Modelo predictivo del rendimiento académico para identificar estudiantes universitarios con alto riesgo de abandono en cursos de fundamentos de programación

Isis Karina Antolinez Ramírez
Oscar Emmanuel Antolinez Ramírez

Documento presentado para optar por el título de Ingeniero de Sistemas

Director:
Mgtr. José Miguel Llanos Mosquera

Codirector:
Mgtr. Julián Andrés Quimbayo Castro

Corporación Universitaria del Huila – CORHUILA
Facultad de Ingeniería
Programa de Ingeniería en Sistemas
2025

Cita	(Antolinez & Antolinez, 2025)
Referencia	Antolinez, I. K., & Antolinez, O. E. (2025). Modelo predictivo del rendimiento académico para identificar estudiantes universitarios con alto riesgo de abandono en cursos de fundamentos de programación. Corporación Universitaria del Huila - CORHUILA.
Según normas APA 7ª edición	

Grupo de Investigación INPROTI Código COL0183316.
Semillero de Investigación Mamba.

Repositorio Institucional: http://bxxxxxxx

Corporación Universitaria del Huila – CORHUILA https://corhuila.edu.co/

El contenido de este documento se ampara en el derecho de expresión de sus autores y no representa el pensamiento ni la posición institucional de la Corporación Universitaria del Huila – CORHUILA. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Nota de Aprobación

El presente trabajo opción de grado para obtener el título de Ingeniero de Sistemas ha sido revisado y calificado con nota.

APROBADO
(Acuerdo 232 de 2023 del Consejo Académico)

Firma
Álvaro Hernán Alarcón
Docente Tiempo Completo
Jurado

Firma
Edisney García Perdomo
Docente Tiempo Completo
Jurado

"El Director y el Jurado del presente trabajo, no son responsables de las ideas y conclusiones expuestas en éste; ellas son exclusividad de sus autores".

Dedicatoria

A nuestro papá y mamá, por el amor incondicional y por enseñarnos con su ejemplo a luchar por nuestros sueños, a ser perseverante y a no rendirnos ante los desafíos. Gracias por su apoyo.

A nuestras queridas abuelas, quienes con su sabiduría y cariño han sido una fuente de inspiración en cada paso de nuestra vida. Su apoyo y amor nos impulsó a seguir adelante.

A nuestras hermanas, por ser una fuente de motivación, admiración y apoyo para terminar nuestros estudios. Gracias por estar con nosotros en este camino, por sus palabras de aliento y por creer en nosotros.

Agradecimientos

Agradecemos profundamente a nuestros padres, por su amor, apoyo incondicional y enseñarnos el valor del esfuerzo. A nuestras hermanas, por ser nuestra inspiración y respaldo constante.

Expresamos nuestra gratitud al profesor José Miguel Llanos por guiarnos en la creación del proyecto como director de este y por su valioso apoyo en el desarrollo del modelo predictivo. Al profesor Julián Andrés Quimbayo, por su apoyo constante en el proyecto y su ayuda como codirector, cuyo conocimiento y dedicación nos impulsaron a dar lo mejor de nosotros mismos.

Agradecemos también al grupo de investigación INPROTI y al semillero MAMBA por su acompañamiento y contribuciones, que enriquecieron significativamente este trabajo.

Expresamos nuestro más sincero agradecimiento al área de Ciencia, Tecnología e Innovación (CTEI) y a la Corporación Universitaria del Huila - CORHUILA por su invaluable apoyo en el desarrollo de este proyecto de investigación. Su respaldo y compromiso con la educación y la investigación han sido fundamentales para la realización de esta tesis.

Sin ustedes, este logro no habría sido posible.

Resumen

Para enfrentar el desafío del alto abandono estudiantil en los cursos iniciales de programación, esta tesis se enfocó en crear un modelo predictivo para detectar tempranamente a aquellos estudiantes universitarios que podrían abandonar un curso de Fundamentos de Programación en la Corporación Universitaria del Huila - CORHUILA. Como punto de partida, utilizamos una versión adaptada del Cuestionario de Motivación y Estrategias de Aprendizaje (MSLQ-Colombia) para comprender mejor cómo estudian y qué motiva a los alumnos. En el desarrollo del proyecto aplicamos la metodología estándar conocida como CRISP-DM para organizar todo el proceso de minería de datos, desde entender la necesidad al igual que evaluar los resultados del modelo y, por otro lado, realizamos un estudio cuasi-experimental donde comparamos un grupo de estudiantes que recibió intervenciones educativas basadas en los resultados del modelo de predicción con otro grupo que no las recibió, para ver el impacto en su permanencia y calificaciones.

En la construcción del modelo predictivo, se evaluaron siete algoritmos de clasificación (Naive Bayes, Support Vector Machine, Decision Tree, K-Nearest Neighbor, Multilayer Perceptron, Random Forest y XGBoost) los cuales tenían la labor de predecir si un estudiante tenía una alta, media o baja probabilidad de abandonar teniendo en cuenta los resultados del MSLQ y la calificación obtenida en el primer, segundo y tercer corte del curso. En el modelado se tuvo que limpiar y preparar los datos, balancear la información, seleccionar las características más importantes y ajustar los parámetros de cada modelo para hacerlos lo más precisos posible. De todos los modelos probados, Random Forest fue el que demostró un mejor rendimiento, alcanzando un F1 Score del 74% para la semana cuatro del desarrollo del curso, una medida que indica un buen balance para identificar correctamente a los estudiantes en riesgo de abandono.Esperamos que esta investigación no solo aporte al conocimiento en educación superior y ciencias de la computación, sino que también entregue herramientas concretas a las instituciones para apoyar a sus estudiantes y mejorar el aprendizaje en estas áreas fundamentales.

Palabras clave: Abandono universitario, Fundamentos de programación, CRISP-DM, Cuasi-experimento, Modelo de predicción, Clasificación.

Abstract

To face the challenge of the high dropout rate in the initial programming courses, this thesis focused on creating a predictive model. The objective is to detect early those university students who might drop out of the Programming Fundamentals courses at the Corporación Universitaria del Huila - CORHUILA.

As a starting point, we used an adapted version of the Motivation and Learning Strategies Questionnaire (MSLQ-Colombia) to better understand how students study and what motivates them. The development of the project followed a two-fold path. On the one hand, we applied the standard CRISP-DM methodology to organize the entire data mining process, from understanding the need to evaluating the results of the model. On the other hand, we conducted a quasi-experimental study: we compared a group of students who received educational interventions based on the MSLQ results with another group who did not, to see the impact on their retention and grades.

In building the predictive model, we explored seven different classification algorithms (Naive Bayes, Support Vector Machine, Decision Tree, K-Nearest Neighbor, Multilayer Perceptron, Random Forest and XGBoost). We took special care in preparing the data, balancing the information, selecting the most important features and adjusting the parameters of each model to make them as accurate as possible.

Of all the models tested, Random Forest proved to be the most effective, achieving an F1 Score of 74%, a measure that indicates a good balance between correctly identifying students at risk without generating false alarms. Most importantly, this model allows us to pinpoint the students most likely to drop out as early as the fourth week of the course.

We hope that this research will not only contribute to knowledge in higher education and computer science, but also provide concrete tools for institutions to support their students and improve learning in these fundamental areas.

Keywords: University dropout, Fundamentals of programming, CRISP-DM, Quasi-experiment, Prediction model, Classification.

Tabla de contenido

Lista de tablas

Lista de figuras

Siglas, acrónimos, abreviaturas

APA	American Psychological Association
Et al.	Y otros
P.	Página
Pp.	Páginas
Párr.	Párrafo
MSc.	Magister Scientiae
PhD	Philosophiae Doctor
CORHUILA	Corporación Universitaria del Huila
MSLQ - Colombia	Cuestionario de Motivación y Estrategias de Aprendizaje Colombia
CRISP-DM	Cross-Industry Standard Process for Data Mining
NB	Naive Bayes
SVM	Support Vector Machine
DT	Decision Tree
KNN	K-Nearest Neighbor
MLP	Multilayer Perceptron
RF	Random Forest
XGBoost	Extreme Gradient Boosting
TP	True Positives
FP	False Positives
FN	False Negatives
TPR	True Positives Rate
FPR	False Positives Rate
AUC	Area Under the Curve
ROC	Receiver Operating Characteristics
LMS	Sistemas de Gestión del Aprendizaje
H_0	Hipótesis nula

H_1	Hipótesis alternativa
GE	Grupo Experimental
GC	Grupo de Control
SMOTE	Synthetic Minority Over-sampling Technique
STD	Desviación Estándar
GBM	Gradient Boosting Machines

Introducción

La evolución de la tecnológica y la digitalización han posicionado las competencias en programación como elementos fundamentales en el ámbito de la educación superior y el contexto laboral, convirtiéndose en habilidades imprescindibles para el desarrollo profesional (Mai et al., 2023). Sin embargo, muchos estudiantes se encuentran con dificultades justo al empezar: los cursos introductorios de programación presentan tasas de abandono preocupantemente altas, a menudo entre el 30% y 50% (Verma et al., 2022). Este es un desafío serio para la retención estudiantil, especialmente en Iberoamérica. En Colombia, por ejemplo, la deserción universitaria general alcanzó el 10% durante 2021 y los cursos de programación suelen ser un punto crítico (*SPADIES,* 2021).

Aprender a programar es un paso fundamental en muchas carreras universitarias, a pesar de esto, los cursos introductorios a la programación suelen tener un alto índice de abandono en varias carreras universitarias. Diversos estudios señalan que una de las razones se debe a que los estudiantes perciben el contenido como muy complejo y sienten que no reciben suficiente apoyo por parte de los docentes (Jamjoom et al., 2021; Schefer-Wenzl et al., 2024). La frustración, la falta de motivación y la sensación de estar sobrecargados pueden frenar el avance inicial, haciendo que se desconecten del curso y del ambiente académico (Alsulami et al., 2023; Lakanen & Isomöttönen, 2023). Se trata de un problema complejo que se ve influenciado por una mezcla de factores tanto académicos, socioeconómicos y motivacionales, como la percepción de dificultad, la falta de experiencia previa y el acceso limitado a recursos de ayuda (Schefer-Wenzl et al., 2024).

Considerando esto, la literatura ha analizado a fondo las causas del abandono en carreras de Ciencias de la Computación, coincidiendo en la importancia de identificar a los estudiantes en riesgo de manera temprana (Llanos et al., 2023a). Sin embargo, todavía hace falta investigar a profundidad la creación de modelos predictivos diseñados para prevenir el abandono en estos cursos fundamentales de programación, que son la base de la formación tecnológica (Alboaneen et al., 2022).

Es aquí donde la motivación y las estrategias de aprendizaje de los estudiantes juegan un papel crucial en el desempeño académico. La motivación, que evaluamos con el instrumento MSLQ-Colombia, nos indica cuánto valor aporta a los estudiantes lo que aprenderán en el curso y cuánta confianza tienen en sus propias capacidades (Lakanen & Isomöttönen, 2023). Al mismo tiempo, las estrategias de aprendizaje nos indican como gestionan el tiempo de estudio y regulan su esfuerzo, para así saber cómo afecta directamente en sus resultados académicos (Jamjoom et al., 2021).

Las calificaciones que un estudiante muestra durante en el curso es un reflejo del progreso y rendimiento en el desarrollo del mismo (Llanos et al., 2023), Si consideramos conjuntamente sus estrategias, motivación y calificaciones, podemos diseñar estrategias academicas más personalizadas para prevenir el abandono y mejorar el desempeño (Nabil et al., 2021; Schefer-Wenzl et al., 2024). Estos elementos nos dan una base sólida para construir modelos predictivos que identifiquen a los estudiantes que necesitan apoyo y nos orienten sobre cómo intervenir eficazmente (Llanos et al., 2023).

El propósito central de esta investigación es, precisamente, desarrollar y validar un modelo predictivo del rendimiento académico. Utilizando técnicas avanzadas de aprendizaje automático, buscamos identificar de forma temprana a los estudiantes con riesgo de abandonar los cursos de fundamentos de programación en CORHUILA. Los resultados que obtengamos servirán para diseñar e implementar estrategias de apoyo personalizadas y proactivas. El objetivo final no es solo reducir las tasas de abandono, sino también fomentar la permanencia y asegurar que los estudiantes adquieran las competencias técnicas esenciales durante su formación profesional.

Planteamiento del problema

Hoy en día, saber programar es una habilidad clave, tanto para la vida diaria como para el trabajo. Esto se debe a que vivimos en un mundo cada vez más digital, podemos acceder fácilmente a recursos en línea y hay una constante demanda de nuevas tecnologías como software, páginas web y aplicaciones (Lakanen & Isomöttönen, 2023). Sin embargo, existe una gran falta de profesionales en esta área, según un informe hecho por (La República, 2023), analizando los resultados del informe Talent Gap, que fue realizado en el 2017, hay un total 60.000 puestos de desarrollador sin cubrir en Colombia, y se proyecta que para este año 2025 ese valor aumentara aproximadamente hasta los 153.000. Como consecuencia a esta necesidad, cada vez más estudiantes se inscriben en cursos básicos de programación en universidades de todo el mundo, reconociendo que estas competencias tecnológicas son fundamentales para su futuro académico y profesional (Mai et al., 2023).

Debido al creciente número de estudiantes que ingresan a carreras de tecnología, los cursos de programación enfrentan un gran desafio: altas tasas de abandono y fracaso académico (Kocsis & Molnár, 2024), las cuales oscilan entre el 30% y el 50% respectivamente (SPADIES, 2021). Esta problemática también se presenta a nivel de Iberoamérica, según datos estadísticos de un estudio realizado por la Agencia Unal en 2021, el 66% de los estudiantes iberoamericanos permanecen activos en sus estudios, mientras que el 33% abandonan (Periódico UNAL, 2022). En Colombia la tasa de deserción anual para el 2021 fue del 10.08%, ubicándose 1.22% por encima de la tasa de deserción anual del 8.85% en 2020 (SPADIES, 2021). Explorando la situación en el departamento del Huila, la región se enfrenta al desafío de mejorar el acceso y la permanencia en la educación superior, según datos del Plan de Desarrollo Huila Crece 2020-2023, la tasa de abandono anual es del 8.63% en la educación universitaria para el año 2017 (Gobernación del Huila, 2020). Esta situación no solo limita las oportunidades profesionales de los estudiantes, sino que también afecta la calidad educativa de las instituciones y la eficiencia de las inversiones gubernamentales en la formación de talento humano cualificado (Llanos et al., 2023a; Schefer-Wenzl et al., 2024)

En este contexto, una de las principales causas del abandono en los cursos de programación es la percepción de que los contenidos son excesivamente complejos (Lu et al., 2021). Cuando los estudiantes se sienten frustrados, desmotivados o simplemente abrumados, es fácil que pierdan el interés por aprender programación, lo que conduce al abandonando temprano del curso. Esta situación deja claro que necesitamos enfoques de enseñanza que hagan la materia menos intimidante y que ayuden a mantener alta la motivación de los alumnos (Schefer-Wenzl et al., 2024).

Por otro lado, las implicaciones van más allá del ámbito educativo, a nivel individual, se interrumpe el desarrollo académico y profesional de los estudiantes, afectando su capacidad de competir en un mercado laboral demandante (SPADIES, 2021), siendo este un factor que a nivel institucional, afectan directamente la retención estudiantil y también repercuten negativamente en la reputación de las universidades, comprometiendo además los recursos significativos que estas destinan a la formación de sus estudiantes (Prasanth & Alqahtani, 2023), tal es el caso de la tasa de perdida en cursos de programación en el programa de Ingeniería de Sistemas, durante los últimos dos años, se mantiene por debajo del 30 % (Castro et al., 2019). Este porcentaje es similar a las cifras observadas en Iberoamérica a nivel de abandono, lo que refleja el comportamiento común de los estudiantes frente a los desafíos de comprensión, motivación y dedicación que implica el aprendizaje en este campo.

El objetivo principal de este trabajo es crear un modelo capaz de predecir el rendimiento académico. La idea es poder identificar, lo antes posible, a aquellos estudiantes que corren un mayor riesgo de abandonar los cursos introductorios de programación. Con esto en mente, nos planteamos la siguiente pregunta de investigación: ***¿Cómo puede un modelo predictivo del rendimiento académico identificar a los estudiantes universitarios con alto índice de abandono en cursos de programación?***

Justificación

Dado lo importante que es conocer y tener habilidades en programación, tanto en lo académico como en lo profesional, es fundamental que logremos retener a los estudiantes en los cursos introductorios (Jamjoom et al., 2021). Necesitamos superar este reto, debido a que la programación es la base de la economía digital, es indispensable para desarrollar nuevas tecnologías y para cubrir la enorme necesidad de profesionales en el sector (Alsulami et al., 2023a).

Para lograr que más estudiantes terminen estos cursos, y así cerrar la brecha tecnológica, es necesario utilizar estrategias efectivas contra el abandono. Ayuda a reducir el gran déficit de desarrolladores que existe, que solo en Colombia podría superar los 153.000 en 2025 (La República, 2023). Este esfuerzo no solo prepara mejor a los futuros profesionales, sino que fortalece nuestra capacidad como sociedad para adaptarnos y sacar provecho del desarrollo digital (Schefer-Wenzl et al., 2024).

Una de las estrategias prometedoras es utilizar modelos predictivos y herramientas análisis de datos para detectar de manera temprana a los estudiantes en riesgo de abandono, analizando información clave al inicio y durante el curso (Lakanen & Isomöttönen, 2023). Con esta información, las instituciones pueden diseñar intervenciones a la medida para fomentar la permanencia y el éxito académico (Alsulami et al., 2023).

Este trabajo de investigación aborda directamente el problema del abandono en los cursos de programación. Desarrollamos un modelo predictivo, basado en análisis de datos y aprendizaje automático, para identificar cuanto antes a los estudiantes que podrían necesitar apoyo. Al enfocarnos en esta detección temprana, atacamos una de las causas raíz del abandono: la dificultad de ofrecer ayuda personalizada y a tiempo a quienes enfrentan obstáculos académicos o de motivación (Jamjoom et al., 2021; Schefer-Wenzl et al., 2024).

El valor de este estudio está en su enfoque basado en datos, que usa técnicas avanzadas de aprendizaje automático para crear un modelo capaz de transformar cómo se da apoyo académico en programación (Guzmán-Castillo et al., 2022).

Finalmente, realizar este estudio es viable. Contamos con herramientas y datos cada vez más accesibles para el análisis predictivo en educación, y además existe un claro interés por parte de las instituciones en reducir el abandono y mejorar los resultados de los estudiantes (Lazo Jaque, 2021). Esto garantiza que el proyecto no solo se pueda llevar a cabo, sino que también pueda tener un impacto real y duradero en la calidad educativa y en la formación de los profesionales que tanto necesita este campo (Kocsis & Molnár, 2024).

