

La enseñanza-aprendizaje de programación en computadora con PSeint: Una revisión sistemática.
Teaching Learning Computer Programming Whith Pseint: A Systematic Review

MSc. Luesner Hernán Navarrete Mora; MSc. Giovanni Antonio Freire Jaramillo; MSc. Giovanni David Fernández Unuzungo;
MSc. Edder Joffre Gilces Loo; Ph.D. Carlos Mego Cubas

INVESTIGACIÓN Y
DESARROLLO TECNOLÓGICO

Julio - diciembre, V⁴-N²;
2023

- ✓ **Recibido:** 21/12/2023
- ✓ **Aceptado:** 29/12/2023
- ✓ **Publicado:** 30/12/2023

PAÍS

- Pa Guayaquil-Ecuador
- Pa Daule-Ecuador
- Pa Santo Domingo-Ecuador
- Pa Guayaquil-Ecuador
- Pa Lima-Perú

INSTITUCIÓN

- Universidad César Vallejo-Perú
- Universidad César Vallejo-Perú
- Universidad César Vallejo-Perú
- Universidad César Vallejo-Perú
- Universidad César Vallejo-Perú

CORREO:

- M lnavarretemo26@ucvvirtual.edu.pe
- M p7002270553@ucvvirtual.edu.pe
- M p7002489954@ucvvirtual.edu.pe
- M p7002272558@ucvvirtual.edu.pe
- M cmegoc@ucvvirtual.edu.pe

ORCID:

- ORCID <https://orcid.org/0000-0001-5909-509X>
- ORCID <http://orcid.org/0000-0002-2986-2865>
- ORCID <http://orcid.org/0000-0002-3157-8556>
- ORCID <http://orcid.org/0000-0002-9802-8401>

FORMATO DE CITA APA.

Navarrete, L. Freire, G. Fernández, G. Gilces, E. Mego, C. (2023). La enseñanza-aprendizaje de programación en computadora con PSeint: Una revisión sistemática. Revista G-ner@ndo, V⁴ (N²). 923 – 936.

Resumen

El presente artículo aborda la revisión sistemática de literatura sobre la enseñanza aprendizaje de la programación en Pseint, Concretamente con este trabajo se da respuesta a interrogantes como: ¿Cuáles es el programa, para la enseñanza aprendizaje de la programación, de código abierto diseñado para ayudar a los estudiantes a aprender los fundamentos de la programación? ¿Cuáles son los tipos de estrategias para el análisis y la solución de un problema de programación por medio de un programa computacional? Se analizaron producciones científicas en el período de 2014 a 2022. El estudio se apoya en la metodología propuesta Moher, Liberati, Tetzlaff & Alman (2014). Las producciones revisadas aportan datos teóricos en el ámbito de la enseñanza de la programación, Los resultados muestran que la enseñanza y el aprendizaje a la programación han sido de gran interés que, por medio del desarrollo del pensamiento, en pseudocódigo y diagrama de flujo, ayudan para llegar a la solución de un problema.

Palabras claves: enseñanza, aprendizaje, programación, computacional, resolver problemas.

Abstract

This article deals with the systematic review of literature on the teaching of programming, specifically with this work answers to questions such as: What is the program, for teaching learning programming, open source designed to help students to learn the fundamentals of programming? What are the types of strategies for the analysis and solution of a programming problem by means of a computer program? Scientific productions in the period from 2014 to 2022 were analyzed. The study is based on the constructivist methodology proposed by Moher, Liberati, Tetzlaff & Alman (2014). The revised productions provide data theoretical in the field of teaching programming, the results show that teaching and learning programming have been of great interest that, through the development of thinking, in pseudocode and flowchart, help to reach the solution of a problem.

Keywords: teaching, learning, programming, computational, problem solving.

Introducción

La relevancia de instruir en programación durante la etapa inicial de educación de docentes de Informática radica en la resolución de problemas mediante lenguajes de programación. Para lograr esto, es esencial que los estudiantes adquieran habilidades específicas como programación lógica, análisis de flujos de datos, y técnicas de programación. Esto incluye también el trabajo colaborativo para la creación de programas funcionales, fomentando el intercambio de conocimientos y la comunicación efectiva, según Jiménez, Collazos y Revelo (2019). A pesar del progreso tecnológico actual, se identifican falencias en la base del aprendizaje de programación, usualmente empezando en el primer año de educación técnica. Estas deficiencias pueden atribuirse a una falta de comprensión de los principios básicos de programación, limitada habilidad en desarrollo de algoritmos, dificultades en la modelación y construcción de programas, y una insuficiente disciplina cognitiva para abordar los fundamentos de la programación.

