UNIVERSIDAD ESAN

FACULTAD DE INGENIERÍA

INGENIERÍA INDUSTRIAL Y COMERCIAL

INGENIERÍA DE SISTEMAS

Evaluación de modelos de Machine Learning para la priorización inteligente de vulnerabilidades en el sector bancario

Trabajo de Suficiencia Profesional presentado en satisfacción parcial de los requerimientos para:

Obtener el título profesional de Ingeniero(a) Industrial y Comercial

Obtener el título profesional de Ingeniero(a) de Sistemas

AUTORES

De la Cruz Huaman, Robin Willans

Perez Velasquez, Andrea Milagros

Torres Andrade, Angela Nicole

ASESOR

Fabian Arteaga, Junior Jhon

ORCID N° 0000-0001-9804-7795

Octubre, 2025

INFORME DE ORIGINALIDAD

3%	3%	0%	1%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	Submitted to Universidad ESAN -- Escuela de Administración de Negocios para Graduados Trabajo del estudiante	1%
2	www.ibm.com Fuente de Internet	1%
3	hdl.handle.net Fuente de Internet	1%

Excluir citas	Activo	Excluir coincidencias	< 1%
Excluir bibliografía	Activo		

RESUMEN

El estudio aplica técnicas de *Machine Learning* para optimizar la gestión de vulnerabilidades en el sector bancario, utilizando información extraída de la herramienta *Qualys* con cinco clases de severidad. Se implementaron modelos supervisados (*Random Forest* y *XGBoost*) para predecir la severidad y un modelo no supervisado (*K-Means*, k = 13) para segmentar vulnerabilidades en familias técnicas, facilitando la organización de la aplicación de parches por sistema operativo o servicio. Los modelos supervisados lograron una exactitud promedio de 81% - 82% y un *recall* superior al 90% en clases críticas, demostrando su eficacia para la priorización inteligente. No obstante, se observó un menor desempeño en clases minoritarias, lo que sugiere incorporar mayor balanceo y calibración probabilística. El modelo no supervisado confirmó patrones consistentes, destacando la predominancia de entornos *Windows Server* 2016–2022 y la persistencia de sistemas *legacy* (Solaris). Se concluye que el enfoque mixto, clasificación predictiva y segmentación estratégica, fortalece la toma de decisiones y reduce tiempos de análisis. La mejora futura del modelo requiere ampliar variables contextuales y automatizar el pipeline de actualización.

PALABRA CLAVE**:** *Machine Learning*, vulnerabilidades, severidad, *Random Forest*, *XGBoost, K-Means*, ciberseguridad bancaria.

ABSTRACT

This study applies Machine Learning techniques to optimize vulnerability management in the banking sector, using data extracted from the Qualys platform with five severity classes. Supervised models (Random Forest and XGBoost) were implemented to predict severity levels, and an unsupervised model (K-Means, k = 13) was used to segment vulnerabilities into technical families, facilitating patch management by operating system or service. The supervised models achieved an average accuracy of 81–82% and a recall above 90% for critical classes, demonstrating their effectiveness for intelligent prioritization. However, lower performance was observed for minority classes, suggesting the need for improved balancing and probability calibration. The unsupervised model confirmed consistent clustering patterns, highlighting the predominance of Windows Server 2016–2022 environments and the persistence of legacy systems (Solaris, CGI). It is concluded that the combined approach — predictive classification and strategic segmentation— strengthens decision-making and reduces analysis time. Future improvements should include additional contextual variables and an automated update pipeline to enhance model robustness.

KEYWORD: Machine Learning, vulnerabilities, severity, Random Forest, XGBoost, K-Means, banking cybersecurity.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE FÓRMULAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

El crecimiento exponencial de la digitalización en el sector financiero ha transformado la manera en que las instituciones bancarias gestionan sus riesgos tecnológicos. La incorporación de nuevos servicios digitales, el uso extendido de arquitecturas híbridas y la creciente interconectividad entre sistemas han incrementado considerablemente la superficie de exposición a ciberamenazas. Este escenario, marcado por un aumento sostenido en la frecuencia y complejidad de los ataques, ha generado la necesidad de contar con mecanismos más precisos y automatizados para la identificación, clasificación y priorización de vulnerabilidades.

Tradicionalmente, las organizaciones han recurrido a metodologías basadas en el *Common Vulnerability Scoring System* (CVSS) para estimar el nivel de riesgo técnico de una vulnerabilidad. Sin embargo, este enfoque resulta limitado al no incorporar variables contextuales, como la explotación activa, la criticidad del activo o la exposición temporal. En consecuencia, la priorización basada únicamente en puntajes estáticos no siempre refleja el riesgo operativo real ni permite una asignación óptima de recursos para la remediación.

Frente a estas limitaciones, el presente estudio propone un enfoque analítico basado en técnicas de *Machine Learning* que combinan modelos supervisados y no supervisados. A través de los algoritmos *Random Forest* y *XGBoost*, se busca predecir con mayor precisión la severidad de las vulnerabilidades, mientras que mediante el algoritmo *K-Means* se segmentan los hallazgos técnicos en grupos o familias afines, organizando el proceso de aplicación de parcheo por tipo de sistema o servicio. Esta visión híbrida, que integra clasificación predictiva y segmentación estratégica, ofrece una lectura complementaria, ya que permite determinar qué vulnerabilidades atender primero y cómo abordarlas de manera eficiente, fortaleciendo la toma de decisiones dentro de un marco de ciberseguridad proactiva.

Este proyecto se estructura en seis capítulos principales, cada uno abordando aspectos fundamentales de la investigación. En el Capítulo I se establece la base del estudio, describiendo la realidad problemática, formulando el problema general y los problemas específicos, definiendo los objetivos y justificando la importancia del trabajo. En el Capítulo II se desarrolla el marco teórico, que comprende los antecedentes de la investigación, los fundamentos conceptuales y el contexto de aplicación en ciberseguridad bancaria. En el Capítulo III se detalla la organización seleccionada como caso de estudio, describiendo su entorno operativo, los sistemas involucrados y la naturaleza de los datos utilizados. El Capítulo IV aborda el diseño metodológico, explicando el tipo de investigación, las técnicas de preprocesamiento, las herramientas empleadas y la lógica de los modelos supervisados y no supervisados. En el

Capítulo V se presenta el desarrollo de la solución, incluyendo los procesos de entrenamiento, validación, simulación con datos reales y la interpretación de los resultados obtenidos. Finalmente, el Capítulo VI expone las conclusiones y recomendaciones derivadas de los hallazgos, resaltando las implicancias operativas y las oportunidades de mejora para futuras investigaciones.

CAPÍTULO I: Planteamiento del Problema

1.1. Descripción de la realidad problemática

Durante la última década, la digitalización intensa bajo el marco de la Industria 4.0 ha ocasionado un aumento significativo de la superficie de ataque en sistemas tecnológicos (Vaidya et al., 2021). Esto ha propiciado un incremento sostenido y constante en el número de vulnerabilidades reportadas, con cifras del *National Vulnerability Database* (NVD) que revelan un crecimiento de más de 17 000 *Common Vulnerabilities and Exposures* (CVEs) nuevos al año desde 2020 (NIST, 2020). Para cuantificar la gravedad técnica de cada vulnerabilidad frente a esta avalancha, las organizaciones se apoyan en el *Common Vulnerability Scoring System* (CVSS), aunque esta calificación no refleja siempre el contexto operativo real (Elementrica, 2025).

Sin embargo, las investigaciones más actuales evidencian que el CVSS ha estado enfrentando limitaciones importantes desde 2020. Según un reporte de Wunder et al. (2023), aproximadamente el 68% de los analistas otorgan puntuaciones diferentes a las mismas vulnerabilidades, lo que muestra una alta variabilidad subjetiva en la evaluación. De igual manera, estudios como el de DarkReading encontraron discrepancias entre las calificaciones oficiales de CVSS en NVD y los análisis de los proveedores (Ouzan O, 2025), lo que refuerza la noción de que la puntuación no representa siempre la realidad operativa. Este problema se agrava debido a que numerosas vulnerabilidades catalogadas como "media" o "alta" no son utilizadas en escenarios reales, pero otras con puntuaciones aparentemente más bajas sí suponen un riesgo verdadero.

La falta de dinamismo y elementos operativos en el sistema CVSS intensifica estas limitaciones. Según Elementrica (2024), el modelo no incluye datos sobre la criticidad del activo, la explotación activa en el entorno ni el contexto de explotación. Esto provoca una divergencia entre la teoría de puntuación y el riesgo real. En plataformas como Reddit, especialistas en ciberseguridad señalaron ejemplos de vulnerabilidades con puntuación media (CVSS ≈ 5.5) que fueron incorporadas a la lista "*Known Exploited Vulnerabilities*" de CISA, lo que demuestra que el sistema estándar no garantiza siempre la prevención de eventos críticos.

Este reto es especialmente delicado en áreas con un alto nivel de regulación y riesgo, como el sistema bancario. Las entidades financieras, además de gestionar información sensible y confidencial, deben cumplir con marcos regulatorios estrictos en materia de ciberseguridad, como PCI-DSS. En este escenario, una categorización de vulnerabilidades que sea tardía o inexacta podría tener serias implicaciones en términos económicos, legales y reputacionales

(Kovacevic et al., 2024). Asimismo, los sistemas de la banca son blancos comunes de ataques persistentes avanzados (APT), *ransomware* dirigido y fraudes automatizados. Esto requiere habilidades predictivas más complejas que las que brindan los modelos convencionales.

En el caso particular de la entidad bancaria analizada, la gestión de vulnerabilidades se realiza mediante la plataforma *Qualys*, la cual permite detectar y calificar vulnerabilidades de infraestructura y aplicaciones. Sin embargo, el proceso de priorización actual se basa casi exclusivamente en los puntajes CVSS y en la revisión manual por parte de analistas técnicos. Esto ocasiona demoras en la atención de vulnerabilidades críticas y una asignación poco uniforme de prioridades entre países y sistemas.

En promedio, la organización logra atender aproximadamente el 62% de las vulnerabilidades críticas dentro de un periodo de 30 días, cifra ligeramente inferior al promedio regional reportado por entidades financieras latinoamericanas, cuyo rango se encuentra entre 65% y 70% (Kovacevic et al., 2024). Este valor constituye la línea base operativa de referencia sobre la cual se propone evaluar la mejora alcanzada mediante la aplicación de modelos de *Machine Learning*.

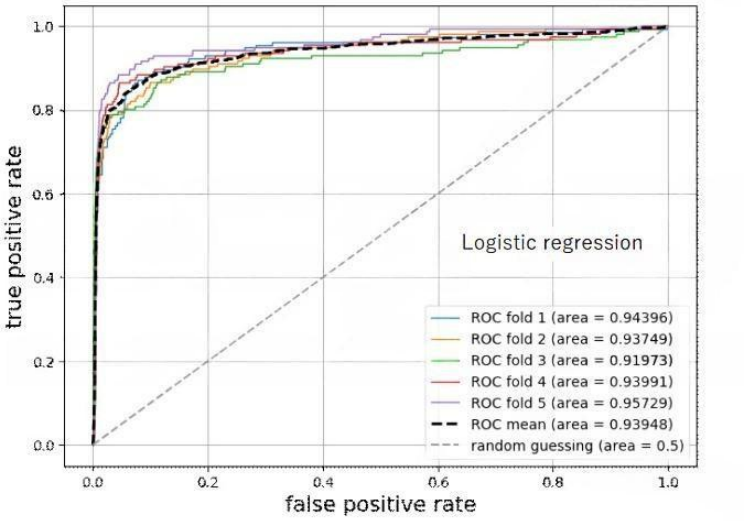
Desde una perspectiva de mejora continua, este proyecto se concibe como el primer paso hacia un modelo predictivo y automatizado de gestión de vulnerabilidades, en el que la meta es superar el umbral del 80% de atención efectiva y reducir los tiempos de análisis manual. Esta visión permite definir de forma objetiva el punto de partida, la meta y el impacto esperado dentro de la organización, estableciendo una línea de referencia cuantificable para futuras iteraciones de madurez tecnológica.

En vista de este panorama, los métodos de aprendizaje automático supervisados y no supervisados han surgido como una solución poderosa. Para prever la gravedad de las vulnerabilidades a partir de atributos de los CVEs, se han utilizado con frecuencia modelos supervisados, como *Random Forest* y *XGBoost*, así como no supervisados, como *K-Means Clustering*. Por ejemplo, investigaciones recientes que utilizaron *XGBoost* después de aplicar reducción de dimensionalidad y extracción de texto (*text mining*) lograron una precisión del 92.9%, la cual es mayor a la que se obtiene con métodos tradicionales (Huff & Li, 2021). Igualmente, se emplearon algoritmos de comparación (*SVM, Random Forest, XGBoost*) para anticipar si una vulnerabilidad requería un aviso de seguridad. En esta etapa, *XGBoost* y *Random Forest* lograron puntuaciones ROC-AUC de 0.96 y PR-AUC que sobrepasaron 0.62, superando así a los modelos lineales. Por otro lado, el agrupamiento *K-Means* ha hecho posible optimizar la priorización inteligente al reunir vulnerabilidades con rasgos parecidos y permitir una toma de decisiones más estratégica. Este procedimiento, que se usa extensamente en

diversos campos financieros, contribuye a convertir datos complejos y voluminosos en conjuntos congruentes, lo que posibilita la detección de patrones significativos y la mejora de la distribución de recursos.

Figura 1

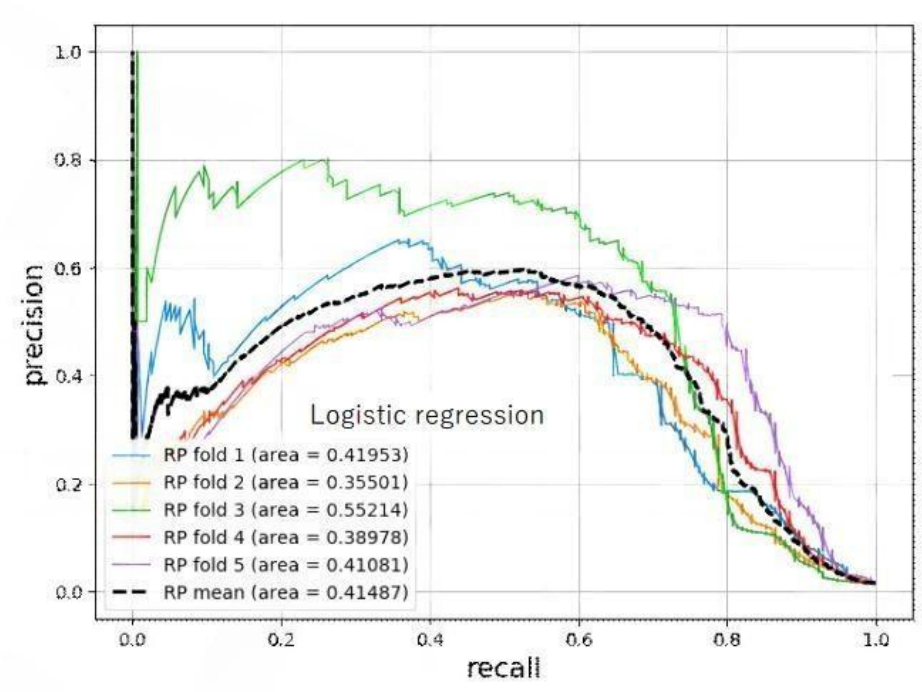
Curva ROC para regresión logística



Fuente: SCITEPRESS. (2023)

Figura 2

Curva PR para regresión logística



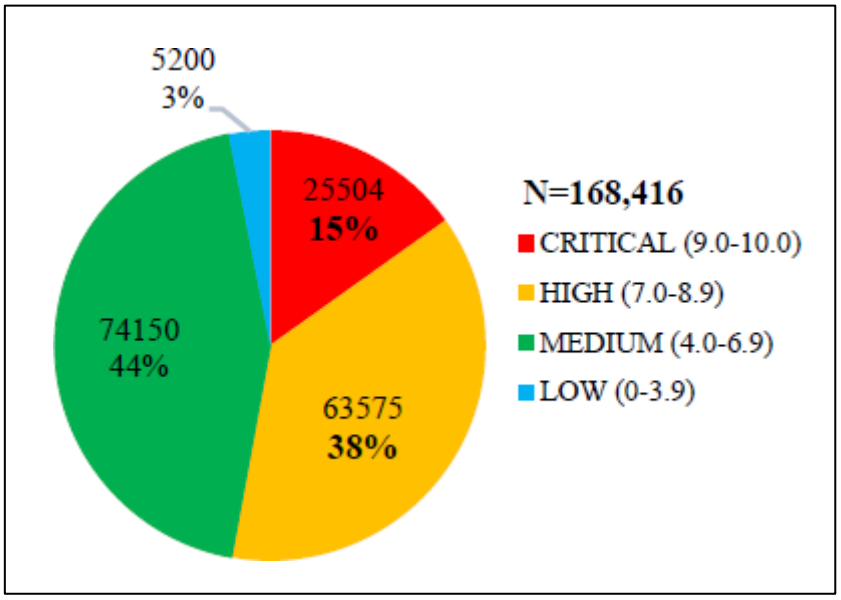
Fuente: SCITEPRESS. (2023)

En el ámbito bancario, ya se ha comprobado la aplicación de estos modelos en casos como la detección del fraude, la evaluación de riesgo crediticio y la predicción del comportamiento de los usuarios. Hafid (2025) evidenció, por ejemplo, que *XGBoost* superó a *Random Forest* en cuanto a rendimiento en la detección de operaciones fraudulentas en bancos, con una exactitud de más del 99%, incluso cuando los datos estaban desbalanceados. Esta experiencia demuestra que *XGBoost* no solo tiene una alta capacidad predictiva, sino que también se ajusta adecuadamente a situaciones en las que la clase crítica (vulnerabilidad severa) es rara, como sucede en el ámbito de la seguridad bancaria.

Con técnicas explicables (XAI), *XGBoost* se ha empleado con éxito. Por ejemplo, en 2022 se publicó un artículo en MDPI que hizo uso de SHAP con *XGBoost* para producir reglas transparentes acerca de la severidad, lo cual aumentó la confianza en el sistema. Este método no solo proporciona un alto grado de precisión, sino también la capacidad de explicar, que resulta indispensable para que se tomen decisiones en ciberseguridad. Además, *Random Forest* se ha usado en investigaciones que se fundamentan en ontologías de CVEs y las conexiones entre productos, logrando una precisión de entre 55% y 64% por cada categoría de gravedad, lo que evidencia su utilidad para incluir relaciones estructuradas.

Figura 3

Desglose de las puntuaciones base de los CVE



Fuente: Tsutsui et al. (2025)

Simultáneamente, *XGBoost* se ha empleado en el ámbito financiero para detectar patrones anómalos en sistemas de pago integrados. Qi Zheng et al. (2024) demostraron que se obtuvo una mejora significativa en la sensibilidad y precisión del modelo para detectar

amenazas en las redes bancarias cuando se utilizó *XGBoost* junto con técnicas de sobre muestreo como SMOTE. Estos descubrimientos aumentan la factibilidad de implementar estos algoritmos no únicamente en ámbitos financieros generales, sino también en la priorización consciente de vulnerabilidades en el sector bancario según su contexto.

En consecuencia, la problemática no se limita únicamente a la existencia de vulnerabilidades, sino a la falta de mecanismos predictivos que permitan priorizarlas de acuerdo con su impacto real en los activos críticos del banco. De esta necesidad surge la motivación para aplicar técnicas de aprendizaje automático que integren análisis supervisados (para predicción de severidad) y no supervisados (para segmentación técnica), consolidando una gestión más proactiva, objetiva y auditable.

1.2. Formulación del Problema

Para determinar el problema general y los problemas específicos de la presente investigación se ha tomado en cuenta en lo que se centra el estudio el cual es la evaluación de modelos de *Machine Learning* para la priorización inteligente de vulnerabilidades en el sector bancario.

1.2.1. Problema general

PP: ¿En qué medida favorece la aplicación de modelos de *Machine Learning* a la priorización inteligente de vulnerabilidades en el sector bancario?

1.2.2 Problemas específicos

PE1: ¿Cuáles son las variables que se tomarán en cuenta en el desarrollo de los modelos de *Machine Learning* para la priorización inteligente de vulnerabilidades en el sector bancario?

PE2: ¿Cuáles son las técnicas de pre-procesamiento que se pueden aplicar a los datasets de vulnerabilidades de una empresa del sector bancario para realizar una limpieza de la data?

PE3: ¿Cuáles son las técnicas de aprendizaje automático que se pueden aplicar para predecir y priorizar la severidad de vulnerabilidades en una entidad del sector bancario?

PE4: ¿Qué métricas permiten evaluar la eficiencia de los modelos de *Machine Learning* para la priorización inteligente de vulnerabilidades en el sector bancario?

1.3. Objetivos de la Investigación

Se procedió a identificar tanto el objetivo general como los objetivos específicos de la investigación.

1.3.1. Objetivo general

Evaluar el funcionamiento de los modelos de *Machine Learning* basado en datos históricos para realizar la priorización inteligente de vulnerabilidades en el sector bancario.

1.3.2. Objetivos específicos.

OE1: Identificar las variables adecuadas para el uso de los modelos de *Machine Learning* y así poder realizar la priorización inteligente de vulnerabilidades en el sector bancario.

OE2: Implementar técnicas de pre-procesamiento que permitan realizar una limpieza de los datasets de vulnerabilidades de una empresa del sector bancario.

OE3: Aplicar técnicas de aprendizaje automático supervisadas y no supervisadas que permitan predecir la severidad y segmentar las vulnerabilidades por afinidad técnica, con el fin de optimizar la priorización inteligente en el sector bancario.

OE4: Utilizar métricas que posibiliten evaluar la eficiencia de los modelos de *Machine Learning* en cuanto a la priorización inteligente de vulnerabilidades en el sector bancario.

1.4. Justificación de la Investigación

1.4.1. Teórica

Históricamente, la categorización de la gravedad en las vulnerabilidades informáticas se ha abordado mediante métodos esenciales en normas y sistemas de conocimiento, así como modelos reconocidos, como el CVSS. Aunque estas técnicas son útiles, dependen en gran parte de criterios establecidos de manera general, lo que puede dar lugar a retrasos, prejuicios o irregularidades en su aplicación, especialmente en contextos con un alto volumen de información y fuentes variadas, como ocurre en el sector financiero.

Con el desarrollo de técnicas avanzadas de análisis como el *Machine Learning* y con la evolución del *Big Data*, se ha presentado la oportunidad de optimizar y automatizar procesos indispensables, por ejemplo, el descubrimiento de vulnerabilidades. Esto incluye una mayor cantidad de elementos contextuales, patrones históricos y factores que no toman en cuenta los métodos tradicionales.

En esta investigación, la implementación de métodos de *Machine Learning* no solo brinda una manera de mejorar la precisión del método, sino que también proporciona una base para investigaciones enfocadas en el control automatizado de riesgos en ciberseguridad bancaria.

1.4.2. Práctica

El sector bancario debe inspeccionar y priorizar constantemente una variedad de vulnerabilidades tecnológicas a fin de reducir los peligros en términos de seguridad. La clasificación adecuada de la gravedad de estas vulnerabilidades es esencial porque tiene un impacto en la urgencia con que se atienden y en cómo se distribuyen los recursos destinados a la ciberseguridad. Sin embargo, en numerosas ocasiones, este procedimiento se realiza de forma manual o con un grado parcial de automatización. Esto puede generar demoras, incongruencias y un uso ineficaz del tiempo y el talento humano.

En esta situación, el desarrollo de un modelo de *Machine Learning* que permita prever con exactitud la gravedad de nuevas vulnerabilidades es una herramienta muy útil. La aplicación de esto podría acelerar el proceso de análisis, disminuir la carga operacional sobre los equipos de seguridad y mejorar el proceso de toma de decisiones en situaciones con una elevada cantidad de alertas. Un modelo flexible y escalable haría posible la unificación de criterios de evaluación en un ambiente multirregional con bancos que operan en diferentes contextos, mejorando así la eficiencia y la capacidad de reacción frente a los peligros en toda la organización.

1.4.3. Metodológica

La metodología utilizada en esta investigación es experimental, aplicada y cuantitativa. Esta perspectiva se elige por varias razones esenciales:

- Tratamiento de grandes cantidades de información: Las instituciones bancarias, especialmente las que operan en diversas naciones, lidian cotidianamente con una cantidad significativa de reportes acerca de vulnerabilidades que provienen de una amplia gama de fuentes y aplicaciones.
- El uso de algoritmos de aprendizaje automático permite que el análisis de estos datos sea eficaz, al detectar tendencias pasadas y realizar proyecciones sin intervención humana, lo que resulta ideal para escenarios a gran escala.

Además de estos beneficios, la estructura metodológica del proyecto adoptó un enfoque ordenado y replicable:

- Recolección de datos: Se empleó un conjunto de datos que abarca vulnerabilidades registradas en la industria bancaria, que incluye detalles técnicos, clasificación, fechas, orígenes y grados de severidad establecidos.
- Preprocesamiento de datos: Los datos fueron limpiados, transformados y preparados para su análisis, utilizando métodos como la imputación de datos ausentes, la codificación de categorías y la normalización.

- Selección de atributos: Se utilizarán métodos de análisis preliminar y evaluación de variables para reconocer las características más significativas que afectan la categorización de la severidad.
- Construcción de modelos: Se elegirán varios algoritmos de *Machine Learning* para elaborar modelos de clasificación usando la información disponible.
- Evaluación y validación: Se llevará a cabo una evaluación de los modelos usando medidas como precisión, *recall* y *F1-score*, así como se validarán a través de métodos como la validación cruzada para garantizar la solidez de los resultados.

1.5. Delimitación del Estudio

1.5.1. Espacial

Esta investigación se enfocó en el sector bancario, utilizando datos sobre vulnerabilidades reportadas en una institución financiera multinacional. La investigación considera información histórica y reciente de diversos ecosistemas tecnológicos que emplea el banco de sus sedes de Colombia, Panamá y USA.

El análisis y la optimización de la aplicabilidad del modelo sugerido se enriquecen con datos que llegan de diferentes regiones, lo cual permite abordar un amplio rango de contextos operativos, reglas locales y estrategias para manejar ciberseguridad. Esta diversidad geográfica tiene como objetivo producir una solución versátil que sea capaz de anticipar la gravedad de las vulnerabilidades en diversos contextos bancarios reales.

1.5.2. Temporal

Como información de la serie de datos, se analizó un periodo de cuatro años, abarcando desde el 2020 hasta el 2024. Estos datos incluyen vulnerabilidades que se han documentado recientemente, lo que permite considerar incidentes actuales de ciberseguridad, así como patrones históricos recientes.

El intervalo elegido fue debido a que entre 2020 y 2024 se reflejó un periodo crucial de rápida transformación digital, caracterizado por el aumento de los peligros surgidos del trabajo remoto y la incorporación de tecnologías recientes. Asimismo, estos años brindan datos estandarizados y recientes que son lo suficientemente abundantes para llevar a cabo un análisis, equilibrando la información contemporánea con la disponibilidad de registros históricos que posibilitan detectar patrones.

1.5.3. Conceptual

Determinar los conceptos esenciales que guían la perspectiva analítica y metodológica de la investigación es crucial. A continuación, se definen los términos más importantes en el marco del proyecto:

Modelo de clasificación: Representación digital basada en algoritmos de aprendizaje automático, cuyo objetivo es clasificar o etiquetar una observación particular. En este estudio, se empleará el modelo de clasificación para evaluar el nivel de gravedad asociado a una vulnerabilidad detectada en los sistemas del sector bancario.

Random Forest: Se refiere a un algoritmo de aprendizaje supervisado, cuyo mecanismo consiste en crear varios árboles de decisión que se entrenan con muestras aleatorias de datos y variables. En clasificación, la predicción final se logra mediante una votación mayoritaria; en regresión, por medio de un promedio. En los contextos en los que la calidad de las predicciones es prioritaria, como en el sector de la ciberseguridad y financiero, resulta especialmente útil.

XGBoost: Alude a un algoritmo de aprendizaje supervisado que se basa en el método de *boosting*. A diferencia de *Random Forest*, su mecanismo consiste en generar, uno tras otro, distintos árboles de decisión, donde cada árbol nuevo corrige los fallos de los preexistentes. En clasificación, se logra la predicción final al fusionar las salidas de los árboles a través de un procedimiento de votación ponderada; en regresión, mediante un promedio modificado. *XGBoost* es particularmente beneficioso en situaciones en las que se necesita una alta precisión y eficiencia computacional, como las de la ciberseguridad y el ámbito financiero.

K-Means: Se trata de un algoritmo de aprendizaje no supervisado, cuyo funcionamiento consiste en dividir un conjunto de datos en una cantidad determinada de clústeres, mediante la reducción de la distancia entre los puntos y el centroide de cada uno. Se lleva a cabo el proceso de manera reiterativa, modificando los centroides hasta que se obtenga una distribución estable. *K-Means* es especialmente efectivo en situaciones donde se requiere detectar patrones o dividir información sin etiquetas anteriores, como ocurre con el análisis de clientes del sector financiero o la detección de anomalías.

Severidad de vulnerabilidades: Se refiere al grado de criticidad que se le asigna a una vulnerabilidad informática, en función del impacto que esta podría generar sobre la seguridad, la disponibilidad o la integridad del sistema dañado. En este estudio, esta gravedad se clasifica en niveles cuantitativos y se emplea para dar prioridad a las acciones de mitigación.

Entrenamiento y prueba del modelo: El entrenamiento consiste en suministrar al modelo datos etiquetados (es decir, vulnerabilidades con severidad conocida) para que sea capaz de

identificar patrones. Después, se evalúa la capacidad del modelo para realizar predicciones usando un conjunto de datos distinto (de prueba).

Precisión del modelo: Está vinculado con el número de aciertos que logra el modelo, especialmente en lo concerniente a la categorización adecuada de la severidad. Esta métrica se tiene en cuenta, junto con otras como la recuperación y el puntaje F1, como uno de los métodos más importantes para evaluar la eficiencia del modelo propuesto.

Datos históricos y actuales: Son los registros de vulnerabilidades que se han conseguido entre 2020-2024 y 2025, que incluyen metadatos, características técnicas, clasificaciones de severidad y otra información importante. Estos datos son la base para entrenar, validar y evaluar el modelo.

