Implementación de un sistema de detección de intrusiones utilizando honeypots especializados en el procesamiento de pagos y generación de transacciones falsas, diseñados específicamente para su aplicación en entornos de transacciones en línea.

Jumbo Salcedo, Karen Lizeth y Garcia Romero, Mario Dario

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Trabajo de Unidad Integración Curricular, previo a la obtención del Título de Ingeniero de Software

Ing. Corral Díaz, María Alexandra, MSc

26 de febrero del 2024

Latacunga

UNIVERSIDAD DE LAS FUERZAS ARMADAS

Copyleaks

Plagiarism report

Tesina_Jumbo - Garcia_V3.pdf

Scan details

Scan time: March 1th, 2024 at 12:21 UTC | Total Pages: 87 | Total Words: 21743

Plagiarism Detection

1%

Types of plagiarism		Words
Identical	0.9%	186
Minor Changes	0%	0
Paraphrased	0%	0
Omitted Words	13.5%	2930

AI Content Detection

N/A

Text coverage: AI text, Human text

Plagiarism Results: (12)

García Ruíz Juan Memoria.pdf?sequence=1&isAllowed=y — 0.3%

https://riuma.uma.es/xmlui/bitstream/handle/10630/20399/garc%c3%ada%20ru%c3%adz%20juan%20memo...

Eduardo Guzmán De los Riscos

GRADO EN INGENIERÍA DEL SOFTWARE CIBERCANARIO: UN HONEYPOT BÁSICO, TANGIBLE Y USABLE PARA ENTORNOS IOT CYBERCANARY: A BASIC, TANGIBLE AND...

Blog elhacker.NET: Los mejores HoneyPots: ejemplos, tipos, característic... — 0.3%

https://blog.elhacker.net/2021/01/los-mejores-honeypots-ejemplos-y-tipos-trampas-rdp-ssh-cowrie-docker-rd...

Tienda Wifi CiudadWireless es la tienda Wifi recomendada por elhacker...

1502-1212_CampoverdeArmijosJI.pdf — 0.3%

http://bibliotecadigital.econ.uba.ar/download/tpos/1502-1212_campoverdearmijosji.pdf

jorge campoverde

Universidad de Buenos Aires Facultades de Ciencias Económicas, Cs. Exactas y Naturales e Ingeniería Carrera de Especialización en Segur...

Random Forest: La Selva de Arbolpedia — 0.1%

https://es.linkedin.com/pulse/random-forest-la-selva-de-arbolpedia-mauricio-mora-caballero-e5nke?trk=articl...

Mauricio Mora Caballero

Acepta...

Certified by Copyleaks | About this report help.copyleaks.com | copyleaks.com

Firma:

.....

Ing. Corral Díaz, María Alexandra, MSc

Directora

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Certificación

Certifico que el trabajo de unidad de integración curricular: **"Implementación de un sistema de detección de intrusiones utilizando honeypots especializados en el procesamiento de pagos y generación de transacciones falsas, diseñados específicamente para su aplicación en entornos de transacciones en línea"** fue realizado por los señores **Jumbo Salcedo, Karen Lizeth y Garcia Romero, Mario Dario**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 26 de febrero del 2024

Firma:

.....

Ing. Corral Díaz, María Alexandra, MSc

0501970487

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Responsabilidad de Autoría

Nosotros, **Jumbo Salcedo, Karen Lizeth y Garcia Romero, Mario Dario**, con cedulas de ciudadanía n° 1725425688 y 0550164412, declaramos que el contenido, ideas y criterios del Trabajo de Unidad de Integración Curricular: **"Implementación de un sistema de detección de intrusiones utilizando honeypots especializados en el procesamiento de pagos y generación de transacciones falsas, diseñados específicamente para su aplicación en entornos de transacciones en línea"**. Es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 26 febrero 2024

Firma

.....

Jumbo Salcedo, Karen Lizeth

C.C: 1725425688

Firma

.....

Garcia Romero, Mario Darío

C.C: 0550164412

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Autorización de Publicación

Nosotros, **Jumbo Salcedo, Karen Lizeth y Garcia Romero, Mario Dario**, con cedulas de ciudadanía n° 1725425688 y 0550164412, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de unidad de integración curricular: **"Implementación de un sistema de detección de intrusiones utilizando honeypots especializados en el procesamiento de pagos y generación de transacciones falsas, diseñados específicamente para su aplicación en entornos de transacciones en línea.**". En el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Latacunga, 26 febrero 2024

Firma

.....

Jumbo Salcedo, Karen Lizeth

C.C: 1725425688

Firma

.....

Garcia Romero, Mario Darío

C.C: 0550164412

Dedicatoria

Quisiera dedicar este proyecto a quienes han sido pilares fundamentales en mi vida. A mis amados padres, Walter y Sandra, cuyo amor incondicional y apoyo inquebrantable han sido mi mayor fortaleza. A Jhonny una persona muy importante que me ha brindado su apoyo inquebrantable y motivación constante para seguir adelante. A toda mi familia, cuyo aliento y respaldo han sido mi motor cada día.

A mis estimados compañeros, quienes han sido parte esencial de mi viaje universitario, gracias por confiar en mí, por compartir conocimientos y por ser una fuente inagotable de inspiración.

Dedico este trabajo de titulación al Ingeniero José Carrillo y a la Ingeniera Alexandra Corral, quienes me han guiado y aconsejado durante la realización de este proyecto

Karen Lizeth Jumbo Salcedo

Ecuador, Febrero del 2024

Agradecimiento

Primero quiero dar gracias a Dios por ayudarme en cada paso que doy en mi vida. A mis padres Walter Jumbo y Sandra Salcedo, por enseñarme el valor de la responsabilidad, por creer en mí y por alentarme en los momentos difíciles. Su amor y comprensión fueron mi motor para seguir adelante y alcanzar este logro.

Quiero expresar mi profundo agradecimiento al Ingeniero José Carrillo y a la Ingeniera Alexandra Corral, por su invaluable orientación y sabios consejos a lo largo de la realización de este proyecto de titulación. Sus conocimientos expertos y su dedicación fueron fundamentales para el desarrollo y éxito de este proyecto.

No puedo dejar de mencionar a mis profesores y profesoras, cuya guía académica y consejos me han inspirado y enriquecido durante mi formación universitaria. Su compromiso con la excelencia académica ha sido una inspiración para mí.

Agradezco también a mis compañeros y compañeras de clase, por sus ideas, debates y por compartir este viaje académico conmigo. Su compañerismo y colaboración fueron un estímulo constante en este camino.

Karen Lizeth Jumbo Salcedo

Ecuador, Febrero del 2024

Dedicatoria

A mis padres, Victoria y Miguel por su amor incondicional y apoyo constante, siendo ellos pilares de mi vida, cuyo sacrificio ha sido el faro que ha iluminado mi camino.

A mis amigos, que me ofrecen un espacio tranquilo y me motivan a seguir adelante.

A mis profesores y mentores, que me han guiado y han compartido conocimiento con paciencia y sabiduría.

Siendo esta tesis más que un logro académico, un fruto de dedicación e inspiración para superar mis limites

Mario Dario Garcia Romero

Ecuador, Febrero del 2024

Agradecimiento

Primordialmente quiero expresar mi agradeciendo a Dios, quien ha guiado cada paso, brindándome fortaleza en momentos de incertidumbre y por colmar mi vida de oportunidades de crecer, siendo este una fuente de inspiración y consuelo.

A mis queridos padres: Victoria mi madre, que siempre ha estado pendiente de mí, brindándome un cuidado incansable y ofreciéndome ser refugio en los momentos más difíciles. Y Miguel mi padre, siendo una guía inquebrantable, que con sabiduría y apoyo, han sido ejemplo trazado de un camino seguido con orgullo, demostrándome así su confianza que me ha inspirado a enfrentar cada desafío con determinación. Siendo esta tesis una muestra a su amor, dedicación y paciencia.

Extendiendo mi gratitud también a mis tutores, la Ingeniera Alexandra Corral y el Ingeniero José Luis Carrillo, cuya guía ha sido un apoyo fundamental para la realización de este trabajo, su compromiso, paciencia y enseñanzas no sólo me han ayudado a superar adversidades académicas, sino que han contribuido a mi desarrollo profesional y personal.

Y finalmente a mis amigos y todos aquellos que, de una forma u otra, han formado parte de este proceso, considerando que cada lección aprendida y cada desafío superado, han sido componentes clave en mi formación y éxito, motivándome día a día a superarme y escapar**.**

Mario Dario Garcia Romero

Ecuador, Febrero del 2024

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Resumen

En el marco de esta tesis, se propone y desarrolla un sistema de detección de intrusiones centrado en la seguridad de transacciones en línea, empleando honeypots especializados con el fin de imitar entornos de procesamiento de pagos y simular posibles atacantes. Este sistema destaca por su enfoque en un análisis del spoofing, técnica utilizada para fabricar una identidad crítica en el sector financiero. El estudio profundiza en el análisis del comportamiento malicioso, empleando algoritmos de aprendizaje automático para reconocer patrones de ataque y tomar medidas proactivas. Este enfoque no solo hace que el sistema sea más capaz de detectar y responder a los incidentes de seguridad, sino que también ayuda a crear un entorno más seguro y confiable para las transacciones digitales. La combinación de metodologías de desarrollo ágil con el desarrollo de sistemas permite una adaptación y evolución continuas en respuesta a las amenazas cibernéticas emergentes. De acuerdo con la constante evolución de las amenazas y la necesidad de soluciones innovadoras y adaptativas en la ciberseguridad financiera, este trabajo no solo aborda los desafíos técnicos inmediatos de la seguridad en transacciones financieras digitales, sino que con la aplicación de estrategias de segmentación adaptativa también plantea un marco de investigación y desarrollo orientado al futuro.

