

ARTÍCULO DE INVESTIGACIÓN / RESEARCH ARTICLE

<https://dx.doi.org/10.14482/inde.43.02.519.852>

SPEI: una herramienta web de intervención temprana para el desempeño académico de estudiantes de cursos de programación

*SPEI: a web-based early intervention
tool for the academic performance of
students in programming courses*

JOSÉ MIGUEL LLANOS MOSQUERA*
VICTOR ANDRES BUCHELI GUERRERO**
FELIPE RESTREPO CALLE***

* Estudiante de doctorado en Ingeniería, Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle (Colombia). M.Sc. Orcid-ID: <https://orcid.org/0000-0003-4642-2770>.
jose.llanos@correounivalle.edu.co

** Profesor titular, Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle (Colombia). Ph.D. Orcid-ID: <https://orcid.org/0000-0002-0885-8699>.
victor.bucheli@correounivalle.edu.co

*** Profesor asociado, Departamento de Ingeniería de Sistemas e Industrial, Universidad Nacional de Colombia. Ph.D. Orcid-ID: <https://orcid.org/0000-0003-4226-1324>.
ferestrepoca@unal.edu.co

Correspondencia: José Miguel Llanos. Carrera 44 n.º 27-35, Neiva (Colombia).
Celular: 3003473415.



Resumen

En los cursos de introducción a la programación (CS1) se observa un bajo desempeño académico entre los estudiantes; problema común en las instituciones de educación superior que se atribuye a la complejidad de los conceptos y a la falta de experiencia en programación. Una manera de abordar esta situación es implementar herramientas que permitan intervenir de manera temprana en los procesos de aprendizaje para mejorar el rendimiento académico. Este estudio presenta Student's Performance Early Intervention (SPEI), herramienta web que integra un modelo de predicción del desempeño académico y dos tipos de intervención temprana para cursos CS1. El objetivo de este artículo es evaluar la efectividad de SPEI en el desempeño académico de los estudiantes en un curso de programación CS1. SPEI se desarrolló sobre la arquitectura Model-Template-View e integra los módulos de predicción, intervención preventiva e intervención proactiva. Los resultados muestran que SPEI identifica con precisión a los estudiantes con bajo rendimiento académico y permite intervenciones personalizadas que contribuyen a mejorar su desempeño. Las conclusiones sugieren que SPEI es una herramienta efectiva para apoyar el proceso formativo y podría replicarse en otros cursos para mejorar el rendimiento académico de los estudiantes en cursos CS1.

Palabras clave: CS1, desempeño académico, herramienta de intervención temprana, intervención temprana, modelo de predicción.

Abstract

In introductory programming courses (CS1), students often exhibit low academic performance, a common issue in higher education institutions attributed to the complexity of the concepts and a lack of programming experience. One way to address this situation is by implementing tools that allow early intervention in the learning process to improve academic performance. This study presents the Student Performance Early Intervention (SPEI), a web-based tool that integrates an academic performance prediction model and two types of early interventions for CS1 courses. The aim of this article is to evaluate the effectiveness of SPEI on the academic performance of students in a CS1 programming course. SPEI was developed using the Model-Template-View architecture and integrates prediction, preventive intervention, and proactive intervention modules. The results show that SPEI accurately identifies students with low academic performance and enables personalized interventions that contribute to improving their performance. The conclusions suggest that SPEI is an effective tool to support the learning process and could be replicated in other courses to enhance students' academic performance in CS1 courses.

Keywords: academic performance, CS1, early intervention, early intervention tool, prediction model.

INTRODUCCIÓN

En los cursos de introducción a la programación (CS1) se observa que los estudiantes presentan un bajo desempeño académico [1], [2]; este problema es muy común en las instituciones de educación superior de todo el mundo y suele atribuirse a varios factores, como la complejidad de los conceptos y la falta de experiencia en programación [3], [4]. Para abordar este desafío es esencial implementar herramientas de intervención temprana, las cuales permiten IDentificar las dificultades específicas que enfrentan los estudiantes y ofrecer apoyo personalizado durante el proceso formativo [5], [6]. Sin embargo, el problema continúa y se ha convertido en un desafío importante para directivos y docentes, quienes investigan estrategias y herramientas con el fin de mejorar el desempeño académico de los estudiantes [7], [8], [9], [10].

Según [7], [8], [9], las herramientas de intervención temprana son importantes, porque permiten evaluar en qué medida los estudiantes y profesores alcanzan los objetivos educativos y la eficacia en la enseñanza. Asimismo, en [5] se menciona que las intervenciones tempranas son necesarias en los procesos formativos, porque orientan al profesor en el apoyo personalizado del estudiante, y en [6] se recomienda implementar medidas correctivas para garantizar que los estudiantes en riesgo alcancen los resultados deseados en términos de aprendizaje y calificación.

Algunas universidades cuentan con herramientas que permiten IDentificar los estudiantes con bajo desempeño académico, brindan intervención oportuna durante el proceso formativo y generan estrategias correctivas que permiten mejorar los métodos pedagógicos [7], [9]. Sin embargo, es necesario desarrollar otras herramientas que permitan intervenir a los estudiantes durante las primeras semanas del curso, de esta manera se pueden generar aportes importantes para el desempeño académico en los cursos de programación CS1 [11], [12].

Por este motivo, basado en los aportes de [3], [13], [14], [15], este trabajo presenta SPEI, herramienta web que integra un modelo de predicción y dos tipos de intervención temprana para el desempeño académico de los estudiantes de cursos de programación CS1.

El modelo predice el desempeño académico en las semanas 3, 5 y 7 de un curso de programación de 16 semanas, e identifica a los estudiantes con bajo, medio y alto desempeño. La primera intervención se denomina *preventiva*, permite generar tutorías grupales, envío de sugerencias y código fuente de referencia al correo electrónico del estudiante intervenido desde la tercera semana del curso. El segundo tipo de intervención se denomina *proactivo*, permite el envío y seguimiento de talleres de refuerzo de programación para los estudiantes con bajo y medio desempeño académico desde la séptima semana del curso. Todo esto con el fin de responder a la pregunta de

investigación: *¿Cómo contribuye una herramienta de intervención temprana a mejorar el desempeño académico de los estudiantes en un curso de programación CS1?*

Este documento se ha organizado de la siguiente manera: en la siguiente sección están los trabajos relacionados; luego se presenta la metodología, en la que se describe la herramienta web SPEI, el procedimiento utilizado, la evaluación de SPEI y las herramientas de apoyo; luego se presentan los resultados obtenidos y las discusiones; y finalmente, las conclusiones.

