

Impact of the use of Scratch and PSeInt on the development of algorithmic thinking in university students

Edwar Saire-Peralta¹; Sonia Calloapaza-Pari¹; Ricardo Calienes-Rodríguez¹; Rene Nieto-Valencia¹; Christian Revilla-Arroyo¹

¹Universidad Nacional de San Agustín de Arequipa, Perú

esaire@unsa.edu.pe, scalloapaza@unsa.edu.pe, rcalienesr@unsa.edu.pe, rnietov@unsa.edu.pe, crevilla@unsa.edu.pe

Abstract- *Algorithmic thinking represents a fundamental pillar of computational thinking and plays a key role in the initial training of university students, even in fields not directly related to computer science. This research aimed to compare the effect of using Scratch and PSeInt on the development of algorithmic thinking in first-year university students majoring in Chemistry. The research employed a quasi-experimental design with pre- and post-test measurements, using the Marcos Román-González Computational Thinking Test as the assessment instrument. The intervention took place during the second unit of the basic computer science course entitled "Fundamentals of Algorithms," where one group of students worked with Scratch and another group with PSeInt. The results showed that both groups demonstrated improvements in their scores for the algorithmic thinking pillar; however, the group using Scratch showed a statistically significant improvement. No statistically significant differences were found when directly comparing the changes between the two groups. These findings suggest that Scratch promotes the development of algorithmic thinking in initial training situations, although the magnitude of the improvement does not differ significantly from PSeInt.*

Keywords-- *Computational thinking, Algorithmic thinking, Scratch, PSeInt, Higher education*

Impacto del uso de Scratch y PSeInt en el desarrollo del pensamiento algorítmico en estudiantes universitarios

Edwar Saire-Peralta¹; Sonia Calloapaza-Pari¹; Ricardo Calienes-Rodríguez¹; Rene Nieto-Valencia¹; Christian Revilla-Arroyo¹

¹Universidad Nacional de San Agustín de Arequipa, Perú

esaire@unsa.edu.pe, scalloapaza@unsa.edu.pe, rcalienesr@unsa.edu.pe, rnietov@unsa.edu.pe, crevilla@unsa.edu.pe

Resumen– El pensamiento algorítmico representa un pilar fundamental del pensamiento computacional y desempeña un rol clave en la formación inicial de estudiantes universitarios, incluso en las carreras no están directamente vinculadas a la informática. La presente investigación tuvo como objetivo comparar el impacto del uso de las herramientas de Scratch y PSeInt en el desarrollo del pensamiento algorítmico en estudiantes universitarios de primer año de la carrera de Química. La investigación tuvo un diseño cuasi-experimental con mediciones pretest y posttest, utilizando el instrumento Test de Pensamiento Computacional de Marcos Román-González como instrumento de evaluación. La intervención se desarrolló durante la segunda unidad del curso de informática básica llamado fundamentos de algoritmia, donde un grupo de estudiantes trabajó con Scratch y otro grupo con PSeInt. Los resultados mostraron que ambos grupos presentaron mejoras en el puntaje del pilar pensamiento algorítmico; sin embargo, el grupo que utilizó Scratch mostró una mejora estadísticamente significativa. No se encontraron diferencias estadísticamente significativas al comparar directamente el cambio entre ambos grupos. Estos resultados sugieren que Scratch favorece el desarrollo del pensamiento algorítmico en situaciones de formación inicial, aunque la magnitud de la mejora no difiere significativamente respecto a PSeInt.

Palabras clave: Pensamiento computacional, Pensamiento algorítmico, Scratch, PSeInt, Educación superior.

I. INTRODUCCIÓN

El pensamiento computacional ha sido identificado como una competencia importante en la educación superior, relacionada al desarrollo de habilidades clave como la resolución de problemas, el razonamiento lógico, la estructuración de procesos mediante la programación y diseño de soluciones creativas[1]. Diversos autores coinciden que el pensamiento computacional se sustenta en cuatro pilares fundamentales: abstracción, descomposición, reconocimiento de patrones y pensamiento algorítmico [2], [3], [4].

