

# Modelo soportado en inteligencia artificial para el desarrollo de actividades de aprendizaje activo basadas en colaboración asistida por computador (M-IDEA)

Carlos G Hidalgo S, Ms.C<sup>1</sup>, Víctor A Bucheli, Ph.D<sup>1</sup>

<sup>1</sup>Universidad del Valle, Colombia, carlos.hidalgo@correounivalle.edu.co, victor.bucheli@correounivalle.edu.co

*Abstract— Los cursos de introducción a la programación son claves en la formación de un profesional de la Ingeniería de Sistemas y/o Ciencias de la Computación, por ello es necesario la implementación de estrategias que permitan mejorar y facilitar la adquisición de competencias de aprendizaje y mitigar problemas de formación del estudiante. Nosotros proponemos un modelo (M-IDEA) que integra el aprendizaje colaborativo asistido por computador y la inteligencia artificial, buscando relacionar al estudiante, profesor y tecnología mediante estrategias de aprendizaje activo. A su vez el modelo permite estudiar las relaciones sociales y las interacciones a nivel micro, para de esta manera entender como emerge el aprendizaje colaborativo en el aula de clase. En este artículo se presenta el modelo, la versión prototipo cero y el diseño experimental para las primeras pruebas en un curso de introducción a la programación*

*Keywords— CSCL, CS1, Aprendizaje activo.*

## I. INTRODUCCIÓN

Los problemas en el aprendizaje de la programación han sido reportados desde los años 50's [1], los cuales están relacionados con la adquisición de las competencias de aprendizaje [2], [3]. Estos problemas se han generado independiente del paradigma o lenguaje de programación [4], [5], de la instrucción tradicional [6] o de tipo de aprendizaje [7], [8]. Sin embargo ¿Cómo hacer que los estudiantes alcancen las competencias académicas de programación? Es una pregunta de investigación abierta, en la literatura se han planteado diferentes propuestas para dar solución, entre las que se encuentran, estilos de aprendizaje [9], metodologías, métodos y estrategias [10]–[12], juegos [13], [14], herramientas y recursos tecnológicos [15], [16], entre otros.

Entre los cursos de programación, el curso de introducción a la programación (CS1) es el que más casos de bajo rendimiento académico presenta, además de ser el curso con mayor deserción y mortalidad académica [17]–[20]. Los estudiantes del primer curso de programación deben adquirir competencias básicas en resolución de problemas, lógica y matemática, entre otras [21]. Las cuales son la base para otras competencias que se adquieren en la formación de todo el programa de Ingeniería de Sistemas y/o Computación, esto parecería verse reflejado en el proceso de formación del estudiante [22]–[24].

Nosotros proponemos un modelo (M-IDEA) de las siglas de Modelo soportado en Inteligencia artificial para el Desarrollo de actividades de aprendizaje activo basadas en

CSCL para la Educación de Aptitudes en programación. El modelo se basa en los procesos del enfoque pedagógico CSCL (Siglas en inglés de Computer Supported Collaborative Learning) el cual busca gestionar el proceso para lograr que los estudiantes obtengan las competencias de aprendizaje de manera colaborativa [25]–[27].

Este artículo esta organizado en 3 secciones. La siguiente sección presenta el modelo M-IDEA. La sección presenta resultados previos, y el finalmente discusión y trabajo futuro

## II. MODELO M-IDEA

Para resolver la pregunta ¿Cómo apoyar las competencias en un curso de programación a través de actividades de aprendizaje activo basadas en el enfoque CSCL? Proponemos resolver el modelo M-IDEA.

En la construcción de M-IDEA se integran diversas estrategias para los procesos CSCL: la formación de grupos, la evaluación formativa y la realimentación. Todo esto como apoyo al diseño de actividades de aprendizaje activo de los cursos introductorios de programación, específicamente en el curso CS1. El modelo propone al proceso de aprendizaje como un espacio social donde las interacciones a nivel micro entre estudiante, profesor y tecnología permiten la emergencia de un sistema de aprendizaje colaborativo. Con el trabajo de investigación integrar los conceptos, el diseño del modelo y el desarrollo de los artefactos y módulos de software que permitan la implementación de actividades de aprendizaje activo y colaborativo, adicionalmente a través de la implementación de analíticas de aprendizaje observar cómo sus integrantes (estudiantes) van alcanzando las competencias en programación.

