

<https://doi.org/10.23913/ride.v10i20.630>

Artículos Científicos

Resolución de problemas algorítmicos y objetos de aprendizaje: una revisión de la literatura

Resolution of Algorithmic Problems and Learning Objects: A Review of the Literature

Resolução de problemas algorítmicos e objetos de aprendizagem: uma revisão da literatura

María Luisa Velasco Ramírez

Universidad Veracruzana, México

lvelasco@uv.mx

<https://orcid.org/0000-0003-1491-7429>

Resumen

El proceso de resolución de problemas algorítmicos ha sido identificado como conflictivo entre los estudiantes universitarios del área de ciencias de la computación e informática. Para tratar de entender cuáles son las dificultades que aparecen reiteradamente en la resolución de dichos problemas, así como si los objetos de aprendizaje se han utilizado como estrategia tecnopedagógica para disminuirlas, y en caso de que así sea identificar cuáles, se realizó un mapeo sistemático de literatura especializada en el tema. A partir de las preguntas de investigación definidas, inicialmente se realizó una búsqueda en bases de datos como la Education Resources Information Center (ERIC) y la del Institute of Electrical and Electronics Engineers (IEEE), y se seleccionaron 35 estudios publicados en los últimos 10 años; en una segunda fase, siguiendo los criterios de calidad definidos, se redujo la cifra a 26. La revisión arrojó la necesidad de desarrollar en los estudiantes la capacidad de abstracción y análisis para identificar y comprender el problema. También se observó el interés en la definición y manipulación de estrategias que fomenten el pensamiento algorítmico como habilidad en la resolución de problemas, distinguiendo a los objetos de aprendizaje como una alternativa para su desarrollo.



Palabras clave: algoritmos, enseñanza superior, mapeo sistemático de literatura, objetos de aprendizaje.

Abstract

The process of solving algorithmic problems has been identified as a great conflict topic between university students in computer science area. To try to understand those difficulties that appear repeatedly in the resolution of algorithmic problems, and if learning objects have been used in order to diminished them, a systematic literature review was carried out to summarize studies on solving problem models and the use of learning objects as a techno-pedagogical strategy. Initially, a search was made in databases such as the Education Resources Information Center (ERIC) and the Institute of Electrical and Electronics Engineers (IEEE), and 35 studies published in the last 10 years were selected. In a second phase, 26 were the final studies that met the defined quality criteria. The review revealed the need to develop the capacity for abstraction and analysis to identify and understand the problem in students, as well as the interest in defining and manipulating strategies that encourage algorithmic thinking as problem-solving skills, distinguishing between learning objects as an alternative for their development.

Keywords: algorithms, higher education, systematic literature mapping, learning objects.

Resumo

O processo algorítmico de resolução de problemas foi identificado como conflituoso entre estudantes universitários na área de ciência da computação e ciência da computação. Tentar entender quais são as dificuldades que aparecem repetidamente na resolução desses problemas, bem como se os objetos de aprendizagem foram utilizados como estratégia técnico-pedagógica para reduzi-los e, se assim for, identificar quais, um mapeamento sistemático de literatura especializada sobre o assunto. Com base nas questões de pesquisa definidas, inicialmente foi realizada uma pesquisa em bancos de dados como o Centro de Informações sobre Recursos Educacionais (ERIC) e o Instituto de Engenheiros Elétricos e Eletrônicos (IEEE), e foram selecionados 35 estudos publicados nos últimos 10 anos; Numa segunda fase, seguindo os critérios de qualidade definidos, o número foi reduzido para 26. A revisão mostrou a necessidade de desenvolver nos alunos a capacidade de abstração e análise para identificar e entender o problema. Também foi observado o interesse na definição e manipulação de estratégias que

promovam o pensamento algorítmico como habilidade de resolução de problemas, distinguindo os objetos de aprendizagem como uma alternativa para o seu desenvolvimento.

Palavras-chave: algoritmos, ensino superior, mapeamento sistemático da literatura, objetos de aprendizagem.

Fecha Recepción: Octubre 2019

Fecha Aceptación: Marzo 2020

Introduction

Computational problems are one of the main objects of study in computer science. The biggest obstacle that students face is the lack of capacity and flexibility in problem solving (Babori, Fihri, Hariri and Bideq, 2016; Luza, 2017). That is, the ability to formulate problems, think creatively about solutions, and express a solution clearly and precisely.

The abstract level of the topic and the lack of connection with practical and significant problems demonstrate the need to achieve greater motivation and stimulate the development of problem-solving skills (Chezzi et al., 2017).

The way in which computing is involved in various aspects of modern life brings advantages to new generations of students, driving the need for introductory programming courses focused on the algorithmic problem solving approach that introduce critical thinking , logical reasoning and problem solving skills in students as essential skills to learn to program.

In the last decade, great interest has been seen in the transition from a classroom-based learning environment in classrooms to an efficient integrated and technological system through techno-pedagogical strategies that facilitate the teaching-learning process.

Chun (2004, cited in Raspopovic, Cvetanovic and Jankulovic, 2016) points out that one way to achieve this is through the segmentation of teaching materials into small units, with the aim of assembling lessons based on existing materials. Learning objects are considered tools that improve the efficiency, performance and the student experience (AlMegren and Yassin, 2013).

The objective of this research is to show the result of a systematic literature mapping (SLM) on algorithmic problem solving and techno-pedagogical strategies to facilitate their development. It should be noted that, as a guide for mapping, the guidelines suggested by Petersen, Vakkalanka y Kuzniarz (2015).



Theoretical foundation

One of the pillars of computer science is the algorithm. The word algorithm comes from Al-Jwarizmi, nickname of the famous mathematician Mohammed Ibn Musa al-Jwarizmi (780-850) (Islamic Culture Foundation [Funci], August 7, 2014). Since the first translation of his books in the 9th century by Adelardo de Bath and Roberto de Chester, the term has been deforming until it reached its current form. The Royal Spanish Academy [RAE] (2018) defines it as a finite group of operations organized in a logical and orderly manner that allows solving a certain problem.

