

Estrategias y recursos didácticos utilizados para aprender programación estructurada. Una revisión sistemática

Teaching strategies and resources used to learn structured programming. A systematic review

J. Jesús Arellano Pimentel • Rocío Solar González • Luis Armería Zavala

RESUMEN

La creciente demanda de aplicaciones digitales requiere formar nuevos profesionistas con sólidos conocimientos y habilidades para el desarrollo de *software*, esta formación suele iniciar durante los primeros cursos de programación estructurada a nivel superior. En este trabajo se realizó una revisión sistemática de la literatura de tres fases (planeación, ejecución y reporte) para identificar, en los últimos cinco años, cuáles son las estrategias y los recursos didácticos más utilizados en la universidad para el aprendizaje de la programación estructurada. Se analizaron 43 fuentes, 14 en español y 29 en inglés, en su mayoría artículos indexados en Scopus, JCR y WoS, pero incluyendo también artículos de congresos, capítulos de libro y tesis de doctorado. Fue posible identificar siete estrategias didácticas entre las cuales sobresalen las orientadas a la resolución de problemas y el uso de algún diseño instruccional; en cuanto a los múltiples recursos identificados destacan el uso de sistemas gestores de aprendizaje y el apoyo en los diagramas de flujo.

Palabras clave: Estrategias didácticas, educación universitaria, recursos didácticos, perfil del estudiante, programación.

ABSTRACT

The growing demand for digital applications requires training new professionals with solid knowledge and skills for software development. This training usually begins during the first structured programming courses at a higher level. In this work, a systematic review of the literature was carried out in three phases (planning, execution, and reporting) to identify, in the last five years, which are the strategies and didactic resources most used in the university for learning structured programming. 43 sources were analyzed, 14 in Spanish and 29 in English, mostly articles indexed in Scopus, JCR and WoS, but also including conference articles, book chapters and doctoral thesis. It was possible to identify seven didactic strategies among which those oriented to problem solving and the use of some instructional design stand out; in terms of the multiple resources identified, the use of learning management systems and the support in flowcharts stand out.

Keywords: Didactic strategies, University education, didactic resources, student profile, programming.

INTRODUCCIÓN

Son numerosos los estudios realizados alrededor del mundo que dan cuenta de lo difícil que resulta a los estudiantes un primer curso introductorio a la lógica de programación estructurada, y esto ocurre prácticamente en la mayoría de las instituciones de educación superior, ocasionando altos índices de reprobación y bajo rendimiento académico de los estudiantes (Çakiroğlu et al., 2018; Juárez-Ramírez et al., 2018; Hartati, 2018; Pérez y Pedroza, 2018; Viola y Gómez, 2019; Jiménez et al., 2019; Molina et al., 2019; Solarte y Machuca, 2019). Para afrontar este desafiante reto de enseñar la lógica de programación estructurada a nivel superior se han planteado diversas estrategias y utilizado recursos didácticos, pero, ¿cuáles son concretamente estas estrategias y recursos didácticos empleados y reportados en la literatura durante los últimos cinco años? Esta es de forma general la interrogante por responder en el presente trabajo.

Un primer curso de programación estructurada está orientado a gestar las bases de la lógica de programación considerando esencialmente el uso de tres tipos de sentencias de control: secuenciales, condicionales e iterativas. No obstante, estas estructuras de control no son exclusivas del paradigma de programación estructurada, pues también están presentes prácticamente en cualquier paradigma de programación moderno. Lo anterior es una de las razones por las cuales se considera que el paradigma de programación estructurada debe ser el primero en enseñarse a nivel universitario (Vasilopoulos y van Schaik, 2018), más aún si la prioridad es el desarrollo de *software* industrial (Kandemir et al., 2020). Por lo tanto, determinar cuáles son las estrategias y

J. Jesús Arellano Pimentel. Profesor-Investigador de la Universidad del Istmo, campus Tehuantepec, México. Se encuentra adscrito a la carrera de Ingeniería en Computación y es Doctor en Educación con Tecnologías del Aprendizaje y el Conocimiento por la Universidad Virtual del Estado de Michoacán. Cuenta con reconocimiento al perfil PRODEP de forma consecutiva en cuatro ocasiones. Entre sus publicaciones recientes se encuentra el artículo “EpAA: entorno para el aprendizaje de algoritmos. Una experiencia educativa desde la perspectiva del aprendizaje flexible” (2022). Es miembro del Cuerpo Académico Realidad Virtual y Aplicaciones Didácticas. Correo electrónico: jjap@sandunga.unistmo.edu.mx. ID: <https://orcid.org/0000-0003-0609-9470>.

Rocío Solar González. Profesora-Investigadora de la Universidad del Istmo, campus Tehuantepec, México. Es Doctora en Ingeniería Química y se encuentra adscrita a la carrera de Ingeniería Química. Obtuvo el reconocimiento al perfil PRODEP y ha sido miembro del SNI. Entre su producción académica reciente está la participación en el SENIE 2023 con la ponencia “Carga cognitiva de una secuencia de videos para desarrollar el juego del Snake en un curso de programación estructurada”. Correo electrónico: solgr@sandunga.unistmo.edu.mx. ID: <https://orcid.org/0009-0001-6499-2059>.

Luis Armería Zavala. Asesor académico de la Secretaría de Educación en el Estado de Michoacán, México. Es Doctor en Tecnologías de Información por la Universidad de Guadalajara. Tiene el reconocimiento de investigador estatal otorgado por el Instituto de Investigación Ciencia y Tecnología e Innovación. Entre sus publicaciones recientes se encuentra el capítulo “Comprar con responsabilidad”, del libro de Proyectos Comunitarios Cuarto Grado, editado por la Secretaría de Educación Pública (2023). Además de su actividad como asesor académico de la Dirección de Proyectos de Investigación e Innovación, es docente en el Instituto Michoacano de Ciencias de la Educación “José María Morelos”. Correo electrónico: luis.armeria@see.gob.mx. ID: <https://orcid.org/0000-0002-4698-8451>.

recursos didácticos más empleados en un primer curso de programación estructurada resulta útil al enseñar otros paradigmas de programación.

Desde la perspectiva de Mansilla y Beltrán (2013), una estrategia didáctica implica la planificación de actividades de aprendizaje orientadas a facilitar el logro de determinados objetivos educativos a través de la interacción de los estudiantes con los contenidos. En la selección de una estrategia didáctica Tobón (2013) establece siete principios de los cuales es posible extraer las características que debe cumplir una estrategia didáctica: 1) permitir al estudiante tomar un papel activo, 2) permitir la reflexión del que aprende y el que enseña, 3) ser inclusiva con los diversos grados de competencia de los estudiantes, 4) adecuarse a los aspectos culturales de los estudiantes, 5) pertinente para abordar procesos del mundo real, 6) congruente en los pasos, técnicas y actividades, y 7) contar con aspectos motivacionales. Así, las unidades de análisis por abordar en esta investigación son estrategia didáctica y recurso didáctico.

El término *recurso didáctico* –también suele denominarse *material didáctico* o *material instruccional*– hace referencia a aquello que facilita las condiciones necesarias para lograr cierto aprendizaje durante la práctica educativa ya sea dentro o fuera del aula (Pastrán et al., 2017). Se podría pensar que cualquier objeto físico o virtual (digital) puede constituirse como un medio facilitador de aprendizajes, y aunque esto es cierto, resulta más apropiado que dichos medios, es decir, recursos didácticos, sean acordes al área disciplinar en la cual se pretende proporcionar apoyos para el aprendizaje y además contribuyan a motivar al estudiante a aprender.

Uno de los recursos didácticos que adquirió mayor auge a partir del confinamiento derivado de la COVID-19 es el denominado *recurso didáctico digital*, utilizado con el propósito de fortalecer los aprendizajes previos, contribuyendo en la integración y asimilación de nuevos conocimientos, pero mediante el uso de plataformas digitales. Entre los recursos didácticos digitales es posible encontrar una gran variedad, por ejemplo: videos, videos interactivos, blogs, chats, foros, *software*, simuladores, infografías, archivos digitales, etc. El uso de uno o diversos recursos didácticos digitales tiene la capacidad de potencializar los aprendizajes cuando se articulan con estrategias didácticas activas centradas en el estudiante (Morales, 2021).