El aprendizaje de la programación va más allá de la mera codificación; implica la resolución de problemas y la automatización mediante computadoras, como señalan Kuz y Ariste (2021). Según Fiallos, Jiménez y Sucursal (2022), esto conlleva el desarrollo de habilidades cognitivas avanzadas. Programar es un proceso integral que involucra múltiples destrezas y competencias. Al aprender a programar, los estudiantes no solo adquieren habilidades técnicas, sino también capacidades cruciales para resolver problemas, aplicables más allá del ámbito de la computación, útiles en diversas facetas de la vida diaria, independientemente de la edad, origen, intereses o profesión, como indican Arellano, Fernández, Rosas y Zúñiga (2014).

La programación informática implica la creación y estructuración de instrucciones que dirigen el funcionamiento de un software o computadora en un idioma que esta última pueda procesar, usualmente mediante lenguajes de programación. Esta práctica sirve como un medio para ensayar soluciones, destacando la importancia de la planificación previa, incluyendo el uso

de técnicas como el bosquejo en papel, enfatizando en la reflexión antes de la programación y ejecución. Según Fracchia, Kogan y Amaro (2016), la habilidad para transferir estos conocimientos y habilidades a futuras situaciones laborales es un beneficio clave en este campo en constante evolución. Machuca, Sampedro, Palma y Cañizares (2021) sugieren que un enfoque efectivo para la enseñanza y aprendizaje de la programación es comenzar con el uso de algoritmos como herramientas esquemáticas para la resolución de problemas.

Un algoritmo es una secuencia definida y limitada de pasos que busca resolver un problema, procesando información de entrada hacia un resultado específico, como define Ramírez (2015). Pinzón y González (2022) consideran el algoritmo como una herramienta esencial en el aprendizaje. Según Cabra y Ramírez (2021), su finalidad es abordar desafíos comunes de manera estructurada. Parra, Amariles y Castro (2016) sugieren que los docentes fomenten la resolución de problemas en los estudiantes, tanto individual como colectivamente, para desarrollar esta competencia. Vidal, Cabezas, Parra y López (2015) recomiendan el uso de recursos visuales y simbólicos para enseñar el proceso de resolución mediante algoritmos.

Pseint es una herramienta de software que interpreta algoritmos escritos en pseudocódigo o representados mediante diagramas de flujo. Es un programa de acceso libre y gratuito, destacando por su editor que presenta sintaxis resaltada en colores, autocompletado y asistencia en pantalla. Ofrece la capacidad de convertir algoritmos en diagramas de flujo y ejecutarlos si son válidos, identificando errores en caso contrario, como describen Huari y Novara (2014).

Pseudocódigos y diagramas de flujo son fundamentales en la representación de algoritmos, siendo esenciales en la educación de programación, según Villalobos, De la Ossa, Rodríguez y Vergara (2020). El Pseudocódigo es adaptable a las preferencias del instructor, incluyendo estructuras de control estándar y funciones avanzadas. PSeInt facilita la conversión entre pseudocódigo y diagramas de flujo, permitiendo ediciones en ambos formatos. Arellano,

Solar, Nieva y Canelo (2022) destacan que los diagramas de flujo mejoran la resolución de problemas y la lógica de programación.

Este artículo busca exponer los resultados de un análisis sistemático de la literatura sobre el proceso de enseñanza y aprendizaje en programación de computadoras usando pseudocódigos. Este estudio examina los desafíos que enfrentan estudiantes y profesores, así como las estrategias actuales en la educación de programadores, según los informes de la literatura científica.

Métodos y materiales

Una revisión sistemática, según Guiao (2015), es un enfoque metódico para abordar una pregunta específica, aplicando procesos claros para evaluar y sintetizar investigaciones relevantes. Esta metodología, alineada con los principios de PRISMA descritos por Moher, Liberati, Tetzlaff y Alman (2014), busca organizar eficientemente la evidencia sobre enseñanza de programación en Pseint. Incluye pasos detallados como definir el propósito, buscar y cribar literatura, evaluar calidad, extraer datos, y sintetizar estudios para garantizar la replicabilidad de la investigación.

