

Hacia el desarrollo del Pensamiento Computacional y la Lógica Matemática a través de Scratch

Towards the development of computational thinking and mathematical logic through Scratch

Benjamín Maraza-Quispe¹, Ashtin Maurice Sotelo-Jump², Olga Melina Alejandro-Oviedo³, Lita Marianela Quispe-Flores⁴ y Lenin Henry Cari-Mogrovejo⁵

¹Universidad Nacional de San Agustín de Arequipa-Perú

²Universidad Nacional de San Agustín de Arequipa-Perú

³Universidad Nacional de San Agustín de Arequipa-Perú

⁴Universidad Nacional de San Agustín de Arequipa-Perú

⁵Universidad Nacional de San Agustín de Arequipa-Perú

Resumen

Actualmente la necesidad de brindar una educación de calidad a las futuras generaciones ha ocasionado el desarrollo de nuevas metodologías de enseñanza, dentro de este hecho las herramientas que nos proporciona las tecnologías de información se han posicionado como el futuro del aprendizaje, en tal sentido, el aprender a programar dejó de ser considerada una habilidad selectiva del campo de la computación, siendo hoy por hoy una necesidad para cualquier estudiante que desea ser competente en este mundo globalizado y dinámico. Dentro de este contexto la presente investigación tiene como objetivo analizar en qué medida el uso del lenguaje de programación Scratch permite el desarrollo de habilidades del pensamiento computacional y lógica matemática.

La metodología consistió en la aplicación de los fundamentos de programación a través de Scratch 3.0 a un grupo experimental compuesto por 25 estudiantes los cuales fueron seleccionados en forma aleatoria de una población de 100 estudiantes, la recolección de datos se realizó a través de un test de razonamiento lógico estandarizado por Acevedo y Oliva y un test de niveles de pensamiento computacional estandarizado por González.

De acuerdo con los resultados, se postula una diferencia significativa en el desempeño de los estudiantes en ambos test, teniendo una mejoría más considerable en los criterios: Bucles, Control de Variables (CV), Probabilidad (PB) y Operaciones Combinatorias (CB). Por consiguiente, se concluye destacando la importancia de la enseñanza de conceptos básicos de la Ciencia de la Computación como son el pensamiento computacional y la lógica matemática, dado que contribuye a la interiorización de conceptos al desarrollar algoritmos en la resolución de problemas.

Palabras clave: Scratch; pensamiento computacional; lógica matemática; enseñanza.

Abstract

Currently the need to provide quality education to future generations has caused the development of new teaching methodologies, within this fact the tools provided by information technologies have positioned themselves as the future of learning, in this sense, learning to program is no longer considered a selective skill in the field of computing, being today a necessity for any student who wants to be competent in this globalized and dynamic world. Within this context, the present research aims to analyze to what extent the use of the Scratch programming language allows the development of computational thinking skills and mathematical logic.

The methodology consisted of applying the fundamentals of programming through Scratch 3.0 to an experimental group composed of 25 students who were randomly selected from a population of 100 students, the data collection was done through a test of logical reasoning standardized by Acevedo and Oliva and a test of computational thinking levels standardized by Gonzalez.

According to the results, a significant difference is postulated in the performance of the students in both tests, having a more considerable improvement in the criteria: Loops, Control of Variables (CV), Probability (PB) and Combinatorial Operations (CB). Therefore, we conclude by highlighting the importance of teaching basic concepts of Computer Science such as computational thinking and mathematical logic, since it contributes to the internalization of concepts by developing algorithms in problem solving.

Keywords: Scratch; computational thinking; mathematical logic; teaching.

¹ **Correspondencia:** Benjamín Maraza-Quispe, bmaraza@unsa.edu.pe

Introducción

Actualmente, dentro de las mallas curriculares de distintas Instituciones Educativas de Educación Básica Regular (EBR), la implementación de herramientas TI en la educación resulta impropia debido a una falta de competencia de docentes en su uso y también en parte por un desconocimiento en las virtudes que traen estas estrategias en el desarrollo cognitivo del estudiante, creando así una percepción de complejidad hacia la programación tanto para alumnos como docentes [1].

Sin embargo, la virtud de la combinación de las metodologías convencionales con las herramientas TI, son determinantes, estableciendo así a jóvenes que emplean el pensamiento computacional y el razonamiento matemático-lógico en su vida diaria.