Cabe mencionar que la presente investigación forma parte de un proyecto más amplio que busca desarrollar e integrar recursos didácticos digitales con estrategias didácticas activas centradas en el estudiante, con la finalidad de valorar y contrastar la percepción de los estudiantes de cursos introductorios a la lógica de programación en la modalidad 100% a distancia versus modalidad 100% presencial para determinar si existen diferencias significativas entre la motivación que produce el uso de los recursos y estrategias didácticas. Pero para lograr dicho fin es necesario indagar en la literatura científica cuáles son los recursos y estrategias didácticas más utilizadas en los cursos introductorios a la lógica de programación. Conocer esto también es importante para

investigaciones cuyo propósito incluya crear, mejorar o simplemente aplicar estrategias y recursos didácticos en asignaturas relacionadas con la programación estructurada.

METODOLOGÍA

En este trabajo de investigación documental se aplicó el proceso de revisión sistemática de la literatura (SLR, del inglés *systematic literature review*), propuesto en Brereton et al. (2007) y ampliado en Kitchenham et. al. (2010), pero aplicado a la búsqueda de información respecto a la temática de estrategias y recursos didácticos empleados en el aprendizaje de la lógica de programación estructurada en educación superior. El proceso de SLR consta de tres fases: 1) planeación de la revisión, 2) ejecución de la revisión y 3) reporte de resultados de la revisión. Cada una de estas tres fases tiene a su vez determinadas etapas.

Durante la fase de planeación ocurren las etapas de: a) establecer la pregunta o preguntas de investigación, b) crear el protocolo de revisión especificando la o las bases de datos y los términos de búsqueda, así como los criterios de inclusión y exclusión, y c) evaluar el protocolo de revisión. En la fase de ejecución de la revisión se dan las etapas de: a) identificar y seleccionar estudios primarios eliminando falsos positivos partiendo del título e información dada por la base de datos, b) aplicar criterios de inclusión y exclusión considerando la lectura del resumen y conclusiones, c) identificar y eliminar fuentes duplicadas, d) asegurar la calidad de las fuentes respecto a la afinidad con la investigación, e) extracción de datos de las fuentes primarias y f) Análisis y síntesis de los resultados. En la última fase de reporte se abordan las etapas de: a) especificación de la estrategia de publicación y b) formato del documento de resultados acorde al foro de publicación.

Fase de planeación

Pregunta de investigación

La pregunta de investigación que surgió como punto de partida para realizar la SLR es: ¿Qué recursos y estrategias didácticas se han empleado en los cursos introductorios a la lógica de programación estructurada a nivel superior para motivar a los estudiantes a mejorar su rendimiento académico? La intención es identificar además qué tan relevante puede ser el perfil de los estudiantes a la hora de seleccionar una estrategia o recurso didáctico en particular.

Protocolo de revisión

Para la búsqueda de fuentes primarias se optó por utilizar la base de datos de Google Académico, ya que, de acuerdo con Martín-Martín et al. (2018b), el enfoque inclusivo de indexación de Google Académico proporciona una cobertura de documentos

científicos y académicos, en diversas disciplinas e idiomas, potencialmente más completa en comparación con bases de datos más selectivas como WoS (Web of Science) y Scopus. Dada la pregunta de investigación y con el fin de abarcar un conjunto significativo de potenciales fuentes primarias se articularon cadenas de búsqueda tanto en el idioma español como en el idioma inglés.

En español las cadenas de búsqueda fueron: a) “curso introductorio” + “lógica de programación”, b) “lógica de programación” + “motivación” + “recursos didácticos”, c) “lógica de programación” + “rendimiento académico” y d) “lógica de programación” + “reprobación”. En inglés las cadenas fueron: a) “introductory course” + “programming logic”, b) “programming logic” + “motivation” + “educational resources”, c) “programming logic” + “academic performance” y d) “programming logic” + “failed students”.

La búsqueda de las cadenas en español se configuró en Google Académico para filtrar automáticamente de acuerdo con los siguientes criterios: desde el año 2018 hasta el 2022, ordenados por relevancia, buscar solo páginas en español, sin incluir patentes o citas; en tanto que para las cadenas en inglés se aplicaron los dos primeros filtros y el último, correspondientes a la fecha, relevancia y exclusión de patentes o citas, respectivamente, pero aceptando publicaciones en cualquier idioma. De los resultados obtenidos se excluyeron aquellas fuentes que en el título o en la breve información proporcionada por el buscador reflejen alguna de estas características:

- Uso del paradigma orientado a objetos u otro paradigma de programación distinto al paradigma estructurado.
- Documentos que no son productos de investigación, como libros de programación, apuntes de asignaturas o programas de estudio.
- Abordaje de asignaturas o tópicos de estudio más avanzados a la lógica de programación, por ejemplo: estructuras de datos, programación paralela, árboles, etc.
- Ni el título de la fuente ni la breve descripción guardan relación alguna con la pregunta de investigación.
- Mismo documento repetido en diferente repositorio o base de datos, ambas indexadas en Google Académico.
- No se aborda esencialmente la temática de estudio, es decir, la lógica de programación estructurada, aun cuando en la breve descripción del buscador sí aparecen las cadenas de búsqueda.
- Las fuentes cuyos documentos no estén en los idiomas español o inglés.
- Aquellas fuentes que al momento de su búsqueda no pudieron ser localizados de forma completa, ya sea por un enlace roto o fuera de línea o incluso porque el antivirus los señale como sitios peligrosos.

Por el contrario, sí se contemplaron aquellas fuentes que en el título y la descripción incluían las cadenas de búsqueda, y que además reflejaran cierta relación fuerte o débil con el tema de estudio y la pregunta de investigación. Posterior a su descarga se analizaron con más detenimiento sus correspondientes resúmenes, metodología y conclusiones para filtrar únicamente aquellos documentos con la mayor afinidad al tema y pregunta de investigación que da origen a la SLR. Cabe mencionar que en el caso de trabajos de tesis solo se incluyeron las de nivel de doctorado. Las búsquedas se realizaron en el transcurso del mes de diciembre del 2022.