Palabras clave: IDS, honeypots, transacciones online, spoofing, transacciones falsas

Abstract

Within the framework of this thesis, an intrusion detection system focused on the security of online transactions is proposed and developed, using specialized honeypots in order to mimic payment processing environments and simulate possible attackers. This system stands out for its focus on an analysis of spoofing, a technique used to fabricate a critical identity in the financial sector. The study delves into the analysis of malicious behavior, employing machine learning algorithms to recognize attack patterns and take proactive measures. This approach not only makes the system better able to detect and respond to security incidents, but also helps create a more secure and trusted environment for digital transactions. The combination of agile development methodologies with system development enables continuous adaptation and evolution in response to emerging cyber threats. In keeping with the constant evolution of threats and the need for innovative and adaptive solutions in financial cybersecurity, this paper not only addresses the immediate technical challenges of digital financial transaction security, but with the application of adaptive segmentation strategies also poses a forward-looking research and development framework.

Keywords: IDS, honeypots, online transactions, spoofing, fake transactions.

Capítulo I

Introducción

Propósito y Contextualización del Tema

Hoy en día, el aumento de las transacciones digitales ha llevado a un aumento significativo de las amenazas cibernéticas. Esto ha dado lugar a una creciente preocupación en el ámbito de la ciberseguridad, lo que a su vez ha impulsado el desarrollo de sistemas de detección de intrusiones. Muchos de estos sistemas se basan en honeypots para abordar efectivamente y proactivamente estas amenazas a fin de garantizar la integridad y la confidencialidad de los negocios y los datos en línea. Teniendo en cuenta esto, el proyecto sugiere la implementación de potes de miel especializados en sistemas de transacciones en línea, procesamiento de pagos y generación de transacciones falsas (Medina Garzón & Vásquez Rodríguez, 2020).

En la era digital, los sistemas basados en la web están enfrentando una oleada creciente de amenazas, especialmente en el ámbito de las transacciones en línea. Estas amenazas, que a menudo permanecen ocultas durante un período prolongado, permiten a los actores maliciosos acceder a información sensible sin ser detectadas (Nomas ISO, 2018).

Las herramientas convencionales de seguridad, como los firewalls, los sistemas de detección de intrusiones y cortafuegos, han demostrado ser útiles, pero su eficacia ha disminuido frente a los atacantes cada vez más calificados (Quintero et al., 2019). Porque, así como las herramientas avanzan también lo hacen los atacantes con sus formas de ataque.

Con el aumento de las transacciones digitales, las ciber amenazas (desde el robo de datos al fraude financiero) se disparan. Los actores maliciosos han evolucionado sus técnicas, pasando de la simple explotación de vulnerabilidades a tácticas avanzadas como el phishing, el skimming y el malware móvil, y ante esta realidad, las respuestas de seguridad tradicionales han resultado insuficientes (Spitzner, 2003).

Se han propuesto soluciones innovadoras para hacer frente a este desafío. Un enfoque prometedor es el uso de honeypots específicamente diseñados en las transacciones en línea (Wang et al., 2023), que simulan plataformas legítimas y son de interés para los ciberdelincuentes y permiten un análisis en profundidad de los métodos de ataque.

Representan un enfoque proactivo que refuerza y adapta las defensas digitales para prepararlas frente a nuevas amenazas y garantizar la integridad de las transacciones en el mundo digital (Popovsky, 2015), El sistema aprende del ataque recibido y puede defenderse en consecuencia.

En este contexto, es importante analizar cómo los sistemas de transacciones en línea pueden beneficiarse de diferentes tipos de honeypots, que van desde aquellos que imitan sistemas de autenticación hasta aquellos que replican sitios de pago legítimos o rutas comerciales (Suarez Restrepo, 2021). Cada variante puede revelar información sobre distintos aspectos de las tácticas de ataque, como intentos de acceso no autorizado, manipulación de transacciones y tácticas de ingeniería social (Priya & Chakkaravarthy, 2023).

El proyecto implica la implementación de un sistema de detección de ataques utilizando honeypots especializados diseñados para el procesamiento de pagos y la generación de transacciones falsas. El objetivo es proporcionar una solución robusta y adaptable capaz de detectar y combatir las amenazas actuales y emergentes en las transacciones en línea.

El trabajo se desarrolló en tres fases: La primera fase consistió en el estudio de artículos científicos sobre el tema, estableciendo objetivos claros para el sistema de detección de intrusos honeypot, el tipo de amenazas a detectar y el nivel de detalle de las transacciones ficticias. Además, se eligió la tecnología adecuada y herramientas especializadas para la creación de honeypots en el procesamiento de pagos y la generación de transacciones ficticias. Estas herramientas pueden incluir software de emulación de sistemas de pago y herramientas de simulación de transacciones.

La segunda fase del proyecto implica la implantación del sistema. Esto incluye el diseño de la arquitectura del honeypot, la creación de la infraestructura necesaria para recopilar y gestionar de manera segura los datos, y la configuración del honeypot para emular perfiles de sistemas de pago y generar transacciones ficticias. Posteriormente, se despliega el honeypot en un entorno de red seguro.

Los honeypots incluyen elementos como formularios de inicio de sesión, campos de datos financieros y detalles de transacciones. También tienen cebos estratégicos para atraer a los atacantes y establecen un sistema de supervisión continua que registra datos detallados sobre las interacciones de los atacantes.

Luego, se analizan los datos recopilados para identificar patrones de ataque y comportamientos sospechosos utilizando algoritmos de aprendizaje automático como máquinas de vectores de soporte (SVM), bosques aleatorios, redes neuronales y técnicas especializadas de detección de anomalías. Esta información permite reforzar las defensas existentes, adaptarse a nuevas amenazas y tomar medidas proactivas para prevenir futuros ataques.

Como siguiente paso, se configuran alertas para informar sobre la detección de actividad maliciosa y se establecen procedimientos de respuesta a incidentes para hacer frente a las amenazas detectadas. Estos procedimientos pueden incluir el bloqueo de direcciones IP maliciosas, análisis forenses y notificación a las partes afectadas.

Finalmente, en la tercera fase, se realizan simulaciones para evaluar la eficacia de los honeypots instalados. Se utilizan técnicas como la validación cruzada y métricas de evaluación como la precisión, la exactitud y la recuperación para asegurar que los modelos puedan detectar con precisión las intrusiones.

El objetivo del proyecto no es solo identificar y abordar las amenazas actuales, sino también anticipar y prepararse para los desafíos futuros. Al comprender profundamente las

tácticas de los ciberdelincuentes, se busca mejorar la seguridad de las transacciones en línea, aumentar la confianza del usuario y establecer un entorno digital más seguro y resistente.

Justificación e Importancia

En la actualidad, con la creciente cantidad de personas y empresas realizando transacciones en línea, la ciberseguridad se vuelve un componente crucial. Sin embargo, estos procesos conllevan riesgos, ya que los ciberdelincuentes buscan acceder a los datos financieros de los usuarios. Para abordar esta problemática, la implementación de un sistema de detección de intrusiones utilizando honeypots especializados para el procesamiento de pagos y la generación de transacciones es esencial para identificar y contrarrestar las técnicas comunes de intrusión.

La importancia de este proyecto radica en su enfoque multifacético, que se centra en los aspectos clave de la ciberseguridad y la protección de datos financieros en las transacciones en línea, convirtiéndolo en una estrategia global para combatir las amenazas cibernéticas. El uso de técnicas de aprendizaje automático mejora significativamente la precisión de la detección, adapta rápidamente a las nuevas amenazas, optimiza los recursos de seguridad, genera alertas tempranas y respuestas eficientes, todo ello mantenido a través de un aprendizaje continuo frente a las amenazas cibernéticas en constante evolución. La implementación de honeypots mejora la seguridad y la protección de las operaciones financieras en línea.

El alcance del proyecto va más allá de la mera detección; los resultados obtenidos de la panadería son analizados para proporcionar insights valiosos en las tácticas y técnicas de los ciberdelincuentes. Estas perspectivas permiten el desarrollo de medidas preventivas y correctivas más eficaces. Además, todo este discernimiento y experiencia culminan en esta tesis, que no solo sirve como testimonio del trabajo realizado, sino que también actúa como

una valiosa referencia para futuras investigaciones y desarrollos en el campo de la ciberseguridad.

Objetivos

Objetivo General

Implementar un sistema de detección de intrusiones utilizando honeypots especializados en el procesamiento de pagos y generación de transacciones falsas, diseñados específicamente para su aplicación en entornos de transacciones en línea.

Objetivos Específicos

- Comprender definiciones, terminología y el funcionamiento de los honeypots, así como los distintos tipos de honeypots diseñados para rastrear actividades en línea, las técnicas computacionales avanzadas empleadas en su implementación y adquirir un conocimiento detallado acerca de su proceso de desarrollo. Además, conocer técnicas avanzadas computacionales para implementar un honeypots especializado en un sistema de transacciones online.

- Implementar un honeypot especializado en el procesamiento de pagos y generación de transacciones falsas aplicado a transacciones online.

- Validar resultados del sistema de detección de intrusos mediante honeypots aplicado a transacciones online.

Metodología

El objetivo de este proyecto es implementar un sistema de detección utilizando honeypots especializados en el procesamiento de pagos y la generación de transacciones falsas. Este sistema se ha diseñado específicamente para su implementación en entornos de transacciones en línea, con el objetivo de mejorar la seguridad y proteger contra las amenazas cibernéticas. Su estructura se compone de tres fases distintas.

En la fase inicial, se realizó una revisión sistemática para obtener una comprensión profunda de las definiciones, la terminología y el funcionamiento de los honeypots. Se llevó a cabo una investigación exhaustiva sobre los diferentes tipos de honeypots diseñados para monitorear actividades en línea, así como sobre las técnicas computacionales avanzadas utilizadas en su creación. Esta revisión se basó en una amplia literatura relacionada con la seguridad en transacciones en línea, honeypots y amenazas cibernéticas. El objetivo era comprender completamente el proceso de desarrollo de un sistema de detección utilizando honeypots especializados, especialmente en el contexto de las transacciones en línea con Secure Pay Guard.

La segunda fase se enfocó en la aplicación práctica del sistema. Se diseñó y se implementó un honeypot especializado para simular transacciones falsas en el ámbito de los pagos en línea. La ejecución de este honeypot fue un paso crucial en la construcción del sistema de detección de intrusiones. Su diseño se alineó cuidadosamente con los escenarios de amenaza típicos en este entorno, lo que permitió una mayor eficacia en la identificación y el seguimiento de posibles ataques.

