

Copyright © 2019

[This article is published under the terms of the Attribution-ShareAlike 4.0 International \(CC BY-SA\)](#)



<https://revistas.udistrital.edu.co/ojs/index.php/Tecnura/issue/view/1075>

DOI: <https://doi.org/10.14483/22487638.16934>

Clasificación del artículo: **Revisión de tema**

Una revisión sistemática sobre aula invertida y aprendizaje colaborativo apoyados en inteligencia artificial para el aprendizaje de programación

A Systematic Review on Flipped Classroom and Collaborative Learning Supported in Artificial Intelligence for Programming Learning

Fecha de recepción: 22 de Noviembre de 2020

Fecha de aceptación: 01 de abril de 2021

Cómo citar: Giovanni-Hidalgo., C.G. Llanos-Mosquera., J.M. y Bucheli-Guerrero., V.A. (2021). *Una revisión sistemática sobre aula invertida y aprendizaje colaborativo apoyados en inteligencia artificial para el aprendizaje de programación. Tecnura*, XX(XX). <https://doi.org/10.14483/22487638.16934>

Carlos Giovanni Hidalgo

Ingeniero de sistemas, magíster en Ingeniería, estudiante de doctorado Universidad del Valle.

Cali, Colombia. Contacto: carlos.hidalgo@correounivalle.edu.co

ORCID: <https://orcid.org/0000-0003-2308-0720>

José Miguel Llanos Mosquera

Ingeniero de sistemas, magíster en Arquitectura de Software, estudiante de doctorado

Universidad del Valle. Cali, Colombia. Contacto: jose.llanos@correounivalle.edu.co

ORCID: <https://orcid.org/0000-0003-4642-2770>

Víctor Andrés Bucheli Guerrero

Ingeniero de sistemas, magíster en Ingeniería y Computación, doctor en Ingeniería. Profesor

asociado Universidad del Valle. Cali, Colombia. Contacto:

victor.bucheli@correounivalle.edu.co

ORCID: <https://orcid.org/0000-0002-0885-8699>

(T2) Resumen

Objetivo: La implementación de estrategias pedagógicas como el aula invertida (AI) y el aprendizaje colaborativo (AC) han contribuido en la enseñanza de programación de computadores. En este sentido, este artículo permite identificar el estado actual de las tecnologías y herramientas basadas en aprendizaje activo y colaborativo, y cómo han apoyado el aprendizaje de la programación, y los aportes que ha generado la inteligencia artificial en este proceso.

Metodología: Se realizó una revisión sistemática bajo la propuesta de Kitchenham para definir las preguntas de investigación, la selección de fuentes de información, el desarrollo de la revisión y el análisis de la información. En el proceso se utilizaron diferentes fuentes de datos y criterios de selección para artículos investigativos y conferencias publicadas entre 2013 y 2020.

Resultados: Se identificó el estado actual del AI y el AC, enfatizando en cómo estas estrategias apoyan el proceso de enseñanza de la programación con métodos educativos y herramientas de *software*. También se identificó el apoyo de la inteligencia artificial en el aprendizaje de la programación a través de diferentes aplicaciones y técnicas computacionales que integran AI y AC.

Conclusiones: Desde la aparición de las herramientas virtuales, la implementación de la inteligencia artificial se ha convertido en una prioridad para la educación virtual, potenciando la forma de entender y comprender las necesidades específicas del estudiante. Las nuevas herramientas y estrategias basadas en inteligencia artificial han contribuido con la gestión y la visualización, donde el docente puede tomar decisiones oportunas del proceso formativo y retroalimentar las actividades desarrolladas en los ambientes de aprendizaje.

Palabras clave: aprendizaje colaborativo, aula invertida, estrategias de aprendizaje, inteligencia artificial, programación de computadores.

ABSTRACT

Objective: Implementing pedagogical strategies such as the Flipped Classroom (FC) and Collaborative Learning (CL) have contributed to the teaching of computer programming. In this sense, this paper allows us to identify the current state-of-the-art techniques based on active and collaborative learning, and how they have supported programming learning, and the contributions that Artificial Intelligence (AI) has generated in this process.

Methodology: Supported by the Kitchenham process, a systematic review was developed, finding the research questions, selecting information sources, the information analysis and processing. Different data sources and selection criteria were used to find research articles and conferences published between 2013 and 2020.

Results: FC and CL current state was identified, emphasizing how these strategies support the teaching process of programming with educational methods and software tools. In addition, it was identified how Artificial Intelligence supports learning programming through different applications and computational techniques that integrate FC and CL.

Conclusions: The novel tools and strategies based on AI have contributed to management and visualization. The teacher can make timely decisions in the training process and provide feedback on the learning environments' activities. Since the appearance of virtual tools, the implementation of Artificial Intelligence has become a basic need for virtual education, enhancing understanding and understanding the specific needs of the student.

Keywords: collaborative learning, flipped classroom, learning strategies, artificial intelligence, computer programming.

(T2) Introducción

Las estrategias pedagógicas cumplen un papel importante en los procesos de enseñanza y aprendizaje, pues ayudan a mejorar la motivación, participación, colaboración e interacción entre estudiantes y profesores (Burch y Melby, 2019; Chinchilla y Gómez, 2020). La implementación de estrategias de aprendizaje puede ser útil en la gestión académica del estudiante, mejorar su rendimiento y aportar en la generación de nuevo conocimiento significativo basado en el autoaprendizaje (Montero y Benavides, 2015). Actualmente dichas estrategias buscan soluciones que ayudan a mejorar las destrezas de los estudiantes para la solución de problemas (Bravo *et al.*, 2013). En este sentido, estrategias como el aula invertida (AI) y el aprendizaje colaborativo (AC) surgen para mejorar hábitos de aprendizaje, pensamiento crítico y la colaboración en el intercambio de conocimiento de los estudiantes (Amresh *et al.*, 2013). El AI se ha enfocado en el aprendizaje mixto, donde el estudiante alcanza sus propias bases a través del autoaprendizaje y en clase resuelve dudas con apoyo del profesor. Por su parte el AC, se enfoca en la construcción de habilidades que le permiten al estudiante tener bases que no se encuentran en la enseñanza tradicional de un curso (Fortanet van Assendelft, 2013).

