



Enseñanza-aprendizaje de programación de computadoras: avances en la última década

Teaching-Learning of Computer Programming: Advances in the Last Decade

Ensino-aprendizagem de programação de computadores: avanços na última década

Ronald-Eduardo Ibarra-Zapata¹

José-Orlando Castillo-Cornelio²

Pedro-Crisolo Trujillo-Natividad³

Christian García-Villegas⁴

Rannoverng Yanac-Montesino⁵

Brian Pando⁶

Recibido: marzo de 2021

Aceptado: agosto de 2021

Para citar este artículo: Ibarra-Zapata, R. E., Castillo-Cornelio, J. O., Trujillo-Natividad, P. C., García-Villegas, C., Yanac-Montesino, R. y Pando, B. (2021). Enseñanza-aprendizaje de programación de computadoras: avances en la última década. *Revista Científica*, 42(3), 290-303. <https://doi.org/10.14483/23448350.18339>

Resumen

La programación de computadoras es una actividad muy demandada a nivel mundial en los últimos años como parte del desarrollo de software. La formación en esta área se da desde los niveles básicos de la educación. Según las publicaciones revisadas existen muchos trabajos a nivel básico y superior, como técnicas, metodologías, herramientas para apoyar el proceso de enseñanza-aprendizaje. A pesar de esto es necesario continuar con los aportes al tema. El objetivo de este trabajo es identificar los artefactos propuestos para

la enseñanza-aprendizaje de programación de computadoras mediante un mapeo sistemático de literatura durante el periodo 2010-2020. Los resultados muestran la existencia de esfuerzos por proponer y validar metodologías y estrategias para la enseñanza a nivel de cursos, dejando abiertas oportunidades de investigación sobre guías, herramientas y métodos de evaluación para acompañar este proceso.

Palabras clave: aprendizaje de programación; codificación de software; enseñanza de programación; programación de software.

1. Universidad Nacional Agraria de la Selva, Tingo María, Perú. ronald.ibarra@unas.edu.pe
2. Universidad Nacional Agraria de la Selva, Tingo María, Perú. joseo.castillo@unas.edu.pe
3. Universidad Nacional Agraria de la Selva, Tingo María, Perú. pedro.trujillo@unas.edu.pe
4. Universidad Nacional Agraria de la Selva, Tingo María, Perú. christian.villegas@unas.edu.pe
5. Universidad Nacional Agraria de la Selva, Tingo María, Perú. rannoverng.yanac@unas.edu.pe
6. Universidad Nacional Agraria de la Selva, Tingo María, Perú. brian.pando@unas.edu.pe

Abstract

In recent years, computer programming has become a highly demanded activity worldwide as part of software development. Training in this area is given from the basic levels of education. According to the reviewed publications, there are many basic and higher-level works involving techniques, methodologies, and tools to support the learning process. Despite this, it is necessary to continue with the contributions to the subject. The purpose of this work is to identify the artifacts proposed for the teaching-learning of computer programming through a systematic mapping of literature during the period 2010-2020. The results show the existence of efforts to propose and validate methodologies and strategies for teaching at the course level, leaving open research opportunities on evaluation guides, tools, and methods to accompany this process.

Keywords: learning software programming; software codification; software programming; teaching software programming.

Resumo

A programação de computadores é uma atividade altamente exigida em todo o mundo nos últimos anos como parte do desenvolvimento de software. A formação nesta área é ministrada desde os níveis básicos da educação. De acordo com as publicações revisadas, existem muitos trabalhos de nível básico e superior, bem como técnicas, metodologias, ferramentas de apoio ao processo de ensino-aprendizagem. Apesar disso, é preciso dar continuidade às contribuições ao tema. O objetivo deste trabalho foi identificar os artefatos propostos para o ensino-aprendizagem da programação de computadores por meio de um mapeamento sistemático da literatura durante o período de 2010-2020. Os resultados mostram a existência de esforços para propor e validar metodologias e estratégias de ensino ao nível do curso, deixando em aberto as oportunidades de investigação sobre guias de avaliação, ferramentas e métodos para acompanhar este processo.

Palavras-chaves: aprendizagem de programação; codificação de software; ensino de programação; programação de software.

Introducción

Hoy en día los programas de computadora son usados constantemente en múltiples dispositivos como PC, smartphones, tabletas, robots y muchos otros dispositivos, por lo que los programas de computadora se han convertido en un elemento muy importante en la vida del ser humano (Mu-tiawani et al., 2017). Por tanto, se debe contar con profesionales que sepan programar para satisfacer las necesidades de programas de computadora en las distintas áreas en las que se desenvuelve el ser humano.

La programación de computadoras, en general, es difícil y requiere una dedicación importante de tiempo para su aprendizaje (Watanabe et al., 2001). El aprendizaje del diseño y programación orientada a objetos es desafiante para principiantes, en especial para estudiantes que ya están familiarizados con el paradigma de los lenguajes procedimentales, por lo que es una tarea difícil pasar de un paradigma a otro (Yulia y Adipranata, 2010).

El propósito de la enseñanza de la programación además de adquirir habilidades en el uso de herramientas también es el de obtener experiencia en la solución de problemas durante el proceso de programación (Watanabe et al., 2001).

Uno de los principales problemas en el aprendizaje de la programación es la habilidad para resolver problemas y el conocimiento de matemáticas, añadiéndose los desafíos de (i) mantener la motivación y el compromiso de los estudiantes en el proceso de aprendizaje, y (ii) encontrar métodos y herramientas adecuados para la enseñanza en las instituciones educativas (Pessoa, Lisboa y Pontual, 2019).

Se han realizado múltiples estudios secundarios que muestran el esfuerzo por reconocer (Saito et al., 2017) y estudiar la enseñanza-aprendizaje de programación en la educación básica (Borges et al., 2018; Szabo et al., 2019; Vinuesa, Córdova y Rodas, 2019) y superior (Lara, Neri y Cota, 2019; Pessoa, Lisboa y Pontual, 2019). En

estos estudios se muestra una diversidad de técnicas y herramientas, una de las más resaltantes es la creación de juegos (Abbasi, Kazi y Khowaja, 2018) para apoyar este proceso. Sin embargo, existe una gran necesidad de continuar buscando alternativas que permitan: (i) fomentar diálogo entre comunidades de educación básica y superior, (ii) clarificar el concepto de resolver problemas de programación, (iii) entender el no uso de los métodos de enseñanza prometedores, (iv) poner al descubierto herramientas para esta labor (Pessoa, Lisboa y Pontual, 2019) y que se ajusten al contexto latinoamericano.