TRABAJOS RELACIONADOS

En la revisión de la literatura se identificaron varias herramientas de intervención temprana que se han utilizado en cursos de programación; por ejemplo, en [11] se describe PreSS#, herramienta web que incluye intervenciones para reducir las tasas de deserción y mejorar el rendimiento académico. En el documento se destaca la eficacia de las intervenciones y la importancia de adaptar el modelo de predicción a diferentes contextos educativos para mejorar los resultados de los estudiantes.

Asimismo, en [16] se menciona VEAP, herramienta que permite a los profesores visualizar los datos de los estudiantes y generar análisis personalizados para evaluar cómo las intervenciones pueden influir en el rendimiento de los estudiantes.

En [17] se describe Caring IDE, herramienta diseñada para ofrecer una experiencia de aprendizaje de programación más fluida en los estudiantes y mejorar la capacidad del profesor para realizar intervenciones informativas de manera rápida y temprana. Los resultados indican que el uso de análisis basados en este IDE puede ser valioso para mejorar el desempeño de los estudiantes en cursos de programación.

De manera similar, en [12] describen el desarrollo e implementación del Módulo ASES, herramienta que utiliza un sistema de información basado en la metodología de semáforos, monitoreando el riesgo de abandono universitario en tiempo real. Esta herramienta emite alertas tempranas del riesgo estudiantil, permitiendo intervenciones preventivas durante los primeros cuatro semestres. Las conclusiones destacan la efectividad de ASES en mejorar la permanencia y graduación de los estudiantes mediante un seguimiento integral, el cual incluye monitores académicos y profesionales de diversas áreas.

En [3] se presenta una herramienta que utiliza analítica de aprendizaje en cursos de programación ofrecidos a través de MOOCs. Esta herramienta proporciona informes mensuales a los profesores, identificando a los estudiantes en riesgo y permitiendo intervenciones tempranas y personalizadas.

Finalmente, en [18] presentan PETIS, herramienta diseñada para cursos de programación en estudiantes de K-12. Esta herramienta permite a los profesores IDENTIFICAR rápidamente las dificultades de los estudiantes y proporcionar retroalimentación inmediata. Las intervenciones están orientadas a mejorar las habilidades de programación y los objetivos de aprendizaje. La tabla 1 presenta las herramientas descritas anteriormente.

TABLA 1. HERRAMIENTAS DE INTERVENCIÓN TEMPRANA IDENTIFICADAS EN LA REVISIÓN DE LA LITERATURA

Herramienta	Tipo de herramienta	Modelo de predicción	Tipos de intervención	Referencia
PreSS#	Herramienta web	Predice el desempeño del estudiante en las primeras semanas de un curso CS1 con una precisión del 71%.	Intervenciones para mejorar el rendimiento académico a partir de tutorías.	[11]
VEAP	Motor de visualización y analizador estadístico de datos	Predice si el estudiante es un programador “débil” o “fuerte”.	Proporciona visualización de datos y análisis para que los profesores intervengan y mejoren el rendimiento de los estudiantes.	[16]
Caring IDE	Entorno de desarrollo integrado (IDE) en la nube	N/A	Proporciona análisis y visualización de datos; permite al profesor realizar intervenciones rápidas y tempranas para mejorar el éxito de los estudiantes.	[17]
Módulo ASES	Sistema de información	Monitorea el riesgo de abandono universitario en tiempo real.	Emite alertas tempranas e intervenciones preventivas durante los primeros cuatro semestres.	[12]
N/A	Plataforma de cursos en línea (MOOCs)	Se basa en la analítica de aprendizaje para identificar estudiantes en riesgo de abandono.	Proporciona informes mensuales a los profesores para intervenir a tiempo los estudiantes en riesgo.	[3]
PETIS	Herramienta de educación	Identificación temprana de dificultades en programación	Proporciona retroalimentación inmediata para mejorar las habilidades de programación.	[18]

Fuente: elaboración propia.

METODOLOGÍA

Herramienta web SPEI

SPEI es una herramienta web de intervención temprana para estudiantes de cursos de programación CS1, incluye un modelo que predice el desempeño académico en las semanas 3, 5 y 7 de un curso de 16 semanas. Asimismo, permite generar intervenciones preventivas como tutorías grupales, envío de sugerencias y código fuente de referencia e intervenciones proactivas con el envío y seguimiento de ejercicios de refuerzo de programación.

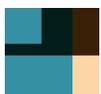
Esta herramienta está diseñada para profesores y estudiantes. Los profesores pueden generar las intervenciones y realizar seguimiento a través de reportes que les permiten tomar decisiones sobre el desempeño académico del curso o un estudiante en particular. Por otro lado, los estudiantes pueden generar reportes de sus actividades para comparar su rendimiento académico con el promedio del curso.

Las entradas de SPEI incluyen las calificaciones de los laboratorios, el tiempo utilizado por el estudiante en sus actividades de programación y la cantidad de intentos de envío. Las calificaciones son cargadas por el profesor mediante un archivo plano con extensión .csv, mientras que el tiempo y la cantidad de intentos se extraen de una herramienta de evaluación automática de código fuente denominada INGIInious M-IDEA [19], que utiliza casos de pruebas definidos por el profesor para evaluar las entregas de programación del estudiante. Todos estos registros se almacenan en la base de datos de la herramienta web SPEI.

SPEI ofrece varias salidas: en el módulo de intervención preventiva se encuentra clasificación del desempeño, la intervención con sugerencias, la intervención con código fuente y el reporte de asistencia de las intervenciones. En el módulo de intervención proactiva se puede enviar ejercicios de refuerzo de programación y generar un reporte de seguimiento del estudiante intervenido.

Metodología de desarrollo

El desarrollo de SPEI se ha gestionado bajo la metodología ágil Scrum; esto ha permitido un enfoque iterativo e incremental para la herramienta. El equipo de trabajo está conformado por desarrolladores de software y docentes, quienes trabajan en sprints de dos semanas. En cada sprint se definen las tareas claves para mejorar las funcionalidades y se garantiza una retroalimentación continua de los usuarios (profesores y estudiantes). Esta metodología ha facilitado la rápida adaptación a los cambios y las mejoras basadas en las necesidades de los usuarios.