M-IDEA será implementado en un ambiente prototipo, que integrará técnicas computacionales de inteligencia artificial, que soporten los 3 procesos de CSCL. Permitiendo apoyar al profesor en el diseño y seguimiento de las actividades de aprendizaje activo. Logrando así identificar las falencias que se generan en el proceso educativo del curso y así tomar decisiones de mejora, para así alcanzar competencias de programación. En la Figura. 1 se muestran los procesos de M-IDEA.

**Digital Object Identifier:** (to be inserted by LACCEI).  
**ISSN, ISBN:** (to be inserted by LACCEI).

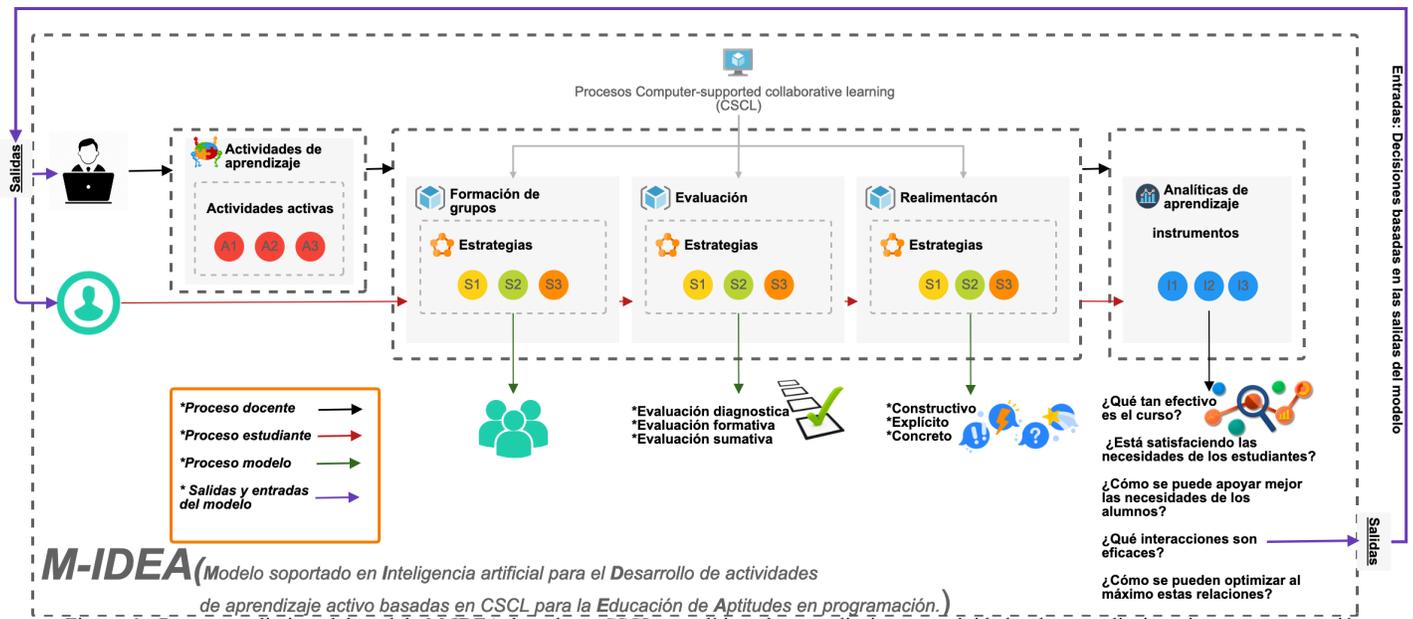


Figura. 1 - Proceso preliminar del modelo M-IDEA, basado en CSCL y analíticas de aprendizaje para actividades de aprendizaje activo en programación

### III. RESULTADOS PREVIOS

En el curso de algoritmia y programación (A&P) de la facultad de Ingeniería que se dicta en la Universidad del Valle, actualmente se cuenta con cerca de 300 estudiantes inscritos de diferentes carreras. En este curso los estudiantes deben alcanzar competencias básicas de la programación las cuales están sujetas al aprendizaje del lenguaje Python. En un previo estudio encontramos que en promedio un 37% de los estudiantes pierden el curso, un 4% pierden por faltas y un 2% pierden por otras razones, siendo este uno de los cursos con mayor deserción y mortalidad académica. Por ello seleccionamos el curso de A&P para nuestro estudio, donde elegimos una población de 73 estudiantes, divididos en 4 cursos, Una vez elegida la población pusimos en práctica una actividad de programación basada en aprendizaje activo y colaborativo suportada por computador.

#### A. Actividad de programación

Se propone una actividad sencilla en el lenguaje de programación Python usando ciclos.