Other important concepts to define are algorithmic thinking, computational thinking and logical thinking, skills needed in solving computational problems.

Algorithmic thinking is the ability to perform the process of abstraction, modeling of a problem, logical deductions and synthesis of the solution that leads to writing the correct algorithm (Chair of Algorithm and Program Concepts. UNLP, 2016). For its part, Lamagna (2015) defines it as the ability to execute, evaluate, understand and create computational procedures.

Computational thinking is based on the power and limits of computer processes, whether they are executed by a human or by a machine; It involves the use of abstraction and segmentation when it comes to solving a complex task or designing a complicated system, using heuristic reasoning to discover a solution (Wing, 2006). Another definition is that offered by Aho (2012), who characterizes it as the thought process involved in the formulation of problems so that their solutions can be represented as discrete computational steps and algorithms.

Logical thinking is an important fundamental skill. Albrecht (1984) says that the basis of all logical thinking is sequential thinking. This process involves taking the important ideas, facts and conclusions involved in a problem and organizing them in a chain-shaped progression that assumes meaning in itself. Logically thinking is thinking in steps.

We cannot forget that one way to solve problems is to resort to heuristics. Anderson (1990, cited in Schunk, 2012) indicates that this “consists of general methods to solve problems that use principles (general rules) that usually lead to the solution” (p. 302).

Table 1 describes some heuristic models for problem solving.

Tabla 1. Modelos heurísticos de resolución de problemas

Autor*	Modelo heurístico	Cómo funciona
Polya (1945, 1957)	<ul style="list-style-type: none"> • Entender el problema. • Diseñar un plan. • Ponerlo en práctica. • Retroceder. 	<p>Entender el problema requiere plantear preguntas como “¿cuál es la incógnita?” y “¿cuáles son los datos?”.</p> <p>Al diseñar un plan, el aprendiz trata de encontrar una conexión entre los datos y la incógnita.</p> <p>Al ponerlo en práctica, es importante verificar cada paso para asegurarse de que se está aplicando en forma correcta. Retroceder significa examinar la solución: ¿es correcta? o ¿existen otros medios para llegar a ella?</p>
Bransford y Stein (1984)	<p>Ideal</p> <ul style="list-style-type: none"> • Identificar el problema. • Definir y plantear el problema. • Explorar posibles estrategias. • Aplicar las estrategias. • Lograr ver hacia atrás para evaluar los efectos de las actividades. 	<p>Darse cuenta de que existe el problema y se le puede dar solución, especificando las condiciones iniciales que debe tener; posteriormente se buscan posibles estrategias de solución; una vez que se seleccionan se aplican y se evalúan.</p>
Treffinger (1985) y Treffinger e Isaksen (2005)	<p>Solución creativa de problemas (SCP)</p> <ul style="list-style-type: none"> • Entender el problema. • Generar ideas. • Prepararse para la acción. 	<p>La comprensión del problema inicia con una meta general para la solución de problemas. Se deben obtener datos importantes como: hechos, opiniones y preocupaciones para formular una pregunta específica. La clave para generar ideas es el pensamiento divergente, el cual es capaz de producir diversas opciones que conduzcan a lograr la meta planteada. Prepararse para la acción incluye examinar las opciones más promisorias y buscar fuentes de ayuda o maneras para superar la resistencia.</p>

Newell y Simon (1972)	Modelo de procesamiento de información que consta de un espacio de problema <ul style="list-style-type: none"> • Estado inicial • Estado final • Vías de solución posibles 	Quien resuelve el problema se forma una representación de este y realiza operaciones para reducir la discrepancia entre el estado inicial y el final.
*Cada uno citado en Schunk (2012)		

Fuente: Elaboración propia con base en Schunk (2012)

Mac Gaul de Jorge, López and Del Olmo (2008) agree with Polya (1945, 1957, cited in in Schunk, 2012) that computational problems are analogous to mathematical problems under the category "Problems to be solved".

Table 2 shows two methods adapted from the Polya method (1945, 1957, cited in Schunk, 2012).

Tabla 2. Métodos para la resolución de problemas computacionales

Autor	Fases
Thompson (1996)	<ul style="list-style-type: none"> • Entender el problema. • Diseñar el problema. • Escribir el programa. • Evaluar la solución.
Mac Gaul de Jorge <i>et al.</i> (2008)	Comprender el problema implica: <ul style="list-style-type: none"> • Reconocer términos. • Identificar entradas, salidas, condiciones. • Diseñar casos prueba. • Seleccionar componentes (concebir un plan conectando con los conocimientos previos adquiridos, identificando los conceptos necesarios para resolver el problema). • Diseñar el algoritmo. • Prueba de escritorio.

Fuente: Elaboración propia

Systematic mapping on resolution of algorithmic problems

The objective of the SLM is to summarize the existing evidence regarding how algorithmic problem solving has been addressed, and some techno-pedagogical strategies used to facilitate it.