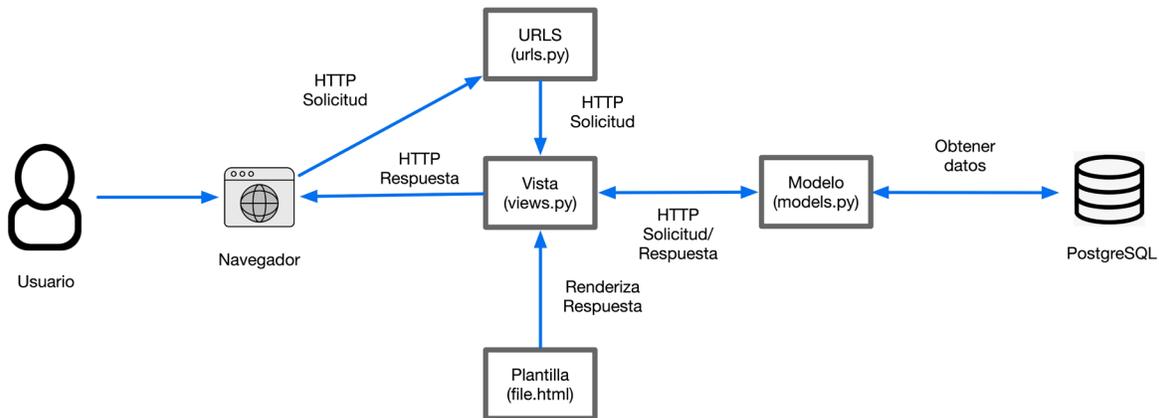


Proceso de pruebas y retroalimentación

SPEI ha sido sometido a un riguroso proceso de pruebas continuas, empleando pruebas unitarias y de integración para garantizar el correcto funcionamiento del modelo predictivo y los módulos de intervención temprana. Además, se han realizado pruebas de usabilidad con un grupo selecto de profesores y estudiantes con el fin de optimizar la experiencia de usuario (UX). Las recomendaciones derivadas de estas pruebas se han incorporado en los *sprints* de desarrollo, siguiendo la metodología Scrum.

Arquitectura de SPEI

La herramienta web SPEI sigue el patrón Model-Template-View (MTV) bajo el *framework* Django, lo que facilita la separación de la lógica de negocio, la interfaz de usuario y el acceso a los datos. Esta estructura no solo mejora la mantenibilidad del sistema, sino que también permite su rápida escalabilidad. El uso de PostgreSQL como base de datos relacional, junto con la librería *psycopg2*, garantiza la fiabilidad de las transacciones y el manejo eficiente de grandes volúmenes de datos. Además, Django proporciona herramientas que agilizan el desarrollo seguro y eficiente de las funcionalidades relacionadas con la predicción y las intervenciones (ver figura 1). El código fuente de SPEI está disponible en el siguiente enlace: <https://github.com/jose-llanos/SPEI-production.git>



Fuente: elaboración propia.

FIGURA 1. ARQUITECTURA DE LA HERRAMIENTA WEB SPEI

Planes de mejora y expansión

El equipo de desarrollo tiene como objetivo ampliar las funcionalidades de la herramienta SPEI, incorporando nuevos módulos que permiten a los profesores persona-

lizar las intervenciones preventivas y proactivas. Asimismo, se están considerando mejoras en el modelo predictivo para integrar nuevas métricas y análisis basados en técnicas de aprendizaje automático. Esto proporcionará a los docentes una visión más precisa y detallada del desempeño académico de los estudiantes.

Modelo de predicción

El modelo implementado en la herramienta SPEI es el propuesto por [20]. Este modelo predice el desempeño académico de los estudiantes durante las primeras semanas de un curso de programación CS1 de 16 semanas. Para su construcción se siguió la metodología CRISP-DM [21], que incluye las fases de comprensión, preparación de datos, modelado y evaluación. En la fase de *comprensión*, se recolectaron los datos, se realizó una exploración inicial y se observaron las relaciones entre las variables. En la fase de *preparación*, se aplicó la técnica de extracción, transformación y carga (ETL, por sus siglas en inglés) con el objetivo de limpiar, formatear e integrar los registros en el conjunto de datos que se utilizará en el modelo de predicción. Las principales tareas realizadas en esta fase fueron: identificar y eliminar los registros NaN, transformar la calificación del estudiante en la salida multiclase del modelo (0 - bajo desempeño, 1 - medio desempeño, 2 - alto desempeño), aplicar la técnica de sobremuestreo para equilibrar los datos y evitar problemas de *overfitting* o *underfitting* en los resultados del modelo. El conjunto de datos final incluyó 3744 registros de 486 estudiantes de los cursos de Programación CS1, impartidos durante los semestres 2021-1, 2021-2 y 2022-1 en la Universidad del Valle (Colombia).

En la fase de *modelado*, se utilizaron siete algoritmos de clasificación: Naive Bayes (NB), Support Vector Classification (SVC), Decision Tree Classifier (DTC), Random Forest Classifier (RFC), K-Neighbors Classifier (KNC), Multi-layer Perceptron Classifier (MLP) y Gradient Boosting Classifier (GBC), los cuales se entrenaron con el 80 % de los registros del conjunto de datos final y se evaluaron con el 20 % restante. Para predecir el desempeño de los estudiantes, se utilizaron características relacionadas con la calificación, el tiempo empleado por los estudiantes en las entregas de programación y el número de envíos. En todos los algoritmos se aplicaron las técnicas de selección de mejores características (*Best Features*), búsqueda en cuadrícula (*Grid Search*) y validación cruzada de 10 pliegues. Cada algoritmo fue entrenado con los mejores hiper-parámetros, generando nuevas predicciones.

Finalmente, en la fase de *evaluación* del modelo, se emplearon las métricas de *precision*, *recall* y *F1 score*. Las pruebas realizadas arrojaron los valores correspondientes para estas métricas en las semanas 3, 5 y 7. En la semana 3, GBC obtuvo los resultados más altos para la métrica *F1 score*, con un valor de 0.86. En la semana 5, RFC y GBC alcanzaron un valor de 0.94 para *F1 score*, mientras que en la semana 7, GBC logró un valor de 0.96 en la misma métrica (ver tabla 2).

TABLA 2. RESULTADO DE LOS ALGORITMOS DE PREDICCIÓN EN LAS SEMANAS 3, 5 Y 7

Semana	Métrica	NB	SVC	DTC	KNC	MLP	RFC	GBC
3	Precision	0.60	0.75	0.78	0.80	0.77	0.84	0.86
	Recall	0.63	0.75	0.78	0.81	0.78	0.84	0.87
	F1 score	0.61	0.75	0.77	0.79	0.77	0.83	0.86
5	Precision	0.67	0.90	0.88	0.91	0.86	0.94	0.94
	Recall	0.68	0.88	0.86	0.91	0.84	0.94	0.94
	F1 score	0.67	0.88	0.86	0.91	0.84	0.94	0.94
7	Precision	0.76	0.95	0.91	0.93	0.90	0.95	0.96
	Recall	0.76	0.94	0.91	0.93	0.89	0.95	0.96
	F1 score	0.76	0.94	0.91	0.93	0.89	0.95	0.96

Fuente: basada en [20].