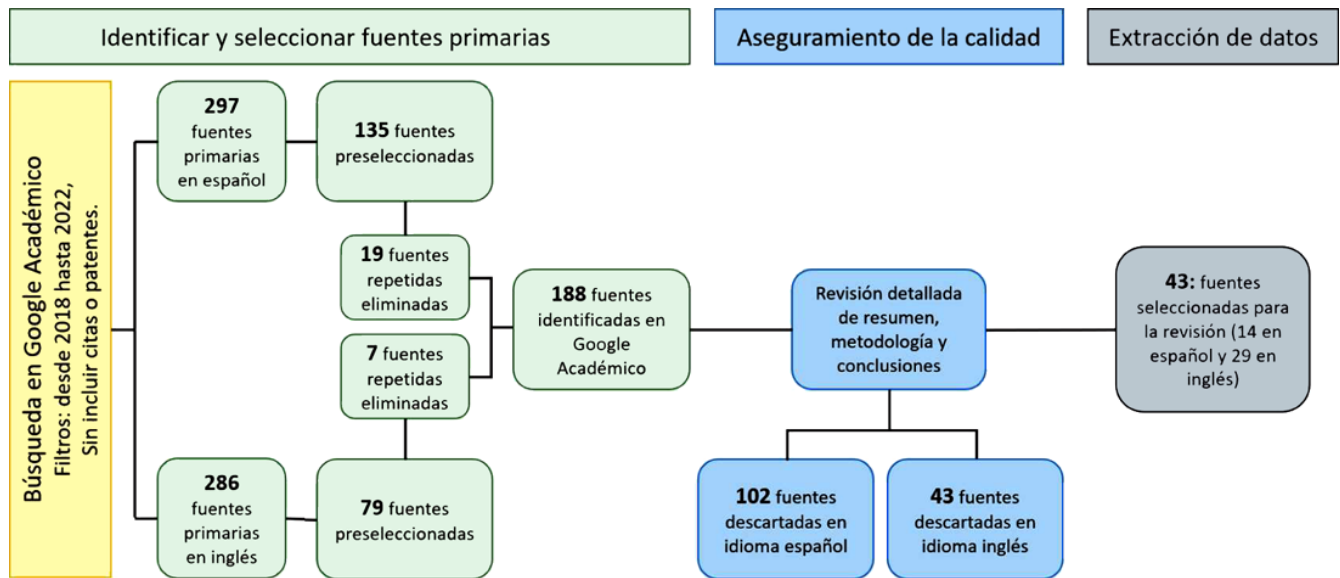
Evaluación del protocolo de revisión

Comúnmente las SLR toman más de un motor de bases de datos para realizar la búsqueda de fuentes primarias, no obstante, de acuerdo con Martín-Martín et al. (2018), Google Académico funciona como un superconjunto de lo que otras bases de datos indexan, en particular de aquellos artículos y revisiones científicas, en diversas áreas del conocimiento, catalogadas como de acceso abierto (OA, del inglés *open access*), así pues, en su estudio demuestran que poco más de la mitad de estos documentos científicos estarán indexados en Google Académico. Por lo tanto, el plan para la SLR se considera válido pues de manera indirecta se recuperarán fuentes primarias de bases de datos especializadas a través de Google Académico. Además existe evidencia de SLR publicadas en revistas indexadas que utilizando un solo motor de búsqueda han recuperado información relevante de su campo de estudio, por ejemplo Montiel y Gomez-Zermeño (2021).

Fase de ejecución

Se procedió de acuerdo con la planificación para la identificación y selección de fuentes primarias considerando los filtros de búsqueda así como los criterios de exclusión e inclusión, obteniéndose los resultados mostrados en la Figura 1. Posteriormente en la fase de aseguramiento de la calidad se descartaron aquellas fuentes que no tienen una estrecha relación con el tema y la pregunta de investigación, es decir, solo se consideraron estudios cuya muestra estuviera compuesta de estudiantes universitarios de nuevo ingreso, inscritos en cursos de lógica de programación estructurada, o estudios de revisiones de la literatura abocadas a cursos introductorios de lógica de programación. Al final, solo 43 fuentes primarias fueron seleccionadas para la extracción de datos.

Figura 1
Resultados de la fase de ejecución de la SLR



Fuente: Construcción personal.

Fase de reporte de resultados de la revisión

Como parte de la estrategia de publicación se tomó la decisión de elegir una revista indexada en el área educativa que aceptara contribuciones de tipo revisiones sistemáticas de la literatura o estudios del estado del arte. La revista *IE Revista de Investigación Educativa* de la REDIECH fue seleccionada por su calidad, además de cumplir con lo establecido en la estrategia de publicación, por lo tanto, los resultados y el texto del artículo en general se apegaron a las guías de autor de esta revista. De acuerdo con la metodología, en el siguiente apartado se presentan los resultados más relevantes de la SLR.

RESULTADOS

Fuentes primarias seleccionadas

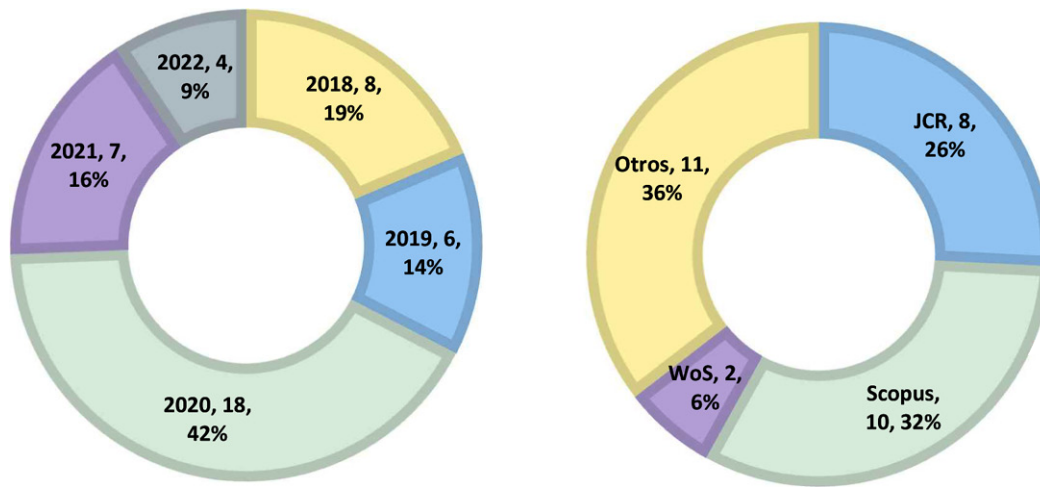
En la Tabla 1 se muestran las fuentes primarias seleccionadas en orden cronológico desde el 2018 hasta el 2022; también se indica el tipo de fuente, así como la indexación, congreso o institución en el caso de las revistas, memorias o tesis, respectivamente. De la información vertida en dicha tabla se obtiene que el 42% (18/43) de las fuentes se publicaron en el año 2020. En cuanto a la indexación de las revistas, el 32% (10/31) se indexan en Scopus, 26% (8/31) se indexan en JCR, 6% (2/31) están indexadas en WoS y el restante 36% (11/31) corresponde a otras indexaciones como Dialnet, Redalyc, Latindex, EBSCO, ERIC, Microsoft Academic y Google Académico (ver Figura 2). Además, la editorial que más sobresale en los congresos es la IEEE, con 57% (4/7).

Tabla 1*Fuentes primarias seleccionadas para la revisión sistemática de la literatura*

No.	Referencia	Tipo de fuente	Indexación/Editorial del congreso/ Universidad
1	Caceffo et al., 2018	Congreso	ACM
2	Gajewski, 2018	Congreso	Univ Silesia
3	Juárez et al., 2018	Congreso	IEEE
4	Çakiroğlu et al., 2018	Revista	EBSCO, ERIC
5	Hartati, 2018	Revista	Google Scholar
6	Pérez y Pedroza, 2018	Revista	Redalyc, Dialnet
7	Yi y Lee, 2018	Revista	Scopus
8	Vasilopoulos y van Schaik, 2018	Revista	JCR, Scopus
9	Viola y Gómez, 2019	Revista	WoS, MIAR
10	Mutanu y Machoka, 2019	Capítulo de libro	Springer
11	Méndez, 2019	Congreso	Universidad de Cundinamarca
12	Jiménez et al., 2019	Revista	WoS, Redalyc
13	Molina et al., 2019	Revista	Redalyc, Dialnet
14	Solarte y Machuca, 2019	Revista	Scopus
15	Anderson y González, 2020	Capítulo de libro	Redie, UPN
16	Begosso et al., 2020	Congreso	IEEE
17	Silva y Silveira, 2020	Revista	Scopus
18	da Costa et al., 2020	Congreso	IEEE
19	Khomokhoana ,2020	Tesis de Doctorado	University of the free state
20	Yücel, 2018	Tesis de Doctorado	Middle East Technical University
21	Dhakshina y Dhakshina, 2020	Revista	JCR
22	González et al., 2020	Revista	Microsoft Academic, ERIC
23	Hernández et al., 2020	Revista	Latindex Catálogo
24	Kandemir et al., 2020	Revista	JCR
25	Kazimoglu, 2020	Revista	JCR
26	Mehmood et al., 2020	Revista	JCR
27	Roa y Martínez, 2020	Revista	Dialnet, Road
28	Servita y Sánchez, 2020	Revista	Google Scholar, REDIB
29	Sharma et al., 2020	Revista	Google Scholar, ProQuest
30	Silva et al., 2020	Congreso	IEEE
31	Zang et al., 2020	Revista	Scopus
32	Castro y González, 2021	Revista	Dialnet, EBSCO, REDIB
33	Correa et al., 2021	Capítulo de libro	UNED
34	Cárdenas et al., 2021	Revista	Scopus, WoS
35	Cheng et al., 2021	Revista	JCR, Scopus, ACM
36	Coban et al., 2021	Revista	ERIC, Index Copernicus
37	Dhayaniithia et al., 2021	Revista	Scopus
38	Kayaalp y Dinc, 2021	Revista	JCR
39	Susanti et al., 2021	Revista	Scopus
40	Arellano y Canedo, 2022	Revista	Scopus
41	Ochoa y Bedregal, 2022	Revista	Scopus, WoS
42	Pérez y Cela, 2022	Revista	JCR, Scopus
43	Ramírez et al., 2022	Revista	Scopus