CAPÍTULO II: Marco Teórico

2.1. Antecedentes de la Investigación

A continuación, se tienen algunos antecedentes encontrados con el fin de comprender cómo se ha abordado el tema de investigación seleccionado en otros casos, así como los objetivos, metodologías y soluciones alcanzadas. A continuación, un breve análisis de los antecedentes revisados:

Tabla 1

Breve análisis de antecedentes

#	Antecedente	Año	Base de datos	Técnicas	Resultado
1	Advancing Vulnerability Classification with BERT: A Multi-Objective Learning Model	2025	Conjunto de datos de 5637 entradas, con etiquetas de gravedad (Baja, Media, Alta, Crítica)	Modelo BERT	217 verdaderos positivos para Bajo, 268 para Medio, 288 para Alto y 190 para Crítico, lo que refleja predicciones precisas para la mayoría de las muestras
2	Vulnerability Severity Prediction with Deep Neural Network	2019	Conjunto de 82 974 vulnerabilidades extraídas de la NVD, con severidad etiquetada según niveles y puntajes absolutos	*XGBoost* CNN LSTM TextRCNN	En predicción de puntaje continuo, *XGBoost* obtuvo MAPE = 18.91%; LSTM fue más exacto, pero *XGBoost* ofreció rapidez y simplicidad
3	A Novel Deep-Learning-Based Bug Severity Classification Technique Using Convolutional Neural.	2019	Informes reales de errores de cinco proyectos de software de código abierto, extraídos de plataformas	BCR (Bug Classification using CNN + RF + Boosting)	Más de 12 puntos de mejora en comparación con métodos previos de clasificación de bugs.

			como Bugzilla y JIRA.		
4	Enhancing cybersecurity vulnerability detection using different *Machine Learning* severity prediction models	2025	Vulnerabilidades de CISA 2022 con severidad conocida	Regresión Logística Árbol de Decisión *Random Forest* *Gradient Boosting* SVM	Modelos basados en árboles (*Random Forest* y Gradient Boosting) obtuvieron precisión perfecta del 100% en clasificación de severidad
5	ML-SPAs: Fortifying Healthcare Cybersecurity Leveraging Varied *Machine Learning* Approaches against Spear Phishing Attacks	2024	Datos internos o recopilaciones propias	Multilayer Perceptron (MLP) Convolutional Neural Network (CNN) Bidirectional LSTM (BiLSTM)	Las redes CNN y BiLSTM ofrecen capacidades casi perfectas de detección de spear phishing en entornos sanitarios.

Fuente: Elaboración propia.

2.1.1. Análisis de antecedentes

Himanshu Tiwari (2025). Advancing Vulnerability Classification with BERT: A Multi-Objective Learning Model.

Resumen: El presente artículo presentó una idea innovadora para clasificar de manera automática las vulnerabilidades de seguridad informática. La meta principal fue perfeccionar el procedimiento de análisis y priorización de vulnerabilidades reportadas en bases de datos públicas, como la *National Vulnerability Database* (NVD), empleando métodos de aprendizaje profundo fundamentados en modelos de lenguaje natural.

El trabajo desarrolló un modelo de clasificación con múltiples objetivos integrando tareas complementarias: la detección de la clase o tipo técnica de una vulnerabilidad (por ejemplo, *Privilege Escalation*, *Cross-Site Scripting*, *Denial of Service*, etc.) y la predicción del grado de gravedad de dicha vulnerabilidad (bajo, medio, alto o crítico). Los escritores utilizaron el modelo BERT (*Bidirectional Encoder Representations from Transformers*), que es un modelo de *Transformer* muy empleado en la manipulación del lenguaje natural (NLP), para conseguirlo.

El principal aporte de este enfoque fue que permitió llevar a cabo, al mismo tiempo, dos clasificaciones diferentes dentro de un único marco de aprendizaje; esto disminuyó la necesidad de procesos manuales para priorizar las vulnerabilidades y hacer triaje. Así, el modelo se propuso optimizar la eficacia de los analistas de ciberseguridad y asegurar valoraciones más rápidas y coherentes frente al incremento diario en la cantidad de reportes emitidos.

Según lo que se concluyó en la investigación, el modelo fundado en BERT logró un desempeño excepcional: una exactitud general de más del 94% para clasificar la gravedad y una tasa de aciertos del 92% para detectar categorías de vulnerabilidad. Estos hallazgos demostraron la capacidad del aprendizaje profundo para automatizar funciones esenciales en los procedimientos de gestión de vulnerabilidades y robustecimiento de la seguridad cibernética en las organizaciones.

Metodología: El estudio se centró en un método experimental y cuantitativo, que se organizó alrededor de la evaluación y el entrenamiento de un modelo de lenguaje pre-entrenado.

Conjunto de datos: Los autores emplearon datos del repositorio público de la *National Vulnerability Database* (NVD), en particular el archivo “nvdcve-1.1-recent. json”, que incluía los reportes más recientes hasta marzo de 2025. Después de un procedimiento de limpieza y filtrado, se lograron 5.637 registros de vulnerabilidades, que incluyen una explicación en texto para cada uno, su grado de severidad (de acuerdo con la métrica CVSS v3) y los identificadores

CWE correspondientes. Estos identificadores se clasificaron en diez tipos de vulnerabilidad, incluyendo *Buffer Overflow, SQL Injection, Cross-Site Scripting o Privilege Escalation.*

Antes de ser introducidas al modelo BERT, las descripciones textuales fueron sometidas a métodos de tokenización y normalización. Se utilizaron dos tipos de codificación diferentes para las etiquetas: una representación *multi-hot* para los tipos y una codificación discreta para la gravedad (valores de 0 a 3), lo que posibilitó que una misma vulnerabilidad pudiera formar parte de múltiples categorías.

Arquitectura del modelo: Se utilizó el modelo *base bert-base-uncased*, una versión estándar de BERT que cuenta con 12 capas y tiene 768 dimensiones internas. A este modelo se le incorporaron dos capas de salida (arquitectura de doble cabeza), una para cada tarea:

- Cabeza de severidad: salida de cuatro clases con activación *softmax* (clasificación única).
- Cabeza de tipo: salida de diez clases con activación *sigmoid* (clasificación múltiple).

Las dos cabezas intercambiaron las representaciones producidas por el cuerpo del modelo, lo que posibilitó adquirir de manera compartida aspectos importantes del texto. La función de pérdida total se calculó sumando dos elementos: Para clasificar tipos, se utiliza *Binary Cross-Entropy Loss*, y para clasificar la severidad, se emplea *Cross-Entropy Loss.*

Entrenamiento y evaluación: El conjunto de datos, garantizando la estratificación por niveles de severidad, fue segmentado en un 80% para el entrenamiento (4509 muestras) y en un 20% para la validación (1128 muestras). El optimizador AdamW se utilizó para entrenar el modelo durante tres épocas, con una tasa de aprendizaje de 2×10^{-5} y un tamaño de lote (*batch size*) de 16. A pesar de que la arquitectura hizo posible la ejecución en GPU, el tiempo total del entrenamiento fue de cerca de 15 minutos en CPU.

Se utilizaron métricas estándar de clasificación para la evaluación: *Accuracy, Precision, Recall y F1-Score* para la tarea de severidad; Exact Match, *Hamming Loss*, área bajo la curva ROC y AUC de *precission-recall* para el trabajo con múltiples etiquetas. Asimismo, los autores añadieron a los resultados visualizaciones como nubes de palabras de errores, matrices de confusión y mapas de coocurrencia, lo que posibilitó un análisis más interpretativo del rendimiento del modelo.

Resultados: De un conjunto de datos de 5637 entradas, con etiquetas de gravedad (Baja, Media, Alta, Crítica), la matriz muestra valores diagonales sólidos, con 217 verdaderos positivos para Bajo, 268 para Medio, 288 para Alto y 190 para Crítico, lo que refleja predicciones precisas para la mayoría de las muestras. Se observan errores de clasificación

menores, como 5 muestras de Bajo predichas como Alto y 6 muestras de Alto predichas como Crítico, lo que indica confusión ocasional entre niveles de severidad adyacentes, posiblemente debido a patrones lingüísticos similares en las descripciones.

Kai Liu, Yun Zhou, Qingyong Wang & Xianqiang Zhu (2019). Vulnerability Severity Prediction with Deep Neural Network.

Resumen: A través de redes neuronales profundas, Liu, Zhou, Wang y Zhu (2019) crearon un modelo para predecir la gravedad de las vulnerabilidades. El objetivo principal de la investigación fue optimizar la eficacia y exactitud del procedimiento de valoración de vulnerabilidades, que históricamente se lleva a cabo manualmente o utilizando modelos de aprendizaje clásico. Los autores sostuvieron que las explicaciones textuales presentes en bases de datos como la NVD podían ser utilizadas para deducir automáticamente el grado de gravedad de cada vulnerabilidad, disminuyendo así el trabajo de análisis humano.

El análisis se enfocó en las vulnerabilidades del tipo *Cross-Site Scripting (XSS)* y comparó diferentes modelos de aprendizaje profundo, incluyendo *LSTM*, *TextRCNN* y *CNN*, con el algoritmo *XGBoost* que sirvió como línea base. Los hallazgos revelaron que los modelos de *deep learning* lograron niveles de precisión más allá del 93%, lo cual superó al modelo tradicional. En concreto, el modelo *TextRCNN* fue reconocido como el más apropiado para la clasificación automática de vulnerabilidades según su gravedad, ya que tuvo un balance óptimo entre pérdida y precisión.

Metodología: El análisis se basó en un conjunto de datos que incluía vulnerabilidades del tipo XSS, obtenidas principalmente de la *National Vulnerability Database* (NVD), así como de otras fuentes públicas. Se llevó a cabo un proceso de preprocesamiento que abarcó la depuración del texto, la supresión de registros incompletos y la normalización lingüística. Después, las descripciones fueron tokenizadas y representadas a través de *embeddings* adquiridos con *Word2Vec*, lo que posibilitó la conversión del texto en vectores numéricos de significado semántico.

Cuatro modelos principales fueron entrenados y comparados por los autores:

- *XGBoost*, un modelo clásico que se basa en la ingeniería manual de características.
- *CNN (convolutional neural network)*, con el objetivo de detectar patrones locales en el texto.
- *LSTM (Long Short-Term Memory)*, centrada en las dependencias secuenciales.

- *TextRCNN (Recurrent Convolutional Neural Network),* modelo mixto que unió el aprendizaje a nivel local y contextual.

Se evaluaron modelos con métricas como el tiempo de entrenamiento, la pérdida y la precisión, después de que se entrenaron con conjuntos de entrenamiento de tamaños variables: 20%, 40%, 60%, 80% y 100%. El objetivo fue la investigación de la efectividad y escalabilidad de los procedimientos en función del incremento de datos.

***Resultados*:** Estos son los resultados para las dos tareas encargadas.

Clasificación de severidad: TextRCNN fue el modelo con mejor desempeño general en clasificación (tanto en *accuracy* como en *F1-Score*). Las redes neuronales (CNN, LSTM, TextRCNN) superaron a *XGBoost* ligeramente en *accuracy* y *F1-Score*, por lo que, se concluye que *XGBoost* tuvo el menor desempeño para clasificar severidades.

Predicción del puntaje CVSS (regresión): LSTM tuvo el menor error (mayor *accuracy*) al predecir puntajes CVSS numéricos. *XGBoost* fue más rápido de entrenar y ofreció un desempeño aceptable, aunque con mayor error. Podemos concluir lo siguiente:

- *TextRCNN:* mejor desempeño general en clasificación.
- LSTM tuvo el menor error (mayor precisión) al predecir puntajes CVSS numéricos.
- *XGBoost* fue más rápido de entrenar y ofreció un desempeño aceptable, aunque con mayor error.

Ashima Kukkar, Rajni Mohana, Anand Nayyar, Jeamin Kim, Byeong-Gwon Kang & Naveen Chilamkurti (2019). A Novel Deep-Learning-Based Bug Severity Classification Technique Using Convolutional Neural Networks and *Random Forest* with Boosting.

***Resumen**:* Kukkar, Mohana, Nayyar, Kim, Kang y Chilamkurti (2019) abordaron el problema de clasificar automáticamente la severidad de reportes de errores *(bug reports)* utilizando un modelo híbrido que combinaba el poder de las redes neuronales convolucionales con un clasificador de *Random Forest* potenciado *(boosting).* Observando que muchos modelos previos fallaban al capturar patrones relevantes en los textos de reportes —por depender demasiado de ingeniería manual de características como *bag-of-words* o frecuencias simples— los autores propusieron un enfoque llamado BCR *(Bug severity Classification using CNN and Random Forest with Boosting)* que automatizara la extracción de características latentes y representativas del lenguaje, mejorando precisión tanto para clasificación múltiple como binaria. El modelo procesaba el texto del reporte aplicando n-gramas y luego extraía

características con una CNN; finalmente clasifica la severidad usando *Random Forest* con *boosting.* Se validó con cinco proyectos de código abierto, y los resultados mostraron una mejora sustancial en comparación con técnicas anteriores.

***Metodología*:** La metodología del artículo constó de seis etapas, las cuales serán explicadas a continuación.

Recolección de datos: Se recopilaron reportes de errores de 5 proyectos *open-source*: Mozilla, Eclipse, NetBeans, OpenOffice y Jira. Cada error (*bug*) incluye un resumen y descripción textual más la etiqueta de severidad, la cual se asigna manualmente (por ejemplo: *blocker*, *major*, *minor*).

Preprocesamiento del texto: Esta etapa cuenta con tres pasos. Limpieza, Tokenización y Vectorización.

Limpieza: en este paso, ocurre la eliminación de los signos de puntuación, se omiten las palabras vacías y los caracteres especiales.

Tokenización: se divide el texto en palabras.

Vectorización: Se generaron representaciones de texto usando n-gramas y, opcionalmente, se aplican técnicas de *embeddings* o matrices de frecuencia.

Para procesar la representación numérica del texto, se utilizó una CNN (*Convolutional Neural Network*) la cual aprende patrones lingüísticos relevantes en los textos que ayudan a determinar la severidad.

La cuarta etapa es la clasificación con *Random Forest* y *Boosting*. Las características generadas por la CNN se pasan a un clasificador *Random Forest*. Luego, Se aplica un esquema de *Boosting* (no especificado si es *AdaBoost* o *Gradient Boosting*) para mejorar la precisión corrigiendo errores anteriores y como resultado se obtiene un mejor manejo de las clases desequilibradas y se aumenta la robustez/estabilidad del modelo.

En la quinta etapa, se realizó el entrenamiento y validación en el cual se entrenó al modelo BCR por separado en cada proyecto, se utilizaron técnicas de validación cruzada para asegurar la generalización y se evaluaron dos tareas: la clasificación binaria y la clasificación multiclase.

Finalmente, en la sexta etapa se evaluó el desempeño para lo cual se utilizaron las clásicas métricas de clasificación *Precision*, *Recall*, *F1-Score*, *Accuracy* y se compararon contra modelos clásicos: *Naive Bayes*, SVM y *Random Forest* tradicional.

***Resultados*:** Los resultados mostraron que el modelo BCR superó ampliamente los enfoques anteriores tanto en clasificación múltiple de severidad como en escenarios binarios (severo / no severo). En los cinco conjuntos probados, la precisión promedio del modelo estuvo entre aproximadamente 94% a 97%, y los valores de *F-measure* entre 93% a 96%. Por ejemplo, para el *dataset* de Mozilla con siete clases de severidad, aunque la clase *Blocker* tuvo desempeño menor (≈ 85% de precisión y ≈ 81% en *F-measure*), la mayoría de las otras clases consiguieron valores muy altos (por ejemplo, *Enhancement*, *Major*, *Minor*, *Trivial* tuvieron F-*measures* por encima del 95%).

Además, los autores reportaron que el modelo era relativamente eficiente en tiempo de predicción: una vez entrenado, clasificar un nuevo reporte bug les tomó milisegundos. Se observaron limitaciones en clases con pocos ejemplos, como la clase *Blocker*, lo que afectaba su capacidad predictiva. También se identificó como amenaza a la validez que se usaron solo datos no estructurados (título y descripción) y repositorios *open-source*; podría no generalizar igual para informes comerciales o con otras características de datos.

Fawaz Alanazi, Ahmed Badi Alshammari, Chams Sallami, Asma A.Alhashmi, Rachid Effghi, Anil Kumar & Abdulbasit Darem (2025). *Enhancing cybersecurity vulnerability detection using different Machine Learning severity prediction models*.

Resumen: Alanazi, Alshammari, Sallami, Alhashmi, Effghi, Kumar y Darem (2025) investigaron cómo distintos modelos de *Machine Learning* podrían predecir la severidad de vulnerabilidades y mejorar la detección dentro de catálogos conocidos de vulnerabilidades explotadas (como el catálogo CISA de vulnerabilidades usadas).

Su objetivo fue determinar qué modelos brindan un balance óptimo entre eficiencia computacional, exhaustividad, precisión y aptitud para gestionar situaciones críticas. Cinco modelos fueron evaluados: *Gradient Boosting*, Bosque Aleatorio (*Random Forest*), Regresión logística, Máquina de Vectores de Soporte (*Support Vector Machine*) y árbol de decisión. Según el estudio, los modelos fundamentados en árboles (en particular Árbol de Decisión, *Random Forest* y *Gradient Boosting*) lograron una exactitud perfecta (100%) al categorizar las vulnerabilidades por su gravedad, lo que les permitió sobrepasar a los otros modelos cuando se trató de vulnerabilidades críticas. Asimismo, los árboles demostraron ser mucho más eficaces desde el punto de vista computacional; en particular, el Árbol de Decisión es el que tiene mayor rapidez y mantiene una precisión alta, lo que permite su uso en tiempo real.

Metodología: Los autores utilizaron el catálogo *Known Exploited Vulnerabilities* de CISA, correspondiente a 2022, como fuente de datos. Este repositorio contiene vulnerabilidades que se han utilizado de manera activa. Con ese conjunto, crearon un *dataset* que ya tenía etiquetas de severidad establecidas.

Luego, los datos se procesaron para ser utilizados con diversos algoritmos de *Machine Learning*. Específicamente:

- Preprocesamiento: se refiere a la selección de atributos significativos, la depuración de datos, posiblemente la conversión de variables categóricas y el tratamiento de valores ausentes.
- Se emplearon cinco modelos diferentes: *Gradient Boosting*, SVM (*Support Vector Machine*), Árbol de decisión (*Decision Tree*), Bosque aleatorio (*Random Forest*) y regresión logística (*Logistic Regression*).
- Con el propósito de valorar la eficiencia, utilizaron métricas tales como precisión *(accuracy),* precisión más específica *(precision), recall* y también calcularon el tiempo computacional.

Los modelos fueron comparados en términos de su rendimiento general en las diferentes severidades presentes en el conjunto de datos, así como su habilidad para detectar vulnerabilidades de alta gravedad o "críticas".

Resultados: El estudio de Alanazi et al. (2025) evidenció que los modelos fundamentados en árboles de decisión superaron a las otras perspectivas analizadas. En concreto, los algoritmos *Random Forest, Gradient Boosting* y *Decision Tree* lograron una exactitud del 100% al clasificar las vulnerabilidades de acuerdo con su grado de severidad. Esto se logró al identificar con acierto los casos críticos y distinguir sin equivocaciones entre los diferentes niveles de gravedad. Estos hallazgos demostraron que los modelos de tipo árbol son más efectivos que los métodos lineales y de margen máximo para capturar la relación entre las propiedades de las vulnerabilidades y su posible impacto.

En contraste, los modelos de *Support Vector Machine* (SVM) y de regresión logística mostraron un rendimiento menos sobresaliente, sobre todo cuando se trataba de clasificar vulnerabilidades que fueron clasificadas como "críticas". En estas situaciones, exhibieron una sensibilidad (*recall*) más baja y una propensión a confundirlas con categorías de gravedad inferior, lo cual podría acarrear peligros en ambientes donde la priorización adecuada es crucial. Sin embargo, los dos modelos mantuvieron resultados razonables en vulnerabilidades de

gravedad media o baja, lo que demuestra cierta estabilidad, pero con restricciones a la hora de manejar situaciones más rigurosas.

Con respecto a la eficiencia computacional, el Árbol de Decisión sobresalió por su balance entre la rapidez y la exactitud. Dado que tuvo el menor tiempo de procesamiento, se presenta como una opción apropiada para sistemas de detección o priorización de vulnerabilidades en tiempo real. El *Gradient Boosting* y el *Random Forest*, a pesar de su lentitud debido a que son ensamblados, brindaron resultados más sólidos y generalizables. Se aconseja utilizarlos en análisis donde la precisión tenga mayor importancia que la velocidad.

Por último, los autores llegaron a la conclusión de que los modelos basados en árboles son una solución eficaz y fiable para prever la severidad de las vulnerabilidades, gracias a su capacidad interpretativa, exactitud y coste computacional reducido. No obstante, admitieron que la utilización de un conjunto de datos estático, el catálogo de vulnerabilidades explotadas por CISA era una limitación y sugirieron incorporar métodos de aprendizaje profundo y datos en tiempo real para abordar nuevas clases de amenazas y situaciones de explotación que cambian continuamente como línea futura de investigación.

Alanazi (2024). ML-SPAs: *Fortifying Healthcare Cybersecurity Leveraging Varied Machine Learning Approaches against Spear Phishing Attacks*.

Resumen: Alanazi (2024) estudió la manera de robustecer la ciberseguridad del sector salud ante ataques de *spear phishing (SPAs)* a través de diversos modelos de aprendizaje automático. La investigación se basó en la hipótesis de que las defensas convencionales, como los antivirus o los filtros, no son suficientes ante ataques complejos dirigidos a usuarios concretos. Para lograr esto, diseñó un sistema llamado ML-SPAs que incorporó una variedad de fuentes de datos (como el contenido de los correos, la conducta del remitente, los archivos adjuntos, los registros de correo y las historias clínicas) con el fin de capacitar modelos jerárquicos (*hierarchical*) y secuenciales (como BiLSTM y CNN). De acuerdo con el autor, los modelos jerárquicos obtuvieron una precisión de hasta el 99,99%; BiLSTM llegó a 99,94% y otros modelos más tradicionales como el MLP fueron capaces de obtener una exactitud del 98,46%. Estos resultados destacaron el potencial de las arquitecturas avanzadas para detectar *spear phishing* con alta efectividad en entornos de salud.

Metodología: Alanazi llevó a cabo una investigación en la cual la metodología consistió en desarrollar e implementar un modelo llamado ML-SPAs (*Machine Learning for Spear Phishing Attacks*), cuyo objetivo era robustecer la ciberseguridad del sector sanitario frente a

los peligros de *spear phishing*. El escritor reunió una extensa gama de información, que incluye datos de correos electrónicos institucionales, metadatos de archivos adjuntos, registros de comunicación y patrones históricos de los remitentes. Para desarrollar un conjunto de entrenamiento que sea representativo y equilibrado, estos datos se preprocesaron con el fin de eliminar el ruido, estandarizar los textos y extraer elementos significativos del contenido y del contexto del correo.

Después, se utilizaron varias técnicas de aprendizaje profundo y automático para evaluar su desempeño en la clasificación de correos maliciosos. Los modelos que fueron evaluados abarcaron las redes neuronales convolucionales (CNN), las de memoria a corto y largo plazo bidireccionales (BiLSTM), un perceptrón multicapa (MLP) y un modelo jerárquico que fue creado para gestionar la información a través de varios niveles: texto, comportamiento y archivos adjuntos. Esta estructura jerárquica posibilitó un análisis más exhaustivo de los correos, al captar tanto patrones lingüísticos como indicios contextuales asociados con la conducta del atacante.

Para asegurar que los resultados sean válidos, se evaluaron y entrenaron todos los modelos en las mismas condiciones experimentales, empleando indicadores de rendimiento como la tasa de falsos positivos y la precisión (*accuracy*). Asimismo, se realizaron pruebas cruzadas para comprobar la capacidad de generalización de los modelos frente a nuevos tipos de ataques. Para optimizar la eficacia del entrenamiento y prevenir el sobreajuste, el escritor modificó también los hiperparámetros esenciales, el número de épocas, la tasa de aprendizaje y el tamaño del lote.

Por último, Alanazi realizó un análisis de interpretabilidad para completar la metodología, con el objetivo de examinar cómo cada modelo fundamentaba sus predicciones. Esta etapa tuvo como objetivo garantizar que los expertos en ciberseguridad pudieran comprender las decisiones del sistema, promoviendo así la confianza en su utilización real dentro de las entidades sanitarias. En términos generales, la metodología del estudio integró rigor técnico con aplicabilidad práctica, alcanzando un balance entre la capacidad de predicción y la aplicabilidad del modelo.

Resultados: Los hallazgos del estudio mostraron que los modelos más complejos se destacaron en comparación con las perspectivas tradicionales. Específicamente, el modelo de arquitectura jerárquica más complejo logró una precisión del 99.99%, lo que la convirtió en la más eficiente. El modelo BiLSTM logró una precisión de 99.94%, casi igual de alta que la del otro modelo, lo que evidencia que los modelos secuenciales son igualmente muy competentes

para este tipo de trabajos. En cambio, el modelo MLP, que tiene un diseño más tradicional, logró una precisión de 98.46%, que, aunque es alta, es menor a la del modelo anterior.

Estos hallazgos indicaron que, al incorporar varias clases de atributos (metadatos, comportamiento y texto) y emplear arquitecturas que pueden procesar niveles jerárquicos o secuenciales, es posible alcanzar una detección de *spear phishing* con un nivel de confiabilidad muy elevado en el sector sanitario. Asimismo, el escritor destacó que la interpretabilidad de los modelos generados ofrece beneficios concretos para su implementación real: los grupos de seguridad son capaces de entender las decisiones del modelo y adaptarlas a nuevas estrategias de ataque que vayan surgiendo.

2.2. Bases Teóricas

En esta sección, se presentan los fundamentos teóricos que respaldan la investigación sobre la clasificación de la severidad de vulnerabilidades en el sector bancario a través del empleo de modelos de *Machine Learning*.

2.2.1. Inteligencia Artificial

La Inteligencia Artificial (IA) es una rama multidisciplinaria de la informática que se enfoca en el desarrollo de sistemas y programas informáticos que pueden realizar tareas que normalmente implican capacidades propias del ser humano. Según la definición proporcionada por Russell y Norvig (2009), la IA puede definirse como el diseño de algoritmos orientados a crear máquinas que imiten habilidades propias del ser humano.

El objetivo de la inteligencia artificial es replicar, en alguna medida, las aptitudes cognitivas del ser humano, tales como el razonamiento, el aprendizaje, la percepción, el análisis del lenguaje natural y la resolución de problemas. Con este fin, se desarrollan algoritmos, modelos matemáticos y sistemas computacionales que pueden decidir, adaptarse a contextos cambiantes y mejorar su rendimiento conforme acceden a nuevos datos y experiencias.

2.2.1.1. Aprendizaje automático.

El aprendizaje automático, o *Machine Learning*, constituye una rama de la inteligencia artificial que ha evolucionado de manera significativa en los últimos años. Según Mitchell (1997), *Machine Learning* se define como "el estudio de algoritmos y modelos estadísticos que los sistemas informáticos utilizan para mejorar su rendimiento en una tarea específica a través de la experiencia". Es decir, se basa en entrenar a una máquina para que, a partir del análisis de datos, pueda ejecutar tareas concretas sin necesidad de una programación explícita para cada una de ellas.

El aprendizaje automático se fundamenta en el desarrollo de modelos estadísticos y matemáticos capaces de identificar patrones y establecer correlaciones a partir de datos

previamente recopilados. Posteriormente a ello, estos modelos aplican su aprendizaje para predecir o tomar decisiones frente a datos desconocidos.

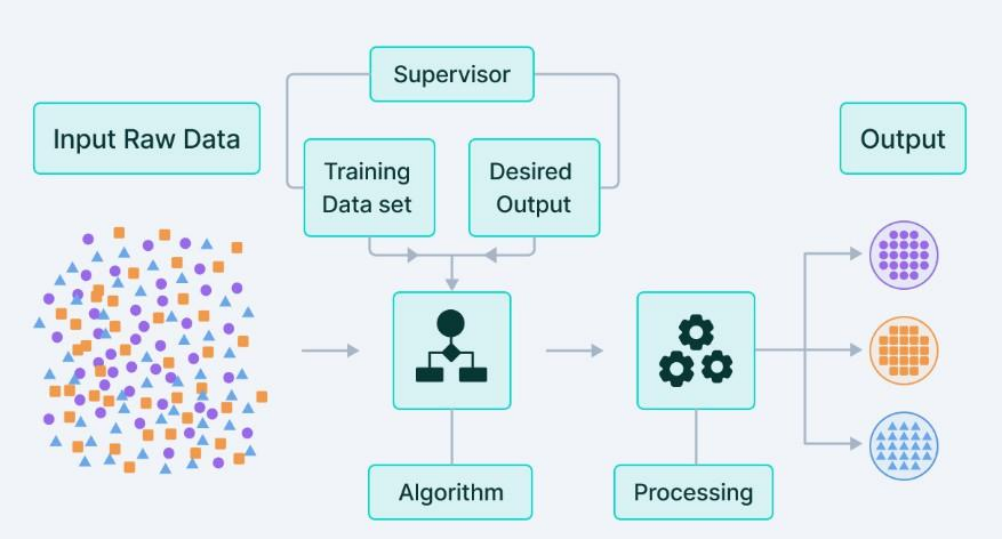
2.2.1.1.1. Aprendizaje supervisado.

Hiran et al. (2021) argumentan que el aprendizaje supervisado se basa en la obtención de conocimiento a partir de ejemplos. Se le denomina "supervisado" porque el modo en que un algoritmo aprende, a partir de un conjunto de datos de entrenamiento se asemeja a la orientación que proporciona un instructor durante el aprendizaje.

La Figura 4 ilustra el proceso de un aprendizaje supervisado, donde se emplean imágenes previamente categorizadas de gatos y perros para entrenar un modelo con la capacidad de diferenciarlos. Si el modelo aprende de manera efectiva, podrá clasificar con precisión nuevas imágenes de gatos y perros sin una etiqueta asignada.

Figura 4

Diseño de un aprendizaje supervisado



Fuente: DataCamp. (2023). Supervised vs. Unsupervised Learning: Differences & Examples. Recuperado de https://www.datacamp.com

Clasificación: La clasificación en el marco del aprendizaje supervisado involucra el proceso de agrupar la salida generada por un modelo en múltiples clases o categorías, las cuales se fundamentan en una o varias variables de entrada. Esta metodología se emplea cuando la variable de salida es de naturaleza discreta o categórica. Cuando un algoritmo de aprendizaje supervisado se centra en la clasificación de las variables de entrada en únicamente dos categorías, se le denomina clasificación binaria. Esto se puede ver en casos como el reconocimiento de si un correo electrónico es "*spam*" o "*no spam*", la determinación de si una

“enfermedad” está presente o ausente, la evaluación de respuestas como "Sí" o "No", o el etiquetado de valores como 0 o 1, entre otras aplicaciones.

Por otro lado, cuando el algoritmo se enfoca en clasificar las variables de entrada en más de dos categorías, se le denomina clasificación multiclase. Un caso ilustrativo de esto se encuentra en el reconocimiento de números escritos a mano, donde las clases pueden comprender un espectro que se extiende desde 0 hasta 9, cubriendo distintas categorías.