Para determinar las metas de la revisión, se partió de las siguientes preguntas de investigación

¿Cuáles es el programa, para la enseñanza aprendizaje de la programación, de código abierto diseñado para ayudar a los estudiantes a aprender los fundamentos de la programación?

¿Cuáles son los tipos de estrategias para el análisis y la solución de un problema de programación por medio de un programa computacional?

Se definió un periodo de búsqueda de ocho años, abarcando desde enero de 2014 hasta octubre de 2022. Durante esta última década, se identificaron tres artículos en 2021 y 2022 sobre enseñanza de programación en pseudocódigo y dos artículos en 2014, 2015 y 2017 enfocados en estrategias para resolver problemas de programación.

La revisión se basó en datos de Scielo, Dialnet, Redalyc y Scopus, utilizando una estrategia de búsqueda específica para hallar información pertinente sobre metodologías de revisión literaria.

La estrategia de búsqueda utilizó combinaciones de términos específicos, por ejemplo, [enseñanza y aprendizaje en programación] junto con [programación en pseudocódigo en la enseñanza]. Para delimitar el alcance de la investigación, se definieron criterios específicos de inclusión y exclusión.

- Trabajos publicados entre 2014 y 2022.
- Estudios cuantitativos, cualitativos o mixtos.
- Artículos en español.
- Disponibilidad de texto completo.

La selección de recursos estuvo estrechamente vinculada con los objetivos del estudio, enfocándose en recursos que contienen términos de búsqueda relevantes para las preguntas definidas en la etapa de planificación.

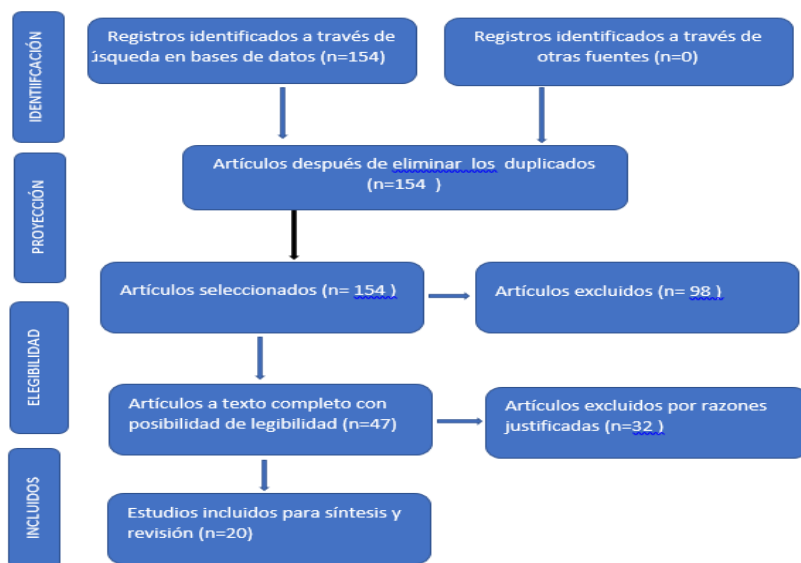


Figura 1. Diagrama de flujo PRISMA. Fuente elaboración propia

Fuente	Descriptor	Resultados
Scopus	Enseñanza aprendizaje en programación	1
	La enseñanza de la programación de computadora	0
Scielo	Enseñanza aprendizaje en programación	80
	Pensamiento algorítmico	3
	Enseñanza aprendizaje en programación	959
	La enseñanza de la programación de computadora	229
Dialnet	La enseñanza de la programación en seudocódigo	6
Redalyc	La enseñanza de la programación de computadora en seudocódigo	403
Total		1681

Gráfico 1: Número de resultados por cada fuente, Fuente elaboración propia.

La definición de estos criterios fue clave para filtrar la información, excluyendo aquellos artículos que no estaban suficientemente alineados con los objetivos de la investigación. La Figura 1 ilustra el proceso de búsqueda y selección, el cual se adhirió a las directrices PRISMA para asegurar transparencia y claridad en el procedimiento.

Análisis de resultado

La siguiente sección detalla los resultados de la investigación, particularmente los hallazgos cuantitativos derivados de las preguntas previamente formuladas. De acuerdo con la metodología establecida, se seleccionaron 20 artículos. Estos fueron catalogados en una matriz que incluía detalles cruciales como autor(es), año de publicación, objetivo, metodología, conclusiones y categoría temática. La Figura 2 exhibe cómo se distribuyen estos artículos a lo largo de los años.