Por todo lo expuesto, es importante hacer un mapeo tanto en educación superior como en el sistema escolar porque la realidad del Perú y Latinoamérica es distinta a la de EE.UU. y otros países que imparten la programación y el pensamiento computacional en los contenidos curriculares de los niveles de educación básica. Por lo que en el contexto de varios países de Latinoamérica puede ser importante conocer las estrategias en todos los niveles de educación para diseñar estrategias que ayuden a cerrar las brechas en la enseñanza-aprendizaje de programación.

En la primera parte se presentan los conceptos de enseñanza-aprendizaje en la programación de computadoras. En la segunda parte se explica la metodología utilizada con la técnica PICO (Petersen, Vakkalanka y Kuzniarz, 2015). En la tercera parte se presenta el análisis y las discusiones de la literatura revisada. Por último, las conclusiones y los trabajos futuros.

Metodología

Se ha realizado un mapeo sistemático de literatura que cubre estudios primarios orientados a contribuir en la mejora de la enseñanza y el aprendizaje de la programación, publicados en el periodo 2010-2020 y teniendo en cuenta la metodología de Petersen, Vakkalanka y Kuzniarz (2015). Este trabajo tiene tres preguntas de investigación generales y diez preguntas específicas.

RQ1. ¿Cuál es la orientación de los artefactos propuestos para mejorar la enseñanza-aprendizaje de programación?

- RQ1.1 ¿Qué tipos de artefactos se han propuesto para mejorar la enseñanza-aprendizaje de programación?: herramientas, métodos, técnicas, guías, enfoque.
- RQ1.2 ¿Qué estrategias didácticas se emplean para la enseñanza-aprendizaje de programación?
- RQ1.3 ¿Qué tipo de herramientas de software se emplean para la enseñanza-aprendizaje de programación?: aplicaciones web, escritorio, móvil.
- RQ1.4 ¿A qué enfoque de programación están orientadas las investigaciones?
- RQ1.5 ¿A qué lenguajes de programación se orientan los artefactos a los que hacen referencia los estudios primarios?

RQ2. ¿Cuál es el alcance pedagógico de los artefactos a los que hacen referencia los estudios primarios?

- RQ2.1 ¿Cuál es el alcance a nivel curricular de los artefactos a los que hacen referencia los estudios primarios?: todo el currículo, un curso, una sesión.
- RQ2.2 ¿A qué nivel de educación están orientados los artefactos a los que hacen referencia los estudios primarios?
- RQ2.3 ¿A qué tipo de entorno de enseñanza-aprendizaje se orientan los artefactos a los que hacen referencia los estudios primarios?: presencial, b-learning o e-learning.

RQ3. ¿Cuál es la metodología de investigación que han seguido los estudios primarios que se han propuesto para mejorar la enseñanza-aprendizaje de programación?

- RQ3.1 ¿De qué manera se han evaluado los artefactos a los que hacen referencia los estudios primarios?: experimentos, casos de estudios, encuestas, etc.

- RQ3.2 ¿Cuál es el alcance a nivel de investigación de los artefactos a los que hacen referencia los estudios primarios?: propuesta, implementación, evaluación.

Las bases de datos utilizadas en esta investigación fueron IEEE-Xplore, ACM Digital Library, ScienceDirect y Scopus. Se determinó la cadena de búsqueda basada en el método PICOC, del que solo se utilizó PIC, adecuándolo al contexto de la investigación.

Tabla I. Cadena de búsqueda según el método PICO

| | |
|--------------|--|
| Population | <i>“teaching programming” OR “programming teaching” OR “programming learning”</i> |
| Intervention | <i>guideline OR device OR tool OR method OR “didactic strategy” OR approach OR “evaluation method”</i> |
| Comparison | - |
| Outcome | - |
| Context | <i>language AND computer</i> |

Fuente: elaboración propia.

La cadena de búsqueda genérica, que fue afinada durante tres iteraciones en las bases de datos para mejorar la precisión de los resultados, es la siguiente:

(guideline OR device OR tool OR method OR “didactic strategy” OR approach OR “evaluation

method”) AND (“teaching programming” OR “programming teaching” OR “programming learning”) AND language AND computer.

Los resultados fueron filtrados mediante los criterios de inclusión y exclusión siguientes: (ci-1) artículos de tipo estudio primario, (ci-2) artículos en idioma inglés, (ci-3) artículos que presentan herramienta, enfoque o guía para la enseñanza-aprendizaje de la programación, y (ci-4) artículos que indiquen haber realizado algún tipo de aplicación práctica.

La cadena de búsqueda fue adaptada a la sintaxis de cada base de datos y los resultados fueron extraídos en una hoja de cálculo para la caracterización según las preguntas de investigación, a la vez fueron filtrados por etapas, aplicando los criterios de inclusión y exclusión.

La búsqueda de artículos relevantes para la revisión siguió el proceso descrito en la [figura 1](#). La búsqueda automática en las cuatro bases de datos arrojó 1984 artículos potencialmente relevantes. En la segunda etapa del proceso de selección, al eliminar artículos duplicados se conservaron 1428. En la tercera fase se realizó una lectura rápida por pares (lector y revisor) del artículo, y así se identificaron aquellos artículos que cumplían los criterios de inclusión y exclusión, quedando 433. En la cuarta y última etapa se realizó la lectura completa y por pares y se procedió a categorizarlos, quedando finalmente 268 artículos categorizados.

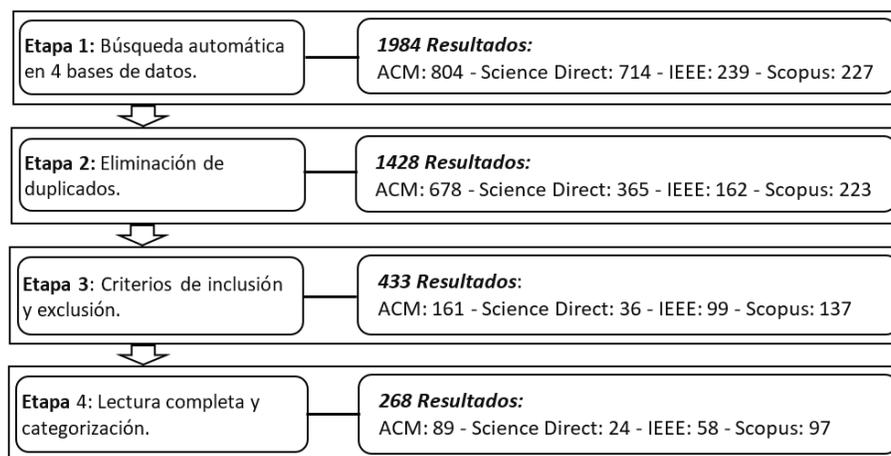


Figura 1. Proceso de búsqueda y resultados en cada etapa.