En particular, el pensamiento algorítmico desempeña un rol central, ya que permite formular soluciones paso a paso de manera secuencial, en base a lógicas y estructuradas [4], [5]. En este contexto, la enseñanza de los principios de algoritmia en carreras no informáticas representa un desafío, pues los

estudiantes muchas veces presentan dificultades para comprender estructuras formales de programación con enfoque textual [6].

Las herramientas educativas como Scratch y PSeInt han sido ampliamente utilizadas para introducir conceptos algorítmicos en contextos educativos como colegios y universidades. La herramienta Scratch, está basado en programación visual utilizando bloques, el cual facilita la enseñanza y comprensión inicial de estructuras algorítmicas mediante representaciones gráficas [7], [8]. Por otro lado, PSeInt, está orientado a la escritura de pseudocódigo con enfoque textual, el cual se aproxima a la sintaxis y lógica de los lenguajes de programación tradicionales, de tal forma que la transición hacia entornos de programación formal se hace más fácil. El uso del pseudocódigo como representación textual para los algoritmos es una herramienta ampliamente utilizada en la enseñanza inicial de programación, ya que permite a los estudiantes comprender y aplicar secuencias básicas de programación antes de abordar lenguajes de programación de alto nivel, funcionando como paso intermedio hacia entornos de programación formal [9], [10].

Existen estudios que han analizado de manera específica el desarrollo del pensamiento algorítmico mediante el uso de herramientas de programación visual y textual. Las investigaciones centradas en Scratch reportan mejoras en la comprensión de secuencias, estructuras de control y lógica algorítmica, al permitir la construcción de algoritmos paso a paso mediante representaciones visuales llamado bloques que reducen la complejidad inicial del aprendizaje [11], [12]. Por otro lado se tienen estudios basados en el uso de pseudocódigo y programación textual, el cual señalan que este enfoque fortalece el pensamiento algorítmico al exigir una formulación explícita literal y ordenada de los pasos de un algoritmo, promoviendo una comprensión más cercana a la lógica de los lenguajes de programación formales [13], [14].

Sin embargo, se han encontrado que la mayoría de las investigaciones existentes analizan las herramientas de programación visual y textual de manera independiente o como parte de evaluaciones generales del pensamiento

computacional. Son escasos los estudios que comparan de forma directa el impacto de entornos visuales como Scratch y herramientas textuales basadas en pseudocódigo como PSeInt, dentro de un mismo diseño experimental, particularmente cuando el análisis se centra exclusivamente en el desarrollo del pensamiento algorítmico en estudiantes universitarios de carreras no informáticas, empleando instrumentos validados y mediciones pretest y postest.

En este contexto de la investigación, el objetivo del presente estudio es comparar el desarrollo del pensamiento algorítmico en estudiantes universitarios de carreras no informáticas que emplean Scratch y PSeInt durante una unidad de algoritmia del curso de informática básica, mediante un diseño cuasi-experimental con mediciones pretest y postest. Se debe señalar de manera clara que el análisis se centra exclusivamente en el pilar del pensamiento algorítmico, sin considerar variables actitudinales ni el desempeño en los demás pilares del pensamiento computacional.

II. METODOLOGÍA

A. Diseño y enfoque de la investigación

El estudio se desarrolló bajo un enfoque cuantitativo, dado que se trabajó con datos numéricos y se emplearon técnicas de estadística descriptiva e inferencial para el análisis de los puntajes obtenidos en el pilar del pensamiento algorítmico.

El diseño de la investigación fue cuasiexperimental de corte longitudinal, donde se trabajó con grupos intactos previamente conformados y se realizaron mediciones antes y después de la intervención pedagógica. Este tipo de diseño de investigación ha permitido analizar los cambios en el puntaje de pensamiento algorítmico a lo largo del tiempo, así como comparar el impacto de las herramientas de programación utilizadas en cada grupo.