Objetivos

Objetivo general

Desarrollar un modelo de predicción del rendimiento académico que permita identificar los estudiantes con alto índice de abandono en cursos de programación

Objetivos específicos

- Identificar las características para predecir el rendimiento académico en cursos de programación.
- Construir el modelo de predicción del rendimiento académico a partir de las características seleccionadas.
- Evaluar el modelo de predicción del rendimiento académico mediante un estudio cuasi-experimental que permita identificar los estudiantes con alto índice de abandono en cursos de programación.

Marco referencial

Estado del arte

Debido a la evolución de la tecnología educativa, las técnicas de predicción se han convertido en aliadas importantes para reducir el abandono estudiantil (Lu et al., 2021). En esta línea, la minería de datos educativos (EDM) ha avanzado hasta el punto de poder incorporar algoritmos potentes como clasificadores y redes de aprendizaje profundo (Jamjoom et al., 2021). Se comprobado que estas herramientas son más efectivas para descubrir patrones complejos en los estudiantes, lo que ayuda a predecir mejor su rendimiento y a detectar temprano a quienes están en riesgo de abandonar (Alsulami et al., 2023a).

Por ejemplo, los clasificadores usan varios modelos a la vez para que las predicciones sean más fiables, mientras que el aprendizaje profundo puede analizar enormes cantidades de datos con más detalle que los métodos tradicionales. Además, herramientas como los Sistemas de Gestión de Aprendizaje (LMS) (F. Chen & Cui, 2020; Jokhan et al., 2019) y los sistemas de alerta temprana (EWS) (Jokhan et al., 2019) han permitido personalizar más las estrategias de enseñanza en plataformas como Moodle y Blackboard. Sin embargo, todavía existen desafíos importantes: los datos que se usan pueden tener sesgos, los modelos no siempre funcionan igual de bien en diferentes universidades o contextos (problemas de escalabilidad) y a menudo faltan sistemas que den alertas automáticas y en tiempo real.

Paralelamente, ha surgido la analítica de aprendizaje (LA) como otra disciplina que ayuda a mejorar tanto la enseñanza como la experiencia de aprender (Sunday et al., 2020). La LA analiza datos de distintas fuentes (plataformas, notas, encuestas) para identificar a estudiantes en riesgo de abandono, dando a profesores y administradores información valiosa para adaptar sus estrategias pedagógicas (Kaunang & Rotikan, 2018). Esta técnica también sirve para evaluar y ajustar continuamente los métodos de enseñanza según lo que necesiten los estudiantes (Hidalgo Suarez et al., 2023; Mai et al., 2023). Todo esto demuestra lo

importante que es integrar la predicción y la analítica en la educación para enfrentar el abandono de forma más consciente y fundamentada.

Actuar rápido con intervenciones tempranas se considera una estrategia clave para reducir el abandono, sobre todo en cursos que exigen mucho esfuerzo mental, como los de programación (Prasanth & Alqahtani, 2023). Los sistemas de alerta temprana (EWS) son un buen ejemplo: están hechos para seguir el progreso de los estudiantes en tiempo real, permitiendo a los profesores intervenir justo a tiempo cuando detectan señales de riesgo. Hay estudios que demuestran que usar EWS puede reducir notablemente las tasas de abandono, porque permiten responder inmediatamente a lo que cada estudiante necesita.

De forma parecida, los sistemas de gestión de aprendizaje (LMS), como Moodle y Blackboard, han ido incorporando módulos predictivos. Estos permiten a los docentes adaptar sus métodos de enseñanza según cómo vean la participación y el rendimiento general de los estudiantes (F. Chen & Cui, 2020). Estas plataformas no solo ayudan a identificar a los estudiantes en riesgo, sino que también facilitan crear ayudas personalizadas, lo que mejora la experiencia de aprendizaje, especialmente en clases online o semipresenciales (Anh et al., 2023).

A pesar de los avances en la predicción del riesgo de abandono y ayuda temprana, aún existen hay desafíos importantes que impiden que estas soluciones se utilicen a gran escala. Uno de los principales problemas es la escalabilidad (Lazo Jaque, 2021). Los modelos predictivos pueden funcionar muy bien en pruebas pequeñas o controladas, pero llevarlos a instituciones grandes, donde cada estudiante tiene su forma de pensar y con recursos variados, sigue siendo difícil (Lazo Jaque, 2021). Este reto nos muestra que necesitamos diseñar modelos más flexibles y generales, que puedan funcionar bien en la gran variedad de contextos educativos que existen.

Otro gran obstáculo, según señalan (Jamjoom et al., 2021), es la falta de evaluación en tiempo real. Aunque algunos sistemas actuales pueden activar intervenciones basándose en datos históricos o patrones ya definidos, son muy pocos los que ofrecen alertas dinámicas y automáticas que respondan a lo que el estudiante está haciendo en ese preciso momento. Sin esa capacidad de reacción inmediata, los educadores pueden perder momentos clave para

intervenir, lo que les resta efectividad a las estrategias preventivas. Para superar estos vacíos, no bastará con innovaciones tecnológicas; se necesitará un enfoque interdisciplinario que combine los avances en analítica de datos con un entendimiento profundo de cómo funciona realmente el aprendizaje en el aula.

Centrándonos en los cursos de programación, (Llanos et al., 2023a) opinan que los Sistemas de Detección Temprana son fundamentales para identificar a estudiantes en riesgo. Estos sistemas suelen apoyarse en plataformas que siguen el aprendizaje, recogiendo y analizando datos sobre el progreso académico. Si estas plataformas se integran con herramientas que evalúan código automáticamente o que monitorean la actividad en los entornos de desarrollo, se puede dar retroalimentación inmediata y personalizada, facilitando intervenciones justo a tiempo. Integrar todo esto con un LMS como Moodle o Blackboard es crucial para tener la información centralizada y coordinar las estrategias de apoyo. Sin embargo, hay desafíos propios de la enseñanza de programación: evaluar bien las habilidades prácticas y la capacidad de resolver problemas requiere métodos más específicos que los habituales (Guzmán-Castillo et al., 2022).

Además, sigue habiendo limitaciones para predecir tempranamente quién tendrá dificultades, debido a la gran variedad en los perfiles de los estudiantes y en los datos que se tienen, lo que complica diseñar modelos predictivos que sean realmente robustos y precisos (Jamjoom et al., 2021). Por otro lado, la escalabilidad de estas soluciones continúa siendo un reto, sobre todo en clases muy numerosas o en instituciones con recursos tecnológicos limitados (Guzmán-Castillo et al., 2022). Finalmente, existen barreras para implementar las intervenciones, como la resistencia al cambio por parte de algunos o la falta de capacitación del personal docente. Esto subraya que necesitamos abordar estos problemas de forma integral, combinando la tecnología con estrategias pedagógicas efectivas y adaptadas a cada entorno educativo (Alboaneen et al., 2022).

Antecedentes

Se ha investigado mucho sobre por qué los estudiantes abandonan las carreras de Ciencias de la Computación, usando a menudo enfoques basados en análisis de datos y

predicción. Por ejemplo, trabajos como los de Llanos et al. (2023) y Martins et al. (2023) han mostrado que algoritmos como Random Forest son bastante efectivos para predecir el riesgo de abandono, analizando factores académicos, sociodemográficos e incluso macroeconómicos. Otros estudios, como el de Guzmán-Castillo et al. (2022), han ido más allá desarrollando sistemas de información que predicen posibles abandonos y ayudan a coordinar las intervenciones de apoyo, lo que subraya lo útiles que son estas tecnologías para la gestión educativa. También hay investigaciones como la de González Rojas (2021) que usan técnicas más avanzadas como redes neuronales y minería de datos para identificar patrones ocultos que influyen en el abandono.

Pero los números no lo cuentan todo. Hay estudios cualitativos, como el de Schefer-Wenzl et al. (2024), que han profundizado en las razones de fondo por las que los estudiantes se van, encontrando problemas como expectativas que no se cumplen o falta de tiempo, y dando recomendaciones prácticas para reducir el problema. Además, otros trabajos se han centrado en adaptar los modelos a realidades locales, usando análisis de datos para crear modelos predictivos ajustados a las características del país. Todos estos antecedentes nos muestran que para entender y combatir el abandono estudiantil se usan enfoques variados, combinando análisis cuantitativos con estudios cualitativos, y que estas herramientas se pueden aplicar con éxito en distintas regiones y contextos educativos.

Antecedentes Internacionales

En el estudio "An Investigation into Student Performance Prediction using Regularized Logistic Regression" desarrollado por (Kurniadi et al., 2023), los autores se enfocaron en el problema del abandono universitario y la importancia de la identificación temprana de estudiantes en riesgo. Para ello, utilizaron modelos de regresión logística con técnicas de regularización, específicamente Lasso (L1) y Ridge (L2), con el objetivo de mejorar la precisión en la predicción del rendimiento estudiantil. Los datos para la investigación se obtuvieron de la plataforma Binus Online Learning de la Bina Nusantara University, centrándose en estudiantes del programa de Sistemas de Información entre 2020 y 2021. El conjunto de datos incluía diversas variables de rendimiento como asistencia

(ATT), participación en foros (FOD), tareas personales (PAS1, PAS2), quizzes (QIZ1, QIZ2) y trabajos en equipo (TAS1-TAS4). La variable a predecir, denominada "Ket", clasificaba a los estudiantes en categorías de "Riesgo" o "Seguro". El hallazgo más revelador fue que los modelos de regresión logística con regularización superaron al modelo de regresión logística tradicional ("Vanilla"). Específicamente, el modelo de Regresión Logística con Lasso obtuvo los mejores resultados, alcanzando una precisión (accuracy) de 0.9947, una puntuación de precisión (precision) de 1, y un F1-score de 0.9697. El modelo de Regresión Logística con Ridge también mostró un buen rendimiento, aunque ligeramente inferior al de Lasso.

Por otro lado, en otro artículo, “A Study on Reasons for Student Dropouts in a Computer Science Bachelor’s Degree Program” desarrollado por (Schefer-Wenzl et al., 2024) se exploran las altas tasas de abandono en los programas de grado en Ciencias de la Computación, destacando el desafío que esto representa para producir suficientes graduados en este campo de alta demanda. El estudio reconoce la complejidad de identificar las razones detrás de estas deserciones, dado el carácter a menudo inaccesible o poco cooperativo de los estudiantes que abandonan. Se empleó un enfoque de investigación que incluyó una revisión de la literatura y entrevistas cualitativas para explorar estas razones, encontrando que las limitaciones de tiempo y las expectativas desalineadas con el programa de grado son los factores principales que conducen al abandono estudiantil. A partir de estos hallazgos, el artículo ofrece recomendaciones para reducir las tasas de abandono en los programas de ciencias de la computación.

Asimismo, "Prediction of Students' Academic Performance in the Programming Fundamentals Course Using Long Short-Term Memory Neural Networks" hecho por (Vives et al., 2024) presenta una investigación sobre la predicción del rendimiento académico para reducir las altas tasas de deserción y desaprobación en el curso "Fundamentos de Programación" en universidades peruanas. El estudio se centró en explorar la eficacia de las Redes de Memoria a Corto y Largo Plazo (LSTM) para predecir si los estudiantes aprobarían o desaprobarían el curso en las semanas 7, 8, 12 y 16 del semestre académico. Para ello, se analizaron 661 registros de estudiantes de las carreras de Ingeniería de Software, Ciencias de la Computación e Ingeniería de Sistemas de Información de dos universidades peruanas,

utilizando 13 atributos académicos.La metodología comparó el modelo LSTM con otros seis algoritmos de aprendizaje automático: Deep Neural Network (DNN), Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Support Vector Classifier (SVM) y K-Nearest Neighbor (KNN). Un desafío clave fue el desequilibrio en los datos (68% aprueban, 32% desaprueban), que se abordó utilizando dos técnicas de sobremuestreo: Synthetic Minority Over-sampling Technique (SMOTE) y Generative Adversarial Networks (GAN). Los resultados demostraron que el modelo propuesto LSTM-GAN fue superior, alcanzando una precisión, recall y F1-Score del 98.3% en la semana 8, la cual se considera estratégica para la toma de decisiones institucionales. Este fue seguido de cerca por el modelo DNN-GAN con un 98.1% de precisión. Se concluyó que la técnica GAN generó datos sintéticos más realistas que se adaptaron mejor a los modelos de aprendizaje profundo como LSTM y DNN, a diferencia de los datos originales o los balanceados con SMOTE, que favorecían a los modelos tradicionales, pero con sesgos.

Otro trabajo interesante es el de (Guzmán-Castillo et al., 2022) titulado "Implementation of a Predictive Information System for University Dropout Prevention". En este caso, los autores no solo estudiaron el problema, sino que implementaron un sistema de información predictivo real para prevenir el abandono universitario. Conscientes de la gravedad del abandono y de los cambios recientes en la educación y la situación socioeconómica, diseñaron este sistema para evaluar diversos factores que influyen en que un estudiante deje la universidad. Este calcula el riesgo de abandono para cada estudiante y genera alertas que ayudan a coordinar la ayuda adecuada. Una de las ventajas que observaron fue que la plataforma permitió reorganizar y priorizar las intervenciones según el nivel de riesgo de cada alumno, lo que ayudó a mejorar la permanencia estudiantil. Este estudio es un buen ejemplo de lo útiles que pueden ser las tecnologías predictivas en la gestión diaria de la educación.

Finalmente, el estudio "Multi-Class Phased Prediction of Academic Performance and Dropout in Higher Education" de (Martins et al., 2023) que explora cómo usar el aprendizaje automático para predecir el rendimiento y el riesgo de abandono en la universidad. Lo interesante de su enfoque es que hicieron estas predicciones en tres momentos distintos

durante el primer año académico. Utilizaron datos de estudiantes de pregrado de una universidad politécnica en Portugal, matriculados entre 2009 y 2017, e incluyeron información académica, sociodemográfica y hasta datos macroeconómicos. Probaron cinco algoritmos diferentes y destacaron que Random Forest funcionó especialmente bien, sobre todo porque sus datos no estaban perfectamente balanceados. Encontraron que el mejor momento para predecir era al final del primer semestre. Concluyeron que sus resultados encajaban bastante bien con lo que ya se sabía por otras investigaciones sobre predicción del rendimiento y abandono estudiantil.

Antecedentes Nacionales

Aquí en Colombia, el abandono estudiantil en la educación superior es un problema serio, y esto afecta también a los cursos de programación. Según datos del Ministerio de Educación Nacional (SPADIES, 2021), la cifra es preocupante: en promedio, 25.47% de estudiantes universitarios deja sus estudios antes de terminarlos. Hay varios factores que contribuyen a esto, como las dificultades económicas, el bajo rendimiento académico o la falta de motivación. Para intentar reducir estas cifras, las universidades están poniendo en marcha estrategias como actualizar sus planes de estudio y ofrecer cursos más flexibles, buscando adaptarse mejor a las necesidades que tienen los estudiantes hoy en día.

Dentro del contexto colombiano, un ejemplo relevante es el trabajo de (González Rojas, 2021) en la Universidad Distrital Francisco José de Caldas. En su estudio, titulado "Construcción de un modelo para predecir el rendimiento académico de los estudiantes de ingeniería electrónica de la universidad distrital Francisco José de Caldas mediante algoritmos de redes neuronales con aprendizaje automático de 2021", el autor desarrolló un modelo para predecir cómo les iría académicamente a los estudiantes de ingeniería electrónica, usando redes neuronales y aprendizaje automático. Para construir y validar el modelo, utilizó datos anónimos de estudiantes que ingresaron entre 2008 y 2020, seleccionando las variables más importantes y evaluando qué tan bien funcionaba el modelo. La idea detrás de este esfuerzo era sentar bases para futuras investigaciones que busquen mejorar el rendimiento de los estudiantes y disminuir el abandono en la carrera.

Por otro lado, el estudio "Early prediction of student performance in CS1 programming courses" desarrollado por (Llanos et al., 2023) se centra en la importancia de poder predecir pronto cómo les irá a los estudiantes en los cursos iniciales de programación, precisamente porque en estas materias suele haber altas tasas de fracaso y bajo rendimiento. Para lograr mejores predicciones, emplearon técnicas avanzadas de clasificación y algoritmos de aprendizaje automático, analizando factores como las calificaciones, los tiempos de entrega y el número de intentos en trabajos prácticos y exámenes. Una de sus conclusiones principales fue que los clasificadores basados en aumento de gradiente (Gradient Boosting) y bosques aleatorios (Random Forest) resultaron especialmente efectivos para predecir con precisión el desempeño de los estudiantes a lo largo de las 16 semanas que duraba el curso.

Finalmente, según el estudio "Predicting the final grade using a machine learning regression model: insights from fifty percent of total course grades in CS1 courses" desarrollado por (Hidalgo Suarez et al., 2023) propone un modelo de aprendizaje automático con un objetivo muy práctico: predecir con buena precisión la nota final que obtendrá un estudiante en un curso inicial de programación (CS1), usando como información solo las calificaciones de la primera mitad del curso. Su método siguió tres fases (entrenamiento, prueba y validación) y probaron varios algoritmos de regresión, incluyendo AdaBoost, Bosque Aleatorio (Random Forest), Regresión de Soporte Vectorial (SVR) y XGBoost. De todos ellos, el SVR fue el que dio mejores resultados, alcanzando valores de R-cuadrado entre el 72% y el 91%. Estos números indican que el modelo desarrollado tiene una alta precisión y es bastante confiable para hacer este tipo de predicción.

Antecedentes Locales

Cuando se habla de antecedentes locales, se identificó una escasez de estudios específicos sobre abandono en cursos de programación en el contexto local. Faltan investigaciones que analicen las particularidades de la enseñanza de la programación y su relación con el abandono. Además, se encontró que se necesitan más estudios sobre intervenciones efectivas para prevenir el abandono en estos cursos, considerando factores contextuales y tecnológicos que influyen en la retención estudiantil.

En este sentido, el artículo titulado "Desarrollo de una aplicación web de seguimiento y deserción para los estudiantes de la Corporación Universitaria del Huila (CORHUILA)", presentado en el III Congreso Internacional en Inteligencia Ambiental, Ingeniería de Software y Salud Electrónica y Móvil desarrollado por (Guzmán-Castillo et al., 2022), Aborda el preocupante problema de la deserción estudiantil en la educación superior en Colombia, donde las tasas alcanzan un alarmante 48. 8%. Este estudio tiene como objetivo mitigar esta problemática a través del desarrollo de una aplicación web basada en el estándar SPADIES del Ministerio de Educación Nacional. La investigación, realizada con 278 estudiantes bajo un diseño no experimental, transversal y descriptivo, indagó en las causas de la deserción en nuestra región. El estudio identificó varios factores clave que contribuyen de forma importante a que los estudiantes abandonen: dificultades económicas, falta de motivación, problemas familiares, cambios de ciudad y una alta exigencia académica. Se observó que este problema afecta predominantemente a hombres entre 19 y 25 años. Como parte de la solución, se desarrolló un prototipo inicial llamado SPACOR. Este sistema, construido con tecnologías como NetBeans y PostgreSQL bajo el patrón MVC, permite guardar y analizar datos para generar informes que ayuden en la toma de decisiones académicas en la institución. El objetivo de SPACOR, en esta primera fase, es ayudar a diseñar estrategias específicas para mejorar la retención, con la mira puesta en integrar inteligencia artificial en el futuro para realizar análisis predictivos y ofrecer apoyo más personalizado. Con iniciativas como esta, CORHUILA busca no solo reducir la deserción, sino también mejorar la calidad de la educación y cumplir con su compromiso social aquí en la región.