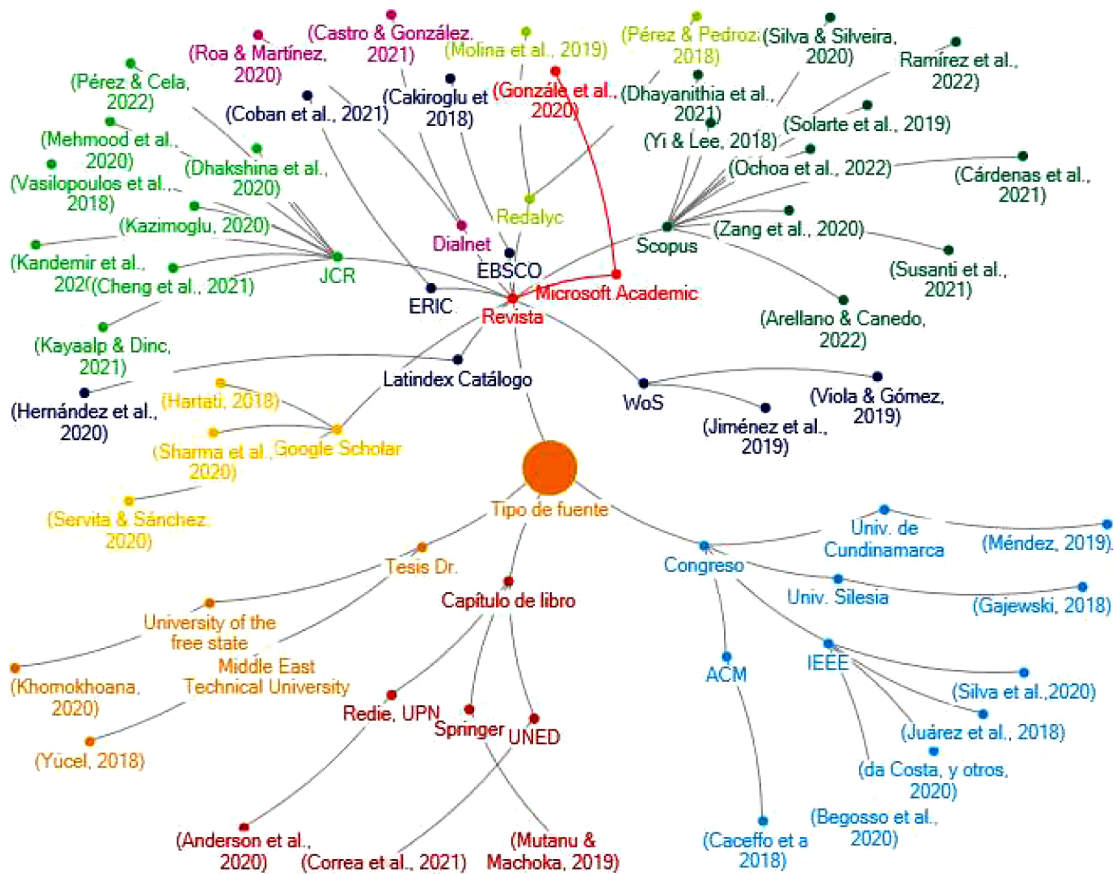
Fuente: Construcción personal.

Figura 2
 Resultados de la SRL por año e indexación de las revistas



Fuente: Construcción personal.

Figura 3
 Representación gráfica de los clusters por tipo de fuente identificada en la SLR



Fuente: Construcción personal.

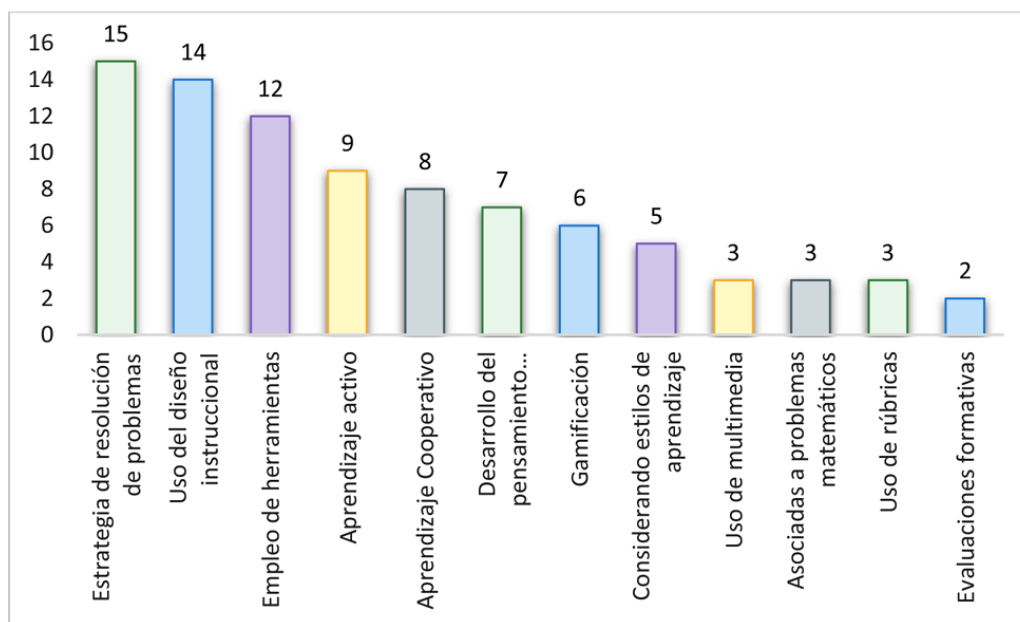
Para contextualizar un poco más la relación entre las estrategias y recursos didácticos para el estudio de la programación estructurada, identificados a partir de la SRL de las diversas fuentes de información, también se presentan gráficamente los resultados de la fase de ejecución. En la construcción de este mapeo se utilizó el *software* NodeXL, el cual se diseñó como una herramienta de visualización y análisis de redes sociales, mediante una plantilla de Microsoft Excel, versión 2017 y posteriores (Ahmed y Lugovic, 2019; Miltchev et al., 2014). La revisión de la literatura permite identificar que, por sus características, NodeXL se utiliza también en minería de datos para procesar la visualización de redes (Zhao et al., 2015), y en el cálculo de métricas simples (Probst y Peng, 2019; Tripathi et al., 2019). Así, mediante dicha herramienta fue factible identificar agrupaciones de datos, o *clusters*, como se muestra en la Figura 3, que sirven de base para establecer categorías de estudio de las fuentes de información. Para facilitar la legibilidad de la figura se ha considerado el apellido del primer autor y el año de publicación de la obra.

Estrategias didácticas identificadas

Las estrategias didácticas identificadas en la SLR se muestran en la Figura 4, cabe mencionar que algunas publicaciones emplean más de una estrategia, pero las cinco que más sobresalen por su frecuencia de uso son: 1) el empleo de estrategias enfocadas en desarrollar habilidades para la resolución de problemas, 2) el uso de algún tipo de diseño instruccional o secuencias didácticas, 3) el empleo de herramientas

Figura 4

Estrategias didácticas identificadas en la SLR



Fuente: Construcción personal.

tecnológicas (*software*) para ayudar al estudiante a comprender y asimilar la lógica de programación estructurada, 4) el uso de estrategias de aprendizaje activo como el aula invertida o el aprendizaje basado en problemas, entre otros, y 5) el aprendizaje cooperativo o trabajo en equipo. En la Tabla 2 se muestran las fuentes que abordan cada una de las cinco estrategias didácticas más utilizadas aquí listadas.