Fuente: elaboración propia

No se utilizaron preguntas de calidad, ya que es un mapeo sistemático del que no se debía descartar ningún artículo luego de los criterios de inclusión y exclusión. El resultado de cada etapa se puede revisar en: <https://github.com/jcastillo85/TeachSoftFramework>.

Resultados

Las investigaciones sobre la enseñanza de programación han tomado fuerza en los últimos cinco años. En la figura 2 se muestra que durante los años 2010 al 2015 las publicaciones se han mantenido en promedio entre 10 y 15, pero se ve una tendencia a crecer a partir del 2015, llegando a un pico de más de 40 artículos publicados en el año 2019. Es de interés global la necesidad de proponer y buscar soluciones para mejorar esta área, sin embargo, Asia tiene un porcentaje superior a las demás regiones.

RQ1. Orientación de los artefactos encontrados

Las investigaciones revisadas muestran que la mayoría de los esfuerzos se ha centrado en crear y validar propuestas de estrategias y metodologías para la enseñanza de programación (Figura 3).

No obstante, hay escasos trabajos encontrados que presenten guías, métodos e instrumentos de evaluación para la enseñanza-aprendizaje, lo que muestra oportunidades de investigaciones para estos componentes. También se aprecian pocos

esfuerzos de estudios del proceso de enseñanza y su interpretación e impacto a través de una malla curricular.

En la Figura 4 se muestra que el 46% de las investigaciones encontradas ha realizado y evaluado los artefactos propuestos, el 21% aplicó en algún contexto y evaluó los resultados, un 20% solo evaluación y un 10% solo propuesta. Pero son pocos los artefactos de tipo herramientas para ayudar al proceso de enseñanza. Esto permite notar una necesidad por hacer investigaciones que generen herramientas que ayuden al proceso de enseñanza de programación, así como para la evaluación de este tipo de enseñanza.

Las estrategias didácticas aplicadas son por mucho las de aprendizaje basado en resolución de problemas, seguido de aprendizaje basado en software y aprendizaje basado en juegos. Por último, se han presentado otras propuestas que no han sido muy exploradas.

Según la Figura 5, el aprendizaje basado en resolución de problemas ha mantenido el interés a largo de los años en comparación con los otros enfoques; sin embargo, el aprendizaje basado en software y basado en juegos también han mantenido un ritmo constante, notándose el interés por estas dos últimas estrategias.

El 55% de las investigaciones encontradas se ha enfocado en la enseñanza-aprendizaje de programación utilizando aplicaciones web, seguido de un 40% de aplicaciones de escritorio (Figura 6). Aún no se han desarrollado muchas aplicaciones móviles; sin embargo, muchas de las aplicaciones

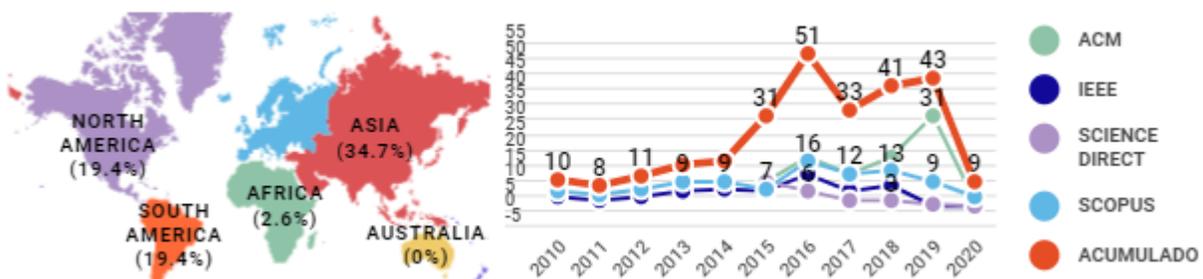


Figura 2. Tendencia de publicaciones en la enseñanza-aprendizaje de programación.

Fuente: elaboración propia

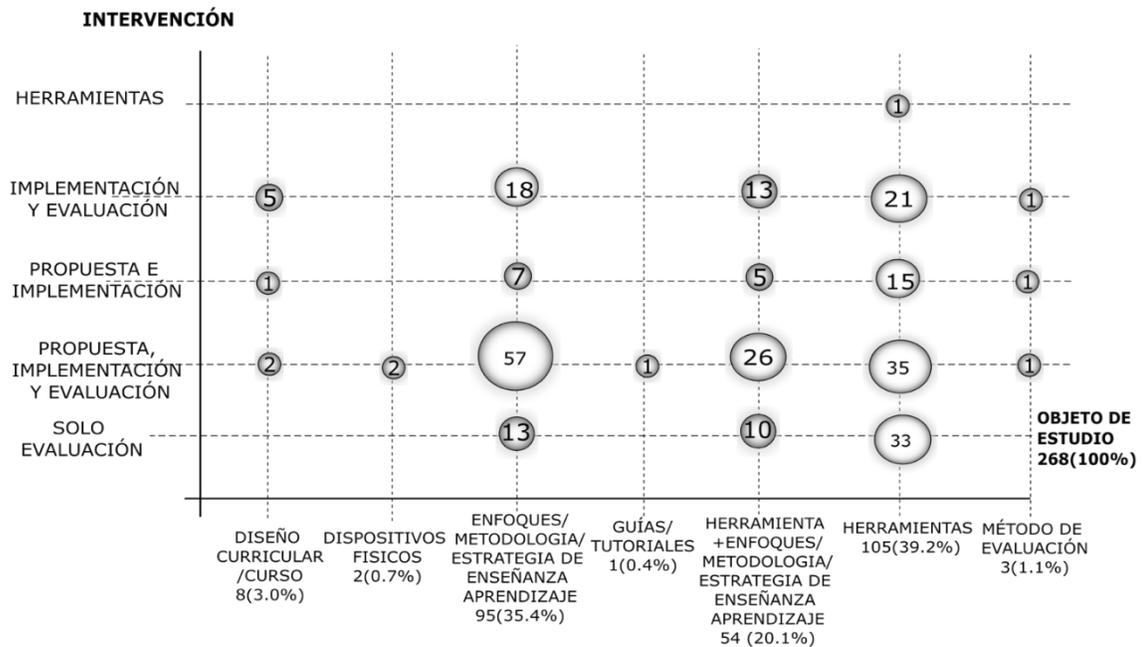


Figura 3. Gráfico de burbuja de los artefactos encontrados.