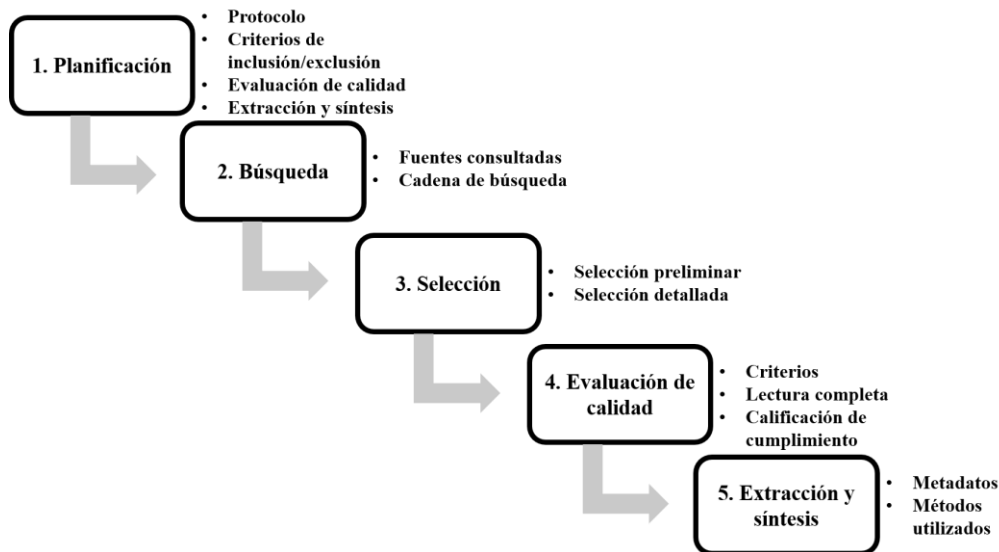
The resources to use are:

- Electronic journals and books available in the library of the Universidad Veracruzana or in databases such as Education Resources Information Center (ERIC) and the Institute of Electrical and Electronics Engineers (IEEE).
- Digital libraries, including that of the Universidad Veracruzana.
- Research and thesis related to the subject here in question.
- Internet, through the use of search engines such as Google Scholar and SemanticScholar.

Method

The review method used in this review is summarized in Figure 1.

Figura 1. Esquema del método de revisión empleado



Fuente: Revelo, Collazos y Jiménez (2018)

Planning

The planning is based on a protocol to perform the SLM that allows defining the scope of the review and the activities to achieve the objectives. In our case, the following questions of interest were specified:



- RQ1. In which magazines and at which events are articles on solving computer problems published?
- RQ2. What research and development (R&D) groups are currently working on solving computational problems?
- RQ3. What topics are addressed in solving computational problems?
- RQ4. What are the most investigated topics?
- RQ5. Have learning objects been used as a teaching strategy in solving computational problems?

Search

For the search phase, the population / intervention / comparison / results search strategy (PICO) was used, which was developed to identify keywords and formulate search strings based on research questions.

In our SLM it was defined as follows:

- Population: studies on algorithmic problem solving.
- Intervention: the techno-pedagogical tools used to favor the resolution of computational problems.
- Comparison: this study aims to analyze the skills that students must have for problem solving and the strategies used to promote it.

Talking about problem solving means taking into account algorithmic thinking, computational thinking, logical thinking, and mathematical thinking. Techno-pedagogical strategies, such as learning objects, were also considered to facilitate problem solving. Therefore, the identified keywords were as follows: solving computer problems, learning objects, computational thinking and algorithmic thinking.

The search strings used were: solving computer problems, learning objects in programming, techno-pedagogical strategies in solving computer problems, learning objects like programming strategy, learning objects like solving computer problems strategy, development of algorithmic thinking with the support of learning objects and the use of learning objects for solving computer problems.

The search was carried out in Spanish and English in the ERIC and IEEE databases; later via the Internet and the Google Scholar and Semantic Scholar browser.

The articles were excluded based on the criteria set out below:



- Title and summary in which no explicit mention was made of algorithmic problem solving.
- Full text reading; His quick review focused on identifying whether the elements of interest were mentioned in this investigation.
- Quality evaluation according to the relevance of the content to answer the research questions.
- Other criteria taken into account for the exclusion of an article were:
 - The year of publication, with a seniority greater than 10 years.
 - Studies published in languages other than English and Spanish.
 - Studies not accessible in full text.

Selection

The first phase consisted of the preliminary selection of the articles based on the revision of the title of the document, as well as the abstract, with the objective of selecting those that made explicit reference to the resolution of algorithmic problems; Subsequently, those articles mentioning algorithmic skills, algorithmic or computational thinking and logical thinking were chosen. Documents containing strategies for problem solving or for developing programming skills, as well as the use of learning objects were also included. In this first selection, 35 bibliographic records were obtained.

In a second phase, an assessment was made based on the exploration of the full text, placing special interest in points of interest regarding the resolution of computational problems. In this phase the final figure of 26 documents was reached.

Table 3 lists the number of items per year. Although preferably the selection of articles from five years to date is suggested, in this review it was considered since 2009 as being within the 10-year limit.

Tabla 3. Documentos seleccionados

Año	Número
2009	1
2010	1
2011	1
2012	1
2013	6
2014	1
2015	6
2016	1
2017	4
2018	4

Fuente: Elaboración propia

Quality assessment

The quality criteria considered when evaluating the selected articles were: place of publication (database, journal), the type of publication, that is, thesis, article or conference, the importance of the content to answer the research questions and the methodological design of research and data analysis.

Table 4 presents a quantitative summary of the selected publication types.

Tabla 4. Resumen cuantitativo de tipos de publicación

Tipo de publicación	Número
Disertación	3
Artículo	13
Conferencia	10

Fuente: Elaboración propia

Data extraction and synthesis

At this stage, a data extraction was performed in a spreadsheet in order to answer the research questions.

Regarding question RQ1. In which magazines and in which events are articles on computer problem solving published? The magazines listed in Table 5 were obtained.