Implementar estas estrategias pedagógicas puede ser un proceso difícil, sin la ayuda de herramientas y tecnologías que controlen el proceso formativo y evaluativo del estudiante. En este sentido, la implementación de tecnologías y técnicas computacionales de inteligencia artificial han mejorado el intercambio de un espacio real a uno virtual de forma natural, logrando adaptarse a las necesidades específicas de cada estudiante (Mok, 2014; Salazar Ospina *et al.*, 2017). Entre las diferentes aplicaciones de la inteligencia artificial como apoyo al aprendizaje de programación, aparecen sistemas de recomendación, plataformas virtuales

de aprendizaje, soporte de colaboración a través de formación de grupos, identificación de perfiles, chat de soporte automático, evaluación y realimentación de código fuente, toma de decisiones a través de visualizaciones, entre otros (Billis y Cubenas, 2020).

El propósito de este artículo es identificar el estado actual del AI y el AC, las tecnologías, estrategias y herramientas existentes, y el papel de la inteligencia artificial como apoyo para el aprendizaje de los cursos de programación. Así, el documento se ha organizado de la siguiente forma. En la metodología se detallan las preguntas investigativas, las fuentes de información, cadenas de búsqueda, filtros para los datos, análisis y presentación de la información. En “Análisis de información” se describe el proceso utilizado para la identificación y el procesamiento de las referencias. En “Resultados” se presenta el estado actual del AI y AC, y los aportes de la inteligencia artificial a los cursos de programación, por último, se mencionan las tendencias y los logros que se proponen alcanzar los autores.

(T2) Metodología

Siguiendo las fases para realizar una revisión sistemática propuesta en Kitchenham *et al.* (2007), se especifican las preguntas de investigación, selección de fuentes de información y se desarrolla un protocolo de revisión y análisis.

(T3) Preguntas de investigación

Se plantearon preguntas de interés que identifican el estado actual a partir de dos estrategias pedagógicas utilizadas para apoyar el aprendizaje de la programación: ¿Cuál es el estado actual de las tecnologías, estrategias y herramientas basadas en aprendizaje activo y el aprendizaje colaborativo como apoyo al aprendizaje de la programación?; ¿cómo el aula invertida y el aprendizaje colaborativo a través de técnicas computacionales de inteligencia

artificial apoyan los cursos de programación?; ¿cuál es el papel de la inteligencia artificial en el aprendizaje de los cursos de programación?

(T3) Fuentes de información y cadenas de búsqueda

En la tabla 1, se muestran cinco funciones de búsquedas específicas y los registros por cada consulta para las fuentes de datos: Scopus, Web of Science (WoS), ACM Digital Library, IEEE Xplore, ScienceDirect y GitHub. Las funciones de búsqueda se construyeron con palabras claves afines al tema de investigación con ayuda de profesores e investigadores expertos en el tema.

Tabla 1. Funciones de búsqueda, fuentes de datos y registros totales por consulta

	Consulta	Fuente de datos	Registros
1	((("flipped classroom") AND ("programming course")) AND ("artificial intelligence"))	Scopus, WoS, ACM Digital Library, IEEE Xplore y ScienceDirect	98
2	((("collaborative learning") AND ("programming course")) AND ("artificial intelligence"))	Scopus, WoS, ACM Digital Library, IEEE Xplore y ScienceDirect	61
3	((programming courses)+(collaborative learning)+(artificial intelligence) (programming courses)+(flipped classroom)+(artificial intelligence) SentenceYear=FROM*2013)	GitHub	73

Fuente: elaboración propia.

Luego de obtener los registros, se usó la metodología PRISMA (Moher *et al.*, 2010), para filtrar la información con base en cuatro criterios: información incompleta, nombre repetido de publicación, calidad de la conferencia o revista, no DOI, bajo número de citas, estrellas, copias y tenedores. Finalmente se obtuvieron 210 registros: 166 documentos y 44 repositorios.

(T3) Análisis de información de referencias

Para responder a las preguntas planteadas, se procesan las referencias utilizando diferentes implementaciones. Se construye un flujo de trabajo basado en tres procesos (figura 2): i) Las referencias; por un lado, con ayuda de NER-Spacy (Vychegzhanin y Kotelnikov, 2019), se extraen las características de los repositorios: citas, año, autor, copias, tenedores, estrellas y actualizaciones; por otro, se extraen las características de los documentos científicos: año, palabras clave, resumen, DOI, autores y organizaciones. ii) Las características extraídas son procesadas con NER-Spacy, para obtener entidades específicas, por ejemplo: técnicas, estrategias, métodos, tipo de tecnología, tipo de *software* y lenguajes de programación. iii) Se construyeron dos visualizaciones: la primera se construye a partir de la librería D3 (Bostock, s. f.) para visualizar la evolución tecnológica de los proyectos más relevantes, estrategias de aprendizaje, tipos de tecnologías y los lenguajes de programación utilizados en los últimos siete años. La segunda se construye con la herramienta SASR (Guerrero, 2019), para clasificar a través de una ontología de datos que contiene las líneas y sublíneas de las ciencias de computación, tomadas del Sistema de Clasificación por Computadora (CCS) (Association for Computing Machinery [ACM], s. f.).