Respecto al nivel de investigación, el estudio profundizó a un nivel descriptivo-comparativo, debido a que se centró en la caracterización y comparación de los puntajes de pensamiento algorítmico obtenidos por los estudiantes en los momentos pretest y postest, así como en la identificación de diferencias significativas entre los grupos mediante el uso de pruebas estadísticas inferenciales.

B. Participantes

La población de estudio estuvo conformada por estudiantes de primer año de la carrera de Química de la Universidad Nacional de San Agustín, matriculados en el curso de Informática Básica. El grupo A estuvo conformado por 28 estudiantes, el cual utilizaron la herramienta Scratch, mientras que el grupo B estuvo conformado por 12 estudiantes el cual utilizaron la herramienta PSeInt tal como encontraron en los registros de matrícula. Ambos grupos fueron evaluados antes y después de la intervención mediante un pretest y un postest, aplicados a los mismos estudiantes en cada grupo. En ambos grupos se desarrolló la unidad de algoritmia, que

corresponde a la segunda unidad del curso de informática básica. Se trabajó con la totalidad de los estudiantes disponibles durante el periodo de estudio, por tales circunstancias se empleó una muestra de tipo censal, por lo que no se realizó un proceso de muestreo probabilístico.

C. Instrumento

Para recolectar los datos para la investigación se utilizó el Test de Pensamiento Computacional propuesto por Román-González, compuesto por 28 ítems de respuesta dicotómica (donde se codificó asignando 1 a la respuesta correcta y 0 a la respuesta incorrecta). Este instrumento ha sido ampliamente utilizado en investigaciones previas y cuenta con evidencia de validez y confiabilidad para la evaluación del pensamiento computacional en contextos educativos.

El test de Román-González evalúa los cuatro pilares del pensamiento computacional: abstracción, descomposición, reconocimiento de patrones y pensamiento algorítmico. Cada ítem está asociado a uno o más de estos pilares, lo que permite un análisis tanto global como específico por dimensión.

En este estudio, el análisis se centró únicamente en el pilar de pensamiento algorítmico, el cual se encuentra presente en 27 de los 28 ítems del instrumento, lo que permitió obtener una medida robusta en dicha dimensión. La elección de trabajar únicamente con el pilar de pensamiento algorítmico, se fundamenta en que la intervención pedagógica desarrollada hacia los estudiantes durante la unidad de fundamentos de algoritmia del curso de informática básica estuvo orientada específicamente al fortalecimiento de dicho pilar, de acuerdo con los contenidos establecidos en el sílabo del curso.

El puntaje obtenido en el pilar del pensamiento algorítmico por cada estudiante, se calculó mediante la suma de los ítems correspondientes, permitiendo comparar el desempeño académico antes y después de la intervención, así como analizar las diferencias entre las herramientas utilizadas en ambos grupos.

D. Procedimiento

El estudio se desarrolló en tres momentos diferenciados. En el primer momento, al inicio de la unidad de fundamentos de algoritmia, se aplicó un pretest a ambos grupos de estudiantes con el objetivo de medir el nivel inicial del pilar de pensamiento algorítmico. Para ello se utilizó el Test de Pensamiento Computacional propuesto por Román-González, el cual evalúa dicho pilar a través de un conjunto de ítems de respuesta dicotómica (correcto/incorrecto). La aplicación del pretest se realizó bajo condiciones controladas y uniformes para ambos grupos, garantizando el mismo tiempo de resolución y las mismas instrucciones.

En el segundo momento se llevó a cabo la intervención pedagógica, donde el grupo A desarrolló las actividades de aprendizaje utilizando la herramienta Scratch, mientras que el grupo B empleó PSeInt. En ambos casos, la intervención estuvo alineada con el sílabo del curso y se centró en el

desarrollo del pensamiento algorítmico. La intervención estuvo basada en contenidos que exige el silabo como secuencias, estructuras de control y resolución de problemas mediante algoritmos. Las sesiones fueron desarrolladas durante la misma unidad académica y con una carga horaria equivalente para ambos grupos, esto con el fin de evitar sesgos derivados del tiempo de exposición o del contenido impartido.