Fuente: elaboración propia

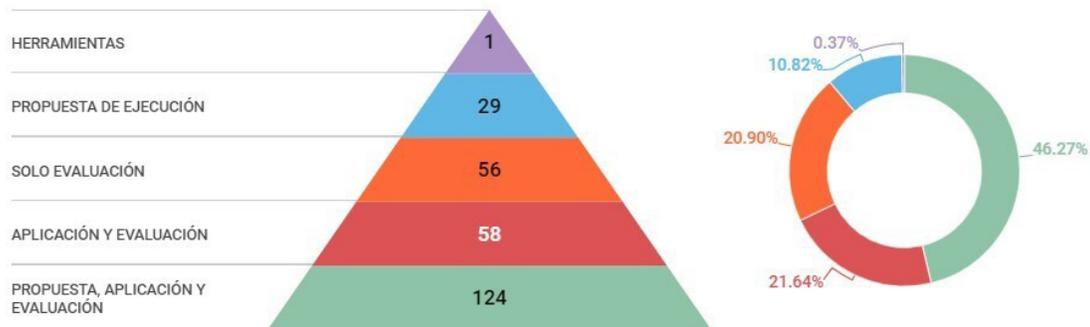


Figura 4. RQ1-1: Tipos de artefactos para mejorar la enseñanza-aprendizaje de programación.

Fuente: elaboración propia

web propuestas podrían haberse desarrollado para ser de diseño responsivo. Las aplicaciones de escritorio para la enseñanza de programación han tenido un ritmo relativamente creciente a lo largo de los años, pero las aplicaciones web empezaron a superar ligeramente a las de escritorio a partir del año 2015.

Según la Figura 7, los enfoques de programación más utilizados han sido la estructurada y la

orientada a objetos, siendo casi el 90% del total de artículos encontrados. Muchos de los artículos se centran en personas que recién inician en la programación, usando programación estructurada para este fin, por ello la gran cantidad de artículos encontrados se basa en la programación estructurada. Estos dos enfoques se han mantenido como los más usados a lo largo de los años.

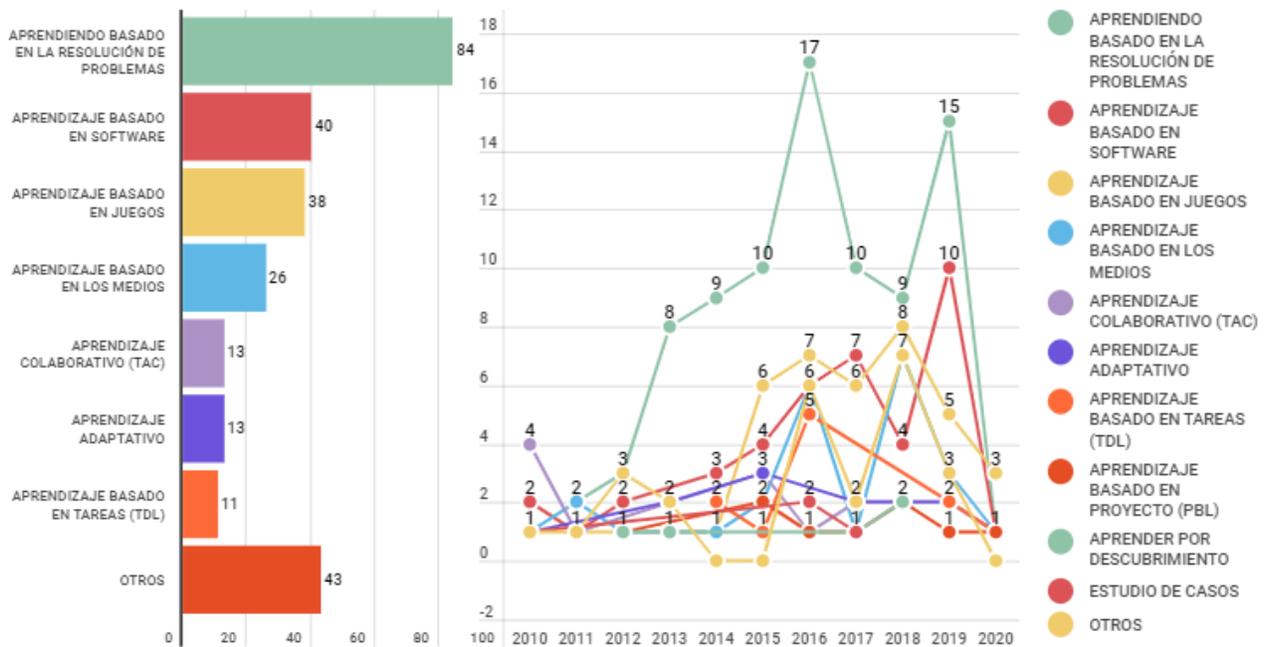


Figura 5. RQ1-2: Estrategias didácticas utilizadas para la enseñanza-aprendizaje de programación.

Fuente: elaboración propia

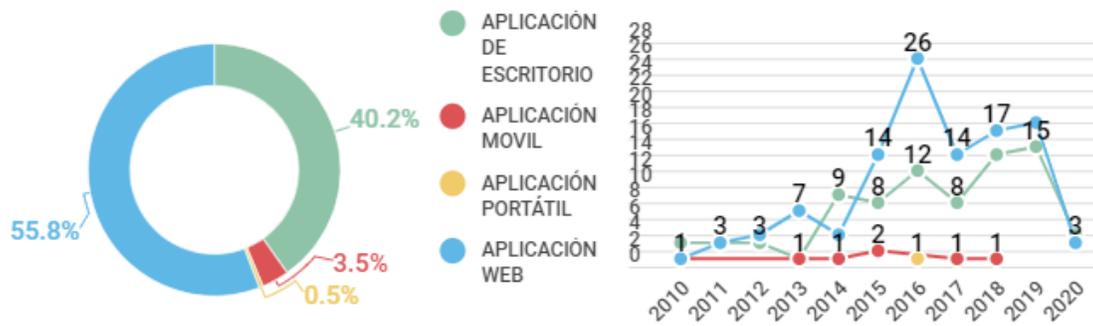


Figura 6. RQ1-3: Tipos de entorno de software para la enseñanza-aprendizaje de programación.

