Revista Internacional de Investigación e Innovación Tecnológica

Página principal: www.riiit.com.mx

Predicción del bajo rendimiento académico en programación de computadoras mediante técnicas de minería de datos educativa en muestras reducidas

Educational Data Mining Techniques for Predicting Low Academic Performance in Computer Programming with Small Samples Sized

Enríquez-Ramírez, C.*, Gordillo-Benavente, L. de J.
Universidad Politécnica de Tulancingo.
carlos.enriquez@upt.edu.mx*; liliana.gordillo@upt.edu.mx

Innovación tecnológica: Se presenta un modelo para detectar el bajo rendimiento académico.
Área de aplicación industrial: En el área de educación, específicamente en las actividades tutoriales en Universidades.

Recibido: 13 diciembre 2024
Aceptado: 03 julio 2025

Abstract

The objective of this research was to demonstrate the feasibility of a predictive model for poor academic performance applicable in contexts with limited data (n=94 first-terms students), integrating psychoeducational variables (personality traits and school climate) and specialized techniques (SMOTE + ReliefF + assemblies). Using Educational Data Mining (EDM) techniques, this study is presented as quantitative and non-experimental applying two Likert-scale instruments: one measuring motivation and another evaluating personality traits based on the Big Five model (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism). The dataset was processed using SMOTE (to balance minority classes) and ReliefF (for feature selection), reducing from 45 variables to 16 key predictors. Four base algorithms (Random Forest, XGBoost, Gradient Boosting, Extra Trees) and a voting ensemble classifier were evaluated. Cross-validation (k=10) and metrics including accuracy, precision, sensitivity, F-measure, and AUC were used to measure performance. However, the sensitivity (50%) revealed difficulties in detecting positive cases, likely due to class imbalance. These results demonstrate the effectiveness of the ensemble method techniques in predicting poor academic under performance, they provide educators with useful information to design early interventions and improve academic outcomes. The ensemble model achieved the highest accuracy (87%), precision (71%), and AUC (72%), outperforming other classifiers.

Keywords: Ensemble model, Random forest, XGBoost, Poor academic performance, Educational data mining.

Resumen

El objetivo de esta investigación fue demostrar la viabilidad de un modelo predictivo de bajo rendimiento académico aplicable en contextos con datos limitados (n=94 estudiantes en su primer ciclo académico), integrando variables psicoeducativas (rasgos de personalidad y clima escolar) y técnicas especializadas (SMOTE + ReliefF + ensambles). Mediante técnicas de minería de datos educativa (MDE), este estudio se presenta como cuantitativo y no experimental aplicándose dos instrumentos de escala Likert: uno midiendo la motivación y otro evaluando los rasgos de personalidad basado en el modelo de los Cinco Grandes (Apertura, Responsabilidad, Extraversión, Amabilidad y Neurotismo). El conjunto de datos se procesó utilizando SMOTE (para equilibrar clases minoritarias) y ReliefF (para selección de características), reduciendo de 45 variables a 16 predictores claves. Se evaluaron cuatro algoritmos base (Random Forest, XGBoost, Gradient Boosting, Extra Trees) y un clasificador de ensamble por votación. Se empleó validación cruzada (k=10) y métricas como exactitud, precisión, sensibilidad, F-measure y AUC para medir el rendimiento. Sin embargo, la sensibilidad (50%) evidenció dificultades para detectar casos positivos, probablemente debido al desbalance de clases. Estos resultados demuestran la eficacia de las técnicas del método de ensamble en la predicción del bajo rendimiento académico, proporcionando a los educadores información accionable para diseñar intervenciones tempranas y mejorar, el modelo de ensamble logró la mayor exactitud (87%), precisión (71%) y AUC (72%), superando los resultados académicos.

Palabra clave: Modelo de ensamble, *Random forest*, *XGBoost*, Bajo rendimiento académico, Minería de datos educativa.

1. Introducción

Este estudio tiene como objetivo principal demostrar la viabilidad técnica de un modelo predictivo de bajo rendimiento académico (BRA) para la materia de programación de computadoras basado en variables psicoeducativas (modelo de los Cinco Grandes + entorno escolar), aplicable en instituciones con acceso limitado a *big data* (n<100).

Para ello, se combinan técnicas de balanceo (SMOTE), selección de características (ReliefF) y modelos de ensamble, evaluando su capacidad para: (a) superar restricciones de muestras pequeñas y desbalanceadas, (b) identificar predictores no académicos y, (c) servir como base para alertas tempranas escalables. Esta aproximación aborda vacíos en la literatura [1, 2] al priorizar aplicabilidad en contextos reales sobre volumen de datos.

El aprendizaje de la programación computacional constituye un pilar fundamental en la formación de profesionales de tecnologías de la información, al equiparles con competencias clave para enfrentar las demandas del mercado laboral digital [3]. Sin embargo, esta asignatura presenta desafíos pedagógicos distintivos que elevan significativamente el riesgo del BRA.

Estudios recientes evidencian que la complejidad cognitiva inherente a la programación, que exige pensamiento abstracto, descomposición de problemas y lógica formal, genera una sobrecarga mental en estudiantes novatos, dificultando la asimilación de conceptos básicos [4]. A esto se suma una curva de aprendizaje no lineal:

errores aparentemente menores, como un punto y coma omitido invalidan soluciones completas, desencadenando frustración temprana y abandono de tareas [5]. Adicionalmente, la falta de retroalimentación inmediata en entornos sin herramientas de autoevaluación permite que malentendidos conceptuales se acumulen progresivamente [6, 7].

Otro desafío significativo es la prevalencia de la ansiedad de programación entre los estudiantes. La ansiedad de programación es particularmente problemática ya que puede crear un ciclo de autorrefuerzo donde ésta lleva a la evasión, lo que resulta en una reducción de la práctica y el desarrollo de habilidades, lo que exacerba aún más los niveles de la misma [8].

Estos factores explican por qué las tasas de BRA en programación son de 50% [9] consolidándose como un predictor crítico de deserción académica [3]. Por ello, la identificación temprana de estudiantes en riesgo resulta esencial para implementar intervenciones oportunas.

La incapacidad de alcanzar las calificaciones mínimas establecidas por la institución educativa se traduce en un indicador de riesgo académico [10]. Para mitigar este problema, es esencial que los docentes cuenten con herramientas que anticipen situaciones de riesgo, como el BRA, y permitan intervenir a tiempo en las dificultades de aprendizaje.

Las causas del BRA son multifactoriales y abarcan aspectos psicológicos (motivación, actitud), pedagógicas (estilos de aprendizaje), sociales, y del entorno académico. Entre los indicadores más estudiados destacan las calificaciones, las evaluaciones de aprendizaje [11] y la gestión de tiempo, principalmente en la educación superior, donde la autonomía del estudiante es crítica [12].

En este contexto, la minería de datos emerge como una herramienta en el ámbito educativo, ya que posibilita el análisis de grandes volúmenes de información generados por las instituciones, identificando patrones ocultos y predictores clave del rendimiento estudiantil [3]. Sin embargo, para transformar estos datos en hallazgos relevantes, se requieren técnicas que integren y analicen variables heterogéneas, permitiendo modelar predictivamente el riesgo académico [13].

Este estudio propone un modelo predictivo de BRA aplicado a la materia de programación de computadoras, basado en atributos individuales de los alumnos (motivación, actitud, estilos de aprendizaje y ambiente escolar). Mediante técnicas de minería de datos, se analizó una colección de información estudiantil para: Identificar patrones asociados al bajo rendimiento, comprender las causas específicas del BRA en programación de computadoras, una materia crítica en la carrera de Sistemas Computacionales de la Universidad Politécnica de Tulancingo (México), durante los primeros cuatrimestres.

La relevancia de este enfoque radica en la capacidad para combinar datos educativos tradicionales (como los resultados de las evaluaciones) con variables cualitativas, optimizando la toma de decisiones institucionales para reducir el abandono escolar [1], ver tabla 1.

En síntesis, los desafíos pedagógicos únicos de la programación como: complejidad cognitiva, curva de aprendizaje no lineal y ansiedad, unidos a tasas de BRA del 50%, exigen herramientas predictivas que trasciendan los indicadores académicos tradicionales.