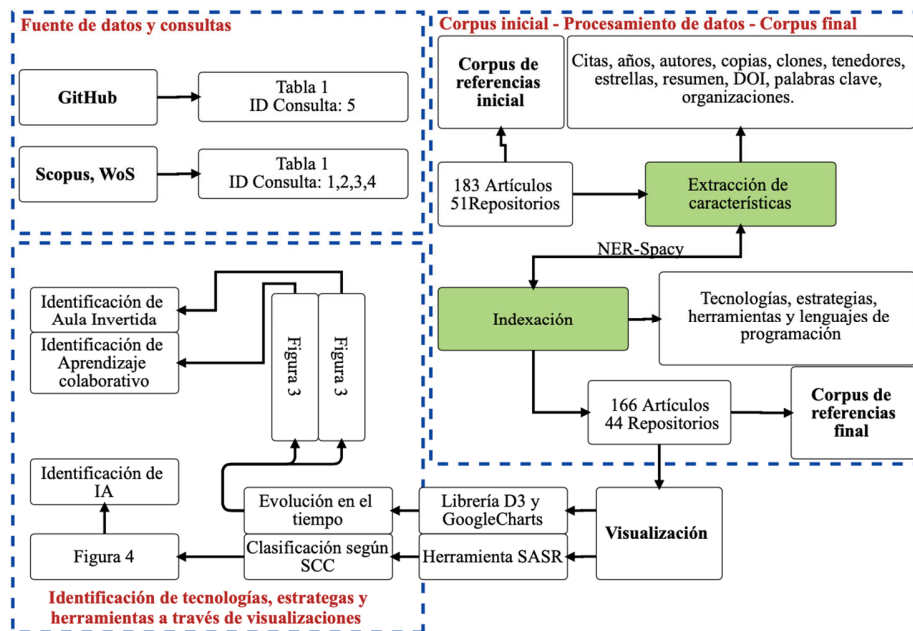


Figura 1. Flujo de trabajo de la identificación, análisis y procesamiento de las referencias

Fuente: elaboración propia.

(T2) Resultados

En esta sección se presentan los hallazgos a partir del año 2013; el análisis muestra el estado actual de las tecnologías, estrategias y herramientas basadas en AI y el AC. También el papel actual y las tendencias de la inteligencia artificial como apoyo en el aprendizaje de los cursos de programación.

(T3) Estado actual del aula invertida y el aprendizaje colaborativo

Para identificar el estado actual de las tecnologías, estrategias y herramientas se realizó la categorización de la información de las referencias, según el año de publicación. En esta sección se responde a la pregunta ¿cuál es el estado actual de las tecnologías, estrategias y

herramientas basadas en aprendizaje activo y el aprendizaje colaborativo como apoyo al aprendizaje de la programación?

(T4) Estado actual del aula invertida

En las figuras 2 y 3, se observan el avance del aula invertida donde encontramos diferentes implementaciones y estrategias que han servido de apoyo en el aprendizaje de la programación.

	2013	2014	2015	2016
Estrategias	Interacciones individuales aprendizaje semipresencial	Videos extra clases orientada a la experimentación	Videos dinámicos	Autocomprobación, Evaluación automática, Tutoría en línea, Ejercicios con retroalimentación
Tecnologías	CMS, LMS, LCMS Procesadores de texto Cloud	Cursos virtuales, MOOC, AVA, OVA	LEGO, REA, OER, Podcast Blended learning	Moocs + Evaluadores automaticos de código fuente
Herramientas	X2 MOODLE Google Drive One Drive PowerPoint	opos FlipClass OpenEdX Coursera Youtube	Flippd Django- ClasseInversee BEN-T1 Mindstorms	flipped-demos Flipped course Flipped Classroom Programming Prezzi

Figura 2. Evolución de tecnologías, estrategias y herramientas basadas en aula invertida (2013-2016)

Fuente: elaboración propia.

Entre los años 2013 y 2016 se utilizaron varias estrategias que aportaron al aula invertida como apoyo al aprendizaje de la programación, el primer aporte se basó en la interacción individual, donde el estudiante adquiere unos conocimientos y desarrolla ejercicios relacionados a código fuente de forma individual, aquí el profesor se enfocó en el análisis y comprensión de los contenidos y la motivación del estudiante por el aprendizaje. Luego se implementó el aprendizaje semipresencial que combina la presencialidad (desarrollo de

actividades en el aula) con el trabajo en línea (uso de internet y herramientas digitales) para cambiar los procesos de enseñanza/aprendizaje, pasando del trabajo individual al colaborativo, apoyando el desarrollo de programas en parejas y pequeños grupos. Después se incorporaron videos realizados por los profesores al proceso de enseñanza, los estudiantes generaban su visualización en jornadas extraclase y usaban la clase presencial para el desarrollo de ejercicios y proyectos de programación, con la retroalimentación del profesor y el apoyo de los compañeros (Bachelor y Bachelor, 2019; Nunn *et al.*, 2016).

Durante estos años se encuentran proyectos relevantes para el apoyo del aprendizaje, tecnologías como sistemas de gestión de contenidos (CMS, por su sigla en inglés), entorno virtual de aprendizaje (EVA) y objetos virtuales de aprendizaje (OVA), han evolucionado desde su implementación con escasos elementos de configuración y poco material disponible, hasta sistemas que integran metodologías para fortalecer el desarrollo del aprendizaje de manera intuitiva, la educación basada en proyectos y las comunidades de aprendizaje autónomo (Barik *et al.*, 2013; Grover *et al.*, 2014). En el desarrollo de herramientas y estrategias se encuentran trabajos basados en aula invertida para la enseñanza de lenguajes de programación como Python, Ruby, Objective-C, JavaScript, R y Swift en cursos gratuitos (Ahmed *et al.*, 2016; Hayashi *et al.*, 2015; Pe Rosiene y Rosiene, 2015), que implementan estrategias como cuestionarios informados, evaluadores de código fuente, juego de roles, entre otras, que se apoyan en videos interactivos, debates virtuales, plataformas con compiladores en línea y tecnologías como recursos educativos abiertos (OER, por su sigla en inglés) (Johnson *et al.*, 2016).

	2017	2018	2019	2020
Herramienta Tecnologías	Cuestionarios autoinformados Auto evaluación	Gamificación, Aprendizaje Basado en Investigación, Preguntas de autograbadas con secuenciación	ABP, Debates online, Juego de Roles, Videos, Aprendizaje Personalizada	Inverso del profesor Clase doble (aula 4.0) Aprendizaje asistido por computador
	MOOC, SPOC, Blackboard, E-learning Taxonomía de Bloom	LMS, Flipped Learning Network	MOOCs automáticos Realidad Aumentada en juegos de aprendizaje	Calificación automática Software adaptativo Tutores online Realimentación online Información embebida
	Fifcos Flipped Classroom Trampoline ludere-old	nodeqb comp495 sp18 flipped tech SOPHIA Facebook	Flipped Classroom PyCrawler ML100Days Psychocognitive-in-flipped-classroom	Kahoot Socrative Cavan

Figura 3. Evolución de tecnologías, estrategias y herramientas basadas en aula invertida (2017-2020)

Fuente: Elaboración propia

En 2017 y 2018 se implementaron varias estrategias a los procesos de enseñanza, por ejemplo, la evaluación automática, que utiliza una metodología de evaluación en línea y que permite detectar de las preguntas a evaluar cuáles se contestan correctamente y cuáles no, tras realizar el estudio previo del material correspondiente a programación. También se implementó la tutoría en línea, donde el estudiante solicita al profesor una orientación a través del sistema de gestión de aprendizaje y el docente la realiza sin necesidad de tener presencialidad con el estudiante, estas tutorías se pueden grabar y sirven de soporte para el aprendizaje de otros estudiantes.