Es importante mencionar que, aunque el artículo de esta investigación no use textualmente el término "abandono estudiantil", sí aborda claramente esta problemática desde una perspectiva local. Representa un paso significativo en la creación de herramientas tecnológicas y métodos para entender mejor y poder mitigar las razones por las cuales los estudiantes dejan sus estudios.

Marco teórico

Cuando un estudiante deja la educación superior, es un problema complejo con muchas caras que afecta tanto al estudiante como a la universidad (Llanos et al., 2023). Para la persona que abandona, esto puede traer consecuencias importantes en su vida personal y profesional, como tener menos oportunidades de trabajo o que su autoestima se vea afectada (González Rojas, 2021). Para las universidades, el abandono genera problemas serios relacionados con su eficiencia, cómo usan sus recursos y su capacidad para mantener los programas académicos funcionando (Kocsis & Molnár, 2024).

En los últimos años, se ha visto un problema clave en la enseñanza de la programación: las tasas de abandono y fracaso son muy altas, moviéndose entre el 30% y el 50% (SPADIES, 2021). Aunque en todo el mundo se necesitan más programadores y más gente se inscribe en carreras tecnológicas, muchos estudiantes sienten que es demasiado difícil o pierden la motivación y terminan dejando los estudios, lo cual perjudica a todos (Margulieux et al., 2020). En Iberoamérica y Colombia, la tasa de deserción sigue siendo un desafío, con un 10.08% en 2021 (SPADIES, 2021), mientras que, en el departamento del Huila, se reportó un 8.63% en 2017 (Gobernación del Huila, 2020). Esta situación limita que tengamos suficientes profesionales bien preparados, lo que impacta nuestra competitividad en el mercado laboral y hace que las inversiones del gobierno en educación no rindan todo lo que deberían (Schefer-Wenzl et al., 2024).

Se entiende que el abandono estudiantil es el resultado de una mezcla de factores académicos, personales y del contexto que rodea al estudiante (Mai et al., 2023). En los cursos de programación, esto se complica aún más por lo técnico de los temas. Los estudiantes que llegan sin una buena base en lógica, resolución de problemas o algoritmos, a menudo se sienten frustrados y desmotivados, sobre todo al principio (Verma et al., 2022). Además, factores externos como la falta de apoyo por parte de la institución o que no haya suficientes recursos para tutorías o repasos, pueden hacer que los estudiantes se sientan desconectados del proceso de aprendizaje (Uhanova et al., 2023). Por eso, para enfrentar de

verdad el abandono en estos cursos, necesitamos entender bien cómo interactúan todos estos diferentes factores que influyen en los estudiantes.

El rendimiento académico de un estudiante suele ser una señal temprana que alerta sobre un posible abandono. Cuando a los estudiantes no les va bien en las notas, pueden sentir que no son capaces de seguir el ritmo del curso, y eso aumenta las ganas de dejarlo (Anh et al., 2023). Pero las calificaciones no son lo único que importa. Factores como la situación socioeconómica, las estrategias que usa el estudiante para aprender y su motivación personal también juegan un papel crucial (Lakanen & Isomöttönen, 2023). Por ejemplo, los estudiantes que tienen que trabajar mientras estudian disponen de menos tiempo para dedicarle a la universidad, lo que puede afectar sus notas y, por lo tanto, hacer más difícil que continúen (Kocsis & Molnár, 2024). Al mismo tiempo, la autosuficiencia es fundamental: aquellos estudiantes que creen que pueden superar los obstáculos tienen más probabilidades de seguir adelante, incluso si las cosas se ponen difíciles (Burgos et al., 2018). Estas relaciones complejas nos muestran que para entender el abandono hay que mirar el panorama completo, considerando tanto al estudiante como a su entorno.

Para enfrentar esta situación, sobre todo en los cursos de programación, es clave analizar por qué los estudiantes los abandonan, por ejemplo, porque los perciben como muy complejos o porque faltan estrategias de enseñanza efectivas (Schefer-Wenzl et al., 2024). Si un estudiante se siente emocionalmente desconectado o mentalmente sobrecargado, le costará más aprender. Esto sugiere que necesitamos implementar modelos educativos innovadores que motiven más a los alumnos y hagan que la programación parezca menos intimidante. Además, si las instituciones entienden mejor estos patrones, podrán diseñar estrategias de retención más inteligentes, optimizar cómo se enseña programación, reducir la pérdida de estudiantes y, así, aprovechar mejor los recursos que se invierten en la educación superior (Guzmán-Castillo et al., 2022).

Modelos de predicción

Dentro del campo educativo, los modelos predictivos se han consolidado como herramientas muy valiosas para adelantarse al abandono estudiantil y poder reducirlo

(Balachandar & Venkatesh, 2023). Estos modelos se crean usando técnicas de minería de datos y aprendizaje automático, las cuales permiten encontrar patrones en el comportamiento de los estudiantes que se asocian con un mayor riesgo de deserción. Además, si se combinan datos de distintas fuentes, como las plataformas de aprendizaje en línea (LMS), los registros de asistencia o las calificaciones, estos modelos se vuelven más precisos. Un buen ejemplo es que varios estudios han mostrado que la frecuencia con la que un estudiante accede a los recursos en un LMS es un indicador fiable de qué tan comprometido está con sus estudios y, por lo tanto, de su riesgo de abandonar SLUS (Alonso et al., 2024).

Naive Bayes (NB)

El Naive Bayes es un algoritmo bastante conocido que se usa para clasificar cosas. Se basa en el teorema de Bayes, una regla matemática que relaciona diferentes tipos de probabilidades (condicionales y marginales) entre dos eventos. Lo que hace "ingenuo" (naive) a este modelo es una suposición clave que hace para simplificar las cosas: asume que todas las características que describen un caso son independientes entre sí, aunque en la realidad no siempre sea así. Esta simplificación es justamente lo que permite que los cálculos de probabilidad sean más sencillos y que el algoritmo pueda clasificar de manera eficiente (S. Chen et al., 2020).

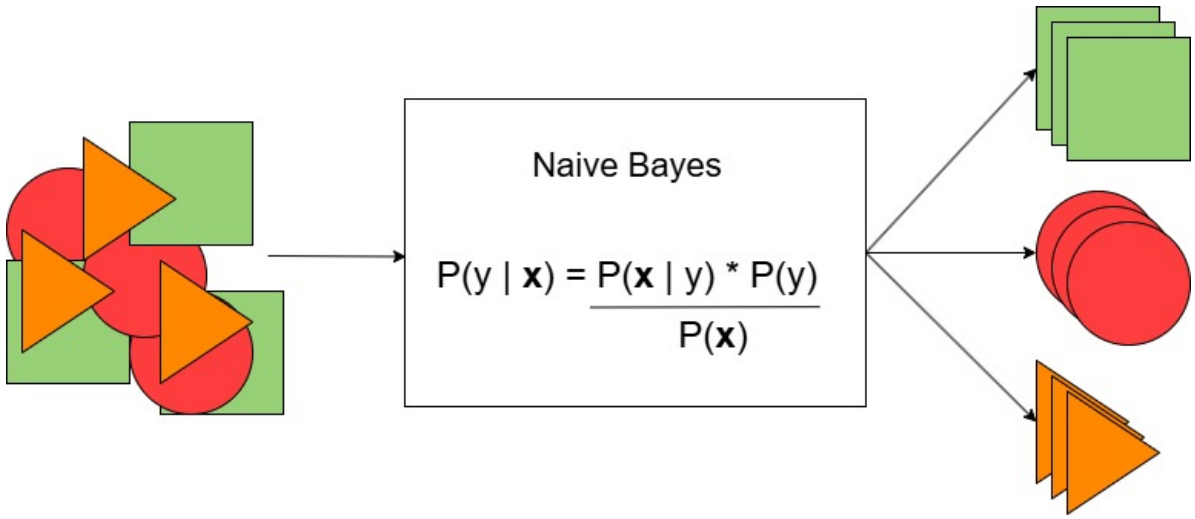
Como se puede ver en la Figura 1, Naive Bayes calcula la probabilidad de que un caso pertenezca a cada clase posible. Para ello, combina la probabilidad "previa" de esa clase (qué tan común es en general) con la probabilidad de observar las características específicas de ese caso dado que pertenece a esa clase (Ismail et al., 2020). El modelo se apoya en los siguientes elementos clave:

- **P(y|x):** Probabilidad posterior de que la instancia pertenezca a la clase y, dado el conjunto de características x.
- **P(x|y):** verosimilitud, es decir, la probabilidad de observar las características x si la instancia pertenece a la clase y.

- **P(y):** probabilidad a priori de la clase y, que representa la frecuencia con la que ocurre dicha clase en el conjunto de datos.
- **P(x):** probabilidad total de las características x, que actúa como un factor de normalización.

Figura 1

Gráfica Naive Bayes (NB)



Nota. Adaptado de (Ismail et al., 2020).

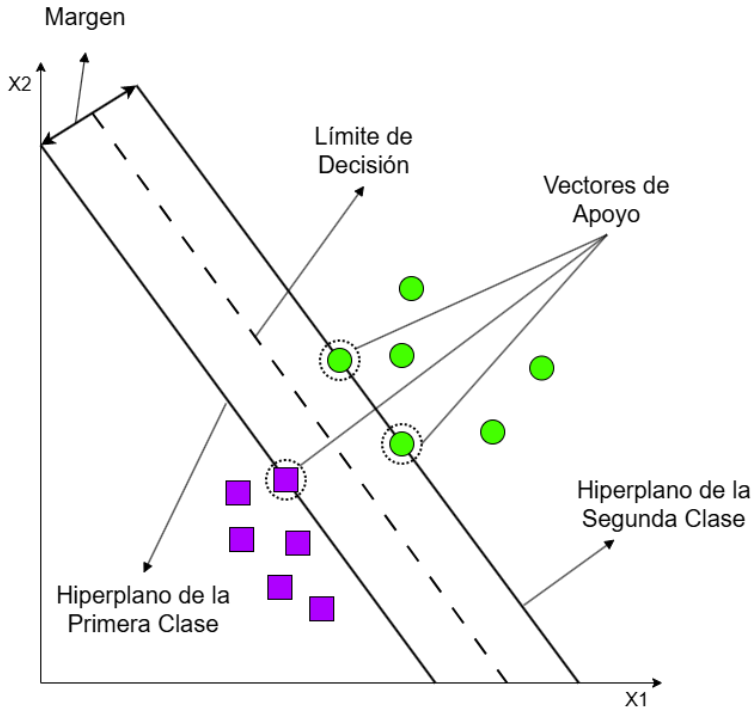
Support Vector Machine (SVM)

Support Vector Machine busca es encontrar un hiperplano óptimo que separe los datos en distintas clases. Este hiperplano se establece de manera que maximiza la distancia mínima entre él y los puntos más cercanos de cada clase, los cuales se denominan vectores de soporte. En situaciones donde los datos no son linealmente separables, las máquinas de soporte vectorial (SVM) recurren a funciones kernel

para transformar los datos en un espacio de mayor dimensión, donde sí es posible realizar la separación, tal como se ilustra en la Figura 2 (Pisner & Schnyer, 2020).

Figura 2

Gráfica de Support Vector Machine (SVM)



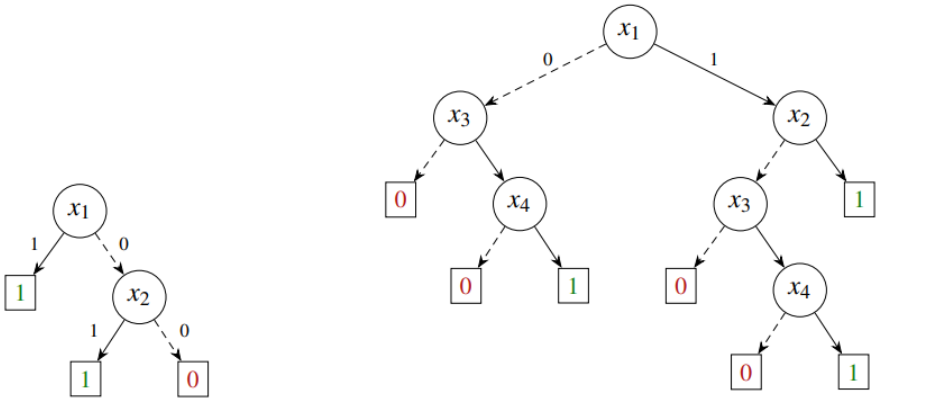
Nota. Adaptado de (Ioannou & Vassiliou, 2021).

Decision Tree (DT)

Un árbol de decisión organiza los datos en una estructura jerárquica que se asemeja a un árbol, donde cada nodo simboliza una decisión o pregunta fundamentada en las características de los datos. Tal como se ilustra en la Figura 3, el algoritmo segmenta los datos en ramas mediante reglas que buscan maximizar métricas como la ganancia de información o la reducción de la entropía. Este proceso se repite hasta obtener una clasificación definitiva (Izza et al., 2020).

Figura 3

Gráfica de Decision Tree (DT)



(a) DT for $f(x_1, x_2) = x_1 \vee x_2$ (b) DT for $f(x_1, \ldots, x_n) = \bigvee_{i=1}^{n/2} x_{2i-1} \wedge x_{2i}$, with $n = 4$

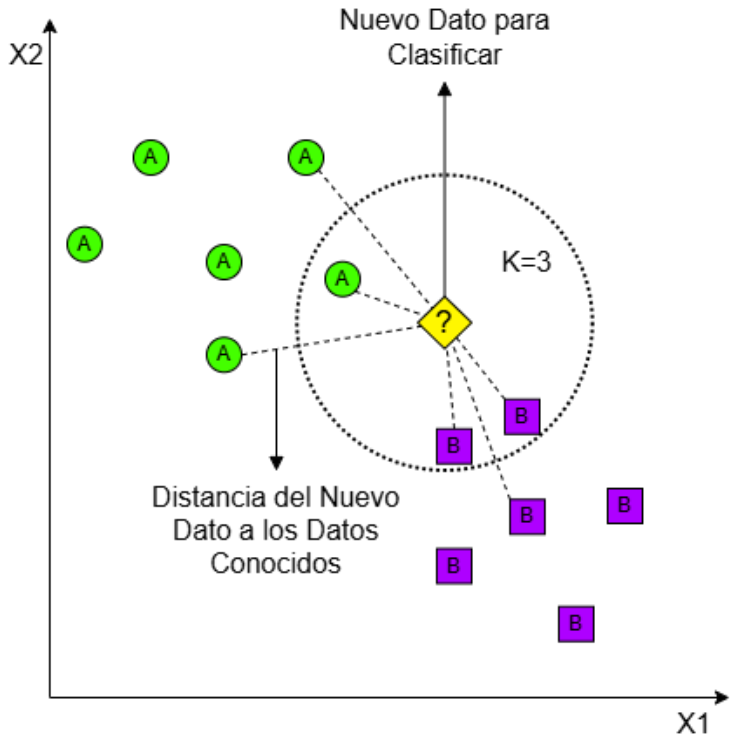
Nota. Tomado de (Izza et al., 2020).

K-Nearest Neighbors (KNN)

K-Nearest Neighbors clasifica una instancia teniendo en cuenta las clases de sus vecinos más cercanos en el espacio de características. Para predecir la clase de una nueva instancia, el algoritmo calcula la distancia entre esta y los puntos del conjunto de datos, selecciona a los "k" vecinos más cercanos y asigna la clase que aparece con mayor frecuencia entre ellos, tal como se ilustra en la Figura 4 (Cunningham & Delany, 2022).

Figura 4

Gráfica de K-Nearest Neighbors (KNN)



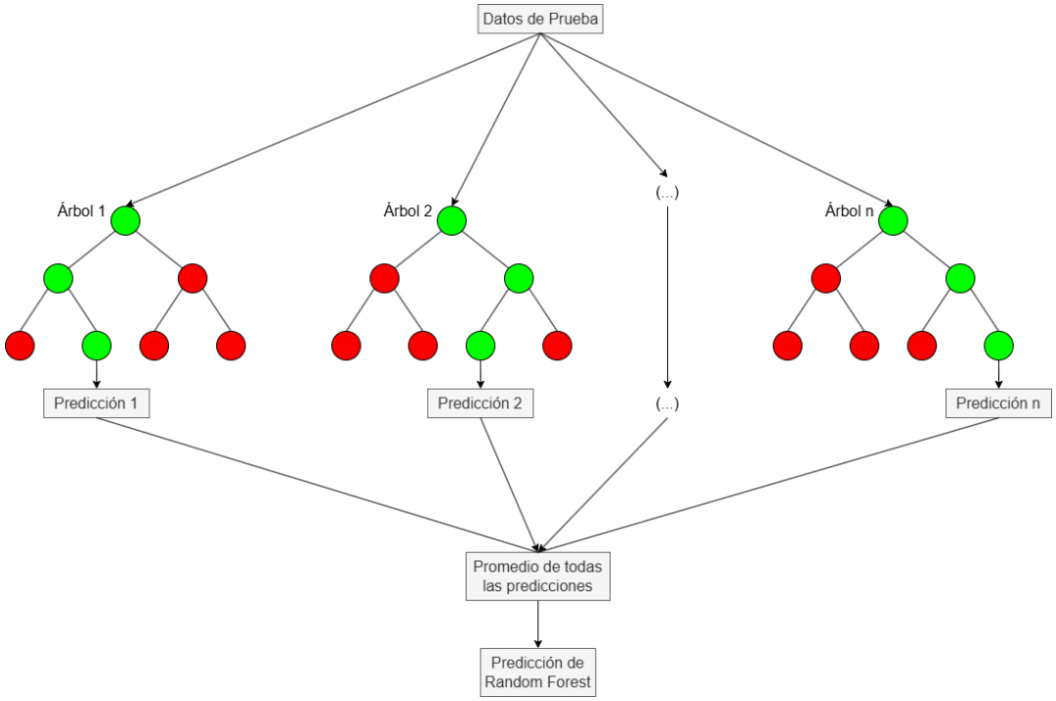
Nota. Adaptado de (Cunningham & Delany, 2022; Uddin et al., 2022)

Random Forest (RF)

Random Forest es un algoritmo de conjunto que utiliza múltiples árboles de decisión para realizar predicciones. Como se observa en la Figura 5, los árboles se construyen de manera aleatoria seleccionando subconjuntos de datos y características. La predicción final se obtiene combinando los resultados de todos los árboles, ya sea mediante un promedio (regresión) o una votación mayoritaria (clasificación) (Parmar et al., 2019).