Se utilizó la metodología SCRUM para el desarrollo de software en esta fase. Esto creó un entorno de desarrollo controlado que consistía en actividades vigiladas, controladas y organizativamente validadas con características ágiles y flexibles que se adaptaban a los requisitos. En caso de eventualidades, el equipo de trabajo correspondiente abordaba cuestiones individuales para garantizar un progreso exitoso en la ejecución del proyecto (Estrada Velasco et al., 2021).

Después, se procedió a validar los resultados obtenidos por el sistema de detección de intrusiones basado en el honeypot. Esta validación se realizó tanto en entornos controlados como en situaciones de la vida real. El objetivo principal es verificar la efectividad del sistema para identificar y mitigar amenazas de seguridad en transacciones en línea y asegurar la

confiabilidad del sistema antes de su implementación definitiva en entornos de transacciones en línea.

Finalmente, se evalúa si el sistema cumple con los objetivos planteados y se identifica áreas de mejora. Las conclusiones resumen los hallazgos clave de la tesis y discuten su relevancia en el contexto de la seguridad en transacciones en línea. En conjunto, esta metodología proporciona una estructura sólida para llevar a cabo la investigación y lograr los objetivos propuestos en esta tesis.

Capítulo II

Marco Teórico

En este capítulo se lleva a cabo una investigación teórica centrada en los sistemas de detección de intrusiones (IDS) utilizando honeypots (HPs), explorando los elementos clave como los factores determinantes, los modelos y algoritmos de Machine Learning aplicados en este contexto, las herramientas utilizadas para su desarrollo y la metodología empleada en su creación (Mokube & Adams, 2007).

Para llevar a cabo este análisis, se llevó a cabo una exhaustiva revisión de la literatura utilizando la base de datos de revistas indexadas en SCOPUS. Se definió una cadena de búsqueda con términos relevantes en este enfoque específico, la cual se empleó con el objetivo de revisar minuciosamente la base de datos en busca de artículos pertinentes, finalmente teniendo en cuenta de realizar varias iteraciones en Scopus, con el fin de encontrar sinónimos y palabras relacionadas, que se ajusten de mejor manera al objeto de estudio.

Sistemas Transaccionales en línea

Los sistemas transaccionales en línea son sistemas informáticos que permiten a los usuarios realizar transacciones comerciales de forma rápida y eficiente. Estos sistemas se utilizan en una amplia gama de aplicaciones, incluyendo banca, comercio electrónico, gestión de inventarios y atención al cliente (IBM, 2020).

Para realizar transacciones en línea en una banca web el usuario debe registrarse en el sitio web o aplicación móvil del banco proporcionando sus datos personales y creando una cuenta con usuario y contraseña. Una vez que inicia la sesión, puede acceder a sus productos bancarios online y puede realizar diversos tipos de transacciones como transferencias y pagos de servicios entre otras acciones.

Antes que cualquier transacción pueda ser completada, los datos necesarios son introducidos, confirmados y autenticados a través de un factor específico. Al finalizar la

transacción, se emite un recibo digital que contiene los detalles de la operación finalizada. (Internacional, 2021).

El proceso de pago en línea implica una serie de pasos entre el cliente, la empresa y varias instituciones financieras. Cuando un consumidor inicia una compra en un sitio web, su información de pago es cifrada y enviada al procesador correspondiente a través de una puerta de pago. El procesador verifica los detalles de la transacción con el banco emisor del cliente, que o bien aprueba o rechaza la operación basándose en los fondos disponibles (Murugappan et al., 2023). Este proceso se basa en las transacciones en línea, ya que estos sistemas poseen propiedades específicas que figuran en la Tabla 1.

Tabla 1

Propiedades de los sistemas Transaccionales en línea

Ord.	***Propiedad***	***Descripción***
1	Transacciones cortas y frecuentes	Las transacciones se ejecutan rápidamente y se repiten con frecuencia.
2	Gran volumen de datos	A menudo manejan grandes volúmenes de datos, ya que necesitan almacenar información sobre las transacciones de los usuarios.
3	Requisitos de rendimiento	Deben ofrecer un alto rendimiento, ya que necesitan procesar un gran número de transacciones de manera rápida y eficiente.

Nota. La tabla 1 presenta las propiedades del funcionamiento de los sistemas transaccionales en línea (Rodrigues, 2021).

Apoyados por diversas tecnologías, los sistemas de transacciones gestionan de manera eficiente grandes volúmenes de operaciones en tiempo real. Para comprender estos fundamentos técnicos, se ha creado una tabla que resume las tecnologías involucradas. La

Tabla 2 incorpora estos aspectos clave de las soluciones tecnológicas para la construcción de plataformas de transacciones robustas y de alto rendimiento.

Tabla 2

Tecnologías que utilizan los sistemas Transaccionales en línea

Ord.	Tecnología	Descripción
1	Bases de datos relacionales	Son un tipo de base de datos que es adecuado para sistemas de transacciones en línea, ya que permiten un almacenamiento y gestión eficientes de grandes volúmenes de datos.
2	Arquitecturas de tres capas	Separan la interfaz de usuario, la lógica de negocio y la capa de datos, lo que facilita el mantenimiento y el escalado de los sistemas transaccionales en línea.
3	Tecnologías de procesamiento en paralelo	Permiten que los sistemas de transacciones en línea procesen simultáneamente un mayor número de operaciones.

Nota. Tecnologías claves que sustentan el funcionamiento de plataformas transaccionales escalables y de alto rendimiento (Martinekuan, 2023).

Procesamiento de pagos en línea

La ejecución segura y eficiente de las transacciones monetarias a través de entornos digitales implica el procesamiento de pagos en línea, que es un componente crucial de las operaciones financieras electrónicas (Alqudhaibi et al., 2023).

Según (Tariq et al., 2023), este proceso implica múltiples etapas, desde la solicitud inicial de transacción por parte del cliente hasta la confirmación y autorización por las instituciones financieras y comerciantes involucrados.

Los clientes, los adquirentes y los comerciantes desempeñan un papel importante en el ciclo sincronizado de las transacciones, pero no son los únicos actores clave en este proceso (Li et al., 2023).

Los procedimientos y reglamentos establecidos garantizan la seguridad del procesamiento de pagos en línea. Las normas como PCI DSS (Seguridad de datos de la industria de las tarjetas de pago) requieren la protección de datos de tarjetas de pago, y tecnologías como SSL/TLS garantizan que la información permanezca confidencial durante la transmisión (Wang et al., 2023).

Sin embargo, las cuestiones de seguridad como el fraude con tarjetas de crédito y el riesgo de interceptar datos confidenciales durante las transacciones persisten a pesar de estas medidas (Vetrivendan & Kumar, 2023).

El procesamiento de pagos en línea requiere no sólo la facilidad de las transacciones financieras a través de medios electrónicos, sino también la consideración cuidadosa de los protocolos de seguridad (Teng, 2022).

Monitorización de transacciones

En el ámbito del procesamiento de pagos en línea, el seguimiento de las transacciones se convierte en una práctica crucial para garantizar la fiabilidad y la seguridad de las operaciones financieras electrónicas (Attia et al., 2019).

La vigilancia implica la observación sistemática y continua de las transacciones financieras para detectar patrones inusuales o actividades sospechosas que puedan indicar posibles amenazas a la seguridad (Alsaeed et al., 2024).

Existen varios métodos disponibles para supervisar las transacciones a este respecto. Detección de patrones extraños se basa en el análisis de comportamientos inusuales que pueden indicar actividades fraudulentas (Alvarez, 2021).

Sin embargo, el análisis del comportamiento requiere examinar el historial de transacciones para identificar cualquier alteración en los patrones de comportamiento típicos del usuario o sistema (Morales, 2018).

La vigilancia en tiempo real se ha vuelto común gracias a la adopción de tecnologías avanzadas como los Sistemas de Detección de Intrusiones (IDS) y los sistemas de prevención de intrusiones (IPS) (Alsaeed et al., 2024).

El seguimiento de las transacciones en el procesamiento de pagos en línea es una medida crucial para reducir los riesgos y detectar las amenazas. Su enfoque integral implica la identificación de patrones inusuales y el análisis de comportamientos, todos respaldados por tecnologías avanzadas de seguridad (IDS, IPS) (Lee et al., 2019).

Tipos de Ataques

Transacciones Falsas

En el ámbito de la seguridad del procesamiento de pagos en línea, la creación de transacciones falsas se destaca como una estrategia clave para evaluar y mejorar la resiliencia de los sistemas frente a posibles amenazas y vulnerabilidades (Balfaqih et al., 2023).

La creación de transacciones fraudulentas controladas y simuladas forma parte de esta práctica para evaluar la capacidad de los sistemas de detección de intrusiones y validar la eficacia de las medidas de seguridad implementadas por el honeypot (Frikha et al., 2023).

Los métodos para generar transacciones falsas deben vigilarse y deben reflejar posibles escenarios de amenaza.

> (Menon et al., 2023) afirman que la simulación de transacciones fraudulentas mediante la introducción de patrones controlados de comportamiento anormal es un método ampliamente utilizado. Este método asegura la evaluación de la capacidad del

sistema para detectar actividades inusuales y adaptarse a las nuevas tácticas de los atacantes.

Considerando la necesidad de fortalecer la resistencia y anticipar posibles amenazas emergentes, es crucial evaluar los sistemas de detección de intrusiones y la robustez de las medidas de seguridad mediante la generación de transacciones falsas (Maseleno, 2021).

Tipos de transacciones falsas

Existen varios tipos de transacciones fraudulentas que pueden clasificarse basándose en el objetivo del atacante, el método utilizado para generar las operaciones y el tipo de sistema transaccional que se está atacando, como se muestra en Tabla 3.

Tabla 3

Tipos de Transacciones falsas

Ord.	Tipo de transacción falsa		Descripción
1	Según el objetivo	Fraude.	El objetivo es lograr un beneficio financiero, como el robo de dinero o bienes.
		Daños	El objetivo es perturbar el sistema de transacciones, por ejemplo, interrumpiendo el servicio o dañando la reputación de la empresa.
		Espionaje	El objetivo es recopilar información sensible, como números de tarjetas de crédito o contraseñas.
2	Método del atacante	Manual	El atacante crea manualmente transacciones usando herramientas o software básicos.