Fuente: elaboración propia

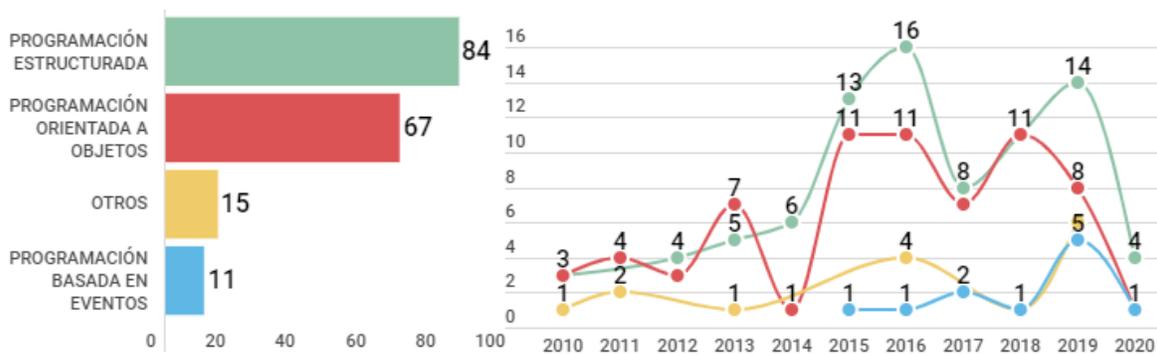


Figura 7. RQ1-4: Enfoques encontrados para la enseñanza-aprendizaje de programación.

Fuente: elaboración propia

Java es uno de los lenguajes de programación más utilizados para la enseñanza de programación, así como C/C++/C#, a lo largo de la última década. La Figura 8 muestra que hay variedad en los lenguajes que se han usado para este fin, pero se debe resaltar que muchas propuestas son agnósticas al lenguaje de programación; es decir, la enseñanza de programación no siempre se debe centrar en un lenguaje de programación. Otro punto para resaltar es la presencia de Scratch como un lenguaje bastante usado para la enseñanza de programación, sobre todo a nivel escolar.

RQ2. Alcances pedagógicos de los artefactos encontrados

Según la Figura 9, el 77% de las investigaciones se ha centrado en el tratamiento y la evaluación durante el periodo de duración de un curso, un 14% en una o algunas de las sesiones del curso. Solo el 8% ha tenido en cuenta a todo el currículo, haciendo notar la escasez de investigaciones en este último alcance. A lo largo de los años la aplicación en un curso ha sido el objetivo de las investigaciones encontradas.

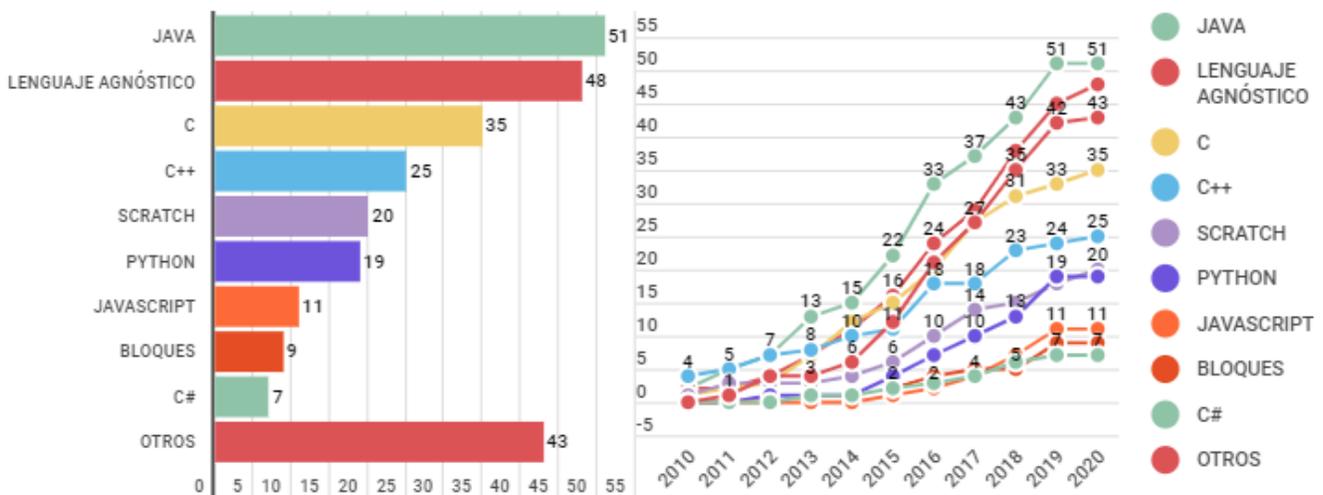


Figura 8. RQ1-5 Lenguajes de programación a los que se orientan los artefactos encontrados.

Fuente: elaboración propia

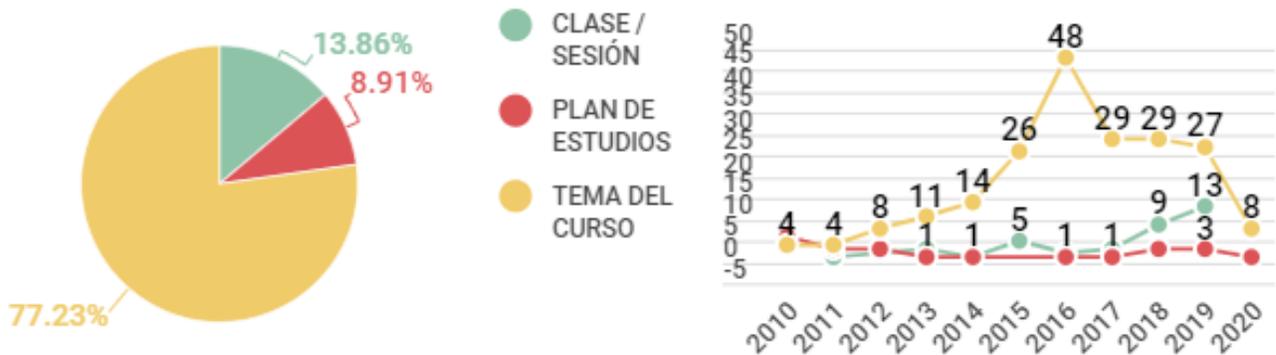


Figura 9. RQ2-1: Alcance a nivel curricular de los artefactos encontrados.

Fuente: elaboración propia

De acuerdo con la [Figura 10](#), casi el 70% de las investigaciones encontradas se ha centrado en la educación a nivel universitario, sobre todo en la región asiática, con menor intensidad a nivel secundaria y un poco menos en primaria. Sin embargo, se evidencia el esfuerzo que se ha realizado en estos dos últimos niveles, indicando que se impulsa el aprendizaje de programación desde la niñez a nivel mundial, con un ligero esfuerzo mayor en Europa. Asimismo, cabe resaltar que alrededor del 3% de las investigaciones se han centrado en el nivel K-12, que va desde la enseñanza en jardín inicial hasta el nivel secundario y es adoptado por algunos países como Estados Unidos, Inglaterra, entre otros.