Tabla 5. Revistas científicas

Revista	Descripción
<i>Algorithms</i>	Es una revista de acceso abierto revisado por pares que proporciona un foro avanzado para estudios relacionados con algoritmos y sus aplicaciones. ISSN 1999-4893. Con un factor de impacto de 0.28 (Scientific Journal Rankings [SRJ], 2018), Q4.
<i>Contemporary Issues in Education Research Journal</i>	Acoge artículos de una amplia gama de temas contemporáneos en la educación. Esta revista publica artículos que contribuyen significativamente a la comprensión de temas de vanguardia en educación. ISSN 1940-5847 (impreso) y 1941-756X (en línea). Tiene una calificación de impacto, de acuerdo con Index Copernicus International (ICI) Journal Master List (2015), de 78.11.
<i>Education and Information Technologies</i>	Revista oficial del Comité Técnico de Educación de la Federación Internacional de Procesamiento de la Información (IFIP, por sus siglas en inglés). Publica artículos de todos los sectores de la educación sobre todos los aspectos de la tecnología de la información y los sistemas de información. ISSN 1360-2357 (impreso) y 1573-7608 (en línea). Con un factor de impacto de 0.60 (SRJ, 2018), Q1.
<i>Informatics in Education</i>	<i>Journal</i> de Europa oriental y central publicado por el Instituto Universitario de Vilnius de Ciencia de Datos y Tecnologías Digitales y Academia de Ciencias de Lituania. ISSN: 1648-5831 (impreso) y 2335-8971 (en línea). Con un factor de impacto de 0.29 (SRJ, 2018), Q3.
<i>Interdisciplinary Journal of Knowledge and Learning Objects</i>	Actualmente se le conoce como <i>Interdisciplinary Journal of E-Learning and Learning Objects</i> . Publica artículos de alta calidad sobre teoría, práctica, innovación e investigación que cubren todos los aspectos de <i>e-learning</i> y <i>learning objects</i> . ISSN 1552-2237. Está indexada por dblp computer science bibliography, LearnTechLib, UlrichsWeb, Ebsco, Google Scholar, Microsoft Academic Search.
<i>Journal of College Teaching & Learning</i>	Publica artículos que contribuyen significativamente a la comprensión de todos los aspectos de la enseñanza y los temas de aprendizaje y campos relacionados. ISSN 1544-0389 (impreso) y 2157-894X (en línea). Indexada por ABI Inform, Directory of Research Journals Indexing, ERIC, Google Scholar, J-Gate, ProQuest, Ulrich's Periodicals. La ICI Journal Master List (2015) le da una calificación de impacto de 78.11.
<i>Journal of Education and Learning</i>	Es una revista internacional de acceso abierto, multidisciplinaria y arbitrada por pares que se ha establecido para la difusión de conocimientos de vanguardia: en el campo de la educación, enseñanza, desarrollo, instrucción, proyectos e innovaciones educativas, metodologías de aprendizaje y nuevas tecnologías en educación y aprendizaje. ISSN 2089-9823 (impreso) y 2302-9277 (impreso).

	Indexada en ERIC. Esta revista está ahora acreditada (reconocida) por el Ministerio de Investigación, Tecnología y Educación Superior, República de Indonesia.
<i>Olympiads in Informatics</i>	Es una revista académica arbitrada que proporciona un foro internacional para presentar investigaciones y desarrollos en el ámbito específico de la enseñanza y el aprendizaje de la informática a través de olimpiadas y otras competencias. ISSN 1822-7732 (impreso) y 2335-8955 (en línea). Con un factor de impacto de 0.22 (SJR, 2018).
<i>Pedagogía Universitaria</i>	Difunde los resultados científico-metodológicos e investigativos alcanzados por los profesores e investigadores universitarios en su quehacer pedagógico. ISSN 1609-4808. Certificada por el Ministerio de Ciencia, Tecnología y Medio Ambiente de Cuba (Citma) e indexada en Latindex.
<i>Píxel-Bit. Revista de Medios y Educación</i>	Es una publicación interdisciplinar de carácter científico-académico y divulgativo que pretende fomentar el intercambio de ideas y trabajos en el campo de los medios audiovisuales, informática y tecnologías avanzadas aplicadas al terreno educativo y de formación en general. ISSN 1133-8482. Con un índice de impacto de 1.170 en 2018, C1, de acuerdo con las métricas de Dialnet.
<i>Procedia Computer Science</i>	Revista indexada en Scopus, cubre los temas del área de ciencias de la computación. Es una plataforma altamente reconocida para acceder a publicaciones de actas de conferencias de alta calidad para los investigadores. ISSN 1877-0509. Factor de impacto 0.28 (SJR, 2018).
<i>Revista Electrónica de Investigación Educativa (REDIE)</i>	La <i>REDIE Revista Electrónica de Investigación Educativa</i> es una publicación del Instituto de Investigación y Desarrollo Educativo de la Universidad Autónoma de Baja California que utiliza las nuevas tecnologías de la información y la comunicación, y tiene como objetivo principal difundir la investigación que se realiza a nivel regional, nacional e internacional en el campo de la educación, aprovechando las ventajas que ofrece el medio electrónico al quehacer académico. ISSN 1604-4041. Factor de impacto 0.43 (SJR, 2018), Q2.
<i>TOJET: The Turkish Online Journal of Educational Technology</i>	Revista científica turca internacional en el campo de la tecnología educativa. <i>TOJET</i> es una revista en línea y revisada por pares que acepta artículos sobre todos los aspectos de la tecnología educativa. ISSN 2146-7242. Ha estado en el índice ERIC desde el volumen 1, número 1, 2002.
<i>Uporabna Informatika</i>	Revista eslovena que promueve el desarrollo y el intercambio de conocimientos de una amplia gama de temas, como la informática empresarial, la administración electrónica y las soluciones informáticas para uso personal. ISSN 1318-1882 (impreso) y 2630-435X (en línea).
<i>WSEAS Transactions on Computers</i>	<i>WSEAS Transactions on Computers</i> publica trabajos de investigación originales relacionados con la ciencia de la información. ISSN 1109-

2750 (impreso) y 2224-2872 (en línea). Con un factor de impacto de 0.12 (SJR, 2017), Q4 y 0 (SJR, 2018).
--

Fuente: Elaboración propia

The selected articles included a) one presented at the XIV International Conference on Cognitive and Exploratory Learning in the Digital Age belonging to the Community Information Service on Research and Development (Cordis) of the European Commission; b) four articles of the Congress on Technology in Education and Technology Education of the National University of La Plata in Argentina, and c) one published in the proceedings of the Innovation and Technology in Computer Science Education (ITiCSE) of 2009.