Figura 5

Gráfica de Random Forest (RF)



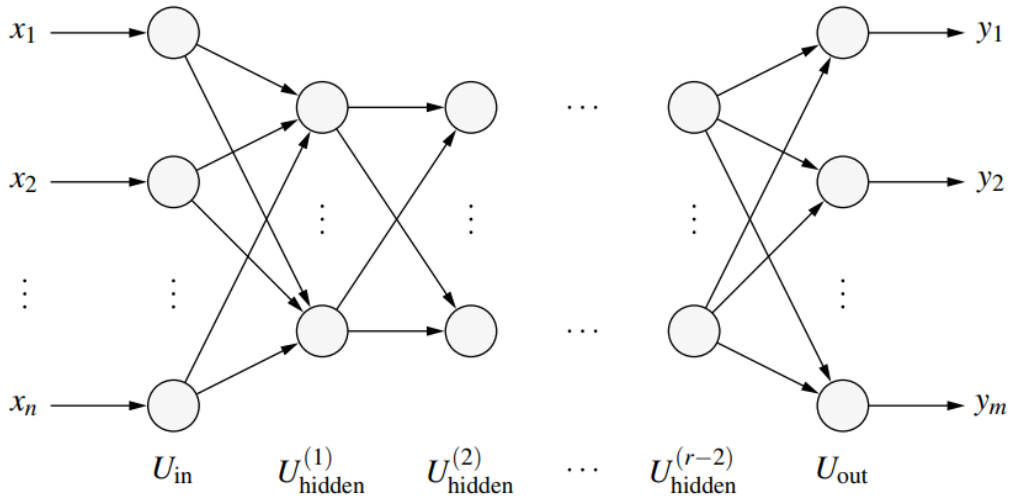
Nota. Adaptado de (Qadir & Abd, 2023).

Multilayer Perceptron (MLP)

Multilayer Perceptron es una red neuronal artificial compuesta por capas de neuronas conectadas entre sí. Como se observa en la Figura 6, cada neurona aplica una función de activación a la entrada recibida y transmite el resultado a la siguiente capa. Durante el entrenamiento, el error en las predicciones se retropropaga por las capas, ajustando los pesos de las conexiones mediante algoritmos de optimización como el gradiente descendente (Kruse et al., 2022).

Figura 6

Gráfica de Multilayer Perceptron (MLP)



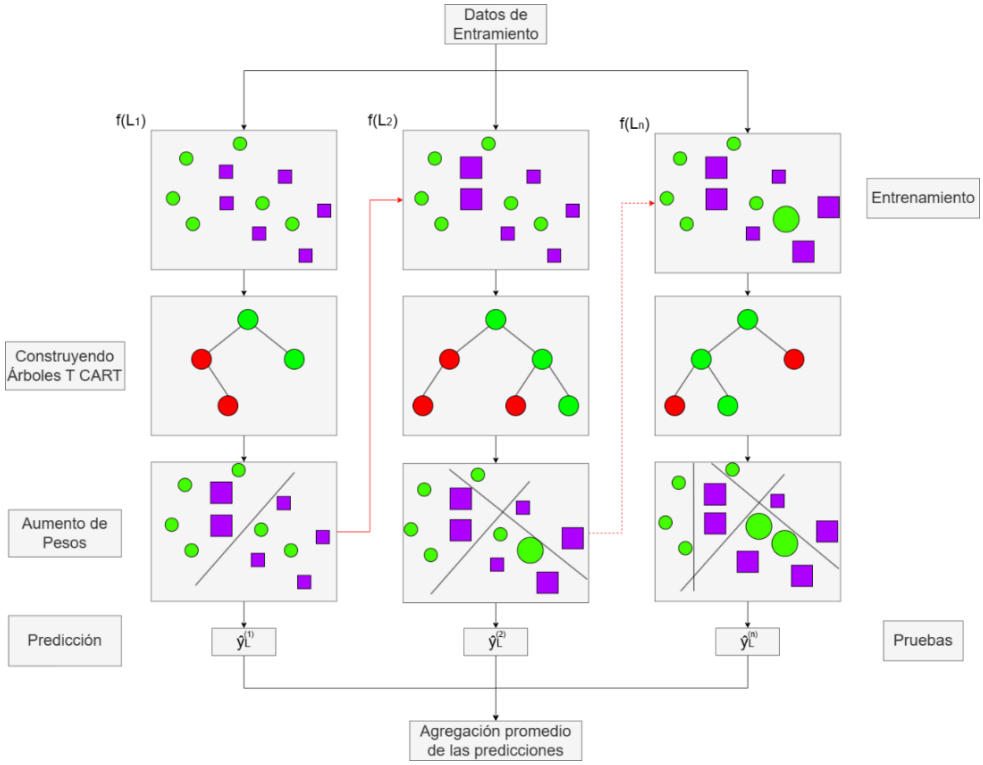
Nota. Tomado de (Kruse et al., 2022).

XGBoost (Extreme Gradient Boosting)

XGBoost es una técnica de boosting que entrena secuencialmente modelos, generalmente árboles de decisión, para corregir los errores de los modelos anteriores. Como se observa en la Figura 7, cada nuevo modelo minimiza los errores residuales utilizando gradiente descendente, y las predicciones finales son una combinación ponderada de todos los modelos entrenados, logrando una alta precisión en el resultado final (Aydin & Ozturk, 2021).

Figura 7

Gráfica de XGBoost (Extreme Gradient Boosting)



Nota. Adaptado de (Yao et al., 2022).

Métricas

Se evalúa la eficiencia de los modelos por medio de tres métricas: *precisión, recall* y *F1 score* y el área bajo la curva:

- **Precisión**

La precisión evalúa la proporción de predicciones positivas correctas realizadas por el modelo en relación con el total de predicciones positivas (Llanos et al., 2023).

Figura 8

Fórmula Precisión

$$Precision = \frac{TP}{TP + FP}$$

Nota. Tomado de (Llanos et al., 2023)

Donde:

- TP (True Positives) son los verdaderos positivos.
- FP (False Positives) son los falsos positivos.

- **Recall**

El recall, también conocido como sensibilidad o tasa de verdaderos positivos, mide la capacidad del modelo para identificar correctamente todos los casos positivos en el conjunto de datos (Llanos et al., 2023).

Figura 9

Fórmula Recall

$$Recall = \frac{TP}{TP + FN}$$

Nota. Tomado de (Llanos et al., 2023)

Donde:

- TP (True Positives) son los verdaderos positivos.
- FN (False Negatives) son los falsos negativos.

- F1 Score

El F1 Score representa la media armónica de la precisión y el recall, fusionando ambas métricas en un único valor. Esta combinación resulta especialmente valiosa en contextos donde es crucial lograr un equilibrio entre los falsos positivos y los falsos negativos (Llanos et al., 2023).

Figura 10

Fórmula F1 Score

$$F1\ Score = 2 * \frac{Precision\ x\ Recall}{Precision + Recall}$$

Nota. Tomado de (Llanos et al., 2023).

- **Área Bajo la Curva (AUC)**

El Área Bajo la Curva (conocida como AUC por sus siglas en inglés) es una medida muy usada para saber qué tan bueno es un modelo de clasificación en general. En esencia, mide la capacidad que tiene el modelo para distinguir correctamente entre los casos positivos y los negativos, sin importar qué umbral o punto de corte de probabilidad estemos usando (F. Chen & Cui, 2020). Este valor se saca de la curva ROC, que es un gráfico que relaciona la proporción de positivos que el modelo acierta (Tasa de Verdaderos Positivos o TPR) con la proporción de negativos que clasifica incorrectamente como positivos (Tasa de Falsos Positivos o FPR). Es fácil de interpretar: un AUC cercano a 1 significa que el modelo es excelente diferenciando las clases, mientras que un valor cercano a 0.5 indica que el modelo no es mejor que adivinar al azar.

Para que los modelos predictivos funcionen bien y sean realmente útiles, es fundamental partir de datos que sean sólidos y que de verdad representen a los estudiantes (Lacave et al., 2018). En este sentido, contamos con herramientas como el cuestionario MSLQ-Colombia, el cual ya ha sido validado para usarse aquí en Colombia. Este cuestionario es valioso porque nos permite medir aspectos clave de los estudiantes, como su motivación, las estrategias que utilizan para aprender y sus emociones (Ramírez Echeverry et al., 2016).

Usar modelos predictivos en la educación no es solo una cuestión técnica, también plantea preguntas éticas importantes sobre la privacidad y la equidad (Alfere & Maghari, 2018). Cuando se recogen datos personales, es absolutamente necesario tener protocolos estrictos de consentimiento informado, asegurándonos de que los estudiantes entiendan bien cómo se usará su información. Además, los algoritmos deben diseñarse con cuidado para minimizar sesgos que puedan empeorar las desigualdades que ya existen. Por ejemplo, si un modelo le da demasiada importancia al historial académico previo, podría ignorar factores del contexto del estudiante, como su situación socioeconómica, y perjudicar injustamente las predicciones (Nabil et al., 2021).

A nivel de la universidad, implementar la analítica del aprendizaje no se trata solo de comprar software avanzado. Requiere un enfoque integral: buena infraestructura tecnológica y también capacitación para los docentes y el personal, para que sepan interpretar y usar los resultados de forma efectiva (Balachandar & Venkatesh, 2023). Además, las políticas de la institución deben promover una cultura donde se innove de manera responsable, asegurando que las decisiones basadas en datos siempre respeten los principios de inclusión y equidad (Martins et al., 2023).

Si miramos estudios internacionales sobre el abandono en cursos de programación, a menudo se destaca que es un fenómeno complejo, señalando factores como la falta de motivación, la poca preparación previa o que los contenidos se perciben como muy difíciles (Mai et al., 2023). Sin embargo, investigaciones hechas aquí en Colombia y en contextos locales similares, frecuentemente resaltan la influencia de otros aspectos cruciales: las condiciones socioeconómicas, barreras culturales o las limitaciones para acceder a recursos tecnológicos impactan mucho en si los estudiantes logran continuar. Estos problemas muchas veces reflejan desigualdades estructurales de nuestra sociedad que afectan el desempeño y la permanencia de los estudiantes en sus cursos (Jamjoom et al., 2021).

Frente a los desafíos que hemos visto, especialmente en cursos como los de programación aquí, la implementación de modelos predictivos se está volviendo una herramienta fundamental. Estos modelos modernos intentan ir más allá de las simples notas, buscando integrar aspectos como la motivación, el compromiso y las emociones de los estudiantes (Kocsis & Molnár, 2024). Hemos avanzado desde enfoques que solo miraban datos históricos hacia el uso de algoritmos más sofisticados, como Random Forest y XGBoost, que pueden detectar patrones complejos, a veces casi en tiempo real (Lázaro Alvarez et al., 2020). Sin embargo, es importantísimo que al adoptar estas herramientas lo hagamos con un fuerte compromiso ético y una visión inclusiva. Debemos tener en cuenta la diversidad de nuestros estudiantes y asegurarnos de que cualquier intervención que se proponga sea justa y realmente adecuada a su contexto y necesidades.

Metodología

Los participantes en este estudio fueron estudiantes de primer semestre que estaban cursando materias introductorias de programación aquí en la Corporación Universitaria del Huila. No los seleccionamos al azar, ya que los grupos de clase estaban definidos desde el inicio del semestre, lo cual encaja con el enfoque cuasi-experimental que seguimos (Fernández et al., 2014). Nos aseguramos de incluir intencionalmente a estudiantes con diferentes niveles académicos y con o sin experiencia previa en programación, para que la muestra representara bien a la población que nos interesaba estudiar.

La forma en que abordamos la investigación fue principalmente descriptiva: observamos el comportamiento de los estudiantes durante todo el curso, registrando datos tanto cualitativos como cuantitativos. Para poner a prueba nuestra hipótesis, comparamos los resultados de un grupo experimental (al que se le aplicó una intervención específica en su aprendizaje) con los de un grupo de control (que siguió con las clases tradicionales impartidas por el profesor).

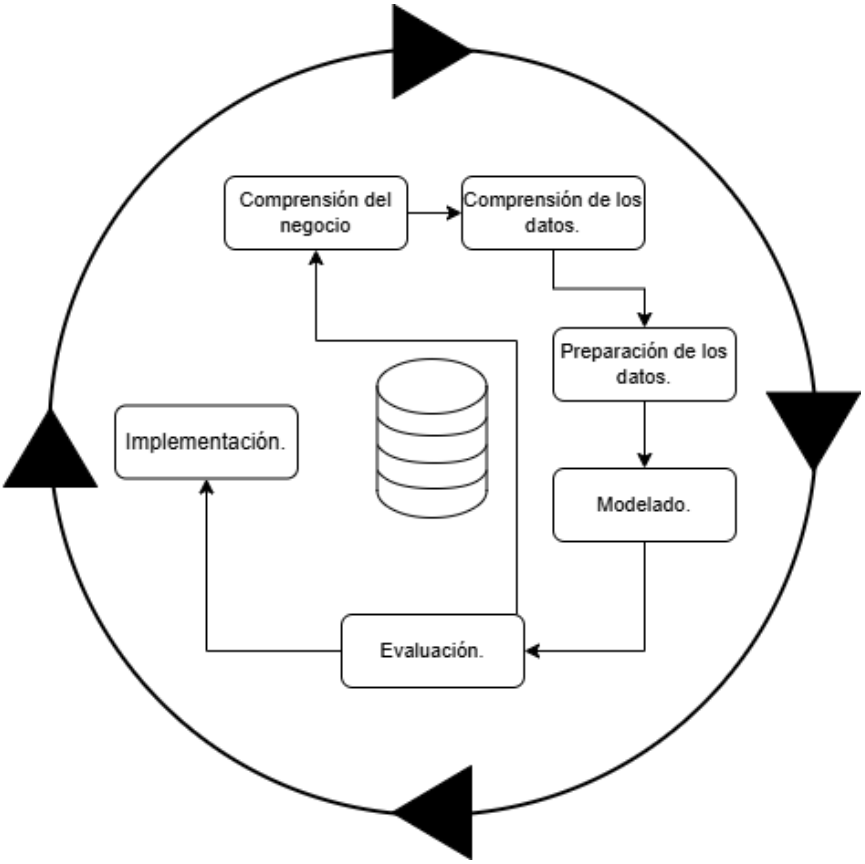
En esta investigación combinamos dos metodologías principales: el proceso estándar para minería de datos conocido como CRISP-DM y un diseño cuasi-experimental. A continuación, describimos cada una.

CRISP-DM

El modelo CRISP-DM (IBM, 2021) organiza el trabajo de minería de datos en seis fases principales, que van desde entender el problema hasta poner el modelo final en funcionamiento. La primera fase, llamada Comprensión del negocio, es fundamental. En esta etapa inicial, nos centramos en definir claramente cuáles son los objetivos generales del proyecto y en determinar qué necesitamos analizar exactamente, asegurándonos de que todo esto encaje bien con el contexto específico de nuestro estudio.

Figura 11

Fases de Crisp-DM



Nota. Adaptado de (IBM, 2021).

En la primera fase *Comprensión del negocio*, definimos el objetivo principal de nuestro estudio: buscar la forma de predecir qué estudiantes podrían abandonar los cursos introductorios de programación aquí en CORHUILA.

Luego, en la *fase de Análisis de datos*, examinamos de cerca la información que teníamos. Usamos el cuestionario MSLQ-Colombia para entender las estrategias de aprendizaje y la motivación de los estudiantes. Aplicamos este cuestionario tanto a un grupo de control como a un grupo experimental, cada uno formado por 40 estudiantes. Al analizar sus respuestas, pudimos empezar a identificar patrones relacionados con la confianza en sí mismos (autoeficacia), las estrategias de aprendizaje que usaban y cómo regulaban su esfuerzo. Hicimos también un análisis estadístico inicial de las respuestas al MSLQ,

calculando la media y la desviación estándar para cada subescala en ambos grupos. Esto nos preparó para usar más adelante la prueba de Wilcoxon después de realizar las intervenciones, una prueba estadística adecuada para datos educativos que a veces tienen valores atípicos, y que nos permitirá ver si hubo cambios significativos entre los grupos durante el curso.

La siguiente etapa fue la *Preparación de los datos*. Aquí aplicamos varias técnicas para asegurarnos de que los datos estuvieran limpios y listos para los modelos. Esto incluyó limpiar los datos, quitando valores extraños y completando la información faltante. Como es común que haya un desbalance entre los estudiantes que abandonan y los que no, probamos tres estrategias para equilibrar las clases (Oversample, Middlesample y Undersample) y también evaluamos otras técnicas como SMOTE y Resample. Finalmente, dividimos los datos para entrenar y probar los modelos, usando dos proporciones distintas (80% para entrenar y 20% para probar; y 70% para entrenar y 30% para probar) para ver cómo esta división afectaba el rendimiento final.

Durante la *fase de Modelado*, seleccionamos y entrenamos siete algoritmos de clasificación diferentes: Naive Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), K-Nearest Neighbors (KNN), Random Forest (RF), Multilayer Perceptron (MLP) y XGBoost. Para intentar que los modelos funcionaran lo mejor posible, primero hicimos una selección de características, usando un Árbol de Decisión para identificar las variables que aportaban más (más del 1%) a la predicción. Después, optimizamos los hiperparámetros de cada modelo usando GridSearch. Esta es una técnica que prueba sistemáticamente muchas combinaciones de configuraciones para encontrar la mejor, basándonos en la métrica f1_weighted en nuestro caso (Llanos et al., 2023), Además, probamos diferentes puntos de partida aleatorios (usando valores de 'random state' del 1 al 100) para evaluar cómo esto influía en los resultados.

Para evaluar la efectividad de nuestros modelos predictivos, realizamos una comparación entre sus predicciones y los resultados reales de los estudiantes, utilizando diversas métricas estándar. Empezamos con la precisión, que nos permitió medir cuántas veces el modelo acertaba al predecir que un estudiante estaba en riesgo. Luego, empleamos el recall (o sensibilidad), que nos ayudó a determinar si el modelo era capaz de identificar a

la mayoría de los estudiantes que realmente estaban en riesgo. Adicionalmente, utilizamos el F1 Score, que combina las dos métricas anteriores y nos ofreció una evaluación equilibrada del rendimiento del modelo. Por último, el área bajo la curva (AUC) nos brindó una visión general sobre la capacidad del modelo para diferenciar entre los estudiantes que probablemente abandonarían y aquellos que no.

Durante la *fase de implementación*, probamos el modelo con los datos de los estudiantes en CORHUILA. Nos sorprendió descubrir que el modelo Random Forest alcanzó una precisión del 70.5% en la identificación de estudiantes en riesgo, ya en la cuarta semana del curso. Además, realizamos una comparación entre los resultados del pre-test y el post-test, es decir, antes y después de la intervención, utilizando la prueba de Wilcoxon. Este análisis reveló mejoras significativas en el grupo que recibió la intervención, especialmente en sus estrategias de aprendizaje, la regulación de su esfuerzo y la metacognición. A la luz de estos resultados, se concluye que el modelo es viable para su implementación en estrategias de intervención temprana en cursos de programación, contribuyendo así a mitigar el abandono estudiantil.