El modelo aquí propuesto responde a esta necesidad al integrar variables psicoeducativas (rasgos de personalidad y

clima escolar) con técnicas especializadas (SMOTE, ReliefF y ensambles). Esta combinación no solo supera las limitaciones de muestras reducidas y desbalanceadas, sino que identifica predictores accionables, como la estabilidad emocional y la interacción docente, para intervenciones tempranas. Así, se ofrece una solución escalable a instituciones con restricciones de datos, priorizando la prevención del BRA y la deserción en programación.

Tabla 1. Organización categórica de factores educativos y psicosociales en programación.

Categoría	Subcategoría	Variables Específicas
1. Historial Académico	Rendimiento previo	- Promedio general de educación primaria - Promedio general de educación secundaria - Promedio general de educación media superior
2. Recursos y Acceso	Infraestructura digital	- Tienes internet en casa - Cuentas con computadora portátil - Cuentas con computadora de escritorio en casa
	Tiempo académico	- Tienes tiempo para practicar la programación de computadoras - Número de horas individuales de estudio
3. Experiencia Educativa	Calidad docente	- Dominio del maestro en clases - Existe habilidad del maestro en la materia - Existe método de enseñanza - Existe involucramiento del maestro
	Interacción educativa	- Te ayuda con dudas el profesor - Existe conexión estudiantes-profesor - El profesor te motiva para estudiar
	Compromiso estudiante	- Pones atención en clase - Estudias para las evaluaciones - Frecuencia de tareas de programación que realizas - Faltas a clase
	Ambiente de aprendizaje	- Medio ambiente de la clase - Dinamismo (contexto educativo) - Ausencia del maestro
4. Perfil Psicosocial	Emociones académicas	- Sufres de ansiedad en clase - Te sientes indefenso - Te sientes nervioso en evaluaciones - Te surge interés por la materia - Disfrutas tu clase de programación - Te sientes motivado por tus logros
	Rasgos de personalidad	- Control de emociones - Control de impulsos - Estabilidad emocional - Apertura mental/culturas/experiencia

Categoría	Subcategoría	Variables Específicas
		- Escrupulosidad
		- Persistencia
		- Afabilidad/Cordialidad
		- Energía
		- Responsabilidad
	Variables sociodemográficas	- Edad
		- Género

2. Trabajos relacionados

Para la construcción de modelos en la MDE, se utilizan diversas técnicas como la asociación, clasificación y la agrupación [14]. Estas permiten obtener patrones en el conjunto de datos que se analiza por medio de algoritmos como son: árboles de decisión, k-vecinos más cercanos (KNN), bayesiano ingenuo, inducción basada en reglas de asociación y muchas otras [15].

En la investigación de [16], se desarrolló un modelo predictivo para estimar las calificaciones finales de estudiantes en cursos introductorios de programación de computadoras. El objetivo del modelo era proporcionar retroalimentación temprana a los alumnos sobre su desempeño, permitiéndoles ajustar sus estrategias de estudio durante el semestre, se evaluaron 11 algoritmos de clasificación, entre los cuales el árbol de decisión CART demostró una mayor precisión (89%) y un F1-score de 0.86, superando a los métodos como SVM (Máquinas de Vector de Soporte) y redes neuronales.

Por su parte, [17] experimentó con diversos algoritmos de aprendizaje automático, regresión logística, redes neuronales, para identificar el mejor modelo predictivo de BRA. Estos enfoques destacan la utilidad de técnicas de clasificación supervisada para detectar patrones asociados al riesgo académico, sirviendo como base para diseñar intervenciones tempranas en contextos educativos.

Contrastando con lo anterior, [18] aplicaron algoritmos como Bayes ingenuo, SMO (Optimización Secuencial Mínima) y regresión logística para predecir el desempeño estudiantil en módulos de programación de computadoras. Durante la fase de preprocesamiento, se utilizó PCA (Análisis de Componentes Principales) para reducir la dimensionalidad de los datos. Los resultados identificaron a Bayes ingenuo como el predictor más efectivo según las métricas de precisión y F1-score usadas para validar el estudio.

En el estudio [19], se analizó 182 características de estudiantes universitarios en la India que cursaban programación en lenguaje C. Para optimizar el modelo, se aplicaron técnicas de selección de atributos, mediante Chi-cuadrado y ganancia de información, con el objetivo de reducir la dimensionalidad y eliminar variables irrelevantes. Entre los algoritmos evaluados (árboles de decisión, SVM, Bayes ingenuo), este último mostró mejor rendimiento (82% precisión, F1=0.78), destacando la importancia de combinar selección de características con modelos robustos.

En el estudio de [20], se diseñó un modelo predictivo de rendimiento académico analizando 324 variables y seleccionando las 15 más influyentes. Se implementaron

técnicas de ensamble (*Bagging*, *Boosting*, Votación) superaron a modelos individuales. En particular, *Stacking* logró una precisión del 85% en entrenamiento y 75% en prueba, mientras que *Blending* obtuvo valores similares (84% y 74%, respectivamente), demostrando una capacidad de generalización robusta. Estos hallazgos subrayan la efectividad de combinar múltiples algoritmos para mitigar el sobreajuste y mejorar la fiabilidad predictiva, un enfoque relevante para estudios centrados en educación.

En un estudio aplicado a estudiantes de Ciencias de la Computación de la Universiti Teknologi MARA [21], se evaluaron técnicas de minería de datos educativa para predecir deserción académica. Entre los algoritmos probados (árboles de decisión, bosques aleatorios, k-vecinos, perceptrón y regresión logística) este último obtuvo la mayor precisión (78%) y recall (72%), destacando en la identificación temprana de estudiantes en riesgo. Si bien este trabajo resalta la necesidad de predictores confiables para programas de retención, su enfoque difiere del presente estudio en tres aspectos clave:

- No emplea técnicas de ensamble, que en nuestra investigación demostraron mayor robustez al combinar múltiples algoritmos.
- No detalla variables específicas, mientras que nuestro modelo integra rasgos de personalidad (modelo de los Cinco Grandes) y factores del entorno escolar, enriqueciendo el análisis con perspectiva psicoeducativa.
- Se centra en deserción, mientras que nuestro objetivo es predecir el BRA en programación.
- Estas diferencias subrayan la contribución original de nuestro trabajo al ampliar el marco predictivo con variables cualitativas.

Un estudio reciente analizó 100 estudiantes de programación utilizando rasgos del modelo de los Cinco Grandes y datos académicos históricos, aplicando SMOTE con *Random Forest* para abordar el desbalance muestral (21% BRA) [22]. Reportaron 84% de precisión y 62% de sensibilidad, identificando la responsabilidad como predictor principal (β=0.72). Nuestra investigación amplía estos hallazgos mediante la incorporación de variables de clima educativo (ausencia docente y motivación del profesor), implementando un modelo de ensamble de votación que mejora la precisión (87%), y aplicando ReliefF para optimizar la interpretabilidad con solo 16 predictores.

Si bien su modelo logró mayor sensibilidad (62% contra 50%), nuestro enfoque demuestra que factores contextuales añaden valor predictivo en muestras reducidas, particularmente en entornos con limitaciones institucionales donde variables como la asistencia docente resultaron determinantes (ReliefF=0.053) [23].

Se confirma el uso de las técnicas de MDE en muestras reducidas como es el caso de [24]; [25]. En un contexto latinoamericano [26] comparable, aplicaron SMOTE con *Random Forest* a 120 estudiantes de programación, usando variables académicas históricas y autoeficacia. Reportaron 85% de precisión y AUC 0.76, identificando la autoeficacia como predictor dominante (importancia=0.81). A comparación en el estudio realizado se supera la precisión del modelo de 87% contra un 85% con un menor tamaño muestral (n=94 contra n=120), además de ofrecer mayor interpretabilidad mediante ReliefF (16 variables contra 25 en su modelo).

A diferencia de estudios previos centrados en variables académicas tradicionales (ej. calificaciones históricas), este trabajo integra rasgos de personalidad (modelo de los Cinco

Grandes) y variables cualitativas del entorno educativo (ej. interacción docente-estudiante, ausencia del profesor). Esta combinación permite identificar predictores únicos del BRA en la materia de programación de computadoras, como la estabilidad emocional y la motivación docente, factores críticos en asignaturas prácticas con alta demanda de tolerancia a la frustración.

Adicionalmente, la sinergia entre SMOTE, ReliefF y modelos de ensamble abordan limitaciones de muestras pequeñas y desbalanceadas, optimizando la generalización sin sacrificar interpretabilidad. Estas innovaciones amplían el alcance de la MDE al demostrar cómo variables no académicas y técnicas especializadas pueden mejorar intervenciones tempranas en contextos específicos.

