutora y en ponencias internacionales organizados por la Universidad Técnica del Norte. Es autora de capítulos de libros y artículos publicados en revistas de alto impacto (Emerging Source Citation Index, Latindex, Redalcy, Scielo). Es autora del Libro Realidad Nacional editado por la Universidad UTE 2019.