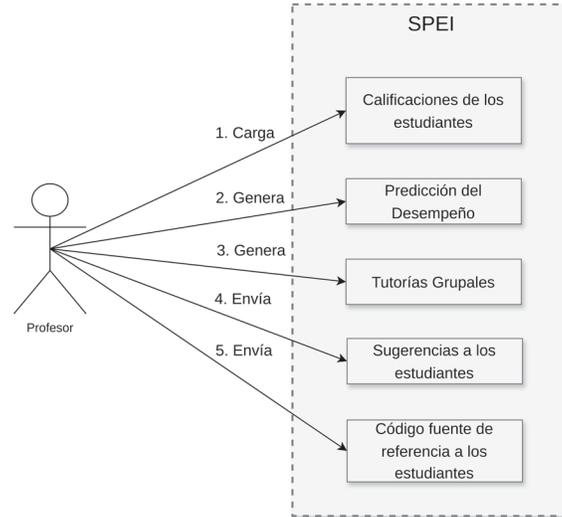
Intervención preventiva

En esta intervención, el profesor ingresa a la opción *Subir Registros Intervención Preventiva* de SPEI y carga un archivo .csv con las calificaciones de los estudiantes (laboratorio 1 y laboratorio 2). Luego selecciona el curso, el semestre y genera la predicción en la opción *Clasificación Desempeño*. Automáticamente, SPEI genera un reporte de los estudiantes con bajo o medio desempeño académico que deben ser intervenidos.

Posteriormente, el profesor realiza la *tutoría grupal*, seleccionando el curso, el semestre, el tema, la fecha, la hora y el lugar de la tutoría. Con esta información SPEI genera una invitación que se envía automáticamente al correo electrónico de todos los estudiantes del curso.

La intervención con sugerencia consiste en enviar una sugerencia o recomendación al estudiante con bajo o medio desempeño académico. En este formulario, el profesor ingresa el curso, el semestre, el indicador de logro y la sugerencia. Luego hace clic en el botón «guardar» y SPEI envía automáticamente un correo electrónico a todos los estudiantes intervenidos.

Finalmente, en la intervención con código fuente, el profesor ingresa el curso, el semestre, el mensaje, la actividad y el código fuente de referencia a partir de una plantilla. Esta plantilla incluye el análisis del problema con sus respectivas entradas y salidas, el pseudocódigo, dos o tres pruebas de escritorio y la solución del problema en C++. Después, el profesor hace clic en «guardar» y SPEI envía automáticamente un correo electrónico a los estudiantes intervenidos con la información descrita. Todo el proceso descrito se encuentra en la figura 2.



Fuente: elaboración propia.

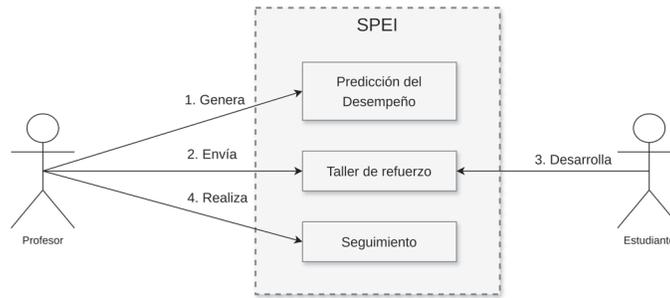
FIGURA 2. PROCESO DE INTERVENCIÓN PREVENTIVA

Intervención proactiva

En este tipo de intervención, el profesor ingresa a la opción *Predicción del Desempeño* y hace clic en el botón «predecir». Luego SPEI genera un reporte con el nombre del estudiante, la calificación promedio y la nota final de la predicción, ordenados según el desempeño: rojo para los estudiantes con bajo desempeño, *naranja* para los estudiantes con desempeño medio y *verde* para los estudiantes con alto desempeño.

Posteriormente, el profesor envía un taller de refuerzo de programación a los estudiantes con bajo y medio desempeño. El taller incluye los objetivos del refuerzo, la metodología para resolver problemas, ejercicios resueltos paso a paso sobre el tema por reforzar y dos problemas de programación que deben entregarse en la herramienta INGIInious M-IDEA.

Finalmente, el profesor realiza el seguimiento, que consiste en generar un informe desde SPEI. Este informe muestra el nombre del estudiante, el taller de refuerzo enviado, la calificación promedio, los ejercicios de refuerzo desarrollados y entregados, la calificación obtenida en la predicción y el desempeño alcanzado tras la intervención. El proceso descrito se encuentra en la figura 3.



Fuente: elaboración propia.

FIGURA 3. PROCESO DE INTERVENCIÓN PROACTIVA

Participantes

En este trabajo se seleccionó un curso de programación CS1 de la Escuela de Ingeniería de Sistemas y Computación de la Universidad del Valle del semestre 2022-2. El curso se denomina «Fundamentos de Programación» (FDP), se dicta en el primer semestre y está conformado por 38 estudiantes del programa de Ingeniería Electrónica, de los cuales 8 son mujeres y 30 hombres. En el curso se adquieren competencias de programación relacionadas con estructuras de decisión, estructuras iterativas, funciones, arreglos y una metodología para resolver problemas en el lenguaje de programación C++.

Conjunto de datos

El conjunto de datos está compuesto por 646 registros correspondientes a tres laboratorios desarrollados por los estudiantes durante el curso de FDP. El laboratorio 1, que se lleva a cabo en la semana 3, incluye cuatro preguntas teóricas y dos ejercicios de programación, tiene un valor del 7.68 % de la nota final y evalúa la implementación de un algoritmo para resolver un problema a partir de entradas y salidas. El laboratorio 2, realizado en la semana 5, incluye tres preguntas teóricas y dos ejercicios de programación, representa el 6 % de la calificación final del curso y evalúa el uso de algoritmos con estructuras de decisión para resolver un problema determinado. Finalmente, el laboratorio 3, que se realiza en la semana 7, incluye cuatro preguntas teóricas y dos ejercicios de programación, tiene un peso del 12 % en la calificación final y evalúa el uso de algoritmos con estructuras de repetición para resolver un problema determinado (ver tabla 3).