Según [2] el pensamiento computacional puede definirse como la habilidad de resolver problemas mediante capacidades como algoritmos y métodos computacionales, este pensamiento se divide en cuatro principales procesos: descomposición, abstracción, reconocimiento de patrones y algoritmos. Los mencionados procesos tendrán un rol fundamental en el análisis de los beneficios que trae para los estudiantes siendo estos evaluados mediante el test diseñado por [3].

Igualmente, el razonamiento está compuesto por diversas capacidades asociadas a la matemática, como la aritmética pitagórica o la geometría euclidiana. Por otro lado, más allá de concepciones tradicionales, en la actualidad la lógica matemática es considerada como el razonamiento causante de la ciencia mediante la validación del conocimiento por el método científico [4].

La importancia del razonamiento lógico siempre ha sido valorada en la educación primaria y secundaria; sin embargo, en el contexto presentado nace la siguiente interrogante: ¿en qué medida la enseñanza de lenguaje de programación mediante Scratch 3.0 desarrolla el Razonamiento Lógico y el Pensamiento Computacional en estudiantes de Educación Básica Regular?

Según la bibliografía existente, parece existir una respuesta tentativa que afirma dicho desarrollo de habilidades, pero aún más existe una mejoría a la aceptación del error, ya que, el éxito de este tipo de software radica en el desarrollo de las habilidades esperadas, gracias al aprendizaje de programación por metas [5].

Marco Teórico

El Lenguaje de Programación

En el mundo de la informática, cualquier tecnología u objeto a la que se le denomina o posee en su interior un computador o procesador, trabaja con él un lenguaje único denomina código binario, es decir que básicamente para un ordenador solo existen unos y ceros.

Por lo tanto, cuando se refiere al lenguaje de programación este se subdivide en dos: el lenguaje de alto nivel y bajo nivel; en este caso, en el presente trabajo se tomará como punto central el primero, ya que los programas que se expondrán más adelante utilizan este tipo de lenguaje de programación. Estos son más cercanos al lenguaje matemático y natural [4].

Existen muchos tipos de lenguaje de programación, a continuación, se presenta una tabla que localiza Scratch dentro de los lenguajes de programación:

Tabla 1

Tipos de Lenguaje de Programación

Tipo de Lenguaje	Description	Software
C++	Destinado al desarrollo de programas o paquetes.	Eve online, Panda3D
Visual	Es un lenguaje de programación en estado creciente, utilizado mayormente para la enseñanza de conocimientos básicos del lenguaje de programación.	Scratch, CODE
JavaScript	Destinado para crear programas que se almacenarán en páginas web, también es ideal para crear e implementar efectos y acciones	Android Studio

Scratch 3.0 software

Scratch se define a sí mismo como un programa que permite crear historias, juegos o animaciones; que luego pueden ser compartidas con la comunidad, que brinda una retroalimentación positiva para los usuarios. Además, ayuda a mejorar la creatividad y el pensamiento sistemático en los jóvenes [6].

Además, entre los numerosos softwares de programación de tipo visual, Scratch es el programa de educación por excelencia, el cual brinda una solución al abandono común de cursos sobre programación, provocado por una percepción elevada dificultad que supone esta actividad [7].

Asimismo, Scratch cuenta con un proceso de enseñanza constructivo y sobre todo activo lo que genera en los estudiantes una mejor experiencia al aprender a programar, y como defendía Papert: las lenguas de programación deben tener un “suelo bajo” y un “techo alto”, es decir que no deben ser un reto el entender cómo empezar a programar, sin embargo, las posibilidades deben ser gigantescas [7].

Pensamiento Computacional

En este mundo moderno y cambiante la tecnología es parte importante del desarrollo de nuevos métodos para mejorar la productividad de ciertos productos [8].

El pensamiento computacional, es aquel modo de resolución de problemas, mediante métodos computacionales o métodos normalmente utilizados por la tecnología como algoritmos [2].

Este proceso consta de varias partes a seguir, las cuales son desarrolladas mediante un razonamiento específico denominado lógica computacional.

el pensamiento computacional es de gran importancia para las exigencias actuales siendo estas caracterizadas por los constantes problemas a los que se someten los empleados, por ejemplo. En pocas palabras, el pensamiento computacional es imperativo si se quiere “sobrevivir” en este mundo globalizado. Debido a que, la enseñanza de este sistema brinda a la población una herramienta para que, como se mencionó anteriormente, mejora la producción y eficacia en cualquier tipo de ámbito laboral y en la vida diaria.

En este caso, se dará una mayor importancia dentro de la descomposición como parte fundamental de la lógica computacional y por ende del pensamiento computacional.