En el tercer momento, al finalizar la unidad de fundamentos de algoritmia, se aplicó un postest a los mismos estudiantes que participaron en el pretest, utilizando nuevamente el Test de Pensamiento Computacional de Román-González. El postest se administró bajo las mismas condiciones que el pretest, lo que permitió comparar de manera directa los resultados obtenidos antes y después de la intervención en cada grupo. Con el desarrollo de los tres momentos se garantizó la consistencia del desarrollo del procedimiento y la validez interna del estudio al recolectar datos del pilar del pensamiento algorítmico.

E. Análisis estadístico de los datos

El procesamiento y análisis de los datos se realizó utilizando el lenguaje de programación Python en el entorno de Google Colaboratory, empleando librerías especializadas para el análisis estadístico. En la primera etapa se llevó a cabo un análisis descriptivo de los puntajes obtenidos únicamente en el pilar de pensamiento algorítmico, calculando medidas de tendencia central y dispersión, tales como la media, valores máximo y mínimo, y desviación estándar, con el fin de caracterizar el desempeño de los estudiantes antes y después de la intervención. Asimismo, se emplearon representaciones gráficas con el fin de visualizar la dispersión y distribución de los datos de manera complementaria, con el fin de comprender los datos obtenidos a nivel descriptivo.

En una segunda etapa, se evaluó el supuesto de normalidad de los datos mediante la prueba de Shapiro-Wilk por tener menos de 50 estudiantes, considerando un nivel de significancia de $\alpha = 0.05$. La prueba de normalidad se aplicó tanto a los puntajes de pensamiento algorítmico obtenidos en el pretest y postest de cada grupo, como a las variables derivadas del cambio entre ambas mediciones.

En una tercera etapa y debido a que las mediciones pretest y postest correspondían a los mismos estudiantes en cada grupo, la comparación intragrupo se realizó mediante la prueba t de Student para muestras relacionadas, siempre que se cumpliera el supuesto de normalidad. Dicha prueba permitió determinar si existían diferencias estadísticamente significativas en el puntaje de pensamiento algorítmico antes y después de la intervención con cada herramienta (Scratch y PSeInt).

En una cuarta etapa, con el propósito de estimar la magnitud de los cambios observados, se calculó el tamaño del efecto utilizando el estadístico de Cohen's d, lo que permitió complementar el análisis inferencial más allá de la significancia estadística.

Finalmente, en una quinta etapa, para comparar el cambio en el puntaje de pensamiento algorítmico entre el grupo que utilizó Scratch y el grupo que empleó PSeInt, se analizó la diferencia individual entre los puntajes postest y pretest. Debido a que una de estas variables no cumplió con el supuesto de normalidad, se aplicó la prueba no paramétrica de Mann-Whitney U, la cual permitió contrastar si el grado de mejora difería significativamente entre ambas herramientas.

III. RESULTADOS

A. Características generales de los datos analizados

El análisis de los datos se realizó considerando únicamente a los estudiantes que completaron tanto el pretest como el postest en cada uno de los grupos. El grupo A, que desarrolló la intervención utilizando Scratch, estuvo conformado por 28 estudiantes, mientras que el grupo B, que utilizó PSeInt, estuvo conformado por 12 estudiantes. Los resultados presentados a continuación corresponden exclusivamente al puntaje obtenido en el pilar de pensamiento algorítmico del Test de Pensamiento Computacional de Román-González.

B. Resultados descriptivos del pensamiento algorítmico

Se desarrolló un análisis descriptivo del puntaje de pensamiento algorítmico en los momentos pretest y postest para ambos grupos. Se calcularon medidas de tendencia central y dispersión, tales como la media y la desviación estándar.

Respecto al grupo de Scratch, se observó un incremento en la media del puntaje de pensamiento algorítmico del pretest al postest. De manera similar el grupo PSeInt mostró un aumento en el puntaje promedio entre ambas mediciones, aunque con una magnitud menor, tal como se muestra en la Tabla I.