Así mismo se implementaron ejercicios con retroalimentación que apoyaron los procesos de aprendizaje de los cursos de programación a través de la identificación de errores comunes de los estudiantes durante la generación de código fuente. De la misma forma se utilizó la gamificación que ayudó a mejorar la motivación de los aprendices frente al desarrollo de las actividades propuestas en los cursos (Criollo-C y Luján-Mora, 2018; Llorente *et al.*, 2020).

Todas estas estrategias se soportan con tecnologías como los MOOC (*massive open online course*), se caracterizaron por dar respuesta y evaluar contenidos en tiempo real, múltiples configuraciones para la formación de grupos y diferentes formas de evaluación en los cursos; todos estos elementos han contribuido a mejorar los procesos educativos y formativos. También se han integrado sistemas virtuales de tutoría en línea con retroalimentación, ejercicios de autocomprobación y tareas calificadas automáticamente (Clark *et al.*, 2016; Seeling, 2016), todo esto soportado con herramientas de evaluación automática de código y uso de herramientas como Blackboard (Blackboard.es, s. f.), Udere, Kattis, Baekjoon y Judge Node (Juan, n. d.).

Desde 2019 se han implementado diferentes estrategias como el aprendizaje basado en problemas, debates en línea, juego de roles, aprendizaje personalizado, clase doble, entre otros, que han aportado a la enseñanza de la programación. Por ejemplo, la comunicación entre el docente y el estudiante mejoró, y en algunos casos es personalizada, esto permite aclarar dudas o necesidades específicas para cada alumno. Los estudiantes avanzan según su ritmo de aprendizaje y el docente puede generar retroalimentación sobre temas específicos del curso. También se mejoró la motivación de los estudiantes, porque están buscando constantemente soluciones a los problemas propuestos por el profesor. Además, las clases se pueden desarrollar sin la presencia de los docentes porque existen plataformas y materiales digitales que se pueden acceder y estudiar según la disposición de los estudiantes. La visión del aula invertida está enfocada en proyectos de plataformas de aprendizaje como Google Classroom, Coursera, Facebook, Learn, Platzi, donde se mejoran los procesos de aprendizaje con la implementación de asistentes virtuales que adaptan las clases a las necesidades del estudiante. Proyectos como Lopukhova *et al.* (2020) y Tong *et al.* (2020) muestran

información, implementaciones y comparaciones de diferentes asistentes virtuales y nuevas formas de educación virtual que ayudan al estudiante tener una clase virtual dinámica, activa y con muy pocos recursos de software.

(T4) Estado actual del aprendizaje colaborativo

Las figuras 4 y 5 muestran los avances del aprendizaje colaborativo, donde se encuentran diferentes implementaciones y estrategias que han servido de apoyo en el aprendizaje de la programación.

	2013	2014	2015	2016
Estrategias	Debates Ejercicios de programación grupal	Programación en Parejas	Trabajo Social colaborativo Discusión entre pares Programación por colaboración	Evaluación automática Retroalimentación grupal
Tecnologías	CMS, LMS, LCMS Cloud	MOOC	REA OER	EduTECAS E-learning
Herramienta	Google Drive	OpenEdX Coursera Youtube	Flippd ClassEInversee BEN-T1 Mindstorms Scratch	flipped-daemonS MyLab programming

Figura 4. Evolución de tecnologías, estrategias y herramientas basadas en aprendizaje colaborativo (2013-2016)

Fuente: elaboración propia.

Entre 2013 y 2016, diferentes estrategias se implementaron y evolucionaron, por ejemplo: los debates, los cuales fueron claves para la programación grupal y la programación en parejas; estas ayudaron a generar estrategias colaborativas para apoyar el aprendizaje de la programación integrando diferentes herramientas y tecnologías (Hayashi *et al.*, 2015), mejorando los procesos académicos, despertando habilidades y destrezas que le permiten al estudiante ser un profesional idóneo en la industria. Entre estos años, además surgen modelos

de colaboración y cooperación que fomentan la inteligencia colectiva (Suárez Valencia *et al.*, 2015), asimismo la implementación de técnicas computacionales para guiar grupos y herramientas de codificación y discusión colaborativa (Sun *et al.*, 2019).

	2017	2018	2019	2020
Estrategias	Videos dinámicos de interacción grupal Programación en Parejas	Aprendizaje colaborativo asistido por computador	Aprendizaje colaborativo basado en reglas y jueces virtuales Rubricas de evaluación grupal	Clase basada en autogestión de grupos Clase doble (aula 4.0) Aprendizaje asistido por computador
Tecnologías	Taxonomía de Bloom	LMSG Flipped Learning Social network	Realidad virtual colaborativa	Software adaptativo Tutores online Realimentación online
Herramienta	Fifcos Flipped Classroom VariantSpark	Flipped Classroom nodeqb SOPHIA Facebook	Algorithm Flipped Classroom PyCrawler flip_table	Kahoot Socrative Cavan M-IDEA UNCcode

Figura 5. Evolución de tecnologías, estrategias y herramientas basadas en aprendizaje colaborativo (2017-2020)

Fuente: elaboración propia.