Razonamiento Lógico

El razonamiento lógico está compuesto por diversas capacidades asociadas a la matemática, como la aritmética pitagórica o la geometría euclidiana. Por otro lado, más allá de concepciones tradicionales, en la actualidad la lógica matemática es considerada como el razonamiento causante de la ciencia mediante la validación del conocimiento por el método científico [9].

Es suma, la importancia del razonamiento lógico siempre ha sido valorada en la educación primaria y secundaria; sin embargo, en los últimos años, el desarrollo de estas habilidades matemáticas es el mismo, los métodos pueden ser variados, en este caso estudiaremos el impacto de las TI en el aprendizaje de la lógica matemática.

Enseñanza de Scratch 3.0 en estudiantes

Referencia [7], tratando específicamente de Scratch, al ser este un tipo de lenguaje de programación visual orientado a bloques cuenta con un proceso de enseñanza constructivo, ideal para iniciar a los estudiantes a la programación. Este paulatino proceso acompañado del desarrollo de habilidades esperadas mediante un sistema de programación por metas individuales logra que Scratch 3.0 sea el software de programación por excelencia para jóvenes.

Por otro lado, vale la pena recalcar lo que dice la bibliografía de los últimos años sobre el desarrollo de ambos pensamientos y su relación con Scratch: en primer lugar, el razonamiento lógico está compuesto por diversas capacidades asociadas a la matemática, como la aritmética pitagórica o la geometría euclidiana. Por otro lado, en la actualidad la lógica matemática es considerada como el razonamiento que da origen a la ciencia mediante a la validación de datos [4], es por ello que autores como [10] presenta un currículum para los estudiantes que esté orientada a sus verdaderas necesidades por ende se plantea como primer eje la globalización.

En cuanto a cómo Scratch logra desarrollar las capacidades mencionadas anteriormente, todo radica en los principios con los que fue ideado, a continuación, se presentan los tres principios fundamentales según [7].

Tabla 2

Principios de Scratch 3.0, (López & Sánchez, 2012)

Principios de Scratch	Descripción
El lenguaje de programación debe ser divertido	La facilidad con la que se puede probar diferentes opciones para completar una determinada acción es fundamental para mejorar la experiencia de lo que es programar.
El lenguaje de programación debe ser significativo	Cuando una persona desea aprender algo nuevo, una de las mejores formas de hacerlo, es si la actividad es significativa para la persona, es decir, tiene un cierto grado de relevancia y autenticidad para cada usuario. Scratch está diseñado justamente para cumplir con este requisito, es diverso y personal a la vez.
El lenguaje de programación debe ser social	Scratch se encuentra muy unida a su propia página web, ya que de esta manera se ha logrado consolidar una buena comunidad alrededor de la plataforma del MIT, siendo así que cada usuario, sin importar la edad, tenga la posibilidad de compartir mediante solo un clic, su trabajo con el mundo entero; pudiendo recibiendo las mejoras necesarias para su animación, historietas, juego o proyecto en general.

El modelo propuesto por [7] se logra gracias a la interface de Scratch. La misma que contó con un diseño original motivada en cumplir las necesidades de aprendizaje y, además, crear interés en niños y jóvenes. Es así como, en primera instancia Scratch se empezó a utilizar en lugares ajenos a las aulas, aunque ineludiblemente llegaría a la currícula de miles de escuelas en todo el mundo, debido a esta invitación a la exploración y al intercambio con pares [11].

La literatura sobre la enseñanza de Scratch 3.0 y su efecto en estudiantes

Especialmente en la última década, se han realizado diversos estudios en las escuelas sobre la adquisición de habilidades de programación y pensamiento computacional en los estudiantes. Scratch se considera una herramienta útil para enseñar programación o para asegurar que los estudiantes adquieran habilidades de pensamiento computacional [12-13].

Las puntuaciones obtenidas al evaluar proyectos de Scratch a través de la herramienta web Scratch y los niveles de pensamiento computacional de los estudiantes no difieren según el género. En otras palabras, el género no influye en los puntajes de evaluación de proyectos de los estudiantes. Si bien la literatura presenta pruebas exhaustivas del impacto del género en las características de los estudiantes relacionados con las computadoras o la programación [14-15]

Existe una relación significativamente alta entre las habilidades de Scratch de los estudiantes y sus habilidades de pensamiento computacional. En otras palabras, el desarrollo de las habilidades de programación de los estudiantes en Scratch provocará aumentos similares en sus habilidades de pensamiento computacional o las mejoras en sus Niveles de pensamiento computacional generarán aumentos en sus habilidades de Scratch. La literatura proporciona una amplia prueba de que el proceso de programación no es un proceso mecánico, sino una disciplina del pensamiento [16].