Tabla 2

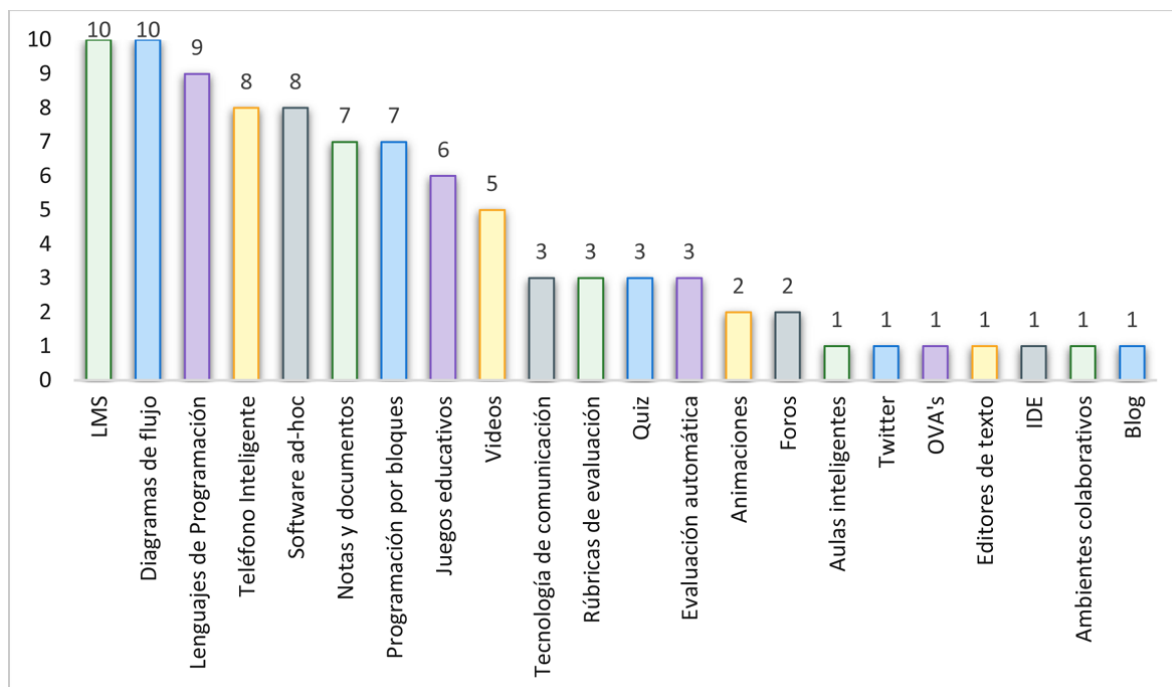
Las cinco estrategias didácticas más utilizadas extraídas de la SLR

Estrategia didáctica	Fuentes que abordan su uso
1) Empleo de estrategias enfocadas en desarrollar habilidades para la resolución de problemas	Juárez-Ramírez et al., 2018; Yi y Lee, 2018; Jiménez et al., 2019; Solarte y Machuca, 2019; Anderson y González, 2020; Da Costa et al., 2020; Kandemir et al., 2020; Mehmood et al., 2020; Servita y Sánchez, 2020; Sharma et al., 2020; Correa et al., 2021; Coban et al., 2021; Susanti et al., 2021; Arellano y Canedo, 2022; Ramírez et al., 2022
2) Uso de algún tipo de diseño instruccional o secuencias didácticas	Pérez y Pedroza, 2018; Viola y Gómez, 2019; Jiménez et al., 2019; Silva y Silveira, 2020; Da Costa et al., 2020; Khomokhoana, 2020; Yücel, 2018; Kandemir et al., 2020; Roa y Martínez, 2020; Servita y Sánchez, 2020; Zang et al., 2020; Cheng et al., 2021; Arellano y Canedo, 2022; Pérez-Suasnavas y Cela, 2022
3) Empleo de herramientas tecnológicas (<i>software</i>) para ayudar al estudiante a comprender y asimilar la lógica de programación estructurada	Vasilopoulos y van Schaik, 2018; Molina et al., 2019; Anderson y González, 2020; Dhakshina y Dhakshina, 2020; Hernández et al., 2020; Mehmood et al., 2020; Silva et al., 2020; Zang et al., 2020; Cárdenas et al., 2021; Kayaalp y Dinc, 2021; Arellano y Canedo, 2022; Ramírez et al., 2022
4) Uso de estrategias de aprendizaje activo como el aula invertida o el aprendizaje basado en problemas, entre otras	Caceffo et al., 2018; Yücel, 2018; Dhakshina y Dhakshina, 2020; Mehmood et al., 2020; Roa y Martínez, 2020; Sharma et al., 2020; Castro y González, 2021; Susanti et al., 2021; Arellano y Canedo, 2022
5) Aprendizaje cooperativo o trabajo en equipo	Caceffo et al., 2018; Méndez, 2019; Da Costa et al., 2020; Yücel, 2018; Servita y Sánchez, 2020; Correa et al., 2021; Cheng et al., 2021; Susanti et al., 2021

Fuente: Construcción personal.

Recursos didácticos identificados

Los recursos didácticos identificados son muy diversos y variados, llegándose a utilizar uno o más de uno en cada fuente, aunque también existen otras fuentes que solo enfatizan en la estrategia y no señalan específicamente algún tipo de recurso didáctico. Considerando las fuentes que sí enfatizan específicamente en uno o más recursos, los que se identificaron se presentan en la Figura 5. Los cinco recursos que más destacan son: 1) uso de algún LMS para suministrar contenidos, actividades y evaluaciones; 2) empleo de diagramas de flujo, ya sea como herramientas especializadas o como parte de la explicación de la lógica de programación; 3) el uso directo de algún lenguaje de programación; 4) el empleo del teléfono inteligente para ejecutar aplicaciones o accederlas en línea, y 5) empleo de *software* construido *ad hoc* para la comprensión y asimilación de la lógica de programación. En la Tabla 3 se presentan las fuentes que abordan cada uno de los cinco recursos más utilizados de acuerdo con la SLR aquí desarrollada.

Figura 5*Tipo de recursos didácticos identificados en la SLR**Fuente:* Construcción personal.**Tabla 3***Los cinco recursos didácticos más utilizados extraídos de la SLR*

Recurso didáctico	Fuentes que abordan su uso
1) Uso de algún LMS para suministrar contenidos, actividades y evaluaciones	Gajewski, 2018; Viola y Gómez, 2019; Molina et al., 2019; Dhakshina y Dhakshina, 2020; Mehmood et al., 2020; Sharma et al., 2020; Roa y Martínez, 2020; Zang et al., 2020; Cheng et al., 2021; Arellano y Canedo, 2022
2) Empleo de diagramas de flujo ya sea a través de herramientas especializadas o como parte de la explicación de la lógica de programación	Gajewski, 2018; Yi y Lee, 2018; Vasilopoulos y van Schaik, 2018; Jiménez et al., 2019; Solarte y Machuca, 2019; Anderson y González, 2020; Mehmood et al., 2020; Kayaalp y Dinc, 2021; Arellano y Canedo, 2022; Ochoa y Bedregal, 2022
3) Uso directo de algún lenguaje de programación	Caceffo et al., 2018; Yi y Lee, 2018; Viola y Gómez, 2019; Jiménez et al., 2019; Solarte y Machuca, 2019; Da Costa et al., 2020; Dhayanithia et al., 2021; Susanti et al., 2021; Ochoa y Bedregal, 2022
4) Empleo del teléfono inteligente para ejecutar aplicaciones o accederlas en línea	Caceffo et al., 2018; Méndez, 2019; Molina et al., 2019; Yücel, 2018; Sharma et al., 2020; Kayaalp y Dinc, 2021; Arellano y Canedo, 2022; Pérez-Suasnavas y Cela, 2022
5) Empleo de <i>software</i> construido <i>ad-hoc</i> para la comprensión y asimilación de la lógica de programación	Yi y Lee, 2018; Vasilopoulos y van Schaik, 2018; Méndez, 2019; Hernández et al., 2020; Kazimoglu, 2020; Silva et al., 2020; Kayaalp y Dinc, 2021; Arellano y Canedo, 2022

Fuente: Construcción personal.

Los resultados, además de responder la pregunta de investigación al identificar las estrategias y recursos didácticos más utilizados en el aprendizaje de la programación estructurada en educación superior, identifican una práctica docente mediada por tecnología. A partir de la SLR, se observa el uso instrumental de diversos recursos digitales y su apoyo en el ejercicio de las habilidades de pensamiento necesarias para desarrollar la lógica de programación.