Según la [Figura 11](#), el 47% de investigaciones se ha enfocado en el aprendizaje presencial, la otra mitad se ha proyectado al aprendizaje e-learning y b-learning. Solo un 14% ha hecho investigaciones exclusivas para e-learning. Esto hace notar la versatilidad de las herramientas construidas para aprendizaje tanto presencial como e-learning. A lo largo de los años las investigaciones se han enfocado en lo presencial, pero en el último año las investigaciones sobre b-learning han superado los esfuerzos del aprendizaje presencial, indicando la necesidad de impulsar esta área.

Según la [Figura 12](#), la mayoría de los estudios se ha centrado en la educación superior tanto en

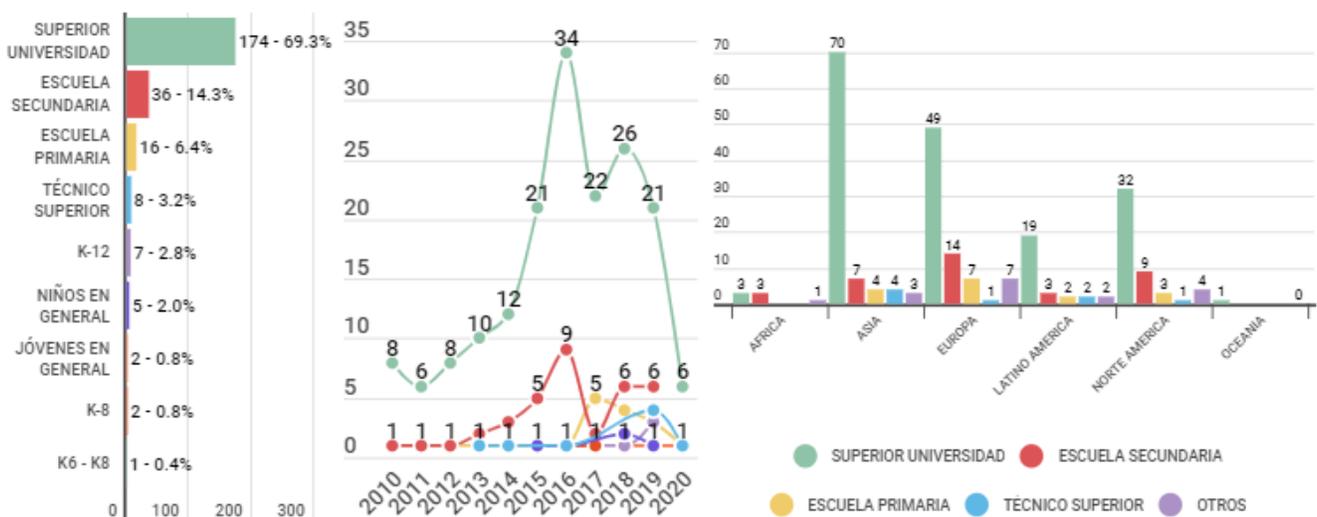


Figura 10. RQ2-2: Enseñanza-aprendizaje de programación por niveles de educación.

Fuente: elaboración propia

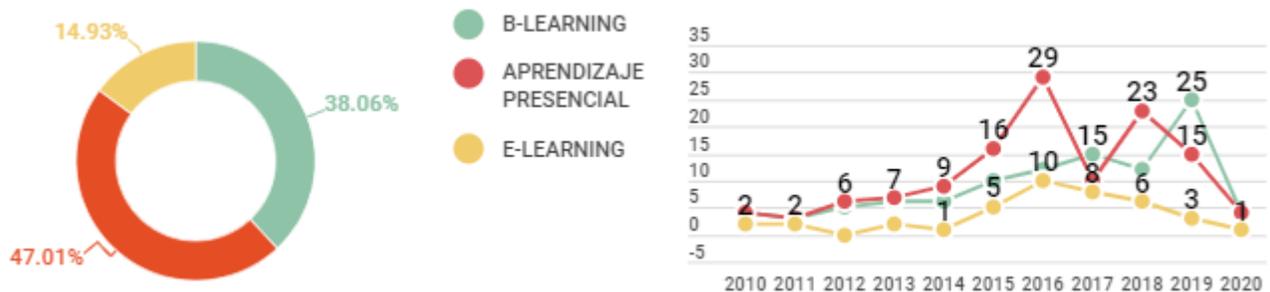


Figura 11. RQ2-3: Tipos de entorno de aprendizaje de los artefactos encontrados.

Fuente: elaboración propia

entornos presenciales como b-learning o e-learning. Aún no hay muchos esfuerzos a nivel escolar. En todos los niveles se encontraron más trabajos de aplicación durante un curso, en su mayoría aplicados a nivel superior universitario.

RQ3. Metodología de los artefactos encontrados

En la Figura 13 se aprecia que el 46% de los artículos encontrados ha realizado propuestas e implementación de evaluación del artefacto propuesto, un 10% ha presentado una propuesta y la ha implementado, pero sin presentar evidencias de su evaluación. Un 22% de trabajos se implementa mediante casos o experimentos, además se evalúan los resultados obtenidos. Un 21% se ha dedicado a evaluar artefactos existentes con resultados favorables. El ritmo de estos tipos de investigación se ha mantenido bajo el mismo patrón a lo largo de los años.

El 50% de las investigaciones revisadas se ha centrado en hacer experimentos académicos, lo cual es lógico debido a que los artículos se basan en la enseñanza de programación a nivel académico. Sin embargo, según la Figura 14 también se han encontrado trabajos de tipo casos de estudio con un 18%, así como pruebas piloto, cuestionarios y otros en menor cantidad.

Según la Figura 15 se ha realizado un gran número de experimentos académicos y casos de estudios aplicados en la educación superior. También existen esfuerzos realizados en los otros niveles de educación. Sin embargo, es notoria la necesidad de explorar aún más la educación a nivel de escuelas y explorar más artefactos a nivel de educación superior.

Respecto a la modalidad de enseñanza se tiene que 47% de la enseñanza de programación es presencial y 15% a distancia o e-learning, lo cual difiere de lo encontrado por Borges et al. (2018), quienes determinaron que el 78% de la enseñanza de programación era presencial y 2% a distancia.



Figura 12. RQ2: Alcance pedagógico de los artefactos encontrados.

Fuente: elaboración propia



Figura 13. RQ3-1: Alcance de los artefactos encontrados.