3. Materiales y métodos

Este estudio adopta un diseño no experimental retrospectivo, bajo un enfoque cuantitativo-correlacional, fundamentándose en la reconstrucción de las causas del BRA sin manipulación de variables, analizando datos históricos de rendimiento académico y rasgos de personalidad lo que permite identificar patrones causales en su contexto natural [3, 10].

El modelo de KDD (*Knowledge Discovery Database*) se integra como herramienta técnica dentro del método científico, permitiendo una explotación sistemática de datos complejos [27]. Su enfoque iterativo (definición del problema, preprocesamiento, minería de datos e interpretación) garantiza la reproducción y la alineación con estándares de investigación en ciencias de la computación [12, 13].

3.1 Definición del problema

Se identificó un BRA en los estudiantes que cursan la materia de programación de computadoras en la institución educativa analizada. Ante esto, se propone identificar las características predictoras que permitan construir un modelo para predecir el BRA. Para ello, se utilizó un conjunto de datos obtenidos mediante encuestas que miden rasgos de personalidad (basado en el modelo de los Cinco Grandes [28]), y el ambiente escolar.

El propósito de este análisis es generar alertas tempranas sobre el desempeño académico de los participantes, con el fin de modificar actividades educativas, incrementar las calificaciones en la asignatura y reducir o erradicar el BRA en programación de computadoras. Para cumplir el objetivo de validar un modelo con datos limitados, se plantearon las siguientes hipótesis, fundamentadas en estudios con muestras reducidas [29, 30] y vacíos en predictores psicoeducativos [16, 19]:

H1: El uso combinado de SMOTE, ReliefF y modelos de ensamble logrará un AUC >70% en la predicción de BRA, pese al tamaño muestral (n=94) y un desbalance en las clases del 19%.

H2: Variables como la estabilidad emocional y la ausencia docente tendrán más peso predictivo (ReliefF >0.03) que los promedios académicos históricos para predecir el BRA en programación de computadoras.

H3: Este enfoque demostrará ser una prueba de concepto viable para universidades con restricciones de datos, permitiendo detectar estudiantes en riesgo con muestras pequeñas (n<100).

3.2 Participantes

Como evidencian [29, 30], la MDE es efectiva en muestras reducidas mediante técnicas como SMOTE y ensambles, sin requerir grandes volúmenes de datos. Estudios previos lo demuestran con 50 [1] y

120 [18] participantes. Nuestra investigación analizó 94 estudiantes de programación (edad media 20±5 años; 54 hombres, 40 mujeres).

3.3 Proceso

Se muestra la arquitectura lógica del estudio realizado, donde se pormenoriza el proceso de KDD usado (Figura 1) que se explica en las siguientes secciones.

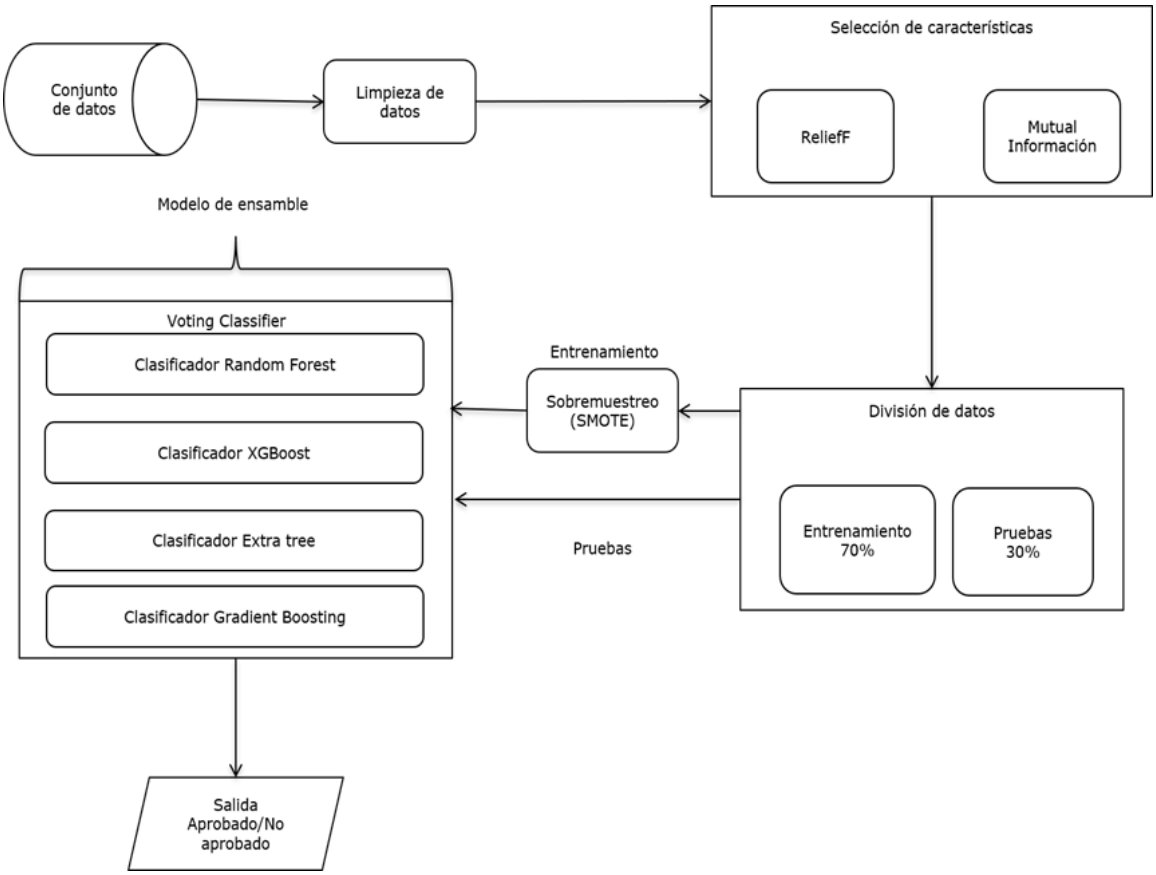

Figura 1. *Arquitectura lógica del proceso de KDD aplicado.*
Fuente: elaboración propia.

3.3.1 Conjunto de datos.

Para obtener la información se aplicaron cuestionarios tanto de clima educativo (tabla 1) donde se evaluó la percepción del alumno sobre su entorno de aprendizaje, incluyendo aspectos como recursos tecnológicos, interacción docente-estudiante y condiciones institucionales. Cuestionario de procesos de enseñanza donde se analizaron los métodos pedagógicos empleados por el docente y el cuestionario de los Cinco Grandes.

El modelo de los Cinco Grandes [28] se seleccionó como marco teórico por su solidez psicométrica, validez transcultural y amplia aceptación científica. Su diseño integral permite analizar rasgos de personalidad que influyen en la percepción del clima educativo y procesos didácticos. El cuestionario aplicado (130 reactivos, escala Likert 1-5) evalúa cinco dimensiones clave:

- Apertura (OPE): de cauteloso/consistente a curioso/inventivo. Refleja la tendencia de una persona a la

curiosidad intelectual, la creatividad y la preferencia por la novedad y variedad de experiencias. Una puntuación alta de apertura implica fuerte grados de imaginación, interés artístico, emoción, aventurero, intelecto y el liberalismo [28].

- Escrupulosidad (COS): de descuidado/tranquilo a organizado/eficiente. Representa un continuo que abarca desde comportamientos más relajados y menos estructurados hasta patrones altamente organizados y eficientes. Este rasgo psicológico refleja fundamentalmente la capacidad de autorregulación [28].
- Extraversión (EXT): de solitario/reservado a extrovertido/enérgico. Es un conjunto que va desde la preferencia por la soledad y la reserva hasta la expresión de energía y sociabilidad. - Este rasgo describe la inclinación de un individuo a interactuar con otros, mostrando comportamientos sociales y asertivos, así como mayor disposición a experimentar emociones positivas, como la alegría, satisfacción y entusiasmo [28].
- Amabilidad (AGR): de frío/cruel a amigable/compasivo este factor refleja la tendencia de una persona a ser amable, preocupada, veraz y cooperativa hacia otros. Una puntuación alta de amabilidad implica altos grados de moralidad, altruismo, simpatía, modestia, confianza, cooperación y conciliación [28].
- Neuroticismo (NEU): desde seguro/calmado hasta inseguro/nervioso: Mide qué tan propensa es una persona a sentir emociones negativas (ira, ansiedad, tristeza) y qué tan estable o vulnerable es emocionalmente. Quienes puntúan alto en este rasgo suelen ser más impulsivos, ansiosos y sensibles al estrés, mientras que quienes puntúan bajo tienden a mantenerse calmados y seguros en diversas situaciones [28].