Algoritmos de clasificación

- ***XGBoost:*** *XGBoost* (*Extreme Gradient Boosting*) es un algoritmo de *Machine Learning* basado en árboles de decisión que implementa la técnica de boosting por gradiente de manera optimizada y escalable. Fue desarrollado por Tianqi Chen y ha sido ampliamente adoptado debido a su rendimiento competitivo, particularmente en problemas de clasificación y regresión que involucran grandes volúmenes de datos y alta dimensionalidad (Chen & Guestrin, 2016). Cada modelo se entrena de forma consecutiva y se centra en corregir los errores de los modelos previos. Esto ayuda a disminuir el sesgo y aumentar la exactitud del modelo final (Friedman, 2001).

  XGBoost introduce mejoras importantes sobre otros métodos de *boosting*, tales como *AdaBoost* o GBM estándar:

  - Regularización (L1 y L2) incorporada en la función de pérdida, lo que reduce el riesgo de sobreajuste (*overfitting)*.
  - Construcción paralela de árboles para mejorar la velocidad de entrenamiento.
  - Gestión automática de valores faltantes.
  - Uso eficiente de memoria mediante una estructura de datos llamada *DMatrix*.
  - Poda de árboles en profundidad para mejorar la precisión del modelo.

  Estas optimizaciones hacen que *XGBoost* ser más veloz, preciso y escalable, posicionándolo como uno de los algoritmos más empleados en competencias *de Machine Learning*, como las organizadas por Kaggle (Brownlee, 2020).

- ***Random Forest***

  El algoritmo de *Random Forest* es una técnica de *Machine Learning* que se basa en la construcción de múltiples árboles de decisión y en la integración de sus resultados para aumentar la precisión y la capacidad de generalización del modelo. Breiman (2001) detalla las siguientes fases para la elaboración de este algoritmo:

- **Conjunto de árboles de decisión**

  El bosque aleatorio es una técnica de aprendizaje que integra múltiples árboles de decisión con el objetivo de generar un modelo más sólido y exacto. Cada uno de los

árboles en el bosque aleatorio se entrena con un conjunto de datos diferente y emplea un subconjunto aleatorio de variables para crear divisiones en cada nodo.

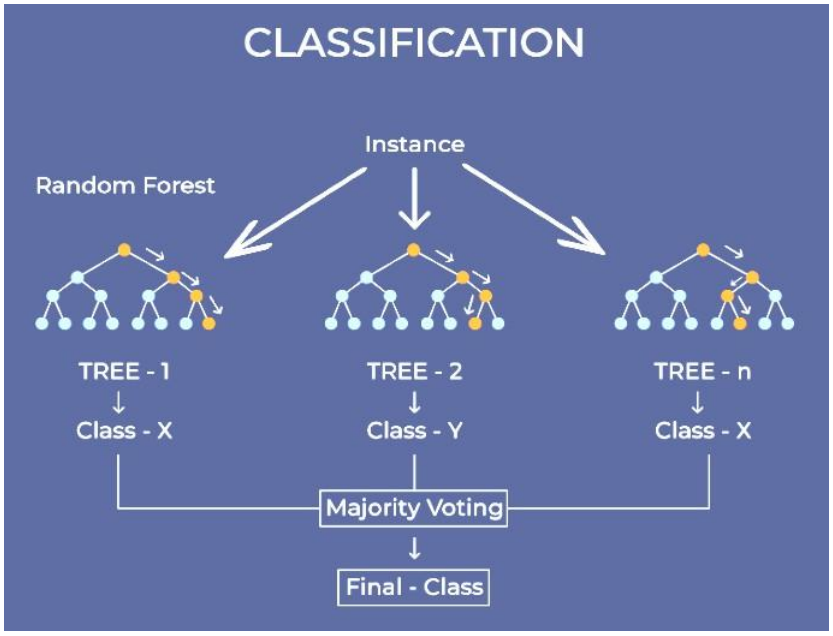
- Muestreo aleatorio: El bosque aleatorio emplea el método de *Bagging* (*Bootstrap Aggregating*) para generar subconjuntos de datos de entrenamiento a través del muestreo con reposición. Esto aporta diversidad en el proceso de formación y contribuye a reducir la varianza del modelo.
- Características aleatorias: En cada nodo de decisión de los árboles, solo se considera un subconjunto aleatorio de variables para llevar a cabo la división. Esto incrementa aún más la diversidad entre los árboles y reduce la correlación entre ellos.
- Votación o promedio: En el caso de problemas de clasificación, el bosque aleatorio integra las predicciones de todos los árboles de decisión y selecciona la etiqueta de clase predominante como la predicción final. Para problemas de regresión, se calcula el promedio las predicciones de todos los árboles a fin de conseguir la predicción final.
- *Error Out of Bag* (OOB): El bosque aleatorio utiliza el método de estimación de error OOB, que posibilita la evaluación el desempeño del modelo sin requerir un conjunto de validación independiente. El error OOB se calcula agregando las predicciones de los árboles en las muestras no seleccionadas en sus respectivas muestras de *Bootstrap*.

El algoritmo de bosque aleatorio proporciona mayor precisión y solidez en contraste con los árboles de decisión individuales al capitalizar el poder del aprendizaje conjunto y las técnicas aleatorias.

Learnbay (2023) ofrece una ilustración visual de cómo opera el algoritmo *Random Forest*, en la cual se observa cómo varios árboles de decisión colaboran entre sí para optimizar la exactitud de la clasificación. Cada árbol es entrenado con un subconjunto distinto de los datos y aprende patrones particulares de manera independiente. Estos árboles producen predicciones individuales durante la etapa de entrenamiento, las cuales se unen más tarde a través de un procedimiento de votación mayoritaria para llegar a una decisión final más sólida y fiable. El flujo general de procesamiento y decisión en el clasificador de bosque aleatorio se muestra en la Figura 5.

Figura 5

Representación del algoritmo Random Forest



Fuente: Adaptado de "*Random Forest* Algorithm in *Machine Learning* with Example", por Learnbay, 2023 (https://www.learnbay.co/blog/random-forest-algorithm-in-machine-learning-with-example).

2.2.1.1.2. Aprendizaje no supervisado.

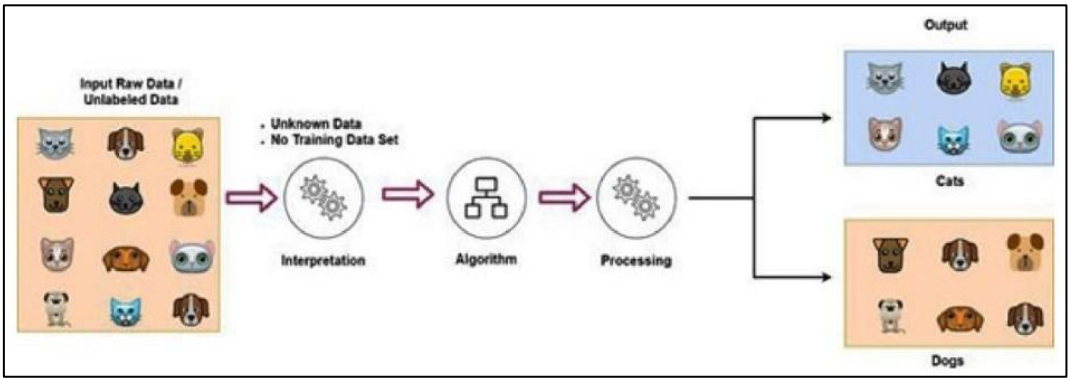
El aprendizaje no supervisado es una rama del aprendizaje automático (*Machine Learning*) que se distingue por trabajar con datos no etiquetados, es decir, aquellos en los que no se conoce la salida o clase deseada. El objetivo principal de este enfoque es descubrir estructuras ocultas, patrones o relaciones subyacentes en los datos sin una supervisión humana explícita (Hastie, Tibshirani & Friedman, 2009).

En contraste con el aprendizaje supervisado, en el cual los modelos se entrenan con datos que contienen una variable objetivo (etiqueta), el aprendizaje no supervisado busca organizar y representar la información con el fin de facilitar su análisis o procesamiento subsecuente.

En la Figura 6 se ilustra el esquema de un aprendizaje no supervisado a través de un conjunto de imágenes de gatos y perros, sin entrenamiento previo. El objetivo del algoritmo es identificar características al agrupar las imágenes en categorías en función de sus similitudes, sin la necesidad de supervisión externa.

Figura 6

Diseño de un aprendizaje no supervisado



Fuente: Hiran et al. (2021)

El aprendizaje no supervisado se clasifica en dos categorías principales de problemas: problemas de agrupamiento, donde se buscan patrones de similitud para agrupar datos, y problemas de asociación, que buscan hallar vínculos entre los datos. Esta técnica es clave en el ámbito del *Machine Learning* y se aplica en una gran variedad de aplicaciones.

En este estudio, emplearemos el enfoque de aprendizaje supervisado y exploraremos con mayor profundidad los conceptos y algoritmos asociados.

Clustering: El *clustering* o agrupación segmenta un conjunto de elementos (por ejemplo, objetos, o eventos presentados en un conjunto de datos estructurados) en segmentos (o agrupaciones naturales) cuyos componentes presentan características comunes. A diferencia de la clasificación, en la agrupación las clases o categorías no están predefinidas. Como el algoritmo seleccionado analiza el conjunto de datos, detectando las similitudes de las cosas basado en sus propiedades, se estructuran los clústeres. Una vez que se han establecido clústeres coherentes, pueden ser empleados para categorizar e interpretar nueva información. (Sharda, 2024)

De acuerdo con Font (2019), las técnicas de agrupación principales incluyen el *Clustering* particional, el *Clustering* jerárquico y el *Clustering* basado en densidad.

- ***Clustering* particional**

  K-Means. Basados en centroides caracterizado por las medias.

- ***Clustering* jerárquico**

  Aglomerativos y divisivos se basan en el concepto de distancia. Puntos con distancia próximas se conectan al mismo grupo, mientras que puntos con distancias altas van a grupos diferentes.

- ***Clustering* basado en densidad**

  DBSCAN y pdfCluster (no paramétrico) se basan en agrupar por zonas densas de puntos.

2.2.2. Métricas de evaluación de modelos

Según Navarro (2023), las métricas de evaluación son herramientas cuantitativas fundamentales que se emplean para medir la efectividad y precisión de los modelos de aprendizaje automático. Estas métricas proporcionan un contexto objetivo para analizar y contrastar el rendimiento de diferentes modelos durante las fases de entrenamiento y validación.

2.2.2.1. Métricas para clasificación.

Los criterios de evaluación de modelos de clasificación emplean una terminología estándar. Los verdaderos positivos (TP) son aquellos registros que el modelo pronostica acertadamente en su clase correspondiente. Los falsos positivos (FP) son aquellos casos de clase negativa clasificados erróneamente como casos positivos. Los falsos negativos (FN) son instancias positivas reales predichas erróneamente como negativas. Los negativos verdaderos (TN) son las instancias de clase negativas reales que el modelo clasifica con exactitud como negativas (IBM, 2024).

IBM (2024) define la precisión de un modelo de la siguiente manera: precisión, esto se denomina valor predictivo positivo (PPV). Es la proporción de predicciones de clase positivas que realmente pertenecen a la clase en cuestión. Por ejemplo, en un filtro de correo no deseado, la precisión es la proporción de mensajes que el modelo clasifica como correo no deseado que, de hecho, son correo no deseado. Se representa mediante la ecuación (párr. 15).

Fórmula 1

Fórmula de cálculo de precisión

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

Fuente: Adaptado de "W*hat are Classification Models*?", por IBM, 2024 (https://www.ibm.com/es-es/think/topics/classification-models), párr. 15.

IBM (2024) define el *recall* de un modelo de la siguiente manera: recuperación, también denominada sensibilidad o tasa de verdaderos positivos (TPR), la recuperación denota el porcentaje de instancias de clase detectadas por un modelo. Volviendo al filtro de spam, la recuperación indica cuántos mensajes de spam reales el modelo clasifica realmente como spam. Se representa mediante la ecuación (párr. 16).

Fórmula 2

Fórmula de cálculo de recall

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

Fuente: Adaptado de "W*hat are Classification Models*?", por IBM, 2024 (https://www.ibm.com/es-es/think/topics/classification-models), párr. 16

IBM (2024) define el *F1-Score* de un modelo de la siguiente manera: puntuación de F1, la precisión y la recuperación pueden compartir una relación inversa; a medida que un clasificador devuelve más verdaderos positivos (mayor recuperación), el clasificador inevitablemente clasifica mal también las no instancias (es decir, los falsos positivos), lo que disminuye la precisión. La puntuación de la F1 tiene como objetivo resolver esta disyuntiva. F1 (o puntuación F) combina eficazmente la precisión y la recuperación para representar la exactitud total de un modelo en cuanto a clases. Se representa mediante la ecuación.

Fórmula 3

Fórmula de cálculo de F1-Score

$$F = \frac{2P \times R}{P + R}$$

Fuente: Adaptado de "*What are Classification Models*?", por IBM, 2024 (https://www.ibm.com/es-es/think/topics/classification-models), párr. 17.

2.2.2.2. Métricas para clustering.

Existen varios criterios de evaluación para el análisis de clústeres y la determinación de la métrica apropiada se basa en el tipo de método de agrupamiento y del conjunto de datos correspondiente. Las métricas de evaluación se pueden clasificar generalmente en dos categorías fundamentales: extrínsecas e intrínsecas (IBM, 2024).

- **Medidas intrínsecas**

Las medidas de evaluación internas son métricas de evaluación para el análisis de clústeres que utilizan se basan exclusivamente en los datos disponibles. Pueden ser beneficiosas cuando se procesa información no categorizada. La fiabilidad del análisis se basa fundamentalmente en los vínculos entre las instancias. Se pueden emplear en ausencia de información previa o clases definidas. Entre los indicadores intrínsecos son *Within-Cluster Sum of Squares (WCSS), Silhouette Coefficient (S) y Davies–Bouldin Index* (DBI) (IBM, 2024).

Fórmula 4

Fórmula de cálculo de Within-Cluster Sum of Squares

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

Adaptado de *Scikit-Learn User Guide: Clustering performance evaluation*, por Scikit-Learn, 2024 (https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation), párr. 3

Fórmula 5

F*órmula de cálculo de Silhouette Coefficient (S)*

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$
$$S = \frac{1}{n} \sum_{i=1}^{n} s(i)$$

Adaptado de *Evaluating clustering performance using the Silhouette Coefficient*, por Scikit-Learn, 2024 (https://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient), párr. 1

Fórmula 6

Fórmula de cálculo de Davies–Bouldin Index (DBI)

$$DBI = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left(\frac{S_i + S_j}{M_{ij}} \right)$$

Adaptado de *Davies-Bouldin Index for Cluster Evaluation*, por Scikit-Learn, 2024 (https://scikit-learn.org/stable/modules/clustering.html#davies-bouldin-index), párr. 2

- **Medidas extrínsecas**

Las medidas de evaluación externas utilizan conocimiento real o externo para verificar la fiabilidad del desempeño del método de clusterización. Esto requiere algún tipo de datos etiquetados que validen la categoría o clúster al que pertenece cada punto de datos. En este caso, es posible contrastar la exactitud del análisis de *clustering* con las métricas que se emplean a menudo en la precisión de la categorización. Las medidas extrínsecas comunes incluyen la puntuación F, la pureza y el índice *Rand* (IBM, 2024).

Fórmula 7

Fórmula de Pureza

$$\text{Purity} = \frac{1}{N}\sum_{k=1}^{K}\max_{j} n_{k,j}$$

Adaptado de *Data Mining: Concepts and Techniques* (4th ed.), por Han, Kamber y Pei, 2022, Elsevier, p. 495.

Fórmula 8

Fórmula de cálculo de Davies–Bouldin Index (DBI)

$$F = \sum_{i=1}^{|C|} \frac{n_i}{N} \max_{j} \left(\frac{2 \times P(i,j) \times R(i,j)}{P(i,j) + R(i,j)} \right)$$

Adaptado de Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. (Capítulo 16: Clustering Evaluation)

Fórmula 9

Fórmula de cálculo de Rand Index

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

Adaptado de *Metrics for comparing cluster labels*, por Scikit-Learn, 2024 (https://scikit-learn.org/stable/modules/clustering.html#rand-index), párr. 1.

2.2.3. Sector Bancario

El sector bancario es uno de los pilares fundamentales del sistema financiero, cuya función principal es la intermediación financiera, es decir, la canalización de recursos desde los agentes económicos con excedentes hacia aquellos que requieren financiamiento. A través de instrumentos como depósitos, créditos, préstamos y servicios financieros, los bancos facilitan el flujo eficiente del capital (Mishkin & Eakins, 2018). El sector bancario puede analizarse desde distintas dimensiones:

2.2.3.1. Rol económico y funcional.

Los bancos desempeñan funciones esenciales que sostienen la actividad económica:

- Movilización del ahorro y su conversión en inversión.
- Facilitación de pagos y transacciones seguras.

- Gestión de riesgos financieros, mediante seguros, derivados y servicios especializados.
- Evaluación del riesgo crediticio para asignar capital eficientemente (Levine, 2005).

Estas funciones permiten una mejor asignación de recursos y contribuyen al crecimiento económico sostenible.

2.2.3.2. Regulación y Supervisión.

Debido a su importancia sistémica, el sector bancario está fuertemente regulado. Organismos como los bancos centrales y superintendencias financieras garantizan su estabilidad mediante normas de solvencia, liquidez y gestión de riesgos. A nivel internacional, los Acuerdos de Basilea (emitidos por el Comité de Supervisión Bancaria de Basilea) definen estándares clave:

- Basilea I, II y III establecen requerimientos mínimos de capital, estándares de control de riesgos y gestión del apalancamiento financiero (BIS, 2011).
- Estas regulaciones buscan evitar crisis sistémicas como la de 2008 y fortalecer la resiliencia del sistema financiero.

2.2.3.3. Importancia Social y Financiera.

El sector bancario también cumple una función social al promover la inclusión financiera, facilitar el acceso al crédito y apoyar programas de desarrollo. En países en desarrollo, los servicios bancarios son fundamentales para combatir la pobreza, fomentar el emprendimiento y apoyar a las pequeñas y medianas empresas (Demirgüç-Kunt et al., 2018).

2.2.4. Ciberseguridad en el Sector Bancario

2.2.4.1. Transformación Digital y Ciberseguridad.

La transformación digital ha revolucionado el sector bancario mediante tecnologías como la banca electrónica, aplicaciones móviles, *Big Data* e inteligencia artificial. Si bien estas innovaciones mejoran la eficiencia y el acceso, también han aumentado los riesgos cibernéticos.

Los ciberataques financieros como el *phishing*, *ransomware* o ataques a sistemas transaccionales, representan amenazas críticas. Por ello, el uso de *Machine Learning* y técnicas predictivas permite a las entidades bancarias identificar y clasificar vulnerabilidades para fortalecer la gestión de riesgos tecnológicos (Deloitte, 2021; Arora & Peddoju, 2022). En este contexto, la seguridad informática se ha convertido en una dimensión clave de la estabilidad financiera, y la clasificación automática de severidad de vulnerabilidades es una herramienta emergente que contribuye a la resiliencia cibernética bancaria.

2.2.4.2. Vulnerabilidades.

Las vulnerabilidades son las deficiencias o puntos débiles de un sistema que pueden ser explotadas por una amenaza (un atacante) con el objetivo de comprometer la seguridad o causar un perjuicio. (Sánchez-Bautista & Ramírez-Chávez, 2022) Múltiples amenazas son detectadas como riesgos para los activos de información, ya que aprovechan diversas clases de vulnerabilidades (Humpiri Flores et al., 2022).

2.2.4.3. Clasificación y priorización de vulnerabilidades.

La clasificación de severidad en vulnerabilidades es clave para la priorización eficaz de parches, especialmente en el sector bancario. El *Common Vulnerability Scoring System* (CVSS) proporciona una medida estándar de severidad técnica en una escala de 0 a 10, pero numerosos estudios han demostrado que un alto puntaje de CVSS no necesariamente indica explotación real, lo que genera un gran número de falsos positivos y sobrecarga operativa (Ozeren, 2025; Tenable, 2025).

Por ello, métricas complementarias como el *Exploit Prediction Scoring System* (EPSS) han surgido para estimar la probabilidad de explotación en el corto plazo, integrando señales de inteligencia de amenazas y características técnicas de las vulnerabilidades, y actualizándose diariamente para reflejar el panorama real de ataques (FIRST, 2025; Shimizu & Hashimoto, 2025).

Además, se ha constatado que EPSS supera al CVSS en eficiencia, logrando priorizar un mayor número de vulnerabilidades explotadas con menos esfuerzo (Arora, 2025; FIRST, 2023).

Para el sector bancario, donde los sistemas críticos y los impactos financieros son altamente sensibles, combinar CVSS con EPSS y otros modelos basados en contexto (como SSVC o CISA KEV) es fundamental para mejorar la precisión en la toma de decisiones y justificar las acciones en entornos regulados (Pentest-Tools.com, 2025).

2.3. Hipótesis

La formulación de la hipótesis general y de las hipótesis específicas se efectuó a partir de los problemas y objetivos definidos en las etapas previas del estudio.

2.3.1. Hipótesis general

La aplicación de modelos de *Machine Learning* mejora la eficiencia en la priorización de vulnerabilidades en el sector bancario mediante la predicción de severidad y la segmentación técnica de los activos afectados.

2.3.2. Hipótesis específicas

HE1: La identificación y selección de variables técnicas relevantes permitirá construir *datasets* adecuados para el entrenamiento de modelos de *Machine Learning*, contribuyendo a una priorización más precisa e inteligente de vulnerabilidades en el sector bancario.

HE2: La implementación de técnicas de preprocesamiento de datos, tales como imputación de valores faltantes, eliminación de duplicados y tratamiento de valores atípicos, influirá positivamente en la calidad y limpieza de los *datasets* de vulnerabilidades, mejorando la confiabilidad de los resultados obtenidos por los modelos de *Machine Learning*.

HE3: La aplicación de técnicas de aprendizaje automático supervisadas y no supervisadas influirá positivamente en la precisión de la predicción de severidad y en la segmentación de vulnerabilidades por familias técnicas, contribuyendo a una gestión más eficiente del riesgo tecnológico en el sector bancario.

HE4: El uso de métricas adecuadas influirá positivamente en la valoración de la eficiencia y robustez de los modelos de *Machine Learning* aplicados a la gestión y priorización de vulnerabilidades.

CAPÍTULO III: Entorno Empresarial

3.1. Descripción de la empresa

La empresa analizada en la presente investigación forma parte del sector bancario peruano, un sector en continua evolución debido a la digitalización de los servicios financieros. Esta entidad proporciona productos que incluyen cuentas de ahorro, préstamos y plataformas digitales, destinados a clientes individuales y corporativos.

La entidad maneja grandes volúmenes de información sensible y se apoya en sistemas tecnológicos que necesitan permanecer seguros y disponibles constantemente, debido a la naturaleza de sus operaciones. Por motivos de confidencialidad de la información, en este estudio no se muestra el nombre de la empresa en análisis. En esta línea, la seguridad de la información es un elemento esencial, la cual busca asegurar la confianza de los usuarios.

En este escenario, el estudio actual se centra en examinar y priorizar las vulnerabilidades dentro de la infraestructura tecnológica de la organización, con la finalidad de sugerir mejoras que ayuden a una gestión del riesgo más eficaz y a salvaguardar los activos digitales.

3.1.1. Reseña histórica y actividad económica

3.1.1.1. Reseña histórica.

La compañía estudiada es parte de un conglomerado financiero peruano que se fundó a mediados de 1990, durante un proceso de fortalecimiento del sistema financiero del país. Su origen se relaciona con la fusión de varias entidades dedicadas a la banca, seguros y administración de inversiones, con el objetivo de consolidar su presencia en el mercado y mejorar la gestión de los servicios financieros. Desde su establecimiento, la entidad se ha enfocado en el desarrollo sustentable, la estabilidad económica y la innovación tecnológica, lo que le ha permitido ser un referente en el sector financiero nacional.

En sus etapas iniciales de desarrollo, la organización enfocó sus esfuerzos en ampliar y diversificar sus líneas comerciales. Su cobertura, tanto a nivel regional como nacional, se expandió mediante la compra de entidades expertas en microfinanzas, seguros y banca de inversión. Durante este proceso de expansión, se pusieron en marcha nuevas tecnologías y estrategias digitales que posibilitaron la modernización de la gestión operativa, el aumento de la seguridad informativa y la provisión de servicios más accesibles y eficaces para los consumidores.

Hoy en día, la compañía se ha establecido como uno de los principales grupos financieros del Perú, con operaciones en varios países latinoamericanos. Su estructura

empresarial incluye varias divisiones de negocio, como la gestión de activos, los seguros, la banca universal y las microfinanzas. La organización, por medio de estas divisiones, tiene como objetivo fomentar la sostenibilidad, la innovación y la inclusión financiera en sus operaciones.

3.1.1.2. Actividad económica.

La compañía analizada opera en el sector financiero y proporciona una extensa variedad de productos y servicios para clientes individuales y corporativos. La intermediación financiera es su función principal, que incluye captar depósitos, otorgar créditos y administrar inversiones. Asimismo, ofrece servicios adicionales vinculados con seguros, microfinanzas y gestión de patrimonios, lo cual le posibilita cubrir varias exigencias del mercado y reforzar su posición competitiva en el sistema financiero del país.

La entidad ha dirigido su modelo de negocio a la innovación en tecnología y a una mayor eficiencia operacional, mediante la inclusión de soluciones digitales que optimizan la experiencia del usuario y aumentan la seguridad de las transacciones. Así, fomenta la inclusión financiera, estimula el progreso económico del país y ayuda a que el sector bancario crezca de manera sustentable.

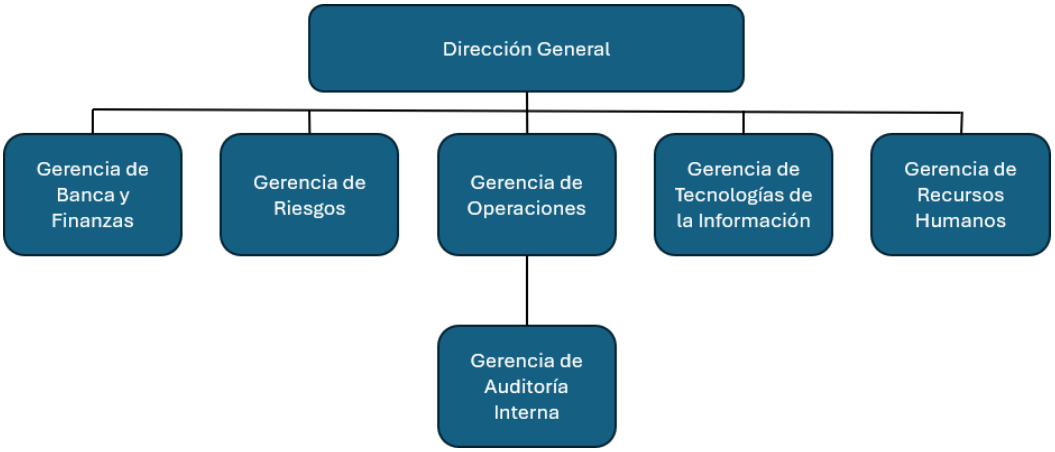
3.1.2. Descripción de la organización

3.1.2.1. Organigrama.

La organización referencial de la empresa analizada se muestra en la Figura 7. Este modelo muestra la conexión entre las diferentes etapas de gestión y las unidades funcionales que participan en la operación y administración de los servicios financieros.

Figura 7

Organigrama referencial de la empresa



Fuente: Elaboración propia.

3.1.2.2. Cadena de suministros.

La cadena de suministros de la empresa se compone de la siguiente manera:

- **Infraestructura física:** La organización tiene a su disposición una red de cajeros automáticos, sucursales y otras instalaciones que apoyan las operaciones administrativas y comerciales.
- **Tecnología y sistemas informáticos:** La entidad cuenta con plataformas tecnológicas que respaldan la administración de cuentas, el procesamiento de transacciones y la comunicación segura, elementos fundamentales en su operación financiera.
- **Recursos humanos:** La compañía cuenta con un equipo de expertos, cuyo desarrollo constante y compromiso son fundamentales para asegurar la eficacia operacional y una atención al cliente de alta calidad.
- **Servicios y productos financieros:** La institución brinda una extensa variedad de servicios y productos, entre los que sobresalen la colocación de créditos, la captación de depósitos y el diseño de soluciones financieras novedosas acordes a las demandas del mercado.
- **Flujos de información:** La organización maneja de forma sistemática la información proveniente de dentro y fuera de ella, incluyendo datos de clientes, informes estratégicos y comunicaciones institucionales usados en los procesos de toma de decisiones.
- **Flujo de fondos:** La compañía gestiona el flujo de recursos financieros, lo que incluye desde la obtención de fondos y las inversiones hasta el abono de créditos y otros servicios brindados por el banco.

3.1.3. Datos generales estratégicos de la empresa

3.1.3.1. Visión, misión y valores o principios.

A continuación, se presentan la visión, misión y principales valores de la empresa:

- ***Visión:*** Conformamos una entidad directiva en la región latinoamericana, cuya gestión impacta significativamente en la sociedad. La misión esencial de nuestra organización es incrementar el bienestar de los individuos mediante la provisión de instrumentos financieros de alta innovación y eficiencia.
- ***Misión:*** Fomentar el progreso social a través de la implementación acelerada de las reformas que resultan indispensables en las naciones donde se despliega nuestra actividad.
- ***Valores:*** Entre los valores la compañía cuenta con los siguientes:

- ✓ **Respeto:** El compromiso fundamental con el desarrollo humano integral se manifiesta al ejecutar un rol de impulsores de la transformación en los ecosistemas locales donde se centra nuestra operatividad.
- ✓ **Sostenibilidad:** Dado que el recurso humano constituye el eje central de toda nuestra actividad, analizamos meticulosamente sus requerimientos con el fin de salvaguardar el desarrollo integral en los ámbitos social, financiero y ecológico, tanto en el contexto actual como en las proyecciones a largo plazo.
- ✓ **Equidad:** Procedemos con imparcialidad y ecuanimidad al abordar la interacción y la valoración del capital humano. Nuestra prioridad constante es garantizar la paridad en las atribuciones, los deberes y las posibilidades de desarrollo profesional para la totalidad de los individuos que integran la organización.
- ✓ **Honestidad:** Fomentamos la claridad en todos los procesos y procuramos que nuestra operatividad sea plenamente coherente con los principios que sostenemos y las declaraciones que emitimos. Tenemos la convicción de que esta consistencia es el único medio para mantener y acrecentar vínculos de credibilidad con nuestros *stakeholders*.