Entre 2017 y 2018 se encuentran diferentes herramientas basadas en la evaluación colaborativa a través de jueces virtuales (Hidalgo Suárez *et al.*, 2018) que permiten evaluar código fuente de manera sintáctica y semántica en proyectos colaborativos, ofreciendo realimentación grupal. En estos años toma fuerza la programación grupal apoyada en herramientas virtuales que realizan el seguimiento del proceso de un grupo de estudiantes (Amarasinghe *et al.*, 2017; Hwei Wang y Liao, 2017). La gestión de proyectos de programación con diferentes formas de pensar lleva a que se construyan metodologías que faciliten la configuración de equipos dependiendo de las habilidades de cada estudiante; en este sentido aparece el colaborativo apoyado por computador (CSCL), que busca integrar profesor, tecnología y estudiantes para obtener lo mejor de cada uno y generar estrategias

para la toma de decisiones en el proceso de aprendizaje (Mader y Bry, 2019; Sánchez-Azqueta *et al.*, 2019; Viberg *et al.*, 2020); cursos de programación basados en las habilidades cognitivas y emocionales (Avry *et al.*, 2020; Awidi y Paynter, 2019); hasta plataformas que unen la gestión y diseño de cursos de programación basados en la colaboración (Suárez Valencia *et al.*, 2015), donde se puede evaluar y realimentar conocimientos grupales e individuales y generar estrategias para la toma de decisiones. Este tipo de plataformas promueven el aprendizaje de la programación a partir de jueces virtuales con capacidad de evaluar el código sintáctico y semántico (Sun *et al.*, 2019), utilizan algoritmos para identificación de perfiles, gestión de grupos y evaluación de conocimientos a través de rúbricas.

A partir de 2019, se han implementado redes de conocimiento, redes sociales de los estudiantes y aplicaciones virtuales didácticas basadas en aprendizaje colaborativo apoyado por computador (CSCL), con el propósito de mejorar el aprendizaje colaborativo de los estudiantes Xie *et al.*, 2020), usando las redes sociales de los estudiantes, aplicaciones virtuales didácticas basadas en aprendizaje colaborativo apoyado por computador (CSCL).

(T3) La inteligencia artificial para el aprendizaje de los cursos de programación

La tecnología y la educación actualmente se encuentran en su punto de exploración, donde la integración del aula invertida y la colaboración cumplen un papel importante en el proceso de formación del estudiante, buscando mecanismos de interacción lo más cercanos al aula de clases tradicional. Las técnicas de inteligencia artificial han ayudado a mejorar los procesos adecuándose a las necesidades de los estudiantes, logrando que la educación tradicional se pueda apoyar en la educación virtual, generando asistentes inteligentes, evaluadores de

código automáticos, identificación de perfiles, sistemas de recomendación, etc. En esta sección se abordan las preguntas ¿cómo el aula invertida y el aprendizaje colaborativo a través de técnicas computacionales de inteligencia artificial apoyan los cursos de programación?, y ¿cuál es el papel que tendrá la inteligencia artificial en el aprendizaje de los cursos de programación?

En la figura 6 se muestran las referencias según las técnicas computacionales a partir del Sistema de Clasificación de Computación de ACM. Se detallan los artículos y proyectos (puntos negros: artículos, puntos blancos: repositorios), esta clasificación indica la tendencia del uso de las técnicas de inteligencia artificial.

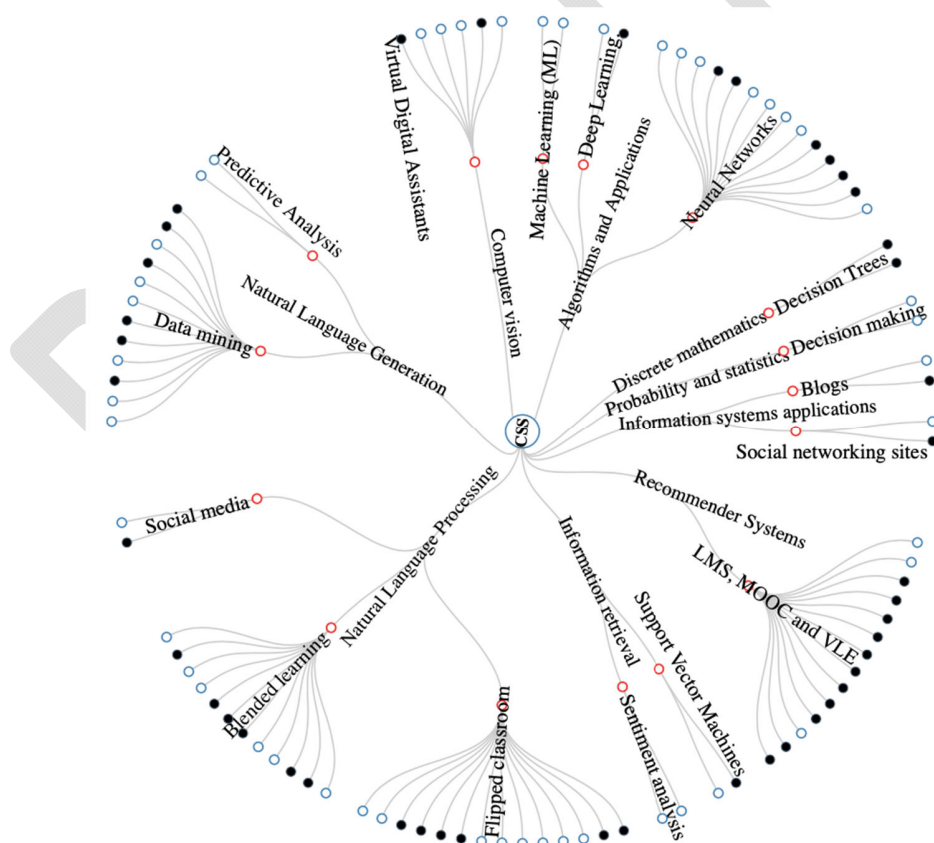


Figura 6. Clasificación de las referencias según las categorías de la CSS-ACM

Fuente: elaboración propia.