Según [17] en una investigación desarrollada que tuvo como objetivo comparar los puntajes de estudiantes de quinto grado obtenidos de proyectos Scratch, los puntajes obtenidos de la Escala de Niveles de Pensamiento Computacional y examinar esta comparación en términos de diferentes variables. Se utilizó un modelo de investigación correlacional en el que participaron 31 estudiantes, a los estudiantes se les enseñó programación básica usando Scratch durante un período de 6 semanas. Al final de la capacitación, las habilidades de programación de los estudiantes se midieron a través de la herramienta web Dr. Scratch. Las habilidades de pensamiento computacional se midieron utilizando una Escala de niveles de pensamiento computacional que incluye 5 factores: creatividad, resolución de problemas, pensamiento algorítmico, colaboración y pensamiento crítico. Los datos se analizaron para determinar la fiabilidad interna para calcular la fiabilidad de la escala. Se encontró que el coeficiente de confiabilidad alfa de Cronbach era 0,809. Se encontró que las puntuaciones obtenidas por los estudiantes mediante el uso de cualquiera de las herramientas de medición no difirieron según el género o el período de uso de la computadora, sin embargo, se observó una relación significativa de alto nivel entre las habilidades de programación de los estudiantes con Scratch y su computacional habilidad de pensamiento.

Metodología

La metodología utilizada está enfocada en un estudio cuantitativo debido a determinar puntos concluyentes respaldados en datos numéricos sometidos a tratados estadísticos para corroborar su validez.

Objetivo

Analizar en qué medida el uso del lenguaje de programación Scratch permite el desarrollo de habilidades del pensamiento computacional y lógica matemática.

Población y Muestra

La población total está compuesta por 100 estudiantes del tercer grado de secundaria de Educación Básica Regular de los cuales se seleccionaron a través de un muestreo aleatorio simple 25 estudiantes para el grupo experimental. Con la muestra descrita anteriormente serán importantes las constantes evaluaciones a las que se le someterá, mediante pruebas destinadas a cuantificar el uso del pensamiento computacional en la resolución de problemas.

Procedimiento

Se diseñaron sesiones de aprendizaje que conformadas por cuatro lecciones como mínimo, cada una con una duración de sesenta minutos y dividida en cuatro partes: propósito, en donde se expresa la meta de la clase; desarrollo, donde se procede con la explicación de los bloques y la estructura de la programación visual; evaluación, momento en el que se plantea un ejercicio con el nombre de "Reto" y finalmente salida, sección final donde se recuperan todos los conceptos y se elaboran conclusiones.

Por último, se evaluó a la muestra con un pre-test, antes de aplicar las lecciones de programación y al término de las lecciones se finalizó con un post-test; ambas evaluaciones como instrumento de recojo de datos.

Instrumentos de recolección de datos

Se utilizaron dos instrumentos de recolección debido a los dos campos de estudio: la lógica matemática y el pensamiento computacional. En el caso del primero se empleó lo propuesto por Tobin y Copie (1981), y su estudio "Test of Logical Thinking" (TOLT), y la posterior conversión y validación al idioma español, realizado por [18], el mismo que evalúa mediante preguntas abiertas y cerradas cinco criterios sobre el razonamiento lógico: proporcionalidad (PP), control de variables (CV), probabilidad (PB), correlación (CR) y operaciones combinatorias (CB).

En el caso del pensamiento computacional se basó en el estudio de Gonzales, en el que a través de una prueba de 28 ítems de elección múltiple tiene como fin la cuantificación estándar de los niveles de pensamiento computacional en los sujetos, en la solución de problemas ayudándose con conceptos computacionales: Direcciones Básicas, Bucles (repetir veces), Bucles (repetir hasta), Condicional Simple, Condicional Compuesta, Mientras (while) y Funciones Simples.

El test tiene una duración de 45 minutos y como población objetiva a estudiantes desde los doce años hasta los quince años [3].

Análisis y Resultados

Recolección, clasificación y análisis de datos:

De acuerdo con los resultados obtenidos, el desempeño de la muestra entre criterios era considerablemente variado por lo que se decidió hacer un estudio detallado por criterio de cada instrumento de recolección: en el caso del razonamiento lógico matemático se clasificaron las notas en cinco criterios que son los que nos presenta el autor del test y en el caso del pensamiento computacional se hizo lo propio, pero en siete criterios de evaluación. De esta manera se busca obtener datos más certeros y específicos sobre en qué campos existe una mejora significativa después del tratamiento (enseñanza de Scratch 3.0).