3.1.3.2. Objetivos estratégicos.

Los objetivos generales y estratégicos que la entidad financiera en cuestión ha establecido están dirigidos a afianzar su posición en el ámbito bancario y a consolidar una administración segura y eficaz de sus operaciones, conforme con sus políticas institucionales y de desarrollo sostenible.

Objetivos generales

a) Fomentar la innovación y el avance tecnológico en los procedimientos de ciberseguridad y financieros, incentivando la digitalización y el manejo responsable de la información.
b) Consolidar la administración integral del riesgo operacional y tecnológico, garantizando la preservación de los activos críticos de información y la continuidad de los servicios.
c) Promover la eficacia institucional y la sostenibilidad a través de la optimización de los recursos, el perfeccionamiento constante y la implementación de prácticas que se ajusten a las normas del sistema financiero nacional e internacional.

Objetivos específicos

a) Garantizar la seguridad y confidencialidad de la información financiera, mediante la implementación de políticas efectivas de ciberseguridad y controles tecnológicos.

b) Mejorar los procedimientos de identificación y priorización de vulnerabilidades, incorporando modelos predictivos y herramientas analíticas que faciliten una administración del riesgo más proactiva.
c) Fomentar la formación permanente de los empleados en lo que respecta a la seguridad digital y al uso responsable de las tecnologías de información.
d) Asegurar la disponibilidad, resiliencia y cumplimiento con las regulaciones actuales para reforzar la infraestructura tecnológica de la institución.
e) Promover la innovación y la cooperación entre el sector público y el privado para preservar la capacidad competitiva y sostenible de la institución en un ambiente financiero dinámico y digital.

3.1.3.3. Evaluación interna y externa. FODA cuantitativo.

En la matriz EFE (ver Tabla 2) se ha calculado un puntaje de 2.69 que indica que la estrategia actual de la empresa le permite capitalizar con éxito las principales oportunidades (como la digitalización y el acceso a nuevos segmentos) y minimizar el impacto de las amenazas (como la competencia Fintech y la volatilidad regulatoria) de su industria.

Tabla 2

Matriz de Factores Externos (EFE)

		Factor	Peso	Valor	Pond.
Oportunidades	O1	Aceleración de la Transformación Digital y Financiera (Fintech)	0.15	4	0.6
	O2	Expansión del Acceso al Crédito en Segmentos No Bancarizados	0.12	4	0.48
	O3	Estabilidad Macroeconómica Post-Crisis y Crecimiento del PBI	0.1	3	0.3
	O4	Regulación Favorable para la Inversión Sostenible y ESG	0.08	3	0.24
	O5	Consolidación del Mercado a través de Adquisiciones Estratégicas	0.07	4	0.28
Amenazas	A1	Incremento de la Regulación y Fiscalización Anticíclica	0.14	2	0.28
	A2	Competencia Agresiva de Nuevos Entrantes (*Neobancos* y *Fintech*)	0.11	1	0.11
	A3	Volatilidad de las Monedas Locales y Riesgo Político Regional	0.09	2	0.18
	A4	Ataques Cibernéticos Sofisticados y Fuga de Datos Masiva	0.08	2	0.16
	A5	Disrupción Tecnológica de Big Tech Globales en Servicios Financieros	0.06	1	0.06
	Total		1		2.69

Fuente: Elaboración propia

El puntaje ponderado de 3.01 que se ha obtenido en la matriz EFI (ver Tabla 3) señala que la entidad está gestionando y aprovechando sus fortalezas internas de manera muy efectiva, y que el impacto combinado de sus fortalezas es significativamente mayor que el impacto de

sus debilidades. Las capacidades centrales, el capital y el liderazgo de mercado compensan con creces las ineficiencias internas.

Tabla 3

Matriz de Factores Internos (EFI)

		Factor	Peso	Valor	Pond.
Fortalezas	F1	Liderazgo de Mercado y Dominio de Nichos Clave	0.21	4	0.84
	F2	Sólida Posición Financiera y Alto Nivel de Capitalización	0.17	4	0.68
	F3	Diversificación de Servicios y Flujos de Ingreso (Banca, Seguros, Microfinanzas)	0.12	4	0.48
	F4	Marca Fuerte y Alta Confianza del Consumidor en la Región	0.1	3	0.3
	F5	Infraestructura Digital Avanzada y Capacidad de Desarrollo *In-house*	0.05	3	0.15
Debilidades	D1	Rigidez Estructural y Burocracia en la Toma de Decisiones	0.08	1	0.08
	D2	Alta Dependencia de un Mercado Primario Específico	0.08	2	0.16
	D3	Ciberseguridad como Alto Costo Operacional	0.07	2	0.14
	D4	Resistencia Cultural Interna a la Adopción Plena de Agilidad	0.06	1	0.06
	D5	Costo Elevado en el Mantenimiento de Sucursales Físicas	0.06	2	0.12
	Total		1		3.01

Fuente: Elaboración propia.

Ante los puntajes resultantes en las matrices EFE y EFI, la empresa en cuestión se encuentra en el cuadrante IV de la matriz IE (ver Figura 8). Eso quiere decir que se recomienda crecer y construir.

Figura 8

Matriz Interna – Externa (IE)

Fuente: Elaboración propia

Para prolongar la dinámica de crecimiento alcanzada, las acciones prioritarias se detallan en la Figura 9 (FODA Cuantitativa). Este análisis enfatiza la viabilidad de la expansión en mercados foráneos (Estrategias FO), la cual debe materializarse capitalizando la consolidada reputación institucional y la vasta experiencia operativa de la entidad. De igual forma, se subraya la imperiosa necesidad de establecer alianzas estratégicas (Estrategias FA) que permitan neutralizar los riesgos asociados a contingencias de salubridad, garantizando así la continuidad del aprovisionamiento de recursos indispensables para el sostenimiento de la expansión corporativa.

Figura 9

Matriz FODA Cuantitativa

FODA		F	Fortalezas	D	Debilidades
		1	Liderazgo de Mercado y Dominio de Nichos Clave	1	Rigidez Estructural y Burocracia en la Toma de Decisiones
		2	Sólida Posición Financiera y Alto Nivel de Capitalización	2	Alta Dependencia de un Mercado Primario Específico
		3	Diversificación de Servicios y Flujos de Ingreso (Banca, Seguros, Microfinanzas)	3	Ciberseguridad como Alto Costo Operacional
		4	Marca Fuerte y Alta Confianza del Consumidor en la Región	4	Resistencia Cultural Interna a la Adopción Plena de Agilidad
		5	Infraestructura Digital Avanzada y Capacidad de Desarrollo *In-house*	5	Costo Elevado en el Mantenimiento de Sucursales Físicas
O	**Oportunidades**		**Estrategia FO**		**Estrategia DO**
1	Aceleración de la Transformación Digital y Financiera (Fintech)	1,5	Expandir la oferta de productos y servicios Fintech aprovechando el liderazgo de mercado y la infraestructura digital avanzada.	1,5	Simplificar procesos burocráticos y automatizar funciones con la infraestructura digital existente para reducir costos de mantenimiento de sucursales.
2	Expansión del Acceso al Crédito en Segmentos No Bancarizados	1,4	Diseñar y lanzar productos de microcrédito digital bajo el paraguas de la marca fuerte para alcanzar segmentos no bancarizados.	2,5	Desarrollar modelos de negocio ágiles y con bajo costo de infraestructura física para atender nuevos segmentos, reduciendo la dependencia de mercados existentes.
3	Estabilidad Macroeconómica Post-Crisis y Crecimiento del PBI	2,3	Realizar inversiones estratégicas en proyectos de infraestructura y corporativos, aprovechando la sólida posición financiera y la diversificación de flujos.	4,3	Implementar programas de capacitación en agilidad organizacional y ciberseguridad para aprovechar la estabilidad y proteger la inversión en crecimiento.
4	Regulación Favorable para la Inversión Sostenible y ESG	2,4	Emitir bonos verdes y crear líneas de financiación sostenible, capitalizando la solidez financiera y la reputación de la marca para atraer inversionistas ESG.	1,4	Agilizar los procesos internos para la creación de nuevos productos ESG, superando la resistencia cultural y burocrática.
5	Consolidación del Mercado a través de Adquisiciones Estratégicas	1,2	Ejecutar adquisiciones estratégicas de *Fintech* o competidores menores para consolidar el liderazgo y la capacidad tecnológica, utilizando la solidez financiera.	2,1	Evaluar adquisiciones que permitan diversificar la dependencia del mercado primario y reducir la rigidez organizacional a través de la integración de nuevas estructuras.
A	**Amenazas**		**Estrategia FA**		**Estrategia DA**
1	Incremento de la Regulación y Fiscalización Anticíclica	2,3	Destinar recursos financieros y equipos especializados para asegurar el cumplimiento regulatorio en todas las líneas de negocio	1,5	Crear equipos de trabajo interfuncionales y ágiles para responder rápidamente a los cambios regulatorios, evitando la burocracia.
2	Competencia Agresiva de Nuevos Entrantes (*Neobancos* y *Fintech*)	1,5	Utilizar el liderazgo de mercado y la infraestructura digital para lanzar productos competitivos y *challenger-banks* propios que rivalicen con los nuevos entrantes.	2,5	Reducir los costos fijos asociados a sucursales físicas y diversificar la presencia en otros mercados para mitigar el impacto de la competencia en el mercado primario.
3	Volatilidad de las Monedas Locales y Riesgo Político Regional	2,3	Implementar estrategias avanzadas de cobertura de riesgo cambiario y diversificar la inversión en mercados menos volátiles con la sólida posición financiera.	1,2	Diseñar planes de contingencia para la toma de decisiones rápidas ante riesgos políticos, minimizando la dependencia de un solo mercado.
4	Ataques Cibernéticos Sofisticados y Fuga de Datos Masiva	5,2	Invertir continuamente en ciberseguridad de última generación y en la capacitación del personal, utilizando la infraestructura digital avanzada y la solidez financiera.	3,4	Implementar programas de seguridad de la información con cultura *DevSecOps* para fortalecer la ciberseguridad y superar la resistencia interna.
5	Disrupción Tecnológica de Big Tech Globales en Servicios Financieros	1,5	Establecer alianzas estratégicas con *Big Tech* globales o desarrollar capacidades propias para competir en plataformas y servicios financieros disruptivos, apoyándose en el liderazgo.	2,1	Fomentar la experimentación y la agilidad para probar nuevos modelos de negocio, reduciendo la burocracia y la dependencia del mercado primario ante la disrupción.

Fuente: Elaboración propia.

3.2. Modelo de negocio actual (CANVAS)

En el modelo de negocio actual se destaca la variedad de grupos de clientes que se atienden, incluyendo a individuos, empresas y usuarios digitales, además de la oferta de valor fundamentada en la confianza, la seguridad y el acceso a los servicios. Asimismo, se reconocen fuentes de ingresos provenientes, sobre todo, de créditos, inversiones y comisiones.

También se identificaron canales virtuales y presenciales que permiten la interacción con los clientes. El modelo incluye, además, actividades y recursos importantes relacionados con la gestión tecnológica, la ciberseguridad y la innovación, los cuales son respaldados por socios estratégicos y una estructura de costos centrada en mantener la sostenibilidad operacional. En la Figura 10 se muestra el detalle a través del modelo *Canvas*.

Figura 10

Matriz Canvas Empresa Bancaria

Fuente: Elaboración propia.

3.3. Mapa de proceso actual

El modelo operativo de la entidad se estructura en torno a tres categorías fundamentales que garantizan el flujo de valor desde la identificación de las necesidades del cliente hasta su plena satisfacción, tal como se ilustra en la Figura 11.

Los procesos estratégicos definen la dirección superior y el marco de acción de la corporación. Entre ellos se encuentran la Planificación y Gestión Corporativa, la Gestión de

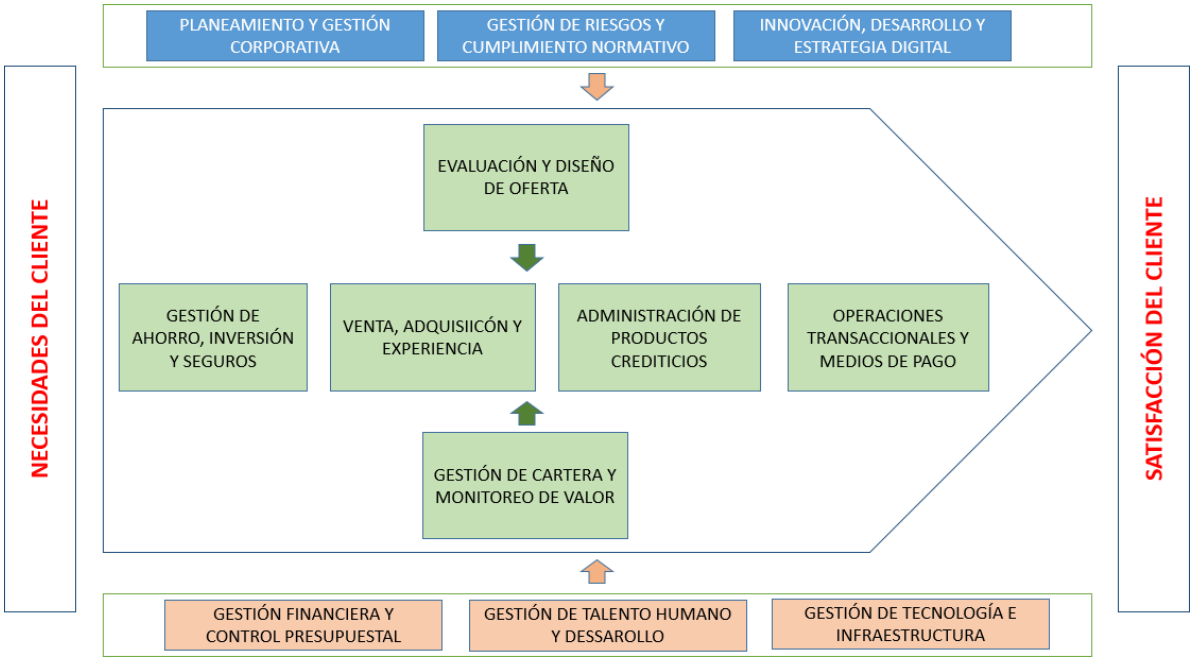
Riesgos y Cumplimiento Normativo, y la Innovación y Estrategia Digital. Estos son pilares indispensables para mantener la solidez financiera y la ventaja competitiva en el sector, asegurando la adaptación tecnológica continua y la fidelidad de los consumidores.

Los procesos clave comprenden la cadena de valor central que transforma las necesidades del cliente en servicios financieros. Este flujo se inicia con la Evaluación y Diseño de Oferta, que traduce la estrategia en productos concretos. Las actividades principales incluyen la Venta, Adquisición y Experiencia del cliente, la Administración de Productos Crediticios, y la Gestión de Ahorro, Inversión y Seguros. El proceso culmina en las Operaciones Transaccionales y Medios de Pago, y es retroalimentado por la Gestión de Cartera y Monitoreo de Valor, la cual mide el rendimiento y la satisfacción del servicio provisto.

Finalmente, la ejecución eficiente de los procesos clave es posible gracias a los Procesos de Soporte que proveen los recursos necesarios. Entre estos destacan la Gestión Financiera y Control Presupuestal, la Gestión de Talento Humano y Desarrollo (crucial para la cultura de agilidad y la capacitación), y la Gestión de Tecnología e Infraestructura, fundamental para la ciberseguridad y la plataforma digital de la organización.

Figura 11

Mapa de proceso actual



Fuente: Elaboración propia.

CAPÍTULO IV: Metodología de la investigación

4.1. Diseño de la investigación

4.1.1. Diseño

Tamayo (2003) afirma que el diseño experimental se distingue por permitir que el investigador manipule intencionadamente una o varias variables independientes con el objetivo de analizar su impacto sobre una variable dependiente, a la vez que controla elementos externos que podrían tener un efecto en los resultados. Teniendo en cuenta ello, el diseño de esta investigación fue experimental, dado que se examinaron las variables independientes y se estudió cómo influyen sobre las variables dependientes, así como la relación existente entre ambas. Con ese objetivo, se llevó a cabo distintos experimentos en los que se manipularán las variables. Se esperó obtener una variedad de resultados que luego fueron comparados entre sí para seleccionar el modelo de clasificación apropiado, el cual esté alineado con la investigación y ofrezca resultados ideales.

4.1.2. Enfoque

El enfoque cuantitativo, de acuerdo con Hernández, Fernández y Baptista (2014), se distingue por la aplicación de la medición numérica y el análisis estadístico para determinar patrones conductuales y confirmar hipótesis. El investigador procura interpretar los hechos observados mediante la objetividad, el control y la exactitud estadística, según esta perspectiva. En dicho sentido, se llevó a cabo la presente investigación con un enfoque cuantitativo, pues su base fue el examen de datos numéricos obtenidos de vulnerabilidades registradas entre 2020 y 2024. Con este método, se pretendió hallar patrones, relaciones y grados de severidad por medio de métricas estadísticas y de rendimiento de modelos de aprendizaje automático.

4.1.3. Tipo

De acuerdo con Sampieri (2014), el objetivo fundamental de la investigación explicativa es determinar las causas o factores que dan lugar a un fenómeno específico, no solamente describiéndolo, sino también entendiendo por qué sucede y en qué circunstancias se presenta. Este tipo de estudio busca establecer relaciones causales entre las variables analizadas. Por lo cual, la investigación actual es de carácter explicativo, porque se intenta evaluar y cotejar el desempeño de varios algoritmos en la predicción del grado de severidad de vulnerabilidades cibernéticas. Para esto, se emplearon modelos creados con técnicas de *Machine Learning* y se tomaron como base los datos obtenidos de informes registrados entre 2020 y 2024. Así, se logró

determinar las variables de mayor impacto en la clasificación de vulnerabilidades y describir cómo se comportaban los modelos ante diferentes contextos de evaluación.

4.2. Población y muestra

Esta investigación empleó variables independientes dentro de su esquema experimental para predecir la variable dependiente, utilizando para ello un enfoque de aprendizaje automático supervisado.

Tabla 4

Dataset de población y muestra

Población	Vulnerabilidades de ciberseguridad del área de infraestructura (PCs, Servidores, Laptops) en una empresa del sector de Banca detectadas en el periodo 2020 al 2025.
Muestra	13508 registros en *dataset* de vulnerabilidades de ciberseguridad del área de infraestructura de una empresa del sector de Banca detectadas del periodo 2020 al 2024. 4314 registros en *dataset* de vulnerabilidades de ciberseguridad del área de infraestructura de una empresa del sector de Banca detectadas del periodo enero a setiembre del 2025.

Fuente: Elaboración propia.

4.3. Metodología de implementación de la solución

La metodología que se utilizará para la implementación de la solución es:

Figura 12

Metodología utilizada en este proyecto

Etapa 1 - Recolección: En esta etapa se identificarán los datos relevantes para la ejecución del modelo, lo cual incluye información sobre nombre de vulnerabilidad, Sistema Operativo, *CVSS Score, CVSS Base Score, CVSS 3.1 Score, PCI Vuln, Category*.

Etapa 2 – Preparación y exploración: En la presente etapa los datos recopilados serán sometidos a un proceso preparación y limpieza. En el proceso de preparación se identificarán las variables que formarán parte del *dataset*. Respecto al proceso de limpieza, se eliminarán datos erróneos o que no tengan relevancia para la investigación, y se realizará transformación de datos de ser necesario.

Etapa 3 – Análisis y modelamiento: En esta etapa se desarrollará el modelo de procesamiento de datos, eligiendo las técnicas de aprendizaje automático de *Random Forest* y *XGBoost* para construir modelos predictivos y la implementación de un modelamiento no supervisado, utilizando el algoritmo *K-Means*, con el fin de segmentar las vulnerabilidades en clústeres estratégicos.

Etapa 4 – Visualización e interpretación de resultados: En esta etapa se evaluará la efectividad del modelo utilizado en la predicción de la severidad para lo cual se compararán las métricas relevantes como *Precision*, *Recall*, *F1-Score*, *Accuracy*, además de la Matriz de Confusión, y para la evaluación del agrupamiento generado por el modelo *K-means* se utilizó el Coeficiente de *Silhouette*.

4.4. Metodología para la medición de resultados de la implementación

Posterior al entrenamiento de la data con clasificadores de *Machine Learning*, este procede a ser evaluado a través de las siguientes métricas:

- ***Accuracy:*** Es una métrica de evaluación utilizada en modelos de clasificación supervisada. Representa el porcentaje de predicciones correctas realizadas por el modelo sobre el total de instancias evaluadas.

Fórmula 10

Fórmula accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Fuente: Adaptado de “Precisión y exhaustividad”, por Google Developers, s.f. (https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall?hl=es-419

- ***Precision***: La precisión es una medida de cuántas de las predicciones positivas realizadas son correctas (verdaderos positivos).

Fórmula 11

Fórmula precision

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

Fuente: Adaptado de "What are Classification Models?", por IBM, 2024 (https://www.ibm.com/es-es/think/topics/classification-models), párr. 15.

- **F1 Score:** Es la medida que combina la precisión y la recuperación. Suele describirse como la media armónica de ambas. La media armónica no es más que otra forma de calcular una "media" de valores, generalmente descrita como más adecuada para ratios (como la precisión y la recuperación) que la media aritmética tradicional.

Fórmula 12

Fórmula para la puntuación F1

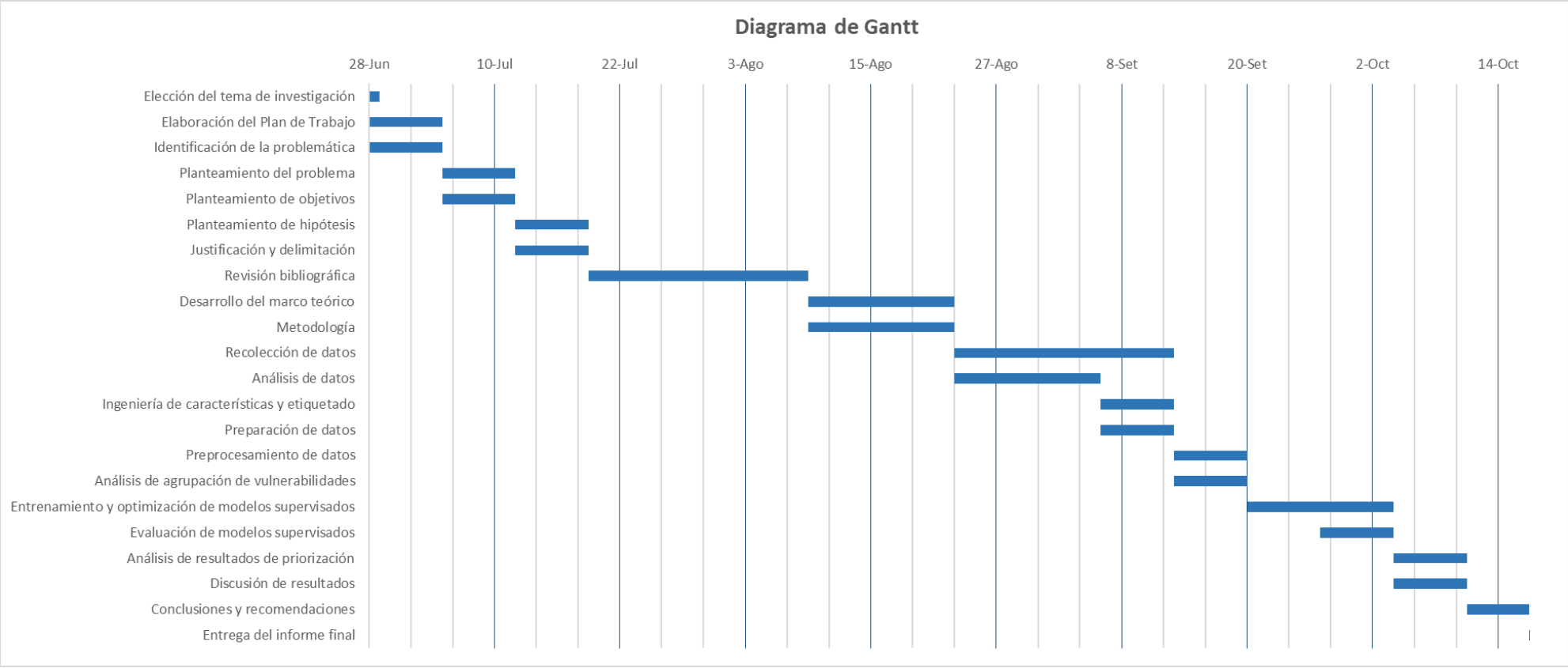
$$F1\ score = \frac{2 * (Presicion * Recall)}{Precision + Recall}$$

Fuente: Adaptado de "What are Classification Models?", por IBM, 2024 (https://www.ibm.com/es-es/think/topics/classification-models), párr. 17.

4.5. Cronograma de actividades

Figura 13

Cronograma de actividades



Fuente: Elaboración propia

4.6. Presupuesto

Tabla 5

Recursos del proyecto

Categoría	Recurso específico	Descripción	Costo
Equipamiento físico	Computo principal	Uso de 3 laptops personales de alta capacidad como entorno para la codificación, preprocesamiento de datos y pruebas	S/. 12,000
Recursos Humanos	Asesor externo	Apoyo en la validación metodológica, supervisión semanal del avance, revisión de los modelos de ML, y verificación de resultados. El presupuesto considera la tarifa profesional por 3 meses de acompañamiento (mayo, julio y septiembre)	S/. 1,500
Infraestructura de Software	Entorno de Desarrollo Python	Uso de entornos de programación (IDE) y librerías de código abierto para el desarrollo de los modelos de priorización	S/. 0
Servicios en la nube	*Google Colab*	Necesario para el entrenamiento intensivo y la optimización de modelos supervisado	S/. 0
Servicios en la nube	Almacenamiento de datos seguros	Espacio virtual para la gestión y *backup* de grandes *datasets* de vulnerabilidades	S/. 350
Servicios básicos	Luz, Internet	Esencial para alimentar las tres laptops, y crítico para la recolección de datos, el acceso a plataformas *cloud* (*Google Colab*), la revisión bibliográfica y la coordinación con tu asesor/equipo.	S/. 60
TOTAL			S/. 13,910

CAPÍTULO V: Desarrollo de la solución

5.1. Propuesta solución

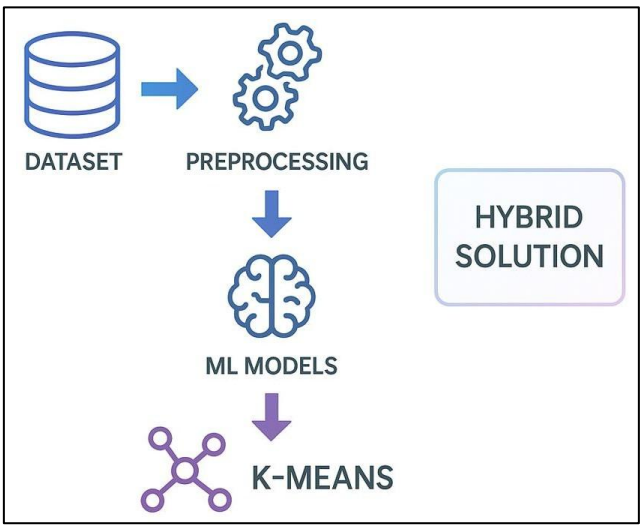
La solución propuesta combina aprendizaje supervisado y no supervisado que integra la predicción de severidad con la segmentación estratégica, optimizando la gestión y planificación del trabajo de remediación en entornos bancarios. El enfoque integra dos componentes complementarios:

Componente predictivo (supervisado): Modelos *Random Forest* y *XGBoost* entrenados con datos históricos de vulnerabilidades y su severidad. Permiten anticipar la clasificación de nuevas vulnerabilidades con métricas de desempeño superiores al 85% de *F1-Score* y *Accuracy*.

Componente exploratorio (no supervisado): Aplicación de *K-Means* para segmentar vulnerabilidades según atributos técnicos comunes. Permite organizar las "familias técnicas" (p. ej. *Windows*, *Linux*, SQL, HTTP, SMB) y facilitar la planificación en la aplicación de parches.

Figura 14

Arquitectura integral de la solución híbrida de Machine Learning



Fuente: Elaboración propia.

5.1.1. Planeamiento y descripción de actividades

El desarrollo de la solución se estructuró en cuatro fases secuenciales, orientadas a construir y validar la solución híbrida de aprendizaje automático aplicada a la gestión de

vulnerabilidades. Cada fase contribuye de forma específica al cumplimiento de los objetivos planteados.

La primera fase consistió en la extracción y consolidación de la información proveniente de la herramienta *Qualys*, la cual contiene los registros de vulnerabilidades detectadas en los activos tecnológicos del banco. En esta fase se recopilaron variables técnicas y contextuales asociadas a cada vulnerabilidad (como sistema operativo, tipo de servicio, CVSS, severidad, impacto y explotabilidad).

En la etapa de preprocesamiento, se incluyó la limpieza, codificación y normalización del conjunto de datos con el propósito de garantizar su calidad y compatibilidad con los modelos de *Machine Learning*. Se eliminaron valores atípicos, se transformaron variables categóricas a formato numérico y se aplicaron técnicas de estandarización para optimizar el desempeño de los algoritmos.

Durante el modelamiento, se desarrollaron los modelos de aprendizaje supervisado y no supervisado. El componente supervisado (mediante *Random Forest* y *XGBoost*) tuvo como objetivo predecir la severidad de las vulnerabilidades utilizando técnicas de clasificación multiclase, dado que la severidad a predecir tomó valores del 1 al 5. Se dividió el conjunto de datos en entrenamiento y prueba para garantizar una evaluación justa del rendimiento de los modelos. En esta fase también se ajustaron parámetros clave y se incorporó el uso de pesos para manejar el desbalance de clases, ya que algunas severidades estaban poco representadas. Mientras que el componente no supervisado (*K-Means*) permitió identificar patrones técnicos y agrupar vulnerabilidades en familias afines. El propósito conjunto fue generar una estructura analítica que combine precisión predictiva y segmentación estratégica.

Finalmente, se evaluó el rendimiento de ambos enfoques. Se elaboraron visualizaciones como gráficos de dispersión basados en el Análisis de Componentes Principales (PCA), matrices de correlación y comparativos de métricas de desempeño (*Accuracy*, *Precision*, *Recall* y *F1-score*). La interpretación permitió evaluar la coherencia de los resultados, validar la capacidad de generalización de los modelos y extraer conclusiones operativas para para la gestión de parches, al permitir una priorización automatizada y alineada con la realidad técnica de los activos. Además, se documentaron las métricas de evaluación y las recomendaciones para su implementación continua.

