

<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](mailto: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

---

## Introducción

Los problemas computacionales constituyen uno de los principales objetos de estudio de la ciencia de la computación. El mayor obstáculo al que se enfrentan los estudiantes es la falta de capacidad y flexibilidad en la resolución de problemas (Babori, Fihri, Hariri y Bideq, 2016; Luza, 2017). Es decir, la capacidad de formular problemas, pensar de forma creativa sobre las soluciones y expresar una solución de manera clara y precisa.

El nivel abstracto del tema y la falta de vinculación con problemas prácticos y significativos evidencian la necesidad de lograr una mayor motivación y estimular el desarrollo de habilidades para la solución de problemas (Chezzi *et al.*, 2017).

La manera en que la computación está inmiscuida en varios aspectos de la vida moderna provoca ventajas en las nuevas generaciones de estudiantes, lo que impulsa la necesidad de cursos de introducción a la programación centrados en el enfoque de resolución de problemas algorítmicos que introduzcan el pensamiento crítico, el razonamiento lógico y las habilidades de resolución de problemas en los estudiantes como habilidades indispensables para aprender a programar.

En la última década se ha visto un gran interés por la transición de un entorno de aprendizaje basado en clases presenciales en aulas a un sistema integrado y tecnológico eficiente a través de estrategias tecnopedagógicas que faciliten el proceso de enseñanza-aprendizaje.

Chun (2004, citado en Raspopovic, Cvetanovic y Jankulovic, 2016) señala que una manera de lograrlo es a través de la segmentación de materiales didácticos en pequeñas unidades, con el objetivo de ensamblar lecciones basadas en materiales ya existentes. Los objetos de aprendizaje se consideran herramientas que mejoran la eficacia, rendimiento y la experiencia de los estudiantes (AlMegren y Yassin, 2013).

El objetivo de la presente investigación es mostrar el resultado de un mapeo sistemático de literatura (SLM, por sus siglas en inglés) sobre la resolución de problemas algorítmicos y las estrategias tecnopedagógicas para facilitar su desarrollo. Cabe señalar que, a modo de guía para el mapeo, se utilizaron las pautas sugeridas por Petersen, Vakkalanka y Kuzniarz (2015).

## Fundamentación teórica

Uno de los pilares de la ciencia informática es el algoritmo. La palabra *algoritmo* viene de *Al-Jwarizmi*, sobrenombre del célebre matemático Mohammed Ibn Musa al-Jwarizmi (780-850) (Fundación de Cultura Islámica [Funci], 7 de agosto de 2014). Desde la primera traducción de sus libros en el siglo IX por Adelardo de Bath y Roberto de Chester se ha venido deformando el término hasta llegar a su forma actual. La Real Academia Española [RAE] (2018) lo define como un grupo finito de operaciones organizadas de manera lógica y ordenada que permite solucionar un determinado problema.

Otros conceptos importantes de definir son *pensamiento algorítmico*, *pensamiento computacional* y *pensamiento lógico*, habilidades necesarias en la resolución de problemas computacionales.

El pensamiento algorítmico es la capacidad para realizar el proceso de abstracción, modelización de un problema, deducciones lógicas y síntesis de la solución que conduzca a escribir el algoritmo correcto (Cátedra de Conceptos de Algoritmos y Programa. UNLP, 2016). Por su parte, Lamagna (2015) lo define como la capacidad de ejecutar, evaluar, entender y crear procedimientos computacionales.

El pensamiento computacional se basa en la potencia y los límites de los procesos informáticos, ya sea que se ejecuten por un humano o por una máquina; implica el uso de la abstracción y segmentación cuando se trata de resolver una tarea compleja o se diseña un sistema complicado, utilizando el razonamiento heurístico para descubrir una solución (Wing, 2006). Otra definición es la ofrecida por Aho (2012), quien lo caracteriza como el proceso de pensamiento involucrado en la formulación de problemas de tal manera que sus soluciones puedan ser representadas como pasos computacionales discretos y algoritmos.

El pensamiento lógico es una importante habilidad fundamental. Albrecht (1984) dice que la base de todo pensamiento lógico es el pensamiento secuencial. Este proceso implica tomar las ideas importantes, los hechos y las conclusiones involucradas en un problema y organizarlas en

una progresión en forma de cadena que asume un significado en sí mismo. Pensar lógicamente es pensar en pasos.

No podemos olvidar que una forma de resolver problemas consiste en recurrir a la heurística. Anderson (1990, citado en Schunk, 2012) indica que esta “consiste en métodos generales para resolver problemas que utilizan principios (reglas generales) que suelen conducir a la solución” (p. 302).