TABLA 3. ACTIVIDADES Y NÚMERO DE REGISTROS DEL CONJUNTO DE DATOS

Semana	Actividad	# Registros
3	Laboratorio 1	228
5	Laboratorio 2	190
7	Laboratorio 3	228

Fuente: elaboración propia.

Proceso de intervención implementado

El proceso de intervención temprana implementado se basó en la propuesta de [22], que define un tratamiento con tutorías grupales y la predicción del desempeño académico para realizar el seguimiento del estudiante durante el curso. Asimismo, se incorporó la propuesta de [23], que contempla el envío de notificaciones personalizadas al estudiante, incluyendo recomendaciones de código fuente.

La estrategia de intervención temprana implementada consta de varios pasos. En primer lugar, se utilizó la herramienta SPEI para predecir el desempeño académico de los estudiantes en las semanas 3 y 5 del curso. Con esta información, el profesor organizó tutorías grupales presenciales y envió a los correos electrónicos de los estudiantes intervenidos sugerencias y código fuente de referencia, basados en las entregas de programación. Todas estas actividades se llevaron a cabo entre las semanas 5 y 7.

En la semana 7 se realizó nuevamente la predicción del desempeño académico; el profesor gestionó desde SPEI el envío de ejercicios de refuerzo de programación para los estudiantes con bajo y medio desempeño académico. Entre las semanas 10 y 12 se realizó el seguimiento de los ejercicios de refuerzo, y en la semana 13 se generó un reporte de seguimiento sobre los ejercicios desarrollados por los estudiantes intervenidos.

La figura 4 ilustra el procedimiento implementado en el curso junto con la intervención temprana. En ella se menciona el nombre de la actividad, la intervención preventiva o proactiva implementada y las herramientas utilizadas. Las actividades en color verde corresponden al pretest y postest del cuestionario MSLQ-Colombia, que consta de 19 preguntas sobre estrategias de aprendizaje y 17 preguntas sobre motivación. Cada ítem se califica en una escala Likert de 1 a 7, donde 1 indica que la afirmación no lo describe en absoluto y 7 que la afirmación lo describe completamente.

Las actividades en color gris corresponden a los laboratorios y predicciones generadas a lo largo del curso. Los laboratorios se realizaron en parejas en las semanas 3,

5, 7 y 11. Las actividades resaltadas en color naranja claro representan los exámenes y opcionales realizados en el curso. Las actividades en amarillo hacen referencia al desarrollo de los ejercicios de refuerzo, y la actividad en color azul corresponde a la presentación del proyecto final.

Actividad	Pretest MSLO	Laboratorio 1, Predicción Semana 3	Desarrollo de ejercicios	Laboratorio 2, Predicción Semana 5	Desarrollo de ejercicios	Laboratorio 3, Predicción Semana 7	Examen 1	Opcional 1	Desarrollo ejercicios de refuerzo	Laboratorio 4	Desarrollo ejercicios de refuerzo	Postest MSLO	Examen 2	Opcional 2	Presentación proyecto final
Intervención Preventiva	N/A	N/A	Tutoría grupal, envío de sugerencias y código fuente de referencia (Lab 1) al correo del estudiante	Tutoría grupal	Tutoría grupal, envío de sugerencias y código fuente de referencia (Lab 2) al correo del estudiante	Tutoría grupal	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Intervención Proactiva	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Envío ejercicios de refuerzo por indicador de logo	Primer seguimiento ejercicios de refuerzo	N/A	Segundo seguimiento ejercicios de refuerzo	Resultados seguimiento ejercicios de refuerzo	N/A	N/A	N/A
Herramienta	Google Form	Moodle INGINIOUS SPEI	SPEI	Moodle INGINIOUS SPEI	SPEI	Moodle INGINIOUS SPEI	Moodle INGINIOUS	Moodle INGINIOUS SPEI	INGINIOUS SPEI	Moodle INGINIOUS	INGINIOUS SPEI	Google Form SPEI	Moodle INGINIOUS	Moodle INGINIOUS	Moodle
Semana	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Fuente: elaboración propia basada en [22].

FIGURA 4. PROCEDIMIENTO IMPLEMENTADO EN EL CURSO JUNTO CON LA INTERVENCIÓN TEMPRANA

Evaluación de la herramienta SPEI

Para evaluar la herramienta SPEI se diseñó un cuestionario en Google Forms de seis preguntas con escala likert de 1 a 5, donde 1 indica que está totalmente en desacuerdo y 5 totalmente de acuerdo. El instrumento evalúa las tutorías grupales desarrolladas durante la intervención, las sugerencias y código fuente enviado al correo electrónico del estudiante, los ejercicios de refuerzo desarrollados y la estrategia de intervención como apoyo al proceso de aprendizaje. Los resultados de la evaluación se presentan en la sección de resultados y discusión.

RESULTADOS Y DISCUSIÓN

Para dar respuesta a la pregunta de investigación planteada: ¿Cómo contribuye una herramienta de intervención temprana a mejorar el desempeño académico de los estudiantes en un curso de programación CS1?, se presentan los resultados obtenidos en el curso de FDP seleccionado.

Módulo de predicción

Después de cargar los registros se procedió a generar la predicción para el curso de FDP. La primera predicción se realizó en la semana 3 y en los resultados se observa que el modelo identificó 9 estudiantes por intervenir, de los cuales 7 presentaron

bajo desempeño académico y 2 desempeño medio (ver figura 5). Teniendo en cuenta que el grupo es de 38 estudiantes, la herramienta SPEI sugiere intervenir al 23.6 % de los estudiantes, de los cuales el 18.4 % corresponde a desempeño bajo y el 5.2 % a desempeño medio.

Predicción del Desempeño del Estudiante (Clasificación)

Curso * Campo para seleccionar el curso
Fundamentos de Programación

Semestre * Campo para seleccionar el semestre
2022-2

Semana Predicción * Campo para seleccionar la semana de la predicción
3

Botón para generar la predicción del desempeño
Predice

Predicción del desempeño según el modelo que incluye SPEI

Show 50 entries

#	Nombre	Lab 1	Tiempo de Entrega Lab1 (Días)	Número Intentos Lab1	Resultado Lab1	Lab 2	Predicción
1	JAIR ANDRES SOLARTE ORTIZ	0.0	0.00	0	Sin envío	0.0	0 - Bajo Desempeño
2	JUAN EDUARDO PEDROZA CRIOLLO	0.0	0.00	0	Sin envío	0.0	0 - Bajo Desempeño
3	JUAN FELIPE CHAPAL NARVAEZ	0.0	0.00	0	Sin envío	0.0	0 - Bajo Desempeño
4	FARLEY STIVEN ANDRADE MORALES	0.0	0.00	0	Sin envío	5.0	0 - Bajo Desempeño
5	DANNA ARIANA GUZMAN RIVAS	0.0	0.00	0	Sin envío	0.0	0 - Bajo Desempeño
6	HANNER YEFREO CHIRAN TAIMAL	1.7	0.00	0	Sin envío	3.8	0 - Bajo Desempeño
7	JAIDER ANDRES RODRIGUEZ GUERRERO	2.5	4.73	6	Fallo	5.0	1 - Medio Desempeño
8	DARWIN OBREGON CAMPAZ	2.9	0.00	0	Sin envío	5.0	0 - Bajo Desempeño
9	MIGUEL ANGEL CAMACHO DAZA	3.3	10.94	22	Fallo	5.0	1 - Medio Desempeño