Diseño Cuasi-experimental

El diseño cuasi-experimental propuesto por (Fernández et al., 2014), que permite a los investigadores comparar grupos y observar resultados bajo condiciones controladas, aunque sin alcanzar el rigor absoluto de un diseño experimental aleatorizado. En contextos donde la aleatorización no es posible, como en estudios educativos donde los grupos ya están establecidos, la evaluación cuasi experimental resulta útil y práctica.

El diseño cuasi-experimental se caracteriza por su enfoque descriptivo y la inclusión de grupos de tratamiento y de control. Aunque no se basa en una selección aleatoria de sujetos, permite contrastar hipótesis y medir el impacto de ciertas intervenciones o tratamientos mediante la observación de datos cualitativos y cuantitativos. Decidimos optar por un diseño cuasi-experimental para llevar a cabo este estudio. Aunque este enfoque requiere un análisis cuidadoso de los resultados para garantizar su validez, suele ser más

económico y fácil de implementar que un experimento tradicional, ya que permite trabajar con grupos que ya existen (Fernández et al., 2014).

En nuestra investigación en CORHUILA, esto se tradujo en la colaboración con dos grupos: uno experimental y otro de control. Al grupo experimental se le aplicó una intervención específica, basada en nuestro modelo predictivo, que tenía como objetivo identificar a los estudiantes con riesgo de abandonar el curso. Por su parte, el grupo de control continuó con el método tradicional de clases, sin recibir esta intervención particular.

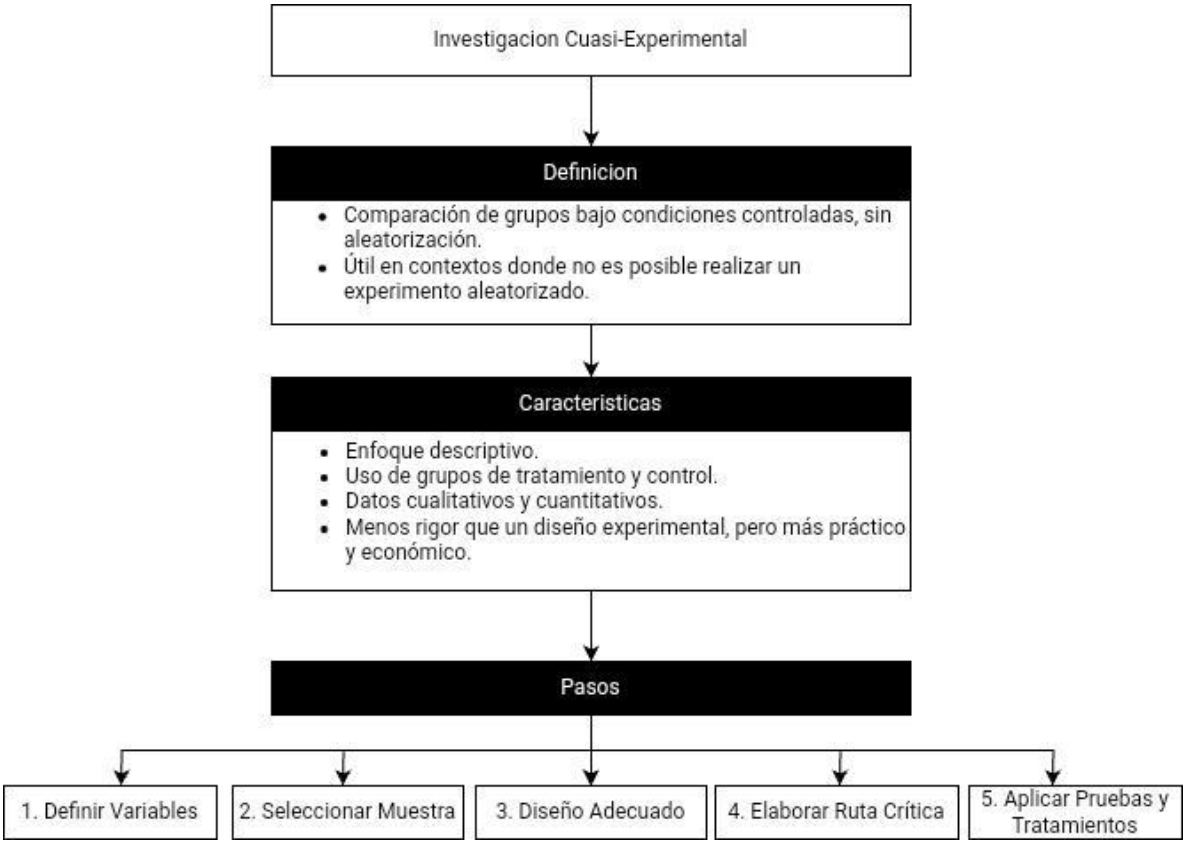
Esta estructura nos brindó la oportunidad de evaluar si el modelo predictivo y el apoyo adicional realmente contribuían a reducir la deserción estudiantil. El proceso se llevó a cabo de la siguiente manera: inicialmente, identificamos a los participantes, quienes ya estaban organizados en sus respectivas clases del curso Fundamentos de Programación. Comenzamos aplicando a todos el cuestionario MSLQ-Colombia como pre-test, para tener una idea inicial de su motivación y estrategias de aprendizaje.

Con base en esos datos, implementamos el modelo predictivo solo en el grupo experimental. A los estudiantes de este grupo que el modelo identificó como de alto riesgo de abandono, les brindamos retroalimentación temprana, así como estrategias de apoyo y seguimiento personalizado. En contraste, el grupo de control no recibió ni esta retroalimentación ni el apoyo adicional. Al final del semestre, volvimos a aplicar el cuestionario MSLQ-Colombia como post-test para medir el impacto de la intervención.

La Figura 12 muestra un diagrama de flujo que describe el procedimiento de la investigación cuasi-experimental.

Figura 12

Diagrama Investigación Cuasi Experimental



Nota. Adaptado de (Hidalgo Suarez et al., 2023).

La recolección de datos en este estudio se llevó a cabo mediante un conjunto de técnicas e instrumentos diseñados para capturar información académica, motivacional y conductual de los estudiantes en cursos de programación.

- **Recolección de datos académicos:** Incluyó calificaciones, tasas de asistencia, y participación en actividades de tutoría y aprendizaje. Estos datos se obtuvieron de los registros institucionales.
- **Cuestionarios:** Se utilizaron para recopilar datos motivacionales y de percepción sobre la programación, evaluando actitudes, frustraciones y expectativas de los estudiantes.

- **Herramientas de análisis de datos:** Se aplicaron algoritmos de minería de datos y aprendizaje automático para identificar patrones y variables clave, utilizando herramientas como Python, R y plataformas especializadas en análisis estadístico.
- **Pretest y postest:** Se aplicaron para evaluar los cambios en los conocimientos y habilidades de los estudiantes antes y después de la intervención.

El análisis de datos se llevó a cabo en varias etapas:

- **Limpieza y preparación de los datos:** Se eliminaron valores faltantes y se transformaron variables categóricas para asegurar la calidad de los datos.
- **Modelado predictivo:** Se emplearon algoritmos de aprendizaje automático, tales como regresión logística, árboles de decisión y redes neuronales, con el fin de identificar patrones y prever el riesgo de abandono estudiantil.
- **Evaluación del modelo:** Se validó la precisión y confiabilidad del modelo mediante métricas como precisión, sensibilidad, especificidad y área bajo la curva ROC.

Motivated Strategies for Learning Questionnaire (MSLQ)

Para la recolección de datos en el contexto colombiano, se empleó una versión adaptada del MSLQ propuesta por (Ramírez-Echeverry et al., 2016). Esta versión consta de 36 preguntas cuidadosamente seleccionadas de las sub-escalas de Estrategias de Aprendizaje y Motivación. El cuestionario se divide en dos secciones principales:

1. Estrategias de Aprendizaje (19 preguntas):
    - Administración del tiempo de estudio (5 ítems)
    - Aprendizaje con pares (7 ítems)
    - Regulación del esfuerzo (4 ítems)
    - Metacognición – seguimiento del aprendizaje (3 ítems)
2. Motivación (17 preguntas):
    - Valoración de la Tarea (6 ítems)

- Metas intrínsecas (3 ítems)
- Expectativas de autoeficacia para el rendimiento (4 ítems)
- Creencias de control del aprendizaje (4 ítems)

Cada pregunta del cuestionario MSLQ se contestó utilizando una escala Likert de 7 puntos. Un '1' significaba que la afirmación no representaba en absoluto al estudiante, mientras que un '7' indicaba que lo describía de manera perfecta. Esta escala nos ofreció una evaluación detallada y matizada de las percepciones y comportamientos de los alumnos.

Implementamos el MSLQ dentro de un diseño longitudinal, lo que implica que queríamos examinar cómo las estrategias de aprendizaje y la motivación de los estudiantes impactaban en su rendimiento académico a lo largo del semestre. Para ello, aplicamos el cuestionario en tres momentos clave: en la semana 4, la semana 8, y la semana 12 del periodo académico.

Estudiar sus respuestas a lo largo del tiempo nos brindó una perspectiva dinámica sobre cómo estos factores podían influir no solo en sus calificaciones, sino también, posiblemente, en su decisión de seguir adelante o abandonar el curso.

Evaluamos el rendimiento académico a través de las calificaciones obtenidas en las actividades del curso de Fundamentos de Programación. Utilizamos la escala habitual de 0.0 a 5.0, donde 3.0 es la nota mínima para aprobar. Esto nos proporcionó un indicador objetivo del desempeño que luego pudimos relacionar con los resultados del MSLQ.

A partir de los datos recopilados con el cuestionario MSLQ-Colombia en estas semanas clave, así como de las calificaciones de los estudiantes durante esos mismos periodos, elaboramos una clasificación del riesgo de abandono:

- 0: Alto riesgo de abandono (calificaciones entre 0.0 y 2.9)
- 1: Medio riesgo de abandono (calificaciones entre 3.0 y 4.0)
- 2: Bajo riesgo de abandono (calificaciones entre 4.1 y 5.0)

Esta clasificación permite identificar de manera temprana a los estudiantes que podrían estar en riesgo de abandonar el curso, facilitando intervenciones oportunas y personalizadas.

Prueba de Wilcoxon

La prueba de Wilcoxon, también conocida como la prueba de rango con signo de Wilcoxon (Turcios, 2015), es una herramienta estadística de tipo no paramétrico. Se utiliza principalmente para comparar dos grupos de datos que están relacionados entre sí, o para ver si hay diferencias significativas en un mismo grupo antes y después de aplicar alguna intervención (como en nuestro caso con el pre-test y post-test). En el contexto de investigaciones como esta, que buscan predecir el rendimiento académico, la prueba de Wilcoxon es útil para confirmar si las diferencias que observamos en las predicciones del modelo, son realmente significativas desde el punto de vista estadístico, o si podrían deberse simplemente al azar.

La prueba parte de dos hipótesis:

- **Hipótesis nula (H_0):** No existen diferencias significativas entre las dos muestras relacionadas, es decir, cualquier diferencia observada es atribuible al azar.
- **Hipótesis alternativa (H_1):** Existen diferencias significativas entre las dos muestras relacionadas, lo que sugiere que los cambios observados son sistemáticos y no producto del azar.

Una gran ventaja de la prueba de Wilcoxon es que es robusta, es decir, funciona bien incluso si los datos no siguen una distribución normal perfecta. Esto la hace muy adecuada para analizar datos educativos, donde es común encontrar distribuciones asimétricas o valores atípicos. Al aplicar esta prueba en nuestro estudio, podemos respaldar la eficacia del modelo predictivo. ¿Cómo? Demostrando que las categorías de riesgo (alto y bajo) que predice el modelo están claramente separadas en función de los indicadores que usamos (como las notas iniciales o los datos del MSLQ). Esto refuerza la utilidad del modelo para identificar a los estudiantes en riesgo de abandono en nuestros cursos de fundamentos de programación aquí en CORHUILA, porque nos permite validar que las predicciones no solo son precisas, sino que las diferencias que encuentra son estadísticamente significativas, no fruto del azar.

Cronograma de Actividades

El cronograma de nuestro proyecto abarcó desde febrero de 2024 hasta enero de 2025, y lo estructuramos en varias fases clave, como se puede ver en las Figuras 13 y 14.

La primera fase, que iniciamos en febrero de 2024, se centró en seleccionar las bases de datos donde buscaríamos artículos, formular la pregunta de investigación y definir la ecuación de búsqueda. Esta etapa fue fundamental para sentar una base sólida para la revisión de la literatura y el marco conceptual del estudio.

Durante marzo y abril de 2024, nuestro enfoque principal fue la recolección de artículos siguiendo la metodología PRISMA. Este proceso incluyó identificar, seleccionar preliminarmente (tamización), elegir e incluir los artículos más relevantes para nuestro tema. Al mismo tiempo, empezamos a construir el Thesaurus, que nos serviría como herramienta de referencia al analizar la literatura, y también elaboramos un informe sobre los algoritmos y métricas que más se utilizaban en los estudios publicados.

Desde mayo hasta septiembre de 2024, nos dedicamos al modelado del abandono estudiantil. Esto implicó dividir los datos en diferentes particiones para entrenar y probar los modelos de predicción. En este periodo, también aplicamos técnicas para balancear los datos (cuando las clases no están equilibradas), como la librería SMOTE, y seleccionamos las características más importantes para el modelo evaluando diferentes estados aleatorios.

Entre mayo y noviembre de 2024, llevamos a cabo la fase de hiperparametrización. Aquí ajustamos finamente los modelos predictivos, usando métricas como el F1 Score y validación cruzada para evaluar su rendimiento. También aplicamos otras técnicas de balanceo de datos como oversampling, middlesampling y undersampling. Esta etapa fue crucial para refinar el modelo y asegurar que fuera lo más preciso posible.

Finalmente, desde junio de 2024 hasta enero de 2025, se lleva a cabo la evaluación del modelo. Esta última fase incluye la selección del modelo final, construcción de pruebas, y aplicación de cuestionarios post-test. El proyecto concluye la redacción del documento de tesis en enero de 2025.

Figura 13

Cronograma de Actividades del Proyecto

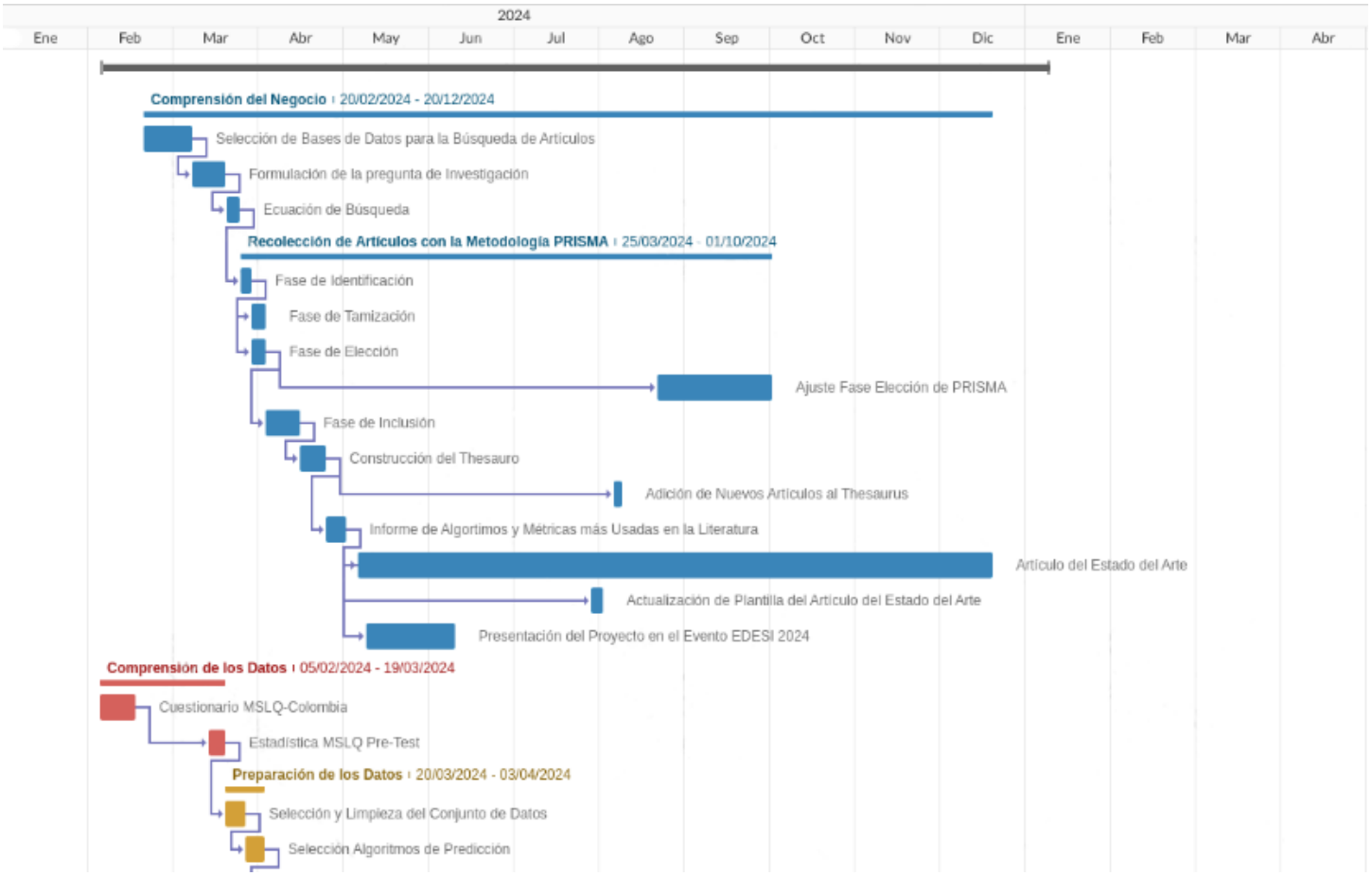
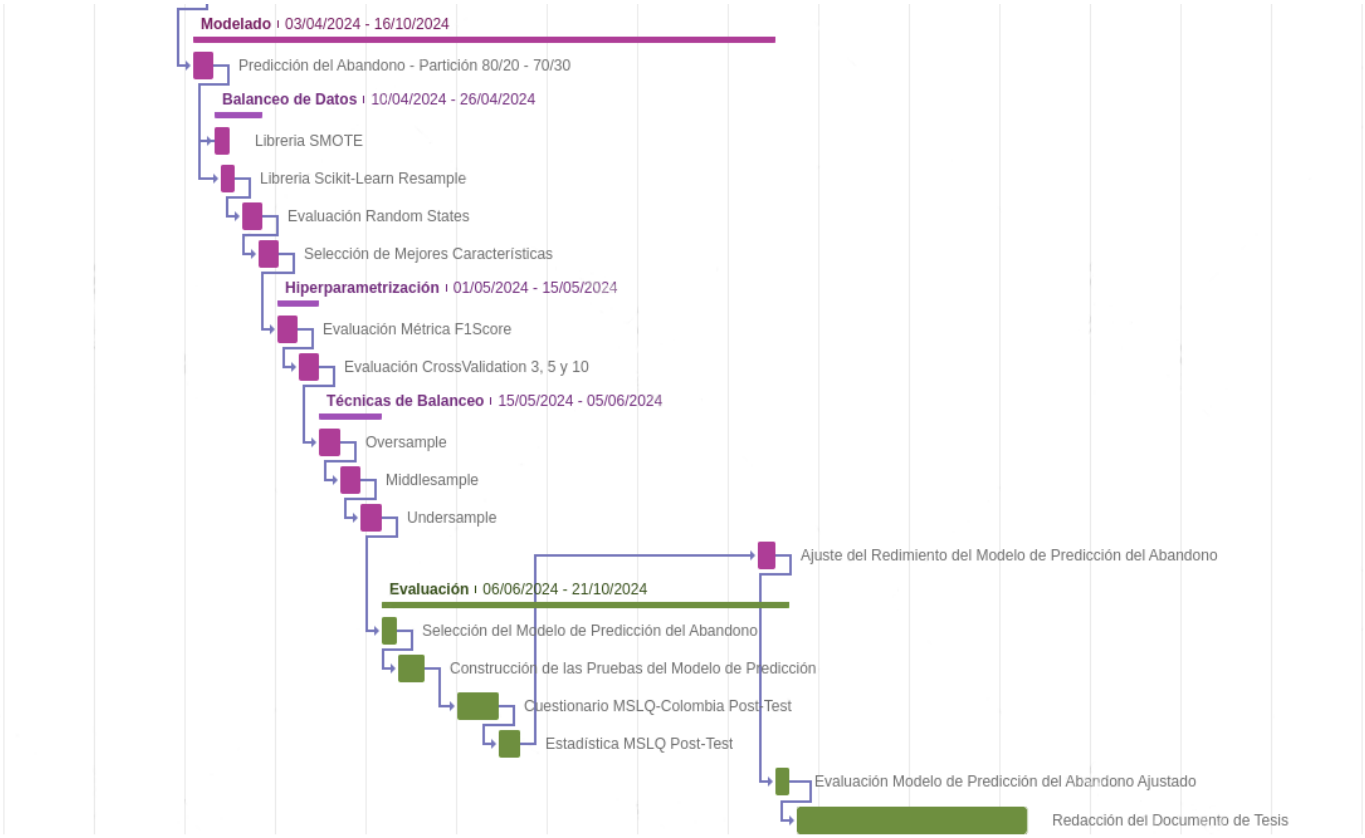