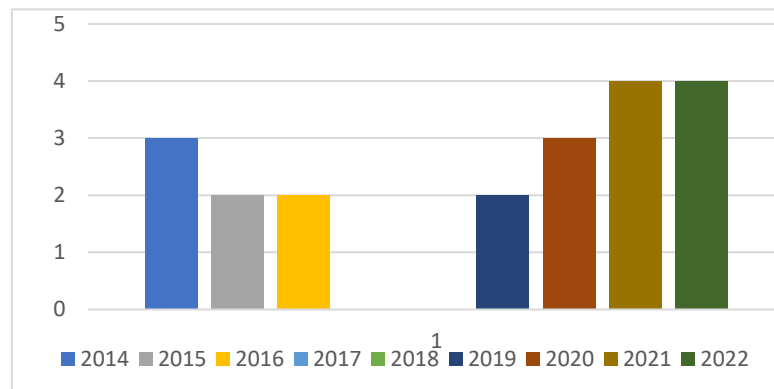


Gráfico 2. Distribución artículos por año. Fuente: elaboración propia

El Gráfico 2 ilustra que en 2021 y 2022 se incrementó el número de estudios sobre la enseñanza y el aprendizaje de la programación. Durante 2014, 2015, 2016 y 2019, se registraron dos publicaciones anuales sobre el tema. Sin embargo, en 2017 y 2018 no se identificaron

AÑO	N°	AUTORES
2014	3	Huari Evangelista, F., & Novara, P. J; Arellano, N., Fernandez, J., Rosas, M. V., & Zuñiga, M. E Salgado Castillo, A., Alonso Berenguer, I., & Gorina Sánchez, A.
2015	2	Vidal, C., Cabezas, C., Parra, J., & López, L. Ramirez, E
2016	2	Parra Castrillón, J. E., Amariles Camacho, M. J., & Castro Castro, C. A; Fracchia, C. C., Kogan, P., & Amaro, S.
2017		
2018		
2019	2	Jiménez Toledo, J. A., Collazos, C., & Revelo Sánchez, O; Beúnes Cañete, J. E., & Vargas Ricardo, A
2020	3	Villalobos Fernández, L. A., De la Ossa Osegueda, Á., Rodríguez-Villagra, O. A., & Vergara Heidke, A. E. Ramírez Prada, P. C., Ortiz Beltrán, A. O., & Lobo Quintero, R. A. Velasco Ramírez, M. L.
2021	4	Machuca Vivar, S. A., Sampedro Guamán, C. R., Palma Rivera, D. P., & Cañizares Galarza, F. P; Kuz, A., & Ariste, M. Cabra Páez, M. L., & Ramírez Gamboa, S. A; Fonden Calzadilla, J. C.
2022	4	Fiallos Quinteros, J., Jiménez Builes, J., & Sucursal Bedoya, J; Arellano Pimentel, J., Solar González, R., Nieva García, . Martínez Allende, L., García Monroy, A. I., & Linares González, E. E. O. S., & Canedo Ibarra, S. P; Pinzón Pérez, D. F., & González Palacio, E. V.

Gráfico 3. Autores de los artículos clasificado por año de edición. Fuente: Elaboración propia

publicaciones relevantes, y en 2020, solo se encontró una publicación sobre este ámbito.

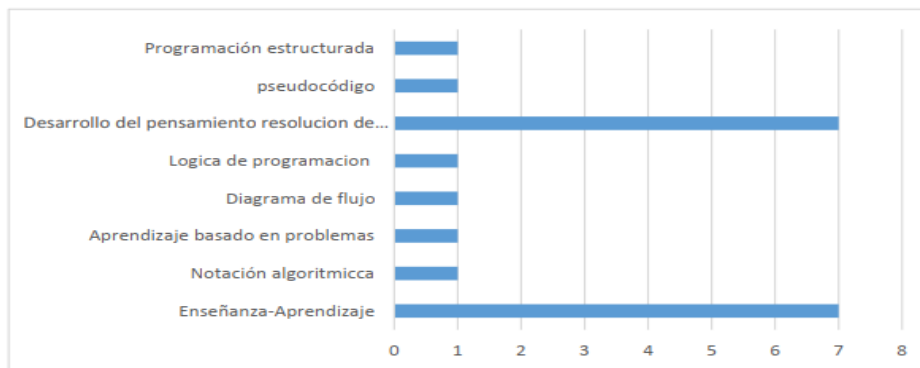


Gráfico 4. Palabras clave más utilizadas. Fuente: elaboración propia

En el análisis temático inicial basada en las palabras clave, se encontró que 7 artículos se centraron en la enseñanza y el aprendizaje de la programación, mientras que otros 7