DISCUSIÓN Y CONCLUSIONES

A pesar de solo emplear Google Académico como motor de búsqueda para realizar esta SLR, es de notar que la mayoría de las publicaciones de revistas están indexadas en Scopus, seguido de las indexadas en JCR y un pequeño porcentaje en WoS, la suma en total de estas indexaciones es del 64% (20/31), esto de alguna forma confirma lo argumentado por Martín-Martín et al. (2018b) respecto a que Google Académico funciona como un superconjunto de lo que otras bases de datos especializadas indexan; además se incluyen artículos publicados en revistas arbitradas e indexadas que por lo común suelen quedar fuera de las SLR, pero que también pueden aportar datos interesantes, en particular para esta SLR se consideraron 11 artículos más, que representan el 36% del total de artículos de revista. Gracias al motor de búsqueda de Google Académico también fue posible incluir otros *clusters* constituidos por memorias de congresos, capítulos de libros y tesis de doctorado, de tal forma que se contó con una cobertura de fuentes primarias más amplia que cuando solo se utilizan bases de datos especializadas.

Dados los resultados del análisis, es importante mencionar que tanto las estrategias como los recursos didácticos deberían adaptarse al perfil de los estudiantes, los cuales ahora más que nunca están habituados a interactuar de forma natural con la tecnología a su alrededor. A nivel universitario se habla de un nuevo perfil de estudiantes catalogado como generación Centennial, I-Gen, D-Gen o Generación Z (Bertuzzi, 2021). En este sentido, no son pocos los autores que señalan la importancia de considerar este nuevo perfil, por ejemplo, Dhakshina y Dhakshina (2020) reconocen la valía de las herramientas didácticas basadas en la tecnología enfocadas a las nuevas generaciones de aprendices. Hernández et al. (2020) proponen el desarrollo de *software ad hoc* para adecuarse a la forma de aprender de los nativos digitales. Kazimoglu (2020) sugiere el uso de nuevos métodos motivacionales, como la gamificación, para adaptarse a los nuevos perfiles. En cuanto a Kayaalp y Dinc (2021), enfatizan el hecho de que el perfil de los nuevos usuarios debe abordarse desde el tipo de tecnología que están habituados a utilizar.

De los cinco recursos didácticos identificados de mayor uso en la SLR, solo uno de ellos pudiera estar o no relacionado con el uso directo de tecnología, se trata de los diagramas de flujo, cuyo uso puede generar errores en su diseño, frecuentemente esto

ocurre a la hora de finalizarlos (Narváez y López, 2022). Aunque también es cierto que herramientas de *software ad hoc* ayudan precisamente a evitar este tipo de errores frecuentes en la construcción de los diagramas de flujo (Arellano et al., 2022). Tres de los cuatro recursos didácticos restantes están vinculados intrínsecamente al uso de tecnología y uno más es precisamente el uso del teléfono móvil.

Sin embargo, también debe considerarse que recurrir solo a la tecnología que emplean habitualmente los estudiantes, incorporándola como recurso didáctico durante el proceso de enseñanza-aprendizaje, no es algo que por sí solo tenga la capacidad de mejorar los aprendizajes. De acuerdo con Sánchez et al. (2018), tiene mayor preponderancia el uso de estrategias educativas centradas en el estudiante que un uso excesivo de la tecnología sin fundamento pedagógico. Por lo tanto, no solo es cuestión de recursos didácticos *ad hoc* al perfil del estudiante, también es de suma importancia el fundamento pedagógico que los sustenta (Arellano y Canedo, 2022), es decir, la estrategia didáctica debe ser también acorde al perfil y contexto del estudiante para favorecer su motivación por aprender.

Particularmente, por el lado de las estrategias didácticas, con frecuencia se enfocan en desarrollar habilidades para la resolución de problemas, lo cual es algo inherente a la asignatura ya que el profesor plantea un problema de forma textual o verbal y el estudiante debe poner en marcha sus habilidades del pensamiento de orden superior para diseñar, implementar y probar su estrategia de solución. Otra estrategia estriba en emplear un diseño instruccional o secuencia didáctica, donde primero se abordan conceptos más sencillos y sobre ellos se construyen conceptos más complejos. También se suele recurrir a herramientas tecnológicas de *software* como apoyo al proceso de enseñanza-aprendizaje, y toda la estrategia didáctica gira en torno a dichas herramientas. Algo que empieza a tener mayor fuerza es aprovechar estrategias de aprendizaje activo, como el aula invertida, tomando ventaja de recursos como los LMS y otras plataformas de uso masivo en línea, por ejemplo YouTube. Una estrategia más que suele usarse con frecuencia es el aprendizaje cooperativo, por ejemplo, en su modalidad de aprendizaje por pares.

Un hallazgo interesante en esta SLR fue que las publicaciones entre los años 2018 y 2019 no realizaron sus estudios en un contexto de confinamiento por la pandemia derivada de la COVID-19, mientras que en el resto de los años sí existe esa posibilidad. De hecho, para el año 2022 deberían existir publicaciones donde ya se ha considerado el retorno a las aulas universitarias, sin embargo, esto no es así. Al momento de realizar la SLR ninguno de los trabajos identificados aborda el retorno a las aulas, esto en sí configura un área de oportunidad para investigar si las estrategias implementadas durante la pandemia se heredaron a la presencialidad y determinar cuál puede ser su eficacia.

Por otro lado, al ser el uso de las herramientas de *software* una de las estrategias didácticas más utilizadas, así como los diagramas de flujo uno de los recursos didác-

ticos a los que más se recurre cuando se trata de enseñar la lógica de programación estructurada a nivel superior, resulta de interés como trabajo a futuro indagar con mayor profundidad este binomio de estrategias y recursos didácticos.

REFERENCIAS

- Ahmed, W., y Lugovic, S. (2019). Social media analytics: analysis and visualization of news diffusion using NodeXL. *Online Information Review*, 43(1), 149-160. <https://doi.org/10.1108/OIR-03-2018-0093>
- Anderson, J. A., y González, D. L. (2020). PSeInt como herramienta para mejorar el proceso de enseñanza aprendizaje de algoritmos, pseudocódigo y diagramas de flujo. En D. L. González, M. I. Domínguez, y E. A. Flores, *Tecnologías de la información en educación. Sistematización de experiencias docentes* (pp. 91-102). Redie. <http://dx.doi.org/10.6084/m9.figshare.12498095.v1>
- Arellano, J. J., y Canedo, S. P. (2022). EpAA: entorno para el aprendizaje de algoritmos. Una experiencia educativa desde la perspectiva del aprendizaje flexible. *EduTec*, (79), 63-79. <https://doi.org/10.21556/edutec.2022.79.2451>
- Arellano, J. J., Solar, R., Nieva, O., y Canedo, S. (2022). Compilador e intérprete en línea de diagramas de flujo con fines didácticos. *Revista de Investigación en Tecnologías de la Información*, 10(20), 63-79. <https://doi.org/10.36825/RITI.10.20.007>
- Begosso, L. C., Begosso, L. R., y Aragao, N. (2020). *An analysis of block-based programming environments for CS1*. IEEE Frontiers in Education Conference (FIE). IEEE. <https://ieeexplore.ieee.org/document/9273982>
- Bertuzzi, M. F. (2021). Centennials en la universidad: prosumidores de contenido en el aula. En M. Veneziani y P. de la Sota, *Cuadernos del Centro de Estudios en Diseño y Comunicación No. 134* (pp. 161-173). Universidad de Palermo. https://fido.palermo.edu/servicios_dyc/publicacionesdc/cuadernos/detalle_articulo.php?id_libro=903&id_articulo=18131
- Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., y Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4), 571-583. <https://doi.org/10.1016/j.jss.2006.07.009>
- Caceffo, R., Gama, G., y Azevedo, R. (2018). *Exploring active learning approaches to computer science classes*. Proceedings of the 49th ACM Technical Symposium on Computer Science Education (pp. 922-927). ACM. <https://doi.org/10.1145/3159450.3159585>
- Çakiroğlu, Ü., Er, B., Uğur, N., y Aydoğdu, E. (2018). Exploring the use of self-regulation strategies in programming with regard to learning styles. *International Journal of Computer Science Education in Schools*, 2(2), 14-28. <https://doi.org/10.21585/ijcses.v2i2.29>
- Cárdenas, J., Puris, A., Novoa, P., Parra, Á., Moreno, J., y Benavides, D. (2021). Using scratch to improve learning programming in college students: A positive experience from a non-WEIRD country. *Electronics*, 10(10), 1180. <https://doi.org/10.3390/electronics10101180>
- Castro, A., y González, J. S. (2021). Experiencias universitarias de aula en la introducción a la programación. *Cuaderno de Pedagogía Universitaria*, 18(35), 85-94. <https://doi.org/10.29197/cpu.v18i35.415>
- Cheng, Y. P., Shen, P. D., Hung, M. L., Tsai, C. W., Lin, C. H., y Hsu, L. C. (2021). Applying online content based knowledge awareness and team learning to develop students' programming skills, reduce their anxiety, and regulate cognitive load in a cloud classroom. *Universal Access in the Information Society*, 21, 557-572. <https://doi.org/10.1007/s10209-020-00789-6>
- Coban, E., Kolburan, A., y Deveci, A. (2021). The effect of programming with scratch course on reflective thinking skills of students towards problem solving. *Journal of Learning and Teaching in Digital Age*, 6(1), 72-80. <https://dergipark.org.tr/en/pub/joltida/issue/59433/854205>
- Correa, A. J., Amórtegui, M. P., y Jiménez, J. H. (2021). Clase espejo como estrategia pedagógica complementaria para el aprendizaje de algoritmos y el fomento de habilidades interculturales en el escenario de redes académicas internacionales. En M. Barrientos, y H. E. (eds.), *II Encuentro Regional de AIESAD 2021* (pp. 45-54). UNED. <https://hemeroteca.unad.edu.co/index.php/memorias/article/view/5733>