A continuación, se muestra la tabla de clasificación final del Test de Razonamiento Lógico (TRL), con un puntaje máximo de 50 por criterio.

Tabla 3

Evaluación por criterio en el TRL

Criterios	Resultado total obtenido	
	Pre-test	Post-test
Proporcionalidad	48	50
Control de variables	26	37
Probabilidad	27	36
Correlación	27	29
Operaciones combinatorias	33	33

A partir de los datos procesados en la Tabla 3 no todos los criterios evaluados por el test de Acevedo y Oliva tuvieron el mismo cambio. Se rescatan específicamente los criterios segundo y tercero los cuales explican: uso de variables y de proporcionalidad donde existe un cambio más significativo con relación a los demás. Esto debido al mismo proceso de uso de Scratch [7].

Con respecto al segundo campo de estudio, como se menciona anteriormente se aplica el mismo principio con la diferencia de que cada criterio del Test de Pensamiento Computacional tiene una nota máxima de 100.

Tabla 4

Criterios de evaluación en la prueba de Pensamiento Computacional

Criterios	Resultado total obtenido	
	Pre-test	Post-test
Direcciones Básicas	99	100
Bucles (repetición de veces)	41	71
Bucles (repetir hasta)	39	69
Condicional simple	42	50
Condicional compuesto	33	48
Mientras	40	55
Funciones simples	32	42

Como se analiza a partir de la Tabla 4, de manera específica, en el criterio segundo y tercero, existe una mejoría más significativa. Dichos criterios miden la capacidad de los alumnos en el uso de bucles tanto en repeticiones numéricas (repetir cuanto) como en repeticiones con condicionales al final (repetir hasta). Este concepto está relacionado con lo que afirma [19]: (...) Al escribir código, los estudiantes aprenden a organizar un proceso, reconocen rutinas o repeticiones y descubren errores en su pensamiento computacional cuando su programa no funciona según la idea o expectativa con la que fue concebido. Todos ellos son características clave del pensamiento computacional (...). Como se analiza a partir de la Tabla 3, de manera específica, en el criterio segundo y tercero, existe una mejoría más significativa. Dichos criterios miden la capacidad de los alumnos en el uso de bucles tanto en repeticiones numéricas (repetir cuanto) como en repeticiones con condicionales al final (repetir hasta). Este concepto está relacionado con lo que afirma.

Validación de la propuesta

Con motivo de validar la propuesta presentada, se empleó el uso del análisis de varianza para muestras emparejadas, mediante este tratado estadístico se busca concluir si el tratamiento realizado a tenido un impacto significativo o de lo contrario los resultados no son lo suficientemente buenos como para afirmar que la enseñanza de Scratch mejora el razonamiento lógico y el pensamiento computacional.

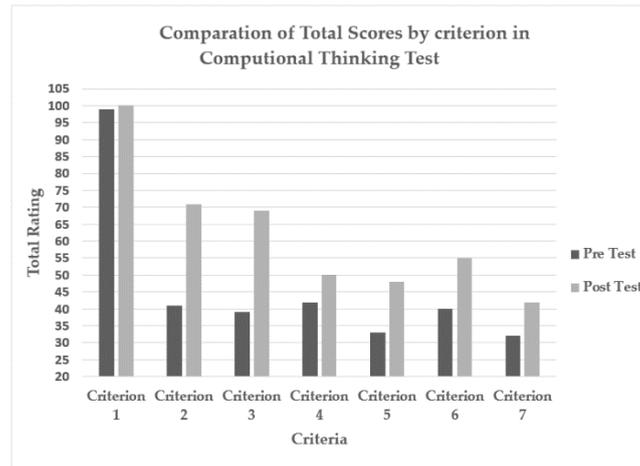


Figura 1. Comparación por criterio en el Test de Razonamiento Lógico.