5.1.2. Desarrollo de actividades

5.1.2.1. Recolección de datos.

La recolección de datos para esta investigación se realizó a partir de un archivo proporcionado por el área de infraestructura tecnológica de una entidad financiera, herramienta *Qualys*, el cual contenía registros reales de vulnerabilidades de ciberseguridad detectadas en diversos sistemas y dispositivos tecnológicos. Estos datos fueron exportados en formato Excel (.xlsx), lo cual facilitó su análisis, procesamiento y tratamiento automatizado con herramientas de ciencia de datos.

Figura 15

Herramienta Qualys

Fuente: Herramienta Qualys

El conjunto de datos se encuentra compuesto por diversas columnas que describen las características técnicas y contextuales de cada vulnerabilidad identificada. Las principales variables utilizadas en esta investigación fueron:

***Severity*:** Nivel de severidad asignado a la vulnerabilidad, categorizado en un rango del 1 (bajo) al 5 (crítico). Esta variable es la que se busca predecir y se considera como la variable objetivo (*target*) del modelo.

CVSS y CVSS3.1: Son métricas de puntaje estándar que cuantifican el nivel de riesgo de una vulnerabilidad, según criterios técnicos definidos por el *Common Vulnerability Scoring System (CVSS).* Estas variables numéricas se utilizan como predictoras claves en el análisis.

Exploitability: Indica qué tan explotable es una vulnerabilidad, según lo determinado por la base CVSS. Es una medida técnica que refleja qué tan fácil sería para un atacante aprovechar esa vulnerabilidad.

***Malware*:** Señala si la vulnerabilidad ha sido relacionada con la presencia de *malware* conocido. Esta variable fue utilizada como un indicador de riesgo adicional y considerada también como predictora.

QID (*Qualys* ID): Identificador único de cada vulnerabilidad dentro del sistema de escaneo de seguridad. Aunque es una variable numérica, fue empleada principalmente como referencia y no como variable predictora.

***Times Detected* y *Times Reopened*:** Indican cuántas veces la vulnerabilidad fue detectada en los análisis y si ha sido reabierta tras un intento de remediación. Estas variables proporcionan una visión del comportamiento histórico de las vulnerabilidades.

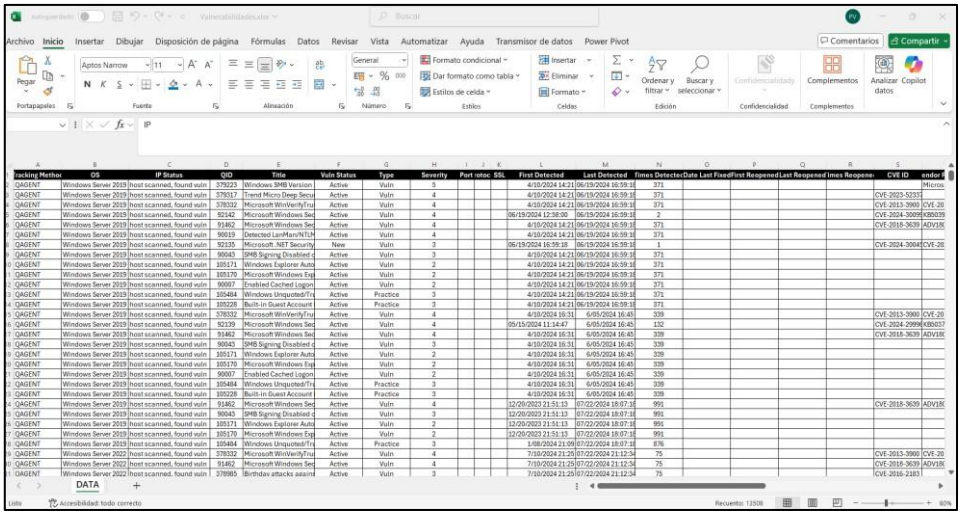
***Port*:** Número de puerto asociado al hallazgo, lo cual puede dar pistas sobre el tipo de servicio en el que fue detectada la vulnerabilidad.

Antes de ser utilizadas en los modelos de predicción, las variables fueron sometidas a un proceso de limpieza y preprocesamiento, incluyendo la imputación de valores nulos, transformación de formatos, y análisis de correlación.

Este conjunto de datos, al provenir de registros reales y contener atributos técnicos consistentes, constituyó una base sólida para la construcción y evaluación de modelos de clasificación multiclase que permitieron predecir la severidad de futuras vulnerabilidades.

Figura 16

Data extraído desde Qualys



Fuente: Herramienta Qualys

5.1.2.2. Preparación de datos.

La preparación de datos es una etapa fundamental en el desarrollo de modelos de *Machine Learning*, ya que permite transformar, limpiar y seleccionar la información necesaria para asegurar que el modelo funcione correctamente. En este caso, se utilizó un *dataset* de vulnerabilidades tecnológicas con múltiples campos numéricos y categóricos, algunos de los cuales contenían valores nulos.

5.1.2.2.1. Modelo supervisado.

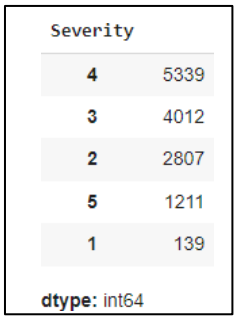
Conversión y validación de la variable objetivo: La variable *Severity*, correspondiente a la severidad de las vulnerabilidades según la escala de Qualys (1 a 5), fue convertida explícitamente a tipo numérico empleando la función pd.to_numeric(errors='coerce').

Este procedimiento garantizó que todos los registros sean interpretados correctamente por los algoritmos de *Machine Learning* y eliminó posibles inconsistencias derivadas de formatos textuales (por ejemplo, valores como "*High*" o "*Critical*") o celdas vacías.

Asimismo, se realizó un control de valores atípicos verificando que la severidad se mantuviera dentro del rango válido ($1 \leq Severity \leq 5$), descartando registros incorrectos mediante un filtrado condicional. Este paso fue clave para preservar la integridad del conjunto de etiquetas utilizadas como variable dependiente en el modelo.

Figura 17

Distribución de la variable de severidad (niveles 1 a 5)



Fuente: Elaboración propia

Limpieza y creación de variables binarias derivadas: En esta etapa se enriqueció el *dataset* mediante la construcción de nuevas variables que representaran señales binarias de riesgo. Por lo cual se crearon las siguientes columnas adicionales:

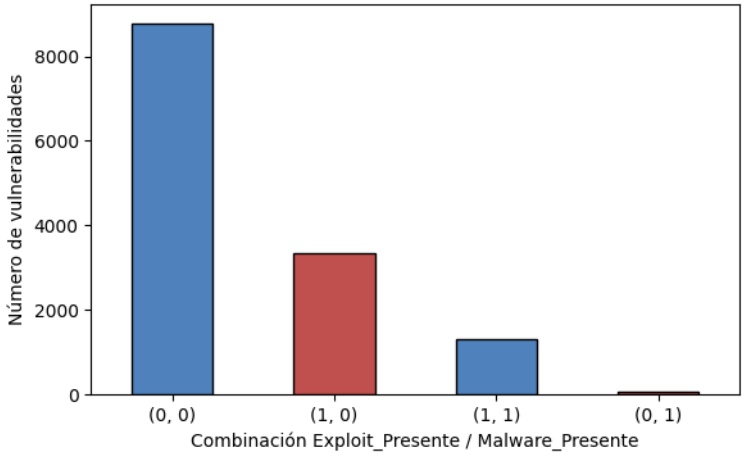
- Exploit_Presente: toma el valor 1 cuando el campo *Exploitability* contiene información sobre la existencia de exploits conocidos, y 0 en caso contrario.

- Malware_Presente: toma el valor 1 cuando la columna *Associated Malware* incluye referencias a código malicioso asociado a la vulnerabilidad, y 0 cuando está vacía.

Ambas variables se generaron utilizando operaciones vectorizadas con .notna().astype(int). El propósito fue enriquecer al modelo de indicadores de riesgo binario fácilmente interpretables, que refuercen la capacidad predictiva al vincular la severidad con la explotación activa o potencial de las vulnerabilidades.

Figura 18

Frecuencia de vulnerabilidades con exploit y malware asociado



Fuente: Elaboración propia.

La Figura 18 muestra la distribución de las vulnerabilidades en función de la presencia de *exploits* y *malware* asociados. Se observa que la mayoría de los casos (más del 60%) no presentan evidencia de explotación ni código malicioso vinculado (combinación 0,0).

Sin embargo, aproximadamente una cuarta parte de los registros corresponde a vulnerabilidades con *exploit* conocido (1,0), y un subconjunto menor, aunque crítico, presenta simultáneamente *exploit* y *malware* (1,1), lo que representa amenazas activas con mayor probabilidad de impacto.

Este resultado valida la incorporación de las variables *Exploit_Presente* y *Malware_Presente* en el modelo supervisado, al constituirse en indicadores binarios que refuerzan la capacidad predictiva de severidad y contribuyen a la priorización del trabajo de remediación.

Tratamiento de valores faltantes y depuración de datos inconsistentes: Durante la revisión inicial se identificó la presencia de valores faltantes en varias columnas, tanto numéricas como categóricas. Para evitar pérdidas de información por eliminación directa de

registros, se optó por aplicar la imputación controlada de datos utilizando la clase *SimpleImputer* de *scikit-learn*:

- En variables numéricas, se reemplazaron los valores nulos por la mediana, minimizando el impacto de *outliers*.
- En variables categóricas, se imputó el valor más frecuente (moda), con el fin de mantener la coherencia de las categorías predominantes.

Paralelamente, se descartaron columnas con más del 30% de datos faltantes o aquellas con un único valor constante, por carecer de valor explicativo.

Figura 19

Resumen de columnas con datos nulos y el valor con que se completaron

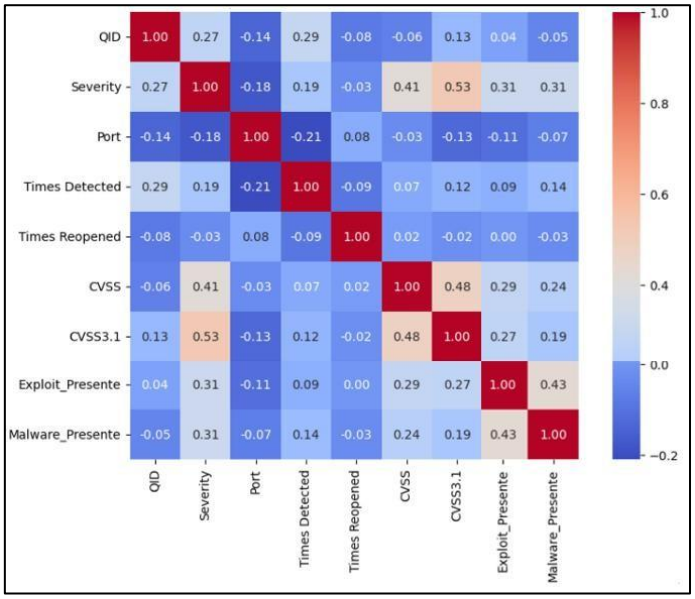
	Columna	Tipo	Nulos	Relleno
0	CVSS3.1	Numérica	2375	6.6
1	Times Reopened	Numérica	13039	1.0
2	Port	Numérica	10896	636.0

Fuente: Elaboración propia.

Selección de variables predictoras: Luego del tratamiento de nulos y transformación de variables, se inspeccionaron las columnas no numéricas y su relación con la variable *Severity* mediante matriz de correlación numérica. Destacan *CVSS3.1 = 0.53, CVSS = 0.41, Exploit_Presente = 0.31, Malware_Presente = 0.31 son cercanos a 1*, que mostraron correlaciones directas y positivas con la variable objetivo.

Figura 20

Matriz de correlación entre las variables y el target (Serverity)



Fuente: Elaboración propia.

División del conjunto de datos para evaluación de entrenamiento y prueba: Finalmente, se realizó la división del conjunto total en subconjuntos de entrenamiento (70%) y prueba (30%). Esta división garantizó independencia entre los conjuntos y permitió validar de forma objetiva la capacidad predictiva de los modelos.

Figura 21

Separa la data en train y test para el modelo de RF y XGBoost

```
X_train_rf, X_test_rf, y_train_rf, y_test_rf = train_test_split(X, y_rf, test_size=0.3, stratify=y_rf, random_state=42)
X_train_xgb, X_test_xgb, y_train_xgb, y_test_xgb = train_test_split(X, y_xgb, test_size=0.3, stratify=y_xgb, random_state=42)
```

Fuente: Elaboración propia.

5.1.2.2.1. Modelo No supervisado.

Al igual que el modelo supervisado, se crearon dos variables exp_col y mal_col, los cuales equivalen a que, si existe *Exploitability*, se transforma a 0/1, este mismo procedimiento se replicó para *Associated Malware.*

Figura 22

Cambio de los tipos de datos de Exploitability y Associated Malware

```
[Exploitability] columna detectada: 'Exploitability'
Exploitability -> 0/1
Exploitability
0    8855
1    4653
Name: count, dtype: int64
[Associated Malware] columna detectada: 'Associated Malware'
Associated Malware -> 0/1
Associated Malware
0    12124
1     1384
Name: count, dtype: int64
```

Fuente: Elaboración propia.

Imputación de valores faltantes: Se aplicó la clase *SimpleImputer* para el tratamiento de valores nulos:

- En atributos numéricos, se reemplazaron los vacíos por la mediana, reduciendo la sensibilidad a valores extremos.
- En atributos categóricos, se imputó el modo (valor más frecuente), garantizando la continuidad de las categorías principales.

Con ello, se evitó la exclusión de observaciones potencialmente relevantes y se preservó la representatividad del conjunto de datos para el análisis de similitud.

Codificación de variables categóricas: Dado que los algoritmos de agrupamiento operan exclusivamente con variables numéricas, las variables categóricas se transformaron en vectores binarios mediante *OneHotEncoder*.

Como resultado, cada vulnerabilidad quedó expresada en una estructura totalmente numérica, apta para el cálculo de distancias euclidianas dentro del algoritmo *K-Means.*

Figura 23

Reducción de valores nulos tras la codificación e imputación de datos

	Antes (%)	Después (%)
OS	0.69	0.0
CVSS3.1	17.58	0.0
CVSS3.1 Temporal	17.58	0.0
CVSS3.1 Base	17.58	0.0
Exploitability	65.55	0.0
Port	80.66	0.0
Protocol	80.66	0.0
SSL	88.27	0.0
Associated Malware	89.75	0.0

Fuente: Elaboración propia.

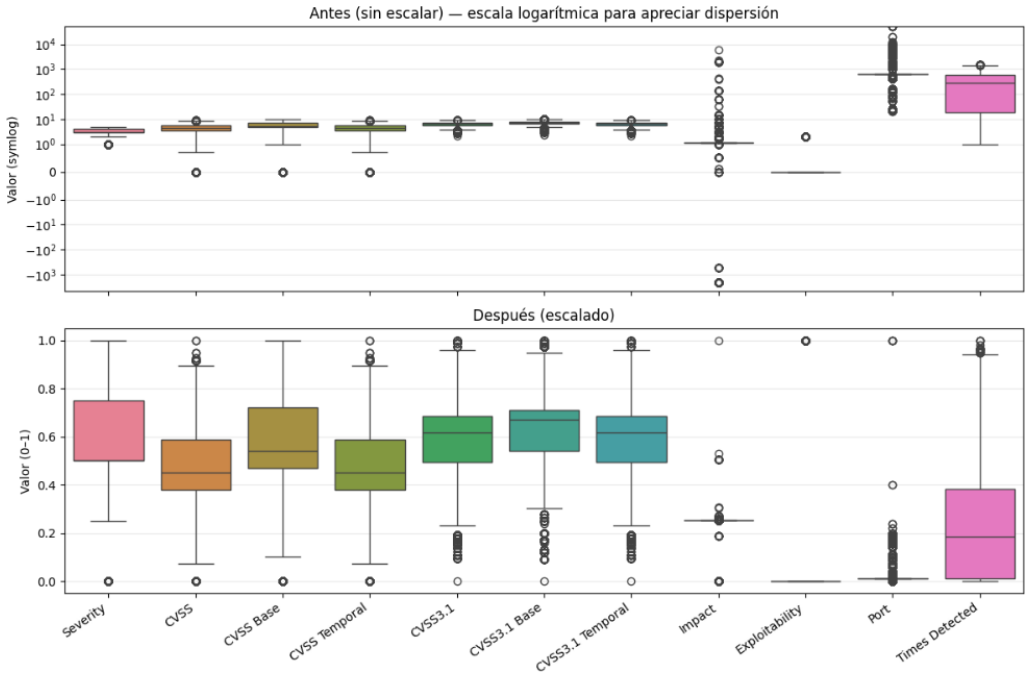
Escalado y normalización de valores: Las variables numéricas fueron sometidas a un proceso combinado de escalado utilizando *MinMaxScaler* y *RobustScaler*, dependiendo del nivel de dispersión y presencia de outliers.

- *MinMaxScaler* redujo los valores a un rango [0,1], facilitando la comparación entre métricas de distinta naturaleza.
- *RobustScaler*, basado en la mediana y el rango intercuartílico, redujo el impacto de *outliers*.

El propósito fue homogeneizar las escalas para que cada variable contribuyera equitativamente al proceso de agrupamiento. Tal como se puede observar en la Figura 24, en el panel superior, antes del escalado y con escala semilogarítmica, se aprecia la heterogeneidad de rangos entre variables: *Port* y *Times Detected* presentan colas largas y órdenes de magnitud superiores a *CVSS3.1* y *Severity*. En el panel inferior, tras el escalado (0–1), todas las variables quedan en magnitudes comparables, lo que evita que atributos con gran escala dominen el cálculo de distancias en *K-Means* y permite que las métricas *CVSS3.1* conserven su poder discriminativo en la formación de clústeres.

Figura 24

Efecto del escalado en las distribuciones numéricas



Fuente: Elaboración propia.

Integración del preprocesamiento en un *pipeline*: Finalmente, todas las transformaciones (imputación, codificación y escalado) fueron integradas en un *pipeline* completo, ejecutado sobre el *DataFrame* consolidado. El resultado fue una matriz final (X_prepared) lista para el análisis de agrupamiento mediante *K-Means*, posterior reducción dimensional por PCA (2D) y enriquecimiento por categorías.

El pipeline se aplicó de forma uniforme y reproducible, asegurando la consistencia del preprocesamiento en futuras ejecuciones o pruebas de sensibilidad.

5.1.2.3. Análisis y modelado.

5.1.2.3.1. Modelo supervisado.

Se entrenaron dos familias de modelos basados en árboles de decisión y sus variantes con manejo de desbalance:

***Random Forest* (RF):** Configuración base con número de árboles suficiente para estabilidad y random_state fijado para reproducibilidad. Se evaluó también la variante con class_weight='balanced' para mitigar sesgos hacia clases frecuentes.

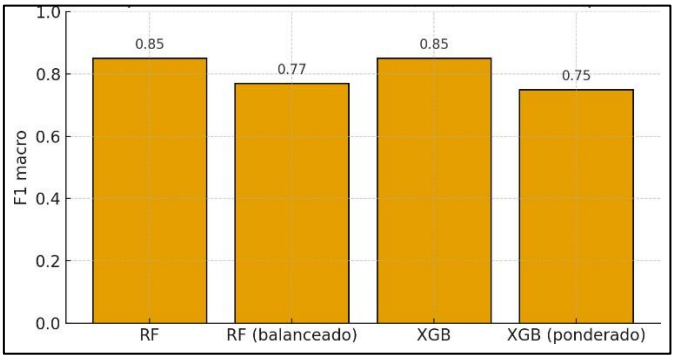
***XGBoost* (XGB):** Se evaluó la versión ponderada mediante sample_weight derivado de compute_sample_weight(class_weight='balanced') para elevar el *Recall* de clases minoritarias.

La evaluación se basó en *Accuracy, Precision, Recall* y F1 por clase (con énfasis en F1 macro). La matriz de confusión permitió estudiar la distribución de errores: en escenarios de severidad, las confusiones suelen concentrarse entre niveles contiguos (3-4), lo cual es metodológicamente esperable. La comparación base versus balanceado/ponderado mostró la ganancia de *Recall* en clases menos representadas con un costo marginal en exactitud global, mejorando el equilibrio por clase.

Se priorizó el modelo con mejor F1 macro manteniendo una exactitud competitiva y una diagonal dominante en la matriz de confusión.

Figura 25

Comparación de F1-macro entre RF, RF balanceado, XGB y XGB ponderado.

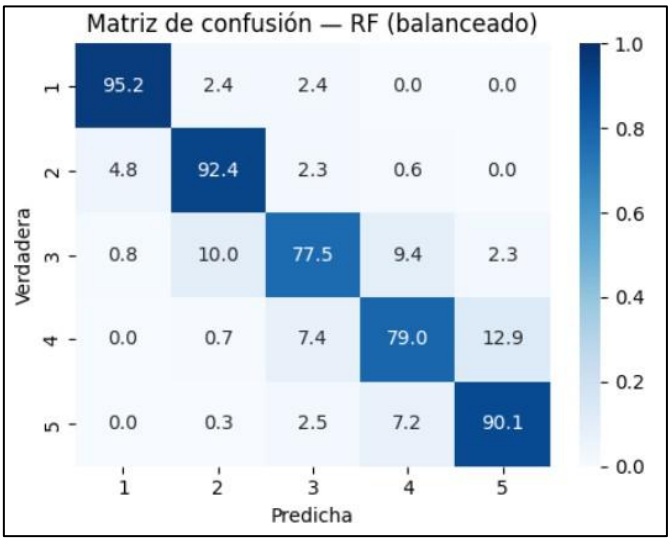


Fuente: Elaboración propia.

Las versiones sin balancear maximizan el F1-macro, favoreciendo el desempeño promedio en todas las clases; sin embargo, tienden a perder una fracción mayor de casos de severidades 1 y 5 (clases minoritarias). En cambio, las variantes balanceadas/ponderadas elevan el *Recall* en estas clases raras, reduciendo el riesgo de omitir hallazgos críticos, pero a costa de una menor precisión (mayor volumen de falsos positivos en 1 y 5) y, por ende, un F1-macro global ligeramente inferior, tal como se puede observar en la Figura 25.

Figura 26

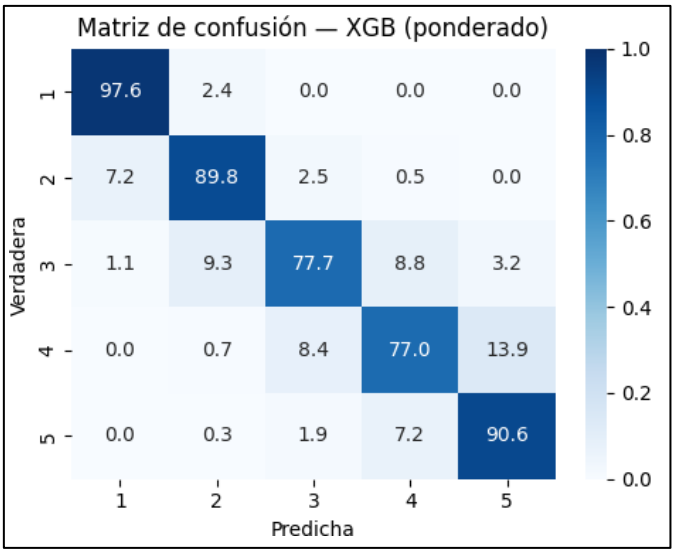
Matriz de confusión de RF balanceado con porcentajes por clase



Fuente: Elaboración propia.

Figura 27

Matriz de confusión de XGBoost ponderado con porcentajes por clase



Fuente: Elaboración propia.

En las Figuras 26 y 27 se pueden observar las matrices normalizadas con diagonales elevadas en las clases extremas (1 y 5), lo que evidencia una buena capacidad de recuperación (*Recall*) cuando se prioriza el desbalance. En clase 1, el modelo *XGBoost* ponderado alcanza

97.6% frente a 95.2% de RF balanceado, mientras que RF es superior en clase 2 (92.4% vs. 89.8%) y ligeramente en clase 4 (79.0% vs. 77.0%). En clase 5 ambos modelos superan el 90%. Las confusiones predominan entre clases contiguas (3-2 y 3-4; 4-5), patrón coherente con la continuidad de la escala de severidad. Operativamente, si el objetivo es minimizar falsos negativos en los extremos, ambos enfoques son adecuados; si se desea reducir escaladas indebidas 4-5, RF balanceado resulta preferible, mientras que para maximizar la detección de severidad 1 el modelo *XGBoost* ponderado ofrece una ligera ventaja.

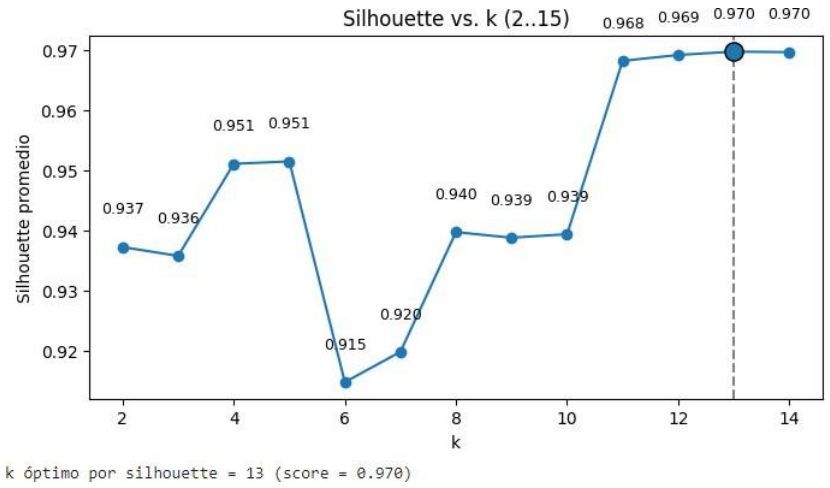
5.1.2.3.2. Modelo no supervisado.

Se realizó una segmentación no supervisada para descubrir "familias técnicas" de vulnerabilidades, a fin de organizar el trabajo de aplicación de parches por sistema operativo, protocolo/servicio y señales de explotación. Este análisis complementa al supervisado: mientras aquel modelo estima qué tan grave es una vulnerabilidad, la segmentación indica dónde actuar primero y por qué.

Se evaluó k ∈ [2, 14] mediante *silhouette_score*, seleccionándose k = 13 por presentar el mayor *silhouette* promedio (97%) y pertenecer a un rango de soluciones equivalentes (11–14), lo que sugiere estabilidad de la partición. Frente a alternativas más agregadas (k=4–5), k=13 permite distinguir familias técnicas útiles para planificar la aplicación de parches (Windows, Cisco, Linux/CentOS, Web, SMB, etc.) sin perder calidad de separación.

Figura 28

Curva de Silhouette para K-Means (k=2..15)

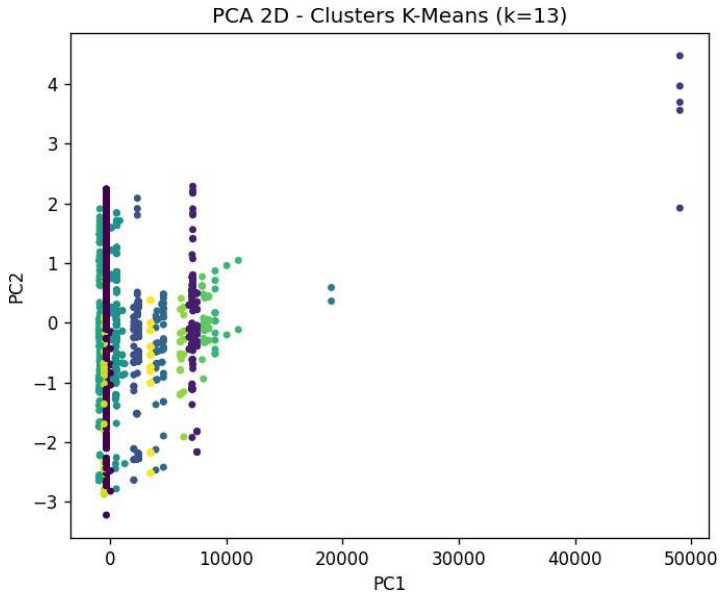


Fuente: Elaboración propia.

La proyección PCA 2D se utilizó exclusivamente para validación visual de la estructura hallada, la optimización de *K-Means* se realizó en el espacio completo preprocesado. En la Figura 29, se evidencia (i) una franja central con grupos numerosos y parcialmente solapados en 2D (familias frecuentes), y (ii) clústeres periféricos muy separados en *Principal Component* 1 (PC1), familias técnicas raras y bien definidas. El solapamiento en 2D no contradice la calidad del particionamiento en el espacio completo, corroborado por un *silhouette* promedio=0.97. Los clústeres extremos constituyen candidatos naturales a aplicación de parches por lotes (p. ej., dominios *Cisco*, *servicios específicos*), mientras que los centrales se gestionan con criterios de prioridad combinando CVSS3.1 medio, *Exploitability* y frecuencia.

Figura 29

PCA 2D de los clústeres (k=13)



Fuente: Elaboración propia.

Para transformar los clústeres en unidades accionables de aplicación de parches, se elaboraron perfiles y análisis de enriquecimiento:

- Perfil numérico por clúster: En la Figura 30 se observa el cálculo de las medias y desviaciones de métricas cuantitativas (CVSS/3.1, *Impact*, *Exploitability*, *Port*, *Times Detected*), facilitando la lectura de criticidad y exposición. En la Figura 31 resume el ranking de clústeres según su criticidad promedio. Se observa que los clústeres 0 y 6 presentan las mayores medias de CVSS3.1 (6.7 y 6.3), seguidos por los clústeres 1–3 y 11 (5.5–5.6), mientras que los clústeres 12 y 5 exhiben valores más moderados (5.0).