TABLA I
ESTADÍSTICOS DESCRIPTIVOS

Grupo	Momento	Media	Des. Est.
Scratch	Pretest	18.2	3.7
Scratch	Postest	19.6	3.8
PSeInt	Pretest	12.6	7.1
PSeInt	Postest	17.0	4.6

Para complementar el análisis descriptivo de los datos, se emplearon representaciones gráficas mediante diagramas de caja (boxplots), los cuales permitieron visualizar la distribución y dispersión de los puntajes en cada grupo y momentos de evaluación que corresponden al pretest y postest, tal como se muestra en la Fig 1 y Fig 2.

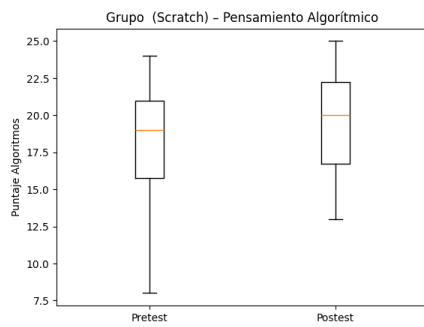


Fig. 1 Distribución y dispersión – Grupo Scratch

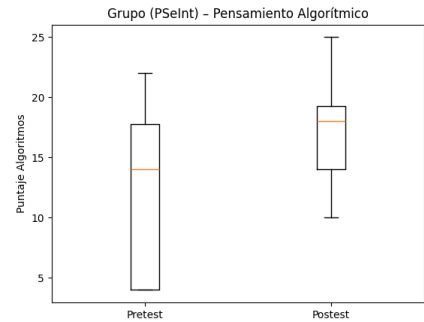


Fig. 2 Distribución y dispersión – Grupo PSeInt

C. Análisis de normalidad de los datos

Para verificar la normalidad de los puntajes de pensamiento algorítmico, se utilizó la prueba de Shapiro–Wilk. Los resultados indicaron que los puntajes correspondientes al pretest y posttest de ambos grupos presentaron una distribución aproximadamente normal ($p > 0.05$), lo que permitió el uso de pruebas estadísticas paramétricas para las comparaciones intragrupo, tal como se muestra en la Tabla II.

TABLA II
PRUEBA DE NORMALIDAD

Grupo	Momento	P-Value
Scratch	Pretest	0.177
Scratch	Posttest	0.125
PSeInt	Pretest	0.065
PSeInt	Posttest	0.505

Seguidamente, al analizar la variable correspondiente al cambio individual ($\Delta = \text{posttest} - \text{pretest}$), se observó que el cambio en el grupo Scratch no cumplió con el supuesto de normalidad ($p < 0.05$), mientras que el cambio en el grupo PSeInt sí presentó una distribución normal ($p > 0.05$) tal como se muestra en la Tabla III. Debido a esta condición, se optó por una prueba no paramétrica para la comparación del cambio entre ambos grupos.

TABLA III
PRUEBA DE NORMALIDAD

Grupo	P-Value
Scratch	0.005
PSeInt	0.639

D. Comparación pretest–posttest por grupo

Grupo Scratch: La comparación entre los puntajes de pensamiento algorítmico obtenidos en el pretest y posttest del grupo Scratch se realizó mediante la prueba *t* de Student para muestras relacionadas. Los resultados mostraron una diferencia estadísticamente significativa entre ambas mediciones ($t = 3.97$, $p = 0.00048$), indicando un incremento significativo del puntaje de pensamiento algorítmico tras la intervención con Scratch.

Grupo PSeInt: Para dicho grupo también se aplicó la prueba *t* de Student para muestras relacionadas en el grupo PSeInt. Sin embargo, para este grupo los resultados no evidenciaron una diferencia estadísticamente significativa entre el pretest y el posttest ($t = 2.02$, $p = 0.068$), al considerar un nivel de significancia de $\alpha = 0.05$.

E. Tamaño del efecto de la intervención

Para poder estimar la magnitud de los cambios observados en cada grupo, se calculó el tamaño del efecto mediante el estadístico de Cohen's *d*. En el grupo Scratch, el tamaño del efecto fue $d = 0.75$, lo que corresponde a un efecto de magnitud moderada a grande. Por otro lado, el grupo PSeInt presentó un tamaño del efecto de $d = 0.58$, correspondiente a un efecto de magnitud moderada. Dichos resultados permiten complementar el análisis inferencial, proporcionando información adicional sobre la relevancia práctica de los cambios observados.