Ord.	Tipo de transacción falsa		Descripción
		Automática	El atacante utiliza herramientas o software sofisticados para generar transacciones automáticamente.
3	Tipo de sistema	Transacciones en línea	Las transacciones se llevan a cabo a través de Internet, como en el comercio electrónico o los pagos en línea.
		Transacciones móviles	Las transacciones se realizan a través de dispositivos móviles, como teléfonos inteligentes o tabletas.
		Transacciones en persona	Las transacciones se realizan en persona, como en un cajero automático o en una tienda.

*Nota. V*ariedad de transacciones fraudulentas en sistemas de transacciones en línea.

Identifica los rasgos distintivos en relación al objetivo (Elizabeth, 2023), método del atacante (Ayuda Ley, 2020), y tipo del sistema (La Hora, 2023), considerándolo fundamental para facilitar la implementación de controles y prevenir o detectar estos ataques a tiempo.

Suplantación de Identidad o Spoofing

El spoofing es la práctica de suplantar la identidad de un remitente o receptor en una comunicación electrónica. Esto se logra cambiando la dirección IP, correo electrónico o número de teléfono de un dispositivo. Es una forma de fraude de identidad que ocurre en entornos digitales y electrónicos, donde los delincuentes obtienen información confidencial de sus víctimas para participar en actividades delictivas. Este documento se enfocará en la detección de los ataques de spoofing en las transacciones en línea falsas (Proofpoint, 2024).

El término "spoofing" proviene del inglés "spoot", que se traduce como imitación. Con intenciones maliciosas, los perpetradores pueden acceder a la información privada de las víctimas. Los datos extraídos pueden ser utilizados para el robo financiero, perjudicar a otros, o infectar un dispositivo. Para contrarrestar este tipo de ataques, se recomienda proteger la dirección IP del cliente para evitar la identificación y autenticarla mediante algoritmos criptográficos (Fernández, 2024).

El spoofing es uno de los ataques cibernéticos más comunes en la actualidad. Es una práctica peligrosa en la que el ciberdelincuente se hace pasar por otro individuo o empresa, lo que potencialmente puede resultar en el robo de datos personales e instalación de malware (Oracle, 2023).

Se puede utilizar el spoofing para enviar correos electrónicos fraudulentos en campañas de phishing o para enviar mensajes de texto de phishing con el objetivo de redirigir a una página falsa para robar datos. El cibercriminal aparenta ser una empresa de la que te puedes fiar, por celebridades o personas reconocibles, con la finalidad de engañarte, diciéndote que hay alguna gestión u oferta urgente, y que tienes que ir a determinada página web y completarla. Estas páginas a las que te llevan son falsificadas, mostrando la ilusión de estar en la web real del banco o de la oferta, donde habrá cuestionarios en los que solicita que ingreses tus datos de tu cuenta de banco (Walsh, 2023).

Tipos de spoofing

La comprensión de los diversos métodos de falsificación spoofing es crucial debido a su aparición común en los sistemas en línea. Se ha creado una tabla que clasifica los principales tipos de engaño según sus características y el componente que intentan suplantar para abordar este tema de manera integral. Con el fin de mejorar la detección de ataques y fortalecer la seguridad en los sistemas, el lector podrá comprender las particularidades de cada uno consultando la Tabla 4.

Tabla 4

Tipos de Spoofing

Ord.	***Tipos de spoofing***	***Descripción***
1	Correo electrónico	Cuando un cibercriminal se pone en contacto contigo suplantando la identidad de una persona o empresa que conoces, para pedirte información privada o dinero. El engaño se realiza usando direcciones o remitentes muy parecidos a los originales y archivos descargables infectados con virus.
2	Páginas web	el estafador crea la réplica de una página web de confianza, usando logotipos y colores similares, para robar información privada de los usuarios o instalar malware.
3	Identificador de llamadas	consiste en la suplantación de identidad de una empresa usando sus números de teléfono o su ubicación geográfica. Los cibercriminales usan un método de llamadas de voz en línea denominado Voz sobre Protocolo de Internet, que envía las voces de manera digital y no análoga, haciendo que la información se pueda vulnerar más fácilmente**.**
4	Mensajes de texto	Cuando un cibercriminal se hace pasar por una organización o persona y te envía mensajes de texto con links que te dirigen a páginas web fraudulentas, o solicitando él envió de dinero e información personal.

Ord.	*Tipos de spoofing*	*Descripción*
5	Dirección IP	cuando los ciberdelincuentes cambian deliberadamente la ubicación geográfica de una computadora instalando una IP desconocida. Luego, inyecta paquetes de datos de IPs fraudulentas en la red, a través de las cuales puede acceder a información privada.
6	Servidor DNS	Consiste en cambiar la ruta por la que accedes a una página web para enviarte a un sitio malicioso.
7	Ataque de intermedio	es un método en el que el cibercriminal intercepta un canal de comunicación para obtener información sensible de una o varias víctimas.

Nota. Categorización de los ataques comunes de robo de identidad y falsificación según los elementos y métodos que pretenden falsificar (Pichincha, 2021).

Honeypots

Un honeypot se utiliza para llevar a cabo el proceso de detección del ataque Spoofing. Honeypot es una tecnología de defensa activa en la que ciertos hosts y servicios de red se configuran para detectar y analizar los comportamientos de ataque, sirviendo como un complemento eficaz a los sistemas de seguridad pasiva tradicionales (Tian et al., 2020).

Un honeypot no es un sistema de detección de intrusiones, pero puede mejorar los métodos de detección y proporcionar nuevos patrones de ataque. Diseñado para engañar a los intrusos, estudiar sus actividades, y aprender sus métodos, basado en el concepto de "conocer al enemigo" y combatirlo (Gonzales Gomez, 2003).

También conocidos como trampas de miel, es un sistema de engaño diseñado que atrae a los atacantes. La mayoría de estos sistemas se instalan detrás de un firewall, aunque también es posible colocarlos delante de él. El firewall que gestiona el tráfico de un honeypot

normalmente está configurado para permitir conexiones entrantes al sistema y restringir las conexiones salientes (Martin et al., 2017).

Un software conocido como "honeypot" está diseñado con el fin de atraer a los atacantes pretendiendo que el sistema es débil o susceptible y podría verse comprometido. Tienen la esencia de la contrainteligencia ya que invita a los atacantes a sistemas vulnerables, pero siendo beneficioso para el administrador de la red u organización (Campoverde Armijos, 2018).

Tipos de Honeypots

Existe una variedad de clases de honeypots, cada una muestra sus propias características. Se ha creado una tabla que clasifica los honeypots según su nivel de interactividad y funciones para que los usuarios puedan comprender mejor estos sistemas de señuelo. La Tabla 5 tiene como objetivo facilitar la identificación del tipo más apropiado en función de las prioridades de cada honeypot.

Tabla 5

Tipos de Honeypots

Ord.	Tipos de Honeypots	Descripción
1	Honeypot de investigación	Se utilizan con fines educativos para mejorar la seguridad. Contiene datos rastreables que pueden ser rastreados si son robados para analizar el ataque
2	Honeypot de producción	Desvían la atención delictiva del sistema real mientras analizan la actividad maliciosa para contribuir a la mitigar las vulnerabilidades, siendo como sistemas señuelo dentro de redes y servidores en pleno funcionamiento.

Ord.	Tipos de Honeypots	Descripción
3	Honeypot de alta interacción	Aquí se encuentran aquellos sistemas que utilizan un entorno real sin simular los diferentes servicios, sino que trabaja con servicios de verdad, este es una herramienta avanzada que replica en su totalidad un sistema operativo real, que permite una interacción profunda con los posibles atacantes. Particularmente estos honeypots son útiles para estudiar tácticas avanzadas y técnicas complejas de intrusión, volviéndolos así en sitios atractivos para los atacantes y permitiendo su estudio complejo. Sin embargo, el nivel de riesgo que estos presentan es elevado, debido a que deben estar constantemente monitorizadas, ya que existe la posibilidad de que el atacante pueda disponer del sistema.
4	Honeypot de media interacción	Crece la relación entre el atacante y el sistema incluyendo recursos falsos, como sistemas de ficheros o servidores de FTP o SSH. Pueden resultar útiles para capturar comandos que puedan ejecutar los atacantes.
5	Honeypot de baja interacción	Ofrecen un sistema con las mismas características que un dispositivo real, con acceso total al atacante, ya que las vulnerabilidades son completamente explotables. Debido a la facilidad de su implementación es el tipo de honeypot que se utilizó para la detección de intrusos en el procesamiento de transacciones falsas.

Nota. La Tabla 5 (El Brujo, 2021), (Atico, 2020), (Verdejo Alvarez, 2004) *clasifica las variedades de honeypots según su sofisticación técnica y objetivos, lo que facilita la selección de sistemas de detección de amenazas según necesidades particulares.*

Ventajas

Falsas alarmas: Estas herramientas de seguridad sólo deben recibir actividades sospechosas. Reducción del número de falsas alarmas y falsos negativos en la detección de ataques (Eduard & Daniel, 2013).

Pocos y valiosos datos: Se registran una gran cantidad de datos de mucho valor en el contexto de honeypots. Esos sistemas sólo son utilizados como objeto de ataques y no registra cantidades importantes de información, sin quitarle importancia, porque está relacionada con actividades hostiles, haciendo que los datos sean claros y fáciles de analizar. Estos sistemas registran toda la actividad, permitiendo la identificación de tales ataques mientras soporta el protocolo IPv6 (Hans710, 2009).

Recursos: Los honeypots no realizan análisis sobre las actividades que registran. Dado que consumen recursos limitados, a diferencia de muchos IDS, que pueden descartar información por esta razón. Estos sistemas se centran en la infraestructura necesaria para registrar toda la actividad que ocurre dentro de ellos (kaskpersky, 2023).

Simplicidad: Son simples de entender, configurar e instalar al no poseer algoritmos complejos, la simplicidad de la implementación es tan sencilla como conectarlo a la red (Esteban & Ignacio, 2013).