Fuente: elaboración propia

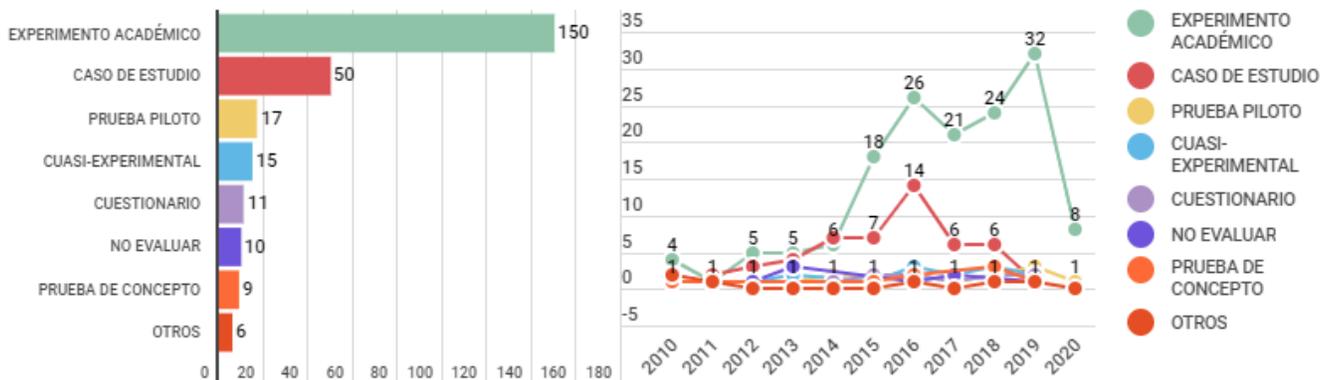


Figura 14. RQ3-2 Tipos de investigación de los trabajos encontrados.

Fuente: elaboración propia

Sobre el nivel educativo en el que se enseña programación, se encontró que más del 50% de los estudios se centró en la enseñanza de programación a nivel universitario, seguido de la enseñanza a nivel secundario, lo cual concuerda con lo encontrado por [Borges et al. \(2018\)](#) y [Pessoa, Lisboa y Pontual \(2019\)](#), quienes encontraron que la enseñanza de programación se enfoca mayormente en educación superior y que además se han incrementado los proyectos de enseñanza de programación para niños y adolescentes.

La estrategia didáctica que se encontró en un porcentaje del 14% es el de aprendizaje basado en juegos, lo que también había identificado [Abbas, Kazi y Khowaja \(2018\)](#) al analizar estudios que

buscan aportar en el aprendizaje de la programación orientada a objetos mediante “juegos serios”.

Además, [Saito et al. \(2017\)](#) encontraron que el software de juegos es muy adecuado para el aprendizaje introductorio de programación, identificando 24 herramientas de software de juegos para enseñar programación, dentro de estas herramientas estaban Alice y Scratch, herramientas que también se identificaron en el presente trabajo. Respecto a Scratch el 8.7% de los artículos estudiados muestra a Scratch como la herramienta utilizada para la enseñanza de programación. Respecto a Alice, [Costa y Miranda \(2017\)](#) encontraron que la herramienta tuvo un impacto positivo en el rendimiento de los estudiantes y [Vinueza,](#)

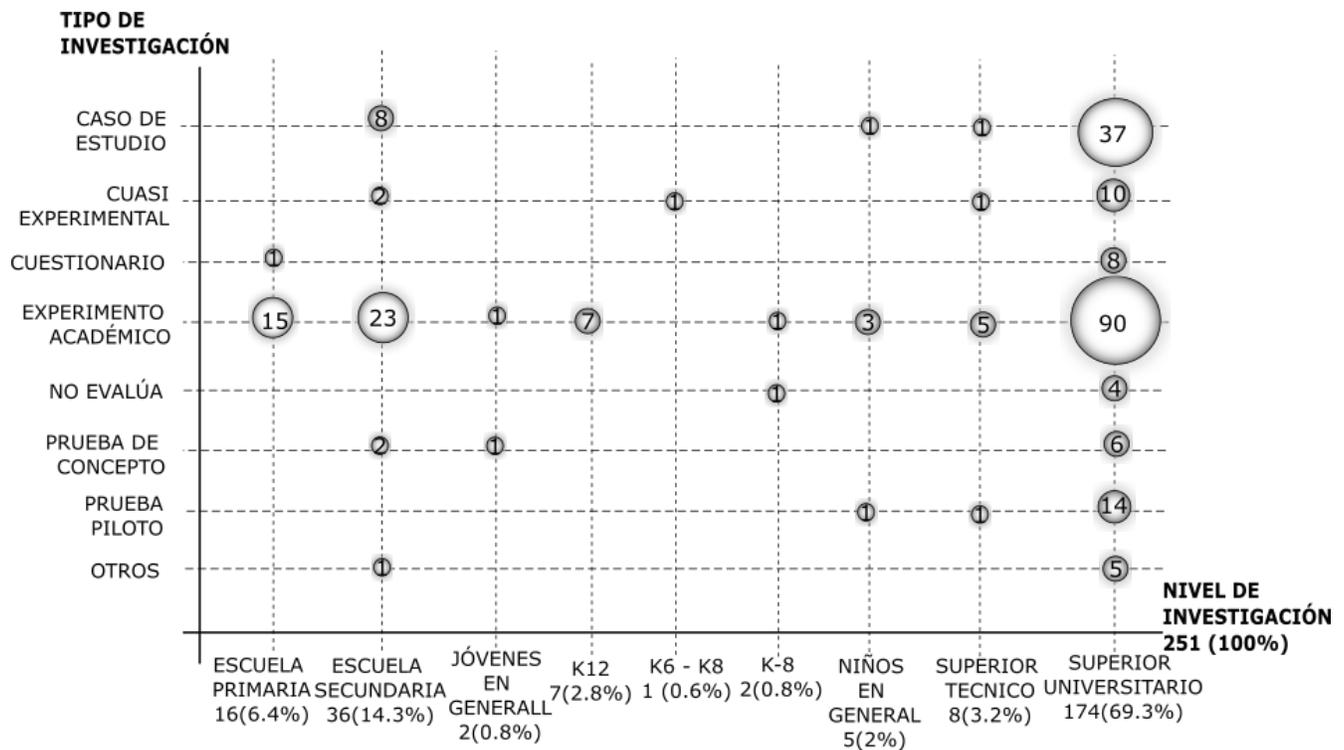


Figura 15. RQ3: Tipos de investigación sobre los artefactos encontrados.

Fuente: elaboración propia

Córdova y Rodas (2019) identificó que Alice es efectivo para el aprendizaje de introducción a la programación, pero ambos coinciden en la existencia de pocas investigaciones en Latinoamérica, lo que se evidencia en la presente investigación con dos estudios (que representa el 0.7%) que reportan a Alice como herramienta para enseñar programación.