- Da Costa, R. C., Vargas, A., Billa, C., Santos, R., Barwaldt, R., y Bothelho, S. (2020). Methodological changes in teaching algorithms in the early years of the Computer Engineering course. *IEEE Frontiers in Education Conference (FIE)* (pp. 1-6). IEEE. <https://ieeexplore.ieee.org/document/9274109>
- Dhakshina, A., y Dhakshina, M. (2020). Online coding event as a formative assessment tool in introductory programming and algorithmic courses—An exploration study. *Computer Applications in Engineering Education*, 28(6), 1-11. <https://doi.org/10.1002/cae.22330>
- Dhayanithi, J., Balamurugan, D., Marimuthu, M., Vidya-bharathi, D., Basker, N., Vidhya, G., Theetchenya, S., y Mohanraj, G. (2021). An evaluation framework and analysis of auto assessing the programming courses during the COVID-19 pandemic. *Turkish Journal of Computer and Mathematics Education*, 12(10), 4918-4923. <https://turcomat.org/index.php/turkbilmate/article/view/5254>
- Gajewski, R. R. (2018). Algorithms, programming, flowcharts and flowgorithm. En E. Smyrnova-Trybulska (ed.), *E-learning and smart learning environment for the preparation of new generation specialist* (pp. 393-408). University of Silesia. <http://druk.studio-noa.pl/ig/pub/us/E-1-10/10-393.pdf>
- González, E. I., Trujillo, V., y Bautista, J. (2020). Procesos metacognitivos en el aprendizaje de programación, base del futuro. *European Journal of Education Studies*, 7(9), 174-193. <https://oapub.org/edu/index.php/ejes/article/view/3243>
- Hartati, S. J. (2018). Framework design learning of introduction to computational algorithms by using the theory learning by doing. *Journal of Physics: Conference Series*, 1028(1), 1-5. <https://dx.doi.org/10.1088/1742-6596/1028/1/012133>
- Hernández, I. Y., González, R., y Nieto, A. D. (2020). Percepción de estudiantes universitarios sobre el uso de software educativo en la enseñanza de la programación. *Revista Electrónica ANFEI Digital*, 7(12), 1-9. <https://anfei.mx/revista/index.php/revista/article/view/695/1336>
- Jiménez, J. A., Collazos, C., y Revelo, O. (2019). Consideraciones en los procesos de enseñanza-aprendizaje para un primer curso de programación de computadores: una revisión sistemática de la literatura. *Tecnológicas*, 22, 84-118. <https://doi.org/10.22430/22565337.1520>
- Juárez-Ramírez, R., Navarro, C. X., Tapia-Ibarra, V., Macías-Olvera, R., y Guerra-García, C. (2018). What is programming? Putting all together - A set of skills required. *2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT)* (pp. 11-20). IEEE. <https://doi.org/10.1109/CONISOFT.2018.8645956>
- Kandemir, C. M., Kalelioglu, F., y Gülbahar, Y. (2020). Pedagogy of teaching introductory text-based programming in terms of computational thinking concepts and practices. *Computer Applications in Engineering Education*, 29(1), 29-45. <https://doi.org/10.1002/cae.22374>
- Kayaalp, F., y Dinc, F. (2021). A mobile app for algorithms learning in engineering education: Drag and drop approach. *Computer Applications in Engineering Education*, 30(1), 235-250. <https://onlinelibrary.wiley.com/doi/epdf/10.1002/cae.22453>
- Kazimoglu, C. (2020). Enhancing confidence in using computational thinking skills via playing a serious game: A case study to increase motivation in learning computer programming. *IEEE Access*, 8, 221831-221851. <https://doi.org/10.1109/ACCESS.2020.3043278>
- Khomokhoana, P. J. (2020). *Source code comprehension: Decoding the cognitive challenges of novice programmers* [Tesis de doctorado]. University of the Free State, Bloemfontein, Sudáfrica. <https://scholar.ufs.ac.za/handle/11660/10946>
- Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., y Linkmana, L. (2010). Systematic literature reviews in software engineering – A tertiary study. *Information and Software Technology*, 52(8), 792-805. <https://doi.org/10.1016/j.infsof.2010.03.006>
- Mansilla, J., y Beltrán, J. (2013). Coherencia entre las estrategias didácticas y las creencias curriculares de los docentes de segundo ciclo, a partir de las actividades didácticas. *Perfiles Educativos*, 35(139), 25-39. <https://doi.org/10.22201/iisue.24486167e.2013.139.35709>
- Martín-Martín, A., Costas, R., van Leeuwen, T., y López-Cózar, E. D. (2018a). Evidence of open access of scientific publications in Google Scholar: A large-scale analysis. *Journal of Informetrics*, 12(3), 819-841. <https://doi.org/10.1016/j.joi.2018.06.012>
- Martín-Martín, A., Orduna-Malea, E., Thelwall, M., y López-Cózar, E. D. (2018b). Google Scholar, Web