Encriptación: Se caracterizan por lograr la actividad ocurrida en cada momento, estos sistemas suelen implicar como uno de los extremos de la comunicación cifrada durante un intrusión o ataque (Fortinet, 2024).

Permiten el estudio de nuevos tipos de ataques, representando una gran ventaja, ya que permite comprender los ataques que se producen en la organización y en base a esto crear un plan de mitigación (Crowd, 2022).

Desventajas

Al ser elementos pasivos, si no están colocados de manera correcta en la red no logran capturar ningún dato y no tendrán ningún valor (Yamin et al., 2021).

Si un atacante es capaz de identificar uno de estos sistemas, puede deshabilitar toda la eficacia del honeypot, impidiéndoles ser atacados, incluso si se encuentran en la misma red que otros sistemas de producción que podrían estar siendo atacados (Avilés Bajaña, 2017).

Si un honeypot es atacado con éxito, el intruso puede usarlo para acceder a los otros sistemas de la red donde está instalado. El riesgo puede variar dependiendo del nivel de complejidad del honeypot; los honeypots más simples conllevan menores riesgos. Siendo un aspecto crucial al implementar este tipo de sistemas (Espí Luis, 2017).

Al ser vulnerables intencionalmente, pueden ser de gran atracción para los atacantes novatos, que usan herramientas públicas automáticas que tratan de vulnerar los sistemas, y si no poseen un ambiente controlado el honeypot puede ser utilizado como puente de ataques a otras redes internas (Tovar Sergio, 2018).

Tienen una visión limitada, por lo que se afirma que los honeypots no sustituyen ningún mecanismo de seguridad, sino que trabajan en conjunto a efecto de mejorar el perímetro de seguridad de la organización. El honeypot puede ser usado como un zombi para llegar a otros sistemas y ponerlos en peligro (Takhion, 2018).

Honeypots para sistemas transaccionales

Los honeypots para sistemas transaccionales son un tipo de honeypot que se utiliza con el fin de proteger los sistemas que procesan transacciones financieras. Estos sistemas

suelen ser objetivos atractivos de los atacantes, ya que pueden obtener acceso a información confidencial o realizar transaccionales fraudulentas (Jaramillo Ramos & Ospina Beltrán, 2019).

Con el objetivo de entender mejor el honeypot se ha creado una tabla que describe los principales usos de estos señuelos en contextos transaccionales que facilitan su comprensión. Como resultado, la Tabla 6 enfatiza la capacidad de los honeypots para mejorar la ciberseguridad de las plataformas transaccionales en respuesta a una variedad de modalidades de ataques e intrusiones.

Tabla 6

Uso de Honeypots para sistemas transaccionales

Ord.	Uso	Descripción
1	Detectar ataques	Pueden ayudar a detectar ataques antes de que afecten a los sistemas reales.
2	Recopilar información sobre los atacantes	Recopilar información sobre los atacantes
3	Disuadir a los atacantes	Puede disuadir a los atacantes de atacar los sistemas reales, ya que saben que serán detectados y atrapados.

Nota. Para facilitar su comprensión, se ha creado una tabla que describe los principales usos de estos señuelos en contextos transaccionales. Como resultado, la Tabla 6 destaca la capacidad de los honeypots en relación a mejorar la ciberseguridad (Verdejo Alvarez, 2004).

Tipos de honeypots para sistemas transaccionales.

Los honeypots tienen diseños específicos para usos financieros y transaccionales. Se ha creado una tabla con los tipos de sistemas de señuelo orientados a servicios bancarios y transacciones electrónicas con el fin de facilitar la comprensión de las opciones disponibles.

Por lo tanto, la Tabla 7 permite elegir el honeypot más adecuado según las prioridades y los recursos existentes.

Tabla 7

Tipos de honeypots para sistemas transaccionales.

Ord.	Tipos de honeypots para sistemas transaccionales	Descripción
1	Honeypots de aplicaciones	Estos honeypots se configuran para simular aplicaciones específicas, como cajeros automáticos o sistemas de pago en línea.
2	Honeypots de red	Estos honeypots se configuran para simular dispositivos de red, como routers o switches.
3	Honeypots de datos	Estos honeypots se configuran para simular datos confidenciales, como números de tarjetas de crédito o contraseñas.

Nota. La Tabla 7 clasifica las diferentes variedades de honeypots para aplicaciones financieras y transacciones en línea enfocado a los tipos de honeypots para sistemas transaccionales (Ciberseguridad, 2009).

Los honeypots son una herramienta valiosa para la seguridad de los sistemas transaccionales. Pueden ayudar a las organizaciones a detectar ataques, recopilar información sobre los atacantes y disuadir a los atacantes (Garg et al., 1998).

Sistema de detección de intrusiones (IDS)

Uno de los mecanismos de defensa más utilizados para reducir el riesgo de ataques dirigidos contra activos informáticos ha sido los sistemas de detección de intrusiones. Un sistema de detección de intrusiones (IDS) es una herramienta de seguridad diseñada que

supervisa los eventos que ocurren en un sistema informático con el propósito de detectar intentos de intrusión (Mira Alfaro, 2003).

Definiendo intento de intrusión como cualquier esfuerzo por comprometer la confidencialidad, integridad, disponibilidad o eludir los mecanismos de seguridad de una computadora o red (Perez Carlos, 2005).

Un sistema de detección de intrusiones (IDS) es un componente que monitora y analiza toda la información que fluye a través de una red de datos para identificar posibles ataques. Cuando se produce un ataque, el sistema responde notificando al administrador y cerrando las puertas al potencial intruso reconfigurando elementos de la red como firewalls y routers (Gómez López, 2009).

IDS ha sido ampliamente utilizado por muchas empresas en los últimos años porque han proporcionado una capa adicional de seguridad. Sin embargo, se ha encontrado que estos proporcionan seguridad; la protección requiere un ataque para existir primero. Si un ataque es pequeño y eficaz, el IDS reacciona lentamente y el ataque alcanza su objetivo (López, 2009).

Procesamiento de Datos

El procesamiento de datos es el proceso de recopilación, organización, limpieza, análisis y presentación de datos. Es una parte esencial de muchas actividades humanas, desde la investigación científica hasta la toma de decisiones (Ortegón Cortázar, 2015).

Es una herramienta poderosa que puede utilizarse para una variedad de propósitos. El análisis de grandes cantidades de datos puede ser beneficioso para las empresas en la toma de decisiones, para los científicos en el descubrimiento de conocimientos, y para las personas en la comprensión del mundo que les rodea (Digital, 2021).

El procesamiento de grandes volúmenes de datos analíticamente requiere el uso de una variedad de métodos para extraer su valor. La Tabla 8 proporciona orientación sobre los

procedimientos necesarios para generar información útil en la toma de decisiones, desde la recopilación de datos hasta la interpretación de conjuntos de datos.

Tabla 8

Técnicas para el procesamiento de datos.

Ord.	Técnicas	Descripción
1	Recopilación de datos	Los datos se recopilan de la fuente deseada utilizando diferentes métodos, como encuestas, experimentos, registros históricos, sensores, web scraping y extracción de datos.
2	Organización de datos	Se utilizan para estructurar los datos de una manera que sea fácil de entender y analizar. Se pueden basar en reglas o en aprendizaje automático.
3	Limpieza de datos	Se utilizan para eliminar errores y los datos incompletos. Estas técnicas son importantes para garantizar la precisión y la fiabilidad de los resultados del análisis de datos. Algunas de las técnicas son: la corrección de errores, completamiento de datos y estandarización de datos.
4	Análisis de datos	Se usan para examinar los datos y extraer información, pueden ser estadísticas, matemáticas o de aprendizaje automático.

Ord.	Técnicas	Descripción
5	Técnicas de aprendizaje autónomo	Utilizan algoritmos de machine Learning para analizar los datos. Por ejemplo, los algoritmos pueden utilizarse para identificar patrones en los datos o para clasificar los datos.

Nota. La Tabla 8 resume cadena de técnicas para explotación de big data utilizadas para el análisis del tema desarrollado (Gonzales Kevin, 2013).

Modelo y/o algoritmo de Machine Learning

La adopción de modelos y algoritmos de aprendizaje automático (ML) en la detección de intrusiones en transacciones de procesamiento de pagos en línea es esencial para la prevención y respuesta efectiva ante amenazas cibernética (Zheng et al., 2018).

La literatura revisada afirma que la elección cuidadosa de un modelo de ML adecuado es esencial para garantizar una detección precisa y efectiva de actividades maliciosas (Rasheed et al., 2023).

El proceso de elegir un modelo implica tener en cuenta una variedad de factores, incluidos el tipo de datos disponibles, la complejidad de las transacciones y la capacidad del modelo para adaptarse a nuevos patrones de amenazas (Harrison, 2019).

La implementación de modelos basados en métodos de aprendizaje supervisados, no supervisados o semi supervisados dependiendo de la disponibilidad de etiquetas de datos ayuda a lograr un enfoque más amplio y preciso para detectar anomalías (Kazemian & Shrestha, 2023).

El desarrollo del modelo depende mucho de factores como los patrones de comportamiento, las características de las transacciones y los perfiles de los usuarios. Es por esto que generalizar situaciones del entorno controlado, el entrenamiento y la validación debe llevarse a cabo rigurosamente. La capacidad del modelo en adaptarse a la evolución de las

amenazas cibernéticas es fundamental con el propósito de garantizar la eficacia de la detección a lo largo del tiempo (Sanni et al., 2023).

Random Forest

Es un enfoque de aprendizaje supervisado para clasificación y regresión, Random Forest funciona al construir un conjunto de árboles de decisión en el proceso de entrenamiento y produce la predicción mediante la mediación de las predicciones individuales de cada uno de ellos. Es fundamentado sobre la técnica de ensemble learning, donde se crea múltiples modelos como arboles de decisión y combina sus predicciones (Liu et al., 2012).

Random Forest combina muestreo de datos y feature sampling para construir un modelo robusto y de alto rendimiento, ideal en muchos problemas de clasificación y regresión (Rigatti, 2017).

Regresión Logística

La regresión logística es una técnica que permite analizar la relación entre una variable dependiente dicotómica y un conjunto de variables predictoras o independientes. Esta técnica puede utilizarse con fines explicativos y predictivos (Chitarroni Horacio, 2002).