Showing 1 to 9 of 9 entries

Previous 1 Next

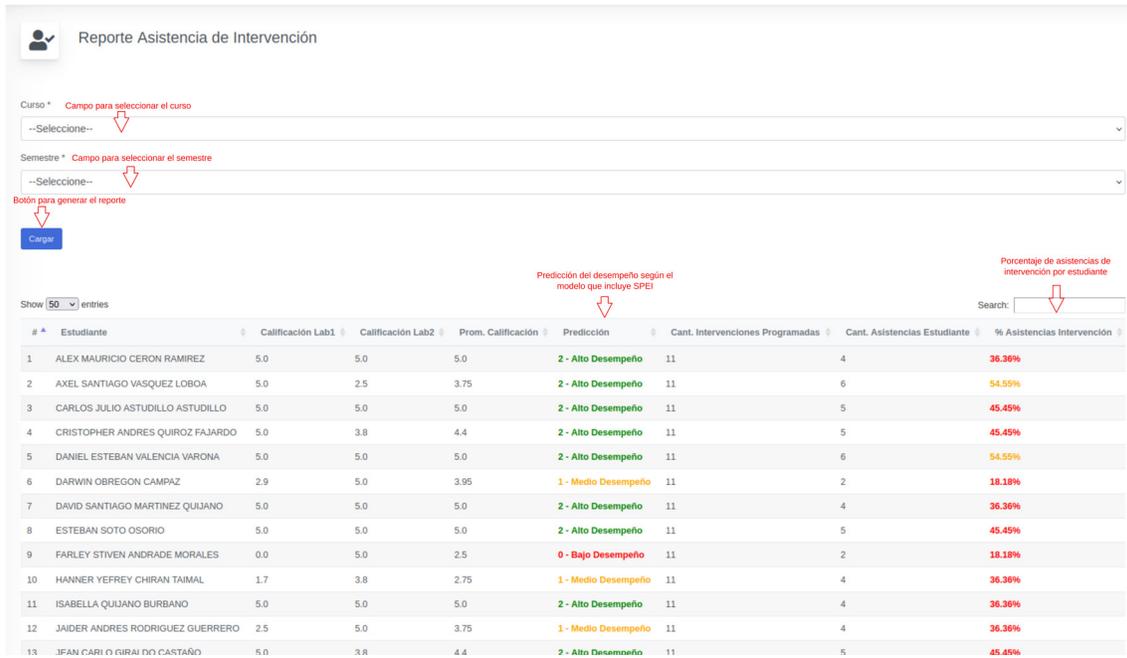
Fuente: elaboración propia.

FIGURA 5. PREDICCIÓN DEL DESEMPEÑO DEL ESTUDIANTE.

Luego se realizó la predicción para la semana 5; en esta, herramienta SPEI identificó 13 estudiantes por intervenir, de los cuales 5 (el 13 %) corresponden a desempeño bajo y 8 (el 21 %) a desempeño medio.

Módulo de intervención preventiva

La primera intervención preventiva que se realizó fue la tutoría grupal. En la figura 6 se observa un reporte que incluye el nombre del estudiante, la calificación de los laboratorios 1 y 2, el promedio de la calificación, la predicción del modelo, la cantidad de intervenciones programadas y las asistencias de los estudiantes. El promedio de asistencia fue de 3.8, lo que indica que los estudiantes asistieron al 34.59 % de las tutorías programadas.



Fuente: elaboración propia.

FIGURA 6. REPORTE DE ASISTENCIAS A TUTORÍAS GRUPALES

Asimismo, durante el semestre académico se enviaron tres sugerencias al correo electrónico de los estudiantes. Estas sugerencias mencionan el indicador de logro evaluado, las falencias IDENTIFICADAS en los ejercicios de programación enviados y recomendaciones sobre buenas prácticas de programación para la definición de variables, estructuras de decisión y estructuras de repetición.

También se realizaron tres intervenciones con código fuente, una después de cada uno de laboratorios 1, 2 y 3. Este tipo de intervención incluye la solución del laboratorio propuesto, con buenas prácticas de programación y código fuente en C++.

Módulo de intervención proactiva

En la intervención proactiva se envió un taller de refuerzo de programación que incluye cinco ejercicios para los estudiantes con desempeño académico bajo y medio, y se realizó el respectivo seguimiento. Los ejercicios están diseñados para reforzar los temas de entradas y salidas, estructuras de decisión y estructuras de repetición.

Luego SPEI identifica el desempeño del estudiante y envía el taller de refuerzo al correo electrónico. Asimismo, genera un informe que incluye el nombre del estudiante, las calificaciones obtenidas en los laboratorios, la predicción de la nota final, el desempeño académico del estudiante y el taller de refuerzo enviado (ver figura 7).

Refuerzo

Curso * **Campo para seleccionar el curso**
Fundamentos de Programación

Semestre * **Campo para seleccionar el semestre**
2022-2

Botón para buscar los refuerzos enviados
Buscar

Mostrar 50 registros

#	Nombre	Laboratorio 1 (7.68%)	Laboratorio 2 (12%)	Laboratorio 3 (6%)	Nota Final (Predicción)	Desempeño	Taller de Refuerzo
1	JAIR ANDRES	0.0	0.0	0.0	0.0	Bajo	1- Guía Asistida
2	DANNA ARIANA	0.0	0.0	0.0	0.0	Bajo	1- Guía Asistida
3	JUAN EDUARDO	0.0	0.0	0.0	0.0	Bajo	1- Guía Asistida
4	JUAN FELIPE	0.0	0.0	0.0	0.0	Bajo	1- Guía Asistida
5	FARLEY STIVEN	0.0	5.0	2.5	2.0	Bajo	1- Guía Asistida
6	DARWIN	2.9	5.0	0.0	2.8	Bajo	1- Guía Asistida
7	LUIS FERNANDO	5.0	2.5	0.0	2.8	Bajo	1- Guía Asistida
8	MAICOL	5.0	2.5	2.5	3.0	Medio	2- Pseudocódigo
9	JAIDER ANDRES	2.5	5.0	0.0	3.1	Medio	2- Pseudocódigo

Fuente: elaboración propia.