abordaron el desarrollo del pensamiento crítico para la solución de problemas, siendo estos los temas más frecuentemente investigados.

En el análisis inicial basado en palabras clave de los 20 artículos examinados, se encontró que el 35% se enfocó en la enseñanza y aprendizaje de la programación, y otro 35% en el desarrollo del pensamiento para resolver problemas. El resto de los estudios, cada uno representando un 5%, se centraron en temas como programación estructurada, pseudocódigo, lógica de programación, diagramas de flujo, aprendizaje basado en problemas y notación algorítmica.

El Gráfico 5 detalla la distribución de artículos según descriptores y año de publicación. En 2014, hubo una publicación sobre resolución de problemas y dos en enseñanza de programación. En 2015, 2016, y 2020, se registró un artículo en cada tema. No se encontraron publicaciones relevantes en 2017 y 2018. En 2019, dos artículos se enfocaron en enseñanza de programación, mientras que, en 2021, hubo un artículo sobre resolución de problemas y tres sobre enseñanza. En 2022, cinco artículos abordaron la resolución de problemas y uno la enseñanza de programación.

Gráfico 5. Descriptores por año. Fuente: elaboración propia



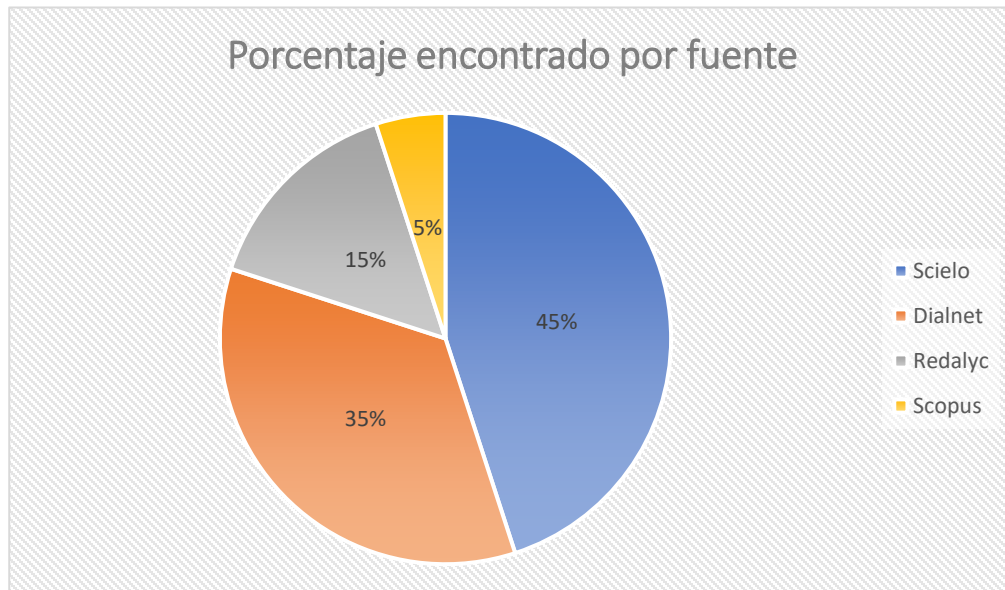


Gráfico 6. Porcentaje de artículos encontrados por fuente. Fuente: elaboración propia

El Gráfico 6 expone el porcentaje de artículos obtenidos de distintas bases de datos. La mayoría de los artículos, un 45%, provinieron de Scielo, seguido por Dialnet con un 35%. Redalyc aportó un 15%, mientras que Scopus contribuyó con un 5% de los artículos, basándose en una estrategia de búsqueda orientada a la metodología para realizar revisiones literarias.