La regresión logística funciona cuando las variables independientes o predictoras son cuantitativas, también conocidas como covariables; o cuando son categóricas. En este caso es necesario transformar las variables categóricas en variables ficticias, que tomas valores de 0 y 1, antes de aplicar el modelo de regresión logística (Alba Vega & Calle Jara, 2020).

Permite modelar y analizar si la presencia o ausencia de una característica o resultado puede ser predicha a partir de un conjunto de variables predictoras que pueden ser numéricas o categóricas (Martos et al., 2017).

Metodología Ágil Scrum

Scrum es una metodología de desarrollo de software ágil reconocida mundialmente que se basa en un enfoque incremental basado en la teoría del control de procesos y la transparencia, la inspección y la adaptación (Estrada Velasco et al., 2021).

En este proyecto se utilizó la metodología Scrum como un marco de trabajo ágil para la gestión de proyectos que se basa en la colaboración, la flexibilidad y la adaptación al cambio. Esta metodología se caracteriza por su enfoque iterativo e incremental, en el que el proyecto se divide en pequeñas tareas que se completan en ciclos cortos, conocidos como sprints (Argumanis Escalante, 2021).

La comprensión de los beneficios de esta técnica requiere un enfoque en sus características cruciales. Para esto resume las características más relevantes del enfoque Scrum, resumiendo sus elementos definidores. Con el fin de comprender mejor las ventajas competitivas de Scrum, que proporciona valor de manera ágil, como se visualiza en la Tabla 9.

Tabla 9

Características marco de trabajo ágil Scrum

Ord.	Característica	Descripción
1	Ciclos de desarrollo cortos	Scrum divide el trabajo en ciclos cortos llamados sprints, típicamente de 2 a 4 semanas, en lugar de proyectos largos y lineales. Esto permite entregar valor al cliente de forma rápida y frecuente, al tiempo que también se adapta a los cambios de una manera ágil.

Ord.	Característica	Descripción
2	Equipos autoorganizados	Los equipos de Scrum son pequeños y multidisciplinares, capaces de tomar decisiones y gestionar su trabajo independientemente. Esto fomenta la colaboración, la responsabilidad y la iniciativa.

Nota. La Tabla 9 enumera las características principales que caracterizan al marco de trabajo ágil Scrum, destacando aspectos como su enfoque iterativo (Integra, 2019).

Dentro de Scrum, los equipos multifuncionales autoorganizados se organizan en torno a tres roles específicos. Se ha creado una tabla que describe estos roles específicos para que todos puedan comprender sus responsabilidades. De esta manera, la Tabla 10 facilita la conformación de equipos utilizando esta metodología al mostrar qué funciones conlleva cada rol según el marco Scrum.

Tabla 10

Roles de Scrum

Ord.	Roles	Descripción
1	Propietario del producto (Product Owner)	Prioriza las funcionalidades del producto y representa la voz del cliente en el equipo
2	Scrum Master	Facilita el proceso de Scrum, elimina obstáculos y asegura que el equipo cumpla sus objetivos.
3	Equipo de desarrollo	Realiza el trabajo del sprint, entregando funcionalidades incrementales del producto

Nota. La Tabla 10 muestra las 3 responsabilidades centrales en los equipos ágiles Scrum: El Product Owner, el Scrum Master y el equipo de desarrollo; detallando las actividades primarias de cada uno (Palacios Jeronimo, 2021).

Scrum ofrece herramientas claras para aumentar la transparencia y la trazabilidad del progreso. Se creó una tabla que describía sus tres artefactos principales que facilitan su comprensión. La Tabla 11 contiene estos instrumentos, que permiten una mejor comprensión de cómo Scrum captura elementos de trabajo, organiza entregas y mide el progreso. Esto fomenta la inspección y adaptación continua en los equipos.

Tabla 11

Artefactos de la metodología Scrum

Ord.	Estado	Artefacto		Descripción
1	Entrada	Proporcionan información sobre el trabajo que se debe realizar	Product Backlog	Una lista priorizada de funcionalidades del producto que el equipo debe desarrollar.
2	Trabajo en curso	Representan el trabajo que se está realizando en el momento actual	Sprint Backlog	Una lista de funcionalidades que el equipo se compromete a realizar durante un sprint.
3	Salida	Representan el trabajo completado	Incremento del producto	El producto funcional que se entrega al final de cada sprint

Nota. En la Tabla 11 se describe las tres herramientas de gestión distintivas de Scrum, estos son el Product Backlog que administra requerimientos, el Sprint Backlog que organiza tareas de

implementación y el Incremento del producto como la versión operativa resultante de cada Sprint (Eslabón, 2011).

Con el propósito de comprender esta metodología de manera más completa, se ha creado una tabla que resume tanto los aspectos positivos como los problemas comunes. Se puede anticipar mejor la preparación y el compromiso necesarios de implementar Scrum de forma exitosa en cualquier tipo de proyecto al examinar estos elementos de la Tabla 12.

Tabla 12

Ventajas y Desafíos de la metodología Scrum

Ord.	Ventajas	Desafíos
1	- Enfoque iterativo e incremental	- Requiere un equipo comprometido.
2	- Colaboración y comunicación	- Puede ser difícil de implementar.
3	- Equipos más comprometidos y motivados.	
4	- Adaptación al cambio	

*Nota***.** La Tabla 12 resume los beneficios y los desafíos comunes de implementar Scrum (Drew, 2019).

Métricas de evaluación

Las métricas de evaluación son medidas empleadas para cuantificar el rendimiento de un modelo de aprendizaje automático. Se utilizan para comparar diferentes modelos y evaluar su desempeño en un conjunto de datos particular (Luisquintanilla, 2023), haciendo uso de las métricas indicadas en la Tabla 13.

Tabla 13

Principales Métricas de evaluación

Ord.	Métricas	Concepto
1	- Precisión	- Es una métrica que indica la proporción de predicciones positivas verdaderas, midiendo la eficacia del modelo para evitar falsos positivos.
2	- Accuracy o exactitud	- Es la precisión de las predicciones con un conjunto de datos de prueba. Se refiere a la proximidad de una medición o predicción al valor real, indicando su precisión y exactitud.
3	- Recall	- Conocido como sensibilidad o tasa de recuperación, este parámetro evalúa la capacidad de un modelo para identificar de manera correcta los ejemplos positivos.

*Nota***.** La Tabla 13 (Nagaesh Singh Chauhan, 2023) define las métricas de evaluación. En esat se detalla la accuracy (Jesus, 2019), la precision (Datos, 2021) y el recall (Díaz, 2020).

Capítulo III

Implementación

En este capítulo se describe el procedimiento realizado para la creación del sistema de detección de intrusiones de spoofing utilizando modelos y algoritmos de Machine Learning. Este sistema se implementó como una página web, con el propósito de detectar transacciones falsas en línea e informar al usuario que se ha realizado un robo en su cuenta del banco.

Secure Pay Guard funciona de la siguiente manera: el usuario inicia sesión en la banca web, selecciona el tipo de cuenta desde la que realiza la transacción. Elige el servicio a pagar o se agrega el identificador de un nuevo servicio, esta información ira a ser analizada por el honeypot accediendo al historial de pagos de la cuenta y este se enfoca en analizar la ubicación de la ip desde donde se solicita la transacción, monto, el identificador del servicio y almacena el puerto y el ataque que detecto. Luego se genera un comprobante, se muestra un mensaje de que el servicio se pagó con éxito y en caso de ser detectado como transacción anómala se envía una alerta sobre el pago.

Métricas de evaluación

A efectos de evaluar Secure Pay Guard se establecieron las siguientes métricas de evaluación que se muestran en la Tabla 14 con sus fórmulas respectivas.

Tabla 14

Fórmulas de Métricas de evaluación

Ord.		Métricas	Formula
1	-	Accuracy	$acurracy = \frac{VP + VN}{VP + VN + FP + FN}$
2	-	Recall	$recall = \frac{VP}{VP + FN}$
3	-	Precisión	$precision = \frac{VP}{VP + FP}$

Nota. En esta tabla se establecen las fórmulas de evaluación del sistema. Se detalla para el cálculo de accuracy (Campus Stellae, 2023), precision (Santos Marco, 2020), y recall (Jesus, 2019).

VP.- Son los valores que el algoritmo clasifica como positivos y que realmente son positivos.

VN.- Son valores que el algoritmo clasifica como negativos (0 en este caso) y que realmente son negativos.

FP.- Son valores que el algoritmo clasifica como positivo cuando realmente son negativos.

FN.- Son valores que el algoritmo clasifica como negativo cuando realmente son positivos.

Para la evaluación del algoritmo se utilizó la matriz de confusión que se muestra en la Tabla 15.

Tabla 15

Matriz de confusión

	Métricas	**Formula**
Positivos	Spoofing clasificados correctamente (VP)	Legítimos mal clasificados (FP)
Negativos	Spoofing mal clasificado (FN)	Legítimos clasificados correctamente (VN)

Nota. Formato para evaluación del sistema.

Metodología Scrum

En este trabajo se aplica la metodología scrum como marco de trabajo ágil, en los últimos años ha ganado una gran popularidad debido a su capacidad de ayudar a los equipos a ser adaptables, flexibles y eficientes en la entrega de productos de software. Se eligió esta metodología como marco de trabajo porque ayuda a los equipos a desarrollar software de forma más eficiente y efectiva.

Roles y Técnicas de Scrum

Para usar la metodología Scrum es fundamental definir y comprender los roles que desempeñan los diferentes miembros del equipo. La definición de roles en la metodología scrum, ayuda a dejar claro quién es responsable de cada aspecto del proyecto, considerando que cada miembro del equipo tiene habilidades y fortalezas únicas, la asignación de roles permite que cada persona se enfoque en las tareas que mejor puede realizar, aumentando la eficiencia y productividad del equipo en su conjunto. Estos roles con el miembro del equipo asignado se muestran en la Tabla 16.