22 estudiantes fueron el grupo control el cual tendrán las dos horas de clase para realizar la actividad y trabajan de manera individual.

Dos grupos de 22 estudiantes cada uno (grupo experimental) realizaron la actividad a través de una herramienta prototipo, en la cual el estudiante debe solucionar en menos de 10 minutos la actividad, si el estudiante no completa la actividad en ese tiempo, al estudiante le ayudara otro estudiante que ya haya terminado en el tiempo establecido (la ayuda se realiza a través de pistas). De esta forma entre más iteraciones ocurran, mayor cantidad de estudiantes estarán ayudando a resolver dicha actividad. La actividad termina cuando todos los estudiantes

hayan terminado de realizar la actividad tanto para el grupo control como para el grupo experimental.

#### B. Resultados de la actividad de programación

Después de realizar las actividades se encontró que la colaboración permitió que los grupos experimentales terminaran la actividad antes que el grupo control. Además de conocer cuales son los estudiantes que tienen mayor y menor desempeño en el curso. En la Figura. 2 se observa como a través del tiempo los estudiantes que terminan la actividad aumentan cuando son expuestos a la colaboración, incluso terminado antes del tiempo propuesto.

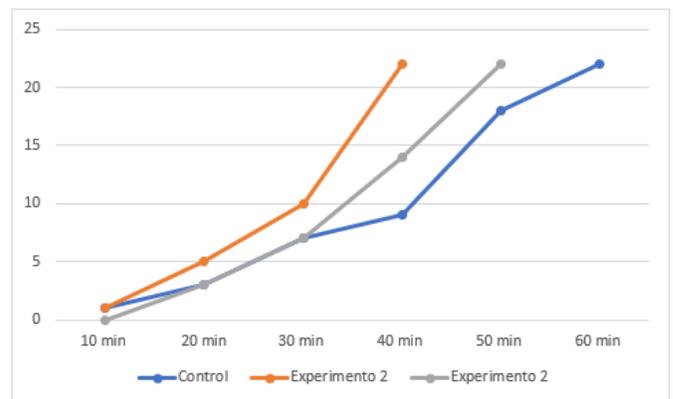


Figura. 2 - Comparación de Tiempo\*Personas entre grupo control y grupos experimentales

Esta actividad es una primera prueba donde se pudo realizar un previo acercamiento a los procesos del CSCL, actualmente se están evaluando a profundidad los resultados para poner en marcha la fase de evaluación y realimentación.

#### IV. DISCUSIÓN TRABAJO FUTURO

Actualmente en la revisión de literatura no hay una propuesta que integre ideas nuevas en los procesos CSCL, por ejemplo, para destacar habilidades individuales de forma grupal, tampoco hay estrategias para emergencia de grupos, donde la unión entre estudiantes genere un rol sin ser especificado. De igual forma no se reporta en la literatura un modelo apoyado en técnicas computacionales de inteligencia artificial para mejorar los procesos CSCL y la toma de decisiones basados en el análisis de datos. Finalmente, no se encuentran estudios que revelen cómo integrar nuevos elementos a los procesos CSCL y el impacto en la adquisición de competencias en cursos de programación. hay un modelo que integre esto.

Actualmente trabajamos en desarrollar estrategias CSCL propuestas en el modelo M-IDEA, las cuales se han empezado a implementar en los cursos de programación, lo que ha permitido que realicemos mejoras. Hemos conseguido alianzas con profesores que desarrollan trabajos similares, lo cual he permitido que el modelo M-IDEA sea un proyecto del cual se beneficien los estudiantes de programación.

Finalmente, el desarrollo de M-IDEA será integrado en una herramienta tecnológica que permitirá realizar prácticas activas de programación que apoyará al docente en la evaluación del código fuente, el seguimiento del estudiante y la toma de decisiones respecto a la competencia de aprendizaje que alcanza el estudiante.