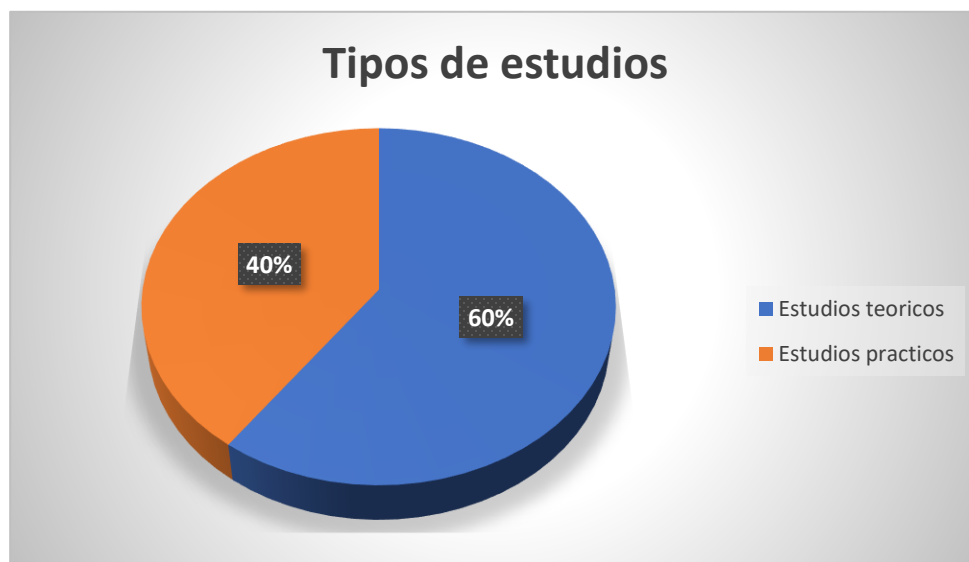


Gráfico 7. Tipo de estudio. Fuente: elaboración propia.

En cuanto a la naturaleza de los estudios, se observó que el 60% eran teóricos y el 40% prácticos. Esto indica una prevalencia de estudios teóricos en la enseñanza y aprendizaje de programación, aunque los estudios prácticos han ganado más relevancia desde el año 2020.

¿Cuáles es el programa, para la enseñanza aprendizaje de la programación, de código abierto diseñado para ayudar a los estudiantes a aprender los fundamentos de la programación?

Los estudiantes muestran desinterés debido a la falta de métodos didácticos innovadores y entornos estimulantes para el aprendizaje. Según Martínez, García y Linares (2022), las clases tienden a ser tediosas, limitándose a copiar códigos. Fonden (2021) afirma que el aprendizaje de programación requiere una planificación y ejecución cuidadosa, integrando aspectos motivacionales y cognitivos. La habilidad para asimilar contenidos y utilizar herramientas tecnológicas como Pseint es fundamental en el proceso educativo, transformando la enseñanza y el aprendizaje.

Es una herramienta eficaz para aprender la lógica de programación y desarrollar programas o algoritmos computacionales. Beunes y Vargas (2019) destacan su utilidad para representar y analizar problemas, especialmente para principiantes en programación. Ofrece un pseudocódigo claro e intuitivo en español, y un editor de diagramas de flujo, facilitando la comprensión de los conceptos básicos de algoritmos. Además, permite ejecutar algoritmos para visualizar su funcionamiento y comprobar resultados, proporcionando un entorno enriquecido con recursos didácticos.

¿Cuáles son los tipos de estrategias para el análisis y la solución de un problema de programación por medio de un programa computacional?

Velasco (2020) enfatiza que analizar y resolver problemas computacionales es un objetivo clave en ciencias computacionales. Estos desafíos se abordan con una mezcla de estrategias de aprendizaje, ayudando a comprender la esencia del problema y sus posibles soluciones. Ramírez, Ortiz y Lobo (2020) señalan que los estudiantes aprenden a evaluar y tomar decisiones

sobre estas cuestiones. Según Salgado, Alonso y Gorina (2014), resolver un problema computacional implica diseñar instrucciones secuenciales para la computadora, fomentando así el desarrollo de habilidades en los estudiantes para crear y ajustar programas.

Thompson (1996), citado por Velasco (2020), describe las etapas para solucionar problemas computacionales como: comprensión, diseño, codificación y evaluación del problema. Según Mac Gaul de Jorge (2008), también citado por Velasco (2020), entender un problema involucra reconocer términos clave, identificar entradas/salidas, diseñar casos de prueba, seleccionar componentes, y conectar con conocimientos previos para diseñar el algoritmo y realizar pruebas de verificación.