- of Science, and Scopus: A systematic comparison of citations in 252 subject categories. *Journal of Informetrics*, 12(4), 1160-1177. <https://doi.org/10.1016/j.joi.2018.09.002>
- Mehmood, E., Abid, A., Farooq, M. S., y Nawaz, N. A. (2020). Curriculum, teaching and learning, and assessments for introductory programming course. *IEEE Access*, 8, 125961-125981. <https://ieeexplore.ieee.org/document/9137246>
- Méndez, J. A. (2019). Aprendizaje colaborativo de programación asistido por computador. En J. E. Márquez, *Educación, ciencia y tecnologías emergentes para la generación del siglo 21* (pp. 43-63). Universidad de Cundinamarca. <https://doi.org/10.36436/9789585203242>
- Miltchev, R., Milchev, G., y Manchev, I. (2014). Approach for integration of the methods of the network analysis in the field of distance and electronic education. *Proceedings of the 15th International Conference on Computer Systems and Technologies* (pp. 460-467). ACM. <https://doi.org/10.1145/2659532.2659604>
- Molina, R. E., Padilla, R. R., y Leyva, M. Y. (2019). Estudio y propuesta metodológica, para la enseñanza-aprendizaje de la programación informática en la educación superior. *Revista Dilemas Contemporáneos. Educación, Política y Valores*, 7, 1-23. <https://doi.org/10.46377/dilemas.v30i1.1294>
- Montiel, H., y Gomez-Zermeño, M. G. (2021). Educational challenges for computational thinking in K-12 education: A systematic literature review of “Scratch” as an innovative programming tool. *Computers*, 10(6), 69. <https://doi.org/10.3390/computers10060069>
- Morales, R. E. (2021). El video como recurso didáctico digital que fortalece el aprendizaje virtual. *Edutec. Revista Electrónica de Tecnología Educativa*, (77), 186-202. <https://doi.org/10.21556/edutec.2021.77.1939>
- Mutau, L., y Machoka, P. (2019). Enhancing computer students' academic performance through explanatory modeling. En B. Tait, J. Kroeze y S. Gruner (eds.), *ICT Education. SACLA 2019. Communications in Computer and Information Science*, vol. 1136 (pp. 227-243). Springer. https://doi.org/10.1007/978-3-030-35629-3_15
- Narváez, L. E., y López, R. E. (2022). Identificación de errores en conceptos básicos de principios de programación. *IE Revista de Investigación Educativa de la REDIECH*, 13. https://doi.org/10.33010/ie_rie_rediech.v13i0.1222
- Ochoa, L. L., y Bedregal, N. (2022). Incorporation of computational thinking practices to enhance learning in a programming course. *International Journal of Advanced Computer Science and Applications*, 13(2), 194-200. <https://dx.doi.org/10.14569/IJACSA.2022.0130224>
- Pastrán, F. R., Montilla, A., y Castillo, A. (2017). Los recursos didácticos desde el constructivismo para la enseñanza de geografía general. *Redine. Red de Investigación Educativa*, 9(1), 28-36. <https://revistas.uclave.org/index.php/redine/article/view/834>
- Pérez, J. A., y Pedroza, O. V. (2018). LM1: una metodología de estudio para la asignatura “Programación 1”. *Educere*, 22(73), 635-648. <https://www.redalyc.org/journal/356/35656676013/html/>
- Pérez-Suasnavas, A.-L., y Cela, K. (2022). Incidencia de la metodología JiTTwT en el rendimiento académico de estudiantes universitarios. *Espiral. Cuadernos del Profesorado*, 15(30). <https://doi.org/10.25115/ecp.v15i30.6500>
- Probst, Y. C., y Peng, Q. (2019). Social media in dietetics: Insights into use and user networks. *Nutrition & Dietetics*, 76(4), 414-420. <https://doi.org/10.1111/1747-0080.12488>
- Ramírez, J. J., Restrepo, F., y González, F. A. (2022). A case study in technology-enhanced learning in an introductory computer programming course. *Global Journal of Engineering Education*, 24(1), 65-71. <http://www.wiete.com.au/journals/GJEE/Publish/vol24no1/10-Restrepo-Calle-F.pdf>
- Roa, K., y Martínez, C. (2020). Diseño de un ambiente virtual de aprendizaje soportado en los estilos de aprendizaje. *Revista Virtu@lmente*, 8(2), 67-86. <https://doi.org/10.21158/2357514x.v8.n2.2020.2761>
- Sánchez, M., Escamilla, J., y Sánchez, M. (2018). ¿Qué es la innovación en educación superior? Reflexiones académicas sobre la innovación educativa. En M. Sánchez, y J. Escamilla, *Perspectivas de la innovación educativa en las universidades de México: experiencias y reflexiones de la RIE 360* (pp. 19-41). Imagia Comunicación. https://www.dee.cuaieed.unam.mx/?sdm_process_download=1&download_id=3883
- Servita, H., y Sánchez, H. (2020). Unidad curricular Introducción a la Programación situada en el enfoque por competencias con apoyo de las TIC. *Revista Conocimiento*

- Libre y Licenciamiento (Clic)*, 22(11), 158-173. <https://convite.cenditel.gob.ve/revistaclac/index.php/revistaclac/article/view/1018>
- Sharma, M., Biros, D., Ayyalasomayajula, S., y Dalal, N. (2020). Teaching programming to the post-millennial generation: Pedagogic considerations for an IS course. *Journal of Information Systems Education*, 31(2), 96-105. <https://aisel.aisnet.org/jise/vol31/iss2/2/>
- Silva, J. P., y Silveira, I. F. (2020). A systematic review on open educational games for programming learning and teaching. *International Journal of Emerging Technologies in Learning (IJET)*, 15(09), 156-172. <https://doi.org/10.3991/ijet.v15i09.12437>
- Silva, J. P., Silveira, I. F., Kamimura, L., y Barboza, A. T. (2020). *Turing Project: An open educational game to teach and learn programming logic*. 15th Iberian Conference on Information Systems and Technologies (CISTI). IEEE; Sevilla, España. <https://ieeexplore.ieee.org/document/9141122>
- Solarte, O., y Machuca, L. E. (2019). Fostering motivation and improving student performance in an introductory programming course: An integrated teaching approach. *Revista ELA*, 16(31), 65-76. <https://doi.org/10.24050/reia.v16i31.1230>
- Susanti, W., Jama, J., Dochi, K., y Nasution, T. (2021). An overview of the teaching and learning process basic programming in algorithm and programming courses. *Turkish Journal of Computer and Mathematics Education*, 12(2), 2934-2944. <https://doi.org/10.17762/turcomat.v12i2.2332>
- Tobón, S. (2013). *Formación integral y competencias. Pensamiento complejo, currículo, didáctica y evaluación* (4a. ed.). ECOE. <https://cife.edu.mx/recursos/2019/12/04/formacion-integral-y-competencias-pensamiento-complejo-curriculo-didactica-y-evaluacion/>
- Tripathi, A., Bharti, K. K., y Ghosh, M. (2019). A study on characterizing the ecosystem of monetizing video spams on YouTube platform. En *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services* (pp. 222-231). ACM. <https://doi.org/10.1145/3366030.3366078>
- Vasilopoulos, I. V., y van Schaik, P. (2018). Koios: design, development, and evaluation of an educational visual tool for Greek novice programmers. *Journal of Educational Computing Research*, 57(5), 1-33. <https://doi.org/10.1177/0735633118781776>
- Viola, J. B., y Gómez, D. T. (2019). Reestructuración pedagógica de la asignatura lógica y algoritmia para el mejoramiento de la enseñanza en los programas de ingeniería de la Universidad Pontificia Bolivariana - Seccional Bucaramanga. *Revista Educación en Ingeniería*, 14(27), 41-47. <https://educacioneningenieria.org/index.php/edi/article/view/940>
- Yi, S., y Lee, Y. (2018). An educational system design to support learning transfer from block-based programming language to text-based programming language. *International Journal on Advanced Science, Engineering and Information Technology*, 8(4-2), 1571-1576. http://ijaseit.insightsociety.org/index.php?option=com_content&view=article&id=9&Itemid=1&article_id=5735
- Yücel, K. (2018). *Developing instructional strategies and recommendations from an introductory programming course in higher education* [Tesis de doctorado]. Middle East Technical University. <https://open.metu.edu.tr/handle/11511/27789>
- Zang, L., Li, B., Zhang, Q., y Hsiao, I. H. (2020). Does a distributed practice strategy for multiple choice questions help novices learn programming? *International Journal of Emerging Technologies in Learning*, 15(18), 234-250. <https://doi.org/10.3991/ijet.v15i18.10567>
- Zhao, Z., Yoo, H., Jo, J., Hong, S., Ryu, S., y Choi, S. (2015). The research trends and comparing about the Big Data between Korea and China using text mining. En *Proceedings of the 2015 International Conference on Big Data Applications and Services* (pp. 121-127). ACM. <https://doi.org/10.1145/2837060.2837078>

Cómo citar este artículo:

Arellano Pimentel, J. J., Solar González, R. y Armería Zavala, L. (2024). Estrategias y recursos didácticos utilizados para aprender programación estructurada. Una revisión sistemática. *IE Revista de Investigación Educativa de la REDIECH*, 15, e1872. https://doi.org/10.33010/ie_rie_rediech.v15i0.1872



Todos los contenidos de *IE Revista de Investigación Educativa de la REDIECH* se publican bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional, y pueden ser usados gratuitamente para fines no comerciales, dando los créditos a los autores y a la revista, como lo establece la licencia.