As for question RQ2. What R&D groups are currently working on algorithmic problem solving? The analysis of the selected articles allowed identifying researchers in the area of algorithmic problem solving. Their professional career and interest in the difficulties encountered by students in the process of solving algorithmic problems and the development of strategies that favor the competition of algorithmic thinking were taken into account.

Marcia Mac Gaul de Jorge, research professor at the University of Salta, Argentina, works mainly in the basic programming subjects of the Systems Analysis degree; He directs, together with his colleagues, research and development projects accredited by the Research Council of Oriented Scientific and Technological Research Projects (Picto) and the National Agency for Scientific and Technological Promotion (Anpcyt). She is the first author of four publications between 2008 and 2015 on problem solving, cognitive processes, algorithmic thinking, logic and the development of a development tool to motivate programming learning in a playful environment.

Eva Milková, professor in the IT department of the University of Hradec Králové, of the Czech Republic; He conducted research on the development of algorithmic thinking, as well as tools to favor it. Between 2005 and 2017 he published articles on the development of algorithmic thinking, including the use of multimedia applications and practical and entertaining tasks to promote it.

Benjamin A. Burton, Bachelor of Science and PhD from the University of Melbourne, he was Director of Training for the Australian Olympic Computer Program from 1999 to 2008, and is now a member of the Scientific Committee of the International Computer Olympiad (IOI), World's leading competition for high school students in computer programming and algorithm design. He has been involved in the Australian Computing Competition (AIC) since its inception in 2005, and edited the AIC documents for the first three years. He is currently a Research Fellow of the QEII Center at the University of Queensland, where he works in computational geometry and topology in three and four dimensions.



Regarding question RQ3. What topics are addressed in solving computational problems? Table 6 lists the authors, topic and type of study.

Tabla 6. Lista de autores, tópicos y tipo de estudio identificados

Autores	Tópico	Tipo de estudio
Adamchik y Gunawardena (2013)	El objetivo de esta investigación es describir un nuevo enfoque para un mecanismo de creación y entrega de contenido para un curso de programación. Este enfoque se basa en el concepto de crear un gran repositorio de objetos de aprendizaje, cada uno de los cuales consta del material principal, ejemplos de código, notas complementarias y preguntas de revisión.	No se indica
Begosso, Begosso, Ribeiro, dos Santos y Begosso (2015)	Desarrollo de un proyecto para crear varios objetos de aprendizaje para ayudar a enseñar conceptos a los estudiantes de los cursos de ciencias que consideran difícil la enseñanza de algoritmos y programación.	Mixto
Biro y Csernoch (2016)	Los autores proporcionan detalles sobre cómo se puede medir la capacidad matemática de las herramientas informáticas basadas en su propia tipología de resolución de problemas.	No se indica
Burton (2010)	Describe la competencia informática australiana a través de cuatro categorías: 1) Áreas algorítmicas, que animan a los estudiantes a desarrollar un algoritmo informal para resolver un rompecabezas determinado, 2) Tareas lógicas, que utilizan rompecabezas no algorítmicos para fomentar un razonamiento riguroso y análisis de casos, 3) Tareas de rastreo, que son tareas simples que les piden a los estudiantes que sigan un conjunto de instrucciones bien definido y 4) Tareas de análisis, donde los alumnos sondean las fortalezas y debilidades de un algoritmo o problema dado.	No se indica

Csernoch, Biró, Máth y Abari (2015)	El Proyecto de Pruebas de Habilidades Algorítmicas y de Aplicación (TAaAS) se lanzó en el año académico 2011/2012 para evaluar a los estudiantes de informática de primer año, centrándose en sus habilidades algorítmicas, en entornos de programación tradicional y no tradicional, y en la transferencia de sus conocimientos de informática.	No se indica
de Raadt (2018)	Tesis doctoral de tipo experimental que describe la integración explícita de estrategias de programación en la instrucción y evaluación de programadores principiantes, y el impacto de este cambio en sus resultados de aprendizaje.	Experimental
Djenic y Mitic (2017)	Este par de autores presenta estrategias y métodos de enseñanza aplicables en entornos mixtos modernos para el aprendizaje de programación dentro de la programación mixta tradicional y en los cursos a distancia de la Escuela de Ingeniería Eléctrica e Informática de Estudios Aplicados en Belgrado.	No se indica
Guerrero y García (2016)	Investigación de tipo cuantitativa con un diseño cuasi-experimental en la que utilizaron objetos de aprendizaje generativos (GLO) para el desarrollo del pensamiento algorítmico en un curso de fundamentos de programación para estudiantes de nuevo ingreso de la ingeniería en Sistemas Computacionales del Instituto Tecnológico de San Luis Potosí.	Cuantitativa
Hu, Tseng y Lee (2013)	Investigación de tipo experimental que utiliza el enfoque de andamios para ayudar a los estudiantes a comprender la estrategia de	Experimental

	resolución de problemas para encontrar la estructura de datos adecuada.	
Konecki (2015)	La adopción de una manera de pensar apropiada que puede llamarse <i>pensamiento algorítmico</i> para la resolución de problemas computacionales.	Cuantitativa
Mac Gaul de Jorge, Fernández, Vargas y Del Olmo (2015)	Estudian los obstáculos y problemáticas en las construcciones conceptuales del diseño específico en el diseño del algoritmo.	Estudios de caso
Mac Gaul, Massé Palermo, Sarmiento (2013)	Presentan un entorno de programación innovador en 3D para motivar al estudiante a aprender programación.	No se indica
Mac Gaul de Jorge y López (2011)	Aluden a una aplicación desarrollada para la construcción de algoritmos aplicados a un caso práctico: la conversión de números enteros, expresados en distintos sistemas numéricos, bajo un enfoque algorítmico que pretende potenciar la competencia de abstracción y la reutilización y adecuación de componentes en el diseño del algoritmo.	No se indica
Malik y Coldwell-Neilson (2016)	Tratan tres entidades del triángulo didáctico, instructores, estudiantes y plan de estudios, para explorar las dificultades de aprendizaje que encuentran los estudiantes cuando estudian introducción a la programación. El modelo enfoque/despliegue/resultado/mejora (ADRI) como parte de la pedagogía utilizada en el curso de introducción a la programación es propuesto para abordar los problemas identificados en la enseñanza de dichos cursos.	No se indica
Matthews (2014)	Investigación sobre los efectos de objetos (LO) de aprendizaje de diferentes tamaños en aprendizaje de programación. Se llevó a cabo un experimento en estudiantes de programación novatos. Se desarrollaron dos tamaños diferentes	Experimental