Conclusiones

La revisión sistemática de literatura sobre la enseñanza-aprendizaje de la programación en PSeint revela un enfoque significativo en la integración de habilidades de pensamiento algorítmico en la resolución de problemas en el contexto de la educación básica secundaria. Existe un interés creciente en el desarrollo del pensamiento computacional y las competencias matemáticas para la resolución de problemas, lo que sugiere la importancia de integrar la programación en el ámbito educativo.

La distribución de artículos según descriptores y año de publicación muestra un aumento en la atención a la resolución de problemas y la enseñanza de programación en los últimos años, lo que refleja un mayor interés en estos temas. La revisión sistemática destaca la importancia de programas de código abierto, como PSeint, en la enseñanza de los fundamentos de la programación, lo que sugiere su relevancia en el ámbito educativo. La metodología utilizada para la revisión sistemática, basada en criterios específicos y directrices PRISMA, garantiza la transparencia y claridad en el proceso de selección de los artículos, lo que fortalece la validez de los hallazgos.

Referencias bibliograficas

- Pinzón Pérez, D. F., & González Palacio, E. V. (2022). Incidencia de las habilidades de pensamiento algorítmico en las habilidades de resolución de problemas: una propuesta didáctica en el contexto de la educación básica secundaria. *Scielo/Estud. pedagóg. vol.48 no.2*. Obtenido de https://www.scielo.cl/scielo.php?pid=S0718-07052022000200415&script=sci_arttext&tlng=en
- Arellano Pimentel, J., Solar González, R., Nieva García, O. S., & Canedo Ibarra, S. P. (2022). Compilador e intérprete en línea de diagramas de flujo con fines. *Dialnet/rit*. doi:<https://doi.org/10.36825/RITI.10.20.007>
- Arellano, N., Fernandez, J., Rosas, M. V., & Zuñiga, M. E. (junio de 2014). Estrategia metodológica de la enseñanza de la programación para la permanencia de los alumnos de primer año de Ingeniería Electrónica. *Scielo/Rev. iberoam. tecnol. educ. educ. tecnol. no.13*.
- Beúnes Cañete, J. E., & Vargas Ricardo, A. (2019). La introducción de la herramienta didáctica PSeInt en el proceso de enseñanza aprendizaje: una propuesta para Álgebra Lineal. *Scielo/trf vol.15 no.1*. Obtenido de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2077-29552019000100147
- Cabra Páez, M. L., & Ramírez Gamboa, S. A. (18 de Febrero de 2021). Desarrollo del pensamiento computacional y las competencias matemáticas en análisis y solución de problemas: una experiencia de aprendizaje con Scratch en la plataforma Moodle. *Redalyc/Revista Educación, vol. 46, núm. 1, 1-30*. doi:<https://doi.org/10.15517/revedu.v46i1.44970>
- Fiallos Quinteros, J., Jiménez Builes, J., & Sucursal Bedoya, J. (2022). Analítica de enseñanza y aprendizaje aplicada a cursos de programación | [Analítica de enseñanza y aprendizaje cursos de programación. *Scopus/Campus Virtuales*, 35-49.
- Fonden Calzadilla, J. C. (2021). El pensamiento analítico - asociativo en la enseñanza y aprendizaje de la Programación. *Scielo/EduSol vol.21 no.7*. doi:<http://orcid.org/0000-0001-7478-8628>
- Fracchia, C. C., Kogan, P., & Amaro, S. (2016). Competir + Motivar + Hornero = aprender programación. *Scielo/Rev. iberoam. tecnol. educ. educ. tecnol. no.18*. Obtenido de http://www.scielo.org.ar/scielo.php?pid=S1850-99592016000200003&script=sci_arttext&tlng=en