En esta investigación se encontró que aproximadamente el 10% de los estudios sobre enseñanza de programación está enfocado a niños de nivel K-12, lo que concuerda con Szabo et al. (2019), quien exploró el desarrollo de la enseñanza de programación introductoria en el nivel K-12, en el periodo 2003 a 2017, identificando un fuerte crecimiento de la enseñanza de programación en este grupo de edad.

Lara, Neri y Cota (2019) encontraron que los lenguajes de programación más citados para enseñanza de programación paralela son C (29%) y C++ (23%), lo que difiere con lo encontrado en la

presente investigación, donde se tienen porcentajes de 13.9% y 9.8% en investigaciones relacionadas con la enseñanza de programación usando C y C++ respectivamente. Lara, Neri y Cota (2019) también encontraron que la educación a distancia para enseñar programación era reciente y por esto solamente el 8% de los estudios utiliza la educación a distancia para la enseñanza de la programación paralela, mientras que en la presente investigación se obtuvo que el 15% de investigaciones se orientan al entorno de enseñanza a distancia o e-learning.

Limitaciones del estudio

Se han utilizado cuatro bases de datos conocidas, sin embargo, existen otras bases de datos donde se podría encontrar artículos que aportan a la investigación. Además, la heterogeneidad de los estudios permite llegar a conclusiones parciales, siendo necesario realizar otros estudios más profundos.

Conclusiones

La investigación en enseñanza de programación de computadoras es un tema que ha empezado a tener fuerza a partir de 2015. A nivel mundial creció el interés por aportar en el tema mediante propuestas y herramientas que sean de utilidad para este fin, con Java como lenguaje predilecto, pero muchas de estas propuestas también son agnósticas al lenguaje de programación. Las investigaciones en su mayoría se han centrado en la enseñanza basada en problemas, basada en software y basada en juegos.

La mayoría de propuestas ha sido a nivel de curso, encontrándose pocas propuestas a nivel de malla curricular. Además, gran parte de los trabajos se centran en la educación superior universitaria y están enfocados en la enseñanza presencial, pero también en el b-learning. Sin embargo, según la apreciación de los autores, en los años sucesivos las investigaciones sobre e-learning deberían aumentar por la necesidad que se ha presentado con la crisis de pandemia por COVID 19 a nivel mundial.

Los artefactos propuestos en su mayoría han sido validados durante la investigación en gran parte por experimentos académicos y casos de estudio, de los que han resultado estrategias y metodologías propuestas para sustentar este proceso de enseñanza. Sin embargo, no se han encontrado muchos aportes sobre guías, herramientas y métodos de evaluación.

Por último, este mapeo sistemático de literatura ha dejado notar la madurez de algunos campos de la enseñanza de programación de computadoras, al tiempo que muestra oportunidades de investigación en otras áreas tales como métodos de evaluación y herramientas. Por lo tanto, como trabajo futuro, se deben proponer guías, herramientas y métodos de evaluación que ayuden al proceso de enseñanza-aprendizaje de programación de computadoras considerando los hallazgos encontrados en este trabajo.

Agradecimientos

Se agradece el apoyo del laboratorio de ingeniería de software LINSOFT FIIS UNAS por los recursos brindados para esta investigación.

Referencias

- Abbasi, S., Kazi, H., Khowaja, K. (2018). A systematic review of learning object oriented programming through serious games and programming approaches. *2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences, ICETAS*, 1-6. <https://doi.org/10.1109/ICETAS.2017.8277894>
- Borges, R. P., Oliveira, P. R., Lima, R. G., de Lima, R. W. (2018). A systematic review of literature on methodologies, practices, and tools for programming teaching. *IEEE Latin America Transactions*, 16(5), 1468-1475. <https://doi.org/10.1109/TLA.2018.8408443>
- Costa, J. M., Miranda, G. L. (2017). Relation between Alice software and programming learning: A systematic review of the literature and meta-analysis. *British Journal of Educational Technology*, 48(6), 1464-1474. <https://doi.org/10.1111/bjet.12496>
- Lara Soares, F. A., Neri Nobre, C., Cota de Freitas, H. (2019). Parallel programming in computing undergraduate courses: A systematic mapping of the literature. *IEEE Latin America Transactions*, 17(8), 1371-1381. <https://doi.org/10.1109/TLA.2019.8932371>
- Mutiawani, V., Juwita, Afidh, R. P. F., Novitasari, D. (2017). Implementing problem-solving method in learning programming application. *2017 International Conference on Electrical Engineering and Informatics (ICELTICS)*, 211-215. <https://doi.org/10.1109/ICELTICS.2017.8253263>
- Pessoa Medeiros, R., Lisboa Ramalho, G., Pontual Falcao, T. (2019). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, 62(2), 77-90. <https://doi.org/10.1109/TE.2018.2864133>

- Petersen, K., Vakkalanka, S., Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64, 1-18. <https://doi.org/10.1016/j.infsof.2015.03.007>
- Saito, D., Sasaki, A., Washizaki, H., Fukazawa, Y., Muto, Y. (2017). Program learning for beginners: Survey and taxonomy of programming learning tools. *Proceedings of the 2017 IEEE 9th International Conference on Engineering Education (ICEED)*, 137-142. <https://doi.org/10.1109/ICEED.2017.8251181>
- Szabo, C., Sheard, J., Luxton-Reilly, A., Simon, Becker, B. A., Ott, L. (2019). Fifteen years of introductory programming in schools: A global overview of K-12 initiatives. *ACM International Conference Proceeding Series*, 1-9. <https://doi.org/10.1145/3364510.3364513>
- Vinueza-Morales, M., Córdova-Morán, J., Rodas-Silva, J. (2019). Using Alice software as a tool for programming learning: A literature review. *17th LACCEI International Multi-Conference for Engineering, Education and Technology*. <https://doi.org/10.18687/LACCEI2019.1.1.161>
- Watanabe, Y., Otobe, Y., Yoshitomi, K., Takahashi, H., Kohiyana, K. (2001). The Effects of Scaffolding-Based Courseware for The Scratch Programming Learning on Student Problem Solving Skill. *IEEE Transactions on Consumer Electronics*, 47(3), 405-411. <https://doi.org/10.1109/30.964127>
- Yulia, Adipranata, R. (2010). Teaching object oriented programming course using cooperative learning method based on game design and visual object oriented environment. *2010 2nd International Conference on Education Technology and Computer*, 2, 1-12. <https://doi.org/10.1109/icetc.2010.5529368>