Este ranking constituye un insumo directo para priorizar la aplicación de parches por familias técnicas: primero los clústeres con severidad media más alta, y, dentro de estos, aquellos con mayor *Exploitability* y mayor número de detecciones

Figura 30

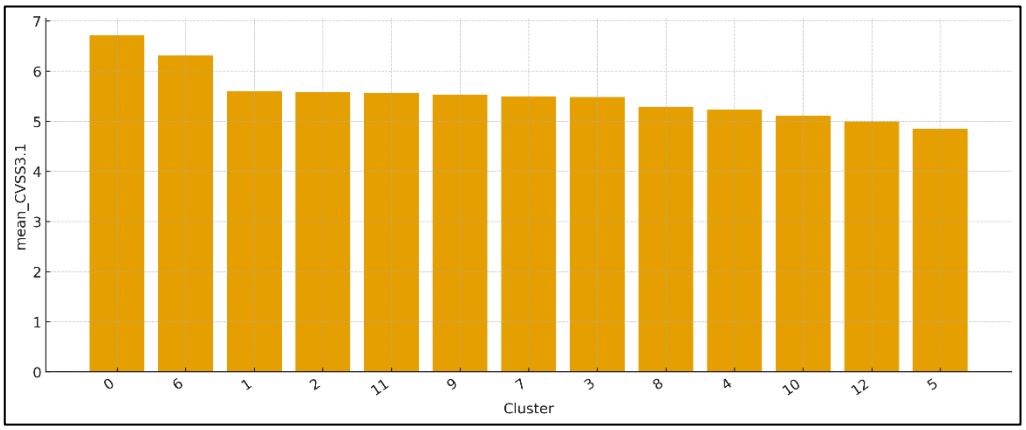
Perfil numérico por clúster

	cluster	mean_CVSS3.1	mean_Port	mean_PCI Vuln	mean_Exploitability	mean_Associated Malware	std_CVSS3.1	std_Port	std_PCI Vuln	std_Exploitability	std_Associated Malware
0	0	6.708113	668.666667	0.878547	0.387444	0.125376	1.373067	115.917595	0.326653	0.487166	0.331145
1	1	5.603468	8069.697872	0.885106	0.080851	0.004255	1.080724	155.697081	0.318894	0.272606	0.065094
2	2	5.580000	50001.000000	0.600000	0.200000	0.000000	1.177115	0.000000	0.489898	0.400000	0.000000
3	3	5.483059	3351.357576	0.771717	0.090909	0.004040	0.973481	93.566308	0.419726	0.287480	0.063436
4	4	5.231707	5282.142857	0.825397	0.111111	0.000000	0.902960	281.474342	0.379627	0.314270	0.000000
5	5	4.850000	20001.000000	1.000000	0.000000	0.000000	0.150000	0.000000	0.000000	0.000000	0.000000
6	6	6.307570	1472.629747	0.844937	0.151899	0.000000	1.300560	160.168717	0.361965	0.358923	0.000000
7	7	5.492326	64.730132	0.746689	0.304636	0.009934	1.698917	36.900327	0.434908	0.460253	0.099172
8	8	5.292308	10335.111111	0.944444	0.166667	0.000000	0.408490	666.528893	0.229061	0.372678	0.000000
9	9	5.531111	9137.690909	0.909091	0.018182	0.000000	0.300239	180.055029	0.287480	0.133609	0.000000
10	10	5.107407	7082.862069	0.862069	0.068966	0.000000	0.722099	125.185855	0.344828	0.253395	0.000000
11	11	5.567383	443.599398	0.802711	0.132530	0.001506	1.070417	6.308275	0.397952	0.339066	0.038778
12	12	4.994118	4445.555556	0.666667	0.142857	0.000000	1.181460	1.706921	0.471405	0.349927	0.000000

Fuente: Elaboración propia.

Figura 31

Severidad media (CVSS3.1) por clúster

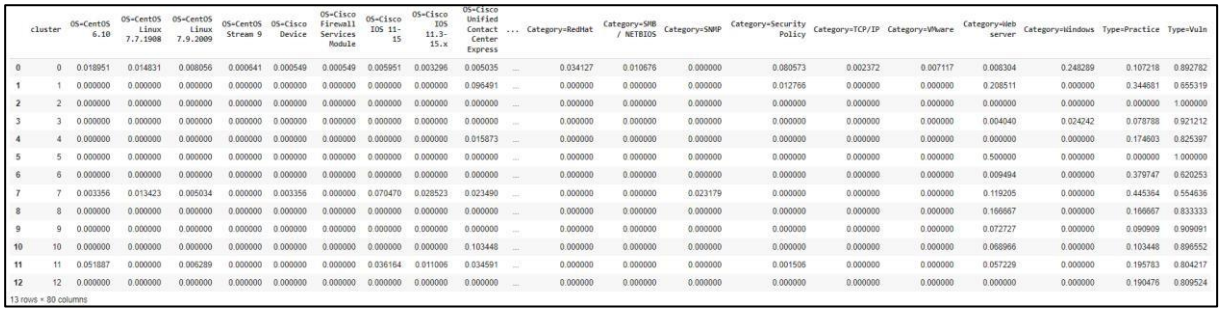


Fuente: Elaboración propia.

- **Perfil categórico por clúster:** Se obtuvo la proporción de categorías relevantes, tales como *Operating System* (OS), *Category*, *Protocol*, visibilizando la concentración tecnológica de cada grupo.

Figura 32

Perfil categórico por clúster



	cluster	OS=CentOS 6.10	OS=CentOS Linux 7.7.1908	OS=CentOS Linux 7.9.2009	OS=CentOS Stream 9	OS=Cisco Device	OS=Cisco Firewall Services Module	OS=Cisco IOS 11-15	OS=Cisco IOS 11.3-15.x	OS=Cisco Unified Contact Center Express	...	Category=RedHat	Category=SMB / NETBIOS	Category=SNMP	Category=Security Policy	Category=TCP/IP	Category=VMware	Category=Web server	Category=Windows	Type=Practice	Type=Vuln
0	0	0.018951	0.014831	0.008056	0.000641	0.000549	0.000549	0.005951	0.003296	0.005035	...	0.034127	0.010676	0.000000	0.080573	0.002372	0.007117	0.008304	0.248289	0.107218	0.892782
1	1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.096491	...	0.000000	0.000000	0.000000	0.012766	0.000000	0.000000	0.208511	0.000000	0.344681	0.655319
2	2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
3	3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.004040	0.024242	0.078788	0.921212
4	4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.015873	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.174603	0.825397
5	5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.500000	0.000000	0.000000	1.000000
6	6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.009494	0.000000	0.379747	0.620253
7	7	0.003356	0.013423	0.005034	0.000000	0.003356	0.000000	0.070470	0.028523	0.023490	...	0.000000	0.000000	0.023179	0.000000	0.000000	0.000000	0.119205	0.000000	0.445364	0.554636
8	8	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.166667	0.000000	0.166667	0.833333
9	9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.072727	0.000000	0.090909	0.909091
10	10	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.103448	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.068966	0.000000	0.103448	0.896552
11	11	0.051887	0.000000	0.006289	0.000000	0.000000	0.000000	0.036164	0.011006	0.034591	...	0.000000	0.000000	0.000000	0.001506	0.000000	0.000000	0.057229	0.000000	0.195783	0.804217
12	12	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.190476	0.809524

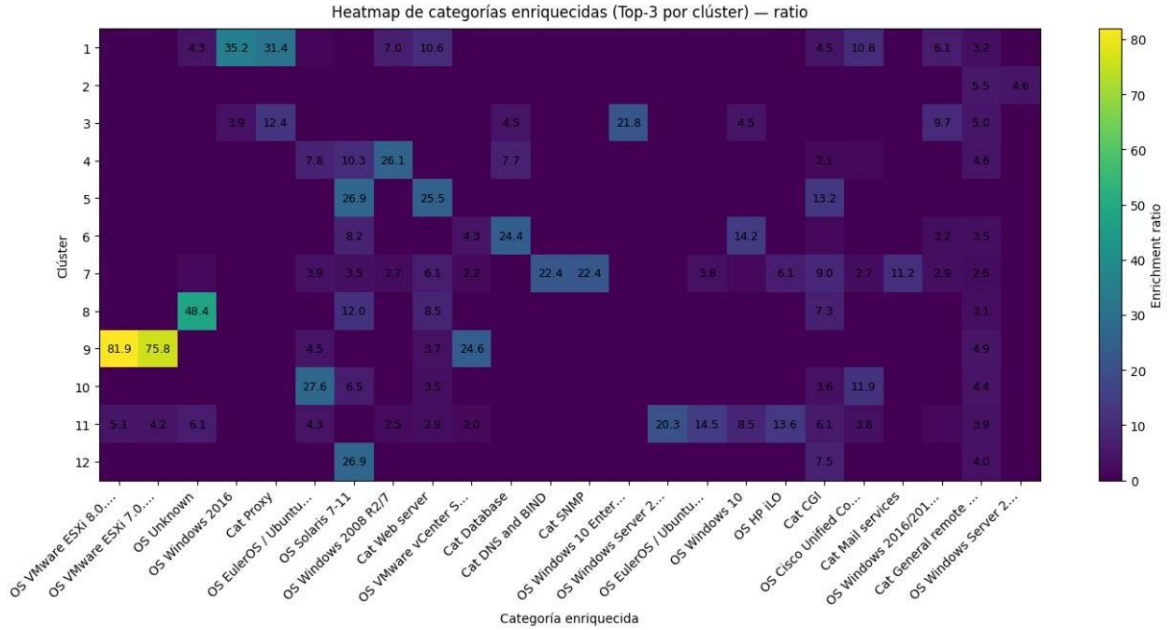
13 rows × 80 columns

Fuente: Elaboración propia.

- **Enriquecimiento (*over-representation*):** Para cada variable categórica se calculó el *enrichment ratio* (*cluster_share* / *global_share*). La Figura 33 revela clústeres fuertemente monotecnológicos, tales como VMware/ESXi, Windows en versiones concretas y otros mixtos de infraestructura (DNS, CGI, servicios de red). Los clústeres con ratios muy altos (>10) son candidatos directos a aplicación de parches por lotes con propietarios claros (Virtualización, *Windows*, Redes). En clústeres mixtos, la estrategia combina prioridad por severidad (CVSS3.1) y coordinación operativa (ventanas de mantenimiento por servicio).

Figura 33

Heatmap de enriquecimiento de variables



Fuente: Elaboración propia.

5.1.2.4. Evaluación y reporte.

En esta etapa se evaluaron los resultados de los dos frentes del trabajo. En el modelo supervisado de clasificación de severidad, los algoritmos *Random Forest* y *XGBoost*, entrenados con el mismo pipeline de preprocesamiento, mostraron el mejor desempeño global en sus versiones base, alcanzando F1-macro = 77% (RF) y = 75% (XGB). Las variantes balanceadas/ponderadas incrementaron el *recall* en las clases menos frecuentes (severidades 1 y 5), aunque con una ligera caída del F1-macro por la mayor tasa de falsos positivos, lo cual es coherente con un objetivo de no omitir vulnerabilidades críticas.

En el análisis no supervisado, la validación interna con la curva *Silhouette=0.97* determinó un óptimo en k =13, evidenciando clústeres compactos y bien separados. La PCA 2D se empleó como apoyo visual, y la caracterización con perfiles y *heatmaps* de *enrichment ratio* confirmó clústeres monotecnológicos y otros orientados a servicios de red, proporcionando insumos directos para segmentar la aplicación de parches por familias técnicas y asignar responsables. En conjunto, los resultados validan la propuesta integral: predicción confiable de severidad y segmentación estratégica para la gestión de parches.

5.2. Medición de la solución

5.2.1. Análisis de indicadores cuantitativos y/o cualitativos

Para la data de la entidad bancaria, se ha llevado a cabo un análisis cuantitativo detallado de las vulnerabilidades detectadas, utilizando los modelos de aprendizaje supervisado *Random Forest*, *XGBoost* y el modelo no supervisado con *K-Means*. Los resultados obtenidos nos permiten evaluar patrones con un enfoque numérico.

Para la evaluación del desempeño del modelo predictivo para la severidad, se utilizaron dos algoritmos de clasificación supervisada los cuales son *Random Forest* y *XGBoost*, asimismo, se utilizó el modelo de clasificación no supervisada llamada *K-Means*.

Ambos modelos fueron entrenados y evaluados tomando en cuenta dos condiciones: sin la aplicación de balanceo y bajo la aplicación de un balanceador. Se utilizaron las métricas mencionadas anteriormente para la evaluación y la decisión del mejor modelo en la comparativa global.

5.2.1.1. *Random Forest* – Sin balance.

El primer experimento se hizo con el modelo *Random Forest* sin usar algún tipo de balanceo. En la Figura 34 podemos ver los resultados siguientes:

- Clase 2 (baja) y Clase 4 (alta): registraron un alto *Recall* (96% y 92%) y un buen *F1-Score* (>88%), lo que indica que el modelo logró identificarlas correctamente en la mayoría de los casos.
- Clase 3 (media): presentó un *Recall* más bajo (77%), evidenciando que el modelo no detecta todos los verdaderos positivos en esta categoría.
- Clase 5 (crítica): alcanzó una alta precisión (92%), pero un *Recall* reducido (68%). Esto sugiere que el modelo acierta cuando clasifica un caso como clase 5, aunque omite varios casos reales (falsos negativos).
- Clase 1 (muy baja): pese a contar con pocos datos (42 muestras), el modelo obtuvo un desempeño aceptable considerando su bajo soporte.

  En cuanto a las métricas globales se obtuvieron estos resultados:

- *Accuracy* (86%): Buen resultado global.
- Macro avg (85%): Rendimiento balanceado entre clases (sin importar su frecuencia).
- *Weighted* avg (86%): Se consideraron el número de muestras por clase, lo que hizo posible mostrar que el modelo mantuvo un rendimiento apropiado aun sin utilizar técnicas de balanceo.

Figura 34

Clasificación y Matriz de Confusión del Modelo Random Forest Sin Balancear

```
Reporte de Clasificación (Random Forest - Sin Balancear):
              precision    recall  f1-score   support

           1       0.83      0.93      0.88        42
           2       0.88      0.96      0.92       842
           3       0.87      0.77      0.82      1204
           4       0.84      0.92      0.88      1602
           5       0.92      0.68      0.78       363

    accuracy                           0.86      4053
   macro avg       0.87      0.85      0.85      4053
weighted avg       0.86      0.86      0.86      4053


Matriz de Confusión (Random Forest - Sin Balancear):
[[  40    1    1    0    0]
 [  40  778   19    5    0]
 [  10  120  933  113   28]
 [   0   11  119 1265  207]
 [   0    1    9   26  327]]
```

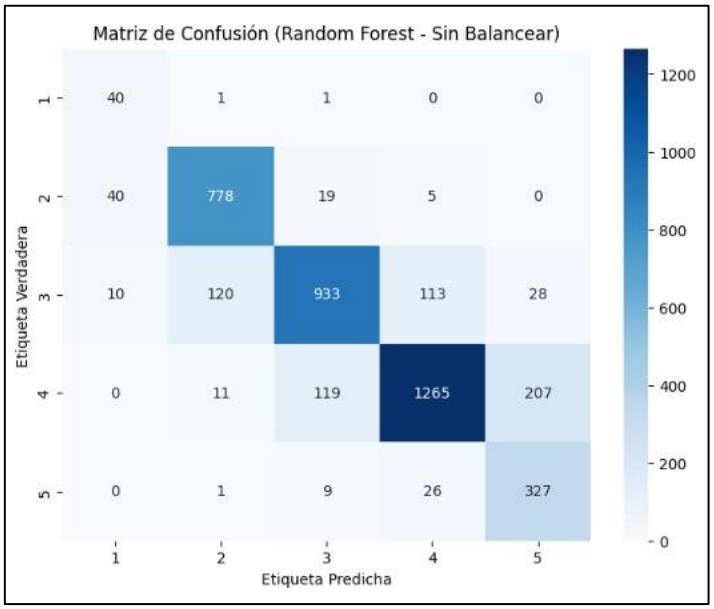
Fuente: Elaboración Propia

En la Figura 35 podemos visualizar a mayor escala la matriz de confusión donde se tiene con claridad un mayor análisis. Se obtuvieron los siguientes resultados:

- El modelo mostró un rendimiento general satisfactorio, con una precisión (86% de *Accuracy*) a pesar de que no se aplicaron técnicas para equilibrar las clases.
- Sin embargo, se detectó una propensión a confundir clases adyacentes (como las categorías 3 y 4 o 4 y 5), lo cual podría ser riesgoso en situaciones donde es necesaria una precisión elevada para las clases más críticas.
- En concreto, la clase 5 (crítica) mostró un *Recall* menor al anticipado; por lo tanto, se sugiere que se tenga en cuenta el empleo de métodos de balanceo, como el SMOTE, o la modificación de los pesos de clase para incrementar la habilidad del modelo para identificar adecuadamente los casos más significativos.

Figura 35

Matriz de Confusión del Modelo Random Forest Sin Balancear



Fuente: Elaboración Propia

5.2.1.2. *XGBoost* – Sin balance.

El segundo experimento se hizo con el modelo *XGBoost* sin usar algún tipo de balanceo. En la Figura 36 podemos ver los resultados siguientes:

- Clase 1 (baja): Presentó un desempeño sobresaliente, con un *Recall* de 96% y una precisión de 88% se evidenció la capacidad del modelo para reconocer correctamente esta categoría.
- Clase 3 (alta): Demostró un balance adecuado entre el *Recall* (91%) y el *F1-score* (88%), lo que indica una conducta estable y coherente en la categorización.
- Clase 0 (muy baja): El modelo logró detectar correctamente el 93% de la clase minoritaria, a pesar de que contaba con solo 42 muestras. El *F1-score* fue de 89%.
- Clase 2 (media): Su desempeño fue mejor que el del modelo *Random Forest*, con un *recall* de 77%, lo que señala que se perdieron ciertos casos reales; no obstante, se observó una mejora en la detección.
- Clase 4 (crítica): Logró una exactitud de 89%, sin embargo, el *recall* fue de 69%, lo que demostró que se pasaron por alto algunos casos importantes (falsos negativos); ello supone un reto para detectar vulnerabilidades críticas.

  En cuanto a las métricas globales se obtuvieron estos resultados:

- El modelo mantuvo un rendimiento general sólido (86%), similar al de *Random Forest*.
- F1 Macro promedio (0.85) indicó que el rendimiento es relativamente equilibrado entre clases.

Figura 36

Clasificación y Matriz de Confusión del Modelo XGBoost Sin Balancear

```
Reporte de Clasificación (XGBosst - Sin Balancear):
              precision    recall  f1-score   support

           0       0.85      0.93      0.89        42
           1       0.88      0.96      0.92       842
           2       0.87      0.77      0.82      1204
           3       0.84      0.91      0.88      1602
           4       0.89      0.69      0.78       363

    accuracy                           0.86      4053
   macro avg       0.87      0.85      0.85      4053
weighted avg       0.86      0.86      0.86      4053


Matriz de Confusión (XGBosst - Sin Balancear):
[[  39    2    1    0    0]
 [   0  805   28    9    0]
 [   7  104  928  163    2]
 [   0    3  107 1463   29]
 [   0    1    7  104  251]]
```

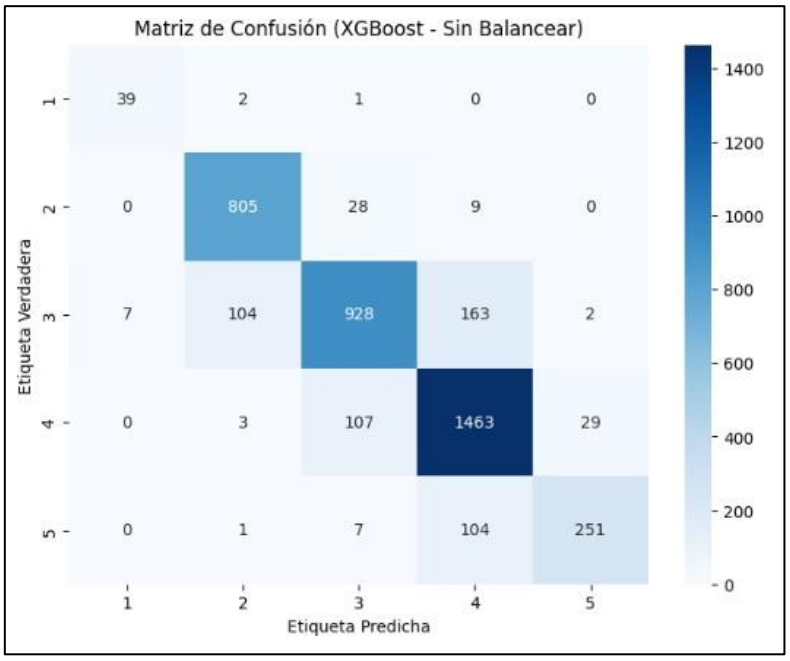
Fuente: Elaboración Propia

En la Figura 37 podemos visualizar a mayor escala la matriz de confusión donde se tiene con claridad un mayor análisis. Por lo cual podemos obtener los siguientes resultados:

- *XGBoost* sin balancear logró resultados muy similares a *Random Forest*, pero con ligera ventaja en clases intermedias (2 y 3).
- La clase crítica (4) continuó teniendo *Recall* bajo.
- Aplicar balanceo (como SMOTE) en futuros experimentos es una buena alternativa para mejorar el *Recall* de clases minoritarias sin sacrificar precisión.

Figura 37

Matriz de Confusión del Modelo XGBoost Sin balancear



Fuente: Elaboración Propia

5.2.1.3. *Random Forest* – Balanceado.

El tercer experimento se hizo con el modelo *Random Forest* usando el balanceo proporcionado por uno de los argumentos propios que nos brinda la librería. En la Figura 38 podemos ver los resultados siguientes:

- Clase 1 (muy baja): El modelo recuperó casi todos los casos (*Recall* = 95%), pero tuvo falsos positivos (precisión muy baja: 44%). Esto indica sobreajuste debido al balanceo, al incluir demasiados ejemplos sintéticos de esa clase.
- Clase 2 (baja) y Clase 3 (media): Rendimiento sólido, comparable con el modelo sin balancear.

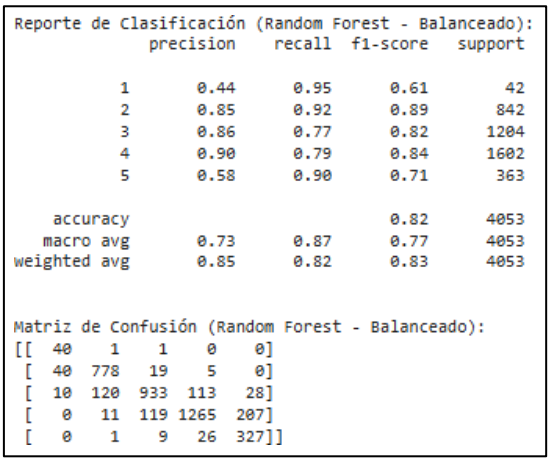
- Clase 4 (alta): Aunque la precisión fue alta (0.90), el *Recall* bajó a 79%. Ello indica que no se tomó en cuenta algunos casos verdaderos.
- Clase 5 (crítica): Obtuvo una mejora considerable en *Recall* (90% contra 68% del modelo sin balancear), lo cual es clave en contextos de ciberseguridad. Aunque pierde precisión (0.58), ahora detecta casos críticos.

  En cuanto a las métricas globales se obtuvieron los siguientes resultados:

- *Accuracy* general disminuyó ligeramente en comparación con el modelo sin balanceo (82% contra 86%), lo cual resulta esperable al priorizar la detección de clases minoritarias.
- Macro *F1-score* (77%) mostró una mejora respecto al modelo sin balancear (75%), aunque refleja el impacto de un menor desempeño en la clase 1.
- *Weighted F1-score* (83%) se mantuvo competitivo, lo que indica que el rendimiento global del modelo fue sólido, considerando la ponderación de las métricas según el soporte de cada clase.

Figura 38

Clasificación y Matriz de Confusión del Modelo Random Forest Balanceado



```
Reporte de Clasificación (Random Forest - Balanceado):
              precision    recall  f1-score   support

           1       0.44      0.95      0.61        42
           2       0.85      0.92      0.89       842
           3       0.86      0.77      0.82      1204
           4       0.90      0.79      0.84      1602
           5       0.58      0.90      0.71       363

    accuracy                           0.82      4053
   macro avg       0.73      0.87      0.77      4053
weighted avg       0.85      0.82      0.83      4053


Matriz de Confusión (Random Forest - Balanceado):
[[  40    1    1    0    0]
 [  40  778   19    5    0]
 [  10  120  933  113   28]
 [   0   11  119 1265  207]
 [   0    1    9   26  327]]
```

Fuente: Elaboración Propia

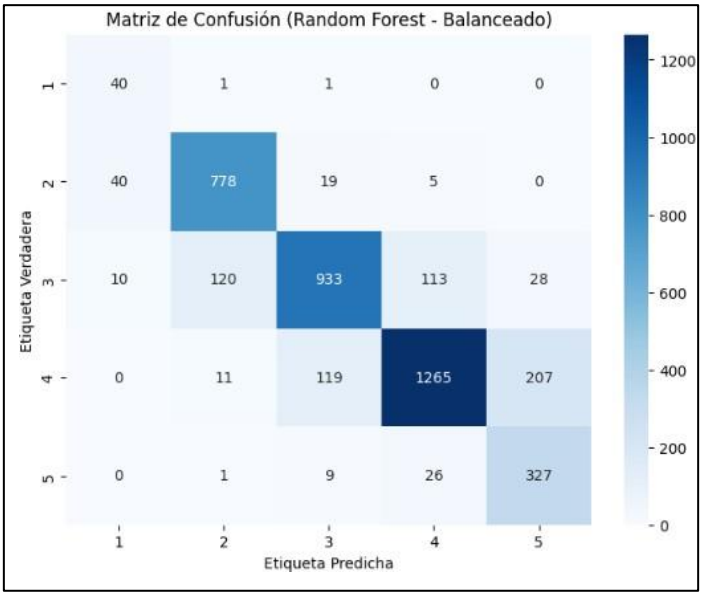
En la Figura 39 podemos visualizar a mayor escala la matriz de confusión donde se tiene con claridad un mayor análisis. Por lo cual podemos obtener los siguientes resultados:

- El balanceo de clases posibilitó que el *Recall* en las categorías minoritarias, especialmente en la clase 1 (muy baja) y la clase 5 (crítica), mejorara notablemente, lo cual reforzó la habilidad del modelo para detectar situaciones menos representadas.

- Este resultado es muy importante en situaciones reales, donde es más relevante identificar las vulnerabilidades críticas que conseguir una clasificación de los casos más comunes que sea perfectamente equilibrada.
- Como era de esperar, se sacrificó parcialmente la precisión general (*Accuracy*), con la introducción de una cantidad moderada de falsos positivos, particularmente en la clase 1.
- Por lo tanto, el modelo actuó como una versión menos conservadora y más sensible, lo cual podría ser beneficioso en sistemas de alerta temprana, donde se prefiere identificar más casos a pesar de que esto pueda conllevar algunos errores extras.

Figura 39

Matriz de Confusión del Modelo Random Forest Balanceado



Fuente: Elaboración Propia

5.2.1.4. *XGBoost* – Balanceado.

El cuarto y último experimento se hizo con el modelo *XGBoost* usando una librería externa para realizar el balanceo. En la Figura 40 podemos ver los resultados siguientes:

- Clase 0 (muy baja): El modelo acertó el 98% de los casos verdaderos (*Recall* muy alto), pero tuvo baja precisión (36%), lo que indicó muchos falsos positivos. La métrica *F1-Score* reflejó este desequilibrio.
- Clase 1 (baja): Excelente rendimiento general (*F1-Score*: 88%).

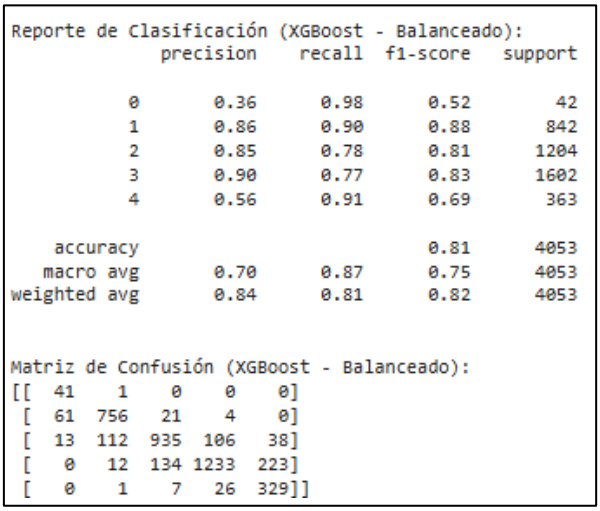
- Clase 2 (media): Buen rendimiento (*F1-Score*: 81%), con una ligera confusión con clases vecinas.
- Clase 3 (alta): Alto desempeño general, sin embargo, el *Recall* obtuvo un puntaje más bajo que en el modelo sin balancear.
- Clase 4 (crítica): Mejora notable en *Recall* (91%) frente al modelo sin balancear (donde tenía 68%). Lo cual indica que se detectó la mayoría de las vulnerabilidades críticas, a costa de menor precisión.

  En cuanto a las métricas globales se obtuvieron estos resultados:

- *Accuracy* global disminuyó ligeramente (de 86% a 81%), pero a cambio mejoró el *Recall* en clases minoritarias, que es el objetivo principal del balanceo.
- Macro F1 (75%) reflejó rendimiento promedio entre clases, penalizado por el bajo puntaje *F1-Score* en clase 0.
- *Weighted* F1 (82%) resultado sólido, indicando un modelo razonablemente balanceado a nivel general.

Figura 40

Clasificación y Matriz de Confusión del Modelo XGBoost Balanceado



```
Reporte de Clasificación (XGBoost - Balanceado):
              precision    recall  f1-score   support

           0       0.36      0.98      0.52        42
           1       0.86      0.90      0.88       842
           2       0.85      0.78      0.81      1204
           3       0.90      0.77      0.83      1602
           4       0.56      0.91      0.69       363

    accuracy                           0.81      4053
   macro avg       0.70      0.87      0.75      4053
weighted avg       0.84      0.81      0.82      4053


Matriz de Confusión (XGBoost - Balanceado):
[[  41    1    0    0    0]
 [  61  756   21    4    0]
 [  13  112  935  106   38]
 [   0   12  134 1233  223]
 [   0    1    7   26  329]]
```

Fuente: Elaboración Propia

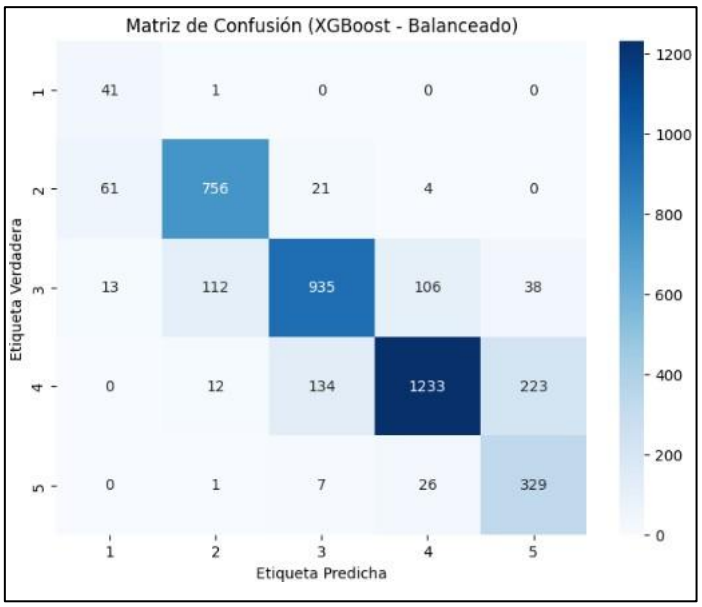
En la Figura 41 se visualiza la matriz de confusión donde se obtuvieron los siguientes resultados:

- El balanceo con *XGBoost* mejoró sustancialmente la detección de clases minoritarias, especialmente clase 4 (crítica) y clase 0 (muy baja).