	de LO como una herramienta de soporte de aprendizaje en un entorno de laboratorio.	
Milková y Hùlková (2013)	Enfoque exitoso en la optimización de la enseñanza y el aprendizaje de una asignatura que desarrolla el pensamiento algorítmico, seguida de una discusión de los beneficios de los rompecabezas resueltos dentro de los temas que tratan con algoritmos gráficos y permiten el desarrollo del pensamiento lógico y la imaginación de los estudiantes, es decir, las habilidades necesarias para comprender más profundamente los algoritmos más complejos	No se indica
Revelo <i>et al.</i> (2018)	Revisión sistemática sobre el trabajo colaborativo en cursos de programación como una estrategia de impacto positivo.	Revisión sistemática de literatura
Román, Pérez y Jiménez (2017)	Abordan la definición de pensamiento computacional y cómo medirlo; validan un instrumento psicométrico para hacerlo.	Cuantitativa
Ruiz (2015)	Investigación descriptiva que tuvo como propósito: 1) identificar las posibilidades de las 15 plataformas de Massive Open Online Course (MOOC) que son más adecuadas para diseñar e implementar cursos de habilidades de programación básica basados en el marco de la comunidad de investigación (CoI), y 2) describir y comparar cómo se implementaron las estrategias de instrucción basadas en el marco de CoI.	Descriptivo
Salgado, Alonso, Sánchez y Tardo (2013)	Análisis epistemológico de la resolución de problemas con el objetivo de revelar nuevas relaciones didácticas que sustentan la lógica algorítmico-computacional de la resolución de problemas de programación.	Análisis epistemológico

Salleh, Shukur y Judi (2013)	Estudio sobre herramientas de enseñanza enfocados en la etapa introductoria de programación; en las técnicas y métodos utilizados en el aprendizaje y evaluación.	No se indica
Sánchez (2018)	Tesis doctoral bajo un enfoque cualitativo que propone una metodología de desarrollo de sistema tutor (ST) enfocada a las habilidades algorítmicas, tomando como base el conocimiento previo del estudiante, acompañándolo en su progreso a través de estrategias de aprendizaje lúdicas que le motiven y ayuden a desarrollar sus habilidades de pensamiento	Cualitativo
Sanford y Naidu (2017)	El modelado matemático como esencia del pensamiento computacional.	No se indica
Tuparov, Tuparova y Tsarnakova (2012).	Desarrollo y uso de objetos de aprendizaje basados en simulación interactiva en un curso introductorio de programación para facilitar la transición del pensamiento concreto al abstracto y la comprensión de los conceptos básicos de algoritmos y programación.	No se indica
Villalobos, Calderón y Jiménez (2009)	Objetos de aprendizaje interactivos (ILOs) como uno de los componentes que refuerzan un modelo pedagógico, al apoyar la generación de habilidades de programación de alto nivel. En este documento se sugiere una taxonomía multidimensional para los ILOs, presentando la experimentación desarrollada para evaluar el impacto de estos objetos dentro de nuestros cursos de ciencias de la computación	No se indica

Zatarain (2018)	En este estudio se presenta un ambiente de aprendizaje que usa técnicas de reconocimiento afectivo con gamificación. El método consistió en evaluar el aprendizaje de la lógica algorítmica usando técnicas tradicionales contra el aprendizaje con reconocimiento automático de emociones y manejo motivacional usando gamificación. Los resultados demostraron que el aprendizaje del estudiante es estadísticamente mejor si se toma en cuenta el estado afectivo del estudiante y si este es motivado por medio de la gamificación.	Cuantitativa, con un diseño cuasi experimental
-----------------	---	--

Fuente: Elaboración propia

In relation to question RQ4. What are the most investigated topics? The following topics were the most recurring:

- Adopting a way of thinking creatively called algorithmic thinking for solving computational problems.
- Strategies for learning programming.
- Strategies for the development of algorithmic thinking.
- How to measure algorithmic thinking

Finally, regarding the questioning RQ5. Have learning objects been used as a didactic strategy in solving computational problems? Through the SLM carried out, it can be seen that learning objects have been used to promote programming learning for more than a decade and recently to develop algorithmic and computational thinking.

Discussion

The main objective of the initial programming courses is the design of algorithms. Students should understand how one works and, after that, begin to describe it in some programming language.

However, these courses have experienced the highest failure rates in the computing area, as evidenced by Begosso et al. (2015) Mac Gaul de Jorge et al. (2015), Tuparov et al. (2012), among others. It is for this reason that various investigations have attempted to adopt new pedagogical

tools and approaches, with the aim of supporting student learning in a more pleasant and efficient way. An instructional tool that has been used in recent years has been learning objects.

Learning objects have been used to promote understanding of abstract programming concepts in both universities and high schools; They have especially served as support in the learning of students who begin their studies in programming, as Matthews (2014) states. Likewise, Guerrero and García (2016) pointed out the success of using learning objects to develop algorithmic thinking; This pair of authors highlighted that the visual and easy-to-navigate characteristics promote the use of learning objects among students, and motivate the student to develop a greater commitment to their training.