Actualmente, la implementación de inteligencia artificial en las estrategias pedagógicas ha mostrado avances significativos en el aprendizaje de la programación, utilizando procesamiento del lenguaje natural, aulas cibernéticas que registran las conferencias de forma automática y se almacenan en línea. Estas herramientas permiten crear mesas de trabajo donde el docente interactúa con sus estudiantes y desarrolla el aprendizaje activo y colaborativo (Burch y Melby, 2019). Basados en aprendizaje automático, por un lado, se encuentran implementaciones de algoritmos de evaluación integral difuso para el análisis de códigos fuente, a partir de los árboles sintácticos (Lei *et al.*, 2017). Por otro lado, proyectos que implementan técnicas de análisis de datos con *agrupamiento jerárquico aglomerado* para analizar la secuencia de aprendizaje exploratorio del estudiante, identificando comportamientos relacionados con el aprendizaje y agrupando secuencias de aprendizaje similar (Trivodaliev *et al.*, 2017). Proyectos basados en minería de datos educativos, que combinan métodos de seguimiento grupal, ayudan a identificar y seguir el proceso de los estudiantes, con el fin de observar las limitaciones y desafíos frente el aprendizaje (Sheard *et al.*, 2013). Estrategias de aprendizaje para reforzar los comportamientos del aprendizaje y las características de los estudiantes en clase de programación (Abirami y Kiruthiga, 2018). Por último, según la clasificación, donde hay mayor aporte de la inteligencia artificial es en la implementación de técnicas de asociación, clasificación y agrupación de estudiantes con asistentes virtuales autónomos (Billis y Cubenas, 2020).

La visión de la inteligencia artificial está basada en la implementación de *framework* educacionales (Fernández-Cerero y Fernández-Montes, 2020) que integran componentes

virtuales como tableros digitales, asistentes virtuales, jueces automáticos de código desde diferentes plataformas. La tendencia está en la implementación de arquitecturas basadas en la nube (educación en la nube) que son nuevas implementaciones que apoyan académicos y estudiantes a través de gamificación, colaboración abierta y distribuida, entre otras técnicas que mejoran el aprendizaje (Zhang *et al.*, 2020): clases adaptativas que involucran la parte cognitiva, abstracta y lógica con técnicas de *deep learning* en el aprendizaje virtual (Gren, 2020; Ramaprasad *et al.*, 2020); plataformas que ayudan a mejorar la experiencia de aprendizaje en los estudiantes, donde su conocimiento se va generando según sus preferencias de estudio y experiencias de aprendizaje (Algayres y Triantafyllou, 2020; Ciolacu *et al.*, 2019).

(T2) Conclusiones

Este trabajo presenta el estado actual del AI y el AC como apoyo para el aprendizaje de la programación de computadores; las tecnologías, estrategias y herramientas existentes, y el papel que tendrá la inteligencia artificial para el aprendizaje de los cursos de programación. Se encontró que los elementos identificados ayudaron a mejorar los procesos formativos de los estudiantes, con evaluación automática de código, ejercicios con retroalimentación, tutoriales interactivos, programación en parejas y ejercicios de programación grupal, que se basaron en la integración de algoritmos de inteligencia artificial.

Luego de realizar la investigación, se puede concluir que, desde la aparición de las herramientas virtuales, la implementación de inteligencia artificial se ha convertido en una necesidad para la educación virtual, porque potencia la forma de entender y aprender, adaptándose a las necesidades específicas del estudiante. A futuro, la inteligencia artificial

tendrá un papel muy importante en la implementación de algoritmos para la identificación de aprendizaje personalizado a partir del aprendizaje automático, procesamiento de lenguaje natural, aprendizaje profundo, analíticas de aprendizaje y minería de datos; que se integran a plataformas virtuales capaces de extraer información de redes sociales con valor agregado para generar estrategias de evaluación y realimentación en los procesos formativos de los estudiantes. Sin embargo, aún quedan preguntas abiertas de exploración, para determinar mejoras en los niveles de aprendizaje tradicional y la implementación de estrategias virtuales. En trabajos previos relacionados al AI y el AC se identificaron aportes significativos relacionados con el aprendizaje de la programación, que han abordado la evaluación automática de código fuente, la identificación de estudiantes en riesgo y el trabajo colaborativo. Sin embargo, como trabajo futuro se propone trabajar en la retroalimentación de los ejercicios de programación, con el fin dar respuesta a los inconvenientes y malas prácticas que se pueden presentar durante la escritura del código fuente. De esta forma se puede apoyar el proceso de aprendizaje, mejorando el interés y la motivación del aprendiz por el desarrollo del *software*.

De las herramientas basadas en aula invertida y la colaboración encontrada en este artículo, las que más se destacan como apoyo al aprendizaje del curso de programación, son: EdX, Coursera, Caccoo, Socrative, y las que usan inteligencia artificial son: M-IDEA, UNCode, MyLAB programming.

(T2) Referencias

Abirami, A. M. y Kiruthiga, P. (2018). Collaborative learning tools for data structures. *Journal of Engineering Education Transformations*, 31(3), 79-83.

Ahmed, B., Aljaani, A. y Yousuf, M. I. (2016). Flipping introductory engineering design courses: Evaluating their effectiveness. En IEEE Global Engineering Education Conference, EDUCON, (pp. 234-239).

<https://doi.org/10.1109/EDUCON.2016.7474559>

Algayres, M. y Triantafyllou, E. (2020). Combining the flipped classroom and simulation games in engineering education: A methodological survey. En SEFI 47th Annual Conference: Varietas Delectat... Complexity Is the New Normality, Proceedings (pp. 83-92).

<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85077815856&partnerID=40&md5=f5e6903495c4235b6a899968f2801dab>

Amarasinghe, I., Hernández-Leo, D. y Jonsson, A. (2017). Intelligent Group Formation in Computer Supported Collaborative Learning Scripts. En Proceedings - IEEE 17th International Conference on Advanced Learning Technologies, ICALT 2017 (pp. 201-203). <https://doi.org/10.1109/ICALT.2017.62>

Amresh, A., Carberry, A. R. y Femiani, J. (2013). Evaluating the effectiveness of flipped classrooms for teaching CS1. En 2013 IEEE Frontiers in Education Conference (FIE) (pp. 733-735). <https://doi.org/10.1109/FIE.2013.6684923>

Association for Computing Machinery (ACM). (s. f.). The 2012 ACM Computing Classification System. <https://www.acm.org/publications/class-2012>