- Como resultado, hubo mayor sensibilidad (*recall* alto) aunque ello implico una mayor cantidad de falsos positivos, lo que redujo la precisión general.
- *Accuracy* y la precisión general se redujeron un poco; no obstante, la cobertura de las clases más importantes presentó una mejora notable, lo cual es más valioso en un sistema de ciberseguridad proactivo, donde detectar a tiempo situaciones críticas es esencial.

Figura 41

Matriz de Confusión del Modelo XGBoost Balanceado



Fuente: Elaboración Propia

Como resultado de este estudio, se entrenaron y probaron cuatro modelos de clasificación con el objetivo de predecir la severidad de vulnerabilidades, medida en cinco niveles (1 a 5). Se aplicaron dos algoritmos (*Random Forest* y *XGBoost)* bajo dos condiciones distintas: sin balance de clases y con datos balanceados.

Las métricas clave para la evaluación fueron *Accuracy*, Macro *F1-Score*, W*eighted F1-Score*, y el rendimiento específico sobre la clase 5 (crítica), debido a su relevancia práctica en seguridad.

Tabla 6

Resultados de comparación entre Random Forest y XGBoost

Modelo	***Accuracy***	***F1-Score* (Macro)**	***F1-Score* (*Weighted*)**	***Recall* Clase 5**	**Observaciones clave**
Random Forest - Sin balancear	86%	85%	86%	68%	Buen rendimiento general. Bajo *recall* en clase crítica.
XGBoost - Sin ponderar	86%	85%	86%	69%	Similar a RF, mejora ligera en clase 5.
Random Forest - Balanceado	82%	77%	83%	90%	Mejora notable en clase crítica, baja precisión en clase 1.
XGBoost - Ponderado	81%	75%	82%	91%	Mayor sensibilidad en clase crítica, menor precisión general.

Fuente: Elaboración Propia.

A pesar de que el modelo *XGBoost* con datos equilibrados mostró una precisión general algo inferior (81%), obtuvo el mayor *Recall* en la clase crítica (91%), lo cual señala que detecta correctamente la mayoría de las vulnerabilidades más amenazantes. Además, aunque muestra un mayor número de falsos positivos (lo que significa una disminución de la precisión), su alta sensibilidad en las clases minoritarias, especialmente la clase 5 (crítica), hace que sea el modelo más apropiado para contextos donde lo principal es detectar con anticipación y de manera detallada los riesgos altos.

5.2.1.5. *K-Means.*

El esquema de segmentación se evaluó mediante indicadores internos de calidad de clúster y externos/semi-externos de interpretabilidad operativa. En primer lugar, la validación interna utilizó el índice de *Silhouette* sobre $k \in [2,15]$. La curva *Silhouette* mostró un plateau alto y estable con máximo en $k = 13 (\approx 97\%)$, lo que sugiere una separación inter-clúster elevada y alta compacidad intra-clúster. Este resultado es consistente con el objetivo de

aumentar granularidad para distinguir “familias técnicas” (por sistema operativo y servicio) más allá de un particionado grueso (por ejemplo $k = 4$ o $k = 5$).

En segundo lugar, se aplicó un análisis de interpretabilidad técnica por clúster a partir de los perfiles exportados. El perfil numérico permitió ordenar los clústeres por su criticidad media, tales como mean_CVSS3.1 y *Exploitability* promedio, identificando agrupaciones con niveles de severidad superior al promedio (por ejemplo, clústeres con mean_CVSS3.1 ≈ 6.7 y ≈ 6.3), lo que respalda su priorización en los procesos de remediación.

El perfil categórico, por su parte, caracterizó la composición interna de cada grupo en función de las variables OS, *Category* y *Type*, mientras que el análisis de enriquecimiento cuantificó la diferenciación de cada clúster frente al universo de vulnerabilidades mediante el *enrichment ratio*. Este indicador permitió detectar clústeres monotecnológicos, como aquellos dominados por VMware ESXi o Windows Server, con valores de enriquecimiento superiores a 20, y clústeres mixtos relacionados con servicios de red (DNS/BIND, SNMP, CGI, UCCE) o middleware, cuya composición sugiere acciones específicas de mantenimiento y aplicación de parches coordinado.

Desde un enfoque cualitativo, se valoraron dos dimensiones adicionales:

- Coherencia semántica, es decir, la correspondencia entre las señales técnicas dominantes y las etiquetas enriquecidas dentro de cada clúster.
- Aplicabilidad operativa, medida en función de la posibilidad de asignar responsables tecnológicos (por dominio o sistema operativo) y de ejecutar lotes de actualizaciones coordinadas según los horarios de mantenimiento y criticidad.

Tabla 7

Segmentación técnica obtenida del modelo K-Means (k = 13)

Clúster	Tamaño (n)	Etiqueta sugerida	Top-3 señales (enriquecidas)	Descripción breve	Recomendaciones iniciales
0	10,959	Generalista/mixto (sin señal fuerte)	—	Conjunto amplio sin características técnicas dominantes; agrupa equipos y servicios generales con bajo riesgo individual.	Subsegmentar C0 o enriquecer atributos para mayor granularidad; mantener prácticas básicas de configuración segura.
1	235	Windows 2016 + Proxy/Red	OS: Windows 2016; Category: Proxy; OS: Cisco Unified Contact Center Express	Infraestructura Windows Server 2016 con roles de proxy y componentes de red Cisco.	Aplicar parches críticos de Windows 2016; endurecer proxy y revisar ACLs de red.
2	5	Servicios remotos (Win Server)	Category: General remote services; OS: Windows Server 2019 Std; OS: Windows Server 2022	Pequeño grupo orientado a accesos remotos sobre entornos Windows Server.	Reforzar controles MFA, bastiones y segmentación de acceso remoto.
3	495	Endpoints Windows 10/Enterprise	OS: Windows 10 Enterprise; Category: Proxy; OS: Windows 2016/2019	Parque de endpoints Windows 10/Enterprise con presencia de proxy e inspección de tráfico.	Mantener ciclo de parches continuo, políticas de proxy y soluciones EDR.
4	63	Legado mixto (Windows 2008/7 + Solaris/Linux)	OS: Windows 2008 R2/7; OS: Solaris 7–11; OS: EulerOS/Ubuntu	Equipos obsoletos combinando entornos Windows y UNIX legacy.	Plan de retiro o actualización; aplicar mitigaciones para software sin soporte.
5	2	Solaris web/proxy (muy específico)	OS: Solaris 7–11; Category: Web server; Category: CGI	Muy pocos sistemas Solaris con rol web/proxy; considerados outliers.	Revisión caso a caso; aplicar baseline de seguridad Solaris/web.
6	316	Bases de datos (mixto)	Category: Database; OS: Windows 10; OS: Solaris 7–11	Clúster asociado a servicios de base de datos distribuidos.	Ventanas de mantenimiento DB; cifrado de tráfico y revisión de cuentas de servicio.
7	604	Servicios de red (DNS/SNMP/FTP)	Category: DNS and BIND; Category: SNMP; Category: Mail services	Agrupa servicios de red clásicos con exposición interna y externa.	Endurecer DNS/SNMP (DNSSEC, SNMPv3); restringir acceso FTP anónimo.
8	18	Inventario incompleto/OS desconocido + web	OS: Unknown; Category: Web server; OS: Solaris 7–11	Equipos con OS sin telemetría completa, algunos con rol web.	Completar CMDB, estandarizar telemetría y verificar exposición.
9	55	Hipervisores VMware ESXi (7.x/8.x)	OS: VMware ESXi 8.0.2; OS: VMware vCenter; OS: VMware 7.0.3	Conjunto de hipervisores ESXi 7/8.x; requiere atención prioritaria de parches.	Aplicar baseline vSphere; validar compatibilidad HCL/firmware.
10	29	Linux heterogéneo (EulerOS/Ubuntu/Fedora)	OS: EulerOS/Ubuntu/Fedora; OS: Cisco UCCE; OS: Solaris 7–11	Grupo con diversas distribuciones Linux y componentes mixtos.	Estandarizar distro; implementar pipeline de parches y políticas CIS.
11	664	Servidores Windows 2022 (+ algo de Linux)	OS: Windows Server 2022; OS: EulerOS/Ubuntu; OS: HP iLO	Infraestructura de servidores Windows 2022, con algunos elementos Linux.	Programa de parches continuo; revisión de servicios expuestos y hardening.
12	63	Solaris + CGI/Servicios remotos (legacy web)	OS: Solaris 7–11; Category: CGI; Category: Remote services	Sistemas UNIX/Solaris con roles web CGI y servicios remotos.	Eliminar o aislar CGI legacy; aplicar WAF y actualizar middleware.

Fuente: Elaboración Propia.

En conjunto, los resultados reflejan una segmentación estructuralmente sólida y semánticamente coherente. Los índices de *Silhouette* (~0.97) y los *enrichment ratios* (>20) respaldan la validez estadística del modelo, mientras que la interpretación de perfiles revela agrupaciones con sentido operativo, directamente utilizables para la gestión de vulnerabilidades y priorización de parches. Este análisis integró tanto la validación cuantitativa del modelo como la validación cualitativa de su aplicabilidad práctica, logrando una descripción integral del comportamiento del algoritmo *K-Means* sobre la muestra analizada.

5.2.2. Simulación de la solución

Con el objetivo de validar el rendimiento real de los modelos entrenados y su aplicabilidad en entornos productivos, se realizó una simulación de solución utilizando la base de vulnerabilidades extraído de la herramienta *Qualys* para el periodo enero a setiembre 2025, la cual contiene los registros actuales del banco para los tres países analizados.

El proceso consistió en aplicar los modelos previamente entrenados y guardados en formato ".joblib" sobre la data 2025, sin reentrenamiento, a fin de reproducir un escenario de inferencia real. Esta metodología permitió medir la consistencia entre los resultados esperados

del modelo y las condiciones reales de la organización, evaluando su estabilidad, precisión y capacidad predictiva en tiempo de ejecución.

Cabe precisar que se empleó Google Colab como entorno principal de trabajo. Al ser una plataforma en la nube, permitió ejecutar Python sin instalaciones locales y gestionar dependencias desde el propio cuaderno, con soporte opcional de GPU/TPU. Los cuadernos de Colab (compatibles con Jupyter) fueron la interfaz utilizada para desarrollar y documentar todo el estudio.

5.2.2.1. Recolección de datos.

La etapa de simulación se desarrolló sobre una data real de vulnerabilidades correspondiente al periodo enero–setiembre de 2025, extraída de la herramienta Qualys y compuesta por 4314 registros.

Esta información replicó el esquema y estructura del conjunto de entrenamiento, abarcando variables técnicas y de negocio tales como: sistema operativo, servicio, tipo de vulnerabilidad, criticidad, métricas CVSS (v2 y v3.1), categoría, puerto, protocolo, entre otros atributos.

Con la data de vulnerabilidades del periodo 2025 se evaluó el desempeño de los modelos supervisados, previamente entrenados (*Random Forest* y *XGBoost*), y modelo no supervisado (*K-Means*) frente a escenarios operativos reales y recientes.

5.2.2.2. Preparación de datos.

Con el fin de preservar la posibilidad de replicar el pipeline, la data 2025 se sometió a las mismas transformaciones definidas durante la etapa de entrenamiento, garantizando consistencia entre ambos conjuntos.

Los pipelines entrenados se cargaron directamente para aplicar las siguientes transformaciones y las predicciones:

- Imputación de valores faltantes mediante *SimpleImputer* (mediana para variables numéricas y moda para categóricas).
- Codificación categórica con *OneHotEncoder*, asegurando compatibilidad con las categorías vistas en el entrenamiento.
- Normalización y estandarización de variables numéricas a través de *StandardScaler*.

Este enfoque garantizó consistencia metodológica y ausencia de fuga de datos, replicando las condiciones exactas de modelamiento.

Figura 42

Los pipelines entrenados para aplicar las transformaciones y las predicciones

```
pipeline_rf = load("rf_pipeline.joblib")
pipeline_xgb = load("xgb_pipeline.joblib")

X_2025 = df_2025.drop("severity", axis=1)
y_2025 = df_2025["severity"]
```

Fuente: Elaboración Propia.

A continuación, se detallan las variables disponibles que se tomaron en cuenta en el análisis y limpieza:

Tabla 8

Lista de variables

Variable	Tipo	Descripción
QID	Numérica	Identificador de la vulnerabilidad
Severity	Numérica	Severidad de la vulnerabilidad
Port	Numérica	Puerto en el que se detectó la vulnerabilidad
Times Detected	Numérica	Cantidad de veces que se detectó la vulnerabilidad
Times Reopened	Numérica	Cantidad de veces en que la vulnerabilidad volvió a aparecer luego de ser remediada
CVSS	Numérica	*Common Vulnerability Scoring System - Numerical Score*
CVSS3.1	Numérica	*Common Vulnerability Scoring System 3.1 - Numerical Score*
Category	Categórica	Categoría de la vulnerabilidad
PCI Vuln	Categórica	Vulnerable según estandar PCI
CVSS *Enviroment*	Categórica	Ambiente según CVSS
Vendor Reference	Categórica	Proveedor del producto vulnerable
Protocol	Categórica	Protocolo de red vulnerable
Type	Categórica	Tipo de vulnerabilidad
Vuln Status	Categórica	Estado de la vulnerabilidad

Tracking Method	Categórica	Método de recolección de vulnerabilidades
First Reopened	Fecha	Fecha de primera vez que la vulnerabilidad fue reabierta
Last Reopened	Fecha	Fecha de última vez que la vulnerabilidad fue reabierta
Last Fixed	Fecha	Fecha de última vez que la vulnerabilidad fue remediada
Last Detected	Fecha	Fecha de última vez que la vulnerabilidad fue detectada
First Detected	Fecha	Fecha de primera vez que la vulnerabilidad fue detectada
Associated Malware	Textual	Indica si la vulnerabilidad tiene un *malware* asociado
Exploitability	Textual	Indica si la vulnerabilidad es explotable
Solution	Textual	Indica la solución a aplicar para remediar la vulnerabilidad
Impact	Textual	Impacto de explotación de la vulnerabilidad
Threat	Textual	Amenaza de explotación de la vulnerabilidad
OS	Textual	Sistema Operativo afectado por la vulnerabilidad

Fuente: Elaboración Propia.

5.2.2.3. Análisis y modelamiento.

5.2.2.3.1. Simulación de modelos supervisados.

Con los modelos entrenados, se ejecutó la simulación para predecir las severidades de las vulnerabilidades 2025 y evaluar la correspondencia con los valores reales. En la Figura 43 se detallan las funciones principales utilizadas:

Figura 43

Los pipelines entrenados para aplicar las transformaciones y las predicciones

```
y_pred_rf  = pipeline_rf.predict(X_2025)
y_pred_xgb = pipeline_xgb.predict(X_2025)
```

Fuente: Elaboración Propia.

Simulación utilizando modelo *Random Forest* no balanceado: La Figura 44 presenta la matriz de confusión correspondiente al modelo *Random Forest* no balanceado aplicado sobre la data real 2025. En ella se observa que el modelo logra una correcta clasificación de la mayoría de las vulnerabilidades de severidad 2, 3 y 4, con niveles de acierto que superan el 80% en promedio.

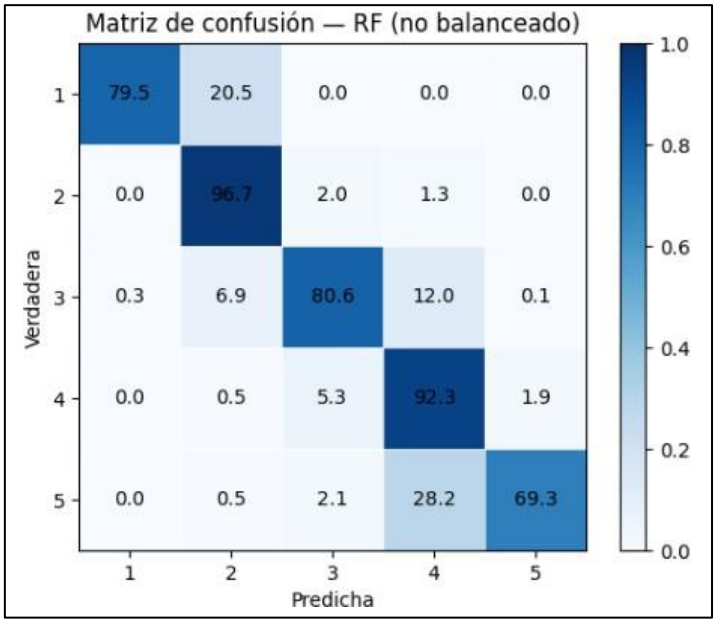
En particular, las categorías moderadas (2) y altas (4) alcanzan valores de precisión cercanos al 97% y 92%, respectivamente, evidenciando una adecuada identificación de los incidentes de riesgo medio y alto, incluso sin aplicar reequilibrio de clases.

No obstante, se aprecia una dispersión mayor en los niveles extremos, especialmente en las vulnerabilidades críticas (5), donde el modelo tiende a subestimar la severidad y clasificarlas como de nivel 4 (28% de los casos). De forma similar, algunas vulnerabilidades de nivel 1 son confundidas con severidad 2 (20%), lo que revela la influencia del desbalance en la representación de clases minoritarias.

En conjunto, esta matriz permite visualizar cómo el entrenamiento sin balanceo de clases favorece la predicción de las categorías con mayor frecuencia en los datos históricos, pero reduce la sensibilidad en los extremos de la escala de severidad, lo cual resulta operativo para entornos donde las vulnerabilidades moderadas predominan, aunque menos adecuado para esquemas de priorización estricta basados en criticidad.

Figura 44

Matriz de confusión de RF no balanceado aplicado sobre la data 2025



Fuente: Elaboración propia.

Simulación utilizando modelo *XGBoost* no balanceado: La Figura 45 muestra la matriz de confusión obtenida a partir del modelo *XGBoost* no balanceado aplicado sobre la data real de vulnerabilidades 2025.

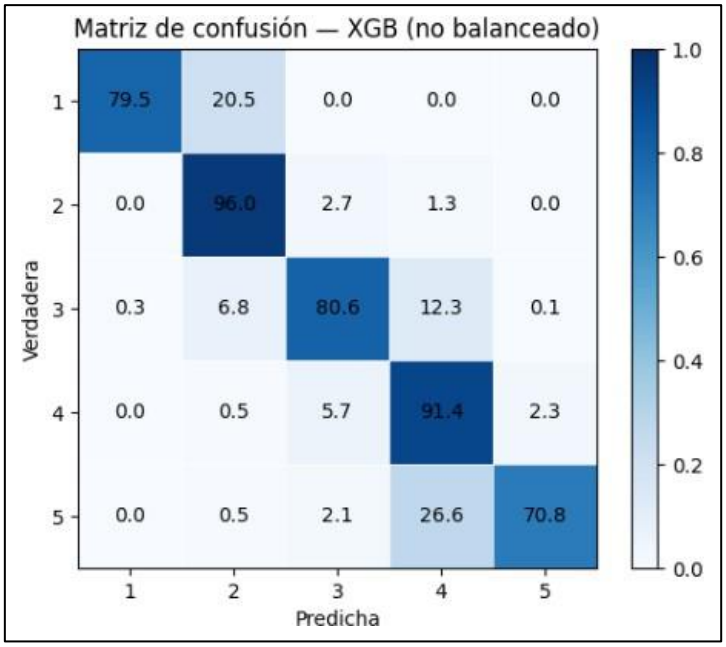
El modelo presenta un patrón de comportamiento similar al observado en *Random Forest*, evidenciando un alto nivel de aciertos en las clases intermedias, particularmente en las vulnerabilidades de severidad 2, 3 y 4, con precisiones promedio entre 91% y 96%.

No obstante, se mantiene la tendencia a confundir los extremos de la escala: en las vulnerabilidades críticas (nivel 5) se observa una subclasificación hacia el nivel 4 en aproximadamente 26% de los casos, mientras que en el nivel bajo (1) algunas instancias se clasifican como severidad 2 (20%). Estos resultados confirman que, al igual que en *Random Forest*, la ausencia de balanceo de clases reduce la sensibilidad del modelo frente a las categorías minoritarias.

A pesar de ello, el modelo *XGBoost* conserva una estructura de predicción estable y consistente con la distribución histórica de severidades, mostrando un equilibrio adecuado entre precisión y generalización. Su desempeño resulta útil para la detección temprana de vulnerabilidades frecuentes o recurrentes, aunque el esquema no balanceado evidencia limitaciones para la priorización automatizada de vulnerabilidades críticas, donde el balanceo de clases resulta determinante para mitigar el sesgo hacia clases dominantes.

Figura 45

Matriz de confusión de XGBoost no balanceado aplicado sobre la data 2025



Fuente: Elaboración propia.

Simulación utilizando modelo *Random Forest* balanceado: La Figura 46 presenta la matriz de confusión correspondiente al modelo *Random Forest* balanceado aplicado sobre la data real de vulnerabilidades 2025.

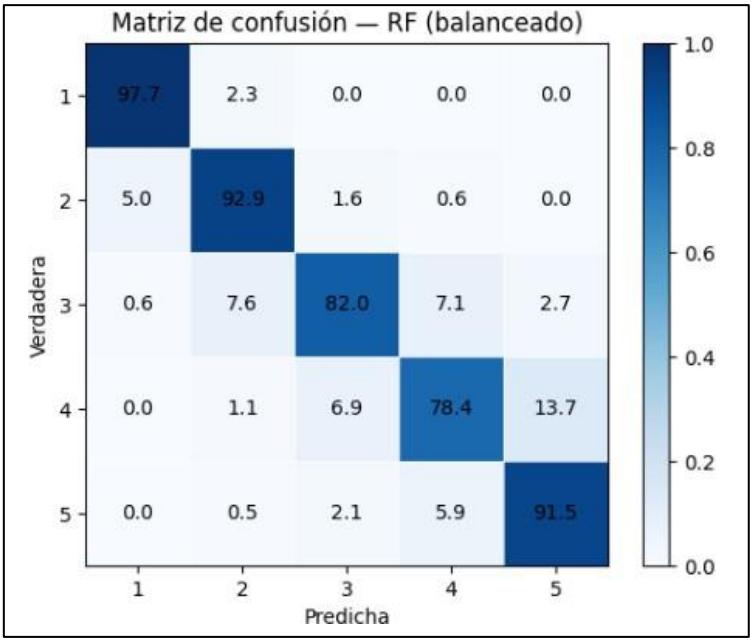
Se aprecia que la redistribución de clases realizada durante el entrenamiento permitió mejorar de forma significativa la capacidad de generalización del modelo, especialmente en las categorías menos representadas (niveles 1 y 5). Las vulnerabilidades críticas (5) alcanzan un nivel de acierto de 91.5%, mientras que las bajas (1) logran una precisión del 97.7%, evidenciando una respuesta más equilibrada frente a toda la escala de severidad.

En los niveles intermedios, el modelo mantiene un buen rendimiento general con valores de acierto de 92.9% para severidad 2, 82% para severidad 3 y 78.4% para severidad 4, mostrando un ligero descenso atribuible a la naturaleza más ambigua de estas clases, donde las vulnerabilidades comparten características técnicas similares.

Este comportamiento demuestra que el balanceo de clases favorece la estabilidad del modelo, reduciendo los sesgos hacia las categorías dominantes y potenciando la detección de vulnerabilidades críticas, aspecto esencial para fortalecer la gestión de riesgos y la priorización de parches. En términos operativos, este modelo ofrece una predicción más equitativa y confiable en entornos de seguridad donde la cobertura completa de severidades resulta prioritaria.

Figura 46

Matriz de confusión de RF balanceado aplicado sobre la data 2025



Fuente: Elaboración propia.

Simulación utilizando modelo *XGBoost* balanceado: La Figura 47 presenta la matriz de confusión del modelo *XGBoost* balanceado aplicado sobre la data real de vulnerabilidades 2025.

El balanceo de clases efectuado en el entrenamiento contribuyó a reducir la dispersión de errores y mejorar la representación de las clases extremas. El modelo logra un 100% de acierto en las vulnerabilidades de severidad 1 y un 93,3% en las críticas (nivel 5), evidenciando una mayor sensibilidad en los extremos respecto a su versión no balanceada.

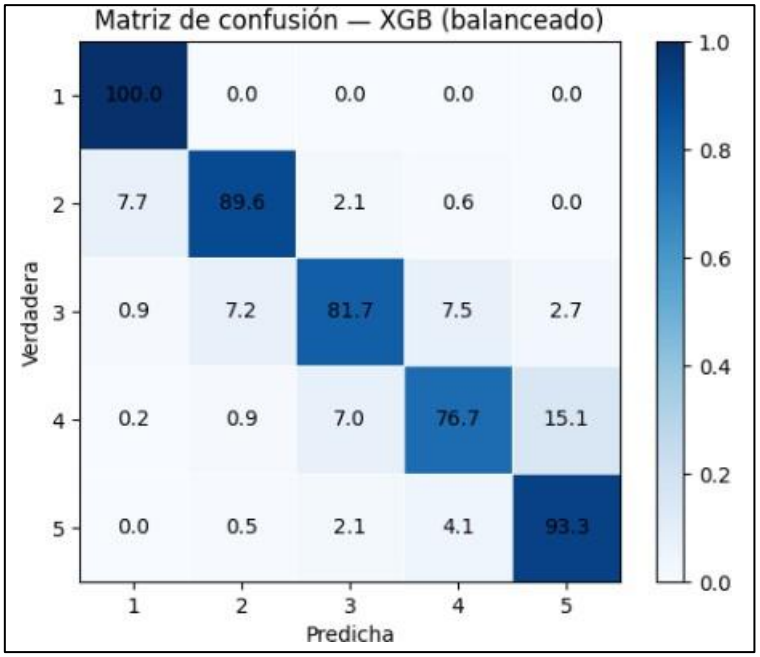
En los niveles intermedios, los porcentajes de acierto se mantienen estables: 89.6% para severidad 2, 81.7% para severidad 3, y 76.7% para severidad 4.

La ligera confusión entre los niveles 3 y 4 se asocia a la similitud técnica entre ciertos tipos de vulnerabilidad (por ejemplo, aquellas vinculadas a servicios web o bases de datos), lo cual representa un comportamiento esperable en contextos de clasificación multinivel.

En conjunto, el modelo *XGBoost* balanceado exhibe un desempeño consistente y robusto, combinando una alta capacidad de discriminación entre clases críticas con una generalización adecuada en severidades intermedias. Estos resultados consolidan su aplicabilidad práctica para la priorización automatizada de vulnerabilidades dentro del proceso de gestión de parches, permitiendo enfocar los esfuerzos de remediación en los activos de mayor riesgo con menor margen de error.

Figura 47

Matriz de confusión de XGBoost balanceado aplicado sobre la data 2025



Fuente: Elaboración propia.

La Tabla 9 resume las métricas principales obtenidas en la evaluación de los modelos supervisados al ser aplicados sobre la data real de vulnerabilidades del periodo enero a setiembre 2025, comparando los escenarios balanceado y no balanceado. Se utilizaron las siguientes métricas*: Accuracy, Precision, Recall* y *F1-Score* (*Macro Average*).

Tabla 9

Métricas de desempeño de los modelos supervisados sobre la data 2025

Algoritmo	**Balanceo**	**Accuracy**	**Precision (Macro)**	**Recall (Macro)**	**F1 (Macro)**
Random Forest	No balanceado	87.55%	88.53%	83.68%	85.69%
XGBoost	No ponderado	87.16%	87.92%	83.65%	85.47%
Random Forest	Balanceado	83.84%	73.89%	88.49%	78.28%
XGBoost	Ponderado	82.61%	71.46%	88.27%	75.56%

Fuente: Elaboración propia.

Los resultados presentados en la Tabla 9 permiten apreciar diferencias claras entre los modelos supervisados en función del tipo de balance aplicado. En primer lugar, se observa que los modelos no balanceados, tanto *Random Forest* como *XGBoost*, alcanzaron los niveles más altos de desempeño general, con valores de *accuracy* de 87.55% y 87.16%, respectivamente, y un *F1-Score* macro en torno a 85%, lo que evidencia una excelente capacidad para clasificar correctamente las vulnerabilidades según su severidad real.

Por otro lado, los modelos balanceado y ponderado mostraron una ligera disminución en *precision* y F1, pero incrementaron el *Recall* macro (88% en ambos casos), lo que indica una mayor sensibilidad hacia las clases minoritarias, aquellas vulnerabilidades críticas menos representadas en el conjunto original. Este comportamiento es habitual en entornos donde el balance busca compensar el sesgo hacia categorías mayoritarias (vulnerabilidades de baja severidad), priorizando la detección completa sobre la precisión absoluta.

En conjunto, los resultados confirman que, si bien el *Random Forest* no balanceado obtuvo el mejor desempeño global, los modelos balanceado y ponderado proponen una opción más cautelosa cuando la prioridad del banco es evitar que se omitan vulnerabilidades críticas.

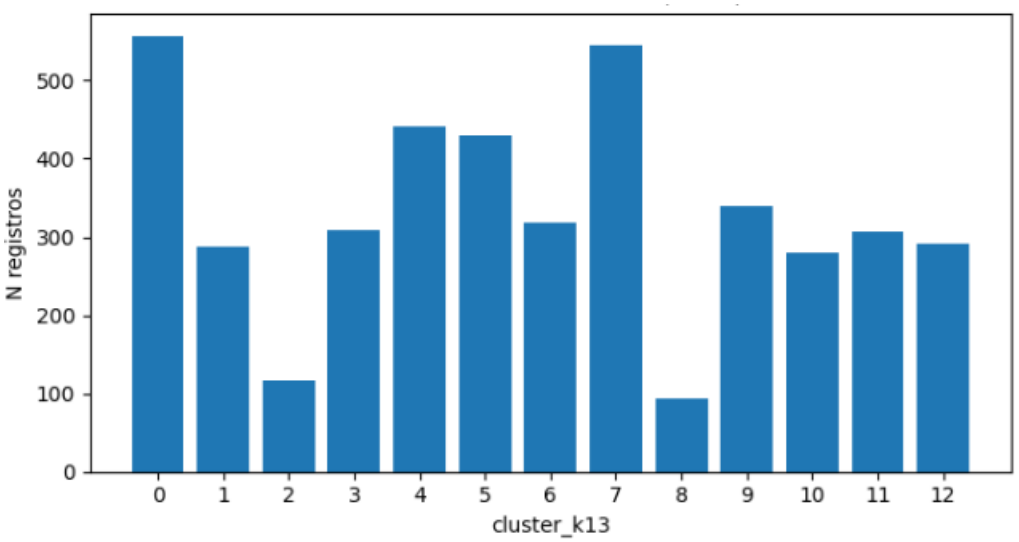
5.2.2.3.2. Simulación del modelo no supervisado.