FIGURA 7. REPORTE DE TALLERES DE REFUERZO DE PROGRAMACIÓN

Finalmente, la herramienta genera un reporte de seguimiento que muestra el nombre del estudiante, el taller de refuerzo enviado, la calificación promedio, la predicción de la nota final, el promedio de la intervención y el desempeño alcanzado. Además, detalla el estado de cada ejercicio enviado, incluyendo el número de intentos, la fecha de creación y la fecha de entrega (ver figura 8).

Intervención

Cantidad de talleres de refuerzo desarrollados
Taller Refuerzo

Calificación final según modelo implementado en SPEI
Nota Final (Predicción)

Promedio alcanzado con la intervención
Promedio Intervención

Desempeño alcanzado según la intervención
Desempeño Intervención

id	Semestre	Nombre Estudiante	Taller Refuerzo	Nota Promedio	Nota Final (Predicción)	Promedio Intervención	Desempeño Intervención
5	2	FARLEY STIVEN	1	2.5	2.0	100	Alto

Detalle Intervención

Ejercicios del taller enviados
Ejercicio

Número de intentos por ejercicios
Número de Intentos

Fecha de entrega del ejercicio de refuerzo
Fecha Entrega

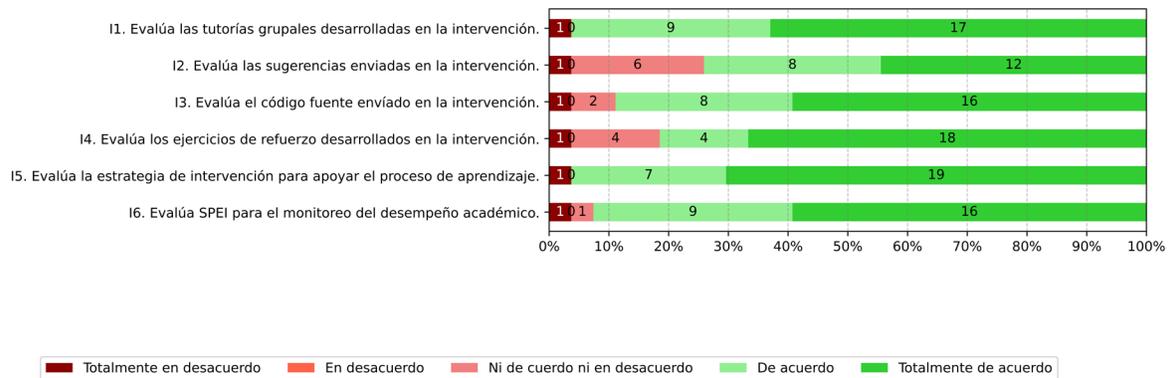
Ejercicio	Estado	Número de Intentos	Fecha Creacion	Fecha Entrega	Fecha Corte
3	100	5	Oct. 13, 2022	Oct. 13, 2022	Oct. 13, 2022
7	100	3	Oct. 12, 2022	Oct. 12, 2022	Oct. 13, 2022
12	100	3	Nov. 3, 2022	Nov. 3, 2022	Nov. 3, 2022

Fuente: elaboración propia.

FIGURA 8. REPORTE DE SEGUIMIENTO DEL ESTUDIANTE

Aplicación del instrumento de evaluación a la herramienta SPEI

La figura 9 presenta de forma agrupada los resultados del cuestionario de evaluación de la herramienta SPEI, respondido por 27 estudiantes del curso FDP. Se observa que en la mayoría de las preguntas los estudiantes evaluaron estar totalmente de acuerdo con los ítems evaluados. En los ítems I2, I3 e I4 varios estudiantes seleccionaron la opción *ni de acuerdo ni en desacuerdo*; esto ocurrió porque la intervención con sugerencias, código fuente de referencia y envío del taller de refuerzo solo se realizó a los estudiantes intervenidos. Finalmente, un estudiante del grupo evaluó todos los ítems con la opción *totalmente en desacuerdo*.



Fuente: elaboración propia.

FIGURA 9. RESULTADOS DEL CUESTIONARIO DE EVALUACIÓN DE LA HERRAMIENTA SPEI

A diferencia de [15], que utiliza una instrucción general y evaluación temprana en la intervención preventiva para identificar dificultades en el aprendizaje, nosotros proponemos tres tipos de intervención preventiva para el profesor, basados en [3], [13]. La primera es la tutoría grupal, la segunda consiste en el envío de sugerencias automáticas al correo electrónico del estudiante y la tercera es la intervención con código fuente de referencia.

En [15], los autores proponen programas de prevención para los estudiantes que no presentan mejoría. En este trabajo proponemos el envío de un taller de refuerzo de programación para los estudiantes con desempeño académico bajo o medio y el seguimiento de sus entregas. Esto permite observar los intentos generados y motivar al estudiante en el desarrollo de los ejercicios propuestos, que están diseñados para fortalecer los indicadores de logro evaluados.

CONCLUSIONES

SPEI demostró que las intervenciones generadas, como las tutorías grupales, el envío de sugerencias y la intervención con código fuente de referencia, pueden adaptarse a las necesidades individuales de los estudiantes, mejorando significativamente su rendimiento en temas relacionados con definición de variables, estructuras de decisión y estructuras de repetición.

El seguimiento detallado que ofrece SPEI incluye el análisis de los intentos, fechas de entrega y desempeño en ejercicios específicos. Esto permite a los profesores tomar decisiones informadas sobre las necesidades de sus estudiantes y generar intervenciones más efectivas.

Como trabajo futuro se pretende evaluar la herramienta en otros cursos de programación CS1 con el fin de comparar los resultados obtenidos en este trabajo con los de otros cursos. De esta forma se puede apoyar el proceso formativo y mejorar el desempeño académico de los estudiantes en cursos de programación CS1.