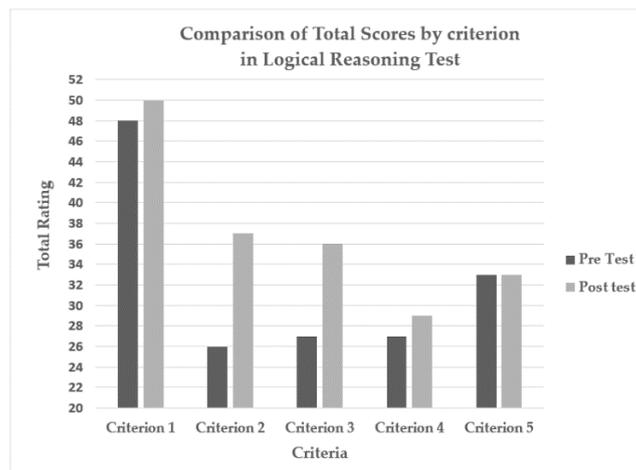


Figura 2. Comparación general en el Test de Razonamiento Lógico.

Los resultados fueron los augurados por la bibliografía, debido a que la muestra satisfizo las expectativas logrando un efecto significativamente superior entre el pre y post test, basado en el postulado de Acevedo y Oliva, como se observa en la Figura 2.

Por otro lado, si bien existe una mejoría considerable en los criterios explicados, analizando las medias de ambos grupos de datos, la diferencia de es 1.16. Recurriendo a los autores estudiados y analizando la Figura 1, se puede explicar: que una posible causa puede ser la orientación de la metodología en el tratamiento, debido al grado determinante del educador en el razonamiento lógico, como concluye [20]: (...) El profesor debe dotar a sus alumnos de las herramientas necesarias para aprender a aprender, mediando así su aprendizaje. Es una necesidad urgente potenciar el desarrollo de capacidades y valores en el aula. (...)

Asimismo, desarrollando en el caso del criterio primero y quinto, al tratarse de Proposiciones y Operaciones Combinatorias, las características de la muestra y el tiempo del tratamiento pueden haber influido en los resultados, debido a que son conceptos más complejos que, comparándolos con la edad media de la muestra, aún no están interiorizados del todo.

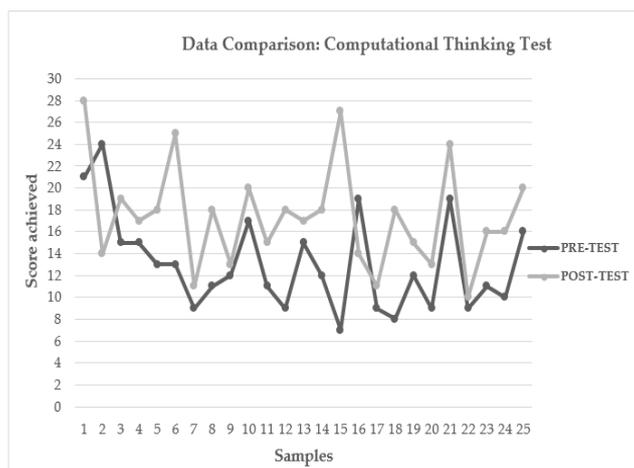


Figura 3. Comparación por criterio en el Test de Pensamiento Computacional.

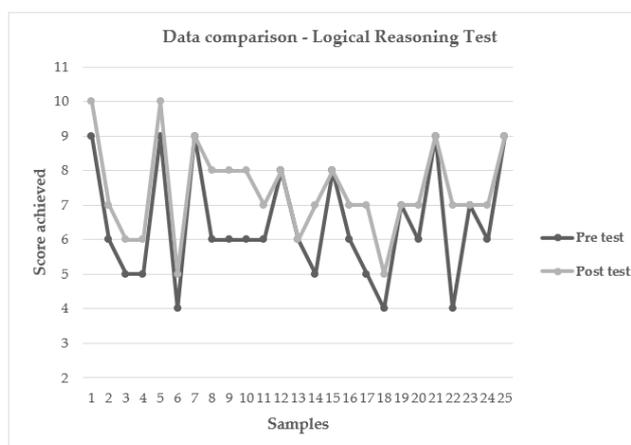


Figura 4. Comparación general en el Test de Pensamiento Computacional.

Se observa en la Figura 3 una diferencia considerable entre el pre y post test, lo que nos augura buenos resultados en el tratamiento. Sin embargo, la mejora no ha sido uniforme en todos los criterios, véase la Tabla 3, en la cual criterios como direcciones básicas y funciones simples cuentan con resultados bastante cercanos.

En conclusión, el tratamiento en la muestra seleccionada ha tenido el impacto esperado, aumentando significativamente el desempeño dentro del TPC, basado del estudio de Rodríguez, validando así el aporte de diversos autores en el desarrollo del Pensamiento Computacional mediante la enseñanza de Scratch.