En la tabla 1 se describen algunos modelos heurísticos para la resolución de problemas.

**Tabla 1.** Modelos heurísticos de resolución de problemas

Autor*	Modelo heurístico	Cómo funciona
Polya (1945, 1957)	<ul style="list-style-type: none"> <li>• Entender el problema.</li> <li>• Diseñar un plan.</li> <li>• Ponerlo en práctica.</li> <li>• Retroceder.</li> </ul>	<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"> <li>• Identificar el problema.</li> <li>• Definir y plantear el problema.</li> <li>• Explorar posibles estrategias.</li> <li>• Aplicar las estrategias.</li> <li>• Lograr ver hacia atrás para evaluar los efectos de las actividades.</li> </ul>	<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"> <li>• Entender el problema.</li> <li>• Generar ideas.</li> <li>• Prepararse para la acción.</li> </ul>	<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</p>

		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.
Newell y Simon (1972)	<p>Modelo de procesamiento de información que consta de un espacio de problema</p> <ul style="list-style-type: none"> <li>• Estado inicial</li> <li>• Estado final</li> <li>• Vías de solución posibles</li> </ul>	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 y Del Olmo (2008) coinciden con Polya (1945, 1957, citado en Schunk, 2012) en que los problemas computacionales son análogos a los problemas matemáticos bajo la categoría “Problemas por resolver”.

En la tabla 2 se muestran dos métodos adaptados a partir del método de Polya (1945, 1957, citado en Schunk, 2012).

**Tabla 2.** Métodos para la resolución de problemas computacionales

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

Fuente: Elaboración propia

### **Mapeo sistemático sobre resolución de problemas algorítmicos**

El objetivo del SLM es resumir la evidencia existente respecto a cómo se ha abordado la resolución de problemas algorítmicos, y algunas estrategias tecnopedagógicas utilizadas para facilitarla.

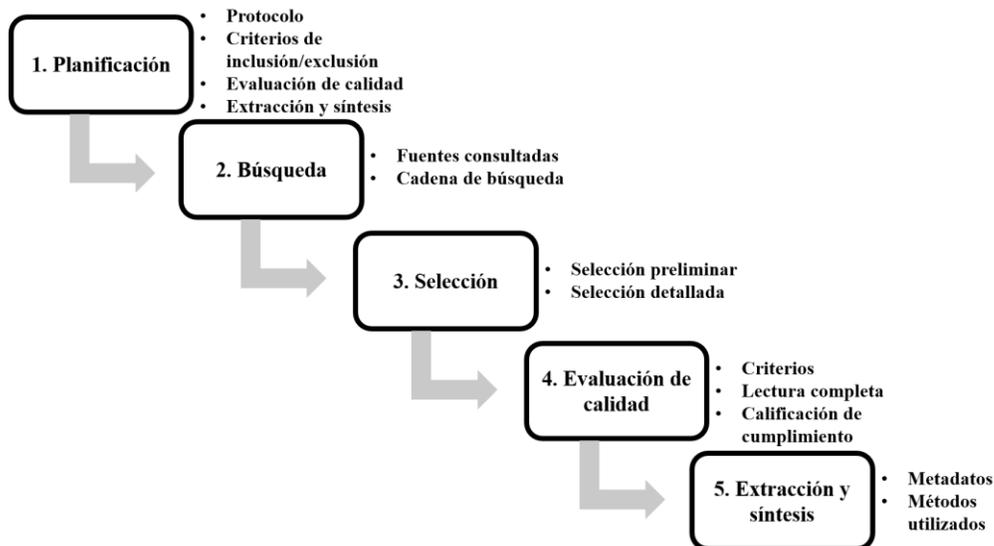
Los recursos para utilizar son:

- Revistas electrónicas y libros disponibles en la biblioteca de la Universidad Veracruzana o en bases de datos como Education Resources Information Center (ERIC) y la del Institute of Electrical and Electronics Engineers (IEEE).
- Bibliotecas digitales, entre ellas la de la Universidad Veracruzana.
- Trabajos de investigación y tesis relacionadas con el tema aquí en cuestión.
- Internet, a través del uso de motores de búsqueda como Google Scholar y SemanticScholar.

## Método

El método de revisión empleado en esta revisión se resume en la figura 1.