Avry, S., Chanel, G., Bétrancourt, M. y Molinari, G. (2020). Achievement appraisals, emotions and socio-cognitive processes: How they interplay in collaborative problem-solving? *Computers in Human Behavior*, 107. <https://doi.org/10.1016/j.chb.2020.106267>

- Awidi, I. T. y Paynter, M. (2019). The impact of a flipped classroom approach on student learning experience. *Computers and Education*, 128, 269-283. <https://doi.org/10.1016/j.compedu.2018.09.013>
- Bachelor, J. W. y Bachelor, J. W. (2019). El aula presencial, semipresencial, virtual e invertida: un estudio comparativo de métodos didácticos en la enseñanza de L2. *Revista Educación*, 43(2), 527-539. <https://doi.org/10.15517/revedu.v43i2.34014>
- Barik, T., Everett, M., Cardona-Rivera, R. E., Roberts, D. L. y Gehringer, E. F. (2013). A community college blended learning classroom experience through Artificial Intelligence in Games. En *Proceedings - Frontiers in Education Conference, FIE* (pp. 1525-1531). <https://doi.org/10.1109/FIE.2013.6685093>
- Billis, S. y Cubenas, O. (2020). Assessing collaborative learning with e-tools in engineering and computer science programs. *Advances in Intelligent Systems and Computing*, 1070, 848-854. https://doi.org/10.1007/978-3-030-32523-7_62
- blackboard.es. (s. f.). ¿Qué es Blackboard Learn? Ayuda de Blackboard. https://help.blackboard.com/es-es/Learn/Instructor/Getting_Started/What_Is_Blackboard_Learn
- Bostock, M. (s. f.). D3.js-Data-Driven Documents. <https://d3js.org/>
- Bravo, L. E. C., Elizalde, I. E. y Ortiz, J. A. T. (2013). Estrategias educativas para el uso de las TIC en educación superior. *Tecnura*, 17, 161-173.
- Burch, M. y Melby, E. (2019). Teaching and evaluating collaborative group work in large visualization courses. En *ACM International Conference Proceeding Series* (pp. 1-8). <https://doi.org/10.1145/3356422.3356447>

Chinchilla, C. M. D. y Gómez, A. A. R. (2020). Aprendizaje activo e innovación en estudiantes de ingeniería. *Revista Colombiana de Tecnologías de Avanzada (RCTA)*, 1(35), 127-135.

Ciolacu, M., Tehrani, A. F., Binder, L. y Svasta, P. M. (2019). Education 4.0-Artificial Intelligence Assisted Higher Education: Early recognition System with Machine Learning to support Students' Success. En 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging, SIITME 2018 - Proceedings (pp. 23-30).

<https://doi.org/10.1109/SIITME.2018.8599203>

Clark, R. M., Kaw, A. y Besterfield-Sacre, M. (2016). Comparing the effectiveness of blended, semi-flipped, and flipped formats in an engineering numerical methods course. *Advances in Engineering Education*, 5(3).
<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85006761527&partnerID=40&md5=3b34e0497418055bc0df77b1a3d4a2b6>

Criollo-C, S. y Luján-Mora, S. (2018). Encouraging student motivation through gamification in engineering education. En *Interactive Mobile Communication, Technologies and Learning* (pp. 204-211). https://doi.org/10.1007/978-3-030-11434-3_24

Fernández-Cerero, D. y Fernández-Montes, A. (2020). Project-based methodology to lecture on web frameworks applied to the management of health-related data. *Advances in Intelligent Systems and Computing*, 951, 321-328. https://doi.org/10.1007/978-3-030-20005-3_33

Fortanet van Assendelft, C. A. (2013). Aprendizaje cooperativo y flipped classroom. Ensayos y resultados de la metodología docente. Universidad de Alicante; Vicerrectorado de Estudios, Formación y Calidad.

Gren, L. (2020). A flipped classroom approach to teaching empirical software engineering. *IEEE Transactions on Education*, 63(3), 155-163.

<https://doi.org/10.1109/TE.2019.2960264>

Grover, S., Pea, R. y Cooper, S. (2014). Promoting active learning & leveraging dashboards for curriculum assessment in an OpenEdX introductory CS course for middle school. En *Proceedings of the First ACM Conference on Learning@ Scale Conference* (pp. 205-206). <https://doi.org/10.1145/2556325.2567883>

Guerrero, V. A. B. (2019). Desarrollo del estado del arte en investigación: una herramienta basada en inteligencia artificial. *Revista Politécnica*, 15(30), 70-81.

<https://doi.org/10.33571/rpolitec.v15n30a7>

Hayashi, Y., Fukamachi, K.-I. y Komatsugawa, H. (2015). Collaborative learning in computer programming courses that adopted the flipped classroom. En *Proceedings - 2015 International Conference on Learning and Teaching in Computing and Engineering, LaTiCE 2015* (pp. 209-212). <https://doi.org/10.1109/LaTiCE.2015.43>

Hidalgo Suárez, C. G., Bucheli, V. A., Restrepo-Calle, F. y González, F. A. (2018). A strategy based on technological maps for the identification of the state-of-the-art techniques in software development projects: Virtual judge projects as a case study. En J. E. Serrano y J. C. Martínez-Santos (Eds.), *Advances in computing* (pp. 338-354). Springer International Publishing https://doi.org/10.1007/978-3-319-98998-3_27

hwei Wang, Y. y Liao, H. C. (2017). Learning performance enhancement using computer-assisted language learning by collaborative learning groups. *Symmetry*, 9(8), 141.

<https://doi.org/10.3390/sym9080141>

Johnson, C., McGill, M., Bouchard, D., Bradshaw, M. K., Bucheli, V. A., Merkle, L. D., Scott, M. J., Sweedyk, Z., Velázquez-Iturbide, J. Á., Xiao, Z. y Zhang, M. (2016). Game Development for Computer Science Education. En Proceedings of the 2016 ITiCSE Working Group Reports (pp. 23-44). <https://doi.org/10.1145/3024906.3024908>

Juan, M. (s. f.). udere-old: EVA orientado al b-learning con gamification para facilitar la modalidad flipped-classroom. <https://github.com/juanmamaffei/ludere-old>

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M. y Linkman, S. (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>

Lei, C.-U., Yau, C.-W., Lui, K.-S., Yum, P., Tam, V., Yuen, A. H.-K. y Lam, E. Y. (2017). Teaching Internet of Things: Enhancing learning efficiency via full-semester flipped classroom. En Proceedings of 2017 IEEE International Conference on Teaching, Assessment and Learning for Engineering, TALE 2017 (pp. 56-60). <https://doi.org/10.1109/TALE.2017.8252304>

Llorente, L. D., Gómez, E. A. V. y Villamizar, S. C. (2020). Gamificación como estrategia pedagógica medida por TIC en educación básica primaria. *Revista Colombiana de Tecnologías de Avanzada (RCTA)*, 2(34), 111-116.

