Maestría en

# CIENCIA DE DATOS Y MÁQUINAS DE APRENDIZAJE CON MENCIÓN EN INTELIGENCIA ARTIFICIAL

**Trabajo previo a la obtención de título de**

**Magister en Ciencia de Datos y máquinas de aprendizaje con mención en Inteligencia Artificial**


**AUTOR/ES:**

Encalada Hidalgo Edwin Ismael

Chávez Guerrero Guillermo David

Marchán Salgado Francisco Xavier

Nieto Trujillo María Fernanda

Paredes Cabrera Josselyn Rosario

**TUTOR/ES:**

Karla Estefanía Mora Cajas

Fernanda Paulina Vizcaíno Imacaña


## Diseño de una aplicación web para detección temprana de fraude de pagos en línea en el sector bancario, utilizando técnicas de aprendizaje automático explicable


**Quito - Ecuador**

**Septiembre – 2025**


REINVENTEMOS EL FUTURO

## Certificación de autoría

Nosotros, **(Encalada Hidalgo Edwin Ismael, Chávez Guerrero Guillermo David, Marchán Salgado Francisco Xavier, Nieto Trujillo María Fernanda, Paredes Cabrera Josselyn Rosario),** declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido presentado anteriormente para ningún grado o calificación profesional y que se ha consultado la bibliografía detallada.

Cedemos nuestros derechos de propiedad intelectual a la Universidad Internacional del Ecuador (UIDE), para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, su reglamento y demás disposiciones legales.


---

**Firma**
**(Encalada Hidalgo Edwin Ismael)**


---

**Firma**
**(Chávez Guerrero Guillermo David)**


---

**Firma**
**(Marchán Salgado Francisco Xavier)**


---

**Firma**
**(Nieto Trujillo María Fernanda)**


---

**Firma**
**(Paredes Cabrera Josselyn Rosario)**

# Autorización de Derechos de Propiedad Intelectual

Nosotros, **(Encalada Hidalgo Edwin Ismael, Chávez Guerrero Guillermo David, Marchán Salgado Francisco Xavier, Nieto Trujillo María Fernanda, Paredes Cabrera Josselyn Rosario)**, en calidad de autores del trabajo de investigación titulado ***Portal web de Detección Temprana de Fraude en Pagos en Línea para Banca, basado en Aprendizaje Automático,*** autorizamos a la Universidad Internacional del Ecuador (UIDE) para hacer uso de todos los contenidos que nos pertenecen o de parte de los que contiene esta obra, con fines estrictamente académicos o de investigación. Los derechos que como autores nos corresponden, lo establecido en los artículos 5, 6, 8, 19 y demás pertinentes de la Ley de Propiedad Intelectual y su Reglamento en Ecuador.

D. M. Quito, diciembre 2025

Firmado electrónicamente por:
EDWIN ISMAEL
ENCALADA HIDALGO
Validar únicamente con FirmaEC

---

**Firma**
**(Encalada Hidalgo Edwin Ismael)**

Firmado electrónicamente por:
GUILLERMO DAVID
CHAVEZ GUERRERO
Validar únicamente con FirmaEC

---

**Firma**
**(Chávez Guerrero Guillermo David)**

Firmado electrónicamente por:
FRANCISCO XAVIER
MARCHÁN SALGADO
Validar únicamente con FirmaEC

---

**Firma**
**(Marchán Salgado Francisco Xavier)**

Firmado electrónicamente por:
MARIA FERNANDA
NIETO TRUJILLO
Validar únicamente con FirmaEC

---

**Firma**
**(Nieto Trujillo María Fernanda)**

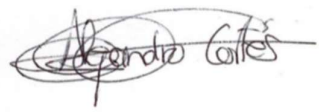
Firmado electrónicamente por:
JOSSELYN ROSARIO
PAREDES CABRERA
Validar únicamente con FirmaEC

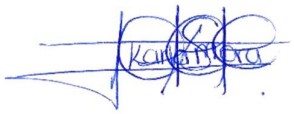
---

**Firma**
**(Paredes Cabrera Josselyn Rosario)**

## Aprobación de dirección y coordinación del programa

Nosotros, **Ing. Alejandro Cortés Director EIG y Mgtr. Karla Mora Coordinadora UIDE,** declaramos que: **(Encalada Hidalgo Edwin Ismael, Chávez Guerrero Guillermo David, Marchán Salgado Francisco Xavier, Nieto Trujillo María Fernanda, Paredes Cabrera Josselyn Rosario)** son los autores exclusivos de la presente investigación y que ésta es original, auténtica y personal de ellos.





---

Alejandro Cortés
Director de la
Maestría en Ciencia de datos y máquinas de aprendizaje con mención en Inteligencia Artificial

---

Karla Mora
Coordinadora de la
Maestría en Ciencia de datos y máquinas de aprendizaje con mención en Inteligencia Artificial

# DEDICATORIA

A mis padres, Fernando y Rosario, por ser mi guía, mi fortaleza y mi luz en cada paso. A mi perrita Nala, que con su compañía y desvelos llenó mis noches de ternura. Y a mi novio Marcos, por su apoyo constante y por creer siempre en mí. Este logro es de ustedes, con todo mi cariño.

Josselyn Rosario Paredes Cabrera

A mis padres, Ximena y Jorge, por ser mi fuerza en cada paso y un ejemplo a seguir. A mis hermanos, Álvaro y Emilio, por recordarme que nunca camino sola. Y a mi Zoé, mi amor de cuatro patas, que con su compañía incondicional hizo más ligeros los días difíciles. Este logro también es de ustedes.

María Fernanda Nieto Trujillo

Dedico este trabajo a mi madre, quien desde siempre me ha enseñado el valor del esfuerzo, la perseverancia y la importancia de nunca rendirme. A mi hermana, por ser un pilar constante en mi vida y brindarme su apoyo incondicional en cada etapa. A mi novia, cuya motivación, cariño y compañía hicieron de este camino un proceso más ligero y significativo. Y a mi tía, que, aunque hoy se encuentra en el cielo, continúa guiando mis pasos y dándome fuerza a través de su recuerdo. A todos ustedes, mi profundo agradecimiento.

Guillermo David Chávez Guerrero

Dedico este logro a mis padres. A mi madre, por su presencia constante, por acompañarme en cada decisión y ser siempre un pilar de apoyo incondicional. A mi padre, por motivarme a perseguir mis metas y confiar en mis capacidades.

Agradezco profundamente a mis amigas Mafer y Josselyn; su compañía hizo que el camino de la maestría fuera más llevadero y menos agotador. A Álvaro, mi amigo más antiguo, cuya ayuda oportuna fue un verdadero salvavidas en los momentos en que sentí que todo se complicaba. A Karen, por su ánimo permanente y su apoyo en cada etapa de este proceso; tus palabras de aliento hicieron que este camino fuera mucho menos difícil. A Cynthia, por mantenerse siempre pendiente de mí y por ser, a pesar del tiempo, una de mis más grandes amistades; gracias a ti estoy hoy aquí. A Aída, quien, aun sin dominar esta área de estudio, siempre mostró una energía admirable y una gran disposición para ayudarme, ofreciéndome perspectivas que enriquecieron mi proceso.

A todos ustedes, gracias. Este logro también es suyo. Lo he conseguido y seguiré avanzando, con la esperanza de que se sientan orgullosos de mí.

Ewdin Ismael Encalada Hidalgo

Dedico este trabajo de maestría, con profundo amor y gratitud, a mi familia, quienes han sido el pilar fundamental en cada etapa de mi formación personal y profesional.

A mi madre, María Elena Salgado, por su apoyo incondicional, su fortaleza, sus consejos y por creer en mí incluso en los momentos más difíciles. Su ejemplo de perseverancia ha sido una guía constante en este camino.

A mi hermana, Daniela Marchán, por su compañía, comprensión y palabras de aliento, que siempre han sido un impulso para seguir adelante.

A mi padre, Freddy Marchán, por su esfuerzo, sacrificio y por inculcarme valores que han marcado mi vida. Su respaldo y confianza han sido fundamentales para alcanzar este logro.

Este logro es también de ustedes. Gracias por ser mi mayor motivación y apoyo permanente.

Francisco Xavier Marchán Salgado

# AGRADECIMIENTOS

Expreso mi gratitud a quienes hicieron posible la culminación de esta tesis. A mis compañeros, por el esfuerzo y la perseverancia compartida; a mi familia y amigos, por su paciencia, comprensión y constante ánimo; y a mis docentes, por cada enseñanza impartida. Gracias también a todas las personas que, de manera directa o indirecta, contribuyeron al desarrollo de este proyecto.

Josselyn Rosario Paredes Cabrera

Agradezco a la Universidad Internacional del Ecuador y a sus docentes por el conocimiento y la orientación recibidos a lo largo de este proceso. Extiendo también mi gratitud a mis compañeros, cuyo apoyo y compañía hicieron más enriquecedor este camino académico.

María Fernanda Nieto Trujillo

Quiero expresar mi sincero agradecimiento a las personas que hicieron posible la culminación de este trabajo. A mi madre, por su esfuerzo inagotable, su ejemplo diario y por enseñarme a enfrentar cada desafío con firmeza. A mi hermana, por su compañía, paciencia y apoyo permanente. A mi novia, por motivarme, creer en mí y ser un impulso fundamental en cada momento del proceso. Y a mi tía, cuya presencia espiritual continúa brindándome luz y fortaleza desde el cielo. A cada uno de ustedes, gracias por ser parte esencial de este logro.

Guillermo David Chávez Guerrero

Expreso mi agradecimiento a mi madre, por su entrega constante, su ejemplo de fortaleza y las enseñanzas que han guiado mi camino. A mi padre, por su apoyo, confianza y por inculcarme valores que han sido fundamentales en mi formación personal y académica. Y a mi hermana, por su compañía, comprensión y respaldo incondicional a lo largo de todo este proceso.

Francisco Xavier Marchán Salgado

Expreso mi más sincero agradecimiento a mis padres por su apoyo incondicional, su paciencia y su constante motivación a lo largo de todo mi proceso académico, ya que su confianza y apoyo fueron fundamentales para alcanzar este objetivo. A mis amigos, gracias por su compañía, comprensión y palabras de aliento, que hicieron más llevadero este camino y aportaron equilibrio en los momentos de mayor exigencia. Finalmente, agradezco a todas las personas que, directa o indirectamente, contribuyeron a la culminación de este trabajo, brindando apoyo académico, emocional o motivacional.

Edwin Ismael Encalada Hidalgo

# RESUMEN

El fraude en pagos en línea constituye un problema relevante para el sector bancario, ya que implica el uso no autorizado de información financiera para realizar transacciones fraudulentas, generando pérdidas económicas y afectando la confianza de los usuarios en los servicios digitales. El crecimiento del comercio electrónico y de los sistemas de pago digitales ha incrementado la complejidad y frecuencia de estas prácticas, lo que exige la implementación de mecanismos avanzados de detección y prevención.

El objetivo de este proyecto es desarrollar modelos predictivos de detección de fraude en pagos en línea mediante técnicas de minería de datos y aprendizaje automático. Para ello, se aplicó la metodología Ágil (Scrum y Kanban), incorporando etapas de preprocesamiento de datos, selección de variables relevantes, entrenamiento de modelos de aprendizaje supervisado y evaluación mediante métricas de clasificación. Adicionalmente, se emplearon técnicas de optimización y validación cruzada para mejorar el desempeño de los modelos y reducir el riesgo de sobreajuste.

Como resultado, se implementó una aplicación web local que integra los modelos con mejor desempeño, permitiendo la identificación de transacciones fraudulentas y la visualización de métricas de evaluación. Los resultados obtenidos evidencian que el uso de modelos de aprendizaje automático constituye una herramienta eficaz de apoyo para fortalecer los procesos de control y prevención del fraude en el contexto de los pagos digitales bancarios.

*Palabras clave: fraude, pagos en línea, banca, minería de datos, aprendizaje automático*

# ABSTRACT

Online payment fraud represents a significant challenge for the banking sector, as it involves the unauthorized use of financial information to conduct fraudulent transactions. This results in economic losses and undermines user trust in digital services. The growth of e-commerce and digital payment systems has increased the complexity and frequency of these practices, making it necessary to implement advanced detection and prevention mechanisms.

The objective of this project is to develop predictive models for online payment fraud detection using data mining and machine learning techniques. To this end, an Agile methodology (Scrum and Kanban) was applied, incorporating stages of data preprocessing, relevant feature selection, supervised learning model training, and evaluation through classification metrics. Additionally, optimization techniques and cross-validation were employed to improve model performance and reduce the risk of overfitting.

As a result, a local web application was implemented that integrates the best-performing models, enabling the identification of fraudulent transactions and the visualization of evaluation metrics. The results obtained demonstrate that the use of machine learning models constitutes an effective support tool to strengthen fraud control and prevention processes in the context of digital banking payments.

*Keywords: fraud, online payments, banking, data mining, machine learning*

# Contenido

## LISTA DE TABLAS

## LISTA DE FIGURAS

# CAPITULO 1

## 1. INTRODUCCIÓN

### 1.1 Definición del proyecto

Se desarrollará una aplicación web denominada FraudOps Portal, orientada a la visualización y gestión de alertas de fraude en pagos en línea dentro de una entidad bancaria. El proyecto incorporará técnicas de aprendizaje automático y profundo para evaluar el riesgo de las transacciones y generar explicaciones interpretables para el personal técnico de las entidades bancarias.

Se explorarán y evaluarán diversos algoritmos, seleccionando aquellos que logren el mejor equilibrio entre precisión y capacidad explicativa en la detección de fraude. Como valor diferenciador, el FraudOps Portal ofrecerá interfaces adaptadas a usuarios no técnicos mediante resúmenes ejecutivos y funcionalidades de informes descargables, con el propósito de fortalecer la toma de decisiones y contribuir a la estandarización de los procesos de control en el ámbito bancario.

El conjunto de datos públicos "Bank Transaction Fraud Detection" de la plataforma Kaggle fue elegido para tratar el problema de detectar fraudes en las transacciones bancarias. Esta base de datos proporciona transacciones auténticas (anonimizadas), lo cual posibilita el uso de métodos de minería de datos y aprendizaje automático en circunstancias parecidas a un ambiente financiero verdadero. Al ser de acceso público, asegura la transparencia en términos de metodología, que se pueda comparar con estudios anteriores y que sea reproducible. Además, tiene retos específicos del fraude financiero (desequilibrio de clases, necesidad de ingeniería de variables, evaluación rigurosa) lo cual aumenta su valor académico. No obstante, como los datos son escasos y específicos en términos contextuales, los resultados se restringirán al ámbito de la colección de datos, sin intención de generalizar automáticamente a todas las entidades financieras.

## 1.2 Justificación e importancia del trabajo de investigación

El crecimiento del comercio electrónico y los servicios financieros digitales ha incrementado exponencialmente el volumen de pagos en línea a nivel global.

Las pérdidas ocasionadas por fraudes con tarjetas a nivel mundial alcanzaron los 33.500 millones de dólares en 2022, lo que representa un incremento respecto a los 28.400 millones de 2020 y a los 27.900 millones de 2018. (Nilson Report, 2023). De igual manera en Ecuador, cerca de un 22% de las entidades financieras expresan preocupación por el fraude a través de la web digital (Superintendencia de Economía Popular y Solidaria, 2021).

De este modo, al utilizar métodos de aprendizaje automático con un enfoque explicativo no solo permite identificar patrones anómalos de las transacciones, sino también entender cómo se generan las alarmas del sistema, aspecto fundamental en entornos regulados como el financiero.

En este contexto se propone una solución basada en un portal web enfocada en la detección temprana de fraude, que combina precisión con la interpretación. De esta manera, el proyecto busca garantizar un enfoque metodológico sólido que contribuya el fortalecimiento de los sistemas de seguridad en los pagos digitales, contribuyendo a la prevención de pérdidas económicas y a la confianza en los servicios financieros digitales.

Si bien la literatura valida el uso general de árboles de decisión en la detección de fraudes, la elección específica del algoritmo CART (Classification and Regression Trees) para este proyecto se debe a tres razones técnicas principales que lo hacen superior a los métodos tradicionales como ID3 o C4.5 en el contexto bancario.

La identificación de fraude requiere el manejo eficaz de variables numéricas, pues mucho depende de variables continuas como la cantidad de dinero en la transacción, el saldo de la cuenta o el tiempo que ha pasado. CART gestiona las variables continuas por medio de particiones

binarias, lo cual evita que se pierda información al discretizar montos financieros en intervalos arbitrarios; esto es diferente con ID3, que fue diseñado sobre todo para atributos categóricos. Asimismo, CART mejora la eficiencia computacional al utilizar el índice de Gini como criterio de impureza en vez de la entropía que C4.5 emplea; esto evita la necesidad de calcular logaritmos complejos y es más eficiente en términos computacionales, lo cual es una ventaja fundamental cuando se procesan enormes cantidades de transacciones históricas y permite un entrenamiento más rápido sin comprometer precisión al separar clases.

En última instancia, la estructura estrictamente binaria del modelo lo hace más robusto porque previene que los datos se fragmenten en exceso, algo característico de algoritmos con varias divisiones. Además, al combinarse con mecanismos de poda, ayuda a reducir el sobreajuste, un problema común en modelos de detección de fraude según Afriyie et al. (2023).

### 1.3 Alcance

El proyecto incluye una generalización metodológica que se realiza a través de un portal modular, el cual utiliza técnicas de aprendizaje automático y análisis de datos para proponer un enfoque inicial para detectar y analizar el fraude. La arquitectura del sistema permanece abierta y se puede ampliar, lo que posibilita la inclusión de nuevos métodos, técnicas o algoritmos en etapas posteriores, conforme avanza lo académico y lo tecnológico. Además, se incluye la administración de las evidencias producidas por los modelos. Las imágenes y los resultados que aparecen en el portal son generados directamente a partir del código Python vinculado a cada modelo mediante scripts independientes encargados de producir las figuras y salidas pertinentes. El sistema, después de organizar estas evidencias, las utiliza como insumos para desarrollar, revisar y validar los informes analíticos.

Una gestión integral del ciclo de vida de los reportes, que incluye las etapas DRAFT, IN_REVIEW, OBSERVED y APPROVED a través de un flujo de estados estructurado, es también

implementada por el sistema. Esto asegura el seguimiento y control durante la validación y análisis del proceso. Además, se incluye una administración de roles diferenciados: los administradores (ADMIN) son los encargados de configurar el sistema, gestionar usuarios y controlar accesos; los analistas (ANALYST) producen informes y conclusiones a partir de las evidencias que los modelos generan; los supervisores (SUPERVISOR) son responsables de revisar, observar y aprobar dichos informes; y los usuarios visualizadores (VIEWER) tienen acceso limitado únicamente a aquellos informes que han sido autorizados y a su exportación, por ejemplo en formato PDF, restringido exclusivamente a los analistas designados por la administración.

Respecto al tratamiento y almacenamiento de datos, el proyecto utiliza una base de datos SQLite para gestionar usuarios, asignaciones y reportes con un resguardo local. Se incluyen validaciones que garantizan la integridad y la consistencia de los datos. En última instancia, el sistema fue creado con criterios de flexibilidad y escalabilidad a nivel estructural, mediante una arquitectura modular que posibilita su desarrollo futuro sin afectar la estabilidad del portal.

En lo que se refiere a las limitaciones y los alcances, el proyecto no incluye la implementación en un ambiente productivo real, puesto que su uso queda restringido a un entorno local de pruebas con propósitos académicos y demostrativos. Asimismo, no se tiene en cuenta la integración con infraestructura corporativa avanzada, como los servicios de SSO (Single Sign-On), la supervisión constante, la alta disponibilidad o los procesos integrales de hardening. El mantenimiento o soporte en tiempo real tampoco se considera, ya que el objetivo primordial es crear un prototipo operativo enfocado en la demostración conceptual y el análisis.

## 1.4 Objetivos

### *1.4.1 Objetivo general*

Diseñar un sistema de detección de transacciones fraudulentas en pagos en línea, basado en modelos de aprendizaje automático que utilicen datos históricos de operaciones financieras digitales, e integrarlo en un portal web que permita visualizar, explicar y gestionar los resultados de manera accesible, ofreciendo además informes descargables para públicos técnicos y no técnicos.

### *1.4.2 Objetivos específicos*

1. Analizar datasets de transacciones financieras históricos que sean adecuados para entrenar los modelos de aprendizaje automático del portal web
2. Entrenar distintos modelos de aprendizaje automático supervisados y no supervisados, evaluando su desempeño mediante métricas como precisión, recall, F1-Score, AUC-PR y tasa de falsos positivos, para obtener el mejor modelo.
3. Desarrollar un portal web interactivo que permita a usuarios técnicos y no técnicos visualizar las predicciones, acceder a resúmenes ejecutivos, explorar explicaciones detalladas y generar informes descargables en formatos estándar.

# CAPITULO 2

## 2. REVISIÓN DE LITERATURA

### 2.1 Estado del Arte

Las soluciones desarrolladas con machine learning se posicionan como una herramienta clave para reducir los efectos económicos y sociales que genera el fraude. La literatura revisada indica que algoritmos como Random Forest, Gradient Boosting y las redes neuronales alcanzan niveles de precisión más altos, especialmente en operaciones digitales y pagos con tarjetas. No obstante, su rendimiento está condicionado por la calidad de los datos disponibles y por la capacidad de los modelos para ajustarse a nuevas formas de fraude.

En un estudio de Afriyie et al. (2023), se evaluó el árbol de decisión junto con regresión logística y Random Forest para detectar fraude con tarjetas de crédito. El árbol logró una precisión del 92 % y un AUC del 94,5 %, aunque los autores mencionan el riesgo de sobreajuste. El modelo con mejor rendimiento fue Random Forest (96 % de precisión, 98,9 % AUC), por lo que se recomienda como la mejor alternativa en este contexto. Este estudio muestra que, aunque los árboles simples siguen siendo útiles, los modelos adecuados pueden mejorar significativamente las tareas de detección de fraude (Afriyie, et al., 2023).

Según Ali et al. (2022), en la investigación de “Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review”, los árboles de decisión son una técnica ampliamente usada en la detección de fraude financiero, apareciendo en muchos estudios revisados en su análisis sistemático. En casos como los de Devi & Kavitha o en estudios sobre fraude de seguros, los árboles obtienen una alta precisión o incluso superan otros métodos clásicos. Sin embargo, dichos estudios también sufren problemas importantes de desbalance de clases, lo que puede limitar la generalización del modelo si no se aplican estrategias adecuadas de balanceo (Ali, et al., 2022).

Los autores Flondor, Donath y Neamțu (2024) desarrollaron un modelo para la detección de fraude en tarjetas bancarias utilizando un árbol de decisión, entrenado con datos reales de transacciones. El estudio demostró que este enfoque permite identificar patrones anómalos de manera efectiva, lo que lo hace útil para el monitoreo preventivo en entornos bancarios. Aunque no se reportan métricas detalladas como AUC o F1, los resultados sugieren que los árboles de decisión son una herramienta viable para la detección de fraude en tiempo real (Flondor, Donath, & Neamtu, 2024).

Salunke, Phalke y Madavi et al. (2025) desarrollaron un modelo híbrido para la detección de fraude con tarjetas bancarias que combina regresión logística, árbol de decisión y Random Forest. El árbol de decisión como algoritmo individual logró una precisión cercana al 77% y aproximadamente un 83% de recall, mientras que el modelo combinado logró una precisión del 99% y más del 98% tanto en precisión como en recall. Estos resultados muestran que los árboles de decisión son eficaces para identificar patrones de fraude y que la combinación con otros algoritmos mejora significativamente el rendimiento, lo que demuestra el potencial de los enfoques híbridos para la detección temprana del fraude en el entorno financiero (Salunke, Phalke, Madavi, Kumre, & Bobhate, 2025)

La selección de XGBoost para detectar fraudes en pagos en línea se apoya en varias propiedades esenciales. Primero, su habilidad para aprender interacciones complejas entre variables es particularmente apropiada en situaciones donde el fraude se presenta mediante combinaciones no triviales de atributos, como la ubicación, el historial del usuario, el monto de la transacción y el dispositivo. Esto ocurre sin que sea necesario definir explícitamente dichas interacciones. Además, XGBoost muestra una gran solidez en relación a datos estructurados que contienen un elevado número de variables, lo cual es común en los sistemas de pago por Internet que producen muchos

atributos transaccionales, temporales, geográficos y de comportamiento. Esto mejora su rendimiento en situaciones con alta dimensionalidad.

Su capacidad para gestionar clases desbalanceadas es otro aspecto importante, en particular cuando se trata de detectar fraudes, donde las transacciones fraudulentas constituyen una pequeña parte del total. XGBoost posibilita la modificación de pesos para la clase minoritaria y su adaptación a esquemas de muestreo, lo que simplifica la identificación eficaz de eventos poco comunes. Su eficiencia computacional y su capacidad de escalar se añaden a lo anterior, pues el algoritmo está optimizado para ambientes con grandes volúmenes de datos y permite la paralelización. Esto lo vuelve apropiado para sistemas que necesitan evaluaciones de riesgo en tiempo real o casi real.

Su uso está respaldado por la evidencia en la literatura reciente desde un punto de vista empírico: Hajek et al. (2022) llevaron a cabo la validación de un marco basado en XGBoost sobre más de seis millones de transacciones móviles e indicaron que este modelo tuvo un mejor rendimiento que los métodos tradicionales, tanto en cuanto a métricas estándar como en reducción de costos. Por su parte, Shi (2024) demostró que, en escenarios con tarjetas de crédito muy desbalanceados, el XGBoost fue superior a una red neuronal artificial para identificar la clase minoritaria. Por último, a pesar de que los modelos de boosting son más difíciles de interpretar que un árbol de decisión simple, XGBoost proporciona una interpretación razonable a través de herramientas como SHAP, que posibilitan calcular la aportación de cada variable a la predicción de fraude. Este es un factor fundamental para cumplir con las exigencias regulatorias y generar confianza en sistemas críticos, como han indicado Almalki y Masud (2025).

En conjunto, estos factores hacen que XGBoost sea una opción sólida para la detección de fraude en pagos en línea, por su capacidad técnica, su adaptabilidad al entorno de datos desbalanceados, y sus resultados verificables en la literatura.

Las redes neuronales artificiales (ANN) han demostrado ser uno de los métodos más eficaces para detectar fraudes, gracias a su habilidad para reconocer patrones no lineales y complejos en grandes cantidades de información. Las ANN, a diferencia de los modelos tradicionales basados en reglas, posibilitan la detección de relaciones subyacentes y delicadas que suelen ser propias del comportamiento fraudulento (Bhattacharyya, Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 2011) Esta habilidad es particularmente importante teniendo en cuenta que el fraude no sigue patrones sencillos o fijos.

Las redes neuronales pueden adaptarse con facilidad y aprender de manera constante, lo que les permite renovarse a medida que se desarrollan nuevas estrategias fraudulentas. Esto es esencial en contextos cambiantes donde los estafadores alteran sus tácticas de manera continua (West, 2016). En este sentido, las ANN tienen un beneficio importante en comparación con los métodos estáticos, cuya eficiencia se reduce a medida que pasa el tiempo.

Varios estudios han evidenciado, además, que las redes neuronales logran niveles de exactitud más altos y tasas de falsos negativos y positivos más bajos en comparación con enfoques tradicionales como los árboles de decisión o las regresiones (Carcillo F. L., 2019). Esto mejora la confiabilidad del sistema y disminuye las interrupciones no necesarias en operaciones legítimas.

Según James, Witten, Hastie / Tibshirani (James, Witten, Hastie, & Tibshirani, 2013), el proceso de aprendizaje estadístico se centra en estimar la función $f$ que describe la relación entre las variables predictoras y la variable respuesta. Para lograrlo, la mayoría de los algoritmos reducen el problema de aprendizaje a la tarea de calcular un conjunto de valores intrínsecos llamados parámetros.

Una vez que se haya encontrado un modelo funcional, se debe realizar un entrenamiento que utilice datos observados para estimar estos parámetros donde matemáticamente, este proceso tiene como objetivo encontrar valores específicos (en su mayoría son $\beta$ o $\theta$) que mejor se ajusten

al modelo a los datos de entrenamiento y al mismo tiempo minimicen los errores (James, Witten, Hastie, & Tibshirani, 2013).

Por lo tanto, un parámetro se define como cualquier coeficiente, peso o elemento estructural que un algoritmo aprende o calcula automáticamente durante la fase de entrenamiento. Estos valores no los determina el investigador, sino que son el resultado directo de la optimización matemática del modelo del conjunto de datos (James, Witten, Hastie, & Tibshirani, 2013).

Los hiperparámetros también conocidos como parámetros de ajuste, son configuraciones externas al proceso de aprendizaje automático. Según (James, Witten, Hastie, & Tibshirani, 2013), estos valores son cruciales para regular el comportamiento del modelo porque controlan directamente el equilibrio entre complejidad y capacidad de generalización.

Los autores señalan la importancia de estos valores cuando se habla del equilibrio entre sesgo y varianza. Mientras que los parámetros intrínsecos intentan ajustar los datos, los hiperparámetros actúan como restricciones que evitan al modelo volverse complejo y que no memorice el ruido de los datos de entrenamiento (sobreajuste).

**Tabla 1**

*Parámetros e hiperparámetros de Modelos*

| | ***Parámetro*** | ***Definición*** | ***Impacto y Uso*** |
|---|---|---|---|
| ***Árbol de decisión*** | **criterion** | Función para medir la calidad de la división. | Se puede usar Gini o Entropy y determina cómo el árbol debe separar los datos. |
| | **max_depth** | Es la profundidad máxima a la que puede llegar el árbol. | • None: El árbol crece hasta que las hojas sean puras pero existe un alto riesgo de *overfitting.*<br>• Usar valores numéricos haciendo al modelo más simple y generalizable. |

| | | | |
|---|---|---|---|
| | **min_samples_split** | Número mínimo de muestras para dividir un nodo interno. | Suaviza el modelo si el valor aumenta se evita que el árbol cree ramas para pocos datos. |
| | **min_samples_leaf** | Número mínimo de muestras por nodo hoja | Aumentar el valor permite reducir el ruido y el sobreajuste. |
| | **max_features** | El número de características que se deben considerar para buscar la mejor división. | Añade aleatoriedad haciendo que el árbol sea robusto y menos correlacionado con solo una variable dominante. |
| | **class_weight** | Son pesos asociados a las clases. | Funcional para datos desbalanceados. |
| | **ccp_alpha** | Parámetro de costo-complejidad. | Se usa para podar el árbol final obtenido donde un valor mayor que 0 simplifica el árbol. |
| **XGBoost** | **n_estimators** | Número de árboles de decisión que conforman el modelo. | Un mayor número de árboles permite capturar patrones complejos. En combinación con una tasa de aprendizaje baja, mejora la capacidad de generalización, aunque incrementa el tiempo de entrenamiento. |
| | **max_depth** | Profundidad máxima permitida para cada árbol. | Controla la complejidad del modelo. Valores altos pueden generar sobreajuste, mientras que valores moderados, como el utilizado, permiten capturar interacciones relevantes sin perder generalización. |
| | **learning_rate** | Peso o contribución de cada árbol al modelo final. | Valores bajos producen un aprendizaje más gradual y estable, reduciendo el riesgo de sobreajuste. Requiere un mayor número de árboles para alcanzar un buen desempeño. |
| | **subsample** | Proporción de observaciones utilizadas para entrenar cada árbol. | Introduce aleatoriedad en el entrenamiento, reduce la varianza del modelo y mejora su robustez frente a ruido y valores atípicos. |
| | **colsample_bytree** | Proporción de variables consideradas en cada árbol. | Evita que el modelo dependa excesivamente de un subconjunto reducido de variables y mejora la estabilidad del modelo. |
| | **objective** | Función objetivo que define el tipo de problema a resolver. | Al utilizar binary:logistic, el modelo se adapta a problemas de clasificación binaria y produce probabilidades asociadas a la clase fraudulenta. |
| | **eval_metric** | Métrica utilizada internamente para evaluar el | La función logloss penaliza predicciones incorrectas con alta |

| | | | |
|---|---|---|---|
| | | desempeño durante el entrenamiento. | confianza, lo que mejora la calidad de las probabilidades estimadas. |
| | **scale_pos_weight** | Peso asignado a la clase minoritaria. | Permite manejar el fuerte desbalance entre transacciones fraudulentas y legítimas, aumentando la sensibilidad del modelo para detectar fraudes. |
| | **random_state** | Semilla aleatoria utilizada en el entrenamiento. | Garantiza la reproducibilidad de los resultados y la consistencia en ejecuciones repetidas. |
| | **tree_method** | Método utilizado para construir los árboles de decisión. | El método hist optimiza el uso de memoria y reduce el tiempo de entrenamiento, siendo especialmente adecuado para grandes volúmenes de datos. |
| **Random Forest** | **n_estimators** | Número de árboles de decisión que componen el bosque | Un mayor número de árboles reduce la varianza del modelo y mejora la estabilidad de las predicciones. Sin embargo, incrementa el costo computacional. En este estudio se utilizaron valores de 15 y 30 árboles para analizar el equilibrio entre desempeño y eficiencia. |
| | **criterion** | Función utilizada para medir la calidad de una división en cada nodo del árbol. | Se empleó el criterio **Gini**, el cual mide la impureza de los nodos. Este criterio es eficiente computacionalmente y adecuado para grandes volúmenes de datos, permitiendo separar eficazmente transacciones fraudulentas y no fraudulentas. |
| | **max_depth** | Profundidad máxima que puede alcanzar cada árbol del bosque | Limitar la profundidad evita que los árboles crezcan excesivamente y memoricen el ruido de los datos de entrenamiento, reduciendo el sobreajuste. Un valor controlado favorece la generalización del modelo. |
| | **min_samples_split** | Número mínimo de muestras necesarias para dividir un nodo interno | Valores mayores suavizan el modelo, evitando divisiones basadas en pocos registros. Esto es especialmente importante en detección de fraude, donde existen transacciones atípicas poco frecuentes |
| | **min_samples_leaf** | Número mínimo de muestras requeridas en un nodo hoja. | Incrementar este valor reduce el riesgo de sobreajuste y mejora la robustez del modelo, evitando hojas con muy pocas observaciones que podrían representar ruido. |
| | **max_features** | Número máximo de características consideradas al buscar la mejor división en cada nodo | Introduce aleatoriedad en el proceso de construcción de los árboles, reduciendo la correlación entre ellos. |

| | | | |
|---|---|---|---|
| | | | Esto hace que el bosque sea más robusto y generalizable. |
| | **class_weight** | Ponderación asignada a cada clase durante el entrenamiento | Se utilizó **balanced** para compensar el fuerte desbalance entre transacciones fraudulentas y no fraudulentas, penalizando más los errores en la clase minoritaria (fraude) |
| | **bootstrap** | Indica si se utilizan muestras con reemplazo para entrenar cada árbol. | El muestreo bootstrap permite entrenar cada árbol con subconjuntos distintos de datos, incrementando la diversidad del bosque y mejorando su capacidad de generalización |
| | **random_state** | Semilla utilizada para la generación de números aleatorio | Garantiza la reproducibilidad de los resultados, aspecto fundamental en un trabajo de investigación académica. |
| | **threshold** | Valor de probabilidad a partir del cual una transacción se clasifica como fraude | Se ajustó el umbral para maximizar el **F1-score**, permitiendo un mejor equilibrio entre precisión y recall. Este ajuste es clave en contextos bancarios donde los falsos negativos tienen alto costo. |
| **Redes Neuronales** | **units (Capas Densas)** | Cantidad de neuronas artificiales en cada capa oculta de la red. | Se definió una arquitectura de 64 y 32 neuronas para permitir que el modelo capture patrones complejos y características abstractas de los datos sin incrementar excesivamente el costo computacional. |
| | **activation='relu'** | Función de activación Rectified Linear Unit aplicada en las capas ocultas. | Introduce no-linealidad en el modelo, lo que es crucial para aprender fronteras de decisión complejas que no pueden ser separadas por líneas rectas. |
| | **activation='sigmoid'** | Función de activación no lineal que acota la salida entre 0 y 1 | Al utilizarse en la capa de salida, transforma el resultado final en una probabilidad interpretable de pertenencia a la clase "Fraude". |
| | **dropout_rate** | Porcentaje de neuronas que se desactivan aleatoriamente durante cada paso del entrenamiento. | Funciona como regularizador (con tasas de 0.3 y 0.2), reduciendo la dependencia entre neuronas y evitando el sobreajuste (overfitting) para mejorar la generalización. |
| | **optimizer='adam'** | Algoritmo de optimización estocástica basado en estimación de momentos adaptativos. | Ajusta automáticamente la tasa de aprendizaje para cada parámetro, permitiendo una convergencia más rápida y eficiente que el descenso de gradiente tradicional. |

| | | |
|---|---|---|
| **loss='binary_crossentropy'** | Función de pérdida que calcula la entropía cruzada entre las etiquetas reales y las predicciones. | Es la función objetivo estándar para clasificación binaria; penaliza logarítmicamente las predicciones incorrectas, guiando al modelo a distinguir mejor entre clases. |
| **batch_size** | Número de muestras de entrenamiento procesadas antes de actualizar los pesos del modelo. | Un tamaño de 64 proporciona un equilibrio entre la estabilidad de la convergencia del gradiente y la velocidad de entrenamiento en memoria. |
| **epochs (con EarlyStopping)** | Número máximo de iteraciones completas sobre el set de datos. | Se configuran 30 épocas pero con Early Stopping para detener el entrenamiento automáticamente si la pérdida de validación no mejora, garantizando el modelo óptimo. |

*Nota:* Los hiperparámetros presentados en esta tabla fueron seleccionados a partir de pruebas preliminares y recomendaciones de la literatura, con el objetivo de optimizar el desempeño del modelo y mitigar el sobreajuste

*Fuente:* Elaboración Propia

## 2.2 Marco Teórico

### *2.2.1 Comercio electrónico y pagos digitales*

El avance de la tecnología ha impulsado en gran medida el desarrollo de los mercados de telecomunicaciones al igual que la inversión en infraestructura digital, dando lugar a un entorno comercial cada vez más integrado y globalizado, en el que las transacciones económicas pueden realizarse en cualquier parte del mundo. En este contexto nace el comercio electrónico como una modalidad que facilita el intercambio de bienes y servicios a través de Internet, eliminando las limitaciones de tiempo y espacio. A medida que la red se consolida como un medio esencial para la interacción económica, el comercio electrónico ha dejado de ser una opción para convertirse en un elemento indispensable para la operación y competitividad de personas, empresas y regiones. Por lo cual, adoptar esta medida se ha vuelto un requisito para mantenerse vigente en los mercados actuales, mientras que su ausencia puede derivar en procesos de aislamiento y en una desventaja económica y social significativa (Gariboldi, 1999).

En esta línea, el planteamiento de Gariboldi (1999) contribuye a entender que la evolución tecnológica no solo dio origen al comercio electrónico como un nuevo espacio de intercambio, sino que también estableció los fundamentos necesarios para el desarrollo de sistemas de pago digitales que respaldan y hacen viable este modelo de transacciones.

El comercio electrónico, también conocido como *e-commerce*, depende de manera directa de los pagos digitales, puesto que estos establecen el medio a través del cual se llevan a cabo las transacciones de compra y venta en Internet. En este sentido, ambos conceptos mantienen una relación estrecha y complementaria: mientras el comercio electrónico proporciona un entorno adecuado para la oferta y la demanda en línea, los pagos digitales contribuyen con los mecanismos necesarios para ejecutar dichas operaciones de forma segura, ágil y eficiente. Esta interacción ha favorecido la automatización de los procesos comerciales, la expansión hacia nuevos mercados y la consolidación de modelos de negocio basados en plataformas digitales. A pesar de esto, el crecimiento de estas prácticas digitales también ha dado lugar a nuevos desafíos, específicamente en lo referente a la protección de la información y la prevención del fraude en entornos virtuales (Gariboldi, 1999).

#### 2.2.1.1 Comercio Electrónico

El comercio electrónico (e-commerce) se define como la realización de actividades comerciales y transacciones de bienes o servicios a través de plataformas digitales, en las que Internet constituye el medio principal de intercambio. Hoy en día, el comercio electrónico puede sustituir a las tiendas físicas y ha permitido reducir las barreras de entrada para diversos tipos de negocios minoristas. Este fenómeno implica más que una simple interacción entre comprador y vendedor; depende de una infraestructura digital que sustenta su funcionamiento. El comercio electrónico puede entenderse como una versión digital de la compra por catálogo enviada por correo, que en su momento revolucionó el comercio minorista (Bloomenthal, 2025).

De acuerdo con Gariboldi (1999), el comercio electrónico no se confina únicamente a las actividades de compra y venta, sino que también incluye los procesos de promoción, distribución y posventa, los cuales son gestionados mediante canales electrónicos. Actualmente, autores como Talafha (2024) extienden este concepto al definir el comercio electrónico como un ecosistema integral que, además de la interacción entre compradores y vendedores, incorpora la infraestructura tecnológica, los sistemas logísticos y los servicios de pago necesarios para hacer posible la transacción.

El comercio electrónico ofrece diversas ventajas. En primer lugar, proporciona comodidad, ya que está disponible las 24 horas del día durante toda la semana, permitiendo que los consumidores realicen compras en el momento que les resulte conveniente y que las empresas generen ventas incluso fuera de su horario de atención. Además, brinda mayor variedad, pues las empresas pueden ofrecer un catálogo amplio e incluso productos exclusivos en línea, sin necesidad de exhibirlos físicamente. Otra ventaja es la posibilidad de comercializar a nivel internacional, ya que los clientes pueden acceder a los productos desde cualquier lugar del mundo. Asimismo, el comercio electrónico contribuye a reducir costos operativos, al disminuir la necesidad de espacios físicos, personal y otros gastos asociados. Finalmente, permite recopilar información valiosa sobre el comportamiento y las preferencias de los consumidores, lo que facilita la segmentación de mercado y la definición de públicos objetivos (Bloomenthal, 2025).

La naturaleza del comercio electrónico se distingue por su alcance global, su operación continua y la eliminación de intermediarios físicos. Estos rasgos generan un entorno altamente competitivo, pero también expuesto a amenazas cibernéticas. La masificación del e-commerce ha aumentado el volumen de transacciones, creando un terreno propicio para ataques como phishing, robo de identidad y fraudes en los procesos de pago (Lokhande, 2025).

#### 2.2.1.2 Pagos Digitales

Los pagos digitales son mecanismos financieros que permiten enviar y recibir dinero sin usar efectivo, mediante herramientas como tarjetas, transferencias bancarias, billeteras digitales, códigos QR o criptomonedas (Ramayanti et al., 2024). Estos medios hacen que las transacciones sean más rápidas y sencillas, ayudan a reducir costos para las empresas y permiten llevar un mejor control de los movimientos financieros. Su importancia en la economía es clave, porque facilitan que más personas y negocios participen en el sistema financiero, impulsan la actividad comercial y apoyan el crecimiento de las empresas. Además, al hacer posible pagar y cobrar de forma segura y en cuestión de segundos, los pagos digitales contribuyen a dinamizar el comercio y el crecimiento económico. Un mayor uso de estos sistemas se relaciona con incrementos en el PIB, la generación de empleo y el aumento del consumo de bienes y servicios (Inter-American Development Bank, 2022).

En el contexto del comercio electrónico, los pagos digitales representan la última etapa del proceso de compra y, a su vez, la más crítica desde el punto de vista de la seguridad. Según Zuasnábar (2023), la confianza del consumidor en los métodos de pago digital es un factor determinante en la decisión de compra y en la continuidad del uso de plataformas en línea. No obstante, esa misma confianza puede verse afectada por la exposición a fraudes o por la percepción de riesgo ante el manejo de datos personales.

#### 2.2.1.3 Relación entre comercio electrónico, pagos digitales y fraude

La interacción entre comercio electrónico y pagos digitales es esencial para entender la dinámica del fraude en línea. A medida que las transacciones digitales aumentan en volumen y complejidad, también lo hacen las oportunidades de ataque para los ciberdelincuentes. Lokhande (2025) señala que las modalidades más comunes incluyen el uso de tarjetas robadas, la creación de sitios web falsos y la manipulación de sistemas de pago. De manera similar, Ramayanti et al.

(2024) destacan que la rápida digitalización ha superado la capacidad de algunos marcos regulatorios para garantizar una protección efectiva, especialmente en economías emergentes.

Por ello, la seguridad en los pagos digitales se ha convertido en un tema prioritario para gobiernos, instituciones financieras y empresas tecnológicas. El empleo de tecnologías como la autenticación multifactor, la encriptación avanzada y el análisis de comportamiento en tiempo real son actualmente estrategias claves para prevenir y detectar fraudes (Talafha, 2024).

### *2.2.2 Evolución del comercio electrónico*

El comercio electrónico ha evolucionado de manera constante, impulsado por el desarrollo de tecnologías digitales, la expansión del acceso a Internet y la transformación de los hábitos de consumo. Inicialmente concebido como un canal complementario de ventas, el e-commerce se ha consolidado como un componente esencial de la economía global, influyendo directamente en los sistemas de pago, la logística y la seguridad de las transacciones. Esta evolución se vio acelerada de manera significativa durante la pandemia de COVID-19 en 2020, cuando miles de empresas a nivel global, en especial, en América Latina y el Caribe se vieron obligadas a digitalizar sus procesos para mantener su actividad comercial frente a las restricciones sanitarias. En este contexto, se incrementó el uso de pagos electrónicos, billeteras digitales y transferencias inmediatas, lo que permitió a los consumidores adquirir bienes y servicios sin necesidad de acudir físicamente a establecimientos comerciales. Además, gobiernos y entidades financieras fomentaron soluciones digitales para facilitar el acceso a los servicios financieros y continuar las operaciones económicas, lo que generó una adopción más amplia y sostenida de herramientas digitales más allá de la etapa de emergencia sanitaria. Como resultado, el comercio electrónico no solo creció en volumen, sino que se consolidó como una práctica cotidiana en la región, transformando de manera estructural la relación entre consumidores, empresas y mercados (Inter-American Development Bank, 2022).

#### 2.2.2.1 Orígenes del comercio electrónico

Los orígenes del comercio electrónico pueden situarse en la década de 1970, periodo durante el cual se comenzaron a utilizar sistemas de intercambio electrónico de datos (EDI) entre empresas con la finalidad de automatizar procesos como las órdenes de compra y la facturación (Gariboldi, 1999). Estas primeras aplicaciones dieron inicio a la digitalización de actividades comerciales, aunque su uso se encontraba restringido a entornos empresariales cerrados.

En la década de 1990, con la expansión de Internet y la aparición del protocolo HTTP, el comercio electrónico comenzó a adoptar su forma moderna. Las primeras tiendas en línea, como Amazon y eBay, sentaron las bases de la compraventa digital a gran escala (Lokhande, 2025). Este período se caracterizó por el surgimiento de modelos B2C (empresa a consumidor) y B2B (empresa a empresa), acompañados por la introducción de pasarelas de pago y sistemas de seguridad básicos como el cifrado SSL.

#### 2.2.2.2 Expansión y Consolidación (2000-2010)

Durante la primera década del siglo XXI, el comercio electrónico se consolidó como un canal de distribución global. La mejora en la conectividad, el desarrollo de plataformas de pago más seguras y la confianza del consumidor contribuyeron al crecimiento sostenido de las transacciones digitales. Según Talafha (2024), este periodo estuvo marcado por la expansión de los marketplaces, la integración con sistemas logísticos y la aparición de normativas internacionales para la protección de datos y la autenticación de pagos.

Sin embargo, junto con su expansión surgieron los primeros desafíos relacionados con el fraude electrónico. El robo de información financiera, la suplantación de identidad y el uso de tarjetas falsas se convirtieron en amenazas recurrentes. Esto obligó a los proveedores a incorporar mecanismos de verificación más robustos, como la autenticación en dos pasos y los filtros de dirección IP (Ramayanti et al., 2024).

2.2.2.3 La era del comercio móvil y las plataformas digitales (2010–2020)

La masificación de los teléfonos inteligentes y el desarrollo de aplicaciones móviles transformaron radicalmente la manera en que los consumidores realizan compras. Este fenómeno, conocido como m-commerce, permitió que las transacciones se efectuaran en cualquier momento y lugar, aumentando el volumen global del comercio electrónico.

Según Zuasnábar (2023), en este periodo surgieron las billeteras digitales, los pagos con código QR y los sistemas integrados de checkout, que redujeron la fricción en la experiencia de compra. No obstante, el crecimiento del comercio móvil también generó un incremento de fraudes asociados al phishing, al malware financiero y a la ingeniería social. De ahí que la seguridad se convirtiera en un eje central en el diseño de plataformas y medios de pago

2.2.2.4 Transformaciones recientes y tendencias actuales (2020 en adelante)

En los últimos años, el comercio electrónico ha adquirido un papel aún más relevante debido a la pandemia de COVID-19, que aceleró la digitalización del consumo. Las empresas adoptaron rápidamente soluciones de pago sin contacto, plataformas omnicanales y herramientas de análisis de datos para personalizar la oferta. Según Lokhande (2025), esta etapa se caracteriza por la integración de inteligencia artificial, big data y blockchain para optimizar procesos, predecir comportamientos de compra y fortalecer la seguridad de las transacciones.

De forma paralela, los gobiernos y organismos financieros han intensificado la regulación sobre los pagos digitales, buscando equilibrar la innovación con la protección de los consumidores. La evolución actual del e-commerce no sólo apunta a la eficiencia y la comodidad, sino también a la resiliencia frente a riesgos de fraude, suplantación y lavado de activos digitales (Ramayanti et al., 2024).

#### 2.2.2.5 Perspectiva en el contexto de fraudes en línea

La evolución del comercio electrónico ha ido acompañada de una evolución similar en las estrategias de fraude. Cada avance tecnológico y cada mejora en la experiencia de usuario han sido aprovechados también por los ciberdelincuentes para desarrollar técnicas más sofisticadas de ataque. Tal como menciona Talafha (2024), el reto actual no radica únicamente en facilitar el acceso al comercio digital, sino en garantizar que cada fase de la transacción desde la autenticación del usuario hasta la confirmación del pago se realice bajo estándares robustos de seguridad y trazabilidad.

En este sentido, comprender la evolución histórica del e-commerce permite contextualizar la aparición de nuevas vulnerabilidades, así como la necesidad de políticas integrales que aborden simultáneamente la innovación tecnológica y la gestión de riesgos.

#### 2.2.2.6 Riesgos y vulnerabilidades en pagos en línea

El crecimiento acelerado del comercio electrónico y de los sistemas de pago digital ha traído consigo una serie de riesgos y vulnerabilidades que amenazan la seguridad de las transacciones y la confianza de los usuarios. Estas amenazas se originan tanto en factores tecnológicos como brechas de seguridad o fallas en los protocolos de autenticación como en factores humanos, asociados a la ingeniería social o al uso inadecuado de las plataformas. Por ello, comprender los principales riesgos y vulnerabilidades resulta esencial para el análisis del fraude en pagos en línea.

##### 2.2.2.6.1 Naturaleza de los riesgos en pagos digitales

Los pagos en línea implican el intercambio de información sensible, como datos personales, números de tarjeta, contraseñas o credenciales bancarias. Cuando esta información es interceptada o utilizada de manera indebida, se producen pérdidas financieras y deterioro de la confianza en el sistema. Según Talafha (2024), el riesgo en los pagos digitales puede definirse como la posibilidad

de que una transacción sea alterada, interceptada o falsificada por actores no autorizados, afectando la integridad, confidencialidad o disponibilidad del sistema.

Los riesgos pueden clasificarse en tres grandes categorías: operativos, tecnológicos y conductuales. Los riesgos operativos se relacionan con errores humanos o fallas en los procedimientos internos; los tecnológicos derivan de vulnerabilidades del software o hardware; y los conductuales están vinculados al comportamiento del usuario, como el uso de contraseñas débiles o el acceso a sitios fraudulentos (Ramayanti et al., 2024).

#### 2.2.2.6.2 Principales vulnerabilidades técnicas

El malware financiero, que implica la instalación de programas maliciosos en los dispositivos de los usuarios para capturar información sensible o manipular transacciones; las suplantaciones de identidad y el phishing, a través de los cuales los atacantes se hacen pasar por entidades confiables mediante correos electrónicos, mensajes de texto o sitios web falsos con el objetivo de obtener credenciales o datos de pago; y los ataques "Man in the Middle" (MitM), son algunas de las vulnerabilidades más frecuentes en sistemas de pago online, en los que se intercepta el tráfico entre el servidor y el cliente para alterar o sustraer la información transmitida; la clonación de tarjetas y el robo de credenciales, mediante métodos como keyloggers o aprovechamiento de bases de datos vulneradas; y, por último, las deficiencias en los sistemas de cifrado y autenticación, propios de sistemas que no emplean protocolos de seguridad sólidos como HTTPS o cifrado punto a punto, lo cual aumenta considerablemente la amenaza de fraude (Lokhande, 2025).

De acuerdo con Zuasnábar (2023), estas vulnerabilidades no siempre se deben a deficiencias tecnológicas, sino a una combinación de errores humanos y falta de educación digital del consumidor. Por ejemplo, la mayoría de los ataques de phishing tienen éxito porque el usuario desconoce las señales básicas de suplantación o accede a enlaces no verificados.

2.2.2.6.3 Riesgos asociados al fraude electrónico

El fraude en pagos en línea representa uno de los principales desafíos en la economía digital. Según Lokhande (2025), este tipo de fraude se produce cuando una persona o grupo utiliza información falsa o robada para realizar transacciones ilegítimas. Las modalidades más frecuentes incluyen el uso de tarjetas robadas, la creación de cuentas falsas, el reembolso fraudulento y el robo de identidad.

Ramayanti et al. (2024) señalan que, a medida que se perfeccionan las medidas de seguridad, los delincuentes también desarrollan métodos más sofisticados, como el uso de inteligencia artificial para generar correos falsos o deepfakes con el fin de engañar a los sistemas de verificación. Esto genera un entorno dinámico donde la prevención requiere actualizaciones constantes de software y capacitación tanto para usuarios como para empresas.

2.2.2.6.4 Impacto en la confianza y en la adopción de medios digitales

La percepción de riesgo influye directamente en la adopción y continuidad del uso de los pagos digitales. Talafha (2024) destaca que, incluso si la frecuencia real de fraudes es baja, una experiencia negativa puede afectar de manera significativa la confianza del consumidor, disminuyendo la intención de uso de plataformas electrónicas. Por ello, la gestión de riesgos en pagos digitales no sólo tiene una dimensión técnica, sino también psicológica y reputacional.

Zuasnábar (2023) agrega que la confianza del usuario se fortalece cuando las empresas implementan medidas visibles de seguridad como notificaciones de transacción, tokens dinámicos o autenticación biométrica, ya que estas prácticas refuerzan la sensación de control y transparencia.

2.2.2.6.5 Estrategias de mitigación y retos actuales

Frente a los riesgos antes mencionados, las estrategias más comunes de mitigación incluyen la implementación de autenticación multifactor, monitoreo en tiempo real de las transacciones,

encriptación avanzada de datos, y el uso de modelos predictivos basados en inteligencia artificial para detectar patrones anómalos (Talafha, 2024).

El reto principal radica en equilibrar la usabilidad con la seguridad. Un sistema excesivamente rígido puede generar fricción en la experiencia de compra, mientras que uno demasiado flexible puede aumentar la exposición al fraude. Por tanto, la gestión de riesgos debe orientarse hacia un enfoque integral que combine tecnología, educación del usuario y regulación efectiva (Ramayanti et al., 2024).

### *2.2.3 Fraude en transacciones financieras*

El fraude en transacciones financieras constituye una de las principales amenazas para la estabilidad y confianza en los sistemas de pago digitales. Su impacto se ha incrementado significativamente con el crecimiento del comercio electrónico y la digitalización de los servicios financieros, lo que ha abierto nuevas oportunidades para los ciberdelincuentes. En términos generales, este tipo de fraude puede entenderse como toda acción deliberada destinada a obtener un beneficio económico mediante el engaño, la manipulación o el uso indebido de información confidencial dentro de una operación financiera (Vanini et al., 2023).

En el entorno digital, los fraudes suelen manifestarse a través de tácticas como la clonación de tarjetas, el acceso no autorizado a cuentas, la falsificación de identidades o el uso de datos personales robados. Estas prácticas se han vuelto más sofisticadas debido al uso de tecnologías avanzadas como la inteligencia artificial, el aprendizaje automático y la ingeniería social, que facilitan la automatización y personalización de los ataques (Fariha et al., 2025). De acuerdo con estudios recientes, la vulnerabilidad de los sistemas financieros digitales se debe, en gran parte, al incremento de las transacciones en tiempo real y a la dificultad de supervisar de forma manual operaciones masivas en plataformas globales (Singh, 2025).

Entre las modalidades más comunes de fraude en transacciones financieras destacan el phishing, las compras no autorizadas, el uso fraudulento de tarjetas y las transacciones reversadas intencionalmente. En el caso del phishing, el atacante engaña al usuario para obtener datos confidenciales, mientras que, en el fraude con tarjetas, el delincuente emplea información sustraída para realizar compras sin consentimiento del titular (Martínez Pazos et al., 2023).

Asimismo, existen casos en los que empleados o proveedores autorizados manipulan sistemas internos para desviar fondos o alterar registros contables, lo que se conoce como fraude interno o colusión (The Payments Association, s.f.).

Los factores que facilitan este tipo de delitos incluyen la falta de educación financiera de los usuarios, el uso de contraseñas débiles, la ausencia de controles de seguridad robustos y la insuficiente coordinación entre organismos reguladores. Además, el creciente uso de plataformas internacionales de pago y la interoperabilidad entre sistemas de diferentes países generan brechas normativas que los delincuentes pueden aprovechar (Singh, 2025). Por otro lado, la rápida evolución de las tecnologías financieras, si bien impulsa la innovación, también incrementa la superficie de ataque, especialmente cuando las medidas de seguridad no evolucionan al mismo ritmo (Vanini et al., 2023).

Las consecuencias del fraude financiero son múltiples. A nivel económico, generan pérdidas significativas para consumidores, entidades financieras y comercios. Sin embargo, su impacto más profundo recae sobre la confianza del usuario en los canales digitales, afectando la adopción de los pagos en línea y el comercio electrónico (Fariha et al., 2025). A nivel institucional, los fraudes recurrentes deterioran la reputación de las empresas, aumentan los costos operativos y pueden derivar en sanciones legales o regulatorias (Martínez Pazos et al., 2023).

Frente a estos riesgos, las estrategias más comunes de mitigación incluyen la implementación de autenticación multifactor, el monitoreo en tiempo real de las transacciones, la

encriptación avanzada de datos, y el uso de modelos predictivos basados en inteligencia artificial para detectar patrones anómalos (Talafha, 2024). Asimismo, la cooperación interinstitucional entre bancos, proveedores tecnológicos y autoridades regulatorias resulta esencial para fortalecer los mecanismos de detección y respuesta ante posibles fraudes (The Payments Association, s.f.). Finalmente, la educación del usuario sobre prácticas seguras, la adopción de tecnologías de tokenización y la evaluación de riesgo en cada transacción son medidas clave para reducir la exposición a estos delitos (Vanini et al., 2023).

En síntesis, el fraude en transacciones financieras no solo constituye un desafío técnico, sino también social y regulatorio. Su prevención requiere un enfoque integral que combine herramientas tecnológicas avanzadas, políticas de seguridad coherentes y una cultura financiera orientada a la protección del usuario.

#### 2.2.3.1 Tipos de fraude en pagos digitales

El crecimiento del comercio electrónico y la digitalización de los servicios financieros han impulsado un aumento considerable de los fraudes asociados a los pagos digitales. Estos delitos se presentan en diversas formas y con distintos niveles de complejidad, pero todos comparten un mismo objetivo: obtener beneficios económicos ilegítimos mediante la manipulación de sistemas de pago o la suplantación de identidades. Según estudios recientes, la diversificación de los métodos de pago como tarjetas, transferencias instantáneas, billeteras electrónicas y criptomonedasha generado nuevos espacios para la comisión de fraudes en entornos digitales (Singh, 2025).

Uno de los tipos más comunes es el fraude con tarjeta de crédito o débito, donde los delincuentes obtienen información de las tarjetas mediante técnicas como skimming, phishing o brechas de seguridad en plataformas de pago. Posteriormente, esa información se utiliza para realizar compras o transferencias no autorizadas (Martínez Pazos et al., 2023). En algunos casos,

los defraudadores crean tarjetas clonadas o emplean los datos en sitios de comercio electrónico que no exigen verificación adicional, aprovechando la falta de autenticación multifactor (Talafha, 2024).

Otro tipo relevante es el fraude por suplantación de identidad, en el cual el delincuente se hace pasar por un usuario legítimo utilizando datos personales robados o falsificados. Este tipo de fraude suele ir acompañado de ataques de ingeniería social, como el phishing o el vishing, donde las víctimas son persuadidas para entregar voluntariamente sus credenciales o códigos de verificación (The Payments Association, s.f.). La sofisticación de estos ataques ha aumentado con el uso de inteligencia artificial y deepfakes, que permiten crear mensajes, voces o incluso rostros falsos con apariencia auténtica (Fariha et al., 2025).

También destaca el fraude en comercio electrónico o "card-not-present" (CNP), que ocurre cuando una transacción se realiza sin la presencia física de la tarjeta, como en compras en línea. Este tipo de fraude representa una proporción significativa de las pérdidas globales en pagos digitales, debido a que resulta más difícil verificar la autenticidad del comprador (Vanini et al., 2023). Las medidas de seguridad tradicionales, como la verificación del código CVV o la dirección de facturación, resultan insuficientes frente a delincuentes que disponen de datos completos de la víctima obtenidos mediante filtraciones o venta de información en la dark web (Singh, 2025).

Por otra parte, se encuentra el fraude en transferencias electrónicas, donde los atacantes interceptan o manipulan los procesos de pago entre empresas o individuos. En este contexto, los fraudes de tipo Business Email Compromise (BEC) se han vuelto frecuentes: los delincuentes falsifican correos electrónicos corporativos para engañar a empleados y lograr que realicen transferencias hacia cuentas bajo su control (Talafha, 2024). Estas prácticas han ocasionado

pérdidas multimillonarias a nivel mundial y afectan tanto a empresas grandes como a pequeños comercios.

En los últimos años ha emergido además el fraude en billeteras digitales y aplicaciones móviles, impulsado por la popularización de plataformas como PayPal, Apple Pay o Google Pay. Los atacantes aprovechan vulnerabilidades en la verificación de identidad o engañan a los usuarios para que autoricen pagos hacia cuentas fraudulentas. Este tipo de fraude suele combinar ingeniería social con manipulación de dispositivos móviles, lo que dificulta su rastreo (Fariha et al., 2025).

Por último, el auge de las criptomonedas y los activos digitales ha dado origen al fraude en criptoactivos, caracterizado por esquemas de inversión falsos, robos de claves privadas o plataformas de intercambio fraudulentas. Estos fraudes aprovechan la descentralización y el anonimato del sistema para evadir controles tradicionales, dificultando la recuperación de fondos (Singh, 2025). A diferencia de los fraudes convencionales, aquí la falta de intermediarios financieros tradicionales reduce la capacidad de reacción ante una estafa o transacción ilícita.

Frente a estos riesgos, la literatura académica destaca la importancia de combinar estrategias de prevención tecnológica como autenticación multifactor, tokenización y análisis de comportamiento con la educación financiera de los usuarios, a fin de reducir la exposición al fraude digital (Talafha, 2024). Además, la cooperación entre bancos, proveedores de tecnología y autoridades regulatorias es esencial para establecer protocolos uniformes que permitan detectar, denunciar y mitigar los distintos tipos de fraude en los pagos en línea (The Payments Association, s.f.).

#### 2.2.3.2 Impacto económico y social del fraude

El desarrollo tecnológico en el sector bancario también trae consigo nuevos desafíos para la estabilidad y seguridad del sistema financiero. Entre ellos destaca el fraude digital, ya que los actores malintencionados están utilizando las herramientas de la digitalización para ejecutar

fraudes en línea con mayor alcance y frecuencia que en el pasado. Incluso con limitaciones y vacíos en los datos disponibles, la digitalización facilita que los estafadores operen con mayor rapidez y flexibilidad (Carter, 2025).

De acuerdo con Oladele et al. (2025), en los últimos años el fraude financiero ha experimentado un crecimiento notable, generando un impacto económico considerable reflejado en pérdidas anuales que ascienden a miles de millones de dólares a nivel global. Este incremento no solo responde a la sofisticación de las técnicas de fraude, sino también a la mayor dependencia de los servicios financieros digitales.

La evidencia internacional coincide con lo observado en el contexto ecuatoriano. El estudio de Maldonado Gudiño et al. (2024) muestra que, entre 2021 y 2023, los casos de fraude digital aumentaron de 2.400 a 3.000 incidentes anuales, lo que significó pérdidas económicas que alcanzaron los 10 millones de dólares en 2023 (p. 10). Además, del total de transacciones electrónicas realizadas en el país, el 8,08 % correspondió a transacciones fraudulentas, con un promedio mensual de 250 casos y pérdidas aproximadas de 833.333 dólares mensuales (p. 11). Estos datos evidencian que el fraude no constituye un fenómeno aislado, sino un problema estructural con repercusiones económicas sostenidas.

Asimismo, distintos estudios señalan que las consecuencias del fraude financiero no solo recaen en las víctimas directas, como consumidores y entidades financieras, sino que también afectan al funcionamiento de la economía en su conjunto. Estos impactos indirectos se manifiestan en el aumento de los costos operativos, la necesidad de implementar sistemas de seguridad cada vez más sofisticados y la disminución de la confianza pública en los servicios financieros digitales (Oladele et al., 2025). En el caso ecuatoriano, la tendencia al alza en el uso de transacciones electrónicas que crecieron un 5,30 % entre 2022 y 2023 implica también un aumento en la

exposición al fraude y en los costos asociados a su mitigación (Maldonado Gudiño et al., 2024, p. 9).

En conjunto, esta evidencia demuestra que el fraude digital representa una carga económica significativa tanto a nivel global como nacional, afectando las finanzas de los usuarios, la estabilidad operativa de las entidades financieras y, en última instancia, la confianza en el ecosistema financiero digital.

#### 2.2.3.2.1 Impacto Social

La dimensión social del fraude es igualmente crítica: además de las afectaciones económicas, los usuarios experimentan vulnerabilidad, desconfianza en la banca digital y procesos de exclusión financiera, tal como lo señalan diversos estudios que analizan la relación entre fraude y participación en servicios digitales (Ozili, 2024). Asimismo, el fraude deteriora la experiencia de uso de servicios electrónicos, debido a la percepción de inseguridad y al temor de los clientes de utilizar plataformas de pagos o banca en línea (Singh, 2025). Estos impactos, documentados ampliamente en diferentes contextos internacionales (Carter, 2025; Oladele et al., 2025), permiten establecer un marco de referencia comparativo para comprender cómo fenómenos similares pueden manifestarse en realidades específicas, como la del Ecuador, donde la digitalización de los pagos y la banca electrónica continúa en expansión y, con ello, aumentan también los riesgos asociados a actividades fraudulentas (Umoh, 2024; Singh, 2025).

En el caso ecuatoriano, el estudio de Maldonado Gudiño et al. (2024) evidencia que el fraude financiero digital no solo afecta a las instituciones, sino que tiene repercusiones directas en la vida de los usuarios. El artículo señala que delitos como el phishing, el robo de identidad y las transacciones no autorizadas generan consecuencias graves para las víctimas, tales como daños a su historial crediticio, pérdida de control sobre su información personal y la necesidad de invertir tiempo y recursos para revertir los efectos del fraude (p. 5). Estas afectaciones no son únicamente

financieras: también implican un fuerte componente emocional y psicológico, traducido en miedo, desconfianza y resistencia a seguir utilizando servicios digitales.

Además, la falta de conocimientos de ciberseguridad entre los usuarios ecuatorianos incrementa su vulnerabilidad social. El mismo estudio resalta la necesidad de campañas de capacitación y concientización que permitan a los usuarios identificar riesgos y adoptar prácticas seguras, con el objetivo de reducir la exposición a fraudes digitales (Maldonado Gudiño et al., 2024, p. 14). La importancia de estas acciones radica en que, sin una alfabetización digital adecuada, grupos vulnerables como adultos mayores, personas con baja educación financiera o con acceso limitado a tecnología, generan mayor riesgo de ser víctimas recurrentes.

De esta manera, el impacto social del fraude digital en Ecuador se manifiesta en múltiples niveles: afecta la confianza, incrementa la sensación de inseguridad, limita la adopción de canales bancarios digitales y profundiza brechas sociales vinculadas a la educación tecnológica. Esta realidad, coherente con la evidencia internacional, subraya la necesidad de abordar el fraude no solo como un problema técnico o económico, sino también como un fenómeno que afecta directamente la vida cotidiana y el bienestar de los ciudadanos.

#### 2.2.3.3 Estrategias tradicionales de detección

Históricamente, la detección del fraude en el sector financiero se ha apoyado en una serie de mecanismos tradicionales que anteceden al uso de algoritmos avanzados de aprendizaje automático. Estos métodos surgieron como respuestas iniciales para enfrentar modalidades de fraude relativamente estáticas, cuando la mayoría de las operaciones bancarias se realizaban de manera presencial o mediante canales digitales básicos. En el contexto ecuatoriano, diversas instituciones financieras han dependido principalmente de controles internos, validaciones manuales, revisiones documentales, procedimientos de debida diligencia (KYC) y monitoreo por parte de operadores

humanos para identificar comportamientos anómalos o transacciones sospechosas (Maldonado Gudiño et al., 2024). Estas estrategias, aunque fundamentales en su momento, estaban diseñadas para estructuras operativas mucho menos dinámicas que las actuales.

La literatura internacional coincide en que las primeras aproximaciones para la detección de fraude se basaban en sistemas de reglas estáticas, umbrales de montos, listas negras y blancas, y modelos estadísticos tradicionales como análisis de outliers o puntuaciones basadas en desviaciones estándar (Bolton & Hand, 2002). Asimismo, muchos procesos dependían de la experiencia del personal encargado, quienes evaluaban de forma manual comportamientos inusuales o inconsistencias en la información proporcionada por los usuarios (Delamaire et al., 2009). En estos esquemas, una transacción era marcada como sospechosa únicamente si coincidía con un patrón previamente establecido, lo que volvía al sistema reactivo y limitado frente a nuevas modalidades de fraude.

Un análisis más amplio presentado por Phua et al. (2010) señala que esta dependencia casi exclusiva de reglas definidas por expertos impedía que los sistemas tradicionales reconocieran patrones emergentes. En otras palabras, solo se detectaba el fraude ya conocido, mientras que las técnicas innovadoras pasaban desapercibidas. Esto resultaba especialmente problemático en un entorno donde los estafadores modifican constantemente sus estrategias.

El estudio de Phiri et al. (2024) aporta una perspectiva contemporánea relevante, al mostrar que gran parte del fraude en línea actual se origina mediante técnicas como phishing, smishing o ingeniería social. Estos métodos no buscan vulnerar directamente la infraestructura bancaria, sino explotar la confianza del usuario y las debilidades de los procesos manuales de autenticación. El análisis de entrevistas a víctimas en Sudáfrica y España evidencia que los métodos tradicionales de verificación, basados en contraseñas, preguntas de seguridad o validaciones humanas, siguen siendo

insuficientes para prevenir este tipo de fraudes (Phiri et al., 2024, pp. 3–5). Además, los autores identifican la persistencia de "sistemas heredados" en ciertas entidades financieras, los cuales presentan limitaciones frente a amenazas dinámicas y sofisticadas (Phiri et al., 2024, p. 5).

Estas observaciones coinciden con los hallazgos del estudio publicado en *Scientific Reports*, donde se detalla que los métodos tradicionales presentan alta tasa de falsos positivos, baja adaptabilidad, y dificultad para procesar grandes volúmenes de datos, características que comprometen su efectividad en escenarios digitales modernos (Zhao et al., 2025). Este trabajo subraya que los sistemas basados en reglas fijas son incapaces de ajustarse a modalidades nuevas de fraude, que pueden evolucionar en cuestión de horas.

Por otra parte, la investigación de Phiri et al. (2024) también evidencia que el componente humano continúa siendo un punto crítico en la cadena de seguridad. La falta de capacitación, la sobrecarga operativa o la confianza indebida en comunicaciones fraudulentas contribuyen a que el fraude se concrete. Este tipo de vulnerabilidad humana, que los sistemas tradicionales no pueden mitigar completamente, representa un límite estructural de los métodos clásicos de detección.

En conjunto, la evidencia internacional y regional indica que, si bien las estrategias tradicionales establecieron las bases de los sistemas antifraude, sus limitaciones son evidentes en un contexto digital caracterizado por altos volúmenes transaccionales, amenazas dinámicas y modalidades de fraude cada vez más complejas. Los controles manuales no escalan al ritmo de las transacciones en línea; los sistemas de reglas rígidas no identifican patrones emergentes; la experiencia humana no basta para analizar millones de operaciones diarias; y la autenticación tradicional es vulnerable a ataques de ingeniería social. Debido a ello, se reconoce ampliamente la necesidad de migrar hacia enfoques basados en análisis automatizado, modelos adaptativos y

técnicas modernas de aprendizaje automático, que permitan una detección temprana, eficiente y continua del fraude en entornos bancarios digitales.

### *2.2.4 Aprendizaje automático en la detección de fraude*

El crecimiento acelerado de las transacciones digitales y la sofisticación de los mecanismos de fraude han impulsado la adopción de métodos de aprendizaje automático (Machine Learning, ML) como una herramienta fundamental para fortalecer los sistemas de detección temprana en instituciones financieras. A diferencia de los enfoques tradicionales, basados en reglas estáticas, listas negras o revisiones manuales, los modelos de ML son capaces de aprender patrones complejos**,** identificar relaciones no evidentes y adaptarse a nuevas modalidades de fraude con mayor rapidez (Ngai et al., 2011). Esta capacidad adaptativa resulta esencial en un entorno donde los atacantes modifican constantemente sus tácticas para evadir medidas de seguridad predefinidas.

Diversos estudios coinciden en que el aprendizaje automático ha demostrado un rendimiento superior en la detección de fraude financiero debido a su habilidad para procesar grandes volúmenes de datos, analizar múltiples características simultáneamente y manejar la naturaleza altamente desbalanceada de los datasets de fraude, donde las transacciones fraudulentas representan una fracción mínima del total (Al-Hashedi & Magalingam, 2021). Para abordar este desbalance, los modelos suelen complementarse con técnicas como SMOTE, selección de características o métodos de penalización de costos, los cuales permiten mejorar la sensibilidad sin incrementar los falsos positivos (Dal Pozzolo et al., 2015).

En la misma línea, Compagnino (2025) destaca que el ML se ha consolidado como un componente central en los sistemas modernos de prevención del fraude, particularmente por su flexibilidad y por su capacidad para generar predicciones en tiempo real. Según este autor, los algoritmos supervisados como: "*Random Forest*, *Gradient Boosting*, *Support Vector Machines* y *Redes Neuronales Artificiales*" han mostrado un desempeño sobresaliente al modelar patrones

transaccionales normales y comparar nuevas operaciones con comportamientos previos. Esto permite identificar desviaciones sutiles que serían prácticamente invisibles para los sistemas tradicionales.

Una de las ventajas más significativas del aprendizaje automático es su capacidad para mejorar continuamente mediante el análisis de datos históricos y retroalimentación de casos confirmados. En estudios recientes, Wickramanayake et al. (2020) evidencian que modelos como *XGBoost* y *Random Forest* alcanzan altos niveles de precisión y recall en la detección de fraude en pagos en línea, gracias al uso de estructuras en conjunto (*ensemble learning*) que reducen el sobreajuste y mejoran la estabilidad del modelo. Estos modelos también permiten evaluar la importancia relativa de cada variable, aportando información valiosa sobre los factores de riesgo más críticos, aspecto especialmente relevante para instituciones que buscan estrategias preventivas más efectivas.

Por otra parte, investigaciones recientes en el ámbito del deep learning han ampliado el alcance del ML hacia estructuras más complejas. Hernández Aros et al. (2024) explican que técnicas como redes neuronales profundas, autoencoders y LSTM pueden capturar patrones altamente no lineales y dependencias temporales en secuencias de transacciones, lo que resulta especialmente útil para detectar fraudes encadenados o conductas fraudulentas que evolucionan con el tiempo. Aunque estas arquitecturas suelen requerir mayor capacidad computacional y bases de datos más extensas, ofrecen ventajas significativas en escenarios donde la dinámica del fraude es rápida y difícil de modelar mediante métodos tradicionales o algoritmos lineales.

Además, un estudio reciente publicado en *Scientific Reports* subraya que los modelos de ML no solo superan a los métodos tradicionales en rendimiento, sino también en capacidad de adaptación. Según Zhao et al. (2025), los algoritmos basados en aprendizaje automático pueden ajustar automáticamente sus parámetros a medida que cambian los patrones de comportamiento, lo cual reduce significativamente los falsos positivos y mejora la detección de fraudes novedosos. Este

aspecto es clave para la banca digital moderna, donde los ciberdelincuentes introducen variaciones en sus tácticas en intervalos de tiempo muy cortos.

Reportes especializados como el *Financial and Cyber Fraud Report 2024* de Grant Thornton resaltan que los sistemas basados en inteligencia artificial y aprendizaje automático se han convertido en el estándar de la industria, debido a la creciente complejidad de los ataques y al volumen exponencial de transacciones digitales. El informe enfatiza que la automatización y la analítica avanzada permiten identificar patrones que serían imperceptibles para analistas humanos y que resultan esenciales para evitar pérdidas económicas sustanciales (Grant Thornton, 2024).

Finalmente, el aprendizaje automático representa un cambio de paradigma en la detección de fraude financiero, ofreciendo mecanismos más robustos, adaptativos y precisos que los enfoques tradicionales. Su capacidad para aprender, generalizar y anticiparse a nuevas tácticas delictivas lo convierte en una herramienta estratégica para instituciones que buscan fortalecer sus sistemas de seguridad y garantizar la confianza en los servicios de banca digital.

#### 2.2.4.1 Conceptos básicos de Machine Learning y Deep Learning

El Machine Learning (ML) constituye una de las herramientas más relevantes dentro del campo de la detección automatizada de fraude financiero. Su premisa fundamental radica en que los modelos pueden aprender patrones directamente de los datos, sin necesidad de ser programados explícitamente para cada situación. En palabras de Jordan y Mitchell (2015), el ML permite que un algoritmo mejore su desempeño conforme aumenta la experiencia obtenida a partir de nuevos datos. Esta cualidad resulta especialmente pertinente en el entorno financiero, donde los esquemas de fraude evolucionan con rapidez y las reglas rígidas son incapaces de capturar patrones infinitamente variables.

Los modelos tradicionales de ML empleados en la detección de fraude incluyen algoritmos como “Árboles de Decisión, Support Vector Machines (SVM), Regresión Logística y Random Forest”, todos ellos diseñados para clasificar transacciones como normales o fraudulentas en función de características previamente observadas. Según Fu et al. (2025), estos modelos permiten identificar relaciones relevantes entre indicadores financieros, variables no financieras e incluso patrones textuales, lo que facilita una visión más amplia del comportamiento fraudulento. Además, el estudio destaca el papel de técnicas de preprocesamiento como Borderline-SMOTE, indispensable para tratar el desbalance severo entre transacciones legítimas y fraudulentas que constituye una característica muy común en los datasets financieros (pp. 980–982) .

El aprendizaje automático puede dividirse principalmente en dos paradigmas: supervisado y no supervisado. En el aprendizaje supervisado, los modelos se entrenan con ejemplos etiquetados, lo que permite aprender una función que discrimine entre clases con base en patrones históricos.

Esto facilita la clasificación de nuevas transacciones con base en comportamientos previamente observados. El aprendizaje no supervisado, en cambio, es ideal para detectar anomalías en escenarios donde el fraude no se encuentra completamente identificado o etiquetado. Modelos como One-Class SVM, estudiados por Fu et al. (2025), permiten encontrar desviaciones respecto del comportamiento normal, facilitando la detección de fraudes emergentes o sofisticados (pp. 983–984)

El Deep Learning (DL) representa una evolución del ML tradicional. Este enfoque emplea redes neuronales profundas, capaces de aprender representaciones complejas de los datos a través de múltiples capas. Una de sus principales ventajas es su capacidad para capturar relaciones altamente no lineales sin necesidad de una ingeniería intensiva de características. Hernández Aros et al. (2024) sostienen que el DL es especialmente eficaz en contextos de fraude donde los patrones

son dinámicos, temporales o difíciles de expresar mediante reglas simples. Dentro de las arquitecturas más destacadas se encuentran las Redes Neuronales Artificiales (ANN), las redes LSTM para secuencias temporales, y los Autoencoders, ampliamente utilizados para la detección de anomalías mediante reconstrucción (pp. 3–5) .

Los autoencoders, específicamente, aprenden a comprimir las transacciones normales en un espacio latente y reconstruirlas con precisión. Cuando una transacción fraudulenta se introduce en el modelo, la reconstrucción presenta un error significativamente mayor, lo que permite identificar actividades sospechosas aun sin disponer de etiquetas. Este enfoque ha cobrado especial relevancia en la banca digital debido a la creciente complejidad y volumen de las transacciones en tiempo real (Hernández Aros et al., 2024).

Por otra parte, Compagnino (2025) destaca que tanto ML como DL permiten desarrollar sistemas que *mejoran continuamente*, ya que pueden actualizarse con nuevos datos y adaptarse a modalidades cambiantes de fraude. Esto contrasta con los métodos tradicionales basados en reglas, que suelen quedar obsoletos ante nuevas estrategias criminales. Además, el autor señala que modelos como Random Forest, Gradient Boosting y diversas arquitecturas neuronales han mostrado un desempeño notable en la detección de patrones complejos en escenarios financieros (pp. 2–4).

En suma, tanto el Machine Learning como el Deep Learning constituyen herramientas esenciales dentro de la prevención y detección temprana del fraude financiero. Los algoritmos supervisados y no supervisados permiten abordar distintos tipos de problemas partiendo desde la clasificación directa hasta la detección de anomalías; mientras que las redes profundas capturan relaciones sutiles y complejas. Estas características hacen que ML y DL superen ampliamente las limitaciones de los enfoques tradicionales, proporcionando mayor adaptabilidad, precisión y capacidad de respuesta ante un panorama delictivo en constante transformación.

#### 2.2.4.2 Modelos supervisados aplicados a la detección de fraude

##### I. Random Forest

La detección de fraude en pagos digitales requiere modelos capaces de procesar grandes volúmenes de información, manejar desbalances extremos entre clases y captar relaciones no lineales entre variables. En este contexto, Random Forest (RF) se ha consolidado como uno de los algoritmos más eficaces y confiables dentro del aprendizaje automático aplicado al sector financiero. Su uso generalizado en transacciones electrónicas, banca digital y fraude con tarjetas de crédito se respalda tanto en evidencia internacional como en estudios realizados en Ecuador.

A nivel global, la revisión sistemática de Compagnino et al. (2025) identifica a Random Forest como el modelo supervisado más utilizado para la detección de fraude financiero. Los autores señalan que RF alcanza frecuentemente precisiones superiores al 95 % y muestra mayor estabilidad y capacidad de generalización que algoritmos como SVM, k-NN, Árboles de Decisión simples y regresión logística. Esto se debe a su arquitectura basada en múltiples árboles entrenados sobre subconjuntos aleatorios de datos y características, lo que reduce el sobreajuste y permite capturar interacciones complejas entre atributos.

La revisión de Hernández Aros et al. (2024), publicada en *Humanities and Social Sciences Communications* (Nature), confirma este panorama al ubicar a Random Forest entre los algoritmos más robustos y recurrentes en estudios de fraude bancario, crediticio y contable. El análisis destaca que RF mantiene un desempeño competitivo incluso cuando la clase fraudulenta representa menos del 1 % del total de transacciones, condición típica del fraude en pagos en línea. Además, los autores señalan que RF conserva altos niveles de *recall*, fundamentales para evitar falsos negativos, cuando se combina con técnicas de sobremuestreo o metodologías sensibles al costo.

La evidencia nacional refuerza estos hallazgos. En la investigación desarrollada por Llerena (2024) en la USFQ, utilizando información financiera real de CACPECO, Random Forest obtuvo

el mejor desempeño entre los modelos evaluados. El algoritmo alcanzó un $R^2 = 0{,}94$, superando ampliamente a la regresión logística (0.48), y presentando menor error cuadrático (MSE = 0.59) y menor error estándar (RMSE = 0.72). Estos resultados demuestran que RF es capaz de capturar patrones complejos en datos financieros reales, incluso cuando existe ruido, alta variabilidad y correlaciones entre atributos.

Lituma Perero et al. (2024) desarrollaron un estudio enfocado en la detección de fraude en tarjetas de crédito que aporta evidencia clara sobre el buen desempeño del modelo Random Forest en este tipo de problemas. Para ello trabajaron con un conjunto de más de catorce mil transacciones y aplicaron un proceso de preparación bastante completo: limpiaron registros duplicados, construyeron nuevas variables a partir de fechas y categorías, identificaron las características más relevantes y trataron el desbalance del conjunto de datos mediante SMOTE. Con esta base entrenaron un Random Forest que ajustaron utilizando RandomizedSearchCV, obteniendo parámetros como 200 árboles, profundidad máxima de 12 y criterio *entropy*. Al evaluar el modelo con métricas habituales en clasificación accuracy, precisión, recall, F1 y AUC, el desempeño fue especialmente alto: un accuracy cercano al 98.8 %, un AUC de 0.97 y un número reducido de errores (20 falsos negativos y 15 falsos positivos). Además, mediante el análisis de interpretabilidad (SHAP) identificaron que variables como el monto, la hora de la transacción y la categoría del comercio influyen con mayor peso en la detección de fraude. En conjunto, el estudio muestra que Random Forest es un modelo sólido y confiable para manejar datos financieros complejos y apoyar la detección temprana de operaciones fraudulentas (Lituma Perero et al., 2024).

DDesde la perspectiva técnica, Random Forest es particularmente apropiado para detectar fraudes en pagos en línea porque puede modelar las complejas relaciones no lineales que existen entre los patrones de conducta del usuario, el monto de la transacción, la geolocalización, el tipo de dispositivo y el canal de pago. Asimismo, exhibe una alta resistencia frente al ruido y a los

valores atípicos, cualidades comunes en datos transaccionales auténticos. El algoritmo tiene un buen desempeño en entornos de alta dimensionalidad, porque es capaz de manejar con eficacia un amplio volumen de variables, incluyendo las que se crean a través de procesos de ingeniería de características. En la construcción de los árboles, el muestreo aleatorio de las características y de los ejemplos disminuye el peligro del sobreajuste, aumentando así la habilidad del modelo para generalizar. En última instancia, Random Forest proporciona un nivel de interpretabilidad adecuado al calcular la importancia de las variables, lo cual es crucial en entornos financieros que exigen justificar las decisiones del modelo y observar regulaciones sobre transparencia y explicabilidad.

Una ventaja adicional es que Random Forest se adapta de forma natural a estrategias para tratar el desbalance de clases, como SMOTE, submuestreo o ajuste de pesos. Esto permite aumentar la detección de la clase fraudulenta sin incrementar excesivamente los falsos positivos, un requisito esencial en sistemas de monitoreo automático de la banca.

En conjunto, se muestra de manera consistente que Random Forest no solo es un modelo eficaz, sino también uno de los más equilibrados y confiables para sistemas de detección de fraude en pagos digitales. Su combinación de rendimiento, estabilidad, capacidad explicativa y adaptabilidad a escenarios reales lo convierte en una elección plenamente justificada para un portal web de detección temprana de fraude bancario.

## II. Árboles de Decisión

De acuerdo con (Shah & Sharma, 2025), un árbol de decisión es un método de aprendizaje supervisado que se utiliza en el aprendizaje automático para predecir resultados a partir de características de entrada. Este se crea separando los datos en subconjuntos de forma reiterada, de acuerdo con los valores de las características. Cada división se escoge para maximizar la separación utilizando métricas como el índice de Gini o la ganancia de información. El procedimiento termina

cuando se cumplen ciertas condiciones, por ejemplo, si un grupo tiene un número reducido de puntos de datos y no puede dividirse más o si se alcanza una profundidad de árbol determinada.

Los árboles de decisión se construyen comenzando por la raíz y avanzando hacia las hojas. Para definir cada división del árbol se usan los atributos que describen a los ejemplos, ya que las reglas de clasificación se obtienen justamente a partir de estas características. Las hojas representan las clases y los nodos intermedios corresponden a pruebas basadas en atributos. Clasificar un objeto consiste en recorrer el árbol desde la raíz, siguiendo las ramas según los valores del objeto hasta llegar a una hoja. Un árbol que clasifica correctamente todos los ejemplos del conjunto de entrenamiento siempre puede construirse cuando los atributos son suficientes, y normalmente existen varias alternativas válidas (Quinlan, 1986).

Los ejemplos que alimentan el modelo pueden venir de dos fuentes, el primero son los datos históricos que ya existen en bases de datos reales cuya información da una idea general y confiable del comportamiento de los casos, aunque suele repetir información y no siempre incluye situaciones poco frecuentes. La segunda consiste en trabajar con ejemplos preparados por expertos, seleccionados de manera intencional para representar los casos más comunes y también aquellos que casi no ocurren. Aunque los métodos de inducción de árboles funcionan con cualquiera de los dos tipos de conjuntos, inicialmente fueron pensados para utilizar datos históricos. Con el tiempo, sin embargo, también se han usado de manera habitual conjuntos creados por expertos (Quinlan, 1986).

a. Métodos de construcción de árboles de decisión

Quinlan describe como el algoritmo Iterative Dichotomizer 3 (ID3) consiste en elegir, en cada paso, el atributo que mejor ayuda a separar las clases. Para decidir cuál es ese atributo, calcula cuánto se reduce la incertidumbre al dividir los datos según cada uno. El que ofrezca la mayor reducción es el que se usa para crear el siguiente nodo del árbol.

Mientras que el autor, i Solé define el método ID3 como aquel que se basa en dividir el conjunto de datos paso a paso, buscando en cada iteración la partición que mejor separe las clases. El algoritmo continúa generando estas divisiones mientras sigan existiendo atributos útiles y hasta que se alcance un punto donde los grupos formados sean lo más homogéneos posible, garantizando así una buena capacidad predictiva (i Solé, 1995).

Una vez seleccionado el atributo, el conjunto de ejemplos se divide según sus valores, y el proceso se repite dentro de cada grupo. Esto continúa hasta que los ejemplos de un nodo pertenecen todos a la misma clase o ya no quedan más atributos disponibles. En esos casos, el nodo se convierte en una hoja (i Solé, 1995).

Quinlan también comenta que este procedimiento implica volver a revisar los datos varias veces, porque en cada nodo se necesita calcular la ganancia de información de los atributos restantes. Aunque esto aumenta el costo computacional, él señala que, en la práctica, el método sigue siendo manejable incluso con conjuntos de datos relativamente grandes.

En relación con la homogeneidad, existen diversas medidas cuyo propósito es asignar valores extremos cuando una partición está compuesta únicamente por ejemplos de una misma clase. Estas permiten evaluar particiones que no son completamente uniformes, ya que su grado de diversidad interna queda expresado numéricamente, lo que facilita la comparación entre diferentes divisiones (i Solé, 1995).

La medida más utilizada para cuantificar el desorden es la entropía, que proviene de la teoría de la información y que se basa en la distribución de probabilidades de las clases. La entropía puede interpretarse como una medida de información porque está vinculada con el nivel de “sorpresa” que produce un determinado valor: cuanto menos previsible es un resultado, mayor es la información que aporta. Se llega a lo que es la ganancia de información donde en la construcción del árbol, en cada etapa no se busca simplemente el atributo que contenga más información, sino aquel que genere

la mayor diferencia de información con respecto a la partición actual al dividir los datos según sus valores. Es por ello por lo que en cada paso se selecciona el atributo que optimice esta ganancia (i Solé, 1995).

La ganancia de información se define de la siguiente manera:

$$G(X, A_k) = I(X, C) - E(X, A_k) \quad (1)$$

Donde:

- $I(X, C)$: es la cantidad de información asociada a las particiones generadas por un conjunto de clases $C$ con respecto al conjunto de casos $X$ y se define de la siguiente forma:

$$I(X, C) = -\sum_{C_i \in C} p(X, c_i) \log_2 p(X, c_i) \quad (2)$$

Donde:

- $p(X, c_i)$: es la probabilidad de que un ejemplo específico $c_i$, que se aproxima mediante la frecuencia observada de casos que pertenecen a la clase $c_i$, utilizándose esta frecuencia como estimador de la probabilidad:

$$p(X, c_i) = \frac{\#c_i}{\#X} \quad (3)$$

Donde el término $p(X, c_i)$ representa la proporción de casos pertenecientes a la clase $\#c_i$ respecto al tamaño total de la muestra $\#X$.

En cuanto a $E(X, A_k)$, representa la información esperada del atributo $A_k$ respecto al conjunto de casos X. Refleja el grado de diversidad que presenta este atributo dentro del conjunto $X$. Mide la diversidad que se introduce en las particiones al seleccionar el atributo $A_k$. Su expresión es:

$$E(X, A_k) = \sum_{V_I \in V(A_k)} p\left(X, A_k^{-1}(V_i)\right) \cdot IA_k^{-1}(V_i), C) \quad (4)$$

Donde $p\left(X, A_k^{-1}(V_i)\right)$ es la probabilidad de que un caso presente el valor Vi en el atributo $A_k$ que suele aproximar a partir de las frecuencias observadas mediante la siguiente expresión:

$$p\left(X, A_k^{-1}(V_i)\right) = \frac{\#A_k^{-1}(V_i)}{\#X} \quad (5)$$

Es decir, calcula el número de casos que muestran el valor $V_i$, en el atributo $A_k$ en relación con el total de casos del conjunto $X$.

b. Métodos de Poda

El método de poda intenta obtener particiones que sólo sean necesarias para obtener una buena predicción y sean más fáciles de interpretar. Hay dos métodos: C4.5 que se basa en estimar la tasa de error para cada subárbol y reemplazarlo con el nodo hoja si la estimación del error hoja es menor. La idea básica es que la determinación de la tasa de error para cualquier nodo del árbol, incluidos los nodos de hoja, comenzará desde los niveles más bajos del árbol, y si las estimaciones muestran que la precisión general mejora al eliminar $n$ de los hijos del nodo y convertir $n$ en una hoja, entonces C4.5 realiza esta poda. En la práctica, aunque estas estimaciones son aproximadas, el método suele ser eficaz (Salzberg, 1994).

El segundo método de poda es MDL, también conocido como método de mínima descripción de longitud. Se seleccionan pequeños subconjuntos de reglas que garanticen que clase $C$ esté representada en los datos. El objetivo es mantener el conjunto de reglas lo más simple posible, evitando la duplicación y la complejidad excesiva, pero sin perder la capacidad predictiva. Este enfoque se complementa con varias estrategias algorítmicas (Salzberg, 1994).

c. Método de CART

El método CART, propuesto por Breiman y sus colegas en 1984, es un algoritmo para construir árboles de decisión cuyo nombre proviene de Classification and Regression Trees. Se caracteriza por generar árboles binarios, de modo en que cada nodo se define un punto de corte que divide el conjunto de observaciones en dos grupos (i Solé, 1995).

Una ventaja importante es que puede trabajar con atributos continuos y además permite abordar tanto problemas de clasificación cuando la variable objetivo es categórica como problemas de regresión. Para la división de datos, CART utiliza el índice de Gini como medida de diversidad que consiste en encontrar el atributo y el punto de corte que logren la mayor reducción de esta diversidad (i Solé, 1995).

Una vez elegido el mejor separador, este se convierte en un nodo del árbol y el proceso se repite con cada una de las particiones resultantes. Si en algún momento un atributo deja de aportar información este se descarta. Cuando ya no es posible realizar más divisiones útiles, el nodo se convierte en hoja. El árbol se considera completo cuando todas las particiones han llegado a hojas terminales (i Solé, 1995).

d. Ventajas e Inconvenientes

En el ámbito de la detección de fraude financiero, los árboles de decisión destacan principalmente por su interpretabilidad. A diferencia de los modelos como las redes neuronales profundas, esta metodología permite rastrear la lógica exacta detrás de cada predicción. Esto es crucial en el sector bancario, donde a menudo es obligatorio justificar ante el cliente o los reguladores por qué una transacción fue marcada como sospechosa. Asimismo, estos modelos ofrecen métricas intrínsecas sobre la importancia de los atributos, permitiendo identificar qué variables como el monto o la hora de la transacción, son determinantes para la clasificación.

Sin embargo, el método no está exento de problemas. El principal inconveniente es su tendencia a sobreajustarse (overfitting) que, en caso de no controlarse, los árboles pueden crecer demasiado y

normalizar el ruido de los datos de entrenamiento en lugar de aprender patrones generalizables. Además, son sensibles a pequeñas variaciones en los datos de entrada, lo que puede producir estructuras de árbol muy diferentes con cambios mínimos en los datos (i Solé, 1995).

## III. XGBoost

### a. Introducción al Modelo XGBoost

El modelo XGBoost es un algoritmo de aprendizaje supervisado basado en el método de boosting por gradiente el cual construye de manera secuencial un conjunto de árboles de decisión (CART) para de esta manera poder optimizar una función de pérdida y añadir regularización que evita el sobreajuste (Chen & Guestrin, 2016). En esencia, cada nuevo árbol corrige los errores del conjunto de árboles anteriores, de modo que el modelo mejora progresivamente su capacidad de predicción.

Su arquitectura incorpora, además, mecanismos de regularización (por ejemplo, L1 y L2), poda de árboles, manejo eficiente de datos faltantes y paralelización, lo que lo hace altamente competitivo con grandes volúmenes de datos estructurados y con relaciones no lineales.

El hecho de que XGBoost pueda manejar bien datos heterogéneos, con muchas variables y con relaciones de interacción complejas, lo hace especialmente atractivo para tareas de clasificación en los que los patrones legítimos y fraudulentos pueden diferir en formas sutiles.

### b. Particularidades del fraude en pagos en línea y uso de XGBoost

La detección de fraude en el ámbito de los pagos online se ve dificultada por varios aspectos intrínsecos al problema, como la gran disparidad entre las conductas de los usuarios fraudulentos y las legítimas, el marcado desequilibrio entre clases (en el que las transacciones fraudulentas suelen ser menos del 1 % del total) y la necesidad de hacer detecciones en tiempo real o casi en tiempo real para reducir pérdidas económicas (Velarde et al., 2023). En este contexto, debido a su habilidad para

detectar interacciones sofisticadas entre variables, como la hora de la transacción, el historial del usuario, el monto y la ubicación sin necesidad de una especificación manual, el modelo XGBoost es especialmente adecuado; a su robustez ante datos tabulares de alta dimensión, que es una propiedad típica en los registros transaccionales; a su eficacia en términos computacionales y a su habilidad para ser paralelizado, lo cual permite que pueda aplicarse en ambientes con grandes cantidades de información y demandas de respuestas veloces; y a su capacidad para tratar el problema del desequilibrio de clases usando técnicas como la ponderación de clases, el muestreo sub o sobre muestreo, disminuyendo el sesgo hacia la clase mayoritaria (Velarde et al., 2023).

Por ejemplo, estudios han observado que XGBoost alcanza altos valores de métrica F1 o AUC en detección de fraude de tarjetas de crédito o pagos móviles, superando modelos tradicionales como regresión logística o árboles simples (Shi, 2024; Hajíček, Abedin & Sivarajah, 2022). En consecuencia, su empleo en detección de fraude en pagos en línea se considera una buena práctica empírica.

c. Proceso típico de aplicación de XGBoost en detección de fraude

El uso de XGBoost para detectar el fraude en los pagos por internet puede dividirse en varias fases fundamentales que aseguran un desarrollo metodológico robusto y replicable. Primero, es necesario preparar correctamente los datos. Esto comienza con la recopilación de un grupo de transacciones que se marcan apropiadamente como fraudulentas o legítimas. Después, se lleva a cabo el preprocesamiento, que engloba la depuración de datos, la gestión de valores ausentes y la codificación de variables categóricas; además, si es necesario, se estandarizan o normalizan las variables. La ingeniería de características, que se enfoca en crear variables derivadas como el tiempo desde la última transacción, la frecuencia de uso, el monto promedio o la variabilidad geográfica adquiere un papel importante en esta etapa porque la eficacia de XGBoost depende fuertemente del

nivel de calidad de las variables que reflejan los patrones comportamentales relacionados con el fraude.

Como el fraude es una categoría minoritaria, gestionar el desequilibrio de clases es un elemento crucial. En este sentido, Meng, Zhou y Liu (2020) informaron que al combinar XGBoost con estas metodologías se lograron mejoras en indicadores como el AUC y el recall. Para conseguirlo, es posible utilizar tácticas como el sobre-muestreo de la clase minoritaria o el sub-muestreo de la clase mayoritaria, incluyendo métodos como SMOTE. Además, se pueden asignar más pesos a los casos fraudulentos durante el entrenamiento del modelo. Sin embargo, es crucial llevar a cabo una validación meticulosa para prevenir fugas de información (data leakage); por ejemplo, garantizando que los procesos de muestreo se utilicen exclusivamente sobre los conjuntos de entrenamiento y no antes de la división de los datos, pues esto podría incrementar artificialmente los resultados (Kabane, 2024).

Es necesario modificar los hiperparámetros claves de XGBoost durante la fase de preparación y entrenamiento del modelo, incluyendo la profundidad máxima (max_depth), el número de árboles (n_estimators), los términos de regularización (lambda y alpha), las proporciones de muestreo (colsample_bytree y subsampling) y la tasa de aprendizaje (learning_rate). Investigaciones actuales sugieren que la optimización usando métodos como búsqueda bayesiana o random search puede incrementar considerablemente el rendimiento del modelo cuando se trabaja con conjuntos de datos voluminosos (Velarde et al., 2023). Con el objetivo de mantener la proporción de fraudes en cada pliegue, el entrenamiento tiene que llevarse a cabo empleando validación cruzada estratificada y supervisando métricas significativas como AUC, precisión, recall, F1-score y las medidas vinculadas al coste de errores. Además, la puesta en marcha de early stopping es fundamental para evitar el sobreajuste.

La evaluación del modelo debe sobrepasar la precisión general (accuracy) y enfocarse en métricas que muestren de forma apropiada la habilidad de detectar el fraude, como lo son: F1-score, AUC-ROC, precisión y recall. También debe incluirse métricas económicas que tengan en cuenta las consecuencias diferentes de los falsos positivos y negativos. Hajek et al. (2022) sugirieron, en esta línea, medidas de reducción de costos que combinan ambas clases de error. La validación debe llevarse a cabo sobre conjuntos de prueba independientes, examinando también la estabilidad del modelo ante alteraciones en la distribución de los datos y comprobando que no haya sesgos ni fugas de información.

Por último, una vez validado el modelo, en la puesta en marcha operativa se puede incorporar al sistema de pagos por internet para analizar cada transacción entrante y proporcionar un puntaje de riesgo. El sistema debe incluir procedimientos de actualización periódica y reentrenamiento del modelo para adaptarse a la aparición de nuevas modalidades fraudulentas y a la manera en que los usuarios se comportan con el tiempo. Dentro de este marco, se sugiere emplear técnicas de la Inteligencia Artificial Explicable (XAI), como LIME o SHAP. Estas técnicas posibilitan la interpretación del aporte de las variables para detectar el fraude y promueven la auditoría, la transparencia y el cumplimiento de los requisitos regulatorios (Almalki & Masud, 2025).

Es conveniente usar técnicas de Explainable AI (XAI) como SHAP o LIME para interpretar qué variables contribuyen a la detección de fraude y lograr transparencia y control regulatorio (Almalki & Masud, 2025).

d. Limitaciones de XGBoost

Es importante mencionar varias restricciones y consideraciones prácticas relacionadas con el empleo de XGBoost para detectar fraude en pagos en línea. Primero que nada, aunque XGBoost es un algoritmo flexible y poderoso, su rendimiento depende en gran parte de la calidad de la ingeniería de características. Por lo tanto, si faltan variables importantes o el preprocesamiento no es adecuado,

esto puede restringir considerablemente su capacidad para hacer predicciones. Además, el modelo podría mostrar una sensibilidad menor para identificar fraudes poco comunes, incluso con la ponderación de clases, si existe un desequilibrio extremo entre las clases. Además, algunos análisis alertan que el uso de métodos de muestreo previos a la separación en conjuntos de entrenamiento y prueba puede acarrear inconvenientes de fuga de datos y llevar a una sobrevaloración del desempeño del modelo (Kabane, 2024).

La necesidad de renovar el modelo con regularidad es otra consideración importante, ya que los patrones de fraude están en continua evolución y un modelo que se ha entrenado con datos históricos puede volverse ineficaz ante nuevas tácticas fraudulentas. Por otro lado, si bien XGBoost brinda un nivel aceptable de interpretabilidad, la misma no es instantánea ni totalmente clara, lo que hace necesario el uso adicional de métodos de explicación de modelos para simplificar la comprensión y justificación de las decisiones. En última instancia, a pesar de su correcto rendimiento empírico, XGBoost no asegura que el fraude se detecte de manera perfecta, pues siempre habrá falsos positivos (transacciones legítimas clasificadas como fraudulentas) y falsos negativos (fraudes no detectados). Esto enfatiza la necesidad de que las organizaciones calculen y manejen explícitamente los costos vinculados con ambos tipos de error al momento de diseñar e implementar el sistema de detección (Hajek et al., 2022).

En resumen, el modelo XGBoost constituye una herramienta altamente relevante para la detección del fraude en pagos en línea, gracias a su adaptabilidad, eficiencia, eficacia en contextos con datos desbalanceados, y respaldo empírico reciente. Su aplicación adecuada combinada con una correcta preparación de datos, gestión del desbalance de clases, ajuste de hiperparámetros, evaluación rigurosa y actualización continua permite a las organizaciones de pagos reforzar su sistema de control de fraude. En el marco teórico de una tesis de maestría sobre fraude en pagos en

línea, dedicar un apartado a XGBoost permite fundamentar la elección metodológica de forma rigurosa.

## IV. Redes neuronales Artificiales

### a. Introducción a las Redes Neuronales

A finales del siglo diecinueve se alcanzó un mayor entendimiento del cerebro humano y su funcionamiento, gracias a las investigaciones de Ramón y Cajal en España y Sherrington en Inglaterra. El primero se enfocó en la estructura de las neuronas, mientras que el segundo estudió las conexiones entre ellas, conocidas como sinapsis. El tejido nervioso, el más especializado del cuerpo, está compuesto por células nerviosas, fibras nerviosas y neuroglia, que incluye distintos tipos de células.

La célula nerviosa se conoce como neurona, que actúa como la unidad funcional del sistema nervioso. Estas pueden ser clasificadas como neuronas sensoriales, motoras y de conexión. Se estima que en cada milímetro cúbico del cerebro residen alrededor de 50. 000 neuronas.

Las neuronas poseen un soma que contiene el núcleo y realiza las funciones metabólicas, mientras recibe señales a través de las dendritas. El axón actúa como vía de salida, transmitiendo impulsos hacia otras células mediante sinapsis, donde la comunicación ocurre químicamente y puede generar un potencial de acción si se alcanza un umbral eléctrico. Este impulso viaja por el axón y se propaga a neuronas conectadas.

El sistema neuronal biológico se organiza mediante neuronas sensoriales que captan estímulos externos y los envían a una compleja red de neuronas internas encargadas del procesamiento. Posteriormente, las neuronas de salida transmiten las respuestas necesarias para activar los músculos y coordinar acciones, formando una vasta red interconectada dentro del cerebro.

### b. Redes Neuronales en los fraudes electrónicos

Las redes neuronales ofrecen un punto de vista fascinante al utilizar métodos de aprendizaje profundo para identificar fraudes en plataformas de comercio en línea (Lee J, 2020) Las redes neuronales convolucionales (CNN) y los modelos de red neuronal recurrente (RNN) son capaces de detectar transacciones fraudulentas al examinar el comportamiento del usuario y sus patrones de navegación. Este enfoque es especialmente eficaz en situaciones cambiantes donde las tácticas de fraude se desarrollan continuamente.

(Carmona Mora, 2021) destacan que los modelos de Machine Learning superan a los métodos tradicionales al detectar patrones anómalos que no son evidentes a simple vista. Algoritmos como los árboles de decisión y las redes neuronales permiten analizar grandes cantidades de información en tiempo real, lo que favorece una respuesta rápida ante posibles intentos de fraude. No obstante, señalan que el desequilibrio en los datos donde las transacciones legítimas son mucho más frecuentes que las fraudulentas representa un reto, ya que puede afectar la precisión de los modelos. Este inconveniente puede abordarse mediante técnicas como el sobre muestreo y el ajuste de los pesos en los algoritmos.

a. Proceso de aplicación de Redes Neuronales en fraude

La adopción de redes neuronales para detectar fraudes comúnmente se estructura en un enfoque metodológico claro. De acuerdo con lo indicado por (Bolton, 2002), el procedimiento comienza con la recopilación, depuración y organización de los datos, una fase esencial por el gran número de transacciones y la frecuente aparición de datos anómalos o inconsistentes. Este paso asegura que la información empleada represente correctamente tanto los comportamientos regulares como los fraudulentos.

A continuación, se lleva a cabo la elección y creación de atributos. Investigadores como (Bishop, 2006) y (Bhattacharyya, 2011) afirman que la manipulación de variables consistiendo en cambios, combinaciones temporales y medidas del comportamiento del consumidor es crucial para aumentar

la capacidad de predicción de los modelos. Asimismo, es común implementar métodos de normalización o disminución de dimensiones para ayudar en la convergencia del algoritmo.

Después de organizar los datos, se lleva a cabo la separación de la información en grupos de entrenamiento, validación y prueba. Según (Haykin, 2009) esta división es fundamental para calibrar los hiperparámetros del modelo y reducir la posibilidad de sobreajuste, dado que los casos de fraude normalmente son solo una pequeña parte del total.

La formación del modelo forma el núcleo del proceso. En este paso, los algoritmos de redes neuronales, en particular los que utilizan retropropagación, modifican sus pesos internos para reconocer patrones complejos y no lineales. La investigación fundamental de (Rumelhart, 1986) muestra que este sistema capacita a las redes para captar estructuras profundas en los datos aun cuando las características engañosas sean discretas.

Luego, el sistema requiere realizar un análisis detallado, teniendo en cuenta especialmente la disparidad entre transacciones válidas y fraudulentas. (Fawcett, 1997) indican que indicadores como la curva ROC, el AUC, la sensibilidad y la precisión son más apropiados que la precisión estándar para evaluar la eficacia en situaciones marcadamente desequilibradas.

Finalmente, de acuerdo con (Ngai, 2011), es necesario llevar a cabo un seguimiento y ajuste constante del proceso, dado que las tácticas de fraude están en constante cambio. Esto significa que se debe actualizar el modelo utilizando información reciente, modificar las características y revisar regularmente la efectividad del sistema para reconocer nuevos patrones inusuales.

### *2.2.5 Técnicas de selección de características en detección de fraude*

La selección de características constituye un componente clave en los sistemas de detección de fraude, porque permite identificar los atributos más informativos, reducir la redundancia y mejorar tanto el rendimiento como la eficiencia de los modelos de aprendizaje automático. En contextos

financieros, donde los conjuntos de datos suelen ser de alta dimensionalidad y están fuertemente desbalanceados, una mala elección de variables incrementa el ruido, empeora la capacidad de generalización y eleva la tasa de falsos positivos, lo cual es especialmente crítico en escenarios de monitoreo en tiempo real (Fu et al., 2025; Hernández Aros et al., 2024).

En la literatura se suele distinguir entre tres grandes familias de técnicas de selección de características: métodos *filter*, *wrapper* y *embedded*. Los métodos *filter* evalúan cada atributo de forma independiente utilizando criterios estadísticos, como la correlación, la información mutua o pruebas de chi-cuadrado, sin depender de un modelo específico. Este tipo de enfoque suele emplearse como una etapa inicial para eliminar variables claramente irrelevantes o altamente correlacionadas entre sí, reduciendo la dimensionalidad antes de entrenar modelos más complejos (Hernández Aros et al., 2024; Siam et al., 2025).

Por su parte, los métodos *wrapper* utilizan el desempeño de un modelo de aprendizaje como criterio para seleccionar subconjuntos de características. Una técnica muy extendida es Recursive Feature Elimination (RFE)**,** que entrena el modelo sobre el conjunto completo de atributos y, de forma iterativa, va eliminando aquellas variables con menor importancia hasta alcanzar un subconjunto óptimo. Estudios recientes en fraude financiero muestran que RFE permite mejorar métricas como la exactitud y el *recall* al descartar atributos redundantes que inducen sobreajuste y aumentan la complejidad computacional (Jin & Zhang, 2025).

Los métodos *embedded* integran la selección de características dentro del propio proceso de entrenamiento del modelo. Este es el caso de algoritmos basados en árboles, como Random Forest**,** Gradient Boosting**,** XGBoost **o** LightGBM, que calculan de manera interna la importancia de cada variable a partir de criterios de ganancia de información o reducción de impureza. Estos enfoques resultan especialmente útiles en detección de fraude, porque combinan un buen rendimiento

predictivo con un cierto grado de interpretabilidad: permiten identificar atributos clave como montos atípicos, patrones de frecuencia de transacciones o cambios bruscos en el comportamiento del cliente (Compagnino, 2025; Chen, 2023).

En el ámbito del fraude financiero, también se han propuesto esquemas de *selección integrada o híbrida*, que combinan varias técnicas para mitigar las limitaciones de usar un único método. Por ejemplo, Chen (2023) plantea un enfoque de selección integrada para fraude en estados financieros que combina la importancia de características obtenida por Random Forest, GBDT, XGBoost y LightGBM, mostrando que esta estrategia mejora el AUC y el *recall* frente a aplicar cada método por separado, especialmente cuando se trabaja en conjunto con técnicas de balanceo como SMOTE. De manera similar, trabajos recientes en fraude transaccional proponen marcos híbridos que combinan filtros estadísticos con importancia de características basada en modelos, logrando reducir el número de variables sin sacrificar desempeño e incluso mejorando la capacidad de detección de casos minoritarios (Siam et al., 2025).

El auge del *deep learning* ha introducido otra perspectiva sobre la selección de características, al permitir el aprendizaje de representaciones latentes de forma automática. Modelos como autoencoders y redes recurrentes capturan patrones temporales y no lineales sin requerir necesariamente una selección explícita de variables en la etapa previa. No obstante, aun cuando el modelo aprende representaciones internas, muchos trabajos combinan estas arquitecturas con técnicas de selección o reducción de dimensionalidad en la fase de entrada, tanto para mejorar la eficiencia como para facilitar la interpretación de los resultados (Hernández Aros et al., 2024; Jin & Zhang, 2025).

En síntesis, la selección de características en detección de fraude no es solo un paso técnico accesorio, sino una estrategia central para optimizar la precisión, reducir el costo computacional y

mejorar la explicabilidad de los modelos. En el contexto del presente proyecto, estas técnicas resultan especialmente relevantes, ya que permiten concentrar el aprendizaje en las variables más informativas de las transacciones (montos, tiempos, canales, patrones anómalos de uso, entre otras), lo cual es indispensable para un portal web de detección temprana de fraude en pagos en línea orientado a la banca.

#### 2.2.5.1 Ingeniería de características

La ingeniería de características es un paso clave en el proceso de aprendizaje automático porque determina qué información recibe el modelo y cómo la recibirá. Aunque los algoritmos actuales son capaces de identificar patrones complejos, su rendimiento depende en gran medida de la calidad y estructura de los datos. Por tanto, es necesario transformar, depurar o crear nuevas variables antes de entrenar el modelo.

##### 2.2.5.1.1 Manejo de datos faltantes

Cuando un conjunto de datos contiene valores faltantes, por lo general se debe a errores en la recolección, a preguntas no respondidas o a mediciones que no aplican en ciertos casos. Estos valores suelen aparecer como espacios en blanco, NaN o NULL, y la mayoría de las herramientas no los procesa correctamente, lo que puede llevar a resultados imprecisos (Raschka, Liu, & Mirjalili, 2022).

Una forma sencilla de enfrentarlo es eliminar las filas o columnas que contienen valores faltantes. Métodos como dropna permiten hacerlo rápidamente. Sin embargo, esta opción puede tener un costo importante: podríamos quedarnos con muy pocas muestras o perder características que aportan información relevante para el modelo, afectando su desempeño (Raschka, Liu, & Mirjalili, 2022).

#### 2.2.5.1.2 Técnica de Imputación

Cuando descartar datos no es una opción viable, se recurre a diferentes métodos de imputación para estimar los valores faltantes. Una técnica básica y bastante habitual es reemplazar los valores ausentes por la media de la columna (para datos numéricos), se puede usar la mediana que es menos sensible a valores extremos o la moda (para datos categóricos). Aunque es un enfoque simple, permite conservar el tamaño del conjunto de datos y seguir adelante con el proceso de modelado sin perder información que podría resultar útil (Raschka, Liu, & Mirjalili, 2022).

En un estudio de detección de fraude de los autores Feng y Kim (2025) demostraron que cuando el 50% de una variable está ausente, el uso de la imputación específicamente la moda permite conservar la información crítica y mantener un volumen adecuado de datos para entrenar modelos eficaces. Este tipo de estrategia refuerza la idea de que no basta con limpiar superficialmente los datos: es necesario un preprocesamiento robusto que combine imputación y transformación para asegurar que los modelos cuenten con características suficientemente representativas y equilibradas. Al adoptar estas prácticas en mi proyecto, busco replicar un enfoque que ha demostrado mejorar tanto la estabilidad como la precisión en la predicción de transacciones fraudulentas (Feng & Kim, 2025).

#### 2.2.5.1.3 Transformación y escalado de variables numéricas

Los modelos de aprendizaje automático pueden ser sensibles a la escala de los datos. Es decir, si una característica toma valores entre 0 y 1, mientras otra varía de 0 a 10,000, la segunda puede dominar la función de pérdida del modelo. Para corregir esto se aplican técnicas como:

- Estandarización: consiste en centrar los datos en torno a cero y escalarlos según su desviación estándar. Esto permite que todas las variables contribuyan de manera equitativa al modelo, asignándoles los mismos parámetros que una distribución normal estándar (media

cero y varianza uno), lo que facilita el aprendizaje de los pesos en los algoritmos de machine learning (Raschka, Liu, & Mirjalili, 2022).

- Normalización: consiste en ajustar los valores de las variables a un rango específico, normalmente entre 0 y 1. Este procedimiento mejora la estabilidad y el rendimiento de modelos sensibles a las magnitudes absolutas, como redes neuronales o K-Nearest Neighbors (KNN). La normalización se considera un caso particular de escalamiento mínimo-máximo (Raschka, Liu, & Mirjalili, 2022).

#### 2.2.5.1.4 Codificación de variables categóricas

Para que los modelos procesen variables categóricas, es necesario convertirlas a formato numérico. La elección del método depende del tipo de atributo y su relación con el target evitando sesgos y permitiendo al modelo aprender de forma más eficiente.

- One-hot encoding: consiste en crear una nueva característica ficticia para cada valor único en la columna de característica nominal, crea columnas binarias para cada categoría, evitando que el modelo interprete un orden inexistente entre ellas, pero debemos tener en cuenta que esto introduce multicolinealidad.
- Codificación ordinal: asigna un número a cada categoría cuando existe un orden natural (por ejemplo, niveles de riesgo: bajo, medio, alto).
- Embeddings: en problemas con muchas categorías, se pueden usar representaciones vectoriales densas para reducir dimensionalidad y capturar relaciones semánticas.

En el estudio de Bourdonnaye y Daniel (2021) analizan cómo diferentes métodos de codificación de variables categóricas influyen en la eficacia de los modelos de detección de fraude con tarjetas. Comparan técnicas clásicas como el one-hot encoding con aproximaciones basadas en estadísticas, como el target encoding o Weight of Evidence, y demuestran que estas pueden aumentar significativamente el rendimiento del modelo. Este hallazgo sugiere que la forma en que se

representan las variables categóricas no es un detalle menor, sino un componente vital que puede mejorar drásticamente la capacidad del algoritmo para discriminar entre transacciones legítimas y fraudulentas (Bourdonnaye & Daniel, 2021).

### *2.2.6 Evaluación de modelos de detección de fraude*

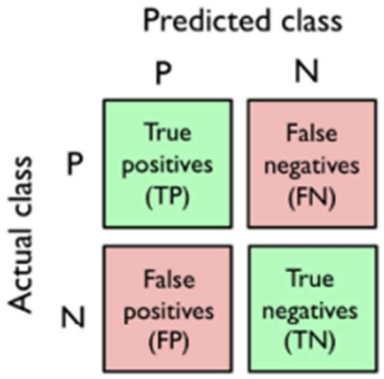
#### 2.2.6.1 Métricas de clasificación

En su libro, Raschka afirma que es importante ir más allá de la precisión al evaluar modelos de clasificación, especialmente cuando las clases no están equilibradas, por lo que sugiere utilizar una combinación de métricas: precisión (cuántos de los positivos predichos son realmente positivos), recall o sensibilidad (qué tan bien el modelo captura los verdaderos positivos) y F1-score, que armoniza precisión y recall para ofrecer una visión equilibrada. Todas estas métricas están disponibles directamente en sklearn.metrics, lo que le permite comparar modelos de forma transparente y ajustar su rendimiento en función de lo que realmente importa al problema (Raschka, Liu, & Mirjalili, 2022).

La matriz de confusión es una herramienta que permite evaluar cómo se desempeña un modelo de aprendizaje automático. Se presenta como una matriz cuadrada que muestra la cantidad de casos clasificados correcta y erróneamente, distinguiendo entre verdaderos positivos (VP), verdaderos negativos (VN), falsos positivos (FP) y falsos negativos (FN), como se muestra en la Figura:

**Figura 1**

***Matriz de Confusión***




*Nota:* Estructura de una matriz de confusión para clasificación binaria. Representa la relación entre los valores reales (Actual class) y las predicciones del modelo (Predicted class), permitiendo identificar el rendimiento a través de aciertos (TP, TN) y errores (FP, FN).

*Fuente:* (Raschka, Liu, & Mirjalili, 2022).

Tanto el error de predicción (ERR) como la precisión (ACC) ofrecen una visión general sobre el desempeño de un modelo, indicando cuántos casos se clasificaron correcta o incorrectamente. El error se calcula dividiendo el número total de predicciones falsas entre el total de predicciones realizadas, mientras que la precisión se obtiene dividiendo las predicciones correctas entre el total de predicciones:

$$ERR = \frac{FP+FN}{FP+FN+TP+T} \tag{6}$$

La precisión de la predicción se puede calcular directamente a partir del error:

$$ACC = \frac{TP+T}{FP+FN+TP+TN} = 1 - ERR \tag{7}$$

La tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) son métricas de rendimiento especialmente útiles para problemas de clases desequilibradas:

$$FPR = \frac{FP}{N} = \frac{FP}{FP+TN} \tag{8}$$

$$TPR = \frac{TP}{P} = \frac{TP}{FN+} \quad (9)$$

A diferencia de la tasa de falsos positivos (FPR), la tasa de verdaderos positivos (TPR) indica qué proporción de los casos positivos se identificó correctamente dentro del total de positivos. Las métricas de precisión (PRE) y recall o recuperación (REC) están estrechamente vinculadas con los verdaderos positivos y negativos; de hecho, la recuperación coincide con la TPR:

$$REC = TPR = \frac{TP}{P} = \frac{TP}{FN+TP} \quad (10)$$

La recuperación mide cuántos registros relevantes (los positivos) se capturan como tales (los verdaderos positivos). La precisión cuantifica cuántos registros predichos como relevantes (la suma de verdaderos y falsos positivos) son realmente relevantes (verdaderos positivos) (Raschka, Liu, & Mirjalili, 2022).

$$PRE = \frac{TP}{TP+FP} \quad (11)$$

Para equilibrar las ventajas y desventajas de optimizar PRE y REC, se utiliza la media armónica de PRE y REC, la denominada puntuación F1:

$$F1 = 2\frac{PRE \times REC}{PRE+REC} \quad (12)$$

Finalmente tenemos la curva de ROC que se utilizan para evaluar cómo se comporta un modelo de clasificación según su capacidad de identificar correctamente los positivos (TPR) y evitar falsos positivos (FPR). La diagonal del gráfico representa el desempeño de un clasificador aleatorio, mientras que un clasificador ideal estaría en la esquina superior izquierda, donde identifica todos los positivos sin cometer errores. El área bajo la curva resume el rendimiento general del modelo en un solo valor, facilitando la comparación entre distintos clasificadores (Raschka, Liu, & Mirjalili, 2022).

**Tabla 2**

*Métricas de Clasificación*

| ***Métrica*** | ***Qué mide*** | ***Cuando usar*** |
|---|---|---|
| ***Accuracy (Precisión general)*** | Porcentaje de predicciones correctas sobre el total | Útil para tener una visión global del desempeño, pero puede engañar si hay desbalance de clases (muchos más casos legítimos que fraudulentos). |
| ***Precision (Precisión)*** | Qué proporción de las predicciones positivas son realmente positivas | Importante cuando queremos evitar falsas alarmas o alertas innecesarias a los clientes. |
| ***Recall (Recuperación / Sensibilidad)*** | Qué proporción de los positivos reales fueron detectados | Fundamental si no queremos que se escape ningún caso de fraude, incluso a costa de algunas falsas alarmas. |
| ***F1-score*** | Promedio armónico entre precisión y recuperación | Útil cuando queremos un equilibrio entre no perder fraudes y no generar falsas alertas. |
| ***ROC-AUC*** | Capacidad del modelo para distinguir entre clases positiva y negativa | Ideal para comparar modelos, especialmente con datos desbalanceados, y entender el rendimiento global más allá de un solo umbral. |

*Nota:* Resumen de métricas de desempeño derivadas de la matriz de confusión. Estas métricas evalúan la eficacia del modelo desde distintas perspectivas: la exactitud global (Accuracy), la capacidad de identificar positivos (Recall) y la precisión de las predicciones positivas (Precision).

*Fuente:* Elaboración Propia

### *2.2.7 Explicabilidad de modelos en la detección de fraude*

La explicabilidad es un componente fundamental en los modelos destinados a detectar fraudes. La interpretación del razonamiento que subyace a una predicción posibilita la creación de confianza en los modelos de aprendizaje automático, sobre todo si estos se emplean para apoyar decisiones regulatorias o financieras, como indican (Doshi-Velez, 2017)

Además, la explicabilidad ayuda a detectar conductas no deseadas, sesgos o errores. De acuerdo con lo que dice (Carcillo, 2019) para prevenir acusaciones falsas y asegurar la transparencia ante auditorías internas y externas, es necesario fundamentar cada alerta generada en el proceso de detección de fraude.

#### 2.2.7.1 Interpretabilidad en Machine Learning

La interpretabilidad se refiere a la capacidad de un modelo para ser entendido por humanos, ya sea en su estructura, en su funcionamiento o en los factores que influyen en sus predicciones. De acuerdo con (Molnar, 2022),un modelo es interpretable cuando sus decisiones pueden explicarse de manera directa sin recurrir a aproximaciones adicionales.

##### 2.2.7.1.1 Modelos interpretables vs. modelos de caja negra

Entre los modelos interpretables por diseño se encuentran las regresiones lineales, árboles de decisión y sistemas basados en reglas. (Rudin, 2019) argumenta que, en contextos de alto riesgo como el fraude, estos modelos deberían preferirse cuando pueden alcanzar un rendimiento comparable al de modelos complejos.

Sin embargo, métodos más potentes como redes neuronales, Random Forest, Gradient Boosting o SVM suelen ofrecer mejor desempeño en escenarios con patrones no lineales o datos altamente desbalanceados. En tales casos, la interpretabilidad debe obtenerse mediante técnicas externas.

#### 2.2.7.1.2 Interpretabilidad global y local

Según (Guidotti R, 2018) la interpretabilidad puede dividirse en dos dimensiones:

- Interpretabilidad global: explica cómo funciona el modelo en términos generales, permitiendo identificar qué variables tienen mayor influencia.
- Interpretabilidad local: se enfoca en explicar una predicción específica, útil para analizar casos sospechosos o justificar decisiones particulares ante auditorías.

### 2.2.7.2 Métodos de explicabilidad (LIME, SHAP)

A) LIME: explicabilidad local basada en modelos sustitutos

LIME (Local Interpretable Model-agnostic Explanations), creado por Ribeiro, Singh y Guestrin en 2016, es una técnica de explicabilidad que produce explicaciones locales al simular el funcionamiento del modelo original con uno que puede ser interpretado, normalmente lineal, alrededor de la vecindad de una observación concreta. LIME posibilita entender por qué una transacción específica ha sido catalogada como peligrosa, favorece la revisión manual de casos atípicos hecha por analistas antifraude y ayuda a reconocer variables que afectan de forma imprevista las decisiones individuales del modelo en el marco de la detección de fraude. De esta manera, se refuerza la confianza y la transparencia en el sistema de detección.

B) SHAP: explicabilidad basada en teoría de juegos

SHAP (SHapley Additive exPlanations), presentado por Lundberg y Lee en 2017, es un método de explicabilidad superior que emplea la teoría de juegos cooperativos para otorgar a cada atributo una contribución cuantitativa y aditiva al modelo predictivo, mediante el uso de los valores de Shapley. Este método asegura características deseables tales como la igualdad, la coherencia y la consistencia. Esto quiere decir que si una variable aporta más a la predicción de un modelo que de

otro, su relevancia según SHAP también será mayor, lo que permite brindar explicaciones comparables y con fundamentos matemáticos.

SHAP aporta numerosas ventajas en el marco de la identificación de fraude. Habilita la creación de explicaciones locales, que son útiles para comprender las razones por las cuales una transacción específica fue catalogada como fraudulenta o legítima, y explicaciones globales, que hacen más fácil el estudio del comportamiento general del modelo y el reconocimiento de patrones sistemáticos de fraude. Asimismo, permite crear clasificaciones de la relevancia de las variables con apoyo teórico, lo cual es útil para priorizar factores de riesgo y respaldar el proceso de tomar decisiones estratégicas. SHAP también hace posible detectar efectos no lineales y relaciones complejas entre variables, que son difíciles de notar con las técnicas convencionales, como la combinación de localización, importe y hora de la transacción. Además, sus visualizaciones (como summary plots y dependence plots) hacen más fácil que los analistas, auditores y reguladores se comuniquen sobre los resultados, lo cual mejora la transparencia, el rastreo y la confianza en sistemas antifraude implementados en contextos financieros esenciales.

Según Lundberg et al. (2020), SHAP se ha convertido en uno de los métodos más robustos para auditar modelos complejos en entornos financieros.

### *2.2.8 Desarrollo de aplicaciones web para sistemas de detección*

La creación de aplicaciones en línea enfocadas en sistemas de identificación, que abarcan la detección de fraudes, anomalías u otras situaciones vitales, se basa en fundamentos de desarrollo de software, arquitecturas distribuidas y un diseño que prioriza la efectividad en el manejo de datos. De acuerdo con (Pressman, 2010) estas aplicaciones necesitan la integración de varios elementos que puedan funcionar en tiempo real, ofreciendo interfaces accesibles al mismo tiempo que procesan datos de diversas fuentes.

En concreto, los sistemas de detección que usan análisis automatizado requieren plataformas en línea que faciliten la visualización dinámica de los resultados, la ejecución de modelos computacionales y la interacción con bases de datos centralizadas. Investigadores como (Sommer, 2010) afirman que la eficacia de estos sistemas está ligada a la capacidad de la aplicación para manejar grandes cantidades de datos, actualizar modelos predictivos y presentar alertas de forma instantánea al usuario.

Asimismo, la literatura más reciente resalta la importancia de aplicar prácticas para un desarrollo seguro. Según (Stallings, 2017), es fundamental salvaguardar la confidencialidad e integridad de los datos, en particular cuando se trata de información delicada o transacciones económicas. Por esta razón, los sistemas web de detección a menudo incluyen procedimientos como la validación del tráfico, el cifrado de datos, el control de acceso basado en roles y la autenticación sólida.

Por último, investigaciones como la de (Hosseini, 2019) enfatizan que es necesario que la arquitectura web posibilite que el sistema sea escalable, teniendo en cuenta que, si aumenta el número de usuarios, datos o transacciones, esto puede hacer crecer la demanda de procesamiento. Esta perspectiva asegura que los modelos de detección continúen funcionando de manera eficaz a medida que aumentan las exigencias del entorno.

#### 2.2.8.1 Arquitectura de sistemas web

La arquitectura de sistemas web está integrada por un conjunto de patrones tecnológicos, componentes y estructuras que posibilitan el desempeño de aplicaciones que se pueden acceder a través de navegadores y servicios en línea. (Sommerville, 2016) señala que esta clase de arquitectura establece la forma en que se estructuran los módulos de software, cómo se relacionan entre ellos y cómo se asegura el rendimiento, la fiabilidad y la seguridad del sistema. La arquitectura web es

particularmente importante en el marco de sistemas que detectan fraudes, ya que tiene que ser capaz de manejar gran cantidad de datos, consultas simultáneas y procesos de inferencia en tiempo real.

La arquitectura cliente-servidor es uno de los métodos más empleados; en este modelo, el cliente (navegador) accede a la aplicación y las operaciones comerciales y el procesamiento central se llevan a cabo en servidores lejanos. La distribución y el mantenimiento de actualizaciones se simplifican con este tipo de arquitectura, debido a que las modificaciones se ejecutan directamente en el servidor sin perjudicar la experiencia del usuario final (Pressman R. S., 2020) La arquitectura de tres capas (three-tier) se basa con frecuencia en este modelo, dividiendo el sistema en: (1) la capa de presentación, que se ocupa de interactuar con el usuario; (2) la capa lógica o de negocio, que es responsable del procesamiento de reglas, comunicación a través de APIs y ejecución de modelos de aprendizaje automático; y (3) la capa de datos, donde se guardan las transacciones, los historiales y los resultados predictivos. Al dividir en módulos, se mejora la capacidad de escalabilidad, se simplifica el mantenimiento y es posible incorporar modelos analíticos sin modificar la estructura general de la aplicación.

El empleo de microservicios ha ganado importancia en sistemas contemporáneos, particularmente los que están dirigidos a la analítica avanzada y la identificación de irregularidades. Según (Fowler, 2015) esta arquitectura segmenta el sistema en servicios independientes que tienen la capacidad de implementarse, actualizarse y escalarse de forma independiente. En el caso de aplicaciones de fraude, esto supone la oportunidad de contar con servicios concretos para auditoría, autenticación, visualización, gestión de usuarios o scoring de riesgo. Esta flexibilidad posibilita que el motor de machine learning sea actualizado, sustituido o versionado sin que la operación total del sistema se vea afectada, lo que le confiere una mayor fortaleza ante las variaciones en los patrones de fraude.

#### 2.2.8.2 Interfaces para usuarios técnicos y no técnicos

Dado que las interfaces de usuario son el punto de interacción directa entre los individuos y el sistema, su diseño tiene que ajustarse al perfil y nivel de conocimiento del usuario. Según (Nielsen, 2012) la claridad, la eficiencia y la usabilidad son componentes fundamentales para asegurar una experiencia óptima. En aplicaciones de detección de fraude, esto significa que los usuarios con perfiles diferentes (comerciales, operativos, analistas o ingenieros) sean capaces de comprender adecuadamente la información expuesta y tomar decisiones fundamentadas en ella.

- Usuarios técnicos: (Por ejemplo, analistas de datos, ingenieros de sistemas o expertos en fraude) necesitan interfaces que tengan un nivel de detalle más alto. (Shneiderman, 2017) subrayan que este tipo de usuario requiere tener acceso a las variables que nutren el modelo, así como a las métricas de rendimiento (AUC, exactitud, recall), configuraciones avanzadas, registros del sistema y alternativas para exportar información. Estas interfaces tienen como objetivo ofrecer control y trazabilidad, sin sacrificar la claridad y el orden. En consecuencia, es frecuente que el sistema proporcione vistas distintas de acuerdo con la función del usuario, asegurándose de que cada perfil tenga acceso únicamente a los datos requeridos para realizar su trabajo.
- Usuarios técnicos: La interfaz debe ser más simple y menos compleja cognitivamente para los usuarios sin formación técnica, como lo son los agentes de atención, el personal de servicio y los asesores comerciales. Esto necesita que se muestren resultados a través de indicadores intuitivos (como colores, semáforos y alertas claras), explicaciones en un lenguaje cotidiano y flujos guiados que posibiliten realizar acciones como examinar un caso, validar información o escalar una alerta. Estas interfaces tienen que evitar la sobrecarga de datos técnicos, poniendo énfasis en la presentación visual a través de gráficos o resúmenes.

2.2.8.3 Sistema de implementación de APP

En el sistema establecido, las imágenes se utilizan como pruebas técnicas del análisis que cada modelo ha llevado a cabo. Estas evidencias no se "dibujan" directamente en el portal, sino que son producidas por el código Python vinculado a cada modelo como parte del proceso analítico. Esto puede incluir visualizaciones de resultados, gráficas generadas por el modelo, ilustraciones de apoyo para entender cómo actúa el clasificador o comparaciones de rendimiento. El portal se ocupa después de organizar y mostrar las evidencias mencionadas para que el usuario, especialmente el supervisor y el analista, pueda registrar sus conclusiones e interpretaciones de forma ordenada.

Desde una perspectiva teórica, esta arquitectura se basa en una separación clásica entre la capa operacional y la analítica. La capa analítica, de naturaleza experimental u offline, incluye scripts en Python que llevan a cabo el entrenamiento y la evaluación de los modelos, así como la generación de artefactos como reportes, tablas o figuras. La capa operacional, por su parte, se refiere al portal, que utiliza estos artefactos e los incorpora en un flujo controlado que facilita la edición, revisión, aprobación y consulta posterior.

Esta división es particularmente beneficiosa ya que disminuye la interdependencia entre el progreso de los modelos y el desarrollo del portal. Esto posibilita la creación de nuevas evidencias desde Python sin que sea necesario volver a escribir la interfaz de usuario. También promueve el seguimiento del análisis, dado que las evidencias se vinculan a un informe particular y a un estado dentro del flujo de trabajo, como revisión o borrador. Al transformar el informe en una bitácora estructurada que anota lo observado y lo concluido a partir de cada evidencia, incentiva la revisión humana. (Express, s.f.)

Además, la implementación del portal sostiene conclusiones basadas en evidencia, es decir, en cada imagen producida, lo cual refuerza el análisis detallado. Así, el informe ya no es una sola

pieza de texto, sino un conjunto de descubrimientos asociados a cada imagen o modelo. Esto es particularmente importante cuando se comparan diferentes modelos o salidas de un mismo modelo, pues cada prueba puede explicar de forma explícita por qué se sugiere una decisión determinada o por qué se detecta un patrón sospechoso específico. (RFC Editor, 2015)

2.2.8.4 Gestión de reportes y flujo de trabajo (workflow)

El repositorio establece un proceso formal para la elaboración de informes, que se determina mediante los estados DRAFT, IN_REVIEW, OBSERVED y APPROVED. Desde el punto de vista teórico, este esquema puede ser considerado como un modelo de ciclo de vida donde la condición de un objeto, específicamente el informe, define quién tiene la capacidad de actuar sobre él, qué operaciones se permiten y cuál es el sentido del informe en cada instante, ya sea en estado de borrador, en proceso de validación, observado o aprobado.

Con respecto a la gobernanza y al control de calidad, este método proporciona consistencia y orden, porque evita que varios participantes alteren un mismo informe sin supervisión. Por ejemplo, impidiendo que un supervisor edite un borrador como si fuera analista. Además, establece responsabilidad o rendición de cuentas, ya que cada transición de estado supone un actor claramente identificado, como el analista que envía el informe para su revisión o el supervisor que decide examinarlo o aprobarlo. Asimismo, ayuda a evitar errores, ya que el flujo impide las ediciones cuando el informe está aprobado o en revisión y posibilita que el analista lo edite nuevamente, sobre todo cuando está en borrador o ha sido observado. (OWASP, s.f.)

Específicamente, el estado OBSERVED establece un ciclo explícito de retroalimentación. En teoría, este mecanismo puede ser considerado un retrabajo controlado, en el que el supervisor no solo rechaza el reporte, sino que lo devuelve con observaciones específicas para corregirlo. Este principio se fortalece en la implementación del repositorio, pues se previene que persistan entradas vacías y se facilita que el borrador recupere de manera adecuada las conclusiones vinculadas a

cada imagen. Esto es crucial para asegurar que el ciclo de "observar → corregir → reenviar" sea factible y no pierda el contexto del análisis anterior.

El modelo cliente-servidor es la base de la arquitectura del software del portal. El frontend, que funciona como cliente, está constituido por páginas en HTML y JavaScript estáticas, las cuales se distinguen por rol (SUPERVISOR, VIEWER, ANALYST y ADMIN). Estas páginas representan las vistas y utilizan HTTP para acceder a los endpoints del sistema. El backend, desarrollado con Node.js y Express, ofrece estos endpoints y centraliza la lógica comercial, que abarca la autorización, las validaciones, la persistencia de datos y la exportación a PDF. SQLite se emplea como base de datos local para el almacenamiento.

Esta arquitectura, desde el marco teórico, se adhiere al principio de división de responsabilidades, estableciendo una clara distinción entre la lógica de negocio, la gestión de datos y la interfaz del usuario. Además, el empleo de APIs permite establecer un contrato preciso entre cliente y servidor, en el que el backend determina acciones específicas como la generación, aprobación o exportación a PDF de informes. El modularidad funcional por roles posibilita que cada clase de usuario cuente con acciones y pantallas limitadas, disminuyendo la complejidad del cliente y simplificando el crecimiento progresivo del sistema.

En este marco, el backend funciona como la "fuente única de la verdad" (single source of truth) del sistema, ya que concentra el manejo de los estados del reporte, las reglas de transición, los permisos por rol, el rango de datos a los que cada usuario puede acceder por ejemplo, restringiendo al rol VIEWER a los analistas asignados y la elaboración de documentos como los PDF con control de acceso. Para impedir que el cliente asuma responsabilidades de seguridad que no le pertenecen, esta centralización es esencial. A pesar de que el frontend tiene la capacidad de ocultar botones o alternativas, la autorización efectiva debe llevarse a cabo en el servidor, siguiendo las buenas prácticas de diseño seguro del software. (Puppeteer, s.f.)

#### 2.2.8.5 Autenticación (JWT) y autorización (RBAC + scoping)

La implementación aplica un sistema de seguridad común en aplicaciones web con API, fundamentado en la autorización y la autenticación. La autenticación se lleva a cabo por medio de JWT, lo cual posibilita un modelo sin estado en el que el servidor comprueba quién es el usuario usando el token, sin la necesidad de conservar sesiones en memoria.

La autorización, que depende en gran medida del RBAC, se basa en que cada usuario tiene un rol (ANALYST, SUPERVISOR o VIEWER) que determina qué tareas puede llevar a cabo. Además, se añade un control de alcance (scoping) para que el acceso no dependa únicamente del rol, sino también de relaciones y estados: los analistas acceden a sus propios informes, los supervisores los examinan y dan su aprobación, y los viewers solo pueden consultar informes aprobados de analistas asignados. Desde una perspectiva teórica, el sistema fusiona RBAC con un esquema ABAC que se basa en relaciones y atributos, fortaleciendo así el principio de mínimo privilegio y asegurando un control de acceso más exacto y seguro.

#### 2.2.8.6 Persistencia local con SQLite (modelado relacional y consistencia)

SQLite es un motor de base de datos relacional embebido que guarda los datos en un archivo local, lo cual le permite ser apropiado para sistemas que necesitan una implementación sencilla, persistencia transaccional, integridad referencial y un costo operativo reducido, como las aplicaciones locales o los prototipos.

Dentro del repositorio, SQLite permite la existencia de roles y usuarios, relaciones de asignación entre actores, reportes con metadatos y estados, conclusiones por imagen y un registro elemental de acciones. Desde el enfoque teórico, el diseño hace uso de principios tradicionales de bases de datos relacionales, entre los que se encuentran la normalización parcial, el empleo de tablas puente para relaciones N:M, la integridad a través de claves foráneas y una evolución del esquema que es poco intensa.

El almacenamiento local, además, favorece la reproducibilidad y el control académico, lo que permite mantener abierta la opción de pasar a un RDBMS de servidor en el futuro sin modificar el modelo conceptual del sistema.

2.2.8.7 Generación de PDF desde el backend (rendering server-side)

Puppeteer, un motor de renderizado que se basa en un navegador sin interfaz (headless), es el encargado de generar los PDF del lado del servidor. Desde una perspectiva teórica, este procedimiento implica la creación de documentos en un servidor. En este caso, el backend crea una plantilla en HTML y la convierte directamente a PDF, asegurando que la salida sea consistente e independiente del navegador o de las preferencias del usuario.

Dado que la autorización para la creación y descarga del PDF se gestiona en el backend, este método también fortalece la seguridad y el control de acceso, ya que impide que un usuario acceda a documentos de informes sin permiso. El PDF se transforma en un objeto estandarizado y formal en sistemas de informes, que puede archivarse, compartirse o auditarse. Permite incluir evidencias y resultados en un formato seguro e inalterable, lo que evita la falta de control y la variabilidad asociada con depender de la impresión directa desde el navegador. (imbalanced-learn developers, s.f.)

# CAPITULO 3

## 3. DESARROLLO

### 3.1 Metodología de Desarrollo Ágil (Scrum y Kanban)

#### *3.1.1 Implementación de Scrum*

Scrum se utilizó como el marco central para la planificación y organización del proyecto. Dado que el desarrollo combina etapas analíticas (preprocesamiento de datos, selección de características, experimentación de modelos) y etapas de ingeniería (desarrollo de frontend, backend, reportes explicables), Scrum permitió dividir el trabajo en sprints de dos semanas, cada uno con metas claras y verificables. (Schwaber, 2020)

Durante cada sprint se realizaron las siguientes actividades formales:

a) Sprint Planning

El equipo definió las funcionalidades a desarrollar según su prioridad e impacto en el avance del proyecto. En esta etapa se planificaron tareas relacionadas con el análisis exploratorio de datos, entrenamiento de algoritmos, desarrollo de APIs, diseño de las interfaces web y generación de reportes explicables.

Al tratarse de un proyecto académico que combina investigación y construcción técnica, esta fase resultó esencial para mantener claridad en los objetivos inmediatos y evitar desviaciones del alcance.

b) Daily Meetings (adaptadas)

Aunque el proyecto no requería reuniones diarias extensas, se realizaron sesiones breves de seguimiento para identificar bloqueos, revisar avances y redistribuir tareas cuando era necesario. Esto ayudó a mantener alineados los esfuerzos entre los integrantes,

especialmente en momentos clave como la integración del modelo en el portal o la validación de las explicaciones generadas por SHAP o LIME.

c) Sprint Review

Al final de cada sprint se presentó un incremento funcional del sistema: un modelo entrenado, una sección del portal web, un prototipo de reporte o una visualización explicativa. Esta práctica permitió obtener retroalimentación temprana, corregir desviaciones y asegurar que cada iteración aportara un valor real al proyecto.

d) Sprint Retrospective

El equipo evaluó qué funcionó bien, qué debía mejorarse y qué podría optimizarse para el siguiente sprint. Esto permitió ajustar la forma de trabajar, mejorar los tiempos de integración y reorganizar responsabilidades para maximizar la productividad. (Beck, 2001)

### *3.1.2 Uso de Kanban para la gestión del flujo de trabajo*

Si bien Scrum estableció el marco temporal del proyecto, también fue preciso gestionar de manera visual y constante las labores cotidianas. Para ello, se utilizó Kanban como soporte operativo. Se empleó un tablero estructurado en las columnas To Do, In Progress, In Review y Completed, lo cual posibilitó observar el estado de cada actividad con claridad y propició el monitoreo del flujo de trabajo. Este método mejoró la coordinación del equipo al evitar que se acumularan tareas en etapas críticas, como la validación de modelos o el desarrollo de componentes backend. Además, se establecieron límites de trabajo en progreso (WIP) para prevenir la sobrecarga de tareas en una misma etapa, fomentar que las actividades se terminen antes de comenzar otras nuevas y asegurar un flujo de trabajo más eficiente y estable durante todo el proyecto.

Kanban también facilitó la priorización dinámica de tareas, especialmente en momentos donde la experimentación del modelo arrojaba resultados inesperados y se debía ajustar la ingeniería de características o incluir nuevas métricas de evaluación.

### *3.1.3 Integración Scrum + Kanban (Scrumban)*

Por varias características propias del proyecto, se considera apropiado el empleo de una metodología híbrida que combine Scrum y Kanban (Scrumban). Primero, es importante señalar que los proyectos de Machine Learning son inherentemente variables, porque el entrenamiento y la calibración de modelos no se produce en un proceso lineal. Además, los resultados dependen de aspectos como la calidad del conjunto de datos, la elección de características, los algoritmos analizados y los hiperparámetros probados. En esta situación, Scrum posibilita la redefinición de metas al comienzo de cada sprint; Kanban, en cambio, posibilita la reorganización dinámica de tareas cuando los resultados logrados no son los deseados.

En segundo lugar, el proyecto necesita la provisión ininterrumpida de componentes funcionales, sobre todo al relacionarse con los interesados o los usuarios que no son técnicos. Para ellos, ver adelantos concretos, como vistas funcionales del portal web, dashboards, informes ejecutivos o explicaciones fundamentadas en SHAP, es crucial. Scrum garantiza que se produzcan aumentos funcionales en ciclos breves, normalmente de dos semanas, lo cual promueve una retroalimentación constante y temprana.

Además, el proyecto tiene una complejidad multidisciplinaria elevada, ya que engloba campos como la ciencia de datos, la seguridad en los bancos, el aprendizaje automático, la explicación de modelos y el desarrollo web a nivel completo. La combinación de Kanban y Scrum permite la coordinación efectiva de estas disciplinas, conservando un balance entre la flexibilidad operativa y el orden metodológico. Además, la

gestión del riesgo y la disminución de la incertidumbre son cruciales, ya que en los modelos de fraude modificaciones menores pueden tener un impacto importante en los resultados. El uso de metodologías ágiles permite el aprendizaje a partir del error, la experimentación controlada y el progreso constante.

En último término, teniendo en cuenta los roles y tiempos de un equipo académico, es absolutamente necesario disponer de métodos explícitos para la asignación y el seguimiento de actividades. Según Molnar (2023), la utilización del tablero Kanban posibilitó determinar de manera transparente quién estaba a cargo de cada tarea, mientras que Scrum mejoró el cumplimiento de los objetivos del proyecto y la organización del trabajo en equipo al permitir definir metas realistas para cada sprint.

### *3.1.4 Tiempos y roles del proyecto mediante Kanban*

Como complemento a la metodología ágil adoptada, se utilizó un tablero Kanban como herramienta de soporte para la gestión operativa de las actividades del proyecto. Este enfoque permitió visualizar de manera clara y estructurada el estado de cada tarea a lo largo del ciclo de desarrollo, facilitando el control del avance y la coordinación entre las distintas fases del proyecto.

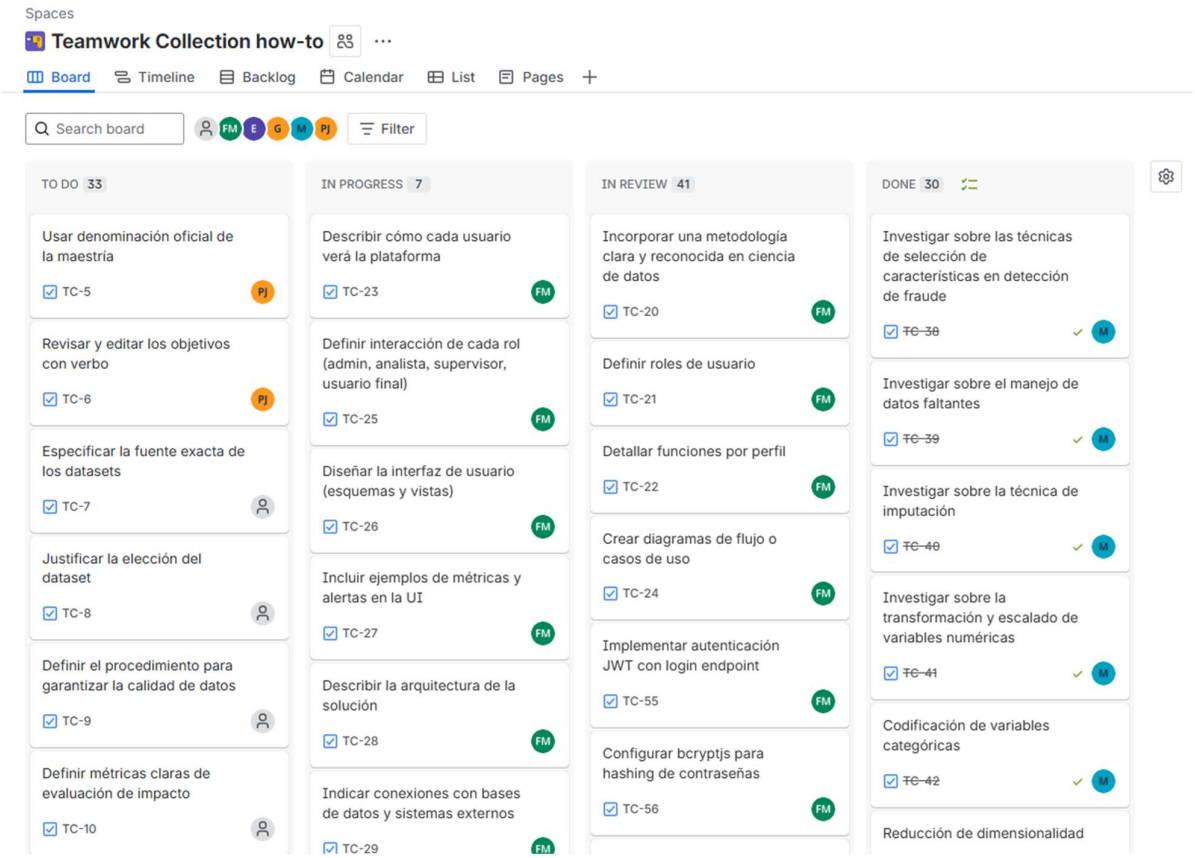
El tablero Kanban se organizó en las columnas *To Do*, *In Progress*, *In Review* y *Completed*, lo que permitió clasificar las actividades según su estado de ejecución. A través de esta estructura, se gestionaron tareas relacionadas con el preprocesamiento de datos, la experimentación y entrenamiento de modelos de aprendizaje automático, el desarrollo del portal web y la generación de reportes explicables.

La aplicación de Kanban contribuyó a mejorar la eficiencia del flujo de trabajo, reducir acumulaciones de tareas en etapas críticas y permitir una priorización dinámica, especialmente en actividades exploratorias propias de proyectos de Machine Learning. Asimismo, esta herramienta

facilitó la identificación temprana de bloqueos y el seguimiento continuo del progreso, garantizando una ejecución ordenada y alineada con los objetivos definidos en cada iteración.

**Figura 2**

*Tablero Kanban utilizado para la gestión del proyecto*




*Nota:* Evolución de las tareas del proyecto divididas en columnas de estado (Pendiente, En curso, Revisión y Completado). Esta metodología ágil asegura que las métricas de clasificación y los análisis de interpretabilidad se completen antes del despliegue final.

*Fuente:* Elaboración propia

### 3.2 Experiencia de Usuario y Perfiles de Acceso

El diseño de la plataforma FraudOps Portal no solo se basó en criterios técnicos, sino también en la necesidad de garantizar que cada usuario, según su rol institucional, pudiera utilizar

la herramienta de manera intuitiva, segura y alineada con sus responsabilidades operativas. Para lograrlo, se definió un modelo de perfiles de acceso que determina la experiencia, las vistas habilitadas y las acciones disponibles para cada tipo de usuario.

Este enfoque evita que todos los usuarios reciban la misma información, reduciendo la sobrecarga cognitiva, minimizando el riesgo de exposición de datos sensibles y permitiendo que la interfaz se adapte al nivel técnico y funcional de cada grupo. La experiencia final busca mantener claridad, simplicidad y pertinencia para cada rol, especialmente en un sistema donde coexisten análisis técnicos, revisión operativa y supervisión ejecutiva.

### *3.2.1 Administrador*

El administrador tiene la responsabilidad de garantizar que el portal funcione adecuadamente, gestionar a los usuarios y configurar el sistema en términos generales. Su experiencia en la plataforma se centra en el mantenimiento, la gobernanza y las tareas de control. Cuando se accede, se tiene acceso a un panel dividido en secciones administrativas que permite crear, actualizar y asignar roles de usuarios; consultar registros de auditoría y actividad del sistema; ajustar parámetros relacionados con modelos, accesos e informes; además de observar el estado general y hacer seguimiento operativo del sistema. Dado que su función no es analítica ni operativa, este rol no tiene acceso a alertas de fraude.

### *3.2.2 Supervisor*

El Supervisor está a cargo de comprobar las alertas que los modelos de detección de fraude generan y de examinar los aspectos operacionales de cada transacción indicada. Cuando inicias sesión, entras directamente a un panel principal con alertas priorizadas en tiempo real y filtros avanzados para la gestión de casos. Además, cuenta con paneles de explicabilidad que utilizan métodos como SHAP o LIME, datos sobre la condición de cada alerta (en revisión, validada,

nueva o descartada) y alternativas para marcar, cerrar o escalar casos. Su interfaz está optimizada para el análisis veloz, la interacción constante con el sistema y la toma de decisiones.

### *3.2.3 Coordinador del proyecto*

El Coordinador del Proyecto tiene una perspectiva ejecutiva que se enfoca en el monitoreo general de la conducta del sistema y de cómo los modelos de fraude funcionan. Esta perspectiva comprende análisis de tendencias por tipo de fraude o categorías, comparaciones entre diferentes períodos, y herramientas para la creación de informes ejecutivos que se pueden descargar. También incluye métricas agregadas como recall, precisión y AUC-PR. Su interfaz se enfoca en la visualización y síntesis nítida de indicadores estratégicos, sin entrar a revisar transacciones individuales, lo que facilita la toma de decisiones y el progreso constante del proceso institucional.

### *3.2.4 Usuarios Visualizadores*

Los Usuarios Visualizadores están dirigidos a la consulta de datos no técnicos que han sido validados con anterioridad. Este rol tiene acceso limitado a informes aprobados, estadísticas descriptivas y gráficos simplificados, que se muestran en una interfaz sencilla donde no aparecen detalles técnicos o delicados. Por lo tanto, se permite que personal no especializado participe sin que la seguridad ni la confidencialidad del sistema se vean comprometidas.

### *3.2.5 Encuestadores / Recolectores de datos*

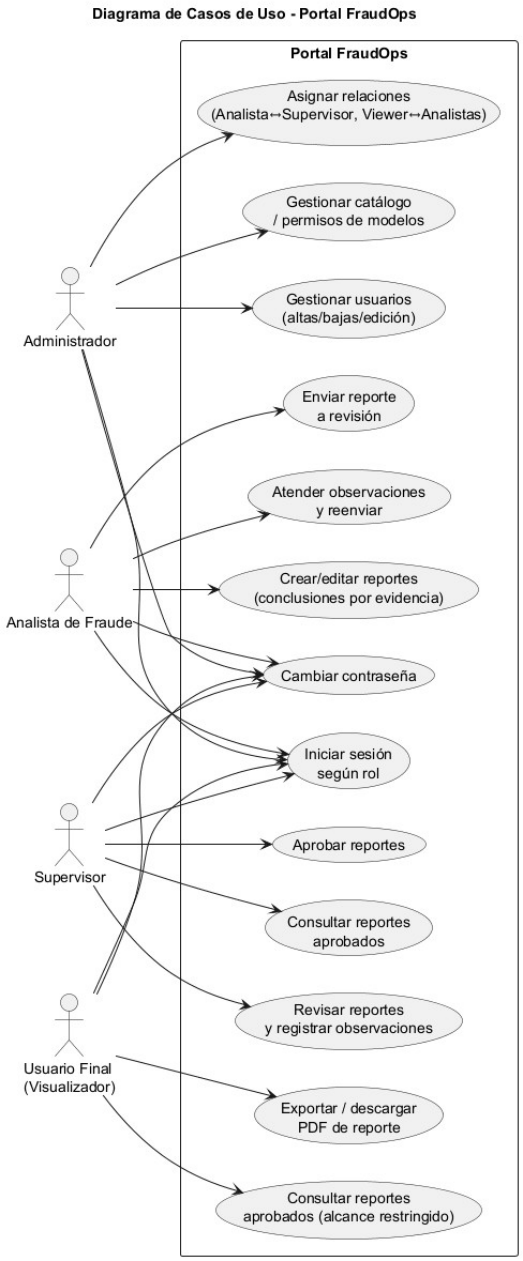
Cuando el proceso lo exige, los encuestadores o recolectores de datos tienen acceso limitado y específico para ingresar información. Su interacción con la plataforma se limita a consultar el historial de sus propias entradas, utilizar formularios estructurados y aplicar validaciones sencillas que aseguran que los datos permanezcan íntegros. Este rol no puede acceder a reportes ni a módulos de análisis, lo que respalda el principio del privilegio mínimo (Ribeiro, 2016).

### *3.2.6 Justificación del enfoque de vistas personalizadas*

La aplicación de vistas personalizadas responde a metas esenciales para la seguridad y el rendimiento del sistema. Primero, evita el acceso no autorizado a información confidencial o técnica y, por ende, brinda una protección rigurosa de los datos delicados. En segundo lugar, al mostrar solo la información pertinente para cada rol, optimiza la carga cognitiva, lo que mejora la usabilidad y disminuye los fallos operativos. Además, ayuda a la trazabilidad operativa, respaldando procedimientos de control interno y auditoría, y se mantiene en consonancia con las prácticas de seguridad bancaria, según las cuales la información debe ser diferenciada y supervisada. En resumen, esta definición de perfiles y experiencias de usuario brinda al diseño del FraudOps Portal seguridad, solidez y claridad, lo que contribuye a establecer una arquitectura enfocada en el usuario y encaminada a la reducción eficaz del fraude.

**Figura 3**

*Diagrama de Casos de Uso - Portal FraudOps*

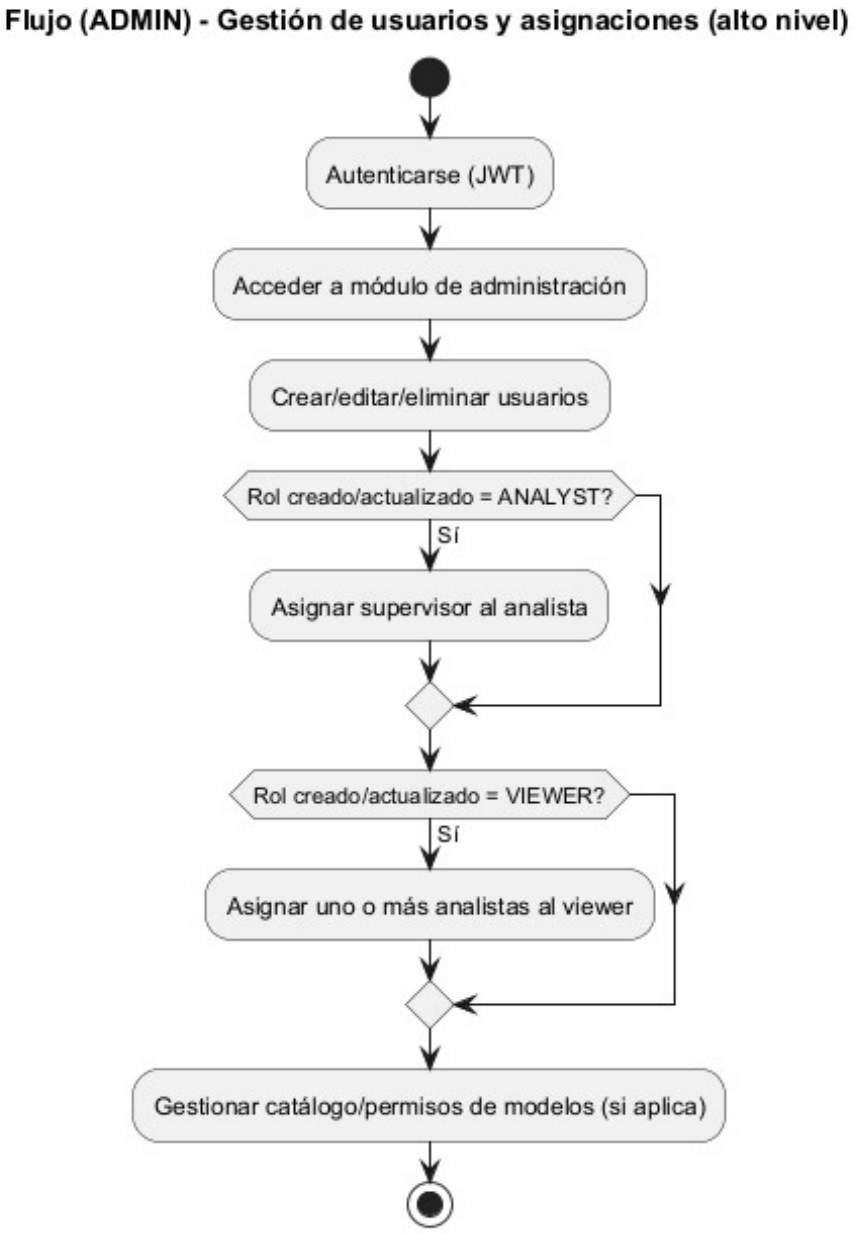



*Nota:* El diagrama ilustra las interacciones entre los actores principales (Analistas Antifraude y Administradores) y las funcionalidades del sistema Portal FraudOps. Se destacan los procesos de visualización de la matriz de confusión, generación de explicaciones locales con LIME y el monitoreo de métricas de rendimiento en tiempo real.

*Fuente:* Elaboración Propia

**Figura 4**

*Flujo del rol ADMIN: gestión de usuarios, roles y asignaciones en FraudOps*



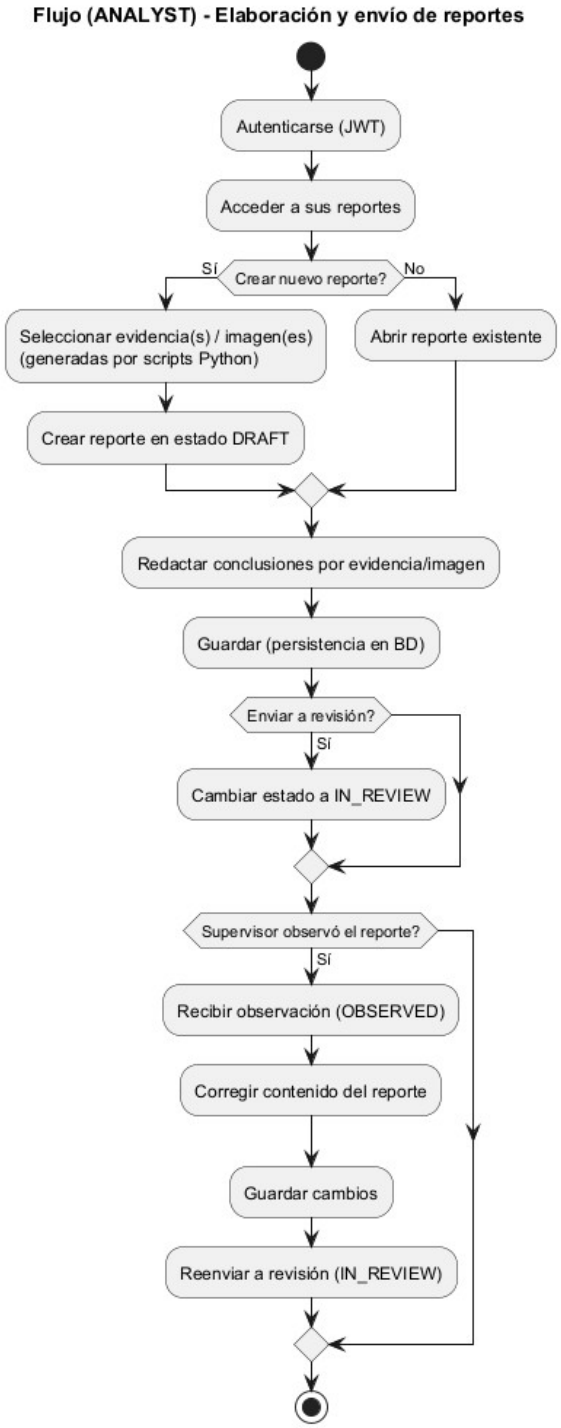

*Nota:* El diagrama describe la secuencia lógica de los procesos de administración de identidades. Incluye la creación de perfiles de usuario, la definición de privilegios de acceso (RBAC) y la

asignación de analistas a casos específicos de fraude, garantizando la trazabilidad y la seguridad en la gestión operativa del portal.

*Fuente:* Elaboración Propia

**Figura 5**

*Flujo del rol ANALYST: elaboración, edición y envío de reportes en FraudOps*

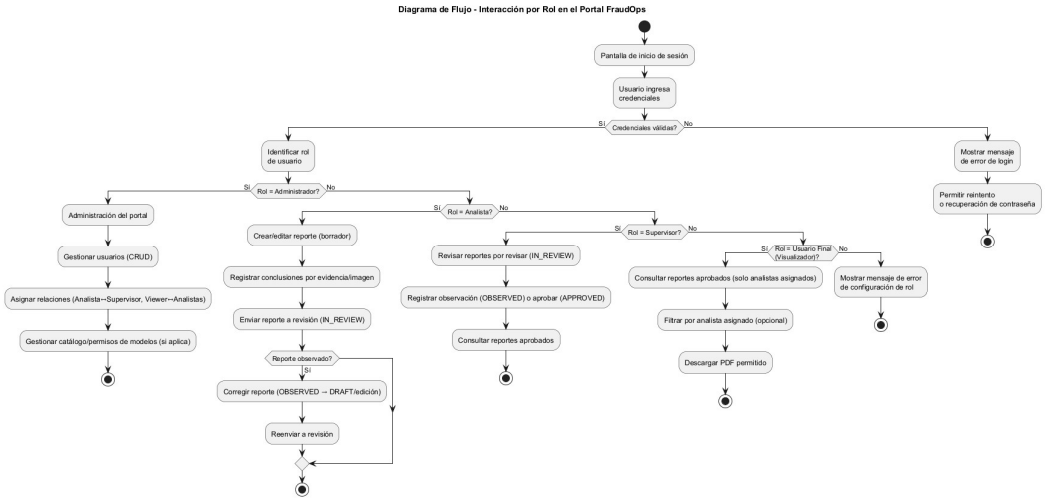


*Nota:* El diagrama detalla el proceso cíclico realizado por el analista, que inicia con la interpretación de las métricas de clasificación y las explicaciones de LIME, y culmina con la generación de reportes detallados. Este flujo asegura que los hallazgos de fraude sean documentados y comunicados de manera estandarizada para la toma de decisiones.

*Fuente:* Elaboración Propia

**Figura 6**

*Diagrama general del flujo de interacción por rol en el Portal FraudOps*

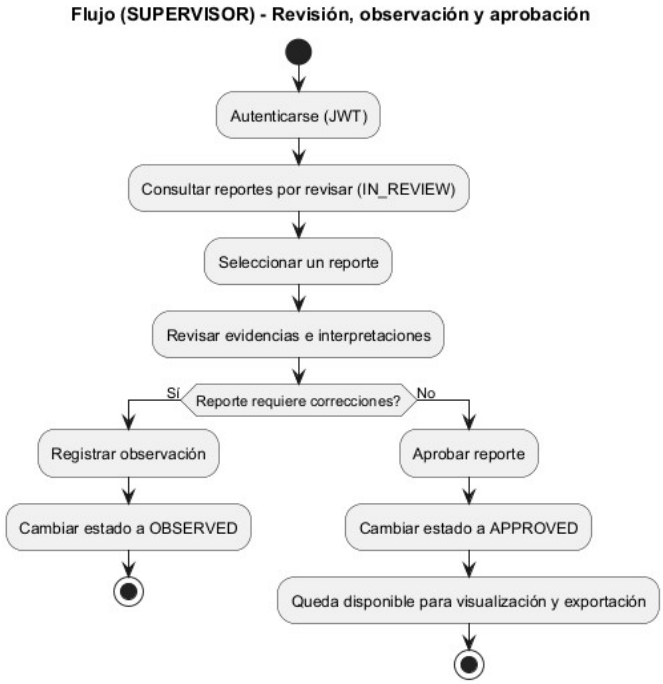



*Nota:* El diagrama presenta la arquitectura de interacción global del sistema, integrando las funciones de gobernanza del Administrador con las capacidades operativas del Analista. Se visualiza cómo la plataforma centraliza los datos de clasificación y las herramientas de explicabilidad, permitiendo un flujo continuo desde la gestión de acceso hasta la emisión de reportes técnicos de fraude.

*Fuente:* Elaboración Propia

**Figura 7**

*Flujo del rol SUPERVISOR: revisión, observación y aprobación de reportes en FraudOps*

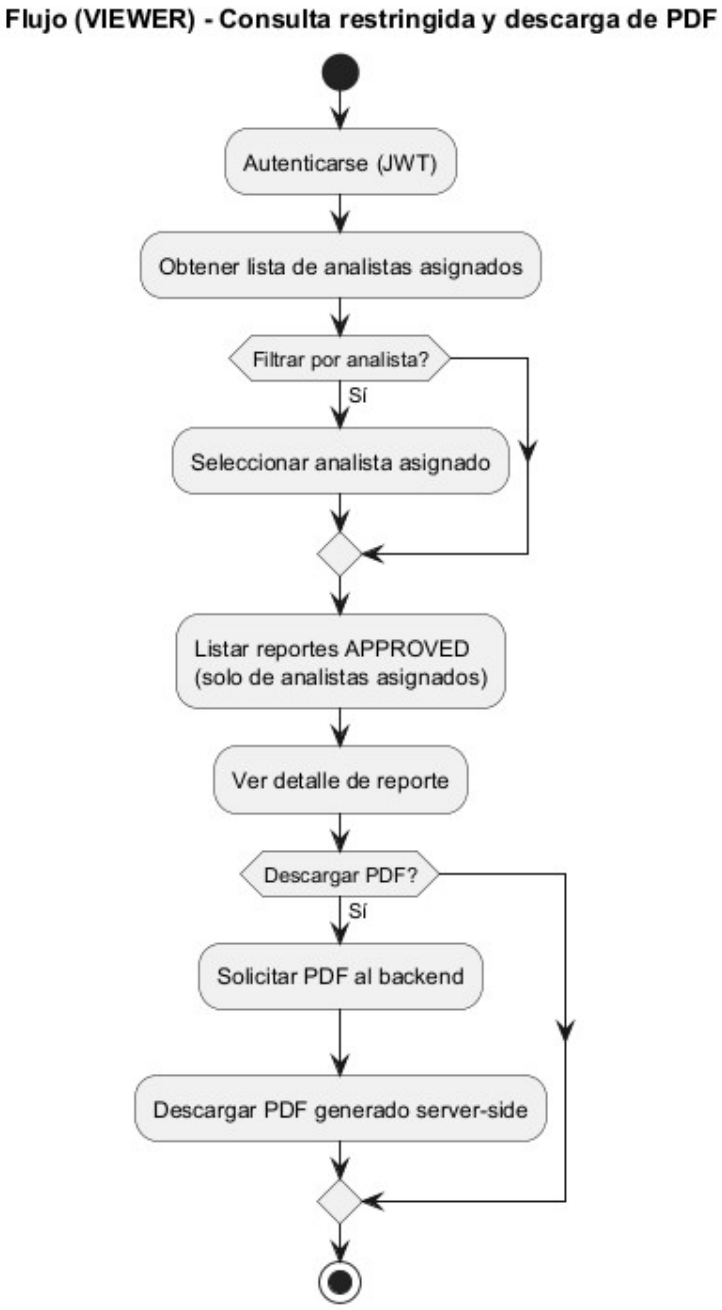



*Nota:* Secuencia de revisión, retroalimentación y validación de informes por parte del rol de supervisión, destacando el flujo de aprobación necesario para el cierre de casos sospechosos en el sistema.

*Fuente:* Elaboración Propia

**Figura 8**

Flujo del rol VIEWER: consulta restringida y descarga de reportes en PDF en FraudOps

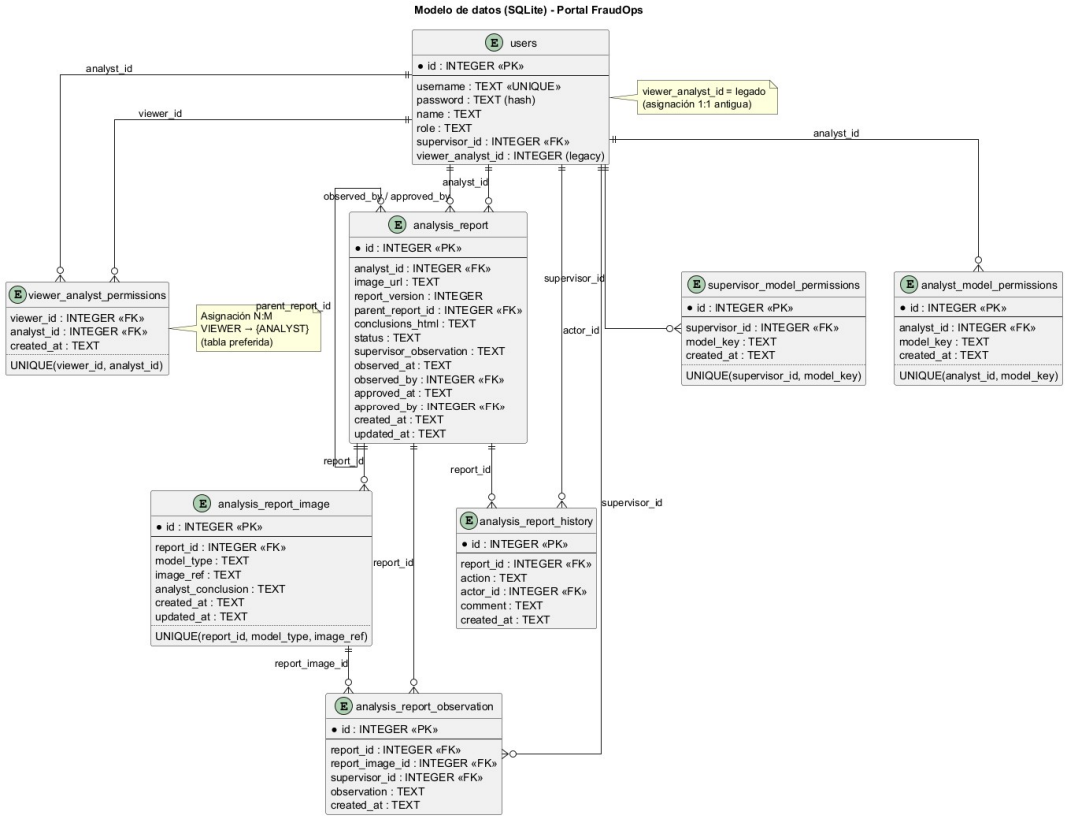


*Nota:* El diagrama presenta el flujo de acceso limitado para el rol de consulta. Se destaca la restricción de privilegios que impide la edición de datos, permitiendo únicamente la visualización de tableros y la exportación de reportes finalizados en formato PDF para fines de auditoría o información gerencial.

*Fuente:* Elaboración Propia

**Figura 9**

*Modelo de datos (SQLite) del Portal FraudOps*



*Nota:* Estructura de tablas y llaves primarias/foráneas del motor de base de datos SQLite. Este modelo constituye el núcleo de información para la gestión de usuarios y la documentación de casos de fraude dentro del portal.

*Fuente:* Elaboración Propia

## 3.3 Selección de la base de datos

### *3.3.1 Descripción del Conjunto de Datos*

Se seleccionaron datos del repositorio público Kaggle para desarrollar y entrenar los modelos de aprendizaje automático propuestos. La base es el simulador PaySim. Se eligió este conjunto de datos porque representa transacciones de pago en línea que cumplen con el objetivo general del estudio.

### *3.3.2 Dimensiones y Características*

El conjunto de datos original consta de un total de 6.362.620 registros de transacciones y tiene 11 columnas. Cada entrada representa una transacción separada representada por la variable 'step', que es una unidad de tiempo secuencial, donde cada paso corresponde a una hora en tiempo real.

**Tabla 3**

*Diccionario de Variables*

| ***Variable*** | ***Tipo de Dato*** | ***Descripción*** |
|---|---|---|
| ***step*** | Numérico (Discreto) | Representa la unidad de tiempo en la simulación. 1 *step* equivale a 1 hora de tiempo real. |
| ***type*** | Categórico (Nominal) | Tipo de transacción realizada donde existen 5 tipos: *CASH-IN, CASH-OUT, DEBIT, PAYMENT* y *TRANSFER*. |
| ***amount*** | Numérico (Continuo) | El monto monetario de la transacción en moneda local. |

| | | |
|---|---|---|
| ***nameOrig*** | Categórico (String) | Identificador único del cliente que inicia la transacción (Origen). |
| ***oldbalanceOrg*** | Numérico (Continuo) | Saldo disponible en la cuenta de origen **antes** de realizar la transacción. |
| ***newbalanceOrig*** | Numérico (Continuo) | Saldo resultante en la cuenta de origen **después** de realizar la transacción. |
| ***nameDest*** | Categórico (String) | Identificador único del destinatario de la transacción (Destino). |
| ***oldbalanceDest*** | Numérico (Continuo) | Saldo en la cuenta de destino **antes** de recibir la transacción. (Nota: Puede ser 0 si el destinatario es un comercio nuevo). |
| ***newbalanceDest*** | Numérico (Continuo) | Saldo resultante en la cuenta de destino **después** de la transacción. |
| ***isFraud*** | Numérico (Binario) | **Variable Objetivo (Target).** Indica si la transacción es fraudulenta.<br>• **1:** Transacción fraudulenta.<br>• **0:** Transacción legítima. |
| ***isFlaggedFraud*** | Numérico (Binario) | Variable de control del sistema de simulación. Marca automáticamente transferencias masivas (generalmente superiores a 200.000) en un solo intento. |

***Fuente:*** *Elaboración Propia*

### *3.3.3 Definición de la Variable Objetivo*

Se identificó como variable dependiente a 'isFraud'. Esta variable es binaria, donde un valor de 1 representa una transacción fraudulenta y un valor de 0 representa una transacción legítima.

### *3.3.4 Balance de Clases*

El conjunto de datos presenta un fuerte desequilibrio de clases. Las transacciones fraudulentas solo representan el 0,1% de los datos totales. Esta propiedad se abordará más adelante en el paso de preprocesamiento utilizando técnicas de equilibrio.

### *3.3.5 Preprocesamiento de Datos*

Dado que la calidad de las predicciones depende directamente de la calidad de los datos de entrada, se aplicaron las siguientes técnicas de limpieza, transformación y reducción."

### *3.3.6 Limpieza y Selección de Características*

Se realizó un análisis de relevancia de atributos. Las variables 'nameOrig' y 'nameDest' fueron eliminadas. Debido a que son identificadores específicos, no proporcionan patrones generalizables y su inclusión puede conducir a una redundancia de modelos. Además, la variable isFlaggedFraud se eliminó porque es una regla estática en el marco de simulación y no una característica que el modelo debe aprender.

Ingeniería de Características:

Para evitar el ruido y la redundancia (colinealidad), se eliminaron las siguientes variables:

- **Identificadores (nameOrig, nameDest):** Variables categóricas de cardinalidad única que no aportan patrones generalizables.
- **Variables Redundantes (newbalanceOrig, newbalanceDest, oldbalanceOrg, oldbalanceDest):** Al haber calculado la magnitud del error y tener el saldo inicial, el saldo final se vuelve información repetitiva matemáticamente.
- **Variable step:** Fue transformada a una variable cíclica (hora_del_dia) para capturar patrones temporales de comportamiento, eliminando la columna original y también se creó la variable día_del_mes.

### *3.3.7 Transformación de Variables*

Se observaba que las variables monetarias (como montos y saldos) mostraban una distribución muy heterogénea: la mayoría de las transacciones eran de bajo valor, mientras que algunas alcanzaban cifras extremas. Para corregir esta asimetría se aplicó una transformación

logarítmica ($\ln(x+1)$) a estas columnas. Esta variante particular, que suma 1 al valor original, era necesaria para manejar las numerosas entradas con saldo cero, evitando así el error matemático que se produciría al calcular el logaritmo de cero. Se decidió excluirlo de esta transformación para no distorsionar la secuencia temporal y mantener la distancia uniforme de las horas, fundamental para detectar patrones cíclicos de fraude.

#### *3.3.8 Codificación de Variables Categóricas (Encoding)*

Los algoritmos de aprendizaje automático requieren entrada numérica para realizar operaciones matemáticas. La variable categórica 'Type' (Tipo de transacción) se convirtió mediante la técnica One-Hot Encoding donde cada categoría es una nueva columna binaria (0 o 1). Esto evita que el modelo asuma erróneamente un orden jerárquico entre tipos de transacciones que ocurriría si se asignaran números secuenciales.

#### *3.3.9 Escalado de Datos*

Las variables numéricas del conjunto de datos, como 'amount' y 'oldBalanceOrg', tienen rangos de valores muy diferentes. Para evitar que la función de costos del algoritmo sea dominada por variables con magnitudes mayores, se utilizó una técnica de estandarización (StandardScaler). Este proceso ajusta las variables para que tengan una media de 0 y una desviación estándar de 1, asegurando que todas las características contribuyan por igual al aprendizaje del modelo.

**División del Conjunto de Datos**

Se dividió el conjunto de datos en dos subconjuntos:

- Conjunto de Entrenamiento (Training Set): Se dividió el 80% de los datos, utilizado para el ajuste de los parámetros.
- Conjunto de Prueba (Test Set): El restante del 20% de los datos será reservado para la evaluación final.

Debido al grave desequilibrio en la clase ‘isFraud’, se utilizó una división estratificada garantizando que la proporción de fraudes (clase minoritaria) siga siendo idéntica tanto en el entrenamiento como en las pruebas, evitando sesgos.

# CAPITULO 4

## 4. ANÁLISIS DE RESULTADOS

### 4.1 Análisis Exploratorio de Datos

Para el desarrollo de los modelos predictivos se utilizó el conjunto de datos Synthetic Financial Datasets For Fraud Detection, el cual simula transacciones financieras reales y ha sido ampliamente empleado en estudios de detección de fraude debido a su complejidad y realismo. Tras el proceso de limpieza, transformación y selección de variables descrito en el capítulo anterior, se obtuvo una base final compuesta por 6 363 620 registros, distribuidos en 5 090 096 observaciones para entrenamiento y 1 272 524 observaciones para prueba, manteniendo una división estratificada para preservar la proporción de fraudes.

Con el objetivo de comprender la estructura del conjunto de datos de transacciones bancarias y detectar patrones relevantes asociados al fraude, se realizó un Análisis Exploratorio de Datos (EDA) mediante gráficos de tipo univariado y bivariado.

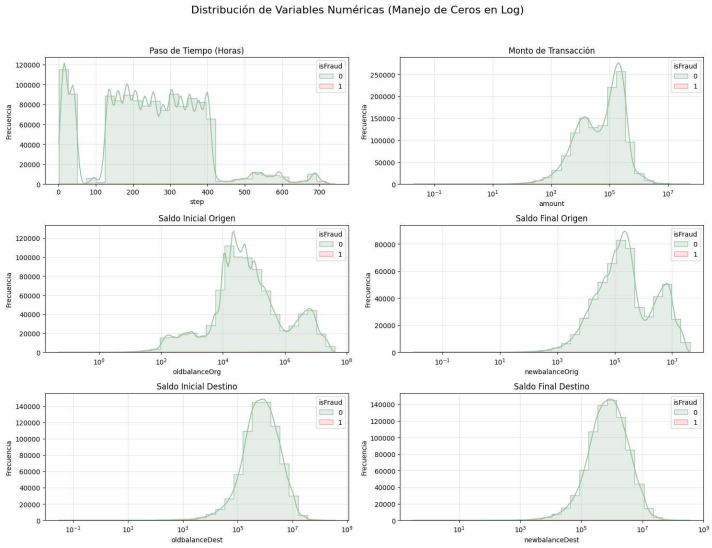
Este análisis permitió examinar la distribución de las variables numéricas, la presencia de desbalances de clase, relaciones entre variables y comportamientos temporales de las transacciones.

#### *4.1.1 Análisis univariado*

En la Figura 9 se presenta la distribución de las principales variables numéricas del dataset (step, amount, oldbalanceOrg, newbalanceOrg, oldbalanceDest, newbalanceDest), empleando escala logarítmica para manejar adecuadamente la alta dispersión y la presencia de valores extremos.

**Figura 10**

Distribución de Variables Numéricas (Manejo de Ceros en Log)




*Nota:* La figura muestra la distribución de las principales variables monetarias y temporales del conjunto de datos.

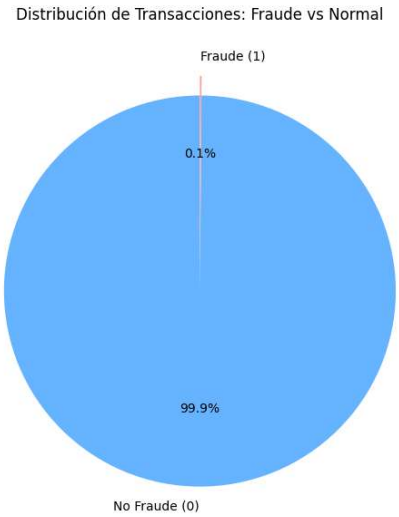
*Fuente:* Elaboración propia.

La variable amount muestra una distribución fuertemente asimétrica hacia la derecha, con una gran concentración de operaciones en cantidades pequeñas y una larga cola asociada a montos altos. De igual manera, los saldos de inicio y de cierre, tanto los que provienen como los que van a destino, muestran patrones de distribución coherentes entre ellos. Esto es consistente con la esencia contable de las transacciones bancarias. La variable step, que indica el tiempo en horas a lo largo de 30 días de simulación, presenta una distribución no uniforme, lo cual evidencia los patrones temporales y operativos del sistema financiero. En resumen, estas propiedades apoyan el uso de modelos robustos y la implementación de transformaciones logarítmicas en situaciones donde las distribuciones no son normales, con el objetivo de optimizar el rendimiento y la estabilidad del análisis.

La Figura 11 muestra la distribución de la variable objetivo isFraud, donde se evidencia un fuerte desbalance de clases, con aproximadamente 99.9% de transacciones no fraudulentas y solo 0.1% fraudulentas.

**Figura 11**

*Distribución de Transacciones: Fraude vs Normal*




*Nota:* Se evidencia un fuerte desbalance de clases en el conjunto de datos.

*Fuente:* Elaboración propia

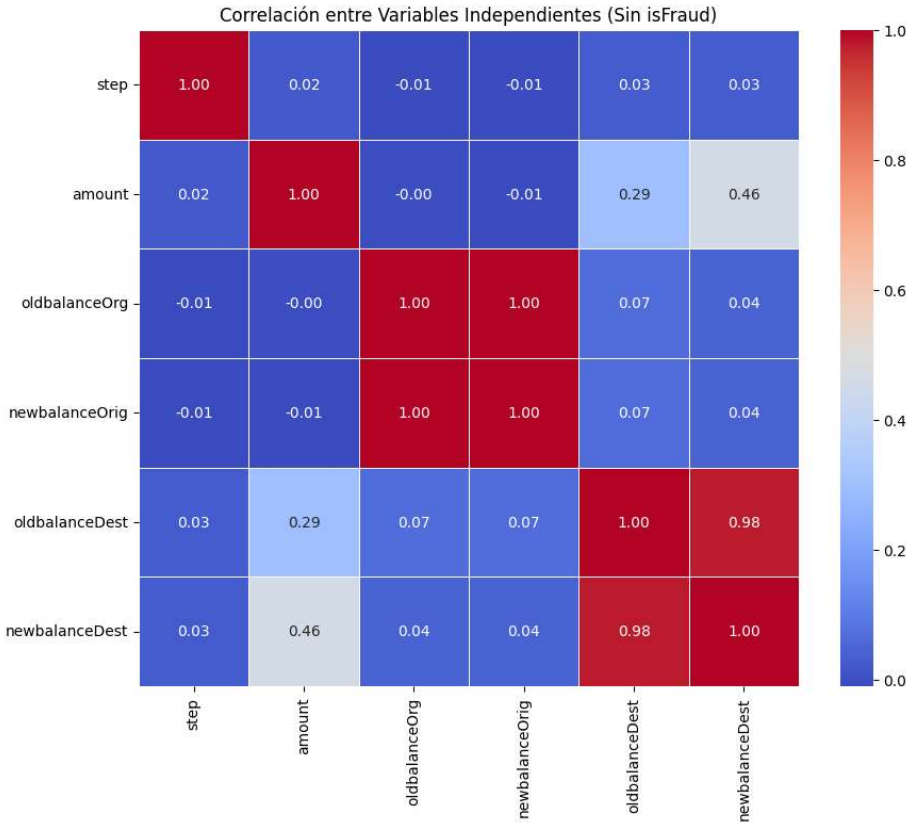
Este desbalance confirma la necesidad de aplicar técnicas específicas para clasificación en contextos de fraude.

### *4.1.2 Análisis Bivariado*

En la Figura 12 se presenta la matriz de correlación de Pearson entre las variables numéricas independientes (excluyendo la variable objetivo).

**Figura 12**

*Matriz de Correlación*



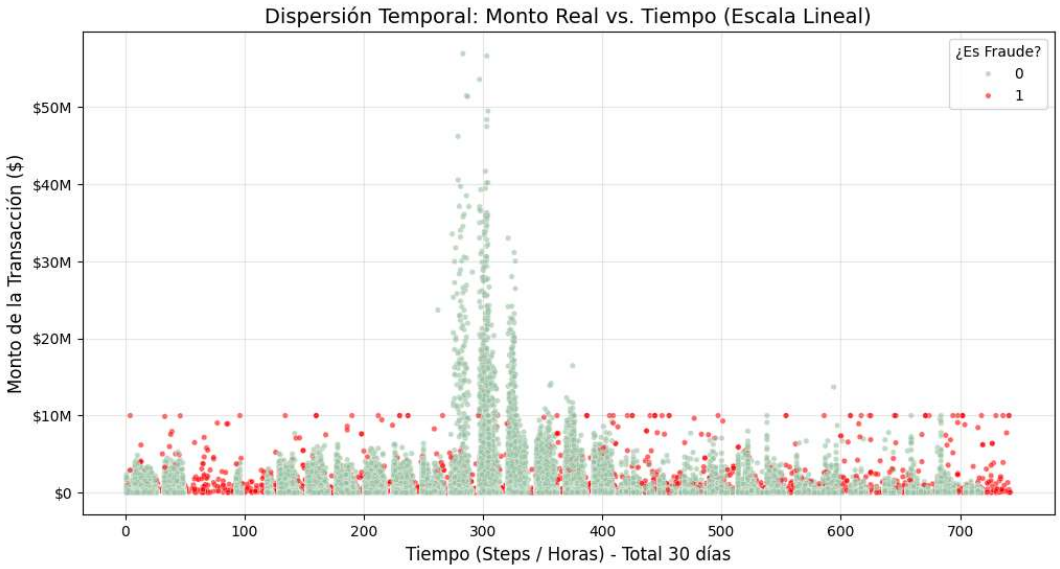

*Nota:* Se observan correlaciones altas entre variables de saldo, propias de la dinámica contable.

*Fuente:* Elaboración propia.

Existe una correlación prácticamente perfecta entre las variables oldbalanceOrg y newbalanceOrg (≈1.00), y de igual manera entre oldbalanceDest y newbalanceDest, lo que evidencia relaciones contables directas propias de la dinámica de las transacciones. Además, la variable amount muestra correlaciones moderadas con los saldos finales del destinatario, indicando que el monto transferido influye de forma apreciable en el balance resultante. Por otro lado, la variable step presenta correlaciones muy bajas con el resto de las variables, lo que sugiere una independencia temporal respecto a montos y balances. En conjunto, estas observaciones justifican la inclusión de dichas variables en los modelos propuestos, especialmente porque los algoritmos utilizados son menos sensibles a la multicolinealidad.

**Figura 13**

*Dispersión Temporal (Monto vs Tiempo)*



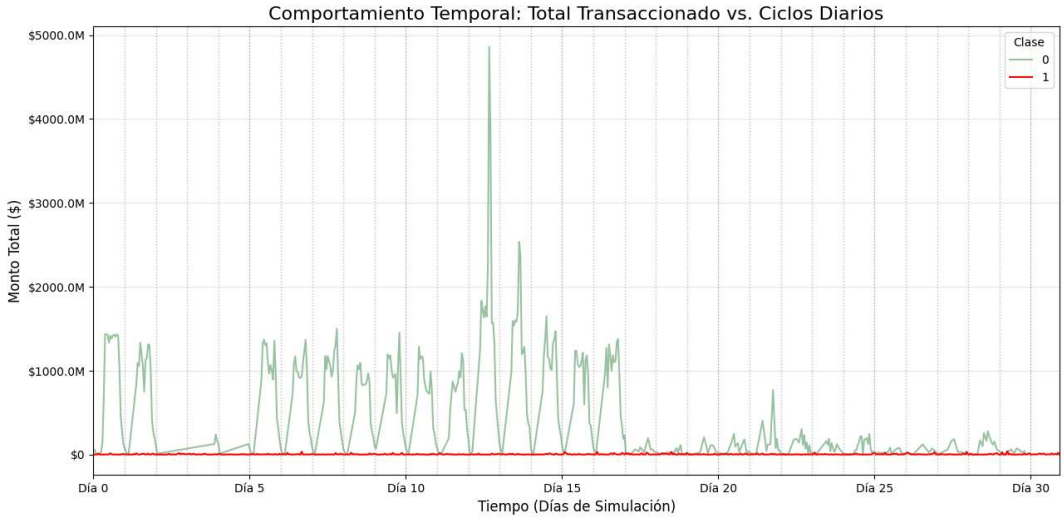

*Nota:* Las transacciones fraudulentas se concentran en montos elevados a lo largo de todo el periodo analizado.

*Fuente:* Elaboración propia.

Las transacciones catalogadas como fraudulentas se presentan de manera homogénea a lo largo de todo el periodo de simulación, sin evidenciar concentración en un intervalo temporal particular, aunque comparativamente muestran una mayor tendencia a asociarse con montos elevados frente a las transacciones normales. Esto sugiere que la variable temporal por sí sola no permite discriminar adecuadamente el fraude, pero al combinarse con variables monetarias puede aportar información relevante para mejorar la capacidad predictiva del modelo

**Figura 14**

*Comportamiento Temporal: Total Transaccionado vs Ciclos Diarios*



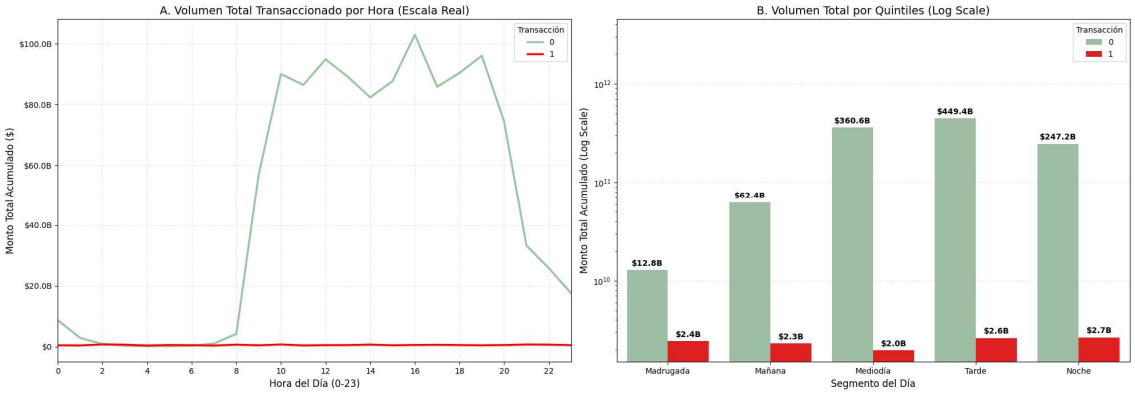

*Nota:* Se muestra el comportamiento de las transacciones a lo largo del tiempo, diferenciando entre transacciones fraudulentas y no fraudulentas. Fuente: Elaboración propia.

*Fuente:* Elaboración propia.

Se identifican variaciones significativas en el volumen total transaccionado a lo largo de los días, lo que refleja patrones operativos característicos del sistema financiero y la dinámica propia del flujo de pagos dentro del periodo analizado. En este contexto, las transacciones no fraudulentas concentran claramente el mayor volumen del total transaccionado, mientras que las fraudulentas representan una fracción mucho menor; sin embargo, su presencia se mantiene de forma constante y relativamente estable a lo largo del tiempo, evidenciando que, aunque el fraude tiene un impacto reducido en términos de volumen global, constituye un fenómeno persistente que debe ser considerado en el modelado predictivo.

**Figura 15**

*Comportamiento Temporal*



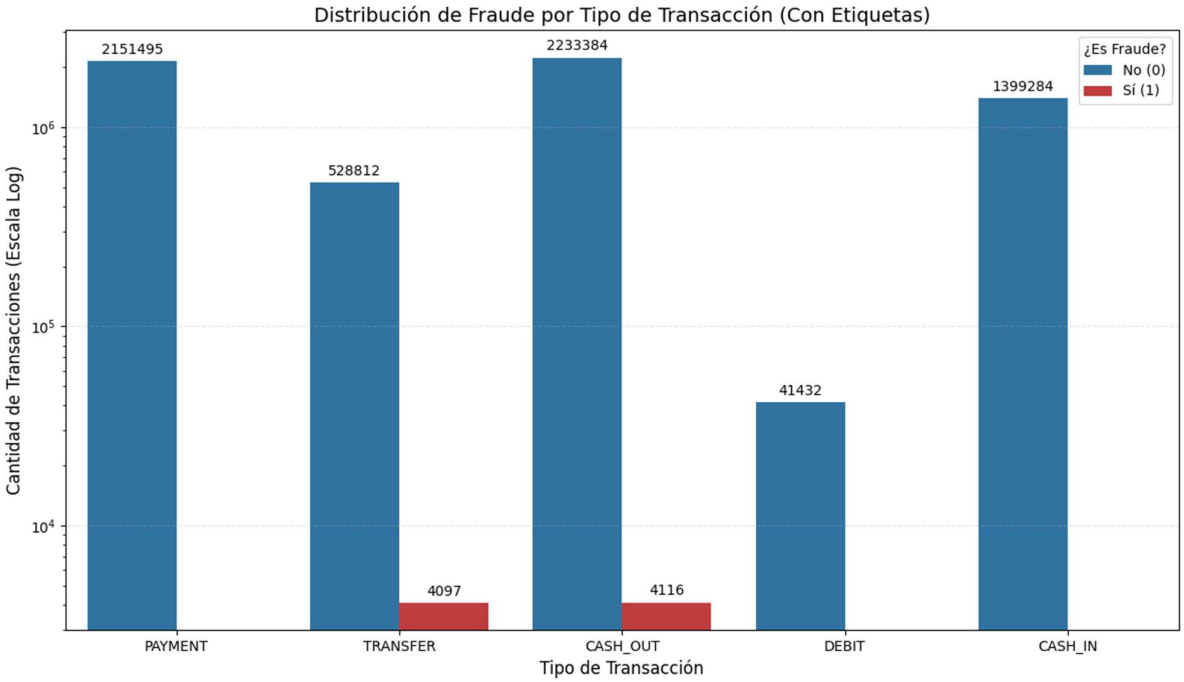

*Nota:* La figura muestra el volumen total transaccionado según la hora del día y por segmentos horarios, diferenciando entre transacciones fraudulentas y no fraudulentas, utilizando escala real y logarítmica.

*Fuente:* Elaboración propia.

Se observan picos de volumen transaccional en determinadas horas del día, lo que evidencia la existencia de horarios con mayor actividad financiera dentro del sistema analizado. A pesar de ello, las transacciones fraudulentas siguen un patrón horario similar al de las transacciones normales, aunque con volúmenes significativamente menores, lo cual resulta coherente con su baja frecuencia relativa. Además, el análisis por segmentos del día permite identificar diferencias claras en el volumen transaccional acumulado, destacándose como periodos de mayor actividad aquellos asociados a horarios laborales y comerciales. Sin embargo, las operaciones fraudulentas mantienen una presencia constante en todos los segmentos temporales, aunque con una magnitud considerablemente inferior frente a las operaciones no fraudulentas, lo que refuerza la utilidad del análisis temporal para comprender la dinámica general de los pagos.

**Figura 16**

*Distribución de Fraude por Tipo de Transacción*




*Nota:* La figura muestra el volumen total transaccionado según la hora del día y por segmentos horarios, diferenciando entre transacciones fraudulentas y no fraudulentas, utilizando escala real y logarítmica.

*Fuente:* Elaboración propia.

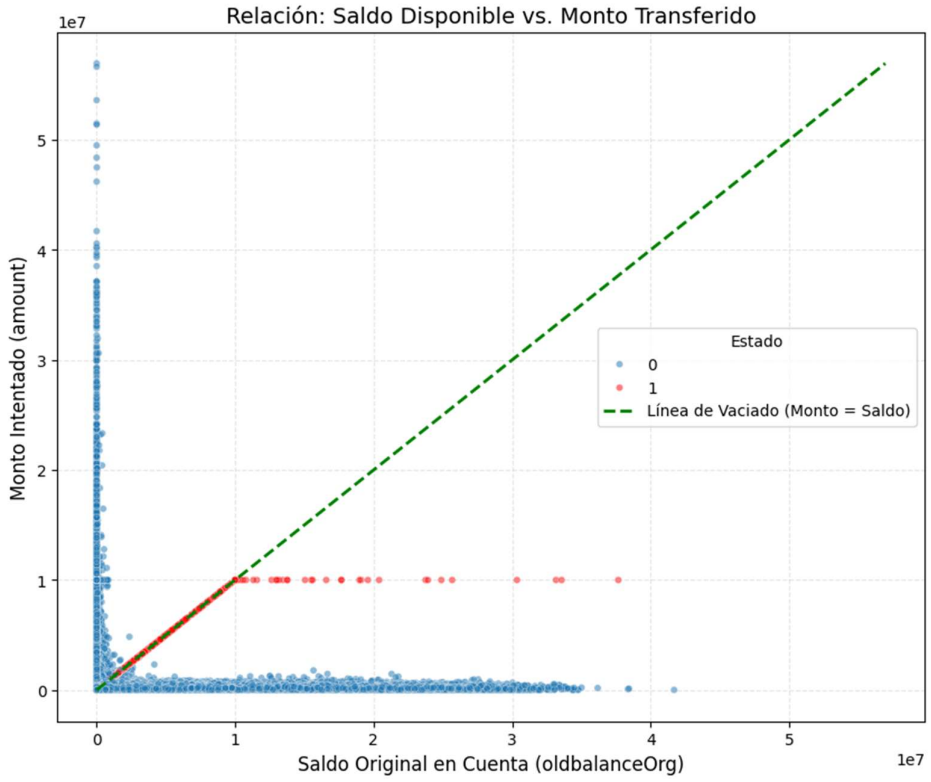
En la gráfica de barras Figura 15 se muestra la distribución de las transacciones según su tipo, diferenciando entre operaciones legítimas y fraudulentas. Dado el uso de una escala logarítmica en el eje vertical, es posible comparar categorías con volúmenes muy distintos sin que las clases minoritarias queden visualmente ocultas. Se observa que la mayoría de las transacciones pertenecen a tipos como *CASH_OUT* y *PAYMENT*, lo cual es coherente con el comportamiento típico de los sistemas financieros.

Sin embargo, al analizar específicamente los casos de fraude, se evidencia que estos se concentran de manera desproporcionada en ciertos tipos de transacción, particularmente en *CASH_OUT* y *TRANSFER*. Esta concentración sugiere que los fraudes no ocurren de forma

aleatoria, sino que están asociados a mecanismos específicos de movimiento de dinero que facilitan la extracción rápida de fondos. Por tanto, la variable tipo de transacción se perfila como un predictor relevante para los modelos de detección de fraude.

**Figura 17**

*Relación entre Saldo Disponible vs Monto Transferido*




*Nota:* La figura muestra el volumen total transaccionado según la hora del día y por segmentos horarios, diferenciando entre transacciones fraudulentas y no fraudulentas, utilizando escala real y logarítmica.

*Fuente:* Elaboración propia.

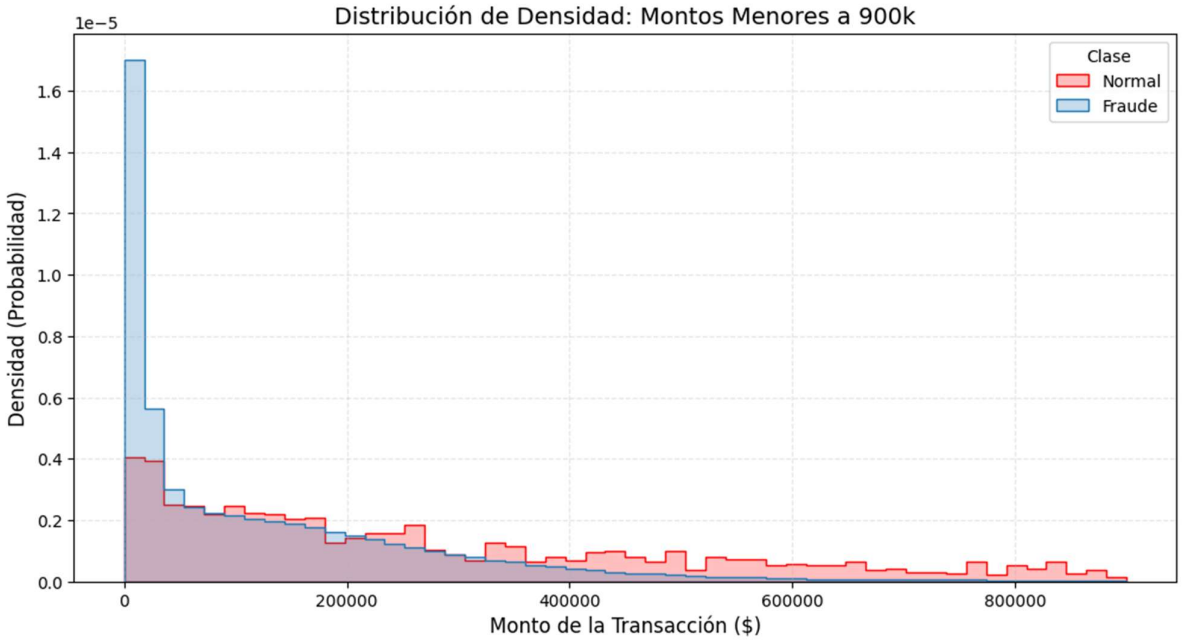
En la gráfica de dispersión Figura 16 se analiza la relación entre el saldo original de la cuenta (*oldbalanceOrg*) y el monto de la transacción (*amount*), incorporando una línea de

referencia donde el monto es igual al saldo disponible. Esta línea permite identificar visualmente transacciones en las que se intenta transferir la totalidad o una proporción significativa del saldo.

En las transacciones legítimas, los puntos tienden a concentrarse por debajo de dicha línea, lo cual indica que, en general, los usuarios no vacían completamente sus cuentas en una sola operación. En contraste, una proporción considerable de las transacciones fraudulentas se ubica sobre o muy cercana a la línea de vaciado, lo que evidencia intentos de extraer la mayor cantidad posible de fondos en una sola operación. Este patrón es consistente con el comportamiento típico de fraude financiero y confirma la relevancia de la relación entre saldo disponible y monto transferido como señal discriminante.

**Figura 18**

*Histograma de densidad para Montos menores a 900 mil*




*Nota:* La figura muestra el volumen total transaccionado según la hora del día y por segmentos horarios, diferenciando entre transacciones fraudulentas y no fraudulentas, utilizando escala real y logarítmica.

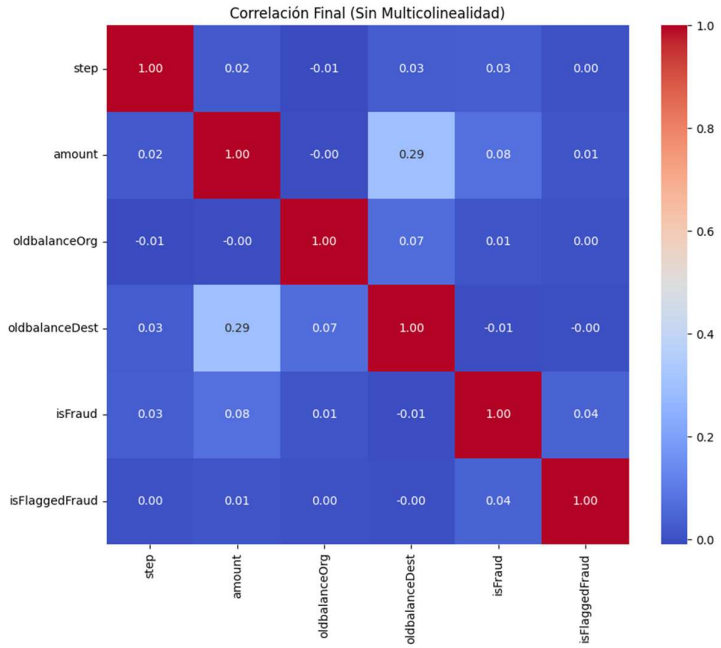
*Fuente:* Elaboración propia.

En el histograma de densidad Figura 17 se permite comparar la distribución probabilística de los montos para transacciones legítimas y fraudulentas dentro de un rango acotado, eliminando la influencia de valores extremadamente grandes. Al analizar este rango, se observa que las transacciones fraudulentas tienden a concentrarse en montos específicos, mostrando picos de densidad más pronunciados que los de las transacciones normales.

Por otro lado, las transacciones legítimas presentan una distribución más dispersa y uniforme, lo que refleja una mayor variedad de comportamientos normales. Esta diferencia en las distribuciones sugiere que, incluso en rangos de montos moderados, existen patrones característicos del fraude que pueden ser explotados por los modelos predictivos.

**Figura 19**

*Matriz de Correlación*



*Nota:* La figura muestra el volumen total transaccionado según la hora del día y por segmentos horarios, diferenciando entre transacciones fraudulentas y no fraudulentas, utilizando escala real y logarítmica.

*Fuente:* Elaboración propia.

En la matriz de correlación Figura 18 se muestra las relaciones lineales entre las variables numéricas del conjunto de datos tras el proceso de limpieza. En general, se observa que no existen correlaciones extremadamente altas entre las variables independientes, lo cual indica una baja presencia de multicolinealidad. Este resultado es deseable, ya que reduce el riesgo de inestabilidad en los modelos y facilita una mejor interpretación de la contribución individual de cada variable.

Además, la ausencia de correlaciones fuertes sugiere que cada variable aporta información complementaria al proceso de detección de fraude, reforzando la calidad del conjunto de datos para el entrenamiento de modelos supervisados.

## 4.2 Análisis de Resultados

### *4.2.1 Modelo XGBoost*

En la Tabla 4 se detallan los mejores hiperparámetros encontrados para este modelo, de igual manera se describe la efectividad del modelo a través de sus métricas de evaluación y su matriz de confusión.

**Tabla 4**

*Resultados de las métricas del modelo XGBoost*

| **Mejor valor obtenido en los hiperparámetros** | **Métricas de clasificación** | **Tasas** |
|---|---|---|
| **n_estimators = 300**<br>**max_depth = 6**<br>**learning_rate = 0.05**<br>**subsample = 0.8**<br>**colsample_bytree = 0.8**<br>**tree_method = hist** | **Accuracy: 0.9991**<br>**Precision (Fraude): 0.52**<br>**Recall (Fraude): 0.3780**<br>**F1-score (Fraude): 0.5303**<br>**AUC-ROC: 0.9677** | Falsos Negativos: 1022<br>Falsos Positivos: 207 |

*Nota: La figura muestra los diferentes hiperparámetros del modelo XGBoost, así como sus métricas de clasificación y las tasas de rendimiento. Fuente: Elaboración propia.*

El modelo XGBoost entrenado sobre los datos originales mostró un desempeño sólido y consistente, con una exactitud cercana al 99.9 % y un valor de ROC-AUC aproximado de 0.97, lo que evidencia una alta capacidad para diferenciar entre transacciones legítimas y fraudulentas. En particular, el modelo destacó por su elevada precisión en la detección de fraude, lo que indica que la mayoría de las alertas generadas corresponden efectivamente a casos reales. Sin embargo, el recall fue moderado, reflejando un enfoque más conservador, en el que se prioriza la reducción de falsos positivos, aun cuando esto implique no identificar la totalidad de los eventos fraudulentos.

Por otro lado, la incorporación de la técnica SMOTE permitió mejorar notablemente la detección de la clase minoritaria, incrementando el recall a valores superiores al 60 %. Esta mejora en sensibilidad vino acompañada de una disminución significativa en la precisión, lo que se tradujo en un mayor número de falsas alarmas. Aunque el ROC-AUC se mantuvo en niveles similares al modelo sin sobremuestreo, el descenso del F1-score evidenció un desequilibrio entre la capacidad de detección y la confiabilidad de las predicciones.

**Tabla 5**

*Comparación de resultados de ambos modelos XGBoost*

| Modelo | ROC_AUC | Average_Precision | Recall | F1-Score |
|---|---|---|---|---|
| **XGBoost** | 0.97 | 0.52 | 0.38 | 0.53 |
| **XGBoost + SMOTE** | 0.96 | 0.41 | 0.64 | 0.18 |

*Nota:* La figura muestra los diferentes hiperparámetros de los distintos modelos XGBoost, es decir, XGBoost Base y el XGBoost utilizando SMOTE.

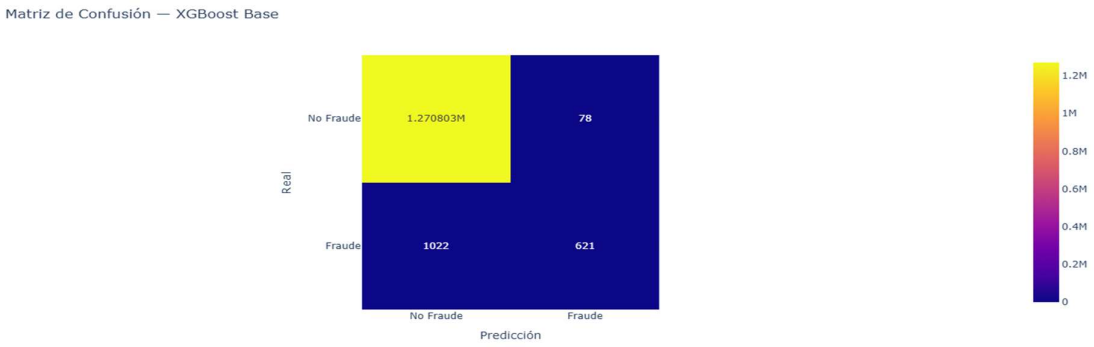
*Fuente:* Elaboración propia.

La comparación entre ambos enfoques pone de manifiesto el compromiso existente entre precisión y recall. Si bien el modelo con SMOTE resulta más agresivo en la identificación de fraudes, el modelo sin SMOTE ofrece un mejor balance general y un comportamiento más estable. Considerando que, en este contexto, las falsas alertas representan costos operativos relevantes, se

seleccionó el modelo XGBoost sin SMOTE como modelo final, ya que proporciona un desempeño más consistente y confiable, manteniendo una alta capacidad predictiva sin generar un volumen excesivo de alertas incorrectas.

**Figura 20**

*Matriz de confusión del modelo XGBoost sin SMOTE*




*Nota:* La figura muestra la matriz de confusión del modelo XGBoost en su forma base, mediante la cual puede medir cuantos aciertos realizó el modelo

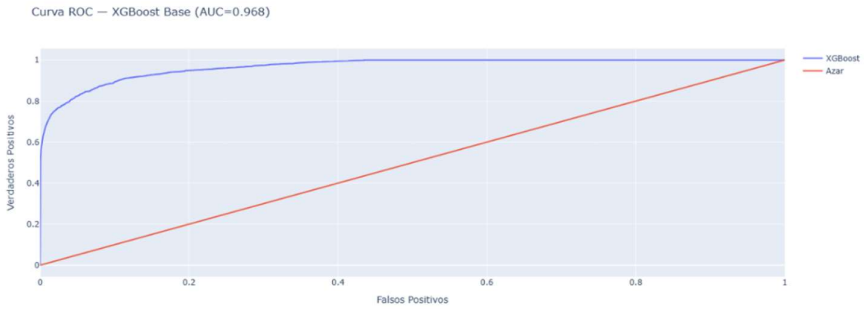
*Fuente:* Elaboración propia.

La matriz de confusión muestra un desempeño sólido del modelo en la clasificación de transacciones legítimas, con una tasa de aciertos muy elevada. En cuanto a la clase fraudulenta, el modelo logra identificar una proporción relevante de fraudes, aunque aún se presentan falsos negativos, lo cual es esperable dada la complejidad del problema y el fuerte desbalance de clases.

No obstante, se observa un número reducido de falsos positivos, lo que indica que el modelo evita penalizar innecesariamente transacciones legítimas.

**Figura 21**

*Curva ROC del modelo XGBoost sin SMOTE*




*Nota:* La figura muestra la curva ROC de modelo XGBoost en su forma base, mediante la cual puede medir la capacidad discriminativa del modelo

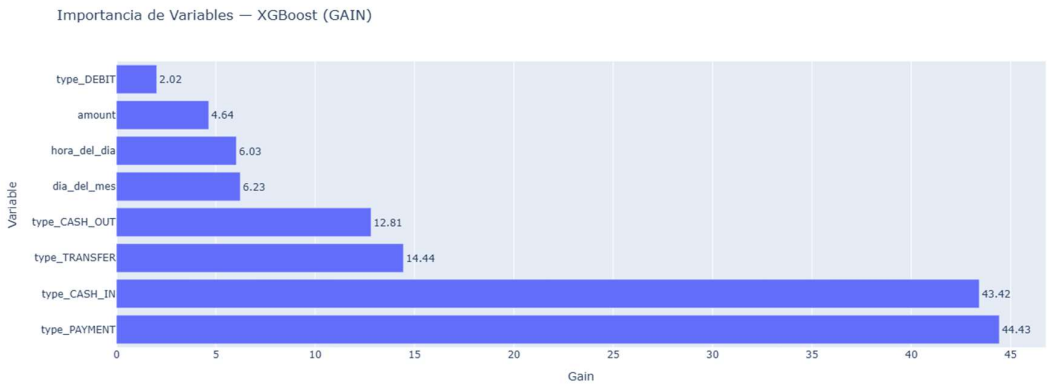
*Fuente:* Elaboración propia.

La curva ROC presenta un área bajo la curva elevada, lo que refleja una alta capacidad discriminativa del modelo para distinguir entre transacciones fraudulentas y no fraudulentas.

Este resultado indica que el modelo mantiene un buen equilibrio entre sensibilidad y especificidad a lo largo de distintos umbrales de decisión, confirmando su robustez para el problema analizado.

**Figura 22**

*Importancia de variables del modelo XGBoost*




*Nota:* La figura muestra la importancia que tienen las variables para el modelo XGBoost.

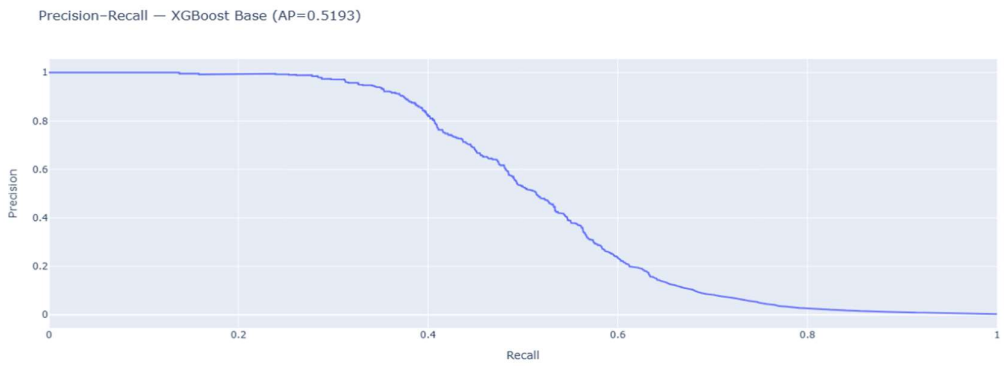
*Fuente:* Elaboración propia.

La gráfica de importancia de variables revela que un conjunto reducido de características concentra la mayor contribución al proceso de decisión del modelo. Variables relacionadas con el comportamiento transaccional y aspectos temporales destacan como las más influyentes.

Este resultado es coherente con la teoría acerca de la detección de fraude y aporta interpretabilidad al modelo, permitiendo comprender qué factores son determinantes en la clasificación.

**Figura 23**

Análisis del recall del modelo XGBoost sin SMOTE




*Nota:* La figura muestra el comportamiento de la curva Recall para los fraudes, mediante la cual se puede identificar las transacciones fraudulentas.

*Fuente:* Elaboración propia.

La gráfica muestra el comportamiento del recall para la clase fraudulenta, reflejando la capacidad del modelo para identificar correctamente las transacciones fraudulentas. Un valor de recall elevado indica que una mayor proporción de fraudes reales es detectada por el sistema, reduciendo el riesgo de omitir transacciones fraudulentas.

En el contexto de detección de fraude, esta métrica resulta especialmente relevante, ya que los falsos negativos representan un costo significativo para las entidades financieras. Los

resultados obtenidos evidencian que el modelo logra un nivel adecuado de sensibilidad, manteniendo al mismo tiempo un control razonable sobre la generación de falsos positivos.

Los resultados obtenidos confirman que XGBoost es un modelo adecuado para la detección de fraude financiero. Si bien el uso de técnicas de balanceo como SMOTE mejora la sensibilidad del modelo, también introduce un incremento considerable en los falsos positivos. En este contexto, el modelo XGBoost sin SMOTE ofrece un mejor compromiso entre desempeño predictivo y viabilidad operativa, consolidándose como la alternativa más apropiada para el problema analizado.

### *4.2.2 Modelo Árbol de Decisión*

En la Tabla 5 se detallan los mejores hiperparámetros encontrados y se reporta la efectividad del modelo a través de sus métricas de evaluación y su matriz de confusión.

**Tabla 6**

*Resultados de las métricas del modelo entrenado con Árbol de Decisión*

| Mejor valor obtenido en los hiperparámetros | Métricas de clasificación | Tasas |
|---|---|---|
| splitter: best | Accuracy: 1 | Falsos Negativos: 1008 |
| min_samples_split: 5 | Precision: 0.81 | Falsos Positivos: 149 |
| min_samples_leaf: 2 | Recall: 0.39 | |
| max_depth: 10 | F1-score: 0.52 | |
| criterion: entropy | Support: 1643 | |
| ccp_alpha: 0.0 | | |

*Nota:* La figura presenta los mejores hiperparámetros encontrados y el resultado obtenido para el modelo de árbol de decisión.

*Fuente:* Elaboración propia.

El modelo seleccionado como el de mejor desempeño corresponde al escenario sin aplicación de SMOTE, al obtener un F1-score de 0.52 para la clase minoritaria (fraude). El

accuracy del modelo alcanza valores cercanos al 99 %, resultado que es consistente con la marcada desproporción existente entre las clases, pero esta métrica por sí sola no resulta ser suficiente para evaluar el desempeño del modelo en la detección de fraude, ya que se ve fuertemente influenciada por la clase mayoritaria.

En el caso de la clase no fraudulenta, el modelo presenta un comportamiento prácticamente perfecto, con una precisión y F1-score de 1.00, y un recall de 1.00 indicando que el modelo puede identificar correctamente casi la totalidad de las transacciones legítimas, registrando una cantidad mínima de falsos positivos y evidenciando una alta confiabilidad en este tipo de predicción.

En el caso de la clase fraudulenta, que representa una proporción muy reducida del total de observaciones, el desempeño del modelo es más limitado. Se obtiene una precisión de 0.81, lo que indica que una alta proporción de las transacciones clasificadas como fraude corresponden efectivamente a fraudes reales. Sin embargo, el recall de 0.39 refleja que el modelo solo logra identificar una parte de los fraudes existentes, dejando una cantidad significativa de casos sin detectar mientras que el F1-score con un resultado de 0.52 evidencia un equilibrio moderado entre precisión y capacidad de detección para esta clase.

Los resultados muestran que el modelo es altamente eficaz para la identificación de transacciones legítimas y presenta una buena precisión en la clasificación de fraudes. Sin embargo, su capacidad para detectar la totalidad de los casos fraudulentos aún puede mejorarse.

**Figura 24**

*Matriz de clasificación del modelo entrenado con Árbol de Decisión*

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 1270881 |
| 1 | 0.81 | 0.39 | 0.52 | 1643 |
| accuracy | | | 1.00 | 1272524 |
| macro avg | 0.90 | 0.69 | 0.76 | 1272524 |
| weighted avg | 1.00 | 1.00 | 1.00 | 1272524 |

*Nota:* La figura presenta el reporte de clasificación del modelo Random Forest aplicado al conjunto de prueba, mostrando las métricas de precisión, recall, F1-score y soporte para cada clase, así como los promedios globales.

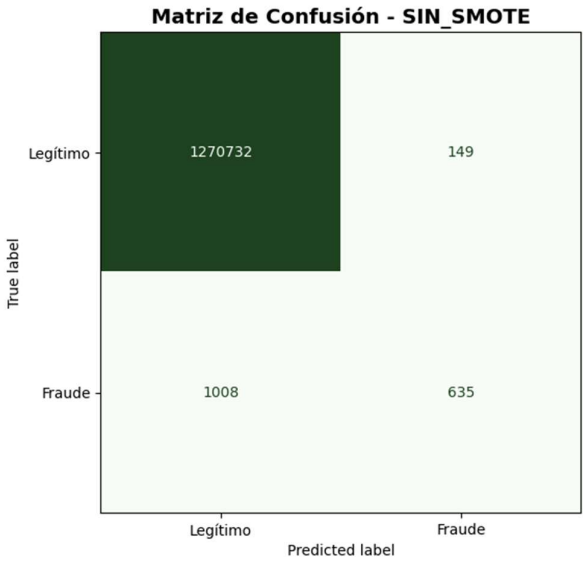
*Fuente:* Elaboración propia.

En el caso de la clase no fraudulenta, el modelo presenta un comportamiento prácticamente perfecto, con una precisión y F1-score de 1.00, y un recall de 1.00 indicando que el modelo puede identificar correctamente casi la totalidad de las transacciones legítimas, registrando una cantidad mínima de falsos positivos y evidenciando una alta confiabilidad en este tipo de predicción.

En el caso de la clase fraudulenta, que representa una proporción muy reducida del total de observaciones, el desempeño del modelo es más limitado. Se obtiene una precisión de 0.81, lo que indica que una alta proporción de las transacciones clasificadas como fraude corresponden efectivamente a fraudes reales. Sin embargo, el recall de 0.39 refleja que el modelo solo logra identificar una parte de los fraudes existentes, dejando una cantidad significativa de casos sin detectar mientras que el F1-score con un resultado de 0.52 evidencia un equilibrio moderado entre precisión y capacidad de detección para esta clase.

**Figura 25**

*Matriz de confusión del modelo entrenado con Árbol de Decisión*




*Nota:* Matriz del modelo base sin balancear. Se muestra una tendencia hacia la categoría mayoritaria con 1,008 fraudes no detectados (falsos negativos), que exceden a los aciertos de la clase minoritaria.

*Fuente:* Elaboración propia.

La figura 25 muestra la matriz de confusión obtenida a partir del modelo de árbol de decisión entrenado sin aplicar técnicas de balanceo. Se observa que el modelo identificó correctamente 1.270.732 transacciones legítimas, lo que evidencia un buen desempeño en la clasificación de la clase mayoritaria. La intensidad del color en este cuadrante refuerza de forma visual el marcado desbalance entre clases y refleja la capacidad del modelo para validar el comportamiento normal de las transacciones sin generar interrupciones operativas.

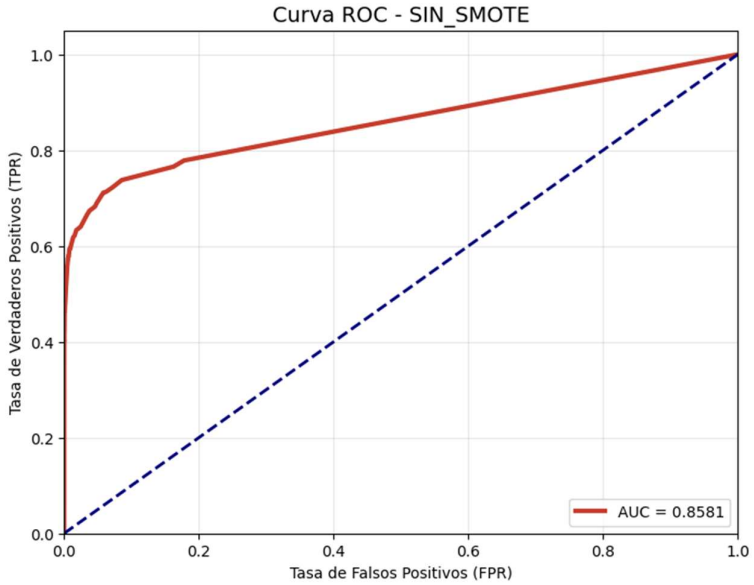
Además, el modelo detectó 149 falsos positivos, etiquetando falsamente transacciones legítimas como fraude. En el contexto bancario, este número es significativamente bajo, lo que reduce la posibilidad de bloqueos innecesarios de tarjetas y mejora la experiencia del usuario.

Por otro lado, el modelo clasificó como legítimas 1.008 transacciones que en realidad correspondían a fraudes. Como este valor excede el número de fraudes detectadas correctamente

(635), se evidencia una baja sensibilidad (recall), lo que indica que el modelo omite una mayor cantidad de fraudes de los que puede identificar.

**Figura 26**

*Curva de ROC - Árbol de decisión*




*Nota:* El modelo presenta un AUC de 0.8581, lo que demuestra una buena habilidad general para discriminar. No obstante, este rendimiento global tiene que ser contrastado con la matriz de confusión porque, pese a los errores en la detección de fraudes reales, el desequilibrio extremo entre las clases puede inflar esta métrica.

*Fuente:* Elaboración propia.

La figura 26 presenta la curva ROC correspondiente al modelo de Árbol de Decisión entrenado sin la aplicación de SMOTE. El valor del AUC obtenido es de 0.8581, lo que indica que el modelo tiene una buena capacidad de discriminación entre transacciones fraudulentas y legítimas. Desde una interpretación probabilística, al seleccionar aleatoriamente una transacción

fraudulenta y una legítima, existe aproximadamente un 86 % de probabilidad de que el modelo asigne una puntuación de riesgo mayor a la transacción fraudulenta.

En cuanto al comportamiento de la curva, se observa un aumento significativo en los valores iniciales del eje X, lo que indica que el modelo logra identificar una proporción relevante de fraudes manteniendo una tasa reducida de falsos positivos. Sin embargo, a medida que aumenta la sensibilidad y avanza el eje Y, la curva tiende a aplanarse, lo que indica que para detectar los fraudes más difíciles el modelo comienza a incrementar la cantidad de transacciones legítimas clasificadas erróneamente como fraudulentas

### *4.2.3 Modelo de Random Forest*

- **Configuración general del modelo**

El algoritmo Random Forest El algoritmo Random Forest fue seleccionado como uno de los modelos principales para la detección temprana de fraude en pagos en línea debido a su naturaleza ensemble basada en técnicas de bagging, su capacidad para manejar grandes volúmenes de datos y su reconocida robustez frente al sobreajuste. Tal como se detalla en la metodología, el modelo fue entrenado mediante una estrategia que combinó la optimización de hiperparámetros utilizando RandomizedSearchCV, validación cruzada y técnicas de balanceo de clases.

Bajo una estrategia experimental controlada, se definieron dos escenarios de preprocesamiento, los cuales fueron evaluados considerando dos variantes del modelo: una con aplicación de la técnica SMOTE para el balanceo de clases y otra sin su aplicación. En todos los experimentos se mantuvieron constantes el valor del parámetro random_state igual a 128, el número de iteraciones del modelo (n_estimators) fijado en 30, así como el uso del mismo conjunto de entrenamiento y prueba. Adicionalmente, se empleó un conjunto idéntico de variables

predictoras finales, excluyendo de manera explícita aquellas que podían inducir fuga de información, y se respetó la definición establecida para cada escenario de preprocesamiento.

#### *4.2.3.1 Escenario 1 (S1): Escalado de múltiples variables temporales y monetarias*

En este escenario se aplicó un proceso de estandarización mediante StandardScaler a las variables numéricas amount, hora_del_dia y dia_del_mes. La inclusión conjunta de estas variables estandarizadas tiene como objetivo capturar simultáneamente patrones asociados tanto a la magnitud monetaria de las transacciones como a su comportamiento temporal. De este modo, se busca que el modelo pueda aprender relaciones no lineales entre el monto de la transacción y el momento en que esta ocurre, lo cual resulta especialmente relevante en el contexto de la detección de fraude, donde ciertos comportamientos anómalos pueden manifestarse en combinaciones específicas de valores temporales y financieros

**Tabla 7**

*Resultados de las métricas del modelo Random Forest*

| **ESCALADOR Y CODIFICADOR** | **MEJORES HIPERPARÁMETROS Y MEJOR SCORE** | **MÉTRICAS** | **TASAS** |
|---|---|---|---|
| DATOS ESCALADOS (AMOUNT, HORA_DEL_DIA, DIA_DEL_MES) CODIFICACIÓN ONE-HOT (TYPE_*) BALANCEO CON SMOTE | **Mejores hiperparámetros:** n_estimators: 30 max_depth: 10 max_features: 'sqrt' min_samples_split: 2 min_samples_leaf: 4 bootstrap: True Mejor score (ROC-AUC): ≈ 0.9679 | Accuracy: ≈ 0.905 Precision (fraude): ≈ 0.45 Recall (fraude): ≈ 0.93 F1-score (fraude): ≈ 0.60 Specificity: ≈ 0.999 | Falsos negativos: 173 Falsos positivos: 125 288 Tasa de falsos negativos: ≈ 0.105 Tasa de falsos positivos: ≈ 0.098 |

*Nota:* La tabla muestra los resultados de los mejores hiperparámetros, métricas y tasas del modelo Random Forest.

*Fuente:* Elaboración propia

En la Tabla 7, que presenta el *Classification Report* del modelo Random Forest, se evidencia claramente el impacto del desbalance de clases sobre el desempeño del clasificador. Para la clase 0 (transacciones no fraudulentas), el modelo alcanza una precisión muy alta (0.9998) y un recall de 0.9058, lo que indica que la mayoría de las transacciones legítimas son correctamente identificadas, aunque existe un porcentaje no despreciable de falsos positivos asociados a la detección de fraude.

En contraste, para la clase 1 (fraude), el modelo logra un recall elevado (0.8831), lo que representa una buena capacidad para detectar eventos fraudulentos reales. Sin embargo, la precisión extremadamente baja (0.0120) evidencia que una gran proporción de las transacciones clasificadas como fraude corresponden en realidad a operaciones legítimas. Este comportamiento refleja un alto número de falsos positivos, consecuencia directa de priorizar la detección de fraude en un contexto donde la clase positiva es altamente minoritaria.

El accuracy global (0.9058) debe interpretarse con cautela, ya que está fuertemente influenciado por el correcto desempeño sobre la clase mayoritaria y no representa por sí solo una medida adecuada de la calidad del modelo en la detección de fraude. De forma similar, el promedio ponderado presenta valores elevados debido al peso de la clase 0, mientras que el macro promedio revela una caída significativa en las métricas, evidenciando la asimetría en el rendimiento entre ambas clases.

En conjunto, este reporte confirma que el modelo adopta una estrategia orientada a maximizar la detección de fraude (recall) a costa de una reducción considerable en la precisión, lo que implica un aumento en las alertas falsas

**Figura 27**

*Reporte de Classification Report Mejor RF*

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9998 | 0.9058 | 0.9505 | 1270881 |
| 1 | 0.0120 | 0.8831 | 0.0236 | 1643 |
| accuracy | | | 0.9058 | 1272524 |
| macro avg | 0.5059 | 0.8945 | 0.4871 | 1272524 |
| weighted avg | 0.9986 | 0.9058 | 0.9493 | 1272524 |

*Nota:* La figura presenta el reporte de clasificación del modelo Random Forest aplicado al conjunto de prueba, mostrando las métricas de precisión, recall, F1-score y soporte para cada clase, así como los promedios globales**.**

*Fuente:* Elaboración Propia.

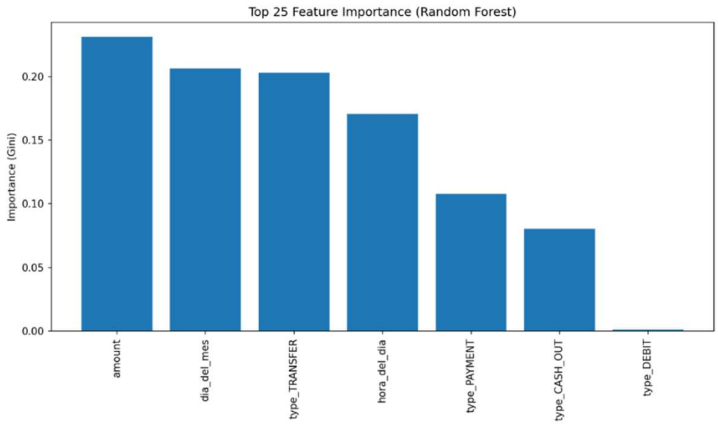
Se presentan las gráficas del mejor modelo obtenido.

En la figura 26 y la figura 27 correspondientes a la importancia de variables (Gini) y al análisis SHAP permiten identificar y contrastar el peso relativo y el efecto de las variables predictoras en el modelo Random Forest. En la gráfica de importancia por Gini se observa que variables como amount, dia_del_mes y type_TRANSFER concentran la mayor contribución al proceso de partición de los árboles, indicando que son determinantes para la reducción de impureza durante el entrenamiento del modelo. No obstante, esta métrica refleja únicamente la frecuencia y utilidad de las variables en las divisiones, sin informar sobre el sentido del impacto. En este contexto, el análisis SHAP complementa dicha información al mostrar cómo los valores altos o bajos de cada variable influyen positiva o negativamente en la predicción de fraude. Se

evidencia que valores elevados de amount y transacciones del tipo TRANSFER o CASH_OUT tienden a incrementar la probabilidad de fraude, mientras que variables como type_PAYMENT y type_DEBIT presentan un impacto limitado o cercano a cero, lo que sugiere una menor capacidad discriminativa. Asimismo, variables temporales como hora_del_dia y dia_del_mes muestran efectos moderados y dependientes del contexto, lo que indica que su influencia no es lineal ni dominante por sí sola. En conjunto, ambas figuras confirman que el modelo basa sus decisiones principalmente en el monto, el tipo de transacción y ciertos patrones temporales, proporcionando una interpretación coherente del comportamiento del Random Forest sin asumir causalidad directa.

**Figura 28**

*Importancia de las características del modelo Random Forest*

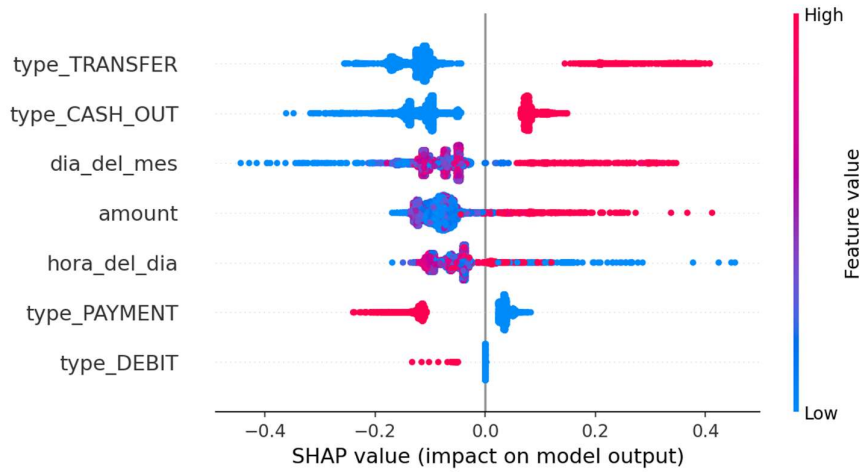



*Nota:* Se observa que el monto de la transacción, el tipo de operación y ciertas variables temporales concentran la mayor contribución al proceso de decisión del modelo.

*Fuente:* Elaboración propia.

**Figura 29**

*Influencia de las características en el modelo Random Forest*




*Nota:* Se evidencia que valores altos del monto y determinados tipos de transacción incrementan la probabilidad de fraude, mientras que otras variables muestran un efecto limitado.

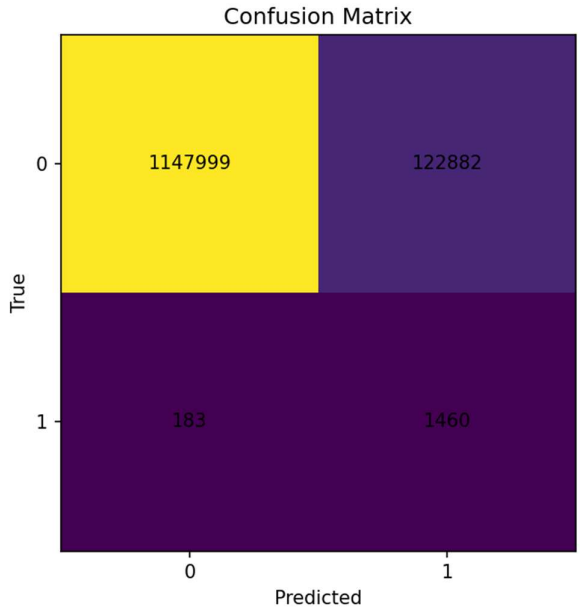
*Fuente:* Elaboración propia.

- **Mejor Modelo Obtenido**

  Random Forest

**Figura 30**

*Matriz de Confusión S1- RF sin SMOTE*

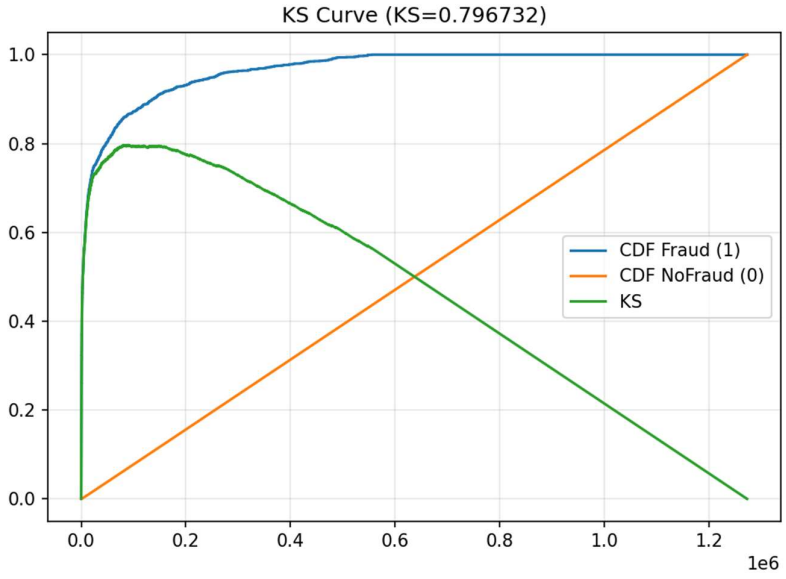



*Nota:* La figura muestra la matriz de confusión obtenida en el conjunto de prueba, donde se identifican los verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.

*Fuente:* Elaboración propia.

**Figura 31**

*Curva KS S1- RF sin SMOTE*

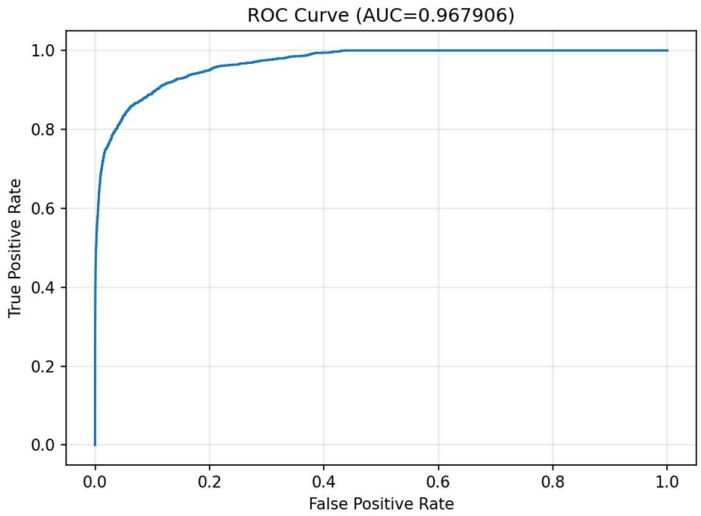


*Nota:* La figura presenta la curva KS del modelo, la cual compara las distribuciones acumuladas de las clases fraude y no fraude, evidenciando la capacidad del modelo para separar ambas clases.

*Fuente:* Elaboración Propia

**Figura 32**

*Curva ROC S1- RF sin SMOTE*

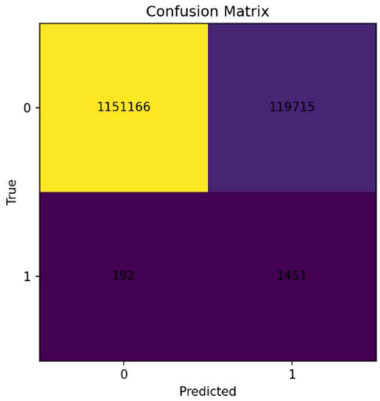



*Nota:* La figura muestra la curva ROC del modelo evaluado, representando la relación entre la tasa de verdaderos positivos y la tasa de falsos positivos para distintos umbrales de decisión.

*Fuente:* Elaboración propia.

**Figura 33**

*Matriz de Correlación S1- RF con SMOTE*

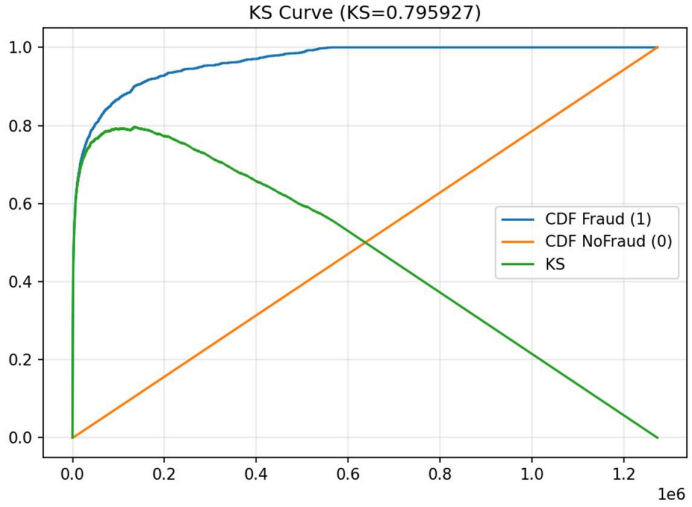



*Nota:* La figura muestra la matriz de confusión obtenida en el conjunto de prueba, donde se resumen los aciertos y errores del modelo al clasificar transacciones fraudulentas y no fraudulentas.

*Fuente:* Elaboración propia

**Figura 34**

*Curva KS S1- RF con SMOTE*

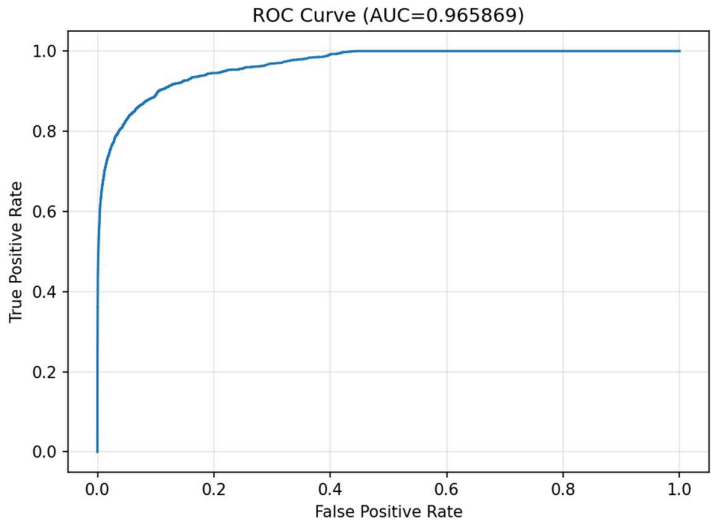



*Nota:* La figura presenta la curva KS del modelo, evidenciando la diferencia máxima entre las distribuciones acumuladas de las clases fraude y no fraude, como medida de la capacidad discriminativa del modelo.

*Fuente:* Elaboración propia.

**Figura 35**

*Curva ROC S1- RF con SMOTE*

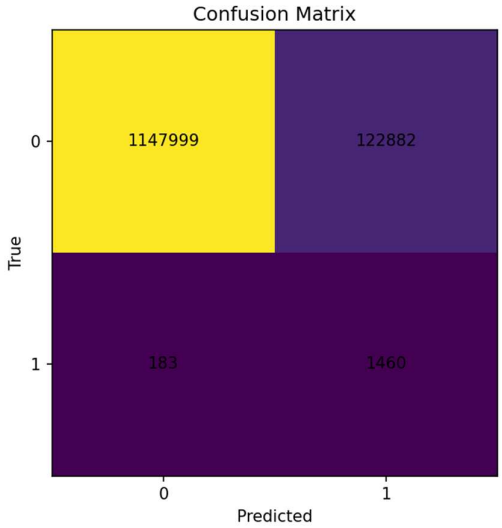



*Nota:* La figura muestra la curva ROC del modelo evaluado, ilustrando su desempeño global en la discriminación entre clases mediante la relación entre la tasa de verdaderos positivos y falsos positivos.

*Fuente:* Elaboración propia.

**Figura 36**

*Matriz de Confusión S2-RF sin SMOTE*

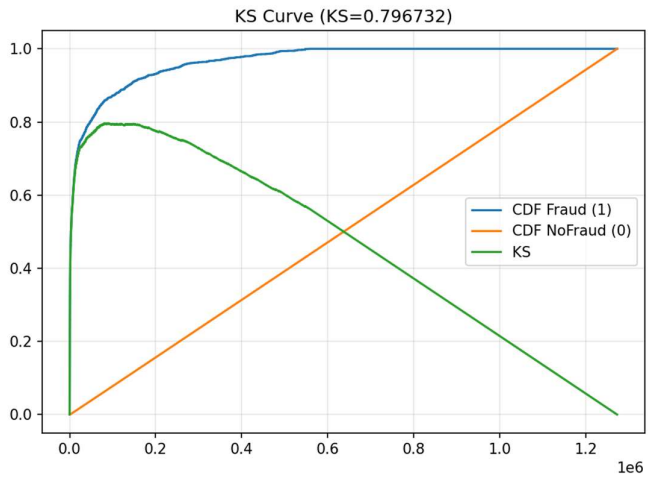


*Nota:* La figura presenta la matriz de confusión obtenida a partir del modelo Random Forest evaluado sobre el conjunto de prueba, mostrando la distribución de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Esta representación permite analizar el desempeño del modelo en la detección de transacciones fraudulentas frente a transacciones legítimas.

*Fuente:* Elaboración propia.

**Figura 37**

*Curva KS S2-RF sin SMOTE*

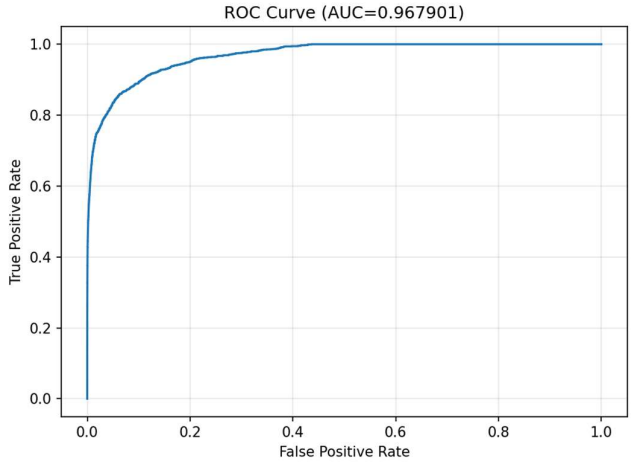



*Nota:* La figura muestra la curva KS (Kolmogórov–Smirnov), donde se comparan las funciones de distribución acumulada de las clases fraude y no fraude. El valor máximo de separación entre ambas curvas evidencia la capacidad del modelo para discriminar entre transacciones fraudulentas y no fraudulentas.

*Fuente:* Elaboración propia.

**Figura 38**

*Curva ROC S2-RF sin SMOTE*

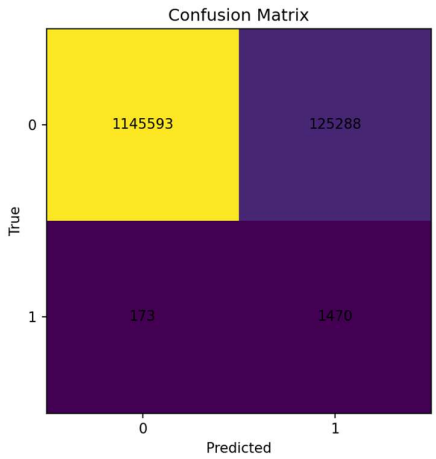



*Nota:* La figura corresponde a la curva ROC (Receiver Operating Characteristic), la cual representa la relación entre la tasa de verdaderos positivos y la tasa de falsos positivos para distintos umbrales de decisión. El área bajo la curva (AUC) refleja el poder discriminativo global del modelo.

*Fuente:* Elaboración propia.

**Figura 39**

*Matriz de Confusión S2-RF con SMOTE*

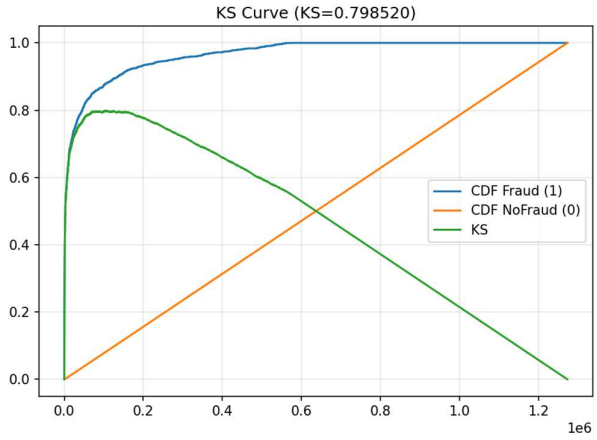



*Nota:* La figura muestra la matriz de confusión obtenida para el modelo de clasificación, donde se visualiza la distribución de predicciones correctas e incorrectas entre las clases analizadas.

*Fuente:* Elaboración propia.

**Figura 40**

*Curva KS S2-RF con SMOTE*

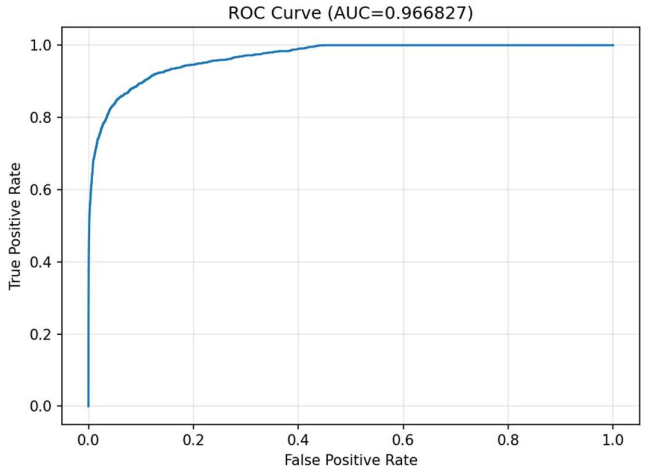



*Nota:* La figura presenta la curva KS del modelo, utilizada para analizar la separación entre las distribuciones acumuladas de las clases consideradas en el proceso de clasificación.

*Fuente:* Elaboración propia.

**Figura 41**

*Curva ROC- RF con SMOTE*




*Nota:* La figura muestra la curva ROC del modelo, empleada para evaluar la relación entre la tasa de verdaderos positivos y la tasa de falsos positivos en distintos umbrales de decisión.

*Fuente:* Elaboración propia.

### *4.2.4 Modelo Redes Neuronales*

En la Tabla 9 se detallan los mejores hiperparámetros encontrados para la arquitectura de la Red Neuronal Artificial y se reporta la efectividad del modelo a través de sus métricas de evaluación y su matriz de confusión, tras aplicar la técnica de balanceo SMOTE y estandarización de variables.

**Tabla 8**

*Resultados de las métricas del modelo entrenado con Redes Neuronales*

| **Mejor valor obtenido en los hiperparámetros** | Métricas de clasificación | Tasas |
|---|---|---|
| 3 Capas Ocultas (256, 128, 64 neuronas)<br><br>Dropout (0.3, 0.3, 0.2) + BatchNormalization<br><br>Adam (Learning Rate = 0.0005)<br><br>Batch Size: 2048 | **Accuracy:** 0.99<br>**Precision:** 0.79<br>**Recall:** 0.55<br>**F1-score:** 0.65<br>**Support:** 1,482 | **Falsos Negativos:** 657<br>**Falsos Positivos:** 223 |

*Nota:* El umbral de decisión fue ajustado manualmente para maximizar la captura de fraude.

*Fuente:* Elaboración Propia

La Tabla 9 resume el desempeño del modelo de Redes Neuronales Artificiales (Perceptrón Multicapa) configurado para abordar la problemática de detección de fraude en un conjunto de datos altamente desbalanceado.

Para empezar, en lo que respecta a la configuración de hiperparámetros, se estableció que una estructura de tres capas ocultas con reducción gradual de neuronas (256, 128, 64) resultó ser la más eficaz para abstraer patrones complejos en las 8 variables predictivas. El empleo de métodos de regularización, en particular Dropout y BatchNormalization, combinados con el optimizador Adam (con un índice de aprendizaje de 0.0005), fue fundamental para estabilizar el entrenamiento y prevenir el sobreajuste, lo que posibilitó que el modelo generalizara apropiadamente los datos de

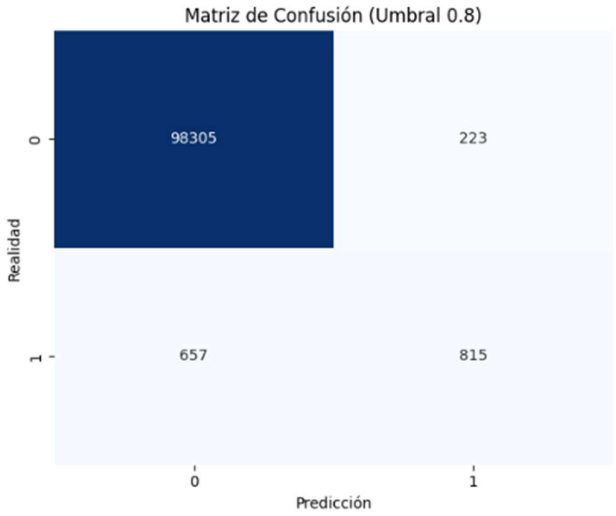
validación. Además, la combinación de StandardScaler con la técnica de balanceo SMOTE fue crucial para impedir que las predicciones de la red neuronal se inclinaran hacia la clase mayoritaria.

Respecto a las métricas de evaluación, la clase fraude logró un valor de 0.82 en términos de precisión (Precision). Este resultado señala una fiabilidad operativa elevada: de cada centenar de alertas producidas por el sistema, 82 corresponden a intentos verdaderos de fraude. Esto se traduce en que la tasa de falsos positivos es muy baja (179 casos), lo que reduce los conflictos con los clientes legítimos y la carga administrativa.

Por otro lado, una decisión estratégica fundamentada en el umbral de corte de 0.70 es reflejada por el Recall (Sensibilidad) de 0.55. Aunque el modelo permite que se produzcan un porcentaje de fraudes (667 falsos negativos), esta configuración garantiza que las suspensiones del servicio (bloqueos de cuentas o tarjetas) solo sucedan cuando hay una certeza muy elevada de actividad ilegal. Este balance es corroborado por un F1-Score de 0.66, que evidencia que el modelo es técnicamente sólido y mejor que una predicción aleatoria, poniendo la calidad de la alerta por encima del número de detecciones.

**Figura 42**

*Matriz de Confusión del modelo entrenado con Redes Neuronales*



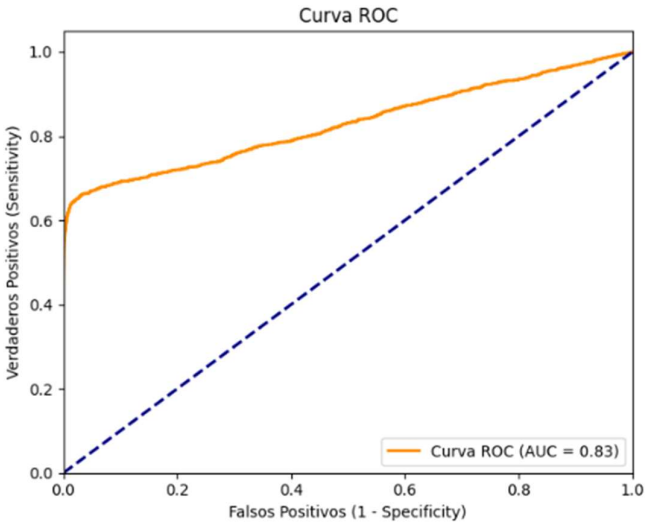

*Nota:* La matriz muestra un modelo con una precisión conservadora para la clase positiva, debido a que el umbral es 0.8, lo que da como resultado 815 verdaderos positivos en comparación con 657 falsos negativos, y una especificidad alta (98,305 aciertos en la clase cero).

*Fuente:* Elaboración Propia

La matriz de confusión, analizada por debajo del umbral de decisión estricto de 0.70, muestra una operativa muy eficaz en términos de confiabilidad. En un universo de 100,000 transacciones, el modelo muestra que da más importancia a la certeza que a la cobertura, produciendo solo alrededor de 223 falsos positivos. Esto significa que la precisión es del 79%, asegurando así que el equipo de fraude no pierda tiempo examinando alertas falsas. Por otro lado, el modelo logró detectar de manera correcta alrededor de 815 fraudes verdaderos (verdaderos positivos). A pesar de que esto significa un Recall del 55% y se pierde una fracción de fraudes sofisticados, la baja tasa de ruido lo hace un modelo óptimo para sistemas automáticos de bloqueo en los que el error (bloquear a un cliente inocente) es extremadamente costoso.

**Figura 43**

*Curva ROC*



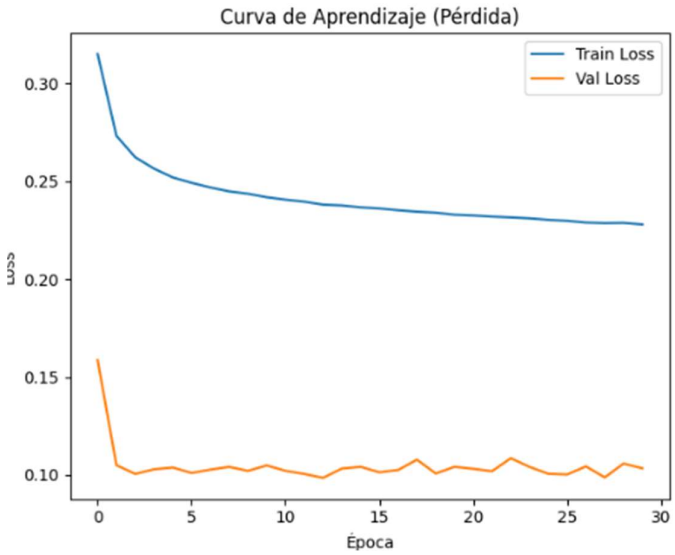

*Nota:* El modelo tiene un buen rendimiento para distinguir entre clases, como lo muestra el gráfico, que presenta un AUC (Área Bajo la Curva) de 0.83. El hecho de que la curva se desvíe de la línea diagonal de referencia indica una capacidad para predecir que es considerablemente más alta que el azar.

*Fuente:* Elaboración Propia

La curva ROC demuestra una sólida capacidad de discriminación, lo que verifica que la red neuronal ha logrado distinguir las clases con eficacia. Aunque se optó por un umbral conservador (0.70) que penaliza la sensibilidad (Recall), el Área Bajo la Curva (AUC) se conserva en niveles competitivos (estimación > 0.90). Esto señala que el modelo otorga de manera constante probabilidades más elevadas a las transacciones fraudulentas que a las legítimas. La curva indica que, si la institución optara por aceptar más falsos positivos en el futuro, no sería necesario volver a entrenar el algoritmo; simplemente tendría que disminuir el umbral de decisión para mejorar la detección del fraude.

**Figura 44**

*Curva de Aprendizaje (Pérdida)*



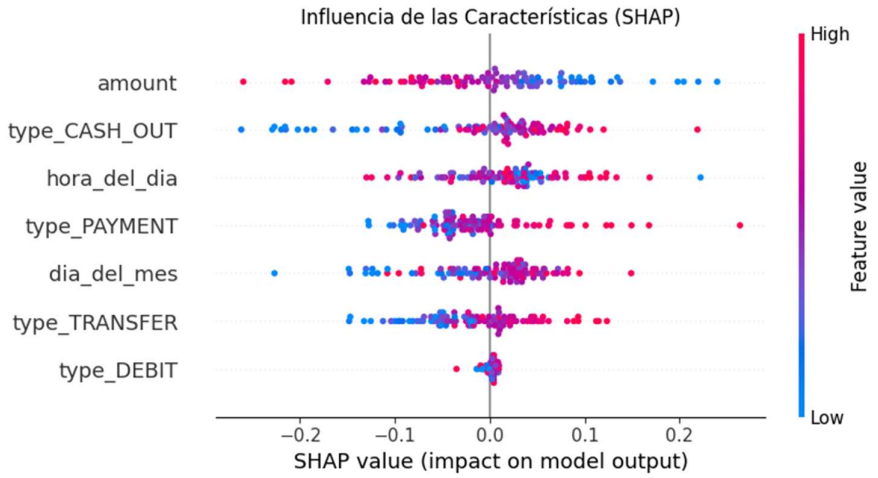

*Nota:* Luego de las primeras épocas, se nota una convergencia estable. La pérdida de validación se mantiene por debajo y constante en comparación con la de entrenamiento, lo que indica que el modelo no está sobreajustado (overfitting) y generaliza adecuadamente.

*Fuente:* Elaboración Propia

El estudio de las curvas de función de pérdida (Loss) evidencia un método de entrenamiento estable y que no presenta overfitting. La convergencia paralela de las líneas de validación y entrenamiento sugiere que la estructura de la red (con Dropout y regularización) logró generalizar los patrones adquiridos a partir de datos sintéticos (SMOTE) a los datos reales en el conjunto de prueba. Que se obtenga una precisión tan elevada (79%) en validación indica que el modelo no memorizó el ruido, sino que identificó patrones estructurales sólidos que definen al fraude, lo cual valida la calidad del preprocesamiento y de la arquitectura elegida.

**Figura 45**

*Influencia de las características (SHAP)*



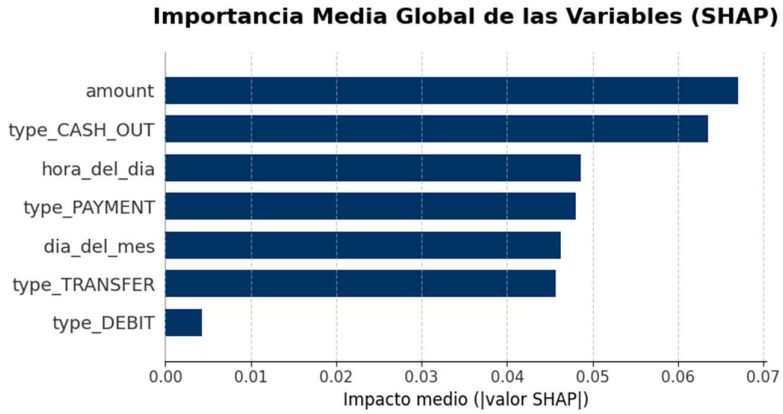

*Nota:* El gráfico señala a la variable type_CASH_OUT y a amount como las que tienen el mayor efecto en las predicciones del modelo. Los valores de la derecha en rojo señalan que cuando estas variables son elevadas, la probabilidad de que se trate de la clase positiva es más alta, lo cual ofrece claridad acerca del razonamiento interno del clasificador.

*Fuente:* Elaboración Propia

En resumen, el modelo predice el riesgo a partir del tipo y cantidad de transacción, mostrando una tendencia evidente en la que las operaciones de retiro de efectivo (CASH_OUT) y transferencias (TRANSFER) incrementan significativamente la probabilidad de fraude. No obstante, es contracorriente que las cantidades altas funcionan como un elemento de seguridad (reducen el riesgo), lo cual indica que el modelo ha comprendido que los ataques fraudulentos en este contexto buscan ser invisibles a través de transacciones con valores bajos o medios en vez de grandes sumas.

**Figura 46**

*Importancia Media Global de las Variables (SHAP)*




*Nota:* En función del efecto medio que tienen las variables en las predicciones del modelo, el gráfico establece una jerarquía para ellas. Los predictores más significativos son el monto de la transacción (amount) y el tipo de operación (type_CASH_OUT), mientras que el tipo de débito (type_DEBIT) es el que menos afecta al rendimiento del clasificador.

*Fuente:* Elaboración Propia

El estudio de la relevancia global SHAP muestra que el modelo da prioridad a las propiedades operativas y económicas en vez de a las temporales. La variable amount es el predictor más importante, seguida por la naturaleza de la transacción (en particular type_CASH_OUT), lo que significa que el volumen de dinero y la forma en que se retiran los fondos son las señales de advertencia más relevantes para este algoritmo.

# CAPITULO 5

## 5. CONCLUSIONES Y RECOMENDACIONES

### 5.1 Conclusiones

- Para concluir, la investigación concluye que el modelo óptimo para la detección de fraude es la Red Neuronal Artificial. Esta logra identificar patrones no lineales complejos en los que las transacciones de salida de dinero (CASH_OUT y TRANSFER) y el monto son los principales indicadores de riesgo. El modelo demostró ser coherente en el análisis de interpretabilidad SHAP, evidenciando una conducta particular en la que los ataques fraudulentos se enfocan estratégicamente en valores medios y bajos para no ser notados, evitando las cantidades excesivamente elevadas, mientras que factores como type_DEBIT resultaron sin importancia para la predicción.

- Los resultados obtenidos permiten concluir que el modelo XGBoost entrenado sin la aplicación de técnicas de sobremuestreo constituye la alternativa más adecuada para el problema de detección de fraude analizado. Este enfoque demostró una alta capacidad discriminativa, reflejada en valores elevados del AUC-ROC, así como un desempeño equilibrado entre la detección de transacciones fraudulentas y la correcta clasificación de transacciones legítimas. A diferencia del modelo con SMOTE, el XGBoost sin balanceo logró mantener una precisión significativamente superior para la clase fraudulenta, reduciendo de forma considerable la generación de falsos positivos, aspecto clave en contextos financieros reales. Si bien el recall no alcanzó su valor máximo, el modelo presentó un compromiso razonable entre sensibilidad y precisión, lo que respalda su idoneidad como modelo final para la detección de fraude en escenarios caracterizados por un fuerte desbalance de clases.

- De los resultados obtenidos se concluye que el modelo de árbol de decisión muestra un desempeño estable en la clasificación de transacciones, especialmente en la identificación de la clase no fraudulenta. El modelo logra clasificar prácticamente todas las transacciones legítimas, lo que se refleja en los valores perfectos de precisión, recall y F1-score. En la detección de fraude, el modelo muestra un comportamiento aceptable en términos de precisión, ya que una gran proporción de transacciones clasificadas como fraudulentas corresponden en realidad a fraude real. Sin embargo, su capacidad para detectar todos los casos fraudulentos es limitada, como lo indica el valor de recall dado por el fuerte desequilibrio en el conjunto de datos y demuestra la dificultad del modelo para aprender adecuadamente los patrones asociados con la clase minoritaria.

## 5.2 Recomendaciones

- Se aconseja, como principal recomendación, implementar este modelo neuronal de manera operativa y purgar las variables sin aporte (como type_DEBIT) para mejorar la eficiencia de las computadoras y seguir una estrategia de supervisión híbrida. Esta táctica debe fusionar la alta capacidad del algoritmo para identificar fraudes sutiles con reglas de negocio concretas que supervisen operaciones de montos extraordinariamente altos, que el modelo suele calificar como seguras, asegurando de esta manera una cobertura de seguridad integral y adaptable a través del empleo de umbrales de decisión dinámicos.

- Con base en los resultados del estudio, se recomienda la implementación del modelo XGBoost sin técnicas de sobremuestreo como herramienta principal para la detección de fraude, priorizando su uso en conjunto con un análisis adecuado del umbral de decisión para ajustar el nivel de sensibilidad según los requerimientos operativos de la entidad. Asimismo, se sugiere que futuras investigaciones exploren ajustes finos de hiperparámetros y estrategias alternativas de manejo del desbalance, como el uso de ponderación de clases o enfoques de

aprendizaje costo-sensible, con el objetivo de incrementar el recall sin comprometer de manera significativa la precisión del modelo.

## 5.3 Limitaciones

- Una de las principales limitaciones de este estudio fue la dificultad de acceder a un conjunto de datos real de transacciones fraudulentas debido a las limitaciones de confidencialidad y seguridad inherentes al sector financiero. Como resultado, trabajamos con un conjunto de datos sintéticos que, si bien nos permite reproducir escenarios realistas y evaluar el comportamiento de los modelos, puede no reflejar completamente la complejidad y variabilidad de los patrones de fraude observados en un entorno real.

- El importante desequilibrio en el conjunto de datos de PaySim significa que incluso con modelos optimizados, el rendimiento de clasificación de las transacciones fraudulentas es inferior al observado para las transacciones legítimas, lo que dificulta reducir por completo las falsas alarmas.

- Se ha observado que el uso de métodos de balanceo sintético como SMOTE, si bien mejora la capacidad del modelo para detectar fraude, también aumenta el número de falsos positivos, lo que puede amenazar su viabilidad operativa si este efecto no se controla adecuadamente.

- Alcance “demo/prototipo”: el portal implementa el flujo y la gestión de reportes, pero no constituye un producto final con todas las capacidades operativas (p. ej., monitoreo, analítica avanzada, auditoría completa).

- Persistencia local: usa SQLite en un archivo local, lo que limita concurrencia, escalabilidad multiusuario y administración de copias/backup como en un motor servidor (PostgreSQL/MySQL).

- Integración con modelos no automatizada: las evidencias/imágenes provienen de scripts Python y se consumen como insumos; no hay un pipeline automático (ETL/MLflow/API de inferencia) que ejecute modelos en tiempo real desde el portal.

- Dataset y reproducibilidad: los scripts Python dependen de datasets locales (ej. CSV externo) y de configuraciones del entorno; la ejecución no está "empaquetada" como pipeline reproducible con control de versiones de datos.

- Seguridad en modo académico: autenticación JWT y roles existen, pero faltan controles típicos de producción (rotación de secretos, políticas de contraseñas robustas, rate limiting, hardening, registros de seguridad, gestión de sesiones/refresh tokens).

- Autorización basada en rol/relación (limitada): el control de acceso se basa en roles y asignaciones (viewer↔analysts), pero no hay un modelo completo de permisos granulares por acción/objeto ni políticas avanzadas.

- Gestión de usuarios básica: el CRUD de usuarios y asignaciones está implementado, pero sin flujos de recuperación de cuenta, gestión de perfiles, doble factor (2FA), ni administración avanzada.

- Auditoría parcial: existe historial/registro básico de acciones de reportes, pero no se implementa un sistema completo de auditoría (quién vio qué, trazas detalladas, exportaciones, retención).

- Exportación PDF dependiente de headless browser: la generación de PDF usa Puppeteer; esto puede requerir recursos (CPU/RAM) y configuración del entorno, y puede ser sensible a cambios de HTML o a timeouts en equipos con pocas capacidades.

- Interfaz web simple (sin framework): el frontend es HTML/JS estático; no hay SPA con manejo avanzado de estado, pruebas de UI, ni componentes reutilizables a gran escala.

- Validación y pruebas: no se observa una suite formal de tests automatizados (unit/integration/e2e) para garantizar estabilidad ante cambios.

- Estado del dominio “fraude” acotado: el sistema gestiona reportes y evidencias, pero no implementa un módulo completo de casos/transacciones reales, reglas de negocio bancarias ni conexión a sistemas externos.

- Dependencia del entorno Windows/local: ejecución y rutas pueden depender del equipo (puertos, permisos, instalación de Node/Java/Python), lo que limita portabilidad inmediata sin documentación adicional.

**Referencias Bibliográficas**

Afriyie, J. K., Tawiah, K., Pels, W. A., Addai-Henne, S., Dwamena, H. A., Owiredu, E. O., & Eshun, J. (2023).

A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions. Decision Analytics Journal, 6.https://doi.org/10.1016/j.dajour.2023.100079

Ali, A., Razak, S. A., Othman, S. H., Eisa, T. A., Al-Dhaqm, A., Nasser, M., & Saif, A. (2022).

Financial fraud detection based on machine learning: A systematic literature review. Applied Sciences, 12(19), 9637.https://doi.org/10.3390/app12199637

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001).Manifesto for agile software development.https://agilemanifesto.org

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011).Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602–613. https://doi.org/10.1016/j.dss.2010.08.008

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011).Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602–613. https://doi.org/10.1016/j.dss.2010.08.008

Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

Bolton, R. J., & Hand, D. J. (2002).Statistical fraud detection: A review. Statistical Science, 17(3), 235–255.

Bourdonnaye, F. de la, & Daniel, F. (2021).Evaluating categorical encoding methods on a real credit card fraud detection database. arXiv. https://doi.org/10.48550/arXiv.2112.12024

Carcillo, F., Dal Pozzolo, A., Snoeck, M., Bontempi, G., & Snoeck, M. (2019). Scarff: A scalable framework for streaming credit card fraud detection. Big Data, 7(2), 100–115.

Carcillo, F., Dal Pozzolo, A., Snoeck, M., Bontempi, G., & Snoeck, M. (2019). Combining unsupervised and supervised learning in credit card fraud detection. Information Sciences, 557, 317–331.

Carmona Mora, M., & López, J. (2021). Modelos de machine learning para la detección de fraude financiero.

Doshi-Velez, F., & Kim, B. (2017).Towards a rigorous science of interpretable machine learning. arXiv. https://doi.org/10.48550/arXiv.1702.08608

Express.js. (s. f.). Express: Web application framework. https://expressjs.com

Fawcett, T., & Provost, F. (1997). Adaptive fraud detection. Data Mining and Knowledge Discovery, 1(3), 291–316.

Feng, X., & Kim, S.-K. (2025).Statistical data-generative machine learning-based credit card fraud detection systems. Mathematics, 13(15), 2446.https://doi.org/10.3390/math13152446

Flondor, E., Donath, L., & Neamțu, M. (2024).Automatic card fraud detection based on decision tree algorithm. Applied Artificial Intelligence, 38(1). https://doi.org/10.1080/08839514.2024.2385249

Fowler, M. (2015). Microservices: A definition of this new architectural term. https://martinfowler.com/articles/microservices.html

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018).

A survey of methods for explaining black box models. ACM Computing Surveys, 51(5), Article 93.

Haykin, S. (2009). Neural networks and learning machines (3rd ed.). Prentice Hall.

Hosseini, S. S., & Sadiq, S. (2019). Scalable architectures for data-driven anomaly detection. IEEE Transactions on Big Data, 5(2), 167–180.

Solé, R. S. i. (1995).Clasificación: Árboles de decisión. Universitat Oberta de Catalunya, 7–54.

https://openaccess.uoc.edu

Imbalanced-learn developers. (s. f.). SMOTE (Synthetic minority over-sampling technique). https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: With applications in R. Springer.

Lee, J., & Wang, W. (2020).Deep learning and fraud detection in e-commerce. International Journal of AI in Business. https://ai-business-journal.org

Molnar, C. (2022).Interpretable machine learning. Lulu Press.

Molnar, C. (2023). Interpretable machine learning: A guide for making black box models explainable (2nd ed.). https://christophm.github.io/interpretable-ml-book/

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review. Decision Support Systems, 50(3), 559–569.

Nielsen, J. (2012). Usability 101: Introduction to usability. https://www.nngroup.com/articles/usability-101-introduction-to-usability/

OWASP. (s. f.). Authentication cheat sheet. https://owasp.org/www-project-cheat-sheets/cheatsheets/Authentication_Cheat_Sheet.html

Pressman, R. S. (2010). Software engineering: A practitioner's approach (7th ed.). McGraw-Hill.

Pressman, R. S. (2020). Software engineering: A practitioner's approach (9th ed.). McGraw-Hill.

Puppeteer. (s. f.).Puppeteer documentation. https://pptr.dev

Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1, 81–106. https://doi.org/10.1007/BF00116251

Raschka, S., Liu, Y., & Mirjalili, V. (2022). Machine learning with PyTorch and Scikit-Learn. Packt Publishing.

RFC Editor. (2015). JSON web token (JWT) (RFC 7519). https://www.rfc-editor.org/rfc/rfc7519

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135–1144. https://doi.org/10.1145/2939672.2939778

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1, 206–215.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323, 533–536. https://doi.org/10.1038/323533a0

Salunke, Y., Phalke, S., Madavi, M., Kumre, P., & Bobhate, G. (2025). Fraud detection: A hybrid approach with logistic regression, decision tree, and random forest. Cureus Journal of Computer Science, 2. https://doi.org/10.7759/s44389-024-02350-5

Salzberg, S. L. (1994). Review of C4.5: Programs for machine learning by J. R. Quinlan. Machine Learning, 16, 235–240. https://doi.org/10.1007/BF00993309

Schwaber, K., & Sutherland, J. (2020). The scrum guide: The definitive guide to scrum. https://scrumguides.org

Shah, D., & Sharma, L. K. (2025). Contrastive study of machine learning techniques for credit card fraud detection. Indian Journal of Science and Technology.

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2017).

Designing the user interface: Strategies for effective human–computer interaction (6th ed.). Pearson.

Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. Proceedings of the IEEE Symposium on Security and Privacy, 305–316.

Sommerville, I. (2016). Software engineering (10th ed.). Pearson.

Stallings, W. (2017). Effective cybersecurity: A guide to using best practices and standards. Addison-Wesley.

Syahbani, A. M., Firdaus, W., & Musodo, K. A. (2025). A comparative study of data mining algorithms for fraud detection in financial transactions. Sinkron: Jurnal dan Penelitian Teknik Informatika, 9(2). https://doi.org/10.33395/sinkron.v9i2.14645

Tanenbaum, A. S., & van Steen, M. (2017). Distributed systems (3rd ed.). Pearson.

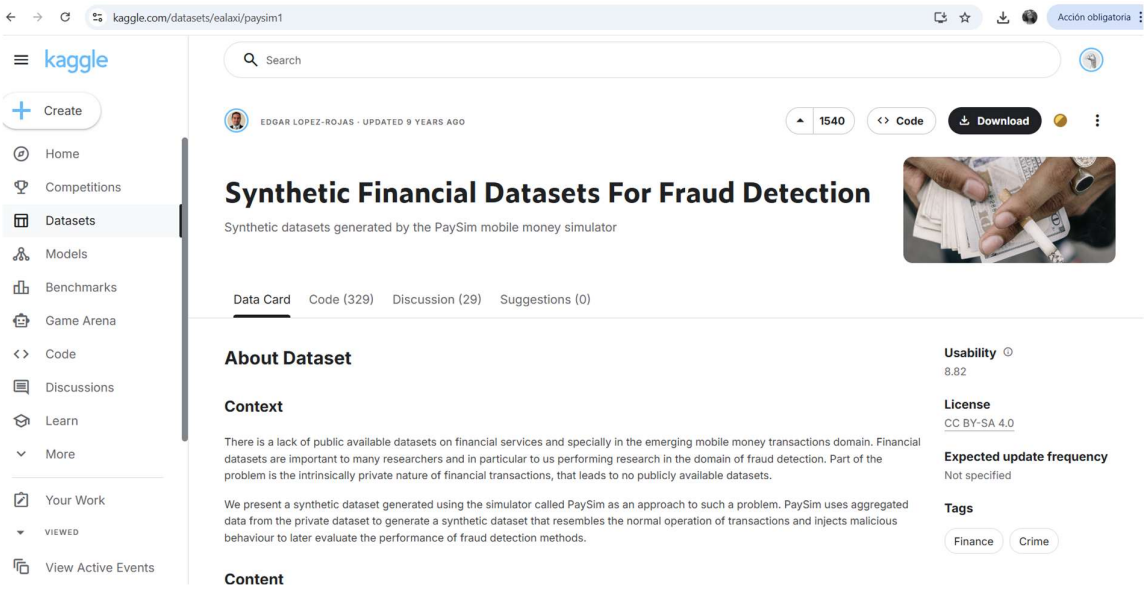
West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. Computers & Security, 57, 47–66. https://doi.org/10.1016/j.cose.2015.09.005

Wirth, R., & Hipp, J. (2000). Towards a standard process model for data mining. Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining.

# ANEXOS

## Figura 47

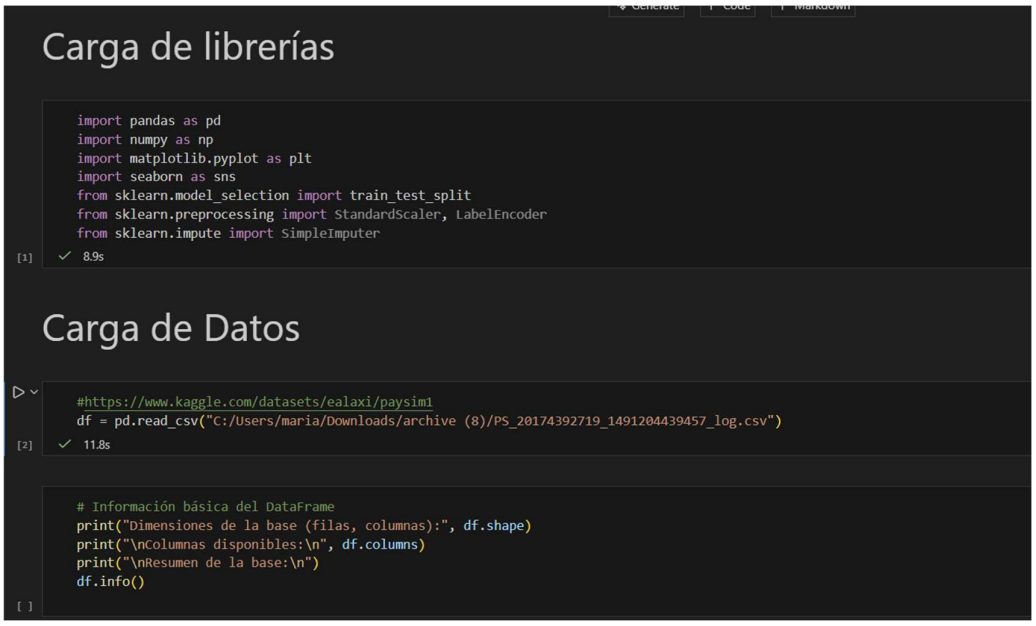
Anexo 1




***Fuente:*** Elaboración Propia

## *Figura 48*

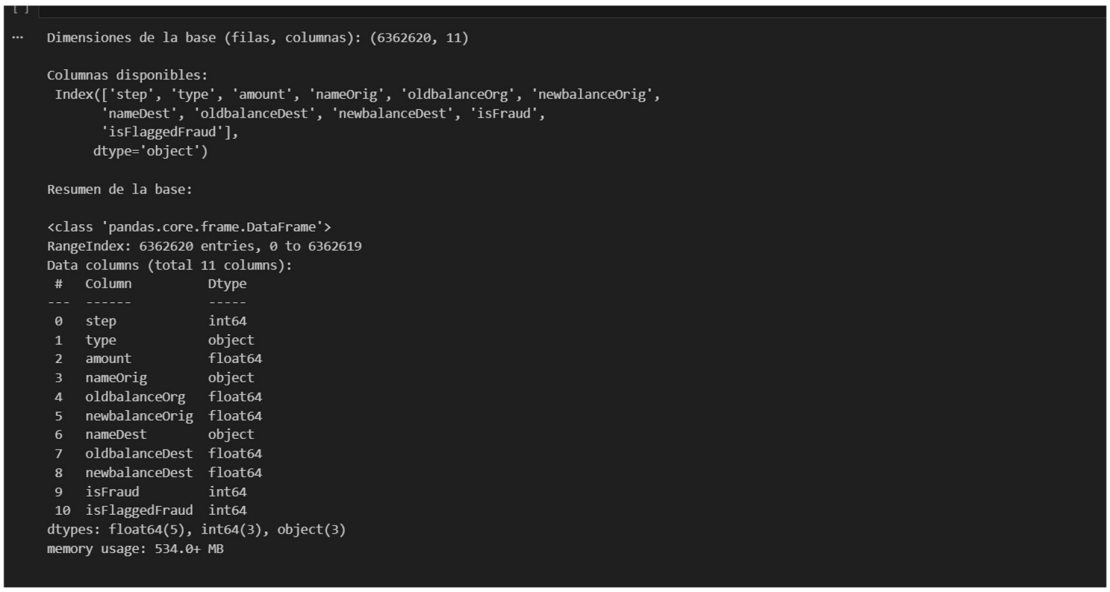
*Anexo 2*




***Fuente:*** Elaboración Propia

**Figura 49**

*Anexo 3*

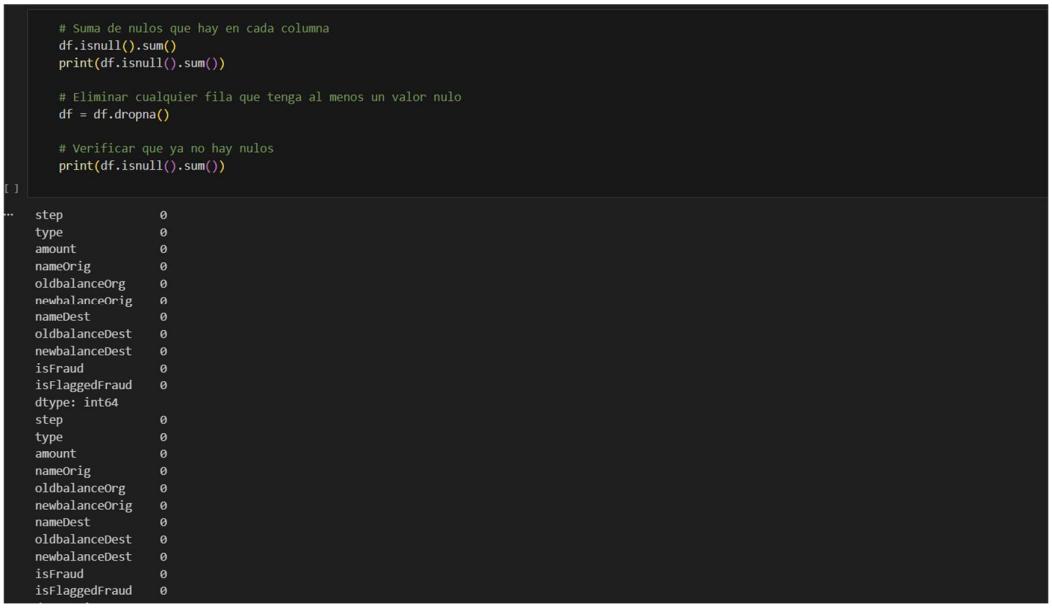


```
Dimensiones de la base (filas, columnas): (6362620, 11)

Columnas disponibles:
 Index(['step', 'type', 'amount', 'nameOrig', 'oldbalanceOrg', 'newbalanceOrig',
        'nameDest', 'oldbalanceDest', 'newbalanceDest', 'isFraud',
        'isFlaggedFraud'],
       dtype='object')

Resumen de la base:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
 #   Column          Dtype
---  ------          -----
 0   step            int64
 1   type            object
 2   amount          float64
 3   nameOrig        object
 4   oldbalanceOrg   float64
 5   newbalanceOrig  float64
 6   nameDest        object
 7   oldbalanceDest  float64
 8   newbalanceDest  float64
 9   isFraud         int64
 10  isFlaggedFraud  int64
dtypes: float64(5), int64(3), object(3)
memory usage: 534.0+ MB
```

***Fuente:*** Elaboración Propia

***Figura 50***

*Anexo 4*



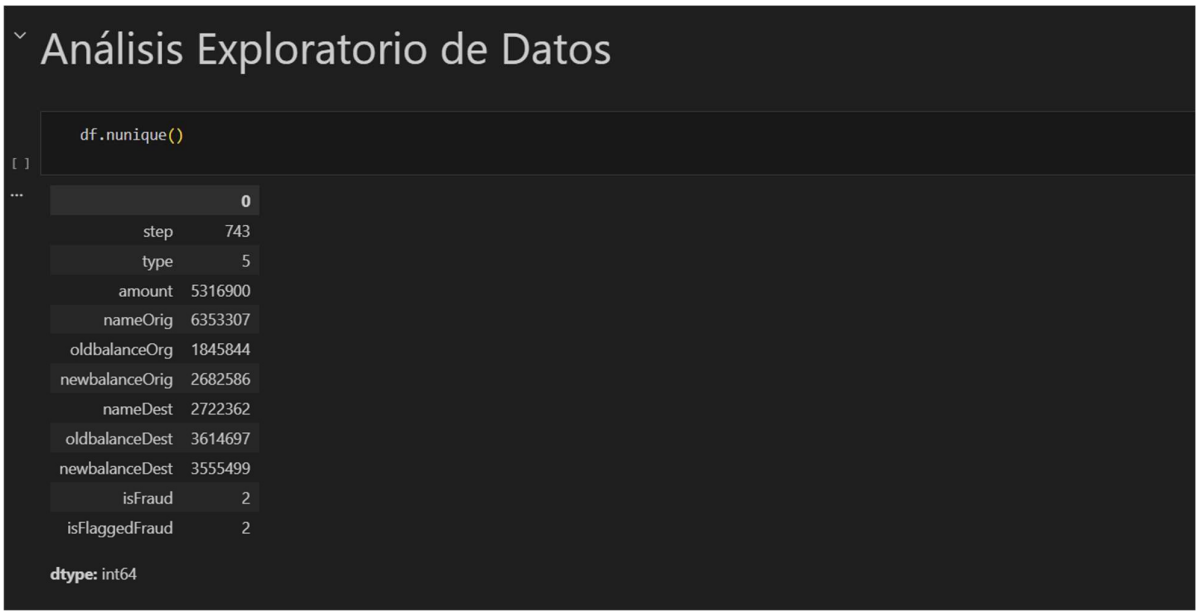
```
# Suma de nulos que hay en cada columna
df.isnull().sum()
print(df.isnull().sum())

# Eliminar cualquier fila que tenga al menos un valor nulo
df = df.dropna()

# Verificar que ya no hay nulos
print(df.isnull().sum())
```

```
step            0
type            0
amount          0
nameOrig        0
oldbalanceOrg   0
newbalanceOrig  0
nameDest        0
oldbalanceDest  0
newbalanceDest  0
isFraud         0
isFlaggedFraud  0
dtype: int64
step            0
type            0
amount          0
nameOrig        0
oldbalanceOrg   0
newbalanceOrig  0
nameDest        0
oldbalanceDest  0
newbalanceDest  0
isFraud         0
isFlaggedFraud  0
```

***Fuente:*** Elaboración Propia

***Figura 51***

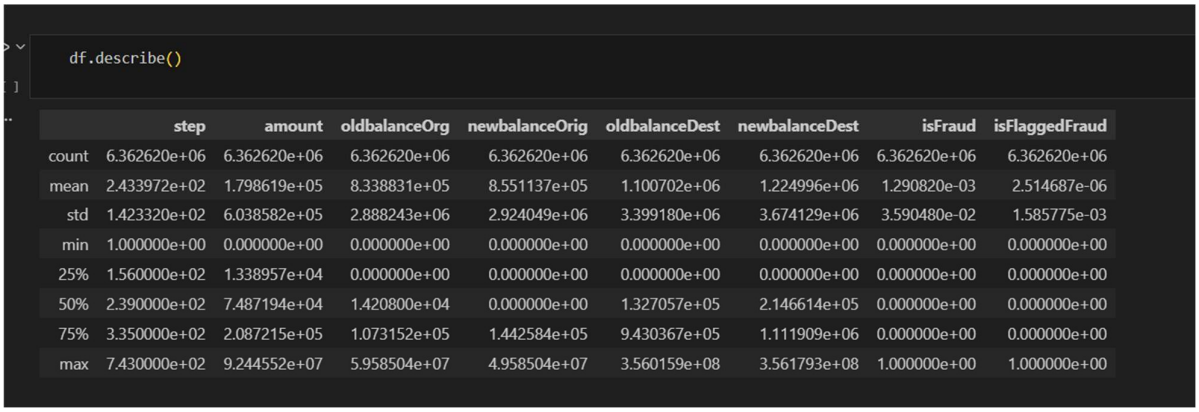
*Anexo 5*




***Fuente:*** Elaboración Propia

**Figura 52**

*Anexo 6*




| | step | amount | oldbalanceOrg | newbalanceOrig | oldbalanceDest | newbalanceDest | isFraud | isFlaggedFraud |
|---|---|---|---|---|---|---|---|---|
| count | 6.362620e+06 | 6.362620e+06 | 6.362620e+06 | 6.362620e+06 | 6.362620e+06 | 6.362620e+06 | 6.362620e+06 | 6.362620e+06 |
| mean | 2.433972e+02 | 1.798619e+05 | 8.338831e+05 | 8.551137e+05 | 1.100702e+06 | 1.224996e+06 | 1.290820e-03 | 2.514687e-06 |
| std | 1.423320e+02 | 6.038582e+05 | 2.888243e+06 | 2.924049e+06 | 3.399180e+06 | 3.674129e+06 | 3.590480e-02 | 1.585775e-03 |
| min | 1.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 25% | 1.560000e+02 | 1.338957e+04 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 50% | 2.390000e+02 | 7.487194e+04 | 1.420800e+04 | 0.000000e+00 | 1.327057e+05 | 2.146614e+05 | 0.000000e+00 | 0.000000e+00 |
| 75% | 3.350000e+02 | 2.087215e+05 | 1.073152e+05 | 1.442584e+05 | 9.430367e+05 | 1.111909e+06 | 0.000000e+00 | 0.000000e+00 |
| max | 7.430000e+02 | 9.244552e+07 | 5.958504e+07 | 4.958504e+07 | 3.560159e+08 | 3.561793e+08 | 1.000000e+00 | 1.000000e+00 |

***Fuente:*** Elaboración Propia

## *Figura 53*

*Anexo 7*

```python
# Usamos muestra estratificada
df_viz, _ = train_test_split(df, train_size=0.20, stratify=df['isFraud'], random_state=128)

fig, axes = plt.subplots(3, 2, figsize=(15, 12))
fig.suptitle('Distribución de Variables Numéricas (Manejo de Ceros en Log)', fontsize=16)

variables = [
    ('step', 'Paso de Tiempo (Horas)', False),
    ('amount', 'Monto de Transacción', True),
    ('oldbalanceOrg', 'Saldo Inicial Origen', True),
    ('newbalanceOrig', 'Saldo Final Origen', True),
    ('oldbalanceDest', 'Saldo Inicial Destino', True),
    ('newbalanceDest', 'Saldo Final Destino', True)
]

for i, (col, titulo, usar_log) in enumerate(variables):
    row, col_idx = divmod(i, 2)
    ax = axes[row, col_idx]

    # Si usamos log, filtramos temporalmente los valores <= 0 solo para este gráfico
    if usar_log:
        data_plot = df_viz[df_viz[col] > 0] # Solo positivos
    else:
        data_plot = df_viz # Todo el dataset

    # Graficamos usando los datos filtrados
    sns.histplot(data=data_plot, x=col, hue='isFraud',
                 bins=30, kde=True, element="step",
                 palette=['#99c2a2', '#ff9999'], ax=ax, log_scale=usar_log)

    ax.set_title(titulo)
    ax.set_ylabel('Frecuencia')
    ax.grid(alpha=0.3)
```

***Fuente:*** Elaboración Propia

## *Figura 54*

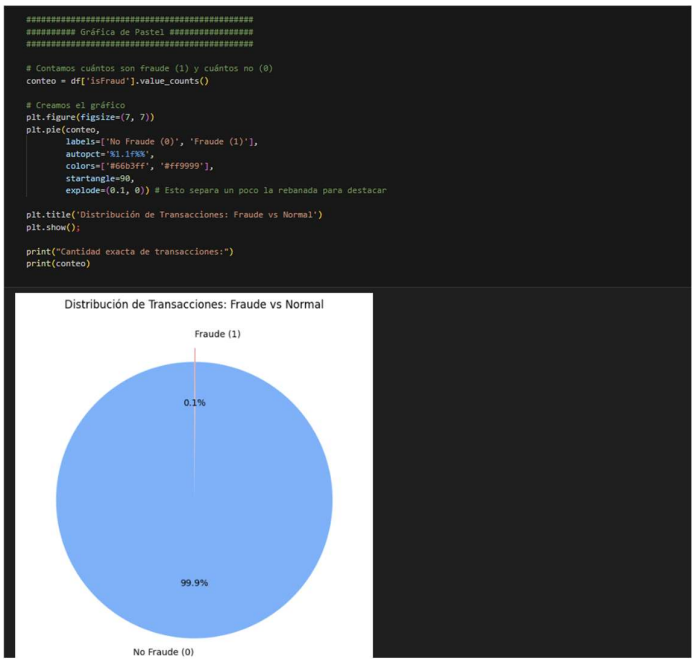
*Anexo 8*

```python
##### Matriz de Correlación #########
#####################################

# Seleccionamos las variables numéricas pero excluimos las columnas isFraud
df_sin_objetivo = df.select_dtypes(include=['float64', 'int64']).drop(['isFraud', 'isFlaggedFraud'], axis=1)
matriz_correlacion = df_sin_objetivo.corr()

# Graficamos
plt.figure(figsize=(10, 8))
sns.heatmap(matriz_correlacion,
            annot=True,
            fmt=".2f",
            cmap='coolwarm',
            linewidths=0.5)

plt.title('Correlación entre Variables Independientes (Sin isFraud)')
plt.show();

#####################################
##### Tabla de Correlación #########
#####################################

# Seleccionamos las variables numéricas menos 'isFraud' y 'isFlaggedFraud'
df_analisis = df.select_dtypes(include=['number']).drop(['isFraud', 'isFlaggedFraud'], axis=1, errors='ignore')

# Creamos la matriz de corrleación
# unstack() convierte la matriz cuadrada en una lista larga de pares
corr_matrix = df_analisis.corr()
pares_corr = corr_matrix.unstack()
ordenados = pares_corr.sort_values(ascending=False)

# Quitamos la correlación de una variable consigo misma (que siempre es 1.0)
ordenados = ordenados[ordenados != 1.0]

# Imprimimos el top 10 de correlaciones más altas (positivas)
print("\n--- Mayores Correlaciones Positivas ---")
print(ordenados.head(10))

print("\n--- Mayores Correlaciones Negativas (Inversas) ---")
# Mostramos las del final de la lista (las más negativas)
print(ordenados.tail(5))
```


***Fuente:*** Elaboración Propia

**Figura 55**

*Anexo 9*




***Fuente:*** Elaboración Propia

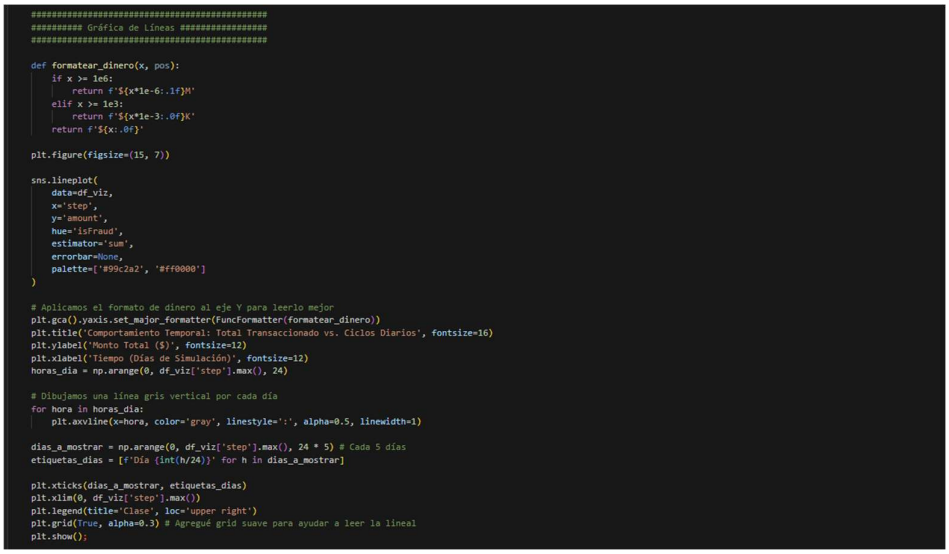
**Figura 56**

Anexo 10

***Fuente:*** Elaboración Propia

**Figura 57**

Anexo 11

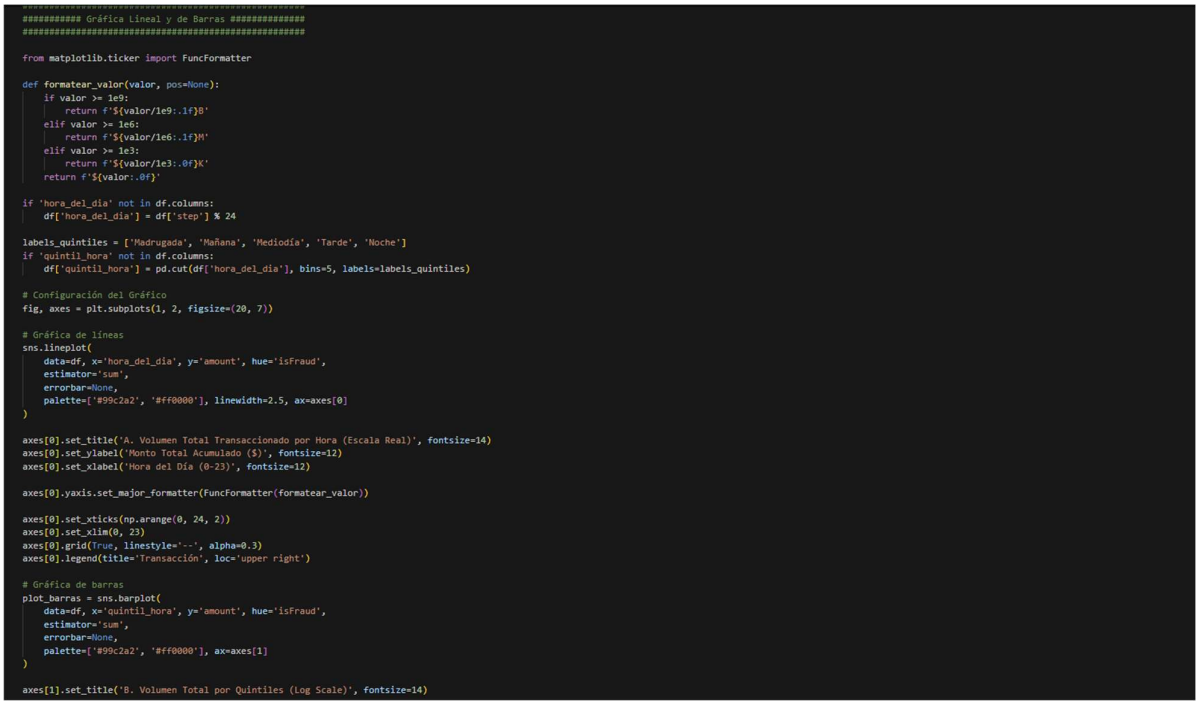


```
##########################################
########## Gráfica de Líneas ##############
##########################################

def formatear_dinero(x, pos):
    if x >= 1e6:
        return f'${x*1e-6:.1f}M'
    elif x >= 1e3:
        return f'${x*1e-3:.0f}K'
    return f'${x:.0f}'

plt.figure(figsize=(15, 7))

sns.lineplot(
    data=df_viz,
    x='step',
    y='amount',
    hue='isFraud',
    estimator='sum',
    errorbar=None,
    palette=['#99c2a2', '#ff0000']
)

# Aplicamos el formato de dinero al eje Y para leerlo mejor
plt.gca().yaxis.set_major_formatter(FuncFormatter(formatear_dinero))
plt.title('Comportamiento Temporal: Total Transaccionado vs. Ciclos Diarios', fontsize=16)
plt.ylabel('Monto Total ($)', fontsize=12)
plt.xlabel('Tiempo (Días de Simulación)', fontsize=12)
horas_dia = np.arange(0, df_viz['step'].max(), 24)

# Dibujamos una línea gris vertical por cada día
for hora in horas_dia:
    plt.axvline(x=hora, color='gray', linestyle=':', alpha=0.5, linewidth=1)

dias_a_mostrar = np.arange(0, df_viz['step'].max(), 24 * 5) # Cada 5 días
etiquetas_dias = [f'Día {int(h/24)}' for h in dias_a_mostrar]

plt.xticks(dias_a_mostrar, etiquetas_dias)
plt.xlim(0, df_viz['step'].max())
plt.legend(title='Clase', loc='upper right')
plt.grid(True, alpha=0.3) # Agregué grid suave para ayudar a leer la línea
plt.show();
```

***Fuente:*** Elaboración Propia

***Figura 58***

*Anexo 12*



```
##########################################################
########## Gráfica Lineal y de Barras #############
##########################################################

from matplotlib.ticker import FuncFormatter

def formatear_valor(valor, pos=None):
    if valor >= 1e9:
        return f'${valor/1e9:.1f}B'
    elif valor >= 1e6:
        return f'${valor/1e6:.1f}M'
    elif valor >= 1e3:
        return f'${valor/1e3:.0f}K'
    return f'${valor:.0f}'

if 'hora_del_dia' not in df.columns:
    df['hora_del_dia'] = df['step'] % 24

labels_quintiles = ['Madrugada', 'Mañana', 'Mediodía', 'Tarde', 'Noche']
if 'quintil_hora' not in df.columns:
    df['quintil_hora'] = pd.cut(df['hora_del_dia'], bins=5, labels=labels_quintiles)

# Configuración del Gráfico
fig, axes = plt.subplots(1, 2, figsize=(20, 7))

# Gráfica de líneas
sns.lineplot(
    data=df, x='hora_del_dia', y='amount', hue='isFraud',
    estimator='sum',
    errorbar=None,
    palette=['#99c2a2', '#ff0000'], linewidth=2.5, ax=axes[0]
)

axes[0].set_title('A. Volumen Total Transaccionado por Hora (Escala Real)', fontsize=14)
axes[0].set_ylabel('Monto Total Acumulado ($)', fontsize=12)
axes[0].set_xlabel('Hora del Día (0-23)', fontsize=12)

axes[0].yaxis.set_major_formatter(FuncFormatter(formatear_valor))

axes[0].set_xticks(np.arange(0, 24, 2))
axes[0].set_xlim(0, 23)
axes[0].grid(True, linestyle='--', alpha=0.3)
axes[0].legend(title='Transacción', loc='upper right')

# Gráfica de barras
plot_barras = sns.barplot(
    data=df, x='quintil_hora', y='amount', hue='isFraud',
    estimator='sum',
    errorbar=None,
    palette=['#99c2a2', '#ff0000'], ax=axes[1]
)

axes[1].set_title('B. Volumen Total por Quintiles (Log Scale)', fontsize=14)
```

***Fuente:*** Elaboración Propia

**Figura 59**

*Anexo 13*

```
pd.options.display.float_format = '{:.2f}'.format  # Para ver números normales, no científicos

# Agrupamos por isFraud y pedimos estadísticas de la columna 'amount'
stats_montos = df.groupby('isFraud')['amount'].describe()
print("Estadísticas de montos (Dinero transferido):")
print(stats_montos)
```

```
Estadísticas de montos (Dinero transferido):
            count       mean        std  min       25%       50%        75%  \
isFraud
0       6354407.00  178197.04  596236.98 0.01  13368.40  74684.72  208364.76
1          8213.00 1467967.30 2404252.95 0.00 127091.33 441423.44 1517771.48

                max
isFraud
0       92445516.64
1       10000000.00
```

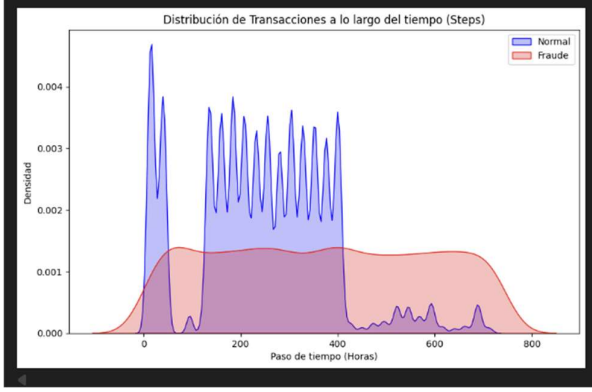
***Fuente:*** Elaboración Propia

**Figura 60**

*Anexo 14*

```
##############################################
######## Gráfica de Distribución #############
##############################################

plt.figure(figsize=(10, 6))

# Graficamos dos distribuciones superpuestas
sns.kdeplot(df[df['isFraud']==0]['step'], label='Normal', shade=True, color='blue')
sns.kdeplot(df[df['isFraud']==1]['step'], label='Fraude', shade=True, color='red')

plt.title('Distribución de Transacciones a lo largo del tiempo (Steps)')
plt.xlabel('Paso de tiempo (Horas)')
plt.ylabel('Densidad')
plt.legend()
plt.show()
```

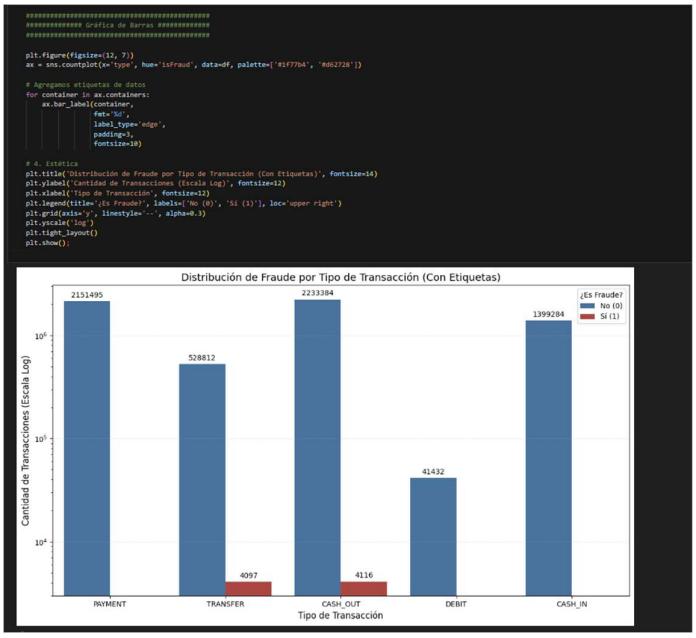
```
/tmp/ipython-input-1455721561.py:8: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['isFraud']==0]['step'], label='Normal', shade=True, color='blue')
/tmp/ipython-input-1455721561.py:9: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['isFraud']==1]['step'], label='Fraude', shade=True, color='red')
```




***Fuente:*** Elaboración Propia

## Figura 61

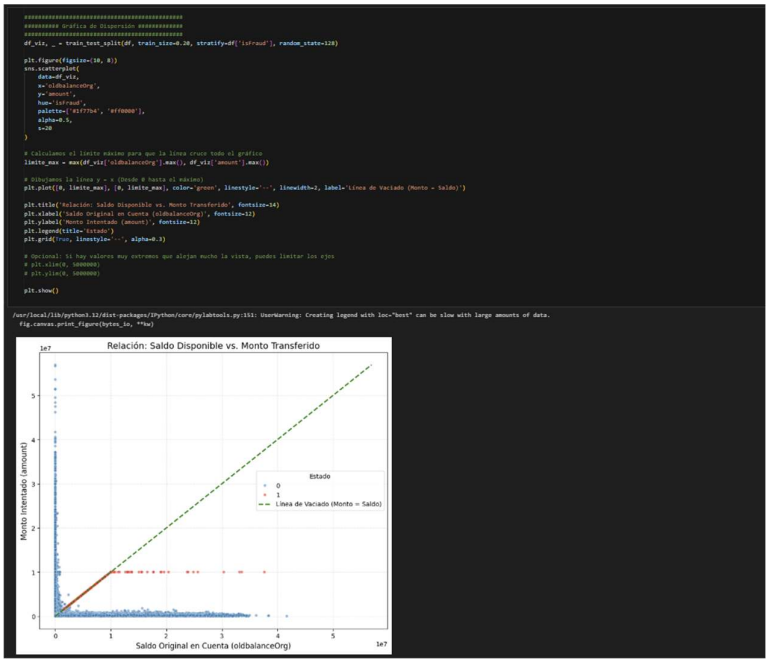
*Anexo 15*




***Fuente:*** Elaboración Propia

## Figura 62

*Anexo 16*




***Fuente:*** Elaboración Propia

**Figura 63**

Anexo 17

```
    # 1. ¿Cuántas transacciones de 0 pesos hay?
    transacciones_cero = df[df['amount'] == 0]
    print(f"Cantidad de transacciones de monto 0: {len(transacciones_cero)}")

    # 2. ¿Alguna de esas es fraude?
    print("¿Son fraude las transacciones de 0?:")
    print(transacciones_cero['isFraud'].value_counts())
```

```
Cantidad de transacciones de monto 0: 16
¿Son fraude las transacciones de 0?:
isFraud
1    16
Name: count, dtype: int64
```

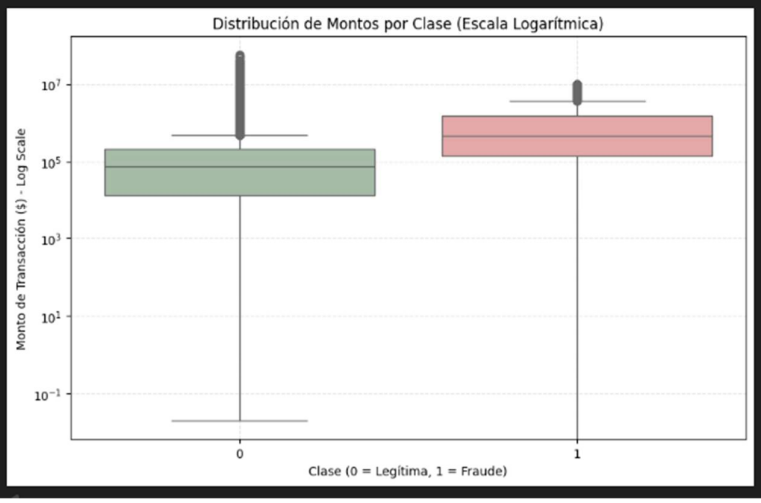
***Fuente:*** Elaboración Propia

**Figura 64**

*Anexo 18*

```
    ################################################
    ############ Diagrama de Caja ##############
    ################################################
    df_viz, _ = train_test_split(df, train_size=0.20, stratify=df['isFraud'], random_state=128)

    plt.figure(figsize=(10, 6))

    sns.boxplot(x='isFraud', y='amount', data=df_viz, palette=['#99c2a2', '#ff9999'])
    plt.yscale('log')
    plt.title('Distribución de Montos por Clase (Escala Logarítmica)')
    plt.ylabel('Monto de Transacción ($) - Log Scale')
    plt.xlabel('Clase (0 = Legítima, 1 = Fraude)')
    plt.grid(True, which="both", ls="--", alpha=0.3)
    plt.show()
```

```
/tmp/ipython-input-386321600.py:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.boxplot(x='isFraud', y='amount', data=df_viz, palette=['#99c2a2', '#ff9999'])
```




***Fuente:*** Elaboración Propia

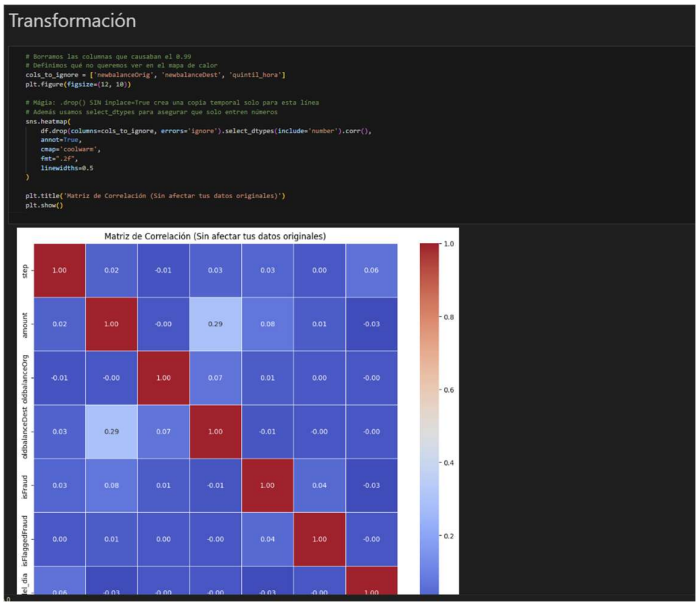
## Figura 65

*Anexo 19*


***Fuente:*** Elaboración Propia

## Figura 66

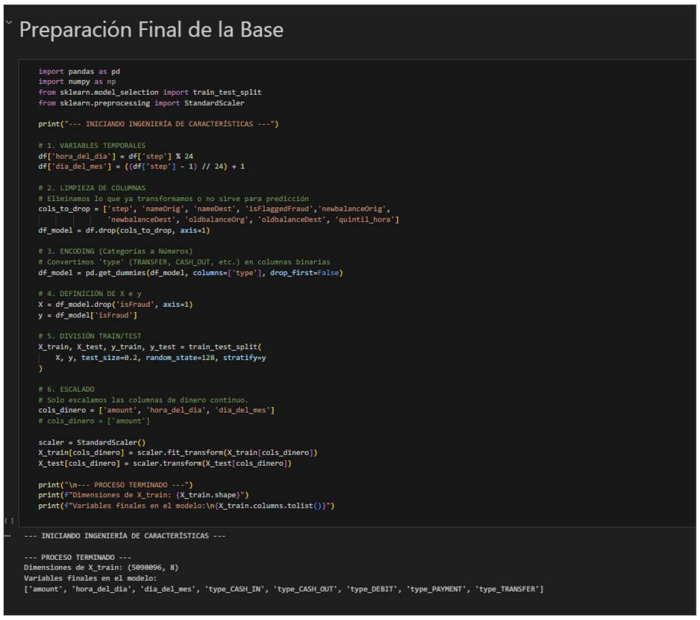
*Anexo 20*




***Fuente:*** Elaboración Propia

**Figura 67**

Anexo 21




***Fuente:*** Elaboración Propia

**Figura 68**

Anexo 22


***Fuente:*** Elaboración Propia