**Figura 1.** Esquema del método de revisión empleado



Fuente: Revelo, Collazos y Jiménez (2018)

## Planificación

La planificación se basa en un protocolo para realizar el SLM que permita definir el alcance de la revisión y las actividades para alcanzar los objetivos. En nuestro caso, se especificaron las siguientes preguntas de interés:

- RQ1. ¿En qué revistas y en qué eventos se publican artículos sobre resolución de problemas computacionales?
- RQ2. ¿Qué grupos de investigación y desarrollo (I+D) trabajan actualmente en resolución de problemas computacionales?
- RQ3. ¿Qué tópicos se abordan en resolución de problemas computacionales?
- RQ4. ¿Cuáles son los tópicos más investigados?
- RQ5. ¿Se han utilizado los objetos de aprendizaje como estrategia didáctica en la resolución de problemas computacionales?

## Búsqueda

Para la fase de búsqueda se utilizó la estrategia de búsqueda población/intervención/comparación/resultados (PICO, por sus siglas en inglés), la cual fue desarrollada para identificar palabras clave y formular cadenas de búsqueda a partir de preguntas de investigación.

En nuestro SLM se definió de la siguiente manera:

- Población: los estudios sobre resolución de problemas algorítmicos.
- Intervención: las herramientas tecnopedagógicas utilizadas para favorecer la resolución de problemas computacionales.
- Comparación: en este estudio se pretende analizar las habilidades con las que deben contar los estudiantes para la resolución de problemas y las estrategias utilizadas para favorecerla.

Hablar de resolución de problemas significa tomar en cuenta el pensamiento algorítmico, el pensamiento computacional, el lógico y el matemático. Se consideraron también las estrategias tecnopedagógicas, tal y como los de objetos de aprendizaje, para facilitar la resolución de problemas. Por lo tanto, las palabras claves identificadas fueron las siguientes: *solving computer problems*, *learning objects*, *computational thinking* y *algorithmic thinking*.

Las cadenas de búsqueda utilizadas fueron: *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* y *the use of learning objects for solving computer problems*.

La búsqueda se realizó en español y en inglés en las bases de datos ERIC e IEEE; posteriormente a través de Internet y el explorador Google Scholar y Semantic Scholar.

Los artículos se excluyeron con base en los criterios puestos a continuación:

- Título y resumen en los que no se hiciera mención explícita de la resolución de problemas algorítmicos.
- Lectura de texto completo; su revisión rápida se enfocó en identificar si se mencionaba los elementos de interés en esta investigación.
- Evaluación de la calidad de acuerdo con la relevancia del contenido para dar respuesta a las preguntas de investigación.

Otros criterios tomados en cuenta para la exclusión de un artículo fueron:

- El año de la publicación, con una antigüedad mayor a 10 años.



- Estudios publicados en idiomas diferentes al inglés y al español.
- Estudios no accesibles en texto completo.

### Selección

La primera fase consistió en la selección preliminar de los artículos a partir de la revisión del título del documento, al igual que del resumen, con el objetivo de seleccionar aquellos que hicieran referencia explícita a la resolución de problemas algorítmicos; posteriormente, se eligieron aquellos artículos que mencionan a las habilidades algorítmicas, el pensamiento algorítmico o computacional y el pensamiento lógico. Se incluyeron también los documentos que contienen estrategias para la resolución de problemas o para el desarrollo de habilidades en programación, así como el uso de objetos de aprendizaje. En esta primera selección se obtuvieron 35 registros bibliográficos.

En una segunda fase se realizó una valoración con base en la exploración del texto completo, poniendo especial interés en los puntos de interés referentes a la resolución de problemas computacionales. En esta fase se llegó a la cifra final de 26 documentos.

En la tabla 3 se enlista el número de artículos por año. Aunque preferentemente se sugiere la selección de artículos de cinco años a la fecha, en esta revisión se consideró desde el año 2009 por estar dentro del límite de 10 años.

**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



## Evaluación de la calidad

Los criterios de calidad considerados al evaluar los artículos seleccionados fueron: lugar de publicación (base de datos, revista), el tipo de publicación, es decir, tesis, artículo o conferencia, la importancia del contenido para contestar las preguntas de la investigación y el diseño metodológico de la investigación y del análisis de los datos.

En la tabla 4 se presenta un resumen cuantitativo de los tipos de publicación seleccionados.

**Tabla 4.** Resumen cuantitativo de tipos de publicación

Tipo de publicación	Número
Disertación	3
Artículo	13
Conferencia	10

Fuente: Elaboración propia

## Extracción de datos y síntesis

En esta etapa se realizó una extracción de datos en una hoja de cálculo con el objetivo de contestar las preguntas de investigación.