The response obtained during the use of learning objects by students was positive in various investigations, as stated Guerrero y García (2016) and Tuparov *et al.* (2012).

Conclusions

The software development area has undergone a rapid expansion and this trend does not stop. Students must be able to create various algorithms to solve specific problems, starting with the simplest, consecutively increasing their algorithmic knowledge until they reach the level at which they understand much more complex groups of operations.

Teaching algorithms and programming concepts for first year students has been a great challenge for universities. New computer science students generally have difficulty understanding and abstracting the understanding of the problem to be solved. One of the great difficulties of learning during the initial semesters is the development of algorithmic thinking, which increases the level of student failure and also the dropout rate of these courses.

As a recap, the SLM began with the analysis of 35 articles; a more critical analysis was continued, where the exclusion criteria defined above were taken into account; consequently, nine articles were discarded. From the remaining 26 articles, the research questions were answered. After all this, it can be concluded that solving computational problems represents a difficulty for students at an introductory stage in programming. It is to highlight the need for the capacity for abstraction and analysis to identify and understand the problem. And it is also observed that there is interest in the definition and manipulation of strategies that encourage algorithmic thinking as a problem solving ability. Finally, learning objects are distinguished as an alternative to improve the teaching-learning process in programming courses and in the development of algorithmic thinking.

The mapping constitutes a strategy of search and selection of literature. In that sense, a first limitation of this study was the preliminary selection of the articles, based on the review of the title of the document and the summary of the investigation, which did not allow to deepen the research findings; A second limitation was not identifying the type of study in the abstract: some selected articles did not include it.

On the other hand, the SLM constitutes a basis for a systematic review of the literature, which in future research will allow to answer specific research questions related to the results of empirical studies.

Problem solving is a recurring aspect of programming that requires the ability to integrate and apply a number of fundamental concepts, cognitive skills, and thinking styles. Developing skills to solve problems in a methodological way is crucial for the development of algorithms as a previous step to the creation of computer programs, so, as another future line of research, the development of a technopedagogical strategy with the objective of encourage algorithmic problem solving competence.

References

- Adamchik, V. and Gunawardena, A. (2013). A learning objects approach to teaching programming. In *Proceedings ITCC 2003. International Conference on Information Technology: Coding and Computing* (pp. 96-99). Las Vegas, United States. Retrieved from <https://ieeexplore.ieee.org/document/1197507>.
- Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835.
- Albrecht, K. (1984). *Brain Building: Easy Games to Develop Your Problem-Solving Skills*. New Jersey, United States: Prentice Hall.
- AlMegren, A. and Yassin, S. Z. (2013). Learning Object Repositories in e-Learning: Challenges for Learners in Saudi Arabia. *European Journal of Open, Distance and E-learning*, 16(1), 115-130.
- Babori, A., Fihri, H., Hariri, A. and Bideq, M. (2016). An e-Learning environment for algorithmic: Toward an active construction of skills. *World Journal on Educational Technology: Current Issues*, 8(2), 82-90.
- Begosso, L. C., Begosso, L. R., Ribeiro, A., dos Santos, R. M. and Begosso, R. H. (2015). The use of Learning Objects for teaching Computer Programming. In *2015 IEEE Frontiers in Education Conference* (pp. 786-791). El Paso, United States: IEEE.
- Biro, P. and Csernoch, M. (2016). The mathability of computer problem solving approaches. In *2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)* (pp. 111-114). Gyor, Hungary: IEEE.
- Burton, B. A. (2010). Encouraging Algorithmic Thinking Without a Computer. *Olympiads in Informatics*, 4(1), 3-14. Retrieved from <http://www.ncss.edu.au/>.
- Cátedra de Conceptos de Algoritmos y Programa. UNLP. (2016). Por qué “pensar algoritmos” es tan importante en informática. *Revista Institucional de la Facultad de Informática | UNLP*, 2(4), 21-22. Recuperado de http://sedici.unlp.edu.ar/bitstream/handle/10915/57362/Documento_completo.pdf-PDFA.pdf?sequence=1.
- Csernoch, M., Biró, P., Máth, J. and Abari, K. (2015). Testing algorithmic skills in traditional and non-traditional programming environments. *Informatics in Education*, 14(2), 175-197.
- Chezzi, C. M., Schenberger, L., Casañas, F., Giuponi, D., Anzardi, A. y Salvarredi, M. (2017). Estrategia de motivación para el razonamiento de algoritmos computacionales mediante

- juegos. Ponencia presentada en la V Jornada de Enseñanza de la Ingeniería. San Nicolás de los Arroyos, octubre 2017. Recuperado de <http://ria.utn.edu.ar/bitstream/handle/123456789/2266/Estrategia%20de%20%20motivaci%C3%B3n%20para%20el%20razonamiento%20de%20algoritmos%20computacionales%20mediante%20juegos.pdf?sequence=1&isAllowed=y>.
- Djenic, S. and Mitic, J. (2017). Teaching Strategies and Methods in Modern Environments for Learning of Programming. Paper presented at the International Association for Development of the Information Society (IADIS) International Conference on Cognition and Exploratory Learning in Digital Age. Vilamoura, Algarve, Portugal, October 18-20, 2017. Retrieved from <https://files.eric.ed.gov/fulltext/ED579455.pdf>.
- de Raadt, M. (2018). *Teaching Programming Strategies Explicitly to Novice Programmers*. (tesis doctoral). University of Southern Queensland, Australia.
- Fundación de Cultura Islámica [Funci]. (7 de agosto de 2014). Al-Jwarizmi y los algoritmos. Recuperado de <https://funci.org/al-jwarizmi-y-los-algoritmos/>.
- Guerrero, M. y García, J. (2016). Desarrollo del pensamiento algorítmico con el apoyo de objetos de aprendizaje generativos. *Píxel-Bit. Revista de Medios y Educación*, 0(49), 163-175. Recuperado de <https://recyt.fecyt.es/index.php/pixel/article/view/61721/37731>.
- Hu, L., Tseng, S. and Lee, T. (2013). Towards Scaffolding Problem-Solving Implementation Process in Undergraduate Programming Course. In *IEEE 13th International Conference on Advanced Learning Technologies* (pp. 417-418). China: IEEE.
- Index Copernicus International (ICI) Journal Master List. (2015). Retrieved from <https://journals.indexcopernicus.com/>.
- Konecki, M. (2015). Algorithmic Thinking as a Prerequisite of Improvements in Introductory Programming Courses. *Uporabna Informatika*, 23(3), 162-169.
- Lamagna, E. A. (2015). Algorithmic thinking unplugged. *Journal of Computing Sciences in Colleges*, 30(6), 45-52.
- Luza, C. (2017). La computación y solución de problemas computacionales. *Perspectiv@S*, 13(12), 23-27. Recuperado de <http://revistas.uigv.edu.pe/index.php/perspectiva/article/view/208>.
- Mac Gaul de Jorge, M. I., Fernández, E. F., Vargas, C. y Del Olmo, A. P. (2015). Indagación de los procesos cognitivos de los estudiantes sobre contenidos básicos de la algoritmia. Ponencia presentada en el X Congreso sobre Tecnología en Educación y Educación en