REFERENCIAS

- [1] D. Buenaño-Fernández, D. Gil y S. Luján-Mora, “Application of Machine Learning in Predicting Performance for Computer Engineering Students: A Case Study”, *Sustainability*, vol. 11, n°. 10, p. 2833, 2019. doi: 10.3390/su11102833.
- [2] A. F. ElGamal, “An Educational Data Mining Model for Predicting Student Performance in Programming Course”, *IJCA*, vol. 70, n°. 17, pp. 22-28, 2013. doi: 10.5120/12160-8163.
- [3] O. Lu, J. Huang, A. Huang y S. Yang, “Applying learning analytics for improving students engagement and learning outcomes in an MOOCs enabled collaborative programming course”, *Interactive Learning Environments*, pp. 78-92, 2018. doi: 10.1080/10494820.2016.1278391.
- [4] M. Sivasakthi, “Classification and prediction based data mining algorithms to predict students’ introductory programming performance”, en *2017 International Conference on Inventive Computing and Informatics (ICICI)*, Coimbatore: IEEE, nov. 2017, pp. 346-350. doi: 10.1109/ICICI.2017.8365371.
- [5] F. D. Pereira, S. C. Fonseca, E. H. T. Oliveira, D. B. F. Oliveira, A. I. Cristea y L. S. G. Carvalho, “Deep learning for early performance prediction of introductory programming students: a comparative and explanatory study”, *RBIE*, vol. 28, pp. 723-748, 2020. doi: 10.5753/rbie.2020.28.o.723.
- [6] M. Kuehn, J. Estad, J. Straub, T. Stokke y S. Kerlin, “An expert system for the prediction of student performance in an initial computer science course”, en *2017 IEEE International*

- Conference on Electro Information Technology (EIT). Lincoln, NE, USA: IEEE, 2017, pp. 1-6. doi: 10.1109/EIT.2017.8053321.
- [7] S. U. Hassan, M. Imran, U. Gillani, N. R. Aljohani, T. D. Bowman y F. Didegah, “Measuring social media activity of scientific literature: an exhaustive comparison of scopus and novel altmetrics big data”, *Scientometrics*, vol. 113, n.º. 2, pp. 1037-1057, 2017. doi: 10.1007/s11192-017-2512-x.
- [8] R. Vilanova et al., “Data-driven tool for monitoring of students performance”, *IFAC-PapersOnLine*, vol. 52, n.º. 9, pp. 165-170, 2019. doi: 10.1016/j.ifacol.2019.08.188.
- [9] O. K. Xin y D. Singh, “Development of Learning Analytics Dashboard based on Moodle Learning Management System”, *IJACSA*, vol. 12, n.º. 7, 2021. doi: 10.14569/IJACSA.2021.0120793.
- [10] C. Lacave, A. I. Molinay J. A. Cruz-Lemus, “Learning Analytics to IDentify dropout factors of Computer Science studies through Bayesian networks”, *Behaviour and Information Technology*, vol. 37, n.º. 10-11, pp. 993-1007, 2018. doi: 10.1080/0144929X.2018.1485053.
- [11] K. Quille y S. Bergin, “CS1: how will they do? How can we help? A decade of research and practice”, *Computer Science Education*, vol. 29, n.º. 2-3, pp. 254-282, 2019. doi: 10.1080/08993408.2019.1612679.
- [12] S. Ramírez-Torres y N. S. Castillo-Flórez, “ALERTAS TEMPRANAS DEL RIESGO DE ABANDONO ESCOLAR: Una estrategia para la permanencia y el éxito estudiantil monitoreado desde las TIC en la Universidad del Valle”, *Repositorio Institucional de la Universidad Tecnológica de Panamá*, p. 8, 2022.
- [13] W. Xing y D. Du, “Dropout Prediction in MOOCs: Using Deep Learning for Personalized Intervention”, *Journal of Educational Computing Research*, vol. 57, n.º. 3, pp. 547-570, junio 2019. doi: 10.1177/0735633118757015.
- [14] J. E. Jiménez et al., “Evaluación del progreso de aprendizaje en lectura dentro de un Modelo de Respuesta a la Intervención (Rtl) en la Comunidad Autónoma de Canarias”, *Escritos de Psicología*, vol. 4, n.º. 2, pp. 56-64, 2011. doi: 10.5231/psy.writ.2011.1207.
- [15] A. R. Andonegi, C. C. Garrido y M. C. Pérez, “Eficacia de un programa de intervención temprana para reducir las señales de riesgo de la dislexia”, *Revista de Educación Inclusiva*, vol. 9, n.º. 2, pp. 186-200, 2017.
- [16] N. Culligan, K. Quille y S. Bergin, “VEAP: a visualisation engine and analyzer for preSS#”, in *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, en Koli Calling '16. New York, NY.: Association for Computing Machinery, Nov. 2016, pp. 130-134. doi: 10.1145/2999541.2999553.
- [17] M. Dorodchi, M. Fallahian, E. Al-Hossami, A. Benedict y A. Benedict, “Custom caring IDE for online offering of CS1”, *Sixth SPLICE workshop at L@S*, 2020.

- [18] H.-Y. Lee, C.-J. Lin, W.-S. Wang, W.-C. Chang y Y.-M. Huang, “Precision education via timely intervention in K-12 computer programming course to enhance programming skill and affective-domain learning objectives”, *IJ STEM Ed*, vol. 10, n.º. 1, p. 52, agosto 2023. doi: 10.1186/s40594-023-00444-5.
- [19] C. G. Hidalgo S. y V. A. Bucheli, “Modelo soportado en inteligencia artificial para el desarrollo de actividades de aprendizaje activo basadas en colaboración asistida por computador (MIDEA)”, *LACCEI International Multi-Conference for Engineering, Education, and Technology*, vol. 17, pp. 1-3, 2019.
- [20] J. Llanos, V. A. Bucheli y F. Restrepo-Calle, “Early prediction of student performance in CS1 programming courses”, *PeerJ Computer Science*, n.º. 9, pp. 1-29, 2023. doi: 10.7717/peerj-cs.1655.
- [21] C. Schröer, F. Kruse y J. M. Gómez, “A Systematic Literature Review on Applying CRISP-DM Process Model”, *Procedia Computer Science*, vol. 181, pp. 526-534, 2021. doi: 10.1016/j.procs.2021.01.199.
- [22] J. M. L. Mosquera, J. Á. V. IturbIDE, M. P. Velasco y V. A. B. Guerrero, “Assessment of a Predictive Model for Academic Performance in a Small-Sized Programming Course”, en *2024 International Symposium on Computers in Education (SIIE)*, La Coruña, España: IEEE, Jun. 2024, pp. 1-6. doi: 10.1109/SIIE63180.2024.10604641.
- [23] D. Azcona, I.-H. Hsiao y A. Smeaton, “An Exploratory Study on Student Engagement with Adaptive Notifications in Programming Courses”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11082 LNCS, pp. 644-647, 2018. doi: 10.1007/978-3-319-98572-5_64.