Tabla 16

Asignación de roles de Scrum

N°.	Rol	Integrante	Descripción
01	Scrum Master	Karen Lizeth Jumbo Salcedo	Líder del equipo de Scrum
02	Product Owner	Dr. José Luis Carrillo Medina	Representa a las partes interesadas
03	Team Development	- Karen Lizeth Jumbo Salcedo - Mario Darío García Romero	Desarrollo y diseño de la aplicación
04	Auditora	- Ing. María Alexandra Corral Diaz	Auditora de los procesos de desarrollo

Nota. Roles asignados a los diferentes actores dentro del proyecto SCRUM.

Épicas

En esta metodología Scrum, el trabajo se organizó en Historias Épicas, que son grandes bloques de funcionalidad que incluyen varias historias de usuario más pequeñas. Las historias épicas se dividen en historias de usuario más pequeñas, lo que permite una gestión más efectiva utilizando técnicas y principios ágiles (Platzi, 2021). La definición de las funcionalidades implementadas se encuentra detalladas en la Tabla 17. Historias Épicas.

Tabla 17

Épicas para la implementación del sistema

ID	Título	Descripción
E. 01	Datos simulados y configuración del entorno	Desarrollar datos de usuario e historial de transacciones simulados para probar la funcionalidad de detección del sistema Secure Pay Guard
E. 02	Implementación del honeypot y detección de transacciones anómalas	Crear e implementar un honeypot especializado con ML que use datos simulados de usuarios e historiales de transacciones para detectar transacciones anómalas.
E. 03	Creación del sistema bancario y gestión de notificación.	Integrar un proceso de emisión de alerta que informe a los usuarios cuando se complete una transacción.

Nota. Épicas para la funcionalidad del sistema.

Esto significa que las Historias Épicas se dividen en historias de usuario más manejables, que son priorizadas y distribuidas en el equipo de desarrollo conforme el avance de cada Sprint. Esto permite un enfoque incremental y adaptativo en la implementación, además, se estimó cada historia de usuario y se monitoreó su progreso (Codeicus, 2023).

Historias de usuario

Las historias de usuario son descripciones breves de funcionalidades del producto desde el punto de vista del usuario, siendo claras, concisas y entregables, ayudan a asegurar que el equipo se centre en las necesidades del cliente

En Scrum, las historias de usuario se utilizan en el contexto de la ingeniería de requisitos ágiles como una herramienta de comunicación que combina los puntos fuertes de la comunicación escrita y verbal. Describe una característica de software desde la perspectiva del

usuario en una o dos frases. Estas historias de usuarios se centran en las necesidades o problemas que tienen como objetivo construir el sistema. También simplifica la gestión de los requisitos reduciendo el número de documentos oficiales y el tiempo necesario (Scrum Manager, 2022).

En cada iteración o sprint, las historias de usuario son una herramienta fundamental que detalla una funcionalidad específica del sistema desde la perspectiva del usuario final. Cada historia de usuario representa una pieza discreta de valor que puede ser desarrollada, probada y desplegada de forma independiente (Atlassian, 2023). Es por esto que en la Tabla 18 se muestra la distribución de historias de usuario en relación a las épicas.

Tabla 18

Historias de usuario en relación a las épicas.

ID Épica	Nombre	Rol	Característica / Funcionalidad	Razón / Resultado
E.02	H.U. 01	Como usuario	Quiero que el sistema use un honeypot que detecte las transacciones anómalas.	Para probar la funcionalidad de detección en el sistema bancario simulado
E.02	H.U. 02	Como usuario	Quiero que el equipo de desarrollo utilice los datos simulados de usuarios e historial de transacciones	Para garantizar mi seguridad y prevenir fraudes.
E.03	H.U. 03	Como usuario	Quiero que el sistema emita alertas cuando se culmina la transacción	Para informar al usuario en caso de que se realice alguna transacción fraudulenta.

Nota. Historias de usuario con el nombre, rol, característica y razón.

Sprint Planning

Dentro de la metodología Scrum, es crucial definir la planificación del sprint, ya que permite una planificación detallada y colaborativa de tareas y actividades que se lleven a cabo durante un sprint específico. Este proceso asegura que todos en el equipo entienden y

colaboran para alcanzar los objetivos, poniendo las bases para un sprint eficiente y productivo donde el equipo trabaja juntos para entregar un incremento funcional del software al final de un período fijado (West Dave, 2022).

Durante la Planificación de Sprint del proyecto, se presenta una lista de historias de usuarios a implementar a lo largo del proyecto. Con un calendario definido y el número de sprint al que pertenece cada historia, como se muestra en la Tabla 19.

Tabla 19

Sprint planning del proyecto

ID	Nombre	Estimación (días)	Fecha de inicio	Fecha de fin	Sprint
01	E. 01 H.U. 01 H.U. 02	15	20/11/2023	19/12/2023	Creación del dataset y modelo de Machine Learning Implementación del Honeypot
02	H.U. 03	15	20/12/2023	15/01/2024	Desarrollo de Sistema bancario simulado.

Nota. Sprint Planning de cada historia de usuario con su cronograma.

Arquitectura del sistema

La arquitectura de un sistema se refiere a la estructura fundamental y las interconexiones entre los diversos componentes de un software y sus características observables (KeepCoding, 2022).

Arquitectura en capas

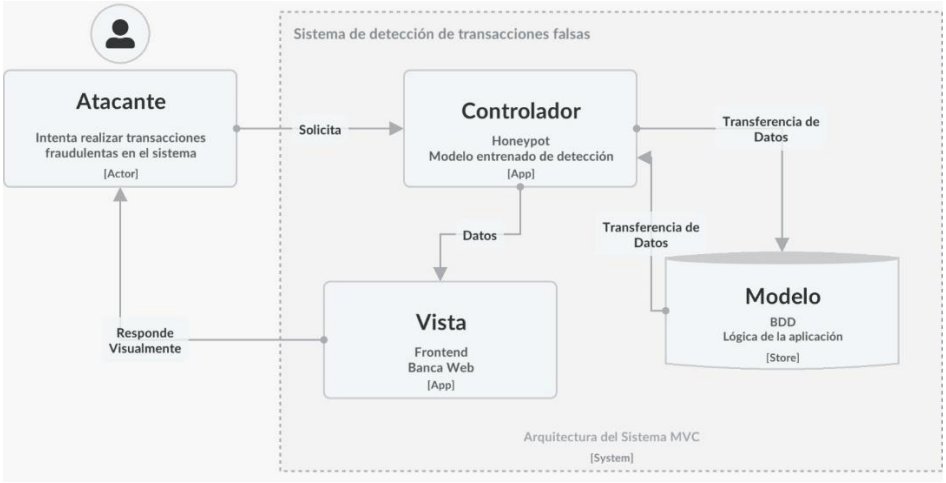
La arquitectura de capas es un modelo de diseño de software que se centra en separar las funcionalidades del sistema en capas o niveles, donde cada capa es responsable de un conjunto específico de tareas y comunica con otros niveles (Durán, 2023).

El sistema de detección de intrusiones Secure Pay Guard se desarrolla utilizando una arquitectura en capas siguiendo el patrón MVC (Model-View-Controller). Este patrón simplifica el desarrollo de aplicaciones complejas. También ayuda a separar el código frontend y backend en componentes distintos, lo que simplifica la gestión y permite que se realicen cambios en ambos lados sin interferencia (Hernandez Rafael, 2021).

Se definió la arquitectura en Modelo, Vista y Controlador como se ilustra en la Figura 1.

Figura 1

Arquitectura MVC del sistema



Nota. Representación gráfica de la arquitectura MVC. Dividida en: i) Modelo, ii) Vista, iii) Controlador.

Tecnología aplicada de la arquitectura

Son herramientas, marcos y lenguajes de programación utilizados en el desarrollo del sistema Secure Pay Guard. La tecnología utilizada en cada componente se explica a continuación:

Modelo. El modelo especifica los datos que la aplicación manejará. Si los datos cambian de estado, el modelo normalmente notifica la vista para que la interfaz de usuario se actualice en consecuencia. En ocasiones, el modelo también notificará al controlador si se necesita una lógica diferente para gestionar la vista actualizada debido a cambios en los datos (WebDocs, 2023).

MongoDB. Es un sistema para la gestión de datos no relacionales o NoSQL. Este es un modelo orientado a documentos que almacena datos en BSON, una representación binaria de JSON, y no utiliza tablas como sistemas SQL (Web development, 2022).

MySQL. Es el sistema de gestión de bases de datos relacionales más utilizado hoy en día debido a su naturaleza de código abierto (Robledano Angel, 2019).

Vista. La vista contiene el código responsable de generar interfaces de usuario que representan al usuario. El propósito principal es mostrar los datos, sin incluir ninguna otra lógica de aplicación. Las vistas son responsables de la presentación y visualización de la información a los usuarios (Desarrollo Web, 2023).

JavaScript. Es un lenguaje de programación alto nivel, interpretado y orientado a objetos, especialmente utilizado en el desarrollo web para ofrecer interactividad y dinamismo a las páginas web.(Coppola, 2023).

React. Es una de las bibliotecas JavaScript más populares para el desarrollo de aplicaciones web. Contiene una colección de fragmentos de código JavaScript reutilizables utilizados con el fin de crear interfaces de usuario llamadas componentes (A, 2020).

Controlador. En el controlador se localizan los componentes encargados de procesar las interacciones del usuario con la aplicación. Son responsables de ejecutar la lógica empresarial para interrogar o manipular el modelo de datos cuando sea necesario, en

respuesta a las acciones del usuario. También elige que vista se debe desplegar al usuario en cada momento, dependiendo del estado actual de la aplicación y la operación solicitada por el usuario. Es decir actúa como intermediario entre el usuario y el sistema (Aguilar, 2019).