Tecnología. Corrientes, 2015. Recuperado de
<http://sedici.unlp.edu.ar/handle/10915/48498>.

Mac Gaul de Jorge, M. I., Massé Palermo, M. L. y Sarmiento, N. (2013). Análisis de Alice para la enseñanza básica de la programación. Ponencia presentada en el VIII Congreso de Tecnología en Educación y Educación en Tecnología. Santiago del Estero, 27 y 28 de junio de 2013. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/27578>.

Mac Gaul de Jorge, M. I. y López, M. F. (2011). Sistemas de numeración: una metodología de enseñanza basada en el enfoque algorítmico. Ponencia presentada en el VI Congreso de Tecnología en Educación y Educación en Tecnología. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/18429>.

Mac Gaul de Jorge, M. I., López, M. F. y Del Olmo, A. P. (2008). Resolución de problemas computacionales. Análisis del proceso de aprendizaje. Ponencia presentada en el III Congreso de Tecnología en Educación y Educación en Tecnología. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/19049>.

Malik, I. S. and Coldwell-Neilson, J. (2016). A model for teaching an introductory programming course using ADRI. *Education and Information Technologies*, 22. Retrieved from <https://doi.org/10.1007/s10639-016-9474-0>.

Matthews, R. (2014). Learning object to enhance introductory programming understanding: Does the size really matter? *Turkish Online Journal of Educational Technology*, 13(1), 174-183.

Milková, E. and Hůlková, A. (2013). Algorithmic and logical thinking development: Base of programming skills. *WSEAS Transactions on Computers*, 12(2), 41-51.

Petersen, K., Vakkalanka, S. and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64, 1-18. Retrieved from 10.1016/j.infsof.2015.03.007.

Raspopovic, M., Cvetanovic, S. and Jankulovic, A. (2016). Challenges of Transitioning to e-learning System with Learning Objects Capabilities. *The International Review of Research in Open and Distributed Learning*, 17(1). Retrieved from <https://www.learntechlib.org/p/173580/>.

Real Academia Española [RAE]. (2018). *Diccionario de la lengua española*. España: Real Academia Española. Recuperado de <http://lema.rae.es/drae/?val=algoritmo>.

- Revelo, O., Collazos, C. y Jiménez, J. (2018). El trabajo colaborativo como estrategia didáctica para la enseñanza/aprendizaje de la programación: una revisión sistemática de literatura. *Revista TecnoLógicas*, 21(41), 115-134.
- Román, M., Pérez, J. C. and Jiménez, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678-691.
- Ruiz, M. A. (2015). *A case study of introductory programming with MOOCS*. (master's thesis). Purdue University, Indiana.
- Salgado, A., Alonso, I., Sánchez, G. y Tardo, Y. (2013). Didáctica de la resolución de problemas de Programación Computacional. *Pedagogía Universitaria*, 18(4), 62-75.
- Salleh, S. M., Shukur, Z. and Judi, H. M. (2013). Analysis of Research in Programming Teaching Tools: An Initial Review. *Procedia - Social and Behavioral Sciences*, 103, 127-135.
- Sánchez, G. (2018). *Metodología de desarrollo Sistema Tutor enfocada en habilidades algorítmicas*. (tesis doctoral). Universidad Benemérita Autónoma de Puebla, Puebla.
- Sanford, J. F. and Naidu, J. T. (2017). Mathematical modeling and computational thinking. *Contemporary Issues in Education Research*, 10(2), 159-168
- Schunk, D. H. (2012). *Teorías del aprendizaje. Una perspectiva educativa*. México: Pearson Educación.
- Scientific Journal Rankings [SRJ]. (2018). Scimago Journal & Country Rank Retrieved from <https://www.scimagojr.com/journalrank.php>.
- Tuparov, G., Tuparova, D. and Tsarnakova, A. (2012). Using Interactive Simulation-Based Learning Objects in Introductory Course Using of Programming. *Procedia - Social and Behavioral Sciences*, 46, 2276-2280.
- Thompson, S. (1996). How to Program It. Retrieved from http://www.ukc.ac.uk/computer_science/Html/Courses/HowToProgIt.html.
- Villalobos, J. A., Calderon, N. A. and Jiménez, C. H. (2009). Developing programming skills by using interactive learning objects. *ACM SIGCSE Bulletin*, 41(3), 151.
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35.
- Zatarain, R. (2018). Reconocimiento afectivo y gamificación aplicados al aprendizaje de lógica algorítmica y programación. *Revista Electrónica de Investigación Educativa*, 20(3), 115-125. Recuperado de <https://doi.org/10.24320/redie.2018.20.3.1636>.