Figura 14

Cronograma de Actividades del Proyecto


Modelado : 03/04/2024 - 16/10/2024
Predicción del Abandono - Partición 80/20 - 70/30
Balanceo de Datos : 10/04/2024 - 26/04/2024
Libreria SMOTE
Libreria Scikit-Learn Resample
Evaluación Random States
Selección de Mejores Características
Hiperparametrización : 01/05/2024 - 15/05/2024
Evaluación Métrica F1Score
Evaluación CrossValidation 3, 5 y 10
Técnicas de Balanceo : 15/05/2024 - 05/06/2024
Oversample
Middlesample
Undersample
Ajuste del Redimiento del Modelo de Predicción del Abandono
Evaluación : 06/06/2024 - 21/10/2024
Selección del Modelo de Predicción del Abandono
Construcción de las Pruebas del Modelo de Predicción
Cuestionario MSLQ-Colombia Post-Test
Estadística MSLQ Post-Test
Evaluación Modelo de Predicción del Abandono Ajustado
Redacción del Documento de Tesis

Resultados y discusión

Comprensión de los Datos

Con el objetivo de empezar a identificar qué variables estaban vinculadas al rendimiento y al abandono de los estudiantes en los cursos de fundamentos de programación, en esta fase de CRISP-DM realizamos un examen a fondo del entorno académico aquí en CORHUILA. Completamos varias tareas en esta etapa para organizar la investigación y sentar unas bases sólidas para las fases siguientes de construcción del modelo predictivo.

El cuestionario MSLQ-Colombia fue igualmente una de las tareas que llevamos a cabo en el grupo de control como el grupo experimental (Ramírez-Echeverry et al., 2016). Esta herramienta, pensada para evaluar motivación y estrategias de aprendizaje, se utilizó para recoger la información relacionada con los hábitos de estudio y la autoeficacia (confianza en sus propias capacidades) y la regulación del esfuerzo del propio estudiante, datos que nos fueron útiles para efectuar las primeras aproximaciones en la detección de tendencias en relación con el rendimiento académico.

Adicionalmente, preparamos el análisis de los resultados del cuasi-experimento. Esto implicó comparar estadísticas clave entre los grupos, como la media, la diferencia de medias y la desviación estándar, con el objetivo de tener todo listo para realizar posteriormente la prueba de Wilcoxon y calcular el p-valor. Esta prueba nos permitiría determinar si existió una diferencia estadísticamente significativa entre el grupo de control y el experimental durante el desarrollo del curso Fundamentos de Programación, comparando los resultados antes y después de haber realizado la intervención con el modelo predictivo, y fijándonos en las subescalas relevantes del MSLQ (Hsu et al., 2016). Además, se realizó un análisis estadístico de los datos obtenidos en el MSLQ en el Pre-Test y Post-Test, como se muestra en la Tabla 1.

Tabla 1

Estadísticas MSLQ Colombia

MSLQ	Subescala	Grupo	Pre-test (media)	Post-test (media)	Diferencia	Pre-test (STD)	Post-test (STD)	Wilcoxon (p-value)
Estrategias de aprendizaje	1. Administración del tiempo de estudio	Grupo de Control	5.23	5.64	0.41	1.35	1.15	**6.349E-03***
		Grupo Experimental	5.15	5.84	0.69	1.31	1.02	**2.776E-05***
	2. Aprendizaje por Pares	Grupo de Control	4.49	5.25	0.76	2.05	1.81	**1.864E-05***
		Grupo Experimental	5.48	5.71	0.23	1.48	1.51	1.309E-01
	3. Regulación del Esfuerzo	Grupo de Control	5.76	6.24	0.48	1.04	0.83	**3.703E-05***
		Grupo Experimental	6.38	6.47	0.09	0.96	1.02	4.891E-01
	8. Metacognición – Seguimiento del Aprendizaje	Grupo de Control	5.81	6.25	0.44	1.23	0.91	**1.046E-02***
		Grupo Experimental	6.49	6.47	-0.02	0.92	0.75	8.856E-01
Motivación	12. Valoración de la Tarea	Grupo de Control	5.98	6.32	0.34	1.21	1.1	**1.133E-03***
		Grupo Experimental	6.44	6.64	0.2	1.26	0.67	6.617E-02
	13. Metas Intrínsecas	Grupo de Control	5.46	5.73	0.27	1.29	1.17	1.491E-01
		Grupo Experimental	6.22	6.31	0.09	1.05	1.06	6.240E-01

16. Expectativas de Autoeficacia para el Rendimiento	Grupo de Control	5.93	5.92	-0.01	0.99	1.34	7.420E-01
	Grupo Experimental	6.54	6.37	-0.17	0.95	1	1.069E-01
17. Creencias de Control del Aprendizaje	Grupo de Control	6.19	6.19	0	0.93	0.91	9.580E-01
	Grupo Experimental	6.69	6.41	-0.28	0.83	0.96	**2.588E-02***

Un valor p (p-value) inferior a 0.05 en esta prueba nos indicó que hubo una diferencia o mejora estadísticamente significativa en la sub-escala analizada. Las sub-escalas donde encontramos estas diferencias significativas fueron:

- **Administración del Tiempo de Estudio:** Aquí observamos una mejora significativa en general. El análisis de los datos sugirió que los estudiantes, después del periodo de estudio, mejoraron en su capacidad para planificar y gestionar su tiempo, lo que indica que se volvieron más hábiles para distribuir y priorizar sus tareas. Este progreso apunta a un mayor compromiso con el estudio y una mejor habilidad para controlar distracciones y organizar sesiones de estudio más productivas.
- **Aprendizaje por Pares:** Observamos cómo trabajaban juntos los estudiantes. En este sentido, el grupo de control fue en el que los estudiantes trabajaron significativamente más con su grupo de compañeros. Una posible explicación es que el grupo control deja un ambiente de compañerismo que facilita que se pregunten y se resuelvan cuestiones mutuamente. Este resultado puede sugerir que la colaboración puede ser muy importante para el rendimiento.
- **Regulación del Esfuerzo:** Esta sub-escala mide en qué medida los alumnos muestran dedicación y atención durante la realización de las tareas, ante la presencia de distracciones o escaso interés por la tarea en sí. En el mismo sentido, el grupo de control mostró niveles de dedicación también significativamente más altos, aunque se podría decir que alcanzaron una mayor capacidad para mantener la atención y

continuar esforzándose durante la tarea que habría de desempeñar aún en los casos de dificultades, lo que tal vez podría reflejar una mayor resistencia y autocontrol.

- **Metacognición – Seguimiento del Aprendizaje:** Esto se refiere a si los estudiantes monitorean y evalúan su propio aprendizaje. El grupo de control demostró una capacidad significativamente mayor para autoevaluarse. Estos estudiantes parecían ser más conscientes de sus fortalezas y debilidades y de identificar áreas donde necesitaban mejorar, lo cual es clave para ajustar las estrategias de estudio.
- **Valoración de la Tarea:** Esto mide qué tan útil consideran los estudiantes las tareas del curso. El grupo de control valoró significativamente más las tareas asignadas como importantes para su formación. Esto sugiere que creían que las actividades les ayudarían en su progreso académico, lo que podría llevar a un mayor compromiso y calidad en su trabajo.
- **Creencias de Control del Aprendizaje:** Esta sub-escala determina si los alumnos piensan que su aprendizaje se debe a su esfuerzo o, en cambio, a factores externos. En este caso, el grupo experimental fue el que mostró niveles significativamente más altos en variables como la confianza en su control. Los alumnos creyeron más firmemente en su capacidad para mejorar sus resultados académicos mediante su esfuerzo y dedicación. Este es un resultado esperanzador, que pone de manifiesto la importancia de fomentar que los alumnos sientan que tienen control sobre su aprendizaje.

Preparación de los Datos

En esta fase de nuestro proyecto, abordamos la preparación de los datos que utilizaríamos para crear el modelo de predicción de la deserción en la asignatura de Fundamentos de Programación. Esto supuso la recolección de la información necesaria; en una etapa posterior limpiamos y equilibramos dichos datos para asegurarnos de que fueran de calidad y que fueran representativos de nuestros estudiantes.

Nuestra principal fuente de información fue una versión modificada del Cuestionario de Estrategias Motivadas para el Aprendizaje (MSLQ), adaptada específicamente para el contexto aquí en Colombia. Esta encuesta constaba de 36 preguntas, donde los estudiantes respondían en una escala Likert de 7 puntos (1 significaba que la afirmación no los describía en absoluto y 7 que los describía perfectamente). Esto nos permitió evaluar en detalle su motivación y las tácticas de aprendizaje que empleaban.

Además del cuestionario, también recopilamos las calificaciones que obtuvieron los alumnos en los tres cortes de evaluación principales del curso: corte 1, corte 2 y corte 3. El riesgo de abandono, la variable objetivo, se clasificó en tres grupos: riesgo bajo (notas entre 4.1 y 5.0), riesgo medio (notas entre 3.0 y 4.0) y riesgo alto (notas entre 0.0 y 2.9). Las respuestas al cuestionario dadas a los alumnos de los grupos de control y experimental, junto con las notas y los resultados de años anteriores, se combinaron finalmente en un conjunto de datos de 207 entradas.

El paso siguiente es realizar una limpieza a la información del dataset, para este caso se eliminaron las columnas que no aportan a la creación del algoritmo de predicción como la hora en que el estudiante completó el cuestionario, su dirección de correo electrónico y el nombre completo. También, se debe realizar una imputación de datos NaN o faltantes, que para nuestro caso no se encontró ninguno.

Dado que es recomendado utilizar modelos de clasificación para predecir el riesgo de abandono de los estudiantes en el curso, es necesario balancear la información para evitar el sobreajuste de los modelos y que no favorezcan una clasificación sobre las otras (Hidalgo

Suarez et al., 2023; Martins et al., 2023). Por este motivo, se utilizó resample como técnica para balancear los datos.

La información fue balanceada de tres formas:

- **Oversample:** Implica aumentar el número de datos de las clases minoritarias para igualar o acercarse al número de datos de la clase mayoritaria.
- **Middlesample:** Método intermedio para lograr un equilibrio más natural entre el Oversampling y el Undersampling, evitando el sobreajuste o la pérdida de información.
- **Undersample:** Implica reducir el número de datos de las clases mayoritarias para igualar o acercarse al número de datos de la clase minoritaria.

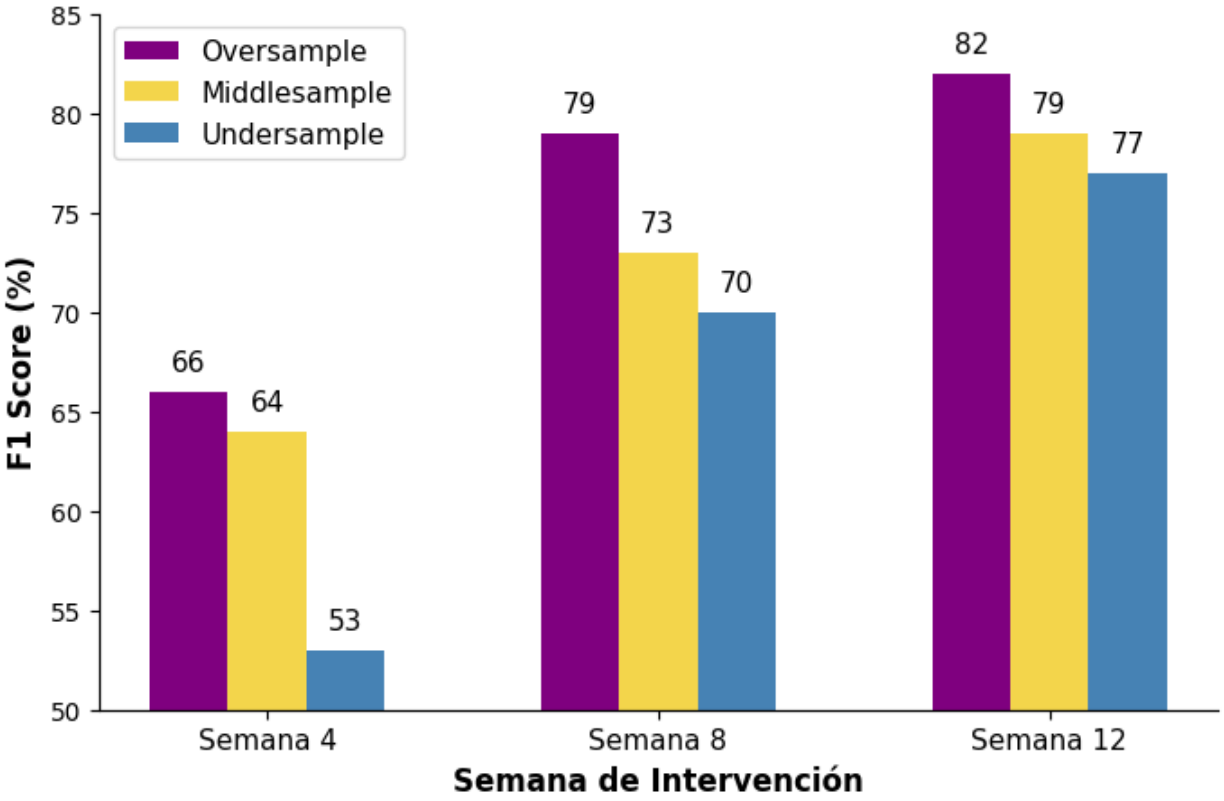
Los resultados mostrados en la Figura 15 hacen referencia al promedio obtenido en la métrica de F1 Score de todos los algoritmos después de balancearlos con random state de 50 y una partición de 80/20. Se puede observar que la técnica de balanceo por *oversample* fue la más efectiva para mejorar el rendimiento de los modelos de clasificación en la predicción del abandono académico donde obtuvo un F1 Score del 66% en la semana 4, lo que lo convierte en la mejor opción para una intervención temprana, en la semana 8 el F1 Score fue del 79%, y en la semana 12 fue 82%, lo que demuestra la estabilidad del balanceo oversample en distintos momentos del curso.

En comparación, middlesample y undersample presentaron un rendimiento inferior en las primeras semanas. Por ejemplo, el promedio con middlesample alcanzó un 64% de F1 Score en la semana 4, y con undersample, el resultado en esa misma semana fue 54% lo que indica un desempeño menos confiable para intervenciones tempranas.

De la Figura 15 también podemos concluir que al pasar las semanas el rendimiento de los modelos aumentaba a medida que se adicionaban más características, en este caso para oversample de la semana 4 a la 8 hubo una mejora del 13% en la predicción gracias a que se agregó la nota del segundo corte del curso. Pasa lo mismo de la semana 8 a la 12 donde el rendimiento aumenta en un 3% al agregar la nota del tercer corte.

Figura 15

Comparación Balanceo de los Modelos



Para garantizar que los modelos de predicción del abandono académico no se vean afectados por un desbalance en las clases, es fundamental evaluar la efectividad de distintas técnicas de balanceo (Llanos et al., 2023). En este sentido, se compararon dos enfoques ampliamente utilizados: SMOTE y Resample. Esta comparación permite determinar cuál de estas técnicas ofrece un mejor desempeño en la clasificación de los estudiantes en riesgo de abandono.

Los resultados que se exponen en la Tabla 2 son el promedio de todos los algoritmos para cada métrica en las particiones 80/20 y 70/30. Se pueden apreciar diferencias significativas entre los datos balanceados con SMOTE y aquellos balanceados con Resample, únicamente para semana objetivo (semana 4) la diferencia en la partición 80/20 fue del 6% en la métrica de F1 Score, de igual forma, para la partición 70/30 la diferencia fue del 10%

en la métrica de F1 Score siendo Resample una mejor técnica de balanceo que SMOTE para nuestro conjunto de datos.