F. Comparación del cambio entre Scratch y PSeInt

Para poder determinar si el grado de mejora en el puntaje de pensamiento algorítmico difería entre ambos grupos, se comparó el cambio individual ($\Delta = \text{posttest} - \text{pretest}$) utilizando la prueba no paramétrica de Mann–Whitney *U*. Los resultados de esta prueba no mostraron diferencias estadísticamente significativas entre el grupo Scratch y el grupo PSeInt en cuanto al cambio del puntaje de pensamiento algorítmico ($U = 116.5$, $p = 0.129$), considerando un nivel de significancia de $\alpha = 0.05$.

IV. DISCUSIÓN

Los resultados obtenidos en el grupo de estudiantes que utilizó la herramienta Scratch muestran un incremento estadísticamente significativo en el pensamiento algorítmico tras la intervención pedagógica en la segunda unidad del curso de informática básica. Esta mejora que mostro los resultados concuerda con trabajos previos que han observado impactos positivos del uso de entornos de programación visual sobre habilidades algorítmicas. Al comparar dichos resultados con Velasco-Ramírez y Otero-Escobar reportaron que una intervención basada en Scratch, aplicada a estudiantes universitarios en una asignatura de algoritmia, produjo una mejora en los puntajes del posttest en comparación con el pretest, lo que respalda la eficacia de Scratch como herramienta para fomentar la lógica y construcción de algoritmos en educación superior [15].

La literatura señala que respecto a estudios que comparan directamente enfoques visuales y textuales, los hallazgos de Rezende y Bispo Jr. sugieren que, al aplicar ambos métodos en la enseñanza de lógica y algoritmos, Scratch tiende a mostrar mejores resultados en las primeras etapas del aprendizaje, mientras que el enfoque con pseudocódigo puede producir mejoras en fases de mayor complejidad cognitiva [16]. Estos resultados mostrados son coherentes con nuestros hallazgos, aunque Scratch produjo una mejora interna significativa en el grupo A, la comparación directa entre las mejoras de ambos grupos (Δ Scratch vs Δ PSeInt) no arrojó diferencias estadísticamente significativas según la prueba de Mann-Whitney U. Esto podría señalar que, si bien cada herramienta es eficaz en contextos particulares del aprendizaje algorítmico, ninguna muestra una ventaja absoluta en términos de magnitud de cambio en el desarrollo de pensamiento algorítmico bajo un diseño cuasi-experimental similar al utilizado en este estudio.

Debido a la ausencia de diferencias significativas en el análisis comparativo entre las mejoras obtenidas con Scratch y con PSeInt (Mann-Whitney U, $p > \alpha$) puede explicarse a partir de las características propias de cada herramienta aplicada. De acuerdo con Rezende y Bispo Jr., puede interpretarse que Scratch facilita la internalización de estructuras algorítmicas iniciales mediante su esquema visual, lo que favorece mejoras tempranas en pensamiento algorítmico, mientras que las ganancias con PSeInt podrían depender de la familiaridad de los estudiantes con la lógica textual y su experiencia previa en programación. Dicha interpretación sugiere que ambos enfoques tienen potencial para desarrollar pensamiento algorítmico, aunque sus beneficios se manifiesten de forma distinta según las condiciones de enseñanza y las características cognitivas de los estudiantes.

Los resultados obtenidos en la presente investigación concuerdan parcialmente con lo reportado por Mladenović *et al.* [17], quienes encontraron que los estudiantes que trabajan con entornos de programación visual presentan menor incidencia de errores algorítmicos en estructuras de control en comparación con aquellos que utilizan programación textual. A pesar de que dicho estudio se desarrolló en el nivel K-12, sus hallazgos refuerzan la idea de que la representación visual de los algoritmos favorece la comprensión inicial de la lógica algorítmica, lo cual resulta coherente con la mejora observada en el grupo que utilizó Scratch en esta investigación, aunque en un nivel educativo distinto, lo que sugiere patrones consistentes en la comprensión algorítmica inicial.