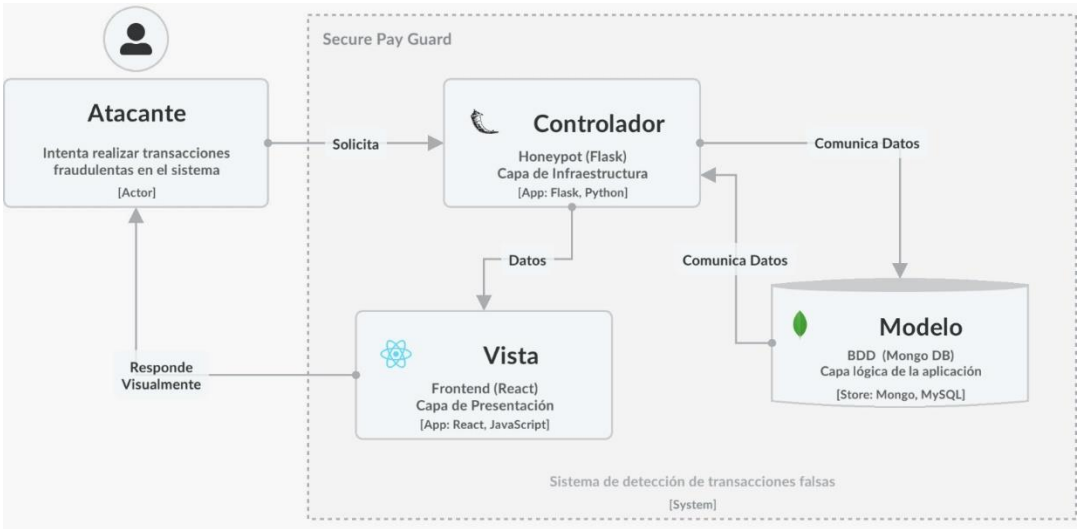
Python. Es un lenguaje de programación de código abierto, orientado a objetos. Tiene una sintaxis que permite leerlo de manera semejante a como se lee el inglés y es fácil de interpretar tanto por los usuarios como por la máquina (Londoño, 2023).

Flask. Es un microframework Python diseñado para simplificar el desarrollo de aplicaciones web siguiendo el patrón MVC. Incluye el motor de plantillas Jinja y una biblioteca llamada tool, ofrece la posibilidad de integrar funciones de terceros (Desarrollo Web, 2023).

Se visualiza los componentes de la arquitectura del sistema de detección de intrusiones, con las tecnologías que se utilizara en cada uno en la Figura 2.

Figura 2

Tecnologías aplicadas en los componentes de la arquitectura MVC



Nota. En el componente de Modelo se utiliza las herramientas Mongo DB y MySQL para la base de datos. En el componente de Vista se utiliza el lenguaje de programación JavaScript y el

framework de React y en el componente de Controlador se utiliza el lenguaje de programación Python y el framework Flask.

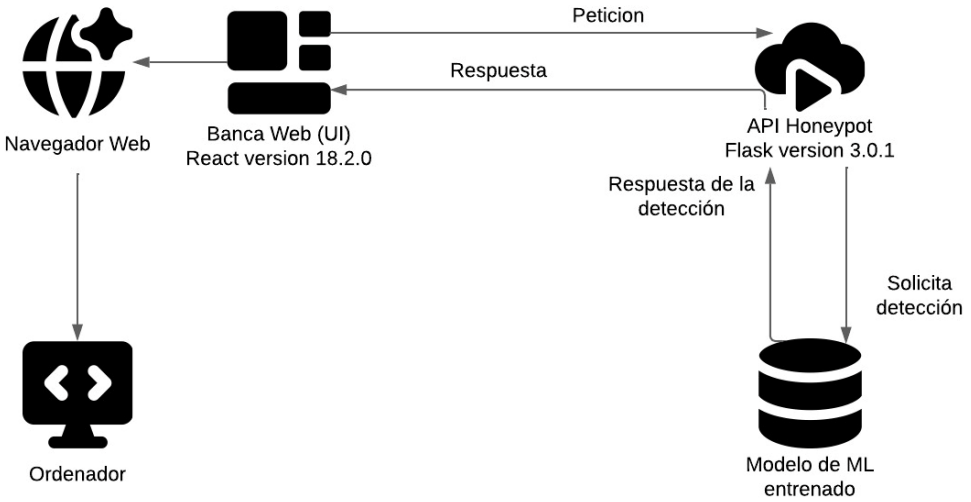
Arquitectura física

El proceso que se sigue dentro de la arquitectura física considera que el usuario inicia sesión en el navegador de la banca web y solicita una transacción de pago. Esta solicitud es analizada por un honeypot y se solicita la detección al modelo de Machine Learning entrenado. La respuesta de la detección se envía al honeypot y se muestra al usuario en caso de ser detectada como anómala.

La Figura 3 muestra la arquitectura física que se usó para desarrollar el sistema de detección de intrusiones Secure Pay Guard.

Figura 3

Arquitectura Física del sistema



Características del Hardware del Sistema, para la ejecución y desarrollo del proyecto se utilizó un procesador AMD Ryzen 3 3200U con una disponibilidad de memoria RAM de 16GB y un sistema operativo de 64 bits.

Implementación de algoritmo y modelos de Machine Learning para prevención de intrusos

Sprint 01: Creación del dataset, modelo de Machine Learning e Implementación del Honeypot

Previo a la implementación se tiene consideraciones de características con el objetivo de la creación del dataset de transacciones financieras para validar y entrenar el sistema de intrusiones. El primer conjunto de datos, contiene información detallada sobre transacciones bancarias, e incluyen elementos importantes como el identificador de la transacción, la fecha y hora en que se realizó, el identificador de pago y el monto de la transacción. Estas características se toman porque representan los datos fundamentales de las transacciones, que se pueden utilizar con el objetivo de identificar patrones de comportamiento normal y anómalo.

En cuanto a la elección de los algoritmos de Random Forest y Regresión Logística se enfoca en la capacidad del modelo para la clasificación y predicción de grupos de datos amplios. Debido a su capacidad de aprender de datos de entrenamiento y generalizar patrones de identificación de actividades anómalas o maliciosas en nuevos datos, estos algoritmos de aprendizaje automático son ampliamente utilizados en la detección de intrusiones. A través de la aplicación de ambos algoritmos a los conjuntos de datos proporcionados, el sistema de detección de intrusiones podrá aprender de manera efectiva los patrones de comportamiento normal y anómalo en el contexto transacciones en línea.

En el Sprint 1 se lleva a cabo el desarrollo de la épica 1, en esta se detalla la generación de datos simulados y el historial de transacciones bancarias. Considerando que es esencial para el desarrollo del sistema porque proporciona la base de datos necesaria del entrenamiento y prueba del sistema de detección de intrusiones. Los criterios de aceptación definidos en esta épica se visualizan en la Tabla 20.

Tabla 20

Épica detallada 01

Épica	**Numero:** E. 01
Nombre: Datos simulados y configuración de entorno	
Programadores responsables: Mario García, Karen Jumbo	
Descripción: • Desarrollar datos de usuario e historial de transacciones simulados para probar la funcionalidad de detección del sistema Secure Pay Guard	
Validación (Criterios de Aceptación): • Los datos de usuario simulados incluyen nombre, ubicación, monto, y entidades de transacción. • El historial de transacciones simulado incluye una amplia gama de transacciones a entidades, como luz, agua, teléfono e internet. • Los datos simulados se pueden usar por el sistema Secure Pay Guard para simular las actividades de usuarios y las transacciones. • Los datos simulados en un entorno de pruebas se utilizan para validar con éxito la funcionalidad de detección del sistema.	

Nota. Detalle de la primera funcionalidad implementada por el equipo de desarrollo.

Así como también, se llevó a cabo la Historia de usuario 1 y 2, donde se muestra cada detalle para el desarrollo del honeypot como: el número de la historia de usuario, el usuario que la realiza, nombre de la historia, prioridad de esta historia del negocio, riesgo en desarrollo, días estimados a invertir, el número de iteraciones asignadas, los programadores que son responsables, la descripción de la historia y los criterios de aceptación teniendo en cuenta que se cumpla la historia de usuario. Como se muestra en la Tabla 21 y 22.

Tabla 21

Historia de Usuario detallada 01

Historias de Usuario	
Número: H.U. 01	**Usuario:** Usuario de la entidad bancaria
Nombre historia: Implementación del honeypot para detectar transacciones anómalas	
Prioridad de negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados (días): 14	**Iteración asignada:** 1
Programadores responsables: Karen Jumbo, Mario García	
Descripción: • Como usuario, quiero que el sistema use un honeypot que detecte las transacciones anómalas para aumentar la seguridad de mis transacciones financieras.	
Validación (Criterios de aceptación): • El sistema debe implementar un honeypot efectivo que pueda detectar transacciones anómalas basadas en patrones predefinidos. • El honeypot debe ser capaz de registrar y notificar al usuario cualquier actividad sospechosa o no autorizada. • Se debe realizar pruebas exhaustivas para garantizar que el honeypot funcione de manera eficaz y no genere falsos positivos ni negativos.	

Nota. Detalles de la historia de usuario 01

Tabla 22

Historia de usuario detallada 02

Historias de Usuario	
Número: H.U. 02	**Usuario:** Usuario
Nombre historia: Uso de datos simulados de usuarios e historial de transacciones.	
Prioridad de negocio: Alta	**Riesgo en desarrollo:** Alto
Puntos estimados (días): 15	**Interacción asignada:** 1
Programadores responsables: Mario García, Karen Jumbo	
Descripción: • Como usuario, necesito que el equipo de desarrollo, utilice los datos simulados de usuarios e historiales de transacciones para propósitos de desarrollo, pruebas y entrenamiento de modelos de Machine Learning.	
Validación (Criterios de aceptación): • Se debe simular diferentes perfiles de usuarios, desde jóvenes adultos hasta personas mayores, con diferentes patrones de comportamiento financiero. • Se deben simular historiales de transacciones como transferencias y pagos. • Las herramientas para la generación dinámica de datos simulados deben permitir ajustar parámetros como la cantidad de usuarios, la duración del historial de transacciones, y la distribución de transacciones por tipo.	

Nota. Detalles de la historia de usuario 02

***Sprint backlog*.** Es una lista de tareas a realizar durante el sprint, en cada una de ellas se estima el tiempo necesario para completarlas, ayudando a que el equipo de trabajo se concentre en cumplir los objetivos planteados (Miro, 2020).

El Sprint Backlog 01 en la Tabla 23 detalla la información del Sprint 02 como las tareas realizadas, persona responsable de su ejecución, planificación para la ejecución del Sprint, el tiempo estimado en horas de