Tabla 2

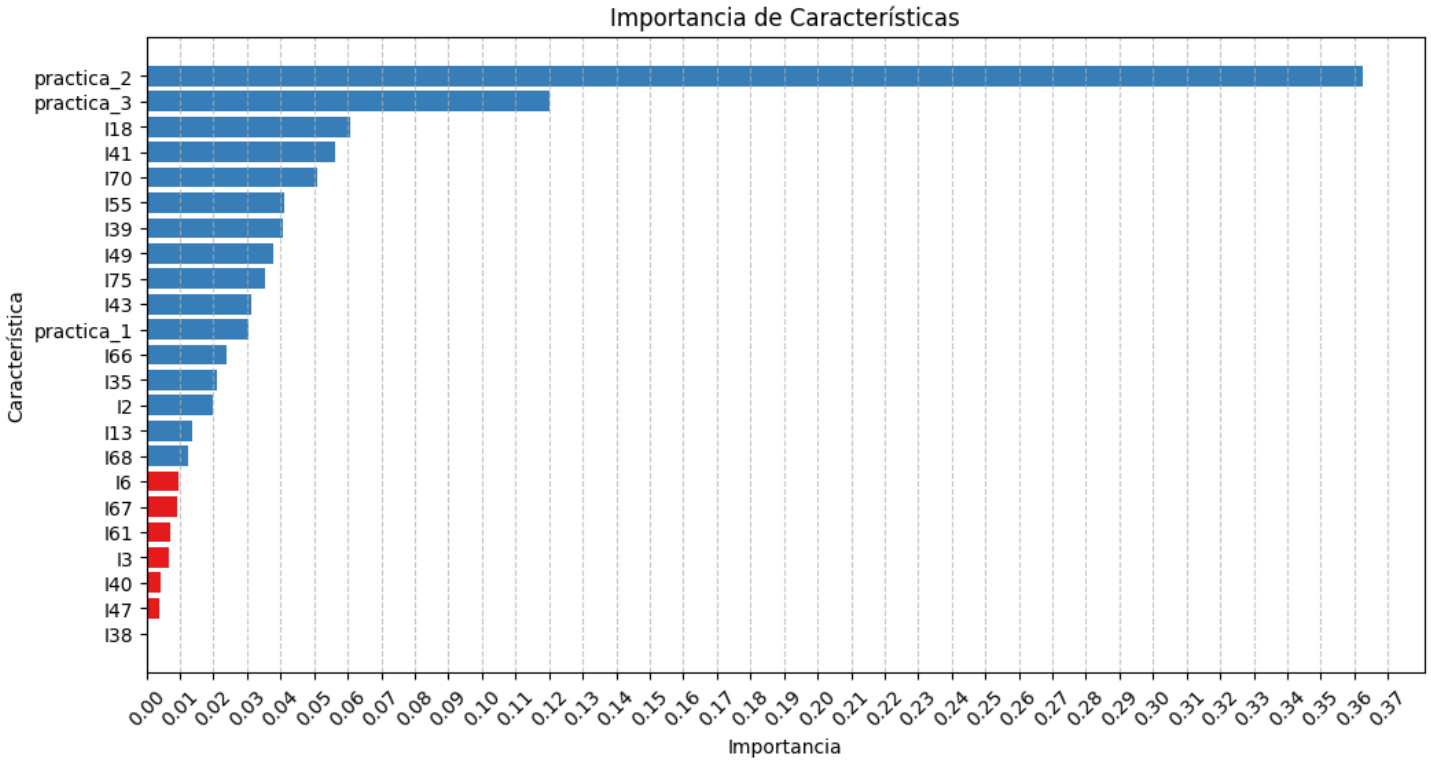
Balanceo Técnicas SMOTE y Resample

Semana	Métrica	Partición 80/20		Diferencia	Partición 70/30		Diferencia
		SMOTE	Resample		SMOTE	Resample	
4	Precisión	0.53	0.58	5%	0.5	0.6	10%
	Recall	0.51	0.57	6%	0.5	0.6	10%
	F1 Score	0.50	0.56	**6%**	0.49	0.59	**10%**
8	Precisión	0.62	0.69	7%	0.61	0.7	9%
	Recall	0.62	0.67	5%	0.6	0.7	10%
	F1 Score	0.62	0.66	**4%**	0.6	0.69	**9%**
12	Precisión	0.67	0.77	10%	0.68	0.74	6%
	Recall	0.65	0.75	10%	0.66	0.75	9%
	F1 Score	0.65	0.75	**10%**	0.66	0.74	**8%**

En el proceso de creación de modelos de clasificación, es fundamental identificar y seleccionar las características que realmente contribuyen al rendimiento del modelo (Llanos et al., 2023). Esto se debe a que algunas características o datos pueden no aportar mejoras significativas, e incluso pueden reducir la eficiencia del modelo (Jamjoom et al., 2021). Para abordar este problema, se llevó a cabo una selección de las mejores características, con el objetivo de identificar cuáles variables tienen un mayor impacto en la creación de un modelo predictivo. Se utilizó el algoritmo *Decision Tree* como base de pruebas para seleccionar las características que aportan más del 1% al rendimiento del modelo, en la Figura 16 se pueden observar las características seleccionadas representadas en azul, mientras que aquellas no seleccionadas se muestran en rojo.

Figura 16

Peso Características Decision Tree



Modelado

En esta etapa, se seleccionaron y entrenaron distintos algoritmos de clasificación con el objetivo de evaluar su desempeño en la predicción del riesgo de abandono. Para ello, se evaluaron 7 algoritmos, siendo estos de tipo matemáticos, computacionales, redes neuronales y ensambles. Además, se analizaron diferentes configuraciones de partición de datos y estrategias de hiperparametrización con el objetivo de optimizar la precisión de las predicciones.

Se probaron siete técnicas de clasificación: matemáticos (Naive Bayes y Support Vector Machine), computacionales (Decision Tree y K-Nearest Neighbor), una red neuronal (Multilayer Perceptron) y de ensamble (Random Forest y XGBoost). Para evaluar la eficacia de los modelos se utilizaron tres métricas: la precisión, la recuperación, la puntuación F1 y

el área bajo la curva (AUC), que muestra la capacidad del modelo para diferenciar entre clases positivas y negativas con distintos umbrales de probabilidad (Llanos et al., 2023).

Los datos se dividieron en dos configuraciones de partición para evaluar el rendimiento de los modelos: 70% para entrenamiento y 30% para pruebas, y 80% para entrenamiento y 20% para pruebas (Abdulazeez & Abdulwahab, 2019). El objetivo de esta estrategia era determinar qué partición apoya mejor el rendimiento de los modelos de predicción y elegir la que produce los mejores resultados.

Tanto la partición de los datos como el proceso de resampling se modificaron mediante un valor conocido como “random state”, el cual permite mezclar o revolver la información para obtener resultados diversos en el rendimiento de los modelos de predicción (Alsulami et al., 2023). Los valores de random state utilizados fueron: 1, 3, 5, 6, 10, 42, 50 y 100.

Cada modelo fue sometido a un proceso de hiper-parametrización mediante GridSearch para encontrar la configuración que proporciona el mejor rendimiento. La variable de puntuación utilizada en GridSearch fue el "f1_weighted", que equilibra el rendimiento considerando la frecuencia de las etiquetas (Hsu et al., 2016). Esta métrica es útil cuando se busca un balance entre la importancia de las etiquetas más frecuentes y las menos frecuentes.

Al final de todo este proceso se generó la Tabla 3, la cual muestra el rendimiento de los siete algoritmos seleccionados a lo largo de las semanas 4, 8 y 12, utilizando la técnica de balanceo Oversample y un random state de 42.

Tabla 3

Evaluación Modelos de Clasificación Corhuila

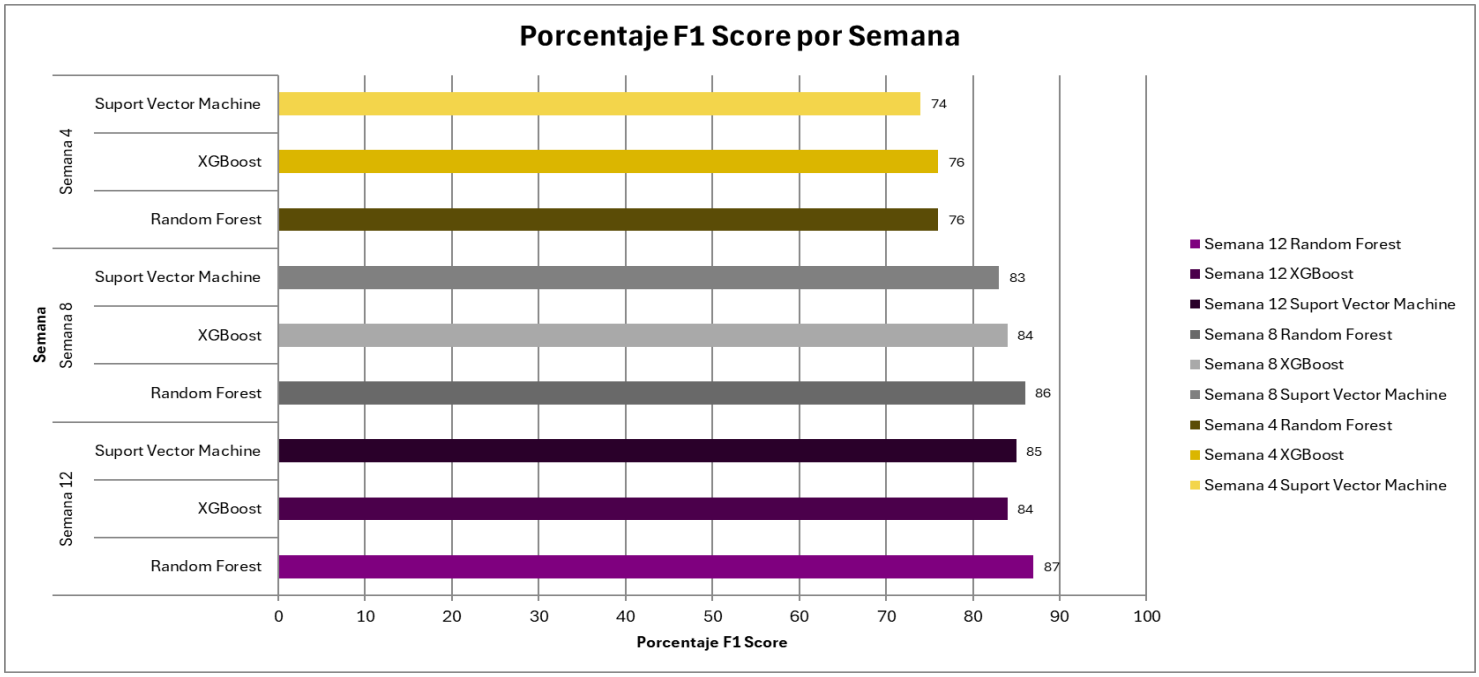
Semana	Métrica	NB	SVM	DT	KNN	RF	MLP	XGBoost
4	Precisión	0.6	0.74	0.71	0.68	**0.78**	0.62	0.78
	Recall	0.57	0.74	0.7	0.68	**0.77**	0.62	0.77
	F1 Score	0.55	0.74	0.69	0.68	**0.76**	0.58	0.76
	AUC	0.73	0.87	0.78	0.76	**0.93**	0.77	0.92
8	Precisión	0.55	0.83	0.79	0.76	**0.88**	0.8	0.87
	Recall	0.57	0.83	0.79	0.75	**0.87**	0.77	0.85
	F1 Score	0.52	0.83	0.79	0.75	**0.86**	0.76	0.84
	AUC	0.84	0.94	0.84	0.82	**0.97**	0.91	0.97
12	Precisión	0.65	0.85	0.79	0.81	**0.88**	0.83	0.87
	Recall	0.66	0.85	0.77	0.75	**0.87**	0.77	0.85
	F1 Score	0.64	0.85	0.76	0.74	**0.87**	0.75	0.84
	AUC	0.86	0.97	0.85	0.82	**0.98**	0.97	0.96

Evaluación

En la fase de evaluación se comparan las predicciones realizadas por los modelos con datos reales, permitiendo determinar su precisión y confiabilidad. En la Tabla 3, podemos observar que Random Forest destaca por su rendimiento a lo largo de todas las semanas, considerando su F1 Score: 76% en la semana 4; 86% en la semana 8 y 87% en la semana 12. En segundo lugar, se encuentra XGBoost, que alcanzó un F1 Score del 76% en la semana 4 y del 84% en las semanas 8 y 12. En tercer lugar, está Support Vector Machine, con un F1 Score del 74% en la semana 4, 83% en la semana 8 y 85% en la semana 12. La Figura 17 muestra el rendimiento de los teniendo en cuenta la métrica de F1 Score.

Figura 17

Modelos con Mejor Rendimiento por Semana Según su F1 Score



Una vez identificados los tres algoritmos con mejor rendimiento para la predicción temprana del abandono, es necesario validar las predicciones de cada uno con datos reales. Para este propósito, se utilizaron 17 datos recolectados del cuestionario MSLQ Colombia en la semana 4. La Tabla 4 muestra el valor real del riesgo de abandono al final del curso y lo compara con las predicciones realizadas por los modelos de Support Vector Machine, Random Forest y XGBoost. El modelo que presentó la mejor predicción en la semana 4 fue Random Forest, con un total de 12 predicciones correctas, lo que equivale a un 70.5% de precisión.

Tabla 4

Evaluación Predicción de los Modelos

	Predicción					
Valor Real	**Support Vector Machine**		**Random Forest**		**XGBoost**	
	Valor Predicción	**Equivalente**	**Valor Predicción**	**Equivalente**	**Valor Predicción**	**Equivalente**
1	1	Si	1	Si	1	Si
2	1	No	1	No	1	No
1	1	Si	0	No	1	Si
2	0	No	2	Si	1	No
1	1	Si	1	Si	1	Si
1	1	Si	1	Si	1	Si
2	1	No	2	Si	2	Si
2	1	No	2	Si	0	No
2	1	No	1	No	1	No
2	0	No	1	No	0	No
1	1	Si	1	Si	2	No
1	1	Si	1	Si	1	Si
1	1	Si	1	Si	1	Si
2	1	No	1	No	1	No
1	1	Si	1	Si	0	No
1	2	No	1	Si	1	Si
1	1	Si	1	Si	1	Si
Porcentaje Equivalente	52.9%		70.5%		52.9%	

Con las pruebas realizadas podemos concluir que el modelo que permite identificar a los estudiantes que presentan una alta probabilidad de abandonar el curso de fundamentos

de programación de la Corhuila en la semana cuatro (4) es Random Forest con un F1 Score del 76% (métrica principal de la investigación) para la técnica de balanceo de datos Oversample, apoyada en Resample; en la partición 80% entrenamiento – 20% pruebas y con random state de 42. Los Hiper-Parámetros utilizados para este modelo fueron:

- bootstrap: False
- max_depth: 10
- min_samples_split: 5

La diferencia de rendimiento observada entre la fase de entrenamiento y la validación con datos reales puede atribuirse a varios factores. En primer lugar, el modelo pudo haber aprendido ciertos patrones específicos del conjunto de entrenamiento que no se replican en el conjunto de prueba, lo que indica una posible sobre adaptación a los datos de entrenamiento (overfitting) (Montesinos López et al., 2022). En segundo lugar, puede haber más variación en la estimación del rendimiento real del modelo debido al tamaño tan reducido del conjunto de validación (17 muestras). Por último, sería necesario examinar otras técnicas de preprocesamiento o evaluar el modelo utilizando un conjunto de validación mayor, ya que la estrategia de equilibrado puede haber modificado la estructura de los datos de entrenamiento, creando un sesgo que no se reflejó en los datos reales (Mohammed et al., 2020).

Conclusiones

Los resultados obtenidos a lo largo de este estudio permitieron el desarrollo del modelo de predicción del rendimiento académico, el cual hizo posible identificar a los estudiantes con alto índice de abandono en cursos de programación, alcanzando así los objetivos, de igual forma, el modelo representa un paso fundamental para la implementación de estrategias preventivas contra el abandono en el ámbito educativo. Así mismo, se examinaron los distintos algoritmos de clasificación y regresión, destacando la efectividad de modelos como Random Forest, Support Vector Machine y XGBoost, los cuales mostraron niveles considerables de precisión en la predicción del rendimiento académico.

También concluimos que el cuestionario MSLQ-Colombia fue esencial para incorporar las estrategias de aprendizaje y la motivación de los estudiantes en la construcción del modelo predictivo. Su aplicación longitudinal en diferentes tiempos del semestre permitió que se obtuvieran datos dinámicos que enriquecieron el modelo al captar cómo estos factores evolucionaban con el tiempo en el contexto de CORHUILA. Asimismo, al tener en cuenta las respuestas del MSLQ y el rendimiento académico nos permitió categorizar a los estudiantes según su riesgo de abandono, lo cual facilita que se haga un apoyo rápido y personalizado.

Tras la aplicación del MSLQ se llevó a cabo un diseño cuasi-experimental con mediciones de pre-test y post-test para evaluar el impacto de las intervenciones, y se utilizó la prueba de rangos con signo de Wilcoxon (Sánchez Turcios, 2015) para muestras relacionadas como método de validación estadística. Los valores de p obtenidos, inferiores a 0.05 en varias subescalas del cuestionario utilizado, revelan que se produjeron mejoras estadísticamente significativas principalmente en la Administración del Tiempo de Estudio, la Regulación del Esfuerzo y las Creencias de Control del Aprendizaje. Tales mejoras reflejan que los estudiantes fortalecieron habilidades clave como la planificación del estudio, la perseverancia frente a distracciones y la creencia en su capacidad para influir en su propio rendimiento académico, lo que refuerza la efectividad del modelo tanto en su capacidad

predictiva como en su impacto positivo sobre factores psicoeducativos relevantes para prevenir el abandono.

La metodología CRISP-DM fue fundamental en cada una de las etapas del proceso de construcción del modelo. Nos permitió asegurar la calidad de los datos, la optimización de los algoritmos y mantener el proyecto alineado con los objetivos. La aplicación de modelos de aprendizaje automático demostró claramente que la capacidad de predicción del sistema mejoraba al combinar factores académicos y motivacionales. En concreto, determinamos que los resultados de las evaluaciones iniciales del curso son un factor predictivo significativo, lo que enfatiza el potencial de las intervenciones iniciales para disminuir el riesgo de abandono.

Asimismo, el enfoque metodológico basado en técnicas de Learning Analytics (LA) y Educational Data Mining (EDM) permitió comprender de forma más profunda los factores que influyen en el rendimiento académico, integrando variables tanto académicas como de comportamiento. Esta perspectiva integral facilitó la construcción de un modelo más robusto, capaz de adaptarse a diferentes contextos educativos.

Nuestra investigación proporcionó pruebas concluyentes de que combinar factores internos de cada estudiante (como la motivación y las estrategias de aprendizaje) con otros parámetros individuales (como el rendimiento en evaluaciones iniciales) conduce a mejores predicciones del rendimiento académico, incluso superando a modelos más antiguos que a menudo se centraban solo en el rendimiento pasado. También demostramos que las técnicas de aprendizaje automático y minería de datos son herramientas viables y útiles en la educación superior cuando se adaptan cuidadosamente a desafíos educativos específicos como los que enfrentamos aquí. Además, esta investigación ofrece un ejemplo metodológico práctico que puede guiar el desarrollo de futuros sistemas de alerta temprana en instituciones académicas y servir de referencia para investigadores que apliquen estos métodos en otros lugares.