En resumen, estos hallazgos encontrados en la investigación sugieren que tanto las herramientas visuales como las textuales pueden contribuir al desarrollo del pensamiento algorítmico, aunque las formas en que lo hacen pueden variar según el diseño instruccional y las características de los estudiantes. La contribución principal de este estudio radica en ofrecer evidencia cuantitativa comparativa en el contexto universitario no informático, utilizando herramientas validadas y un diseño cuasi-

experimental con mediciones previas y posteriores. Este enfoque permite una comprensión más fina del impacto diferencial de Scratch y PSeInt, lo cual representa un avance en la literatura respecto a los estudios existentes que, mayoritariamente, han explorado estos enfoques de manera separada e independiente.

V. CONCLUSIONES

La presente investigación ha permitido analizar comparativamente el efecto de dos herramientas de enseñanza de algoritmia, denominados Scratch y PSeInt, sobre el desarrollo del pensamiento algorítmico en estudiantes universitarios de carreras no informáticas. Los resultados han mostrado que el uso de la herramienta Scratch mostró una mejora estadísticamente significativa en los puntajes de pensamiento algorítmico entre el pretest y el posttest, además presentó un tamaño del efecto de magnitud moderada-alta, lo que sugiere un impacto consistente de esta herramienta en contextos de formación inicial de algoritmos.

Respecto a la herramienta PSeInt, se observó un incremento en el puntaje promedio de pensamiento algorítmico, sin embargo, dicho cambio no alcanzó significancia estadística, aunque el tamaño del efecto obtenido indica una mejora de magnitud moderada. Estos resultados sugieren que el uso de herramientas basadas en pseudocódigo textual también contribuye al desarrollo del pensamiento algorítmico, aunque de manera menos pronunciada en comparación con entornos de programación visual como la herramienta Scratch.

Sin embargo, la comparación directa del cambio en el pensamiento algorítmico entre ambos grupos no mostró diferencias estadísticamente significativas, lo que indica que, desde un enfoque comparativo, ninguna de las dos herramientas supera de forma concluyente a la otra en términos de mejora global. Este resultado refuerza la idea de que tanto Scratch como PSeInt constituyen alternativas pedagógicas válidas para la enseñanza de algoritmia en las carreras de nivel universitario, siendo su efectividad dependiente del contexto educativo y del perfil de los estudiantes.

A la vista de los resultados del estudio sugieren que Scratch puede resultar especialmente adecuado en etapas iniciales del aprendizaje algorítmico, particularmente en carreras no orientadas a la informática, al facilitar la comprensión de estructuras algorítmicas mediante representaciones visuales como bloques. Por otro lado, la herramienta PSeInt continúa siendo una herramienta pertinente para el fortalecimiento progresivo del pensamiento algorítmico hacia la transición de lenguajes de programación textuales tradicionales.

Como recomendación, se plantea extender el análisis a los demás pilares del pensamiento computacional: abstracción, descomposición y reconocimiento de patrones, así como

incorporar variables actitudinales y motivacionales que permitan comprender de manera más integral el impacto de estas herramientas en el proceso de aprendizaje de algoritmos. En términos generales, este estudio aporta evidencia empírica que contribuye a la toma de decisiones pedagógicas en la enseñanza inicial de algoritmia en educación superior, particularmente en contextos no informáticos.

AGRADECIMIENTO/RECONOCIMIENTO

Se agradece a los alumnos de la Universidad Nacional de San Agustín, especialmente de las carreras de ingeniería, así como a los colegas que contribuyeron al desarrollo de este estudio.