#### REFERENCIAS

- [1] W. F. Atchison *et al.*, "Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science," *Commun. ACM*, vol. 11, no. 3, pp. 151–197, 1968.
- [2] S. Reges, "Back to Basics in CS1 and CS2."
- [3] S. Reges, "Back to Basics in CS1 and CS2."
- [4] J. E. Sammet, "Programming languages: history and future," *Commun. ACM*, vol. 15, no. 7, pp. 601–610, Jul. 2002.
- [5] T. J. (Tim) Bergin, "A history of the history of programming languages," *Commun. ACM*, vol. 50, no. 5, p. 69, 2007.
- [6] M. Craig, D. Zingaro, J. Campbell, D. Horton, and P. Gries, "Comparing outcomes in inverted and traditional CS1," in *Proceedings of the 2014 conference on Innovation & technology in computer science education - ITiCSE '14*, 2014, pp. 261–266.
- [7] P. Dillenbourg, "What do you mean by collaborative learning? What do you mean by 'collaborative learning'?"
- [8] Y. huei Wang and H. C. Liao, "Learning performance enhancement using computer-assisted language learning by collaborative learning groups," *Symmetry (Basel)*, vol. 9, no. 8, p. 141, Aug. 2017.
- [9] L. Thomas, M. Ratcliffe, J. Woodbury, and E. Jarman, "Learning styles and performance in the introductory programming sequence," in *Proceedings of the 33rd SIGCSE technical symposium on Computer science education - SIGCSE '02*, 2002, vol. 34, no. 1, p. 33.
- [10] O. Hazzan and Y. Dubinsky, "Teaching a software development methodology: the case of extreme programming," in *Software Engineering Education Conference, Proceedings*, 2003, vol. 2003–Janua, pp. 176–184.
- [11] J. Bennedsen, M. E. Caspersen, and M. Kölling, *Reflections on the Teaching of Programming*, vol. 4821. Springer, 2008.
- [12] S. L. Salcedo and A. M. O. Idrobo, "New tools and methodologies for programming languages learning using the scribbler robot and Alice," in *Proceedings - Frontiers in Education Conference, FIE*, 2011, p. F4G–1.
- [13] R. Rajaravivarma, "A games-based approach for teaching the introductory programming course," *ACM SIGCSE Bull.*, vol. 37, no. 4, p. 98, 2006.
- [14] S. Leutenegger and J. Edgington, "A games first approach to teaching introductory programming," in *ACM SIGCSE Bulletin*, 2007, vol. 39, no. 1, p. 115.
- [15] G. Javidi and E. Sheybani, "Teaching Computer Programming through Game Design: A Game-First Approach," *GSTF J. Comput.*, vol. 4, no. 3, pp. 1–10, 2015.
- [16] S. Voinakis, G. Anastassakis, and P. Koutsabasis, "Teaching and learning logic programming in virtual worlds using interactive microworld representations," *Br. J. Educ. Technol.*, vol. 49, no. 1, pp. 30–44, 2018.
- [17] J. P. Munson and J. P. Zitovsky, "Models for Early Identification of Struggling Novice Programmers," in *Proceedings of the 2018 ACM SIGCSE Technical Symposium on Computer Science Education - SIGCSE '18*, 2018, pp. 699–704.
- [18] S. I. Malik, "Improvements in Introductory Programming Course: Action Research Insights and Outcomes," *Syst. Pract. Action Res.*, vol. 31, no. 6, pp. 637–656, 2018.
- [19] D. Zingaro *et al.*, "Identifying Student Difficulties with Basic Data Structures," in *Proceedings of the 2018 ACM Conference on International Computing Education Research - ICER '18*, 2018, pp. 169–177.
- [20] J. Figueiredo and F. J. García-Peñalvo, "Building Skills in Introductory Programming," in *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM'18*, 2018, pp. 46–50.
- [21] S. Sultana, "Defining the competencies, programming languages, and assessments for an introductory computer science course," 2016.
- [22] A. Gomes and A. J. Mendes, "Learning to program-difficulties and solutions," in *International Conference on Engineering Education*, 2007, pp. 1–5.
- [23] L. Columbus, "Roundup of cloud computing forecasts and market estimates," *Forbes Mag.*, 2015.
- [24] A.-J. Lakanen and V. Isomöttönen, "High school students' perspective to university CS1," in *Proceedings of the 18th ACM conference on Innovation and technology in computer science education - ITiCSE '13*, 2013, p. 261.
- [25] G. Stahl, T. D. Koschmann, and D. D. Suthers, *Computer-supported collaborative learning: An historical perspective*. na, 2006.
- [26] J. Chen, M. Wang, P. A. Kirschner, and C.-C. Tsai, "The Role of Collaboration, Computer Use, Learning Environments, and Supporting Strategies in CSCL: A Meta-Analysis," *Rev. Educ. Res.*, vol. 88, no. 6, p. 003465431879158, 2018.
- [27] J. W. Winter, "Analysis of knowledge construction during group space activities in a flipped learning course," *J. Comput. Assist. Learn.*, vol. 34, no. 6, pp. 720–730, 2018.