Estos instrumentos se administraron al inicio del cuatrimestre, y las respuestas se transformaron en variables independientes. Para la variable dependiente se derivó del promedio final de tres evaluaciones parciales, siguiendo la normativa institucional (calificación mínima aprobatoria 7/10).

3.3.2 Limpieza de datos

Dentro de las fases de importancia en el proceso de MDE se encuentra el preprocesamiento que tiene como fase de acción la reducción de características, es decir, identificar aquellas variables que tienen o carecen de influencia en el uso de los modelos con la finalidad de minimizar costes de procesamiento y evitar la sobrecarga de recursos de software.

Tras el proceso de recolección inicial, se contó con 188 participantes y 45 variables independientes. Sin embargo, durante la fase de preprocesamiento, se identificaron 129 registros válidos (debido a la duplicidad o registros incompletos). De estos, se eliminaron 35 registros por las siguientes situaciones:

- Datos faltantes: 20 registros con respuestas incompletas.
- Inconsistencias: 15 registros con valores atípicos o contradicciones.

Finalmente, se trabajó con 94 registros depurados, asegurando la integridad y calidad del conjunto de datos para el análisis posterior.

3.3.3 Selección de características

La selección de características es un paso importante en la construcción de sistemas clasificadores, ya que permite identificar predictores relevantes y eliminar la redundancia o ruido en los datos. Este proceso persigue tres objetivos [31, 32]: en primer lugar, mejorar la precisión del modelo; en segundo lugar, minimizar los costos computacionales; y, por último, incrementar la interpretabilidad.

En este estudio, se implementó el algoritmo ReliefF, un método supervisado que asigna pesos a las características según su capacidad para discriminar entre clases [33]. A diferencia de métodos como la prueba Chi-cuadrada o el Análisis de Componentes Principales (PCA), ReliefF se adapta mejor a conjuntos de datos con relaciones no lineales y múltiples clases, se define en el pseudocódigo siguiente.

El proceso se implementa con el siguiente pseudocódigo:

Inicio: Asignamos peso de "0" a las 45 variables iniciales.

Repetir 100 veces (para garantizar resultados confiables):

Paso A: Seleccionar un estudiante al azar.

Paso B: Buscar sus 5 "vecinos académicos" más cercanos:

Grupo 1: Estudiantes con su mismo rendimiento (ej: ambos aprobados).

Grupo 2: Estudiantes con rendimiento opuesto (ej: aprobado vs no aprobado).

Paso C: Para cada variable (ej: "ausencia del profesor"):

Si varía mucho entre el estudiante y su Grupo 1: Restar importancia (no ayuda a distinguir rendimiento).

Si varía mucho entre el estudiante y su Grupo 2: Sumar importancia (sí ayuda a distinguir).

Resultado final: Ordenar todas las variables por su peso acumulado y seleccionar las 16 más importantes (con peso > 0.01).

El algoritmo ReliefF opera mediante 100 iteraciones donde en cada ciclo:
(1) se selecciona aleatoriamente un estudiante de referencia,
(2) se identifican sus 5 vecinos más cercanos con rendimiento similar ('hits') y 5 con rendimiento opuesto ('misses'),
(3) para cada variable, su peso se ajusta disminuyendo si presenta diferencias significativas con los 'hits' (indicando baja capacidad discriminativa dentro de la misma categoría) o aumentando si varía notablemente con los 'misses' (señalando relevancia predictiva). Finalmente, se seleccionan las 16 variables con peso acumulado > 0.01 que demostraron mayor poder para diferenciar entre estudiantes con alto y bajo rendimiento.

El algoritmo evalúa cada variable comparando estudiantes similares entre sí. Es decir, si dos estudiantes con rendimiento similar tienen valores distintos en una variable (ej: estabilidad emocional), esa variable probablemente no es relevante. Pero si dos estudiantes con rendimiento diferente tienen valores distintos, la variable podría ser importante.

3.3.4 División de datos.

Para garantizar la validez del modelo predictivo, el conjunto de datos se dividió en dos subconjuntos estratificados en primer lugar, de entrenamiento (70%), utilizado para ajustar los parámetros de los algoritmos y el

conjunto de prueba empleado para evaluar el rendimiento del modelo en datos no vistos. La estratificación se realizó conservando la proporción de clases (aprobado/no aprobado) en ambos subconjuntos, evitando sesgos en la evaluación. Esta estrategia, respaldada por práctica estándar en aprendizaje automático [33, 34], asegura que las distribuciones de las variables independientes sean consistentes entre entrenamiento y prueba.

Se consideró una muestra de 94 estudiantes, aplicando técnicas de minería de datos educativa (MDE) adaptadas a contextos con limitaciones muestrales. Si bien el volumen de datos no alcanza escalas masivas, el preprocesamiento con SMOTE (para balancear clases minoritarias) y ReliefF (para selección de características) permitió extraer patrones predictivos robustos, optimizando la calidad del modelo. Enfocándose en la aplicabilidad de técnicas de MDE como prueba de concepto en entornos educativos específicos, donde la recolección de grandes volúmenes de datos suele ser un desafío operativo.

3.3.5 Submuestreo

Dado que el conjunto de datos estaba desbalanceado en un 19% de registros pertenecientes a la clase “No aprobados” se aplicó el algoritmo SMOTE (*Synthetic Minority Oversampling Technique*). Este método genera instancias sintéticas de la clase minoritaria con el objetivo de equilibrar el conjunto de datos y mejorar el rendimiento del modelo clasificatorio [35].

3.3.6 Modelo de ensamble

La combinación de algoritmos mediante técnicas de ensamblado permite integrar fortalezas predictivas de múltiples modelos base, modelo combinado. Este enfoque, fundamentado en el principio de diversidad algorítmica [35], produce sistemas más robustos al:

(1) reducir la varianza del error mediante promediado de predicciones (*bagging*),
(2) optimizar secuencialmente el sesgo (*boosting*), y
(3) maximizar la precisión global mediante mecanismos de votación ponderada [36].

Como resultado, se mitiga el riesgo de sobreajuste (*overfitting*) y se mejora significativamente la estabilidad predictiva en conjunto de datos complejos. Por ejemplo, los ensambles funcionan como un comité de expertos: *Random Forest*, *XGBoost* y otros algoritmos debaten sus predicciones, y la decisión final (votación) es más precisa que la de un solo experto.

3.3.6.1 Arquitectura del clasificador

Dentro de las técnicas de minería de datos, usadas para construir el modelo de ensamble para este trabajo de investigación se tomaron a los siguientes:

- *Random Forest*, es una técnica conocida de aprendizaje automático supervisado, se basa en la idea de aprendizaje conjunto. Agrega diferentes números de árboles de decisión en diferentes subconjuntos del conjunto de datos proporcionado y luego promedia los resultados para mejorar la precisión prevista del conjunto de datos [16].
- *Extreme Gradient Boosting* (*XGBoost*), es una estrategia de aprendizaje supervisado basado en árboles de decisión, es considerado una evolución de los algoritmos como árboles de decisión, *bagging*, *random forest*, *boosting*, *gradient boostring*. Es destacable por su velocidad de proceso y su precisión de predicción en conjuntos grandes de datos, además de minimizar el error de sesgo del modelo [16].

- *Extratree*. Es un método de aprendizaje automático supervisado. Sus dos principales diferencias con otros métodos de conjuntos basados en árboles son que divide los nodos eligiendo puntos de corte completamente al azar y que utiliza toda la muestra de aprendizaje (en lugar de una réplica) para hacer crecer los árboles. En años recientes, *Extratree* ha sido ampliamente adoptado en aplicaciones prácticas debido a su eficiencia computacional y su capacidad para manejar datos de alta dimensionalidad, además de servir como componente base en enfoques de ensamblado modernos [36].
- *Gradient Boosting*. Es un método que se destaca por la velocidad y precisión de predicción en conjuntos de datos grandes. Dentro de las características es el manejo de minimizar el error de sesgo e incrementar la varianza al aumentar la complejidad de los predictores débiles [36].
- *Voting Classifier* permite llevar a cabo modelos de ensamble en el aprendizaje automático, mediante la combinación de diversos modelos base para tomar una decisión. Admite dos tipos de votaciones fuerte y suave para el primer caso cada modelo base en el ensamblado emite una predicción y la clase que recibe la mayoría de votos se selecciona como la predicción final del ensamblado, por otra parte, se realiza el conteo de votos se toma en cuenta las probabilidades de cada una de las clases promediando las probabilidades y la clase con mayor probabilidad es la que es seleccionada [16].