- Guirao Goris, S. A. (2015). Utilidad y tipos de revisión de literatura. *Scielo/Ene. vol.9 no.2*. doi:<https://dx.doi.org/10.4321/S1988-348X2015000200002>
- Huari Evangelista, F., & Novara, P. J. (2014). Intérprete para probar un programa escrito en pseudocódigo. *Dialnet/Revista de la Facultad de Ingeniería Industrial*. Obtenido de file:///C:/Users/Docente/Downloads/Dialnet-InterpreteParaProbarUnProgramaEscritoEnPseudocodig-8635435.pdf
- Jiménez Toledo, J. A., Collazos, C., & Revelo Sánchez, O. (2019). Consideraciones en los procesos de enseñanza-aprendizaje para un primer curso de programación de computadores: una revisión sistemática de la literatura. *Redalyc/TecnoLógicas, vol. 22, 84-118*. doi:<https://doi.org/10.22430/22565337.1520>
- Kuz, A., & Ariste, M. C. (2021). Un análisis desde la programación estructurada del lenguaje Scratch como entorno lúdico educativo. *Dialnet/Revista Iberoamericana de Informática Educativa*. Obtenido de file:///C:/Users/Docente/Downloads/Dialnet-UnAnalisisDesdeLaProgramacionEstructuradaDelLengua-7985875.pdf
- Machuca Vivar, S. A., Sampedro Guamán, C. R., Palma Rivera, D. P., & Cañizares Galarza, F. P. (2021). Desarrollo de la lógica de programación en estudiantes de sistemas de Uniandes Santo Domingo. *Scielo/Conrado vol.17 no.79*. Obtenido de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1990-86442021000200214
- Martínez Allende, L., García Monroy, A. I., & Linares González, E. E. (2022). El juego, estrategia pedagógica en la enseñanza de la programación y elaboración de algoritmos. *Scielo/RIDE. Rev. Iberoam. Investig. Desarro. Educ vol.13 no.25*. doi:<https://doi.org/10.23913/ride.v13i25.1267>
- Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. G. (2014). Ítems de referencia para publicar Revisiones Sistemáticas y Metaanálisis:. *Dialnet/Revista Española de Nutrición Humana y Dietética*. Obtenido de Dialnet-ItemsDeReferenciaParaPublicarRevisionesSistematica-4821653.pdf
- Parra Castrillón, J. E., Amariles Camacho, M. J., & Castro Castro, C. A. (2016). Aprendizaje basado en problemas en el camino a la innovación en ingeniería. *Dialnet/Ingenierías USBMed, Volumen 7, No. 2, 96-103*. Obtenido de file:///D:/Downloads/Dialnet-AprendizajeBasadoEnProblemasEnEICaminoALaInnovacio-6007713.pdf
- Ramírez Prada, P. C., Ortiz Beltrán, A. O., & Lobo Quintero, R. A. (2020). Diagnóstico y reestructuración pedagógica en la enseñanza de programación de computadores en ingeniería de sistemas. *Redalyc/Revista Interamericana de Investigación, Educación y Pedagogía, vol. 13, núm. 2, 245-270*. doi:<https://doi.org/10.15332/25005421.6000>
-

- Ramírez, E. (2015). Alpha: una notación algorítmica basada en pseudocódigo. *Redalyc/Télématique*, vol. 14, núm. 1, 97-121. Obtenido de <https://www.redalyc.org/pdf/784/78435427007.pdf>
- Salgado Castillo, A., Alonso Berenguer, I., & Gorina Sánchez, A. (2014). Ejemplificación de la solución algorítmica de problemas de programación computacional. *Dialnet*. Obtenido de [Dialnet-EjemplificacionDeLaSolucionAlgoritmicaDeProblemasD-6572946%20\(1\).pdf](#)
- Velasco Ramírez, M. L. (2020). Resolución de problemas algorítmicos y objetos de aprendizaje: una revisión de la literatura. *Scielo/RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo*. doi:<https://doi.org/10.23913/ride.v10i20.630>
- Vidal, C., Cabezas, C., Parra, J., & López, L. (2015). Experiencias Prácticas con el Uso del Lenguaje de Programación Scratch para Desarrollar el Pensamiento Algorítmico de Estudiantes en Chile. *Scielo/Form. Univ. vol.8 no.4*. doi:<http://dx.doi.org/10.4067/S0718-50062015000400004>
- Villalobos Fernández , L. A., De la Ossa Osegueda, Á., Rodríguez-Villagra, O. A., & Vergara Heidke, A. E. (2020). Resolución de diagramas de flujo y pseudocódigo por parte de estudiantes de Ciencias de la Computación de la Universidad de Costa Rica. *Dialnet*, 129-146. Obtenido de <file:///C:/Users/Docente/Downloads/Dialnet-ResolucionDeDiagramasDeFlujoYPseudocodigoPorParteD-7659494.pdf>
-