T-student para dos muestras relacionadas

El tratado estadístico empleado consiste en el formulado de dos hipótesis: la hipótesis nula y la alternativa, según los resultados del tratado se aprueba o refuta una de las hipótesis.

a) Análisis por criterio dentro de la Prueba de Razonamiento Lógico:

Ho = No hay diferencia significativa en las medias de los resultados en el TRL antes y después del tratamiento.

Ha = Hay una diferencia significativa en las medias de los resultados en el TRL antes y después del tratamiento.

$\alpha = 5\%$

Tabla 5

Prueba T para dos muestras pareadas en la prueba de Razonamiento Lógico

	Variable 1	Variable 2
Promedio	6.44	7.4
Diferencia	2.76	1.45
Observaciones	25	25
Coefficiente de correlación de Pearson	0.86	7.4
Diferencia hipotética de medias	0	-
Grados de libertad	24	-
Estadístico T	5.70	-
P (T <= t) una cola	3.49	-
Valor crítico de t (de una cola)	1.71	-
P (T <= t) dos colas	0.0000070	-
Valor crítico de t (dos colas)	2.063	-

Dado que el valor P resultante es menor al 5% o 0.05, entonces se rechaza la hipótesis nula y se acepta la hipótesis alternativa, en otras palabras, si hay una diferencia significativa después del tratamiento.

- Análisis por criterio dentro de la Prueba de Razonamiento Lógico: Una vez validados los datos dentro del TRL, se estudió la significancia por criterio. De esta manera, se obtuvieron conclusiones más específicas en campos singulares y el impacto del tratamiento (docencia de Scratch 3.0 en estudiantes de tercer año de secundaria) en la muestra.

Tabla 6

Prueba T por criterio para TRL

Criteria	P-Value	alpha	Significant difference
Proporcionalidad	48	50	NO
Control de variables	26	37	SI
Probabilidad	27	36	SI
Correlación	27	29	NO
Operaciones combinatorias	33	33	SI

b) Prueba general de la Prueba de Pensamiento Computacional:

Ho = No hay diferencia significativa en las medias de los resultados en el TPC antes y después del tratamiento.

Ha = Hay una diferencia significativa en las medias de los resultados en el TPC antes y después del tratamiento.

$\alpha = 5\%$

Tabla 7

Prueba T para dos muestras pareadas en la prueba de Pensamiento Computacional

	Variable 1	Variable 2
Promedio	13.04	17.4
Diferencia	19.12	22.42
Observaciones	25	25
Coefficiente de correlación de Pearson	0.28	-
Diferencia hipotética de medias	0	-
Grados de libertad	24	-
Estadístico T	-3.98	-
P (T <= t) una cola	0.00	-
Valor crítico de t (de una cola)	1.71	-
P (T <= t) dos colas	0.000553121	-
Valor crítico de t (dos colas)	2.06	-

Como en el caso de TRL, el valor de P es menor que alfa, por lo que se afirma que existe una diferencia significativa.

- Análisis por criterios dentro del Test de Pensamiento Computacional:

Tabla 7

Prueba T por criterio en la prueba de Pensamiento Computacional

Criteria	P-Value	alpha	Significant difference
Direcciones Básicas	0.16	0.05	NO
Bucles (repetición de veces)	0.00	0.05	SI
Bucles (repetir hasta)	0.00	0.05	SI
Condicionales simples	0.31	0.05	NO
Condicionales compuestas	0.07	0.05	NO
Mientras	0.08	0.05	NO
Funciones simples	0.06	0.05	NO

Discusión y Resultados

La presente investigación concluyó que el tratamiento empleado en la muestra ha logrado tener un impacto significativo en los resultados de ambos campos de estudio: el pensamiento computacional y el razonamiento lógico.

En primer lugar, en el caso del Pensamiento Computacional, se afirman los beneficios de la enseñanza de programación en estudiantes, con capacidades relacionadas a la resolución de problemas, así como de todas las edades, sin hacer ningún tipo de distinción debido a que la actividad se amolda a las capacidades individuales de cada persona. Dejando de lado el desarrollo de las capacidades evaluadas durante el presente trabajo, también las mejoras se centran en habilidades blandas dentro de los alumnos acercándolos a un entorno moderno y dinámico, propio del mundo laboral.

En segundo lugar, el aprendizaje de la lógica matemática ha sido confirmado por el análisis estadístico, por lo cual se afirma las ventajas de la implementación de las TI en áreas aparentemente ajenas como es la matemática escolar, todo esto logrado a través de la aplicación del TRL un test estandarizado. De esta manera, se recomienda más específicamente la enseñanza de Scratch para mejorar el rendimiento de estudiantes en el tema de control de variables y probabilidad.