De forma complementaria, la evolución de los resultados por semana (semana 4, 8 y 12) evidencia que la efectividad del modelo mejora a medida que se dispone de más información, especialmente las notas obtenidas en cada corte. Esto tiene implicaciones

prácticas: un modelo como este permite intervenciones diferenciadas según el momento del semestre. Por ejemplo, aunque en la semana 4 Random Forest se alcanzó un F1 Score del 76% (con oversample), lo cual es adecuado para alertas tempranas, en la semana 12 esta métrica alcanzó un 87%, lo que demuestra una mayor precisión con datos acumulados.

Una de las principales conclusiones de nuestro trabajo, también, es la clara necesidad de que las instituciones desarrollen intervenciones, basadas en datos, que ayuden a reducir las tasas de abandono, principalmente en cursos exigentes, como son los de programación. Al poder detectar a los estudiantes en riesgo con suficiente antelación, las instituciones pueden crear planes de acompañamiento académico, lo que permite mejorar de manera considerable las tasas de retención y rendimiento.

Para que este modelo sea verdaderamente útil en la práctica, recomendamos que futuras investigaciones se centren en integrarlo en plataformas reales de gestión académica, quizás comenzando aquí en CORHUILA. Esto nos permitiría evaluar adecuadamente su impacto en las decisiones institucionales y su eficacia real para reducir las tasas generales de abandono. También sugerimos explorar herramientas interactivas de visualización de datos; estas podrían facilitar mucho que los profesores y administradores comprendan las predicciones del modelo y las utilicen para diseñar un apoyo eficaz y basado en evidencia para los estudiantes.

Recomendaciones

A partir de los resultados obtenidos en la evaluación de la probabilidad de abandono, recomendamos aplicar intervenciones adaptadas a los estudiantes que hayamos identificado bajo un umbral del riesgo (en categoría alto y medio de riesgo). Por ejemplo, el asesoramiento académico a medida, la tutoría o el seguimiento/especialización de apoyo psicológico pueden resultar vitales para minimizar el riesgo de abandonar de manera prematura la materia y a la vez para mejorar su rendimiento.

El modelo Random Forest que utilizamos mostró un buen rendimiento (con un F1 Score del 76%); no obstante, sugerimos investigar en el futuro el uso de otros algoritmos de clasificación. Probar con redes neuronales o Gradient Boosting Machines (GBM) nos podría permitir comprobar si existe la posibilidad de incrementar aún más la precisión de nuestras predicciones. También sería beneficioso comparar la eficacia de modelos híbridos, que combinan diferentes enfoques, para mejorar el proceso predictivo.

Dado que descubrimos que combinar diferentes tipos de características mejora la predicción, sería conveniente investigar en futuros trabajos el uso de variables adicionales que podrían influir en el éxito académico. Considerar elementos como el comportamiento del estudiante en plataformas en línea, su participación en actividades extracurriculares o su interacción en foros de debate podría darnos una imagen más completa de su motivación y rendimiento.

Para examinar más a fondo los factores que influyen en el abandono escolar, recomendamos realizar estudios más amplios y multifacéticos que tengan en cuenta elementos como los rasgos psicológicos de los estudiantes, su contexto socioeconómico y las características del entorno de aprendizaje.

Este estudio se centró en alumnos de primer año de Fundamentos de Programación. Sería pertinente ampliar la investigación para incluir a estudiantes de años posteriores (segundo o tercero) y así examinar cómo cambian con el tiempo los factores que influyen en el abandono. Igualmente, nos parece útil la posibilidad de volver a aplicar este estudio en otras instituciones o en otros sitios, de manera que examinemos si los modelos y métodos

utilizados son generalizables a distintas realidades. Si se pudiera implementar este enfoque predictivo a más cursos y programas académicos, su impacto y utilidad podrían ampliarse considerablemente.

Finalmente, se recomienda investigar en futuras investigaciones cómo el ajuste del proceso de aprendizaje a las necesidades y características propias de cada estudiante (posiblemente utilizando la información de modelos predictivos) incrementa el rendimiento educativo y desciende las tasas de abandono.

Referencias

Abdulazeez, Y., & Abdulwahab, L. (2019). Application of classification models to predict students' academic performance using classifiers ensemble and synthetic minority over sampling techniques. *Bayero Journal of Pure and Applied Sciences*, *11*(2), Article 2. https://doi.org/10.4314/bajopas.v11i2.17

Alboaneen, D., Almelihi, M., Alsubaie, R., Alghamdi, R., Alshehri, L., & Alharthi, R. (2022). Development of a Web-Based Prediction System for Students' Academic Performance. *Data*, *7*(2), Article 2. https://doi.org/10.3390/data7020021

Alfere, S. S., & Maghari, A. Y. (2018). *Prediction of Student's Performance Using Modified KNN Classifiers*.

Alonso, M. A., González-Ortiz-de-Zárate, A., Gómez-Flechoso, M. Á., & Castrillón, M. (2024). Effectiveness of a Peer Mentoring on University Dropout and Academic Performance. *Psicología Educativa*, *30*(1), 29-37. https://doi.org/10.5093/psed2024a5

Alsulami, A. A., AL-Ghamdi, A. S. A.-M., & Ragab, M. (2023). Enhancement of E-Learning Student's Performance Based on Ensemble Techniques. *Electronics*, *12*(6), Article 6. https://doi.org/10.3390/electronics12061508

Anh, B. N., Giang, N. H., Hai, N. Q., Minh, T. N., Son, N. T., & Chien, B. D. (2023). An University Student Dropout Detector Based on Academic Data. *2023 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, 1-8. https://doi.org/10.1109/ISIEA58478.2023.10212223

Aydin, Z. E., & Ozturk, Z. K. (s. f.). *Performance Analysis of XGBoost Classifier with Missing Data*.

Balachandar, V., & Venkatesh, K. (2023). Predicting and Analysing University Dropout Rates using Machine Learning Methods. *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, 1-8. https://doi.org/10.1109/ICSES60034.2023.10465449

Burgos, C., Campanario, M. L., Peña, D. D. L., Lara, J. A., Lizcano, D., & Martínez, M. A. (2018). Data mining for modeling students' performance: A tutoring action plan to prevent academic dropout. *Computers & Electrical Engineering*, *66*, 541-556. https://doi.org/10.1016/j.compeleceng.2017.03.005

Castro, J. A. Q., Mosquera, J. M. L., López, Á. H. A., & Suarez, S. B. (2019). *Desarrollo de una aplicación web de seguimiento y deserción para los estudiantes de la corporación universitaria del Huila – CORHUILA*.

Chen, F., & Cui, Y. (2020). Utilizing Student Time Series Behaviour in Learning Management Systems for Early Prediction of Course Performance. *Journal of Learning Analytics*, *7*(2), Article 2. https://doi.org/10.18608/jla.2020.72.1

Chen, S., Webb, G. I., Liu, L., & Ma, X. (2020). A novel selective naïve Bayes algorithm. *Knowledge-Based Systems*, *192*, 105361. https://doi.org/10.1016/j.knosys.2019.105361

Cunningham, P., & Delany, S. J. (2022). k-Nearest Neighbour Classifiers—A Tutorial. *ACM Computing Surveys*, *54*(6), 1-25. https://doi.org/10.1145/3459665

Fernández, P., Vallejo, G., Livacic-Rojas, P., & Tuero, E. (2014). Validez Estructurada para una investigación cuasi-experimental de calidad. Se cumplen 50 años de la presentación en sociedad de los diseños cuasi-experimentales. *Anales de Psicología*, *30*(2), 756-771. https://doi.org/10.6018/analesps.30.2.166911

Gobernación del Huila. (2020). *Plan de desarrollo 2020-2023*. Gobernación del Huila. https://www.huila.gov.co/documentos/1336/plan-de-desarrollo-2020-2023/

González Rojas, K. T. (2021). *Construcción de un modelo para predecir el rendimiento académico de los estudiantes de ingeniería electrónica de la universidad distrital Francisco José de Caldas mediante algoritmos de redes neuronales con aprendizaje automático*.

Guzmán-Castillo, S., Körner, F., Pantoja-García, J. I., Nieto-Ramos, L., Gómez-Charris, Y., Castro-Sarmiento, A., & Romero-Conrado, A. R. (2022). Implementation of a Predictive Information System for University Dropout Prevention. *Procedia Computer Science*, *198*, 566-571. https://doi.org/10.1016/j.procs.2021.12.287

Hidalgo Suarez, C. G., Llanos, J., & Bucheli, V. A. (2023). Predicting the final grade using a machine learning regression model: Insights from fifty percent of total course grades in CS1 courses. *PeerJ Computer Science*, *9*, e1689. https://doi.org/10.7717/peerj-cs.1689

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2016). *A Practical Guide to Support Vector Classification*.

IBM. (2021, agosto 17). *SPSS Modeler Subscription*. https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview

Ioannou, C., & Vassiliou, V. (2021). Network Attack Classification in IoT Using Support Vector Machines. *Journal of Sensor and Actuator Networks*, *10*(3), 58. https://doi.org/10.3390/jsan10030058

Ismail, M., Hassan, N., & Bafjaish, S. S. (2020). *Comparative Analysis of Naive Bayesian Techniques in Health-Related for Classification Task*. *1*(2).

Izza, Y., Ignatiev, A., & Marques-Silva, J. (2020). *On Explaining Decision Trees* (arXiv:2010.11034). arXiv. https://doi.org/10.48550/arXiv.2010.11034

Jamjoom, M., Alabdulkreem, E., Hadjouni, M., Karim, F., & Qarh, M. (2021). Early Prediction for At-Risk Students in an Introductory Programming Course Based on Student Self-Efficacy. *Informatica*, *45*(6), Article 6. https://doi.org/10.31449/inf.v45i6.3528

Jokhan, A., Sharma, B., & Singh, S. (2019). Early warning system as a predictor for student performance in higher education blended courses. *Studies in Higher Education*, *44*(11), Article 11. https://doi.org/10.1080/03075079.2018.1466872

Kaunang, F. J., & Rotikan, R. (2018). Students' Academic Performance Prediction using Data Mining. *2018 Third International Conference on Informatics and Computing (ICIC)*, 1-5. https://doi.org/10.1109/IAC.2018.8780547

Kocsis, Á., & Molnár, G. (2024). Factors influencing academic performance and dropout rates in higher education. *Oxford Review of Education*, 1-19. https://doi.org/10.1080/03054985.2024.2316616

Kruse, R., Mostaghim, S., Borgelt, C., Braune, C., & Steinbrecher, M. (2022). *Computational Intelligence: A Methodological Introduction*. Springer International Publishing. https://doi.org/10.1007/978-3-030-42227-1

Kurniadi, F. I., Dewi, M. A., Murad, D. F., Rabiha, S. G., & Romli, A. (2023). An Investigation into Student Performance Prediction using Regularized Logistic Regression. *2023 IEEE 9th International Conference on Computing, Engineering and Design (ICCED)*, 1-6. https://doi.org/10.1109/ICCED60214.2023.10425782

La República. (2023). *Preocupación del sector público, privado y académico por la falta de programadores*. https://www.larepublica.co/empresas/foro-transformando-vidas-a-traves-de-la-educacion-digital-talento-y-oportunidades-3746249

Lacave, C., Molina, A. I., & Cruz-Lemus, J. A. (2018). Learning Analytics to identify dropout factors of Computer Science studies through Bayesian networks. *Behaviour & Information Technology*, *37*(10-11), Article 10-11. https://doi.org/10.1080/0144929X.2018.1485053

Lakanen, A.-J., & Isomöttönen, V. (2023). CS1: Intrinsic Motivation, Self-Efficacy, and Effort. *Informatics in Education*. https://doi.org/10.15388/infedu.2023.26

Lázaro Alvarez, N., Callejas, Z., & Griol, D. (2020). Predicting Computer Engineering students' dropout in Cuban Higher Education with pre-enrollment and early performance data. *Journal of Technology and Science Education*, *10*(2), Article 2. https://doi.org/10.3926/jotse.922

Lazo Jaque, F. A. (2021). *Modelo predictor del rendimiento académico de los estudiantes de un curso de programación de primer año de la Universidad Andrés Bello*. https://repositorio.unab.cl/xmlui/handle/ria/22174

Llanos, J., Bucheli, V. A., & Restrepo-Calle, F. (2023). Early prediction of student performance in CS1 programming courses. *PeerJ Computer Science*, *9*, e1655. https://doi.org/10.7717/peerj-cs.1655

Lu, S., Wang, X., Zhou, H., Sun, Q., Rong, W., & Wu, J. (2021). Anomaly Detection for Early Warning in Object-oriented Programming Course. *2021 IEEE International Conference on Engineering, Technology & Education (TALE)*, 01-08. https://doi.org/10.1109/TALE52509.2021.9678677

Mai, T. T., Crane, M., & Bezbradica, M. (2023). Students' Learning Behaviour in Programming Education Analysis: Insights from Entropy and Community Detection. *Entropy*, *25*(8), Article 8. https://doi.org/10.3390/e25081225

Margulieux, L. E., Morrison, B. B., & Decker, A. (2020). Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. *International Journal of STEM Education*, *7*(1), 19. https://doi.org/10.1186/s40594-020-00222-7

Martins, M. V., Baptista, L., Machado, J., & Realinho, V. (2023). Multi-Class Phased Prediction of Academic Performance and Dropout in Higher Education. *Applied Sciences*, *13*(8), Article 8. https://doi.org/10.3390/app13084702

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental

Results. *2020 11th International Conference on Information and Communication Systems (ICICS)*, 243-248. https://doi.org/10.1109/ICICS49469.2020.239556

Montesinos López, O. A., Montesinos López, A., & Crossa, J. (2022). Overfitting, Model Tuning, and Evaluation of Prediction Performance. En O. A. Montesinos López, A. Montesinos López, & J. Crossa, *Multivariate Statistical Machine Learning Methods for Genomic Prediction* (pp. 109-139). Springer International Publishing. https://link.springer.com/10.1007/978-3-030-89010-0_4

Nabil, A., Seyam, M., & Abou-Elfetouh, A. (2021). Prediction of Students' Academic Performance Based on Courses' Grades Using Deep Neural Networks. *IEEE Access*, *9*, 140731-140746. https://doi.org/10.1109/ACCESS.2021.3119596

Parmar, A., Katariya, R., & Patel, V. (2019). A Review on Random Forest: An Ensemble Classifier. En J. Hemanth, X. Fernando, P. Lafata, & Z. Baig (Eds.), *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018* (Vol. 26, pp. 758-763). Springer International Publishing. https://doi.org/10.1007/978-3-030-03146-6_86

Periódico UNAL. (2022). *Periódico UNAL - Abandono universitario tarea en la que Iberoamérica se sigue rajando*. https://www.periodico.unal.edu.co/articulos/abandono-universitario-tarea-en-la-que-iberoamerica-se-sigue-rajando

Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. En *Machine Learning* (pp. 101-121). Elsevier. https://doi.org/10.1016/B978-0-12-815739-8.00006-7

Prasanth, A., & Alqahtani, H. (2023). Predictive Modeling of Student Behavior for Early Dropout Detection in Universities using Machine Learning Techniques. *2023 IEEE 8th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, 1-5. https://doi.org/10.1109/ICETAS59148.2023.10346531

Qadir, A. M., & Abd, D. F. (2023). Kidney Diseases Classification using Hybrid Transfer-Learning DenseNet201-Based and Random Forest Classifier. *Kurdistan Journal of Applied Research*, 131-144. https://doi.org/10.24017/Science.2022.2.11

Ramírez Echeverry, J. J., García Carrillo, À., & Olarte Dussan, F. A. (2016). Adaptation and validation of the motivated strategies for learning questionnaire-mslq-in engineering students in Colombia. *Tempus Publications*, *32*(4), 1774-1787.

Ramírez-Echeverry, J. J., García-Carrillo, A., & Olarte Dussán, F. A. (2016). Adaptation and Validation of the Motivated Strategies for Learning Questionnaire—MSLQ—in Engineering Students in Colombia. *2016*, *32*(4), 1-14. http://hdl.handle.net/2117/107554

Sánchez Turcios, R. A. (2015). *Prueba de Wilcoxon-Mann-Whitney: Mitos y realidades*.

Schefer-Wenzl, S., Miladinovic, I., Bachinger-Raithofer, S., & Muckenhumer, C. (2024). A Study on Reasons for Student Dropouts in a Computer Science Bachelor's Degree Program. En M. E. Auer, U. R. Cukierman, E. Vendrell Vidal, & E. Tovar Caro (Eds.), *Towards a Hybrid, Flexible and Socially Engaged Higher Education* (Vol. 911, pp. 391-400). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-53382-2_38

SPADIES. (2021). *ESTADÍSTICAS DE DESERCIÓN Y PERMANENCIA EN EDUCACIÓN SUPERIOR SPADIES 3.0*. SPADIES. http://www.mineducacion.gov.co/sistemasdeinformacion/1783/w3-propertyvalue-68157.html

Sunday, K., Ocheja, P., Hussain, S., Oyelere, S. S., Samson, B. O., & Agbo, F. J. (2020). Analyzing Student Performance in Programming Education Using Classification Techniques. *International Journal of Emerging Technologies in Learning (iJET)*, *15*(02), Article 02. https://doi.org/10.3991/ijet.v15i02.11527

Turcios, R. A. S. (2015). *Prueba de Wilcoxon-Mann-Whitney: Mitos y realidades*.

Uddin, S., Haque, I., Lu, H., Moni, M. A., & Gide, E. (2022). Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. *Scientific Reports*, *12*(1), 6256. https://doi.org/10.1038/s41598-022-10358-x

Uhanova, M., Prokofyeva, N., Katalnikova, S., Zavjalova, O., & Ziborova, V. (2023). The Influence of Prior Knowledge and Additional Courses on the Academic Performance of Students in the Introductory Programming Course CS1. *Procedia Computer Science*, *225*, 1397-1406. https://doi.org/10.1016/j.procs.2023.10.128

Verma, S., Yadav, R. K., & Kholiya, K. (2022). Prediction of Academic Performance of Engineering Students by Using Data Mining Techniques. *International Journal of Information and Education Technology*, *12*(11), Article 11. https://doi.org/10.18178/ijiet.2022.12.11.1734

Vives, L., Cabezas, I., Vives, J. C., Reyes, N. G., Aquino, J., Cóndor, J. B., & Altamirano, S. F. S. (2024). Prediction of Students' Academic Performance in the Programming Fundamentals Course Using Long Short-Term Memory Neural Networks. *IEEE Access*, *12*, 5882-5898. https://doi.org/10.1109/ACCESS.2024.3350169

Yao, X., Fu, X., & Zong, C. (2022). Short-Term Load Forecasting Method Based on Feature Preference Strategy and LightGBM-XGboost. *IEEE Access*, *10*, 75257-75268. https://doi.org/10.1109/ACCESS.2022.3192011