REFERENCIAS

- [1] M. Romero, A. Lepage, and B. Lille, "Computational thinking development through creative programming in higher education," *Int. J. Educ. Technol. High. Educ.*, vol. 14, no. 1, 2017, doi: 10.1186/s41239-017-0080-z.
- [2] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, 2006, doi: 10.1145/1118178.1118215.
- [3] V. J. Shute, C. Sun, and J. Asbell-Clarke, "Demystifying computational thinking," *Educ. Res. Rev.*, vol. 22, pp. 142–158, 2017, doi: <https://doi.org/10.1016/j.edurev.2017.09.003>.
- [4] M. Román-González, J.-C. Pérez-González, and C. Jiménez-Fernández, "Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test," *Comput. Human Behav.*, vol. 72, pp. 678–691, 2017, doi: <https://doi.org/10.1016/j.chb.2016.08.047>.
- [5] S. Grover and R. Pea, "Computational Thinking in K–12: A Review of the State of the Field," *Educ. Res.*, vol. 42, no. 1, pp. 38–43, 2013, doi: 10.3102/0013189X12463051.
- [6] A. Robins, J. Rountree, and N. Rountree, "Learning and Teaching Programming: A Review and Discussion," *Comput. Sci. Educ.*, vol. 13, no. 2, pp. 137–172, 2003, doi: 10.1076/csed.13.2.137.14200.
- [7] M. Resnick *et al.*, "Scratch: Programming for all," *Commun. ACM*, vol. 52, no. 11, pp. 60–67, 2009, doi: 10.1145/1592761.1592779.
- [8] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, "The scratch programming language and environment," *ACM Trans. Comput. Educ.*, vol. 10, no. 4, 2010, doi: 10.1145/1868358.1868363.
- [9] A. López-García and J. Urquiza-Fuentes, "Diseño de pseudocódigo, comprensión y aprendizaje con estudiantes de ciclos formativos de informática," pp. 61–76, 2024, [Online]. Available: <https://www.researchgate.net/publication/385130477>.
- [10] A. López-García and J. Urquiza-Fuentes, "Comprensión de pseudocódigo básico en estudiantes de ciclos formativos," *RISTI - Rev. Ibérica Sist. e Tecnol. Informação*, no. 51, pp. 37–50, 2023, doi: 10.17013/risti.37-50.
- [11] L. A. Calao, J. Moreno-León, H. E. Correa, and G. Robles, "Developing Mathematical Thinking with Scratch BT - Design for Teaching and Learning in a Networked World," 2015, pp. 17–27.
- [12] W. Stewart and K. Baek, "Analyzing computational thinking studies in Scratch programming: A review of elementary education literature," *Int. J. Comput. Sci. Educ. Sch.*, vol. 6, no. 1, pp. 35–58, 2023, doi: 10.21585/ijcses.v6i1.156.
- [13] R. Hijón-Neira, C. Pizarro, J. French, D. Palacios-Alonso, and E. Çoban, "Computational Thinking Measurement of CS University Students," *Appl. Sci.*, vol. 14, no. 12, p. 5261, 2024.
- [14] A. López-García, J. Urquiza-Fuentes, A. J. Mendes, and M. Caeiro-Rodríguez, "Improving University Students' Learning of Programming Using Customised Pseudocode," *Comput. Appl. Eng. Educ.*, vol. 33, no. 4, 2025, doi: 10.1002/cae.70061.
- [15] M. Velasco-Ramírez and A. Otero-Escobar, "El impacto del uso de Scratch para favorecer el pensamiento algorítmico en estudiantes de educación superior," *Rev. Innova Educ.*, vol. 5, no. 4, pp. 105–120, 2023, doi: <https://doi.org/10.35622/j.rie.2023.05v.006>.
- [16] C. M. C. Rezende and E. L. Bispo, "Comparison between pseudocode usage and visual programming with scratch in programming teaching," *Proc. - 13th Lat. Am. Conf. Learn. Technol. LACLO 2018*, pp. 492–498, 2018, doi: 10.1109/LACLO.2018.00087.
- [17] M. Mladenović, I. Boljat, and Ž. Žanko, "Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level," *Educ. Inf. Technol.*, vol. 23, no. 4, pp. 1483–1500, 2018, doi: 10.1007/s10639-017-9673-3.