En cuanto a la pregunta RQ1. ¿En qué revistas y en qué eventos se publican artículos sobre resolución de problemas computacionales?, se obtuvieron las revistas listadas en la tabla 5.

**Tabla 5.** Revistas científicas

<b>Revista</b>	<b>Descripción</b>
<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 &amp; 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

De los artículos seleccionados se incluyó *a)* uno presentado en la XIV Conferencia Internacional sobre Aprendizaje Cognitivo y Exploratorio en la Era Digital perteneciente al Servicio de Información Comunitario sobre Investigación y Desarrollo (Cordis) de la Comisión Europea; *b)* cuatro artículos del Congreso sobre Tecnología en Educación y Educación en Tecnología de la Universidad Nacional de la Plata en Argentina, y *c)* uno en publicado en las actas de la Innovation and Technology in Computer Science Education (ITiCSE) de 2009.

En cuanto a la pregunta RQ2. ¿Qué grupos de I+D trabajan actualmente en resolución de problemas algorítmicos?, el análisis de los artículos seleccionados permitió identificar a investigadores en el área de resolución de problemas algorítmicos. Se tomó en cuenta su trayectoria profesional y su interés en las dificultades que encuentran los estudiantes en el proceso de resolución de problemas algorítmicos y el desarrollo de estrategias que permitan favorecer la competencia del pensamiento algorítmico.

Marcia Mac Gaul de Jorge, docente investigadora de la Universidad de Salta, Argentina, trabaja principalmente en las asignaturas básicas de programación de la licenciatura de Análisis de Sistemas; dirige junto con sus colegas proyectos de investigación y desarrollo acreditados por el consejo de investigación de Proyectos de Investigación Científica y Tecnológica Orientados (Picto) y la Agencia Nacional de Promoción Científica y Tecnológica (Anpcyt). Es la primera autora de cuatro publicaciones entre los años 2008 y 2015 sobre la resolución de problemas, los procesos cognitivos, el pensamiento algorítmico, el lógico y sobre el desarrollo de una herramienta de desarrollo para motivar el aprendizaje de programación en un ambiente lúdico.

Eva Milková, profesora del departamento de Informática de la Universidad de Hradec Králové, de la República Checa; realizó investigaciones sobre el desarrollo del pensamiento algorítmico, así como de herramientas para favorecerlo. Entre el año 2005 y 2017 publicó artículos sobre el desarrollo del pensamiento algorítmico, incluyendo el uso de aplicaciones multimedia y tareas prácticas y entretenidas para favorecerlo.

Benjamin A. Burton, licenciado en Ciencias y doctor por la Universidad de Melbourne, fue director de capacitación para el programa olímpico de informática australiano desde 1999 hasta 2008, y ahora forma parte del Comité Científico de la Olimpiada Internacional de Informática (IOI, por sus siglas en inglés), principal competencia mundial para estudiantes de secundaria en programación de computadoras y diseño de algoritmos. Ha estado involucrado en la Competencia de Informática Australiana (AIC) desde su inicio en 2005, y editó los documentos de la AIC durante los primeros tres años. Actualmente es miembro investigador del QEII Centre en la Universidad

de Queensland, donde trabaja en geometría computacional y topología en tres y cuatro dimensiones.

Respecto a la pregunta RQ3. ¿Qué tópicos se abordan en resolución de problemas computacionales?, en la tabla 6 se enumeran los autores, tópico y tipo de estudio.

**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	No se indica

	fortalezas y debilidades de un algoritmo o problema dado.	
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	Experimental

	para ayudar a los estudiantes a comprender la estrategia de 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 de LO como una herramienta de soporte de aprendizaje en un entorno de laboratorio.	Experimental
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	Análisis epistemológico

	resolución de problemas de programación.	
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

En relación con la pregunta RQ4. ¿Cuáles son los tópicos más investigados?, los siguientes temas fueron los más recurrentes:

- La adopción de una manera de pensar creativamente denominada *pensamiento algorítmico* para la resolución de problemas computacionales.
- Estrategias para el aprendizaje de programación.
- Estrategias para el desarrollo del pensamiento algorítmico.
- Cómo medir el pensamiento algorítmico

Finalmente, en cuanto al cuestionamiento RQ5. ¿Se han utilizado los objetos de aprendizaje como estrategia didáctica en la resolución de problemas computacionales?, a través del SLM realizado se observa que los objetos de aprendizaje se han utilizado para favorecer el aprendizaje de programación desde hace más de una década y recientemente para desarrollar el pensamiento algorítmico y computacional.