Lopukhova, J., Makeeva, E. y Rudneva, T. (2020). Using flipped classroom in foreign language teaching: Implementation of interactive educational technologies. *Advances in Intelligent Systems and Computing*, 1135, 619-630.

https://doi.org/10.1007/978-3-030-40271-6_61

- Mader, S. y Bry, F. (2019). Phased classroom instruction: A case study on teaching programming languages. En CSEDU 2019 - Proceedings of the 11th International Conference on Computer Supported Education, 1, 241-251. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85067114371&partnerID=40&md5=1ddf4f4d561f5919e5a8944c9d03f1bf>
- Moher, D., Liberati, A., Tetzlaff, J. y Altman, D. G. (2010). Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *International Journal of Surgery*, 8(5), 336-341. <https://doi.org/10.1016/j.ijssu.2010.02.007>
<https://doi.org/10.1016/j.ijssu.2010.02.007>
- Mok, H. N. (2014). Teaching tip: The flipped classroom. *Journal of Information Systems Education*, 25(1), 7-11.
- Montero, E. L. P. y Benavides, E. S. (2015). Estructuras de decisión a partir del aprendizaje autorregulado en ambientes B-Learning. *Tecnura*, 19, 15-24.
- Nunn, S., Avella, J. T., Kanai, T. y Kebritchi, M. (2016). Learning analytics methods, benefits, and challenges in higher education: A systematic literature review. *Online Learning*, 20(2). <https://doi.org/10.24059/olj.v20i2.790>
- Pe Rosiene, C. y Rosiene, J. A. (2015). Flipping a programming course: The good, the bad, and the ugly. En *Proceedings - Frontiers in Education Conference, FIE, 2014* (pp. 1-3). <https://doi.org/10.1109/FIE.2015.7344151>
- Ramaprasad, H., Bahamon, J. C., Jones, R. H. y Watson, S. (2020). Using forcing functions to improve student preparedness in an operating systems and networking class. En *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, (p. 1366). <https://doi.org/10.1145/3328778.3372668>

Salazar Ospina, O. M., Rodríguez Marín, P. A., Ovalle Carranza, D. A. y Duque Méndez, N. D. (2017). Interfaces adaptativas personalizadas para brindar recomendaciones en repositorios de objetos de aprendizaje. *Tecnura*, 21(53), 107-118.

<https://doi.org/10.14483/22487638.9287>

Sánchez-Azqueta, C., Celma, S., Aldea, C., Gimeno, C. y Cascarosa, E. (2019). ICT-based didactic strategies to build knowledge models in electronics in higher education. En *Proceedings - IEEE International Symposium on Circuits and Systems, 2019* (pp. 1-5). <https://doi.org/10.1109/ISCAS.2019.8702527>

Seeling, P. (2016). Switching to blend-Ed: Effects of replacing the textbook with the browser in an introductory computer programming course. En *Proceedings - Frontiers in Education Conference, FIE, 2016* (pp. 1-5). <https://doi.org/10.1109/FIE.2016.7757620>

Sheard, J., Simon, Carbone, A., D'Souza, D. y Hamilton, M. (2013). Assessment of programming: Pedagogical foundations of exams. En *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education* (pp. 141-146). <https://doi.org/10.1145/2462476.2465586>

Suárez Valencia, E., Bucheli, V., Zarama, R. y Garcia, Á. (2015). Collective intelligence: Analysis and modelling. *Kybernetes*, 44(6/7), 1122-1133. <https://doi.org/10.1108/K-11-2014-0245>

Sun, Q., Wu, J., Rong, W. y Liu, W. (2019). Formative assessment of programming language learning based on peer code review: Implementation and experience report. *Tsinghua Science and Technology*, 24(4), 423-434. <https://doi.org/10.26599/TST.2018.9010109>

- Tong, Y., Kinshuk y Wei, X. (2020). Teaching design and practice of a project-based blended learning model. *International Journal of Mobile and Blended Learning*, 12(1), 33-50. <https://doi.org/10.4018/IJMBL.2020010103>
- Trivodaliev, K., Stojkoska, B. R., Mihova, M., Jovanov, M. y Kalajdziski, S. (2017). Teaching computer programming: The Macedonian case study of functional programming. En *IEEE Global Engineering Education Conference, EDUCON* (pp. 1282-1289). <https://doi.org/10.1109/EDUCON.2017.7943013>
- Viberg, O., Mavroudi, A., Fernaeus, Y., Bogdan, C. y Laaksolahti, J. (2020). Reducing free riding: CLASS - A system for collaborative learning assessment. *Advances in Intelligent Systems and Computing*, 1008, 132-138. https://doi.org/10.1007/978-3-030-23884-1_17
- Vychegzhanin, S. y Kotelnikov, E. (2019). Comparison of named entity recognition tools applied to news articles. En *Proceedings - 2019 Ivannikov Ispras Open Conference, ISPRAS 2019* (pp. 72-77). <https://doi.org/10.1109/ISPRAS47671.2019.00017>
- Xie, J., Lu, Y., Gao, R., Zhu, S.-C. y Wu, Y. N. (2020). cooperative training of descriptor and generator networks. En *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1), 27-45. <https://doi.org/10.1109/TPAMI.2018.2879081>
- Zhang, M., Yu, Z., Sun, Y., Gong, X. y Liu, Y. (2020). The construction and application of hybrid teaching model of flipping classroom and divided classroom base on project-driven in the teaching of Java programming. *Journal of Physics: Conference Series*, 14 53(1) <https://doi.org/10.1088/1742-6596/1453/1/012016>