En tercer lugar, el entorno de enseñanza que se logra crear mediante Scratch vale la pena ser recalcado, en especial con miras a la futura implementación de entornos virtuales en miles de escuelas alrededor del mundo. Este software permite a los alumnos trabajar como equipo además de permitir un intercambio sociocultural que enriquece su perspectiva del mundo

De este modo, en gran medida la enseñanza de Scratch 3.0 ha permitido desarrollar el pensamiento computacional y razonamiento lógico en estudiantes de secundaria, lo que augura la buena relación entre las TI y la Educación, innovando y regalando nuevas herramientas de aprendizaje por las cuales las nuevas generaciones se formen con capacidades modernas y fundamentales en el mundo actual.

Referencias

- [1] Maraza Quispe, B. (2011). Influencia de un entorno multimedia de simulación por computadora en el aprendizaje por investigación de la Física. *Nuevas Ideas en Informática Educativa*, TISE.
- [2] Moreno León, J. (23 de marzo de 2014). Programamos. Obtenido de Videojuegos y "apps": <https://programamos.es/que-es-el-pensamiento-computacional/>
- [3] Román Gonzalez, M., Pérez Gonzalez, J. C., & Jiménez Fernández, C. (2015). Test de Pensamiento Computacional: diseño y psicometría general. Congreso Internacional sobre Aprendizaje, Innovación y Competitividad.
- [4] Hernandez, L. (2004). Fundamentos de la programación. Madrid, España.
- [5] Sánchez Rey, A. (2016). UNIVERSIDAD DEL PAÍS VASCO. Obtenido de Archivo Digital Docencia Investigación. <https://addi.ehu.es/bitstream/>
- [6] Scratch-MIT. (s.f.). Scratch. Obtenido de Acerca de Scratch: <https://scratch.mit.edu/about>
- [7] López-Escribano, C., & Sánchez-Montoya, R. (2012). *Revistas UM*. Obtenido de <https://revistas.um.es/red/article/view/233521/179471>
- [8] Colegio de Bachilleres. (2008). Lógica Computacional y Programación. Mexico.
- [9] Sangüillo Fernandez-Vega, J. (2008). El pensamiento lógico matemático. Madrid: Ediciones Akal, S.
- [10] Chamorro, C. (2011). *Revistas UM*. Obtenido de <https://revistas.um.es/educatio/article/view/132961/122661>
- [11] Maloney, J., Resnick, M., Rusk, N., Silverman, R., & Eastmond, E. (Noviembre de 2010). The Scratch Programming Language. *ACM Trans*, 15.
- [12] S. Catlak, M. Tekdal. and F. Baz. The Status of Teaching Programming with Scratch: A Document Review Work. *Journal of Instructional Technologies & Teacher Education*, 2015, 4(3), 13-25.
- [13] O. Ozyurt. and H. Ozyurt. A Study for Determining Computer Programming Students' Attitudes Towards Programming and Their Programming Self - Efficacy. *Journal of Theory and Practice in Education*, 2015, 11(1), 51-67.
- [14] S. Toker. An Assessment of Pre-Service Teacher Education Program in Relation to Technology Training for Future Practice: A Case of Primary School Teacher Education Program, Burdur. (Unpublished master's thesis), 2014, Graduate School of Education, METU.
- [15] A.P. Ambrosio. F. M. Costa. L. Almeida. A. Franco. and J. Macedo. Identifying cognitive abilities to improve CS1 outcome. Paper presented at the Frontiers in Education Conference (FIE), 2016.
- [16] Ali OLUK, Özgen KORKMAZ, "Comparing Students' Scratch Skills with Their Computational Thinking Skills in Terms of Different Variables", *International Journal of Modern Education and Computer Science (IJMECS)*, Vol.8, No.11, pp.1-7, 2016. DOI: 10.5815/ijmeecs.2016.11.01
- [17] Acevedo, J. A., & Oliva, J. (1995). Validación y aplicaciones de un test de razonamiento lógico. *Revista de psicología general y aplicada*, 339-351.
- [18] Valverde Berrocoso, J., Fernández Sánchez, M. R., & Garrido Arroyo, M. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia*(46), 18
- [19] Cerillo MM. (2002). Enseñar a Pensar: Desarrollo de Razonamiento Lógico. *Aula Abierta*.