## Discusión

El objetivo principal de los cursos de programación inicial es el diseño de algoritmos. Los estudiantes deben entender cómo funciona uno y, después de eso, comenzar a describirlo en algún lenguaje de programación.

Sin embargo, estos cursos han experimentado las tasas de fracaso más altas el área de informática, como lo manifiestan Begosso *et al.* (2015) Mac Gaul de Jorge *et al.* (2015), Tuparov *et al.* (2012), entre otros. Es por ello por lo que diversas investigaciones han intentado la adopción de nuevas herramientas y enfoques pedagógicos, con el objetivo de apoyar el aprendizaje de los estudiantes de una manera más agradable y eficiente. Una herramienta de instrucción que ha sido utilizada en los últimos años han sido los objetos de aprendizaje.

Los objetos de aprendizaje se han utilizado para favorecer la comprensión de los conceptos abstractos de programación tanto en universidades como en escuelas secundarias; especialmente han servido de soporte en el aprendizaje de los estudiantes que inician sus estudios en programación, tal y como lo manifiesta Matthews (2014). Asimismo, Guerrero y García (2016) señalaron el éxito del uso de los objetos de aprendizaje para desarrollar el pensamiento algorítmico; este par de autores resaltó que las características visuales y de fácil navegación promueven entre los estudiantes su uso, el de los objetos de aprendizaje, y motivan al estudiante a desarrollar un mayor compromiso con su formación.

La respuesta obtenida durante el uso de objetos de aprendizaje por parte de los estudiantes fue positiva en diversas investigaciones, como lo exponen Guerrero y García (2016) y Tuparov *et al.* (2012).

## Conclusiones

El área de desarrollo de *software* ha experimentado una rápida expansión y esta tendencia no se detiene. Los estudiantes deben ser capaces de crear varios algoritmos para resolver problemas específicos, comenzando con los más sencillos, aumentando de forma consecutiva su conocimiento algorítmico hasta lograr el nivel en el que comprendan grupos de operaciones mucho más complejos.

Enseñar algoritmos y conceptos de programación para estudiantes de primer año ha sido un gran desafío para las universidades. Los nuevos estudiantes de ciencias de la computación generalmente tienen dificultades para entender y abstraer la comprensión del problema a resolver.

Una de las grandes dificultades de aprender durante los semestres iniciales es el desarrollo del pensamiento algorítmico, lo que aumenta el nivel de fracaso de los estudiantes y también la tasa de deserción de dichos cursos.

A modo de recapitulación, el SLM se comenzó con el análisis de 35 artículos; se prosiguió con un análisis más crítico, donde se tomaron en cuenta los criterios de exclusión definidos líneas arriba; consecuentemente, se descartaron nueve artículos. A partir de los 26 artículos restantes se contestaron las preguntas de investigación. Luego de todo esto se puede concluir que la resolución de problemas computacionales representa una dificultad para los estudiantes en una etapa introductoria en la programación. Es de resaltar la necesidad de la capacidad de abstracción y análisis para identificar y comprender el problema. Y también se observa que existe el interés en la definición y manipulación de estrategias que fomenten el pensamiento algorítmico como habilidad en la resolución de problemas. Por último, se distinguen los objetos de aprendizaje como una alternativa para mejorar el proceso de enseñanza-aprendizaje en cursos de programación y en el desarrollo del pensamiento algorítmico.

El mapeo constituye una estrategia de búsqueda y selección de la literatura. En ese sentido, una primera limitante de este estudio la constituyó la elección preliminar de los artículos, a partir de la revisión del título del documento y el resumen de la investigación, lo que no permitió profundizar en los hallazgos de las investigaciones; una segunda limitante fue no identificar el tipo de estudio en el resumen: algunos artículos seleccionados no lo incluían.

Por otro lado, el SLM constituye una base para realizar una revisión sistemática de la literatura, lo que en futuras investigaciones permitirá responder a preguntas de investigación específicas relacionadas con resultados de estudios empíricos.

Resolver problemas es un aspecto recurrente en programación que requiere de la capacidad de integrar y aplicar una serie de conceptos fundamentales, habilidades cognitivas y estilos de pensamiento. Desarrollar habilidades para resolver problemas de manera metodológica es crucial para el desarrollo de algoritmos como un paso anterior a la creación de programas de computadora, por lo que, como otra futura línea de investigación, se propone el desarrollo de una estrategia tecnopedagógica con el objetivo de favorecer la competencia de resolución de problemas algorítmicos.

## Referencias

- 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](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](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>.