3.4 Tratamiento de la información

Se implementó validación cruzada *k-fold* (k=10) con *scikit-learn* para construir y evaluar los predictores. Esta técnica divide los datos en 10 particiones, garantizando que cada instancia se use tanto para entrenamiento como validación. El enfoque optimiza el equilibrio sesgo-varianza, mejora la generalización del modelo y previene sobreajuste en datos limitados [38].

Para medir la exactitud de los modelos generados por técnicas de clasificación, es necesario ejecutar el algoritmo y obtener las predicciones de clase. Posteriormente, estas predicciones se contrastan con los valores reales de los datos, permitiendo calcular tanto la precisión del modelo como los errores de clasificación. Mediante, el *F-measure* Ec. (1), la Precisión Ec. (2) y el *recall* (Sensibilidad) Ec. (3), como se describen a continuación:

$$F - measure = 2 \times \left(\frac{Precision \times recall}{Precision + recall}\right) \quad \textbf{(Ec. 1)}$$

$$Precision = \frac{VP}{(FP + VP)} \quad \textbf{(Ec. 2)}$$

$$recall = \frac{VP}{(FN + VP)} \quad \textbf{(Ec. 3)}$$

Verdadero Positivo (VP): Instancias positivas y clasificadas como positivas.
Falso Positivo (FP): Instancias negativas, pero clasificadas como positivas.
Falso Negativo (FN): Instancias positivas, pero clasificadas como negativas.

La elección de las métricas de precisión, *recall* y *F-measure* se fundamenta en la capacidad para abordar los desafíos específicos de clasificación en contextos educativos con clases desbalanceadas, como el caso de BRA, esta decisión se basa en

recomendaciones metodológicos de la literatura [34, 35, 36].

4. Resultados y hallazgos

Los resultados obtenidos demuestran la eficacia del modelo propuesto para predecir el BRA en la asignatura de programación de computadoras. En esta sección, se presentan las métricas de evaluación y los hallazgos clave derivados del análisis comparativo entre los algoritmos base y de ensamble. Para cada técnica evaluada se generaron modelos predictivos orientados a clasificar a los estudiantes en dos estados: En riesgo académico (clase positiva) y sin riesgo académico (clase negativa).

4.1 Desempeño comparativo de modelos

Como se buscaba en el objetivo central, el enfoque propuesto demostró una viabilidad técnica con los datos limitados (94 alumnos) en la H1, el modelo de ensamble demostró superioridad frente a los algoritmos base, alcanzando un 87% de exactitud y un AUC del 72% (Figura 2). Esto valida que la combinación de múltiples clasificadores mitiga el sobreajuste y mejora la generalización en datos desbalanceados, tal como predice la teoría de ensambles [36].

Estos resultados validan la hipótesis de que la diversidad algorítmica mejora la predicción del BRA en contextos educativos complejos.

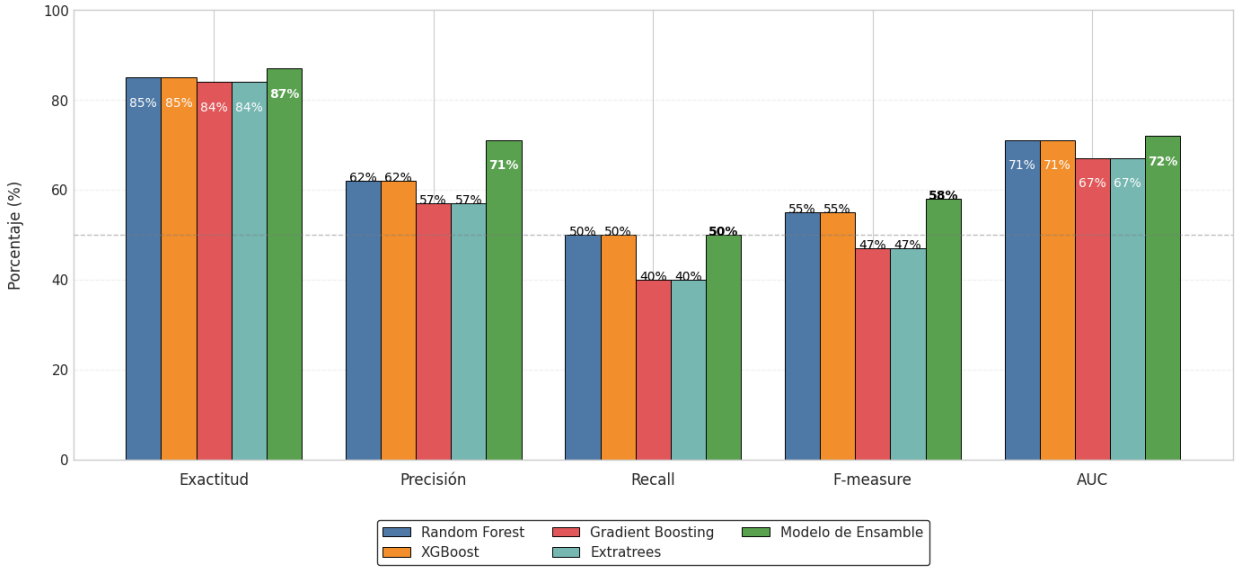

Figura 2. *Comparación de modelos predictivos por métricas.*
Fuente: elaboración propia.

No obstante, el recall del 50% evidenció una limitación inherente a la naturaleza desbalanceada del conjunto de datos (sólo el 19% de la muestra correspondió a la clase de riesgo), donde el modelo priorizó reducir (FP) sin detectar a la mitad de los casos críticos (FN). Para contextos educativos, esto indica que, si bien el sistema es confiable para identificar estudiantes que realmente necesitan el apoyo, requiere ser completado con evaluaciones cualitativas periódicamente para mitigar los falsos positivos [20, 35].

4.2 Implicaciones prácticas.

Con un 71% de precisión, el modelo de ensamble constituye una herramienta confiable para detectar estudiantes en riesgo de BRA desde etapas iniciales del curso. Esto

permite implementar intervenciones personalizadas (tutorías adicionales o ajustes metodológicos) antes que los problemas académicos se agraven.

Su integración en plataformas de gestión académica podría generar alertas automáticas para facilitar decisiones en tiempo real, alineándose con las tendencias actuales en *learning analytics* y educación basada en datos. La Figura 3 ilustra cómo los resultados individuales del modelo guían a los educadores en su aplicación práctica.

Si un estudiante es clasificado en 'riesgo' (clase positiva), identificar las variables con mayor peso (Figura 4) y active intervenciones especificas dependiendo del plan de acción personalizado. Ejemplo alta ausencia docente (0.0534) sugiere coordinar tutorías de recuperación.

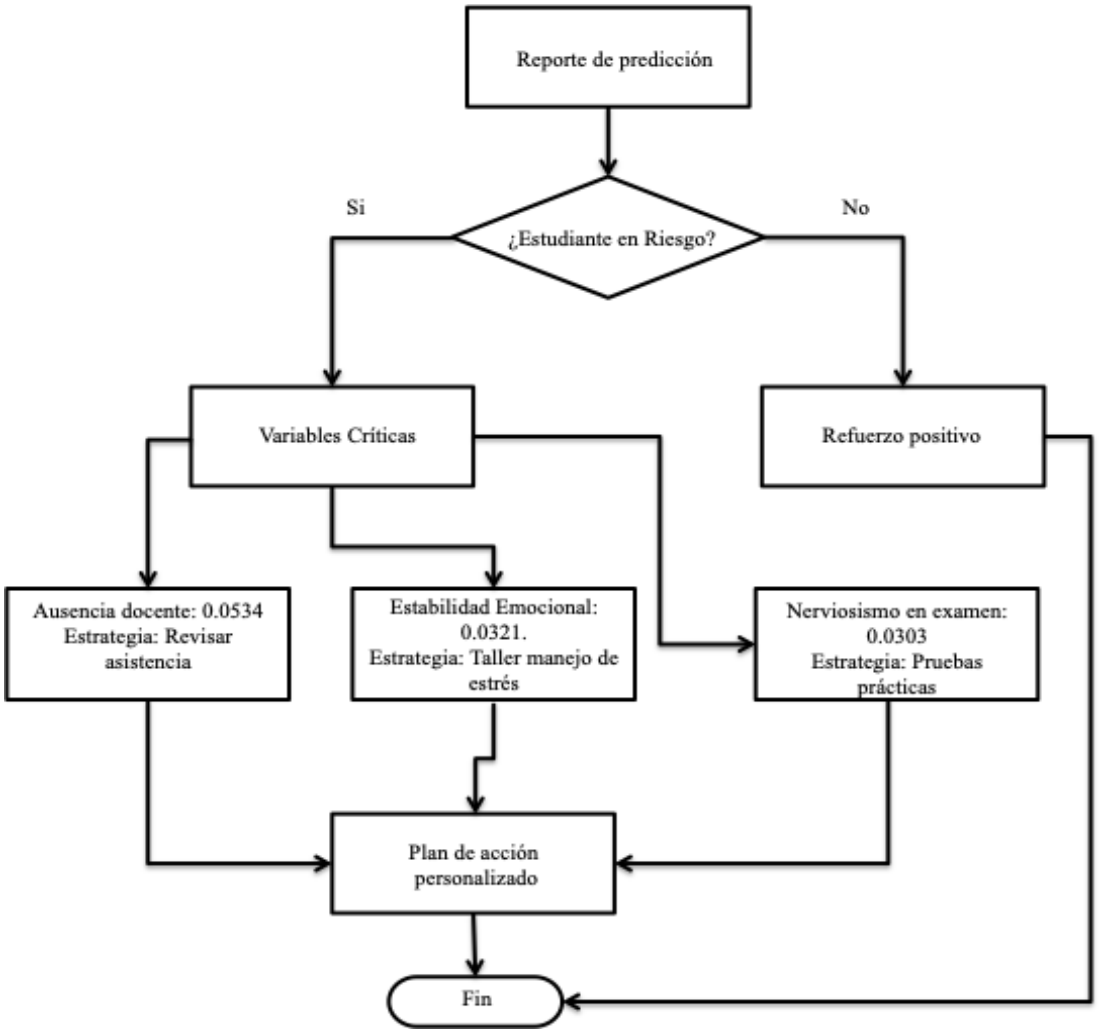

Figura 3. Proceso para la interpretación de resultados.
Fuente: elaboración propia.

Si el modelo alerta que un estudiante tiene riesgo de BRA (probabilidad: 85%), el docente revisa sus variables clave: baja persistencia (0.041) y alta ansiedad en exámenes (0.038). Esto sugiere intervenciones como tutorías en manejo de estrés y ejercicios progresivos de programación.

4.3 Variables identificadas

El proceso de selección de características mediante el algoritmo de ReliefF permitió identificar las variables más relevantes para predecir el BRA en la asignatura de programación de computadoras Figura 4. Este refinamiento buscó optimizar la capacidad explicativa del modelo manteniendo sólo las características más informativas.

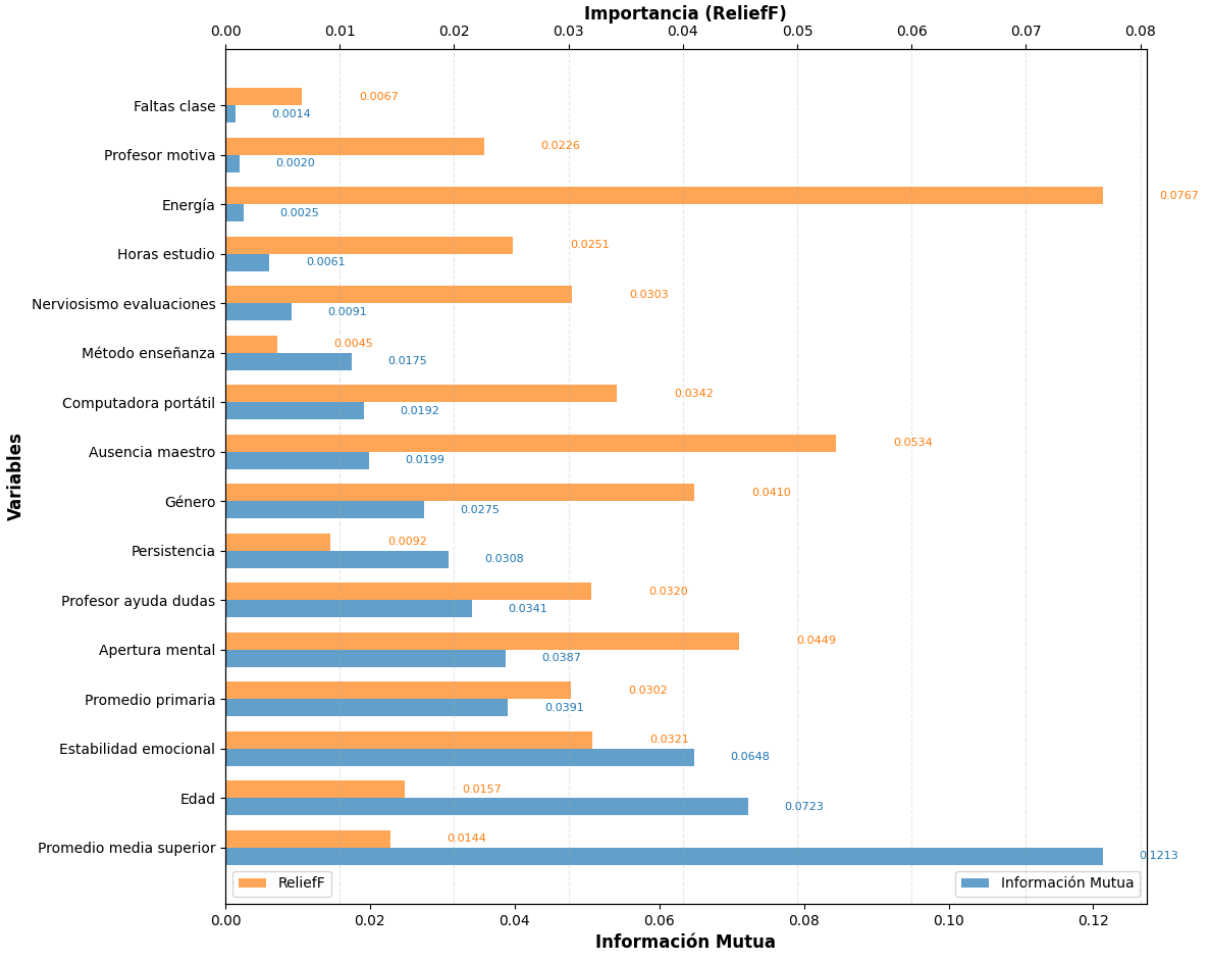

Figura 4. Características obtenidas por el algoritmo ReliefF (Valores más altos = Mayor impacto en el rendimiento. Ej: 'Ausencia docente' (0.053) afecta 3.7 veces más que 'Horas de estudio' (0.014)).
Fuente: elaboración propia.

La elección de las 16 variables predictoras mediante el algoritmo ReliefF se fundamentó en su capacidad para discriminar entre clases en contextos desbalanceados y relaciones no lineales, priorizando atributos con mayor peso predictivo (p. ej., "Ausencia del maestro" y "Energía").

Este modelo supera los enfoques tradicionales centrados en variables académicas convencionales (calificaciones históricas, horas de estudio) al incorporar dimensiones psicoeducativas clave. Dimensiones como la estabilidad emocional (Neuroticismo) y la apertura mental demuestran impacto directo en la tolerancia a la frustración durante la codificación. Simultáneamente, variables del entorno docente (motivación del profesor, asistencia) revelan la influencia del clima educativo, factor subestimado en estudios previos [1, 30].

La integración de rasgos de personalidad y factores contextuales proporciona una perspectiva holística del bajo rendimiento académico (BRA). Esta aproximación multifactorial no solo mejora la capacidad predictiva, sino que también permite diseñar intervenciones basadas en el bienestar emocional y la calidad docente, aspectos esenciales en entornos educativos prácticos.

Así, el modelo genera hallazgos accionables para abordar las causas complejas del BRA, alineándose con avances recientes en psicología educativa [33]. Su valor diferencial radica en trascender las métricas individuales para ofrecer soluciones integradas que responden a la naturaleza multidimensional del aprendizaje en programación.

En línea con la H2, el algoritmo ReliefF identificó que los rasgos de personalidad (estabilidad emocional: 0.0321) y la ausencia del profesor (0.0534) fueron los predictores más influyentes del BRA (Figura 4). Estos hallazgos respaldan la premisa de que factores psicoeducativos, más que académicos históricos, determinan el riesgo en asignaturas prácticas como programación.

Para el caso de la presencia de una alta Información Mutua, como el promedio de educación media (0.1213), se observó con un bajo peso en ReliefF (0.0144), lo que sugiere que, aunque estadísticamente relacionadas con el rendimiento, su capacidad discriminativa para predecir el BRA es limitado [33].

Las variables como “Ausencia del maestro en la clase de programación de computadoras” y “el profesor te motiva para estudiar la materia de programación de computadoras” (0.0534 y 0.0226, respectivamente) destacan el impacto del componente docente en el rendimiento, lo cual coincide con los hallazgos previos sobre la influencia del clima educativo [11, 31]. En cuanto a la selección de las variables, el algoritmo ReliefF [33] demostró ser efectivo al identificar 16 variables relevantes, descartando 29 variables poco informativas y optimizando así el rendimiento computacional del modelo sin perder capacidad predictiva [34].

4.4 Discusión de la investigación

El proceso seguido en este estudio consistió en aplicar técnicas de análisis de información mediante diversas técnicas de minería de datos aplicadas a una pequeña muestra para el logro del objetivo, validar un modelo viable de datos limitados, se refleja en los hallazgos de las primeras dos hipótesis: primero, las métricas competitivas (AUC: 72%) con datos limitados [H1], comparable con muestras mayores [1, 18], la identificación de predictores psicoeducativos [H2], ignorados en modelos tradicionales. Lo que convierte al modelo en una prueba de concepto usable [H3] para universidades con recursos limitados, donde la recolección masiva de los datos es inviable.

Los resultados obtenidos se alinean con hallazgos previos en minería de datos educativa (MDE), pero también introducen matices relevantes. Por ejemplo, el estudio de [1] identificó que los árboles de decisión (CART) alcanzaron una precisión del 89% para predecir el rendimiento en programación, similar a nuestro modelo de ensamble (87%). Sin embargo, a diferencia de [14], nuestro enfoque incorpora variables psicoeducativas (rasgos de personalidad y clima escolar), lo que amplía el marco explicativo del BRA más allá de los datos académicos tradicionales. Esto respalda la premisa de [3] sobre la necesidad de integrar dimensiones cualitativas en modelos predictivos educativos.

En cuanto a las técnicas, nuestro uso de SMOTE y ReliefF coincide con las recomendaciones de [15] para manejar datos desbalanceados y seleccionar características en contextos educativos. No obstante, mientras [18] reportó que el algoritmo Bayes ingenuo era el más efectivo para predecir desempeño en programación (precisión: 82%), nuestro trabajo demuestra que los ensambles (*Random Forest* + *XGBoost*) superan esta precisión (87%), validando las ventajas de la diversidad algorítmica señaladas por [20]. Esto sugiere que, para problemas multifactoriales como el BRA, los modelos híbridos pueden ofrecer mayor robustez.

La precisión de los resultados obtenidos en los cuatro algoritmos base, así como en el modelo de ensamble, se debe en gran medida a la proporción entre las clases de estudiantes aprobados y no aprobados. Este desequilibrio entre clases afecta la capacidad de los modelos para predecir con precisión, lo cual representa un factor clave en la interpretación de las métricas obtenidas.

4.4.1 Limitación de la investigación

Este estudio presenta algunas limitaciones. En primer lugar, el desequilibrio en la distribución de clases (19% BRA) afectó la capacidad del modelo para identificar casos positivos (recall: 50%), a pesar del uso de SMOTE. Además, la muestra reducida (n=94) y el contexto institucional específico limitan la generalización de los resultados.

Una limitación compartida con [13] es el bajo *recall* (50%) en clases minoritarias, atribuible al desbalance de datos. Futuras investigaciones podrían explorar técnicas de aprendizaje sensible al costo [35] o enfoques híbridos como los propuestos por [20], combinando ensambles con análisis cualitativo para mejorar la detección de casos positivos.

5. Conclusiones

Este estudio demuestra cómo los principios teóricos de la MDE y los modelos de predicción del rendimiento académico pueden aplicarse para abordar el BRA en la asignatura de programación de computadoras. Los resultados confirman que el modelo de los Cinco Grandes de personalidad, particularmente los factores de estabilidad emocional y apertura a la experiencia son predictores significativos en el modelo propuesto (pesos ReliefF: 0.0321 y 0.0449, respectivamente).

Además, la relevancia de variables del entorno educativo (ausencia del profesor, peso: 0.0534) corrobora los resultados de [10] sobre el impacto del clima escolar en el rendimiento, particularmente en asignaturas prácticas como es el caso de estudio.

La relevancia de la estabilidad emocional como predictor extiende el trabajo de [28], demostrando que este rasgo no solo afecta el desempeño general, sino específicamente el aprendizaje de programación, posiblemente por su impacto en la tolerancia a la frustración ante errores de codificación.

Este estudio cumplió su objetivo de demostrar la viabilidad de un modelo predictivo de BRA aplicable con datos limitados (n=94), obteniendo tres logros clave: primero la validación técnica en muestras reducidas: La combinación SMOTE-ReliefF-ensamble logró AUC=72% y precisión=71% [H1], superando limitaciones de tamaño muestral y desbalance. Segundo la relevancia de variables psicoeducativas: Factores como ausencia docente (0.0534) y estabilidad emocional (0.0321) fueron predictores más fuertes que académicos históricos [H2], ofreciendo señales accionables para intervenciones. Por último, la prueba de

concepto escalable: El modelo sirve como base para sistemas de alerta temprana en instituciones sin *Big Data* [H3], priorizando calidad analítica sobre volumen.

El éxito del modelo de ensamble (exactitud: 87%, precisión: 71%) valida el marco teórico de [36], el cual propone que la combinación de múltiples algoritmos mitiga los sesgos en conjuntos de datos desbalanceados. Sin embargo, el recall moderado (50%) coincide con las advertencias de [35] sobre los límites de los enfoques puramente cuantitativos en contextos educativos, reforzando la necesidad de integrar métodos (cuanti-cualitativos).

Para futuros estudios de este trabajo se prevé integrar técnicas de *cost-sensitive learning* [35] para mejorar el recall abordando así el desbalance de clases, además de integrar técnicas de *deep learning* [34] para procesar datos no estructurados (comentarios de estudiantes), ampliando así el marco teórico propuesto por [1].

De igual manera se escalará a más participantes y así aumentar la muestra con la ayuda de técnicas de SMOTE con *Cross Validation Repeted*. Además, se propone implementar un piloto institucional para medir el impacto directo en la reducción del BRA y el abandono escolar, complementado con evaluaciones cualitativas que triangulen las alertas del modelo con las percepciones de docentes y estudiantes.

El estudio puede ser replicado no precisamente para la materia de programación sino en un contexto genérico si se aplican las siguientes acciones: (1) Recolecte datos: Aplique cuestionarios del modelo de los Cinco Grandes y clima escolar (Tabla 1). Preproceso: (2) Use SMOTE para balancear datos si hay menos del 20% de reprobados. (3) Seleccione variables: Priorice predictores con peso ReliefF > 0.03 (Figura 4). (4) Genere alertas: Con el modelo de ensamble, identifique estudiantes en riesgo cada 4 semanas. (5) Intervenga: Enfoque tutorías en variables críticas (ej: talleres de estabilidad emocional si Neuroticismo es alto).

No obstante, la adopción institucional a mediano plazo enfrenta desafíos críticos: (a) la sostenibilidad requiere integrar el modelo a sistemas de gestión académica existentes, evitando carga administrativa adicional; (b) la escalabilidad depende de capacitar a docentes en la interpretación de alertas basadas en predictores psicoeducativos (ej. neuroticismo); y (c) la ética del uso de datos sensibles exige protocolos claros de privacidad. Superar estos retos demanda alianzas estratégicas entre departamentos académicos y psicopedagogía, transformando esta prueba de concepto en una política educativa arraigada.

6. Conflictos de intereses

Se manifiesta que los autores no tienen conflictos de intereses al redactar el presente documento para su publicación.

7. Referencias

[1] Khan, I., Al Sadiri, A., Ahmad, A. R., & Jabeur, N. (2019). Tracking student performance in introductory programming by means of machine learning. In 2019 4th mec international conference on big data and smart city (icbdsc) (págs. 1-6). IEEE.

[2] Hasan, R., Palaniappan, S., Mahmood, S., Abbas, A., Sarker, K. U., & Sattar, M. U. (2020). Predicting student performance in higher educational institutions using video learning analytics and data mining techniques. Applied Sciences, 10(11), 3894.

[3] Franco, E. A., Martínez, R., & Domínguez, V. (2021). Modelos predictivos de riesgo académico en

carreras de computación con minería de datos educativos. Revista de Educación a Distancia (RED), 21 (66).

[4] Gabatino, T., Ogawa, M. B. C., & Crosby, M. E. (2022, June). Abstracting the Understanding and Application of Cognitive Load in Computational Thinking and Modularized Learning. In International Conference on Human-Computer Interaction (pp. 273-286). Cham: Springer International Publishing.

[5] Salmon, A., Hammer, K., Santos, E. A., & Becker, B. A. (2025). Debugging Without Error Messages: How LLM Prompting Strategy Affects Programming Error Explanation Effectiveness. arXiv preprint arXiv:2501.05706.

[6] Selvakani, D., & Vasumathi, K. (2025). Enhancing Dropout Prediction in Higher Education using a Hybrid Machine Learning Approach. Journal of Computer Science, 18, 188-212.

[7] Messer, M., Brown, C., Kölling, M., & Shi, M. (2024). Automated Grading and Feedback Tools for Programming Education: A Systematic Review. ACM Transactions on Computing Education, 24 (1), 1-43.

[8] Cheng, Y.-P., Shen, P.-D., Hung, M.-L., Tsai, C.-W., Lin, C.-H., & Hsu, L. C. (2022). Applying online content-based knowledge awareness and team learning to develop students' programming skills, reduce their anxiety, and regulate cognitive load in a cloud classroom. Universal Access in the Information Society, 21(2), 557–572.

[9] 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, 1-16.

[10] Navarro, R. E. (2016). El rendimiento académico: concepto, investigación y desarrollo. REICE. Revista Iberoamericana Sobre Calidad, Eficacia Y Cambio En Educación, 1(2).

[11] López-Torres, R., Pino, R., & Pedrero, E. (2018). Deserción Escolar. En R. Lopez-Torres, R. Pino, & E. Pedrero, Logro Escolar y Deserción desde el pensamiento complejo (Vol. 1, pág. 11). Académica Española.

[12] Reyes, N, Meneses, A y Díaz, A. (2022). Planificación y gestión del tiempo académico de estudiantes universitarios. Form. Univ. vol.15 no.1

[13] Suhas S Athani, Sharath A Kodli, Mayur N Banavasi, and P.G. Sunitha Hiremath. (2017). Student Performance Predictor using Multiclass Support Vector Classification Algorithm. In 2017 International Conference on Signal Processing and Communication (ICSPC). IEEE, 341–346. https://doi.org/10.1109/CSPC.2017.8305866

[14] Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. " O'Reilly Media, Inc.".

[15] Zaffar, M., Hashmani, M., & Savita, K. (2018). A study of prediction models for students enrolled in programming subjects. In 2018 4th International Conference on Computer and Information Sciences (ICCOINS) (págs. 1-5). IEEE.

[16] Bedregal-Alpaca, N., Aruquipa-Velazco, D., & Cornejo-Aparicio, V. (2020). Técnicas de data Mining para extraer perfiles comportamiento académico y predecir la deserción universitaria. Revista Ibérica de Sistemas

e Tecnologias de Informação, (E27), 592-604

[17] Bergin, S., Mooney, A., Ghent, J., & Quille, K. (2015). Using machine learning techniques to predict introductory programming performance. International Journal of Computer Science and Software Engineering, 4, págs. 323-328.

[18] Contreras, L., Fuentes, H., y Rivas E. (2021) Análisis del rendimiento académico mediante técnicas de aprendizaje automático con métodos de ensamble revista Redipe, (10) (13)

[19] Yaacob, W. W., Sobri, N. M., Nasir, S. M., Norshahidi, N. D., & Husin, W. W. (2020). Predicting student drop-out in higher institution using data mining techniques. (I. Publishing, Ed.) Journal of Physics: Conference Series, 1496 (1), 012005.

[20] Redroban, C., Saavedra, J., Leon, M., Nuñez, S., & Echeverria, F. (2023, May). Educational Data Mining: A Predictive Model to Reduce Student Dropout. In Proceedings of International Conference on Information Technology and Applications: ICITA 2022 (pp. 713-721). Singapore: Springer Nature Singapore.

[21] Soto, C. J. & John, O. P. (2017). The next Big Five Inventory (BFI-2): Developing and assessing a hierarchical model with 15 facets to enhance bandwidth, fidelity, and predictive power. Journal of personality and social psychology, 113(1), 117.

[22] Selvakani, D., & Vasumathi, K. (2025). Enhancing Dropout Prediction in Higher Education using a Hybrid Machine Learning Approach. Journal of Computer Science, 18, 188-212.

[23] Santoso, J. T., Ginantra, N. L. W. S. R., Arifin, M., Riinawati, R., Sudrajat, D., & Rahim, R. (2021). Comparison of classification data mining C4. 5 and Naïve Bayes algorithms of EDM dataset. TEM Journal, 10(4), 1738-1744.

[24] Vidhya, R., & Vadivu, G. (2021). Retracted article: towards developing an ensemble based two-level student classification model (ESCM) using advanced learning patterns and analytics. Journal of Ambient Intelligence and Humanized Computing, 12(7), 7095-7105.

[25] Tjahyadi, H., & Tude, K. N. (2025). The Implementation of Educational Data Mining in Predicting Students' Academic Achievement in Mathematics at a Private Elementary School. International Journal of Information and Education Technology, 15(1).

[26] Bello, F. A., Kóhler, J., Hinrechsen, K., Araya, V., Hidalgo, L., & Jara, J. L. (2020, November). Using machine learning methods to identify significant variables for the prediction of first-year Informatics Engineering students dropout. In 2020 39th International Conference of the Chilean Computer Science Society (SCCC) (pp. 1-5). IEEE.

[27] Veerasamy, A.K.; D'Souza, D.; Apiola, M.-V.; Laakso, M.-J.; Salakoski, T. Using early assessment performance as early warning signs to identify at-risk students in programming courses. In Proceedings of the 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 21–24 October 2020.

[28] Đambić, G.; Krajcar, M.; Bele, D. Machine learning model for early detection of higher education students that need additional attention in introductory programming courses. Int. J. Digit. Technol. Econ. 2016, 1, 1–11.

[29] Adhikari, A., Jain, L. C. & Prasad B. (2017). A state-of-the-art review of knowledge discovery in multiple databases. Journal of Intelligent Systems, 26(1), 23-24.

[30] Agrawal, P., Abutarboush, H. F., Ganesh T., & Mohamed, A. W. (2021). Metaheuristic algoritms on feature selection: A survey of one decade of research (2009-2019). Ieee Access, 9, 26766-26791.

[31] Urbanowicz, R., Meeker, M., La Cava, W., Olson, R., & Moore, J. (2018). Relief-based feature selection: Introduction and review. Journal of Biomedical Informatics (85), 189-203.

[32] Coelho, F. Castro, C. Braga, A. P., & Verleysen, M. (2019). Semi-supervised relevance index for feature selection. Neural Computing and Applications, 31, 989-997.

[33] Wang, S., Dai, Y., Shen, J., & Xuan, J. (2021). Research on expansion and classification of imbalanced data based on SMOTE algorithm. Scientific reports, 11(1), 24039.

[34] Kabari, L. G., & Onwuka, U. C. (2019). Comparison of bagging and voting ensemble machine learning algorithm as a classifier. International Journals of Advanced Research in Computer Science and software engineering, 19-23.

[35] Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., ... & Nahavandi, S. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information fusion, 76, 243-297.

[36] Hastie, T., Tibshirani, R., & Friedman, J. (2017). The elements of statistical learning: data mining, inference, and prediction.