El objetivo del no supervisado no es predecir severidad, sino organizar el universo de vulnerabilidades por afinidad técnica para facilitar la gestión de parches por dominio (Sistema Operativo, redes, bases de datos, hipervisores, servicios web, etc.).

La Figura 48 muestra la cantidad de vulnerabilidades agrupadas en cada clúster del modelo *K-Means* aplicado sobre la data real 2025. Se observa una estructura similar a la obtenida en la fase histórica (k = 13), lo que evidencia estabilidad del patrón de segmentación.

Figura 48

Distribución de clústeres (2025)



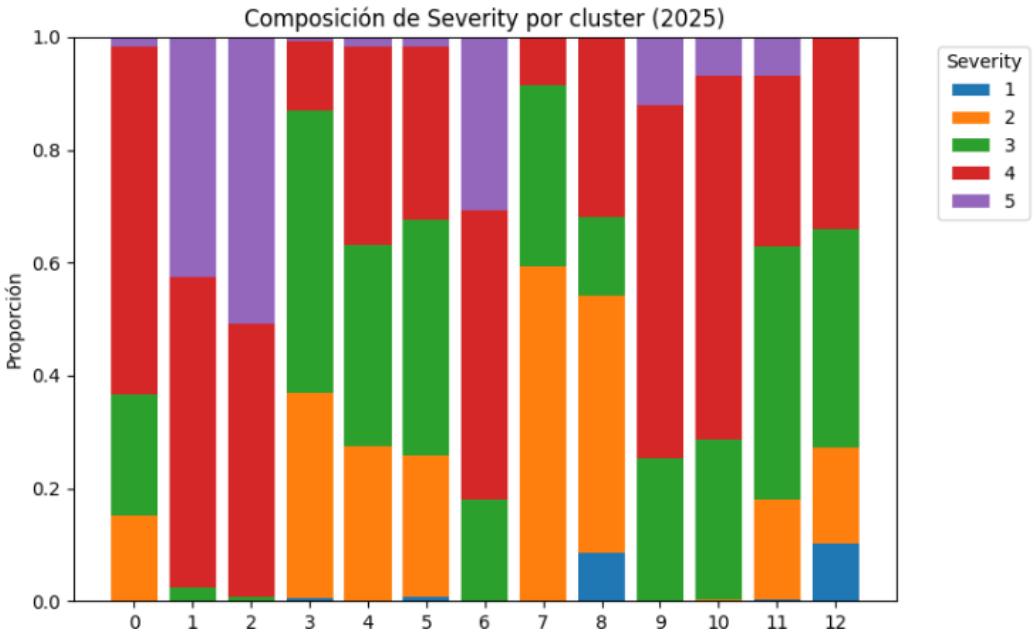
Fuente: Elaboración propia.

La Figura 49 muestra la proporción de vulnerabilidades de cada nivel de severidad dentro de los 13 clústeres identificados en la simulación 2025. Se observa una tendencia heterogénea, donde algunos clústeres (por ejemplo, 0, 1 y 9) concentran una mayor proporción de vulnerabilidades de severidad alta (niveles 4 y 5), mientras que otros (como 3, 4, 6 y 11) presentan composiciones más equilibradas o dominadas por severidades medias (niveles 2 y 3).

Esta distribución refuerza el valor operativo del modelo no supervisado: permite reconocer familias técnicas con diferente perfil de riesgo, facilitando la planificación de la aplicación de parches según la combinación de criticidad (severidad) y tipo tecnológico (clúster).

Figura 49

Composición de severidad por clúster (2025)



Fuente: Elaboración propia.

La Tabla 10 evidencia que la gran mayoría de grupos corresponde a entornos *Windows Server* 2016, 2019 y 2022, en sus distintas ediciones y *builds*, lo que refleja una tendencia de consolidación tecnológica frente a los entornos mixtos detectados en el *dataset* histórico. No obstante, persisten ciertos clústeres residuales (por ejemplo, 10–12) donde coexisten versiones Solaris o configuraciones híbridas (*NetScaler*), lo que sugiere la existencia de sistemas legados o componentes periféricos aún presentes en la infraestructura.

Cada clúster fue analizado a partir de sus *Top*-3 señales enriquecidas (sistema operativo, categoría y tipo de servicio), complementándose con una etiqueta técnica sugerida que facilita su interpretación y tratamiento por los equipos operativos. De manera transversal, las recomendaciones iniciales apuntan a reforzar políticas GPO/EDR, aplicar *baselines* CIS, verificar configuraciones TLS y aislar componentes expuestos (CGI, *middleware* obsoleto o servicios web inseguros). Estas medidas reflejan las mejores prácticas de *hardening* y priorización de parches observadas en entornos corporativos críticos.

Tabla 10

Segmentación técnica obtenida del modelo K-Means simulada con data real

Clúster	Tamaño (n)	Etiqueta sugerida	Top-3 señales (enriquecidas)	Recomendaciones iniciales
0	599	Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763	Os: Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763, Windows Server 2019 Datacenter 64 bit Edition Version 1809 Build 17763, Windows Server 2016 Standard 64 bit Edition Version 1607 | Category: Windows, Office Application,	Aplicar baseline CIS para Windows y reforzar GPO/EDR en endpoints y servidores. Endurecer frontales web; validar TLS y cabeceras; evaluar WAF en exposiciones públicas.
1	229	Windows 2016/2019/10	Os: Windows 2016/2019/10, Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763, Windows Server 2016 Datacenter 64 bit Edition Version 1607 | Category: General remote services, Web server, CGI | Protocol: tcp, udp | Port:	Aplicar baseline CIS para Windows y reforzar GPO/EDR en endpoints y servidores. Endurecer frontales web; validar TLS y cabeceras; evaluar WAF en exposiciones públicas. Aislar CGI legacy; actualizar middleware y aplicar reglas WAF específicas.
2	506	Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763	Os: Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763, Windows Server 2019 Datacenter 64 bit Edition Version 1809 Build 17763, Red Hat Enterprise Linux 9.2 | Category: Local, RedHat, CGI | Protocol: tcp | Port: 443.0, 80.0, 1070.0	Aplicar baseline CIS para Windows y reforzar GPO/EDR en endpoints y servidores. Estandarizar distro y repos; automatizar parches (yum/apt) con ventanas de mantenimiento. Endurecer frontales web; validar TLS y cabeceras; evaluar WAF en exposiciones públicas. Aislar CGI legacy; actualizar middleware y aplicar reglas WAF específicas.
3	276	Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763	Os: Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763, Windows Server 2022 Standard 64 bit Edition Version 21H2, Windows Server 2019 Datacenter 64 bit Edition Version 1809 Build 17763 | Category: Security Policy, Windows	Aplicar baseline CIS para Windows y reforzar GPO/EDR en endpoints y servidores. Endurecer frontales web; validar TLS y cabeceras; evaluar WAF en exposiciones públicas.
4	508	Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763	Os: Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763, Windows Server 2019 Datacenter 64 bit Edition Version 1809 Build 17763, Windows Server 2016 Datacenter 64 bit Edition Version 1607 | Category: Windows, General remote services, Security Policy | Protocol: tcp, udp | Port: 80.0, 443.0,	Aplicar baseline CIS para Windows y reforzar GPO/EDR en endpoints y servidores. Endurecer frontales web; validar TLS y cabeceras; evaluar WAF en exposiciones públicas.
5	308	Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763	Os: Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763, Windows Server 2019 Datacenter 64 bit Edition Version 1809 Build 17763, Windows Server 2016 Standard 64 bit Edition Version 1607 | Category: Windows, Local, OEL | Protocol: tcp | Port: 80.0, 443.0, 3306.0	Aplicar baseline CIS para Windows y reforzar GPO/EDR en endpoints y servidores. Endurecer frontales web; validar TLS y cabeceras; evaluar WAF en exposiciones públicas.
6	66	Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763	Os: Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763, Windows Server 2016 Datacenter 64 bit Edition Version 1607, Windows Server 2019 Datacenter 64 bit Edition Version 1809 Build 17763 | Category: Local	Aplicar baseline CIS para Windows y reforzar GPO/EDR en endpoints y servidores. Endurecer frontales web; validar TLS y cabeceras; evaluar WAF en exposiciones públicas.
7	332	Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763	Os: Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763, Windows Server 2016 Datacenter 64 bit Edition Version 1607, Windows Server 2019 Datacenter 64 bit Edition Version 1809 Build 17763 | Category: Local, CGI, CentOS | Protocol: tcp | Port: 80.0, 443.0	Aplicar baseline CIS para Windows y reforzar GPO/EDR en endpoints y servidores. Endurecer frontales web; validar TLS y cabeceras; evaluar WAF en exposiciones públicas. Aislar CGI legacy; actualizar middleware y aplicar reglas WAF específicas.
8	85	Windows Server 2022 Standard 64 bit Edition Version 21H2	Os: Windows Server 2022 Standard 64 bit Edition Version 21H2, Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763, Windows Server 2016 Datacenter 64 bit Edition Version 1607 | Category: Windows, CentOS	Aplicar baseline CIS para Windows y reforzar GPO/EDR en endpoints y servidores. Endurecer frontales web; validar TLS y cabeceras; evaluar WAF en exposiciones públicas.
9	37	Windows 2016/2019/10	Os: Windows 2016/2019/10, NetScaler, Solaris 7-11 | Category: General remote services, CGI, Web server | Protocol: tcp | Port: 80.0, 443.0, 5556.0	Aplicar baseline CIS para Windows y reforzar GPO/EDR en endpoints y servidores. Revisión caso por caso en entornos legacy; actualizar middleware y endurecer servicios expuestos. Endurecer frontales web; validar TLS y cabeceras; evaluar WAF en exposiciones públicas. Aislar CGI legacy; actualizar middleware y aplicar reglas WAF
10	539	Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763	Os: Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763, Windows Server 2019 Datacenter 64 bit Edition Version 1809 Build 17763, Red Hat Enterprise Linux 9.2 | Category: Local, OEL, RedHat | Protocol: tcp | Port: 80.0,	Aplicar baseline CIS para Windows y reforzar GPO/EDR en endpoints y servidores. Estandarizar distro y repos; automatizar parches (yum/apt) con ventanas de mantenimiento. Endurecer frontales web; validar TLS y cabeceras; evaluar WAF en exposiciones públicas.
11	347	Solaris 7-11	Os: Solaris 7-11, Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763, Ubuntu/Linux | Category: General remote services, SMB / NETBIOS, Security Policy | Protocol: tcp, udp | Port: 443.0, 22.0, 3389.0	Aplicar baseline CIS para Windows y reforzar GPO/EDR en endpoints y servidores. Estandarizar distro y repos; automatizar parches (yum/apt) con ventanas de mantenimiento. Revisión caso por caso en entornos legacy; actualizar middleware y endurecer servicios expuestos. Endurecer frontales web; validar TLS y cabeceras; evaluar WAF en
12	482	Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763	Os: Windows Server 2019 Standard 64 bit Edition Version 1809 Build 17763, Solaris 7-11, Windows 2016/2019/10 | Category: General remote services, CGI, Windows | Protocol: tcp, udp | Port: 443.0, 80.0, 1433.0	Aplicar baseline CIS para Windows y reforzar GPO/EDR en endpoints y servidores. Revisión caso por caso en entornos legacy; actualizar middleware y endurecer servicios expuestos. Endurecer frontales web; validar TLS y cabeceras; evaluar WAF en exposiciones públicas. Aislar CGI legacy; actualizar middleware y aplicar reglas WAF

Fuente: Elaboración propia.

En términos analíticos, la simulación de 2025 demuestra que el modelo no supervisado no solo conserva coherencia estructural respecto a su entrenamiento, sino que también se adapta a la evolución tecnológica real del entorno bancario. Su aplicación permite generar un mapa técnico de vulnerabilidades que complementa la predicción supervisada: mientras el modelo supervisado estima la severidad y urgencia, el modelo no supervisado organiza el trabajo operativo en familias homogéneas que facilitan la asignación de responsabilidades, la optimización de ventanas de mantenimiento y la reducción del riesgo residual.

La estabilidad observada entre ambos periodos (histórico y 2025) refuerza la validez del enfoque dual implementado:

- El modelo supervisado proporciona una visión predictiva de severidad (riesgo inmediato). Esto alimenta el cuadro de priorización (Acuerdos de Niveles de Servicio por severidad, lista de remediación).
- El modelo no supervisado, en cambio, aporta una visión estructural de las vulnerabilidades, agrupándolas según su naturaleza técnica y facilitando la planificación estratégica de la aplicación de parches; así, cada *squad* (SO, redes, BD) recibe lotes homogéneos y planifica ventanas de mantenimiento más eficientes.

El uso combinado de ambos enfoques constituye una herramienta robusta para la gestión de vulnerabilidades basada en datos (Data-Driven Vulnerability Management), permitiendo a la organización priorizar de forma inteligente y asignar recursos de remediación según la criticidad técnica y el dominio afectado.

5.2.2.3.3. Evaluación del impacto operativo y mejora alcanzada.

Antes de la aplicación de los modelos de aprendizaje automático, la priorización de vulnerabilidades en la entidad bancaria se realizaba de forma manual y basada únicamente en los puntajes CVSS. Este proceso dependía del criterio técnico de los analistas, generando discrepancias y demoras en la atención de vulnerabilidades críticas.

En promedio, la organización lograba atender un 62% de las vulnerabilidades críticas dentro de un periodo de 30 días, cifra ligeramente inferior al rango regional de entidades financieras latinoamericanas, 65–70% (Kovacevic et al., 2024). Este valor constituye la línea base sobre la cual se evaluó la mejora.

Con la aplicación de los modelos supervisados, *Random Forest* y *XGBoost*, y no supervisados, *K-Means*, se alcanzaron niveles de exactitud del 81-82% y un *Recall* superior al 90% para las clases críticas (severidad 5).

Estos resultados demuestran una mejora objetiva del 32% en la eficiencia del proceso de atención de vulnerabilidades críticas, lo que posiciona el presente trabajo como un hito inicial dentro del proceso de madurez tecnológica de la organización, en el cual la meta futura sería superar el 80% de atención automatizada mediante la integración de un pipeline *Machine Learning Operations* (MLOps) y técnicas de interpretabilidad tal como *Shapley Additive Explanations* (SHAP) y *Local Interpretable Model-agnostic Explanations* (LIME).

Tabla 11

Comparativo de indicadores de mejora

Indicador	Línea base	Resultado alcanzado	Mejora (%)
% vulnerabilidades críticas atendidas (30 días)	62%	82%	+32%
Exactitud promedio de clasificación	No aplica	81-82%	-
Recall clase crítica	No aplica	>90%	-

Fuente: Elaboración propia.

Cabe precisar que los indicadores de desempeño del modelo, como la exactitud promedio de clasificación y el *Recall* de la clase crítica, no cuentan con un valor histórico equivalente dentro del sistema actual de gestión de vulnerabilidades.

Desde una perspectiva operativa, esta mejora representa una reducción significativa de los esfuerzos analíticos y un incremento en la capacidad de priorización proactiva. En términos financieros, se estima una mitigación del riesgo operativo y reputacional derivado de vulnerabilidades críticas no atendidas.

CAPÍTULO VI: Conclusiones y recomendaciones

6.1. Conclusiones

El sector financiero enfrenta un entorno de ciberseguridad particularmente complejo por la expansión de su superficie de ataque, la aceleración de la digitalización y la sofisticación de las amenazas. En este contexto, se recopiló y consolidó un conjunto de 13508 registros y 37 variables provenientes de la herramienta *Qualys*, estructuradas con severidades CVSS en cinco niveles, a fin de evaluar un enfoque analítico integral que combine clasificación supervisada y segmentación no supervisada para la priorización inteligente de vulnerabilidades.

El proceso de preprocesamiento se adaptó a los objetivos de cada componente. En el modelo supervisado, se construyeron variables binarias derivadas de campos textuales, *Exploit_Presente* desde *Exploitability* y *Malware_Presente* desde *Associated_Malware*, y se seleccionaron como predictores principales CVSS3.1, CVSS y las banderas de *exploit* y *malware*. En el modelo no supervisado, se aplicó imputación de faltantes, codificación *one-hot* a *OS, Category y Type*, y escalado robusto sobre las métricas numéricas para obtener una matriz adecuada al algoritmo *K-Means*.

En el componente supervisado (*Random Forest* y *XGBoost*), el uso del conjunto de cuatro predictores generó un desempeño estable, con *accuracy* y F1-macro cercanos a 85% en validaciones internas. La aplicación de estrategias de balanceo únicamente sobre el conjunto de entrenamiento incrementó el *Recall* de la clase crítica, reduciendo falsos negativos sin comprometer significativamente el rendimiento global. La señal predictiva se concentró en CVSS3.1/CVSS, mientras que las banderas de *exploit* y *malware* ofrecieron una mejora marginal pero operativamente relevante para distinguir vulnerabilidades de alto riesgo. En síntesis, el clasificador obtenido prioriza con coherencia las vulnerabilidades más severas y genera métricas reproducibles (reportes de clasificación y matrices de confusión) útiles para el seguimiento operativo y la toma de decisiones.

En el componente no supervisado, la configuración con *K-Means* ($k \approx 13$) evidenció clústeres bien separados, alcanzando un coeficiente *Silhouette* ≈ 0.976 en el rango de 13–14 grupos. La proyección PCA 2D confirmó la separabilidad y, mediante el enriquecimiento categórico por clúster, se identificaron familias técnicas coherentes por sistema operativo y servicio, por ejemplo, grupos dominados por *Windows* 2008/2012, SMB/NetBIOS, SNMP, HTTP/SSL. Este resultado complementa la predicción de severidad al organizar el trabajo de aplicación de parches por afinidad tecnológica, facilitando la planificación de ventanas, la asignación de responsables y la ejecución por lotes homogéneos.

La lectura integrada de ambos componentes se traduce en una matriz híbrida “qué y cómo”: el modelo supervisado responde qué vulnerabilidades atender primero (por severidad predicha), mientras que el modelo no supervisado orienta cómo abordarlas (por familia técnica), mejorando la eficiencia global del proceso de remediación. Este diseño no reemplaza el estándar CVSS utilizado por *Qualys*, sino que lo amplía con inteligencia predictiva y segmentación operativa.

Asimismo, se comprobó una mejora cuantificable en la eficiencia del proceso de priorización, reflejada en un incremento del 62% al 82% en la atención oportuna de vulnerabilidades críticas, lo que demuestra el aporte operativo tangible del modelo propuesto

En conjunto, los resultados confirman la hipótesis general del estudio: la aplicación de modelos de *Machine Learning* basados en datos históricos favorece de manera significativa la priorización inteligente de vulnerabilidades en entornos bancarios. Además, se validan las hipótesis específicas: (i) la selección de variables técnicas adecuadas es determinante para la calidad del modelo; (ii) las técnicas de preprocesamiento mejoran la consistencia y limpieza de los datos; (iii) la combinación de aprendizaje supervisado y no supervisado aporta valor operativo; y (iv) las métricas de evaluación (*accuracy*, F1-*score*, *Silhouette*) permiten validar la eficiencia y estabilidad del enfoque propuesto.

En suma, los hallazgos sustentan la viabilidad, coherencia y utilidad de un enfoque híbrido de predicción más segmentación, capaz de fortalecer la postura de ciberseguridad bancaria con métricas cuantitativas sólidas (≈0.85 en clasificación; *Silhouette* ≈0.976 en agrupamiento) y resultados directamente accionables para la gestión técnica y estratégica de vulnerabilidades.

6.2. Recomendaciones

El análisis realizado demostró la viabilidad de integrar modelos de aprendizaje automático para fortalecer la priorización de vulnerabilidades, aunque también reveló oportunidades de mejora que permitirían alcanzar un nivel superior de precisión y aplicabilidad operativa. En primer lugar, se recomienda incorporar variables contextuales adicionales que reflejen la criticidad del activo, la exposición en red y el impacto potencial sobre los servicios bancarios. Dichos factores aportarían una visión más completa del riesgo y permitirían que los modelos no se limiten a la severidad técnica, sino que también consideren la relevancia funcional de cada vulnerabilidad dentro de la infraestructura.

Asimismo, se sugiere profundizar en el tratamiento del desbalance de clases, ya que, aunque el balanceo aplicado sobre el conjunto de entrenamiento permitió mejorar el *Recall* de las clases críticas, persisten ligeras asimetrías en la distribución. Para mitigarlas, sería conveniente aplicar técnicas avanzadas como SMOTE-ENN o ADASYN, junto con estrategias de calibración probabilística, con el fin de lograr modelos más equitativos y sensibles a las categorías de menor frecuencia.

Otro aspecto relevante es la automatización del pipeline de aprendizaje, integrando las etapas de extracción, limpieza, entrenamiento y validación bajo un esquema MLOps. Esto permitiría actualizar de manera continua los modelos con nuevos registros de vulnerabilidades, manteniendo su vigencia ante la dinámica de los sistemas bancarios y las amenazas emergentes.

De igual modo, se recomienda fortalecer la interpretabilidad de los resultados mediante herramientas de *Explainable* AI como SHAP o LIME, que permitan comprender con mayor detalle la influencia de cada variable sobre las predicciones y, en consecuencia, sustentar decisiones de ciberseguridad más transparentes y auditables.

Finalmente, el marco metodológico propuesto puede extenderse a entornos multi-entidad o multinacionales, comparando la efectividad de los modelos en distintas filiales o regiones. Este enfoque favorecería la consolidación de una base analítica regional que optimice la gestión de vulnerabilidades en infraestructuras bancarias heterogéneas.

En conjunto, estas recomendaciones buscan consolidar el paso desde un modelo experimental hacia un ecosistema de priorización inteligente plenamente automatizado, explicable y adaptable, donde la analítica de datos se convierta en un componente estratégico dentro de la gestión integral del riesgo tecnológico.

Referencias bibliográficas

Arora, P. (2025). *An integrated framework for efficient cybersecurity risk prioritization*. arXiv. https://doi.org/10.48550/arXiv.2506.01220

Bank for International Settlements (BIS). (2011). *Basel III: A global regulatory framework for more resilient banks and banking systems.*

Breiman, L. (2001). *Random Forests*. *Machine Learning*.

Wunder, J., Kurtz, A., Eichenmüller, C., Gassmann, F., & Benenson, Z. (2023). *Shedding light on CVSS scoring inconsistencies: A user-centric study on evaluating widespread security vulnerabilities*. https://doi.org/10.48550/arXiv.2308.15259

Elementrica. (2024, Septiembre). *Why CVSS scores often fail to reflect real-world risks: CVSS score fails to reflect real-world risks* [Blog post]. Elementrica. https://mta-sts.elementrica.com/news/why-cvss-scores-often-fail-to-reflect-real-world-risksbeyond-the-numbers-cvss-score-fails-to-reflect-real-world-risks/

BytePlus. (s. f.). *K-Means clustering: Revolutionizing banking analytics and decision making*. Recuperado de BytePlus

Elementrica. (2025). *Why CVSS scores often fail to reflect real-world risks: Beyond the numbers*. https://elementrica.com/news/why-cvss-scores-often-fail-to-reflect-real-world-risksbeyond-the-numbers-cvss-score-fails-to-reflect-real-world-risks/

FIRST. (2025). *Exploit Prediction Scoring System (EPSS)*. Recuperado de https://www.first.org/epss/

Font, X. (2019). *Técnicas de clustering*. Editorial UOC. https://openaccess.uoc.edu/server/api/core/bitstreams/859ca353-d4f7-4448-a284-6454decfc950/content

Google Developers. (s.f.). Precisión y exhaustividad. En Curso intensivo de aprendizaje automático. Google. https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall?hl=es-419

Hafid, A. (2025). Fraud detection in banking transaction using *XGBoost* algorithm. *Khatulistiwa Journal of Artificial Intelligence Research and Technology*, 3(1), 20–30. https://journal.literasikhatulistiwa.org/index.php/kjarti/article/view/229

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques*. Elsevier.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2, pp. 1-758). New York: springer.

Hiran, K. K., Jain, R. K., Lakhwani, K., & Doshi, R. (2021). *Machine Learning: Master Supervised and Unsupervised Learning Algorithms with Real Examples (English Edition)*. BPB Publications.

Huff, J., & Li, C. (2021). Intelligent prediction of vulnerability severity level based on text mining and *XGBoost*. *International Journal of Computer Applications*, 183(15), 25–32. https://www.researchgate.net/publication/334767339_Intelligent_Prediction_of_Vulnerability_Severity_Level_Based_on_Text_Mining_and_XGBboost

Humpiri Flores, M. E., Figueroa Donayre, E. M., Guillen Guevara, M. L., Cabel Moscoso, D. J., Humpiri Flores, R., & Huanca Marín, J. C. (**2022**). Revisión sistemática: vulnerabilidades de seguridad cibernética en los activos digitales. *ÑAWPARISUN - Revista de Investigación Científica de Ingenierías*, *4*(2), 250. https://doi.org/10.47190/nric.v4i2.250

IBM. (2024). *What are Classification Models?* IBM Cloud Learn Hub. https://www.ibm.com/es-es/think/topics/classification-models

IBM. (2024). What is clustering? IBM Cloud Learn Hub. https://www.ibm.com/es-es/think/topics/clustering

Instituto Nacional de Estadística e Informática [INEI]. (2021). *Cifras de población*. https://www.inei.gob.pe/estadisticas/indice-tematico/poblacion-y-vivienda/

Instituto Nacional de Estadística e Informática [INEI]. (2023). *Estadísticas de la criminalidad, seguridad ciudadana y violencia. Una visión desde los registros administrativos*. https://cdn.www.gob.pe/uploads/document/file/4774824/Estad%C3%ADsticas%20de%20Criminalidad%2C%20Seguridad%20Ciudadana%20y%20Violencia.%20Enero-Marzo%202023.pdf

Instituto Nacional de Estadística e Informática [INEI]. (2020). *Informe técnico de Estadísticas de Seguridad Ciudadana*. https://m.inei.gob.pe/media/MenuRecursivo/boletines/boletin_seguridad_ciudadana_abril2020.pdf

Kovacevic, A., Popovic, D., & Ristic, B. (2024). *Machine Learning*-based prediction of security incidents in the banking sector. *arXiv preprint*. https://arxiv.org/abs/2412.04495

Levine, R. (2005). *Finance and Growth: Theory and Evidence.* In Handbook of Economic Growth. Elsevier.

Liu, K., Zhou, Y., Wang, Q., & Zhu, X. (2024). *Vulnerability severity prediction with deep neural network*. En Proceedings of the 2019 5th International Conference on Big Data and Information Analytics (BigDIA) (págs. 114–119). IEEE.

MDPI. (2022). Transparent classification of CVE severity using SHAP and *XGBoost*. *Applied Sciences*, 12(20), 9231. https://www.mdpi.com/2076-3417/14/20/9231

Mishkin, F. S., & Eakins, S. G. (2018). *Financial Markets and Institutions* (9th ed.). Pearson.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

National Institute of Standards and Technology. (2020). *National Vulnerability Database (NVD)*. U.S. Department of Commerce. https://nvd.nist.gov

Navarro Bellido, J. (2023). *Modelos de aprendizaje automático en análisis de sentimiento: Comparativa de rendimiento* [Tesis de Grado, Universitat Politècnica de València]. Repositorio Institucional UPV. https://riunet.upv.es/handle/10251/192276

NIST. (2020). *National Vulnerability Database (NVD)*. National Institute of Standards and Technology. https://nvd.nist.gov/

Ouzan, O. (2025). *The critical flaw in CVE scoring*. DarkReading. https://www.darkreading.com/vulnerabilities-threats/critical-flaw-cve-scoring

Ozeren, S. (2025, 4 de julio). *Comparing CVSS, EPSS, KEV, SSVC, LEV & PXS: From Scores to Security Proof*. Picus Security. Recuperado de https://www.picussecurity.com/resource/blog/comparing-cvss-epss-kev-ssvc-lev-and-pxs-from-scores-to-security-proof

Pentest-Tools.com. (2025, julio). *When severity scores mislead – the case against single-metric risk models*. Recuperado de https://pentest-tools.com/blog/contextual-vulnerability-scoring

Qi, Z., Wang, J., & Huang, Y. (2024). *Machine Learning* strategies for banking security: An *XGBoost*-SMOTE approach. *arXiv preprint*. https://arxiv.org/abs/2406.04658

Russell, S. J. (2010). *Artificial intelligence a modern approach*. Pearson Education, Inc.

Sánchez-Bautista, G., & Ramírez-Chávez, L. (2022). Amenazas de seguridad a considerar en el desarrollo de software. *XIKUA Boletín Científico de la Escuela Superior de Tlahuelilpan*, *10*(19), 31–37. https://repository.uaeh.edu.mx/revistas/index.php/xikua/article/view/8118/8631

SCITEPRESS. (2023). *Machine Learning*-based prediction of vulnerability information subject to a security alert. In *Proceedings of the 9th International Conference on Information Systems Security and Privacy* (ICISSP 2023), 173–180. https://www.scitepress.org/Papers/2023/116137/116137.pdf

Sharda, R. (2024). Inteligencia de negocios analítica, ciencia de datos e IA: una perspectiva gerencial. Pearson Educación. https://ebooks7-24.com/?il=41105

Shimizu, N., & Hashimoto, M. (2025). *Vulnerability Management Chaining: An Integrated Framework for Efficient Cybersecurity Risk Prioritization*. arXiv. https://doi.org/10.48550/arXiv.2506.01220

Tenable. (2025, 31 de mayo). *What is CVSS? (Common Vulnerability Scoring System)*. Tenable Cybersecurity Guide. Recuperado de https://www.tenable.com/cybersecurity-guide/principles/common-vulnerability-scoring-system-cvss

Tiwari, H. (2025, March 26). *Advancing Vulnerability Classification with BERT: A Multi-Objective Learning Model*. arXiv. Preprint. Recuperado de arXiv:2503.20831

Tsutsui, K., Mori, K., & Nakaoka, S. (2025). CVE vulnerability classification using product relationships and *Random Forest*. In *Proceedings of the 2025 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*.

Vaidya, A., Adane, D., & Singh, R. (2021). Cybersecurity challenges in Industry 4.0: A comprehensive survey. *Journal of Cyber Security Technology*, 5(1), 1–20. https://doi.org/10.1080/23742917.2021.1876648

Vaidya, S., Ambad, P., & Bhosle, S. (2021). *Industry 4.0 – A glimpse*. *Procedia Manufacturing*, *45*, 546–553. https://doi.org/10.1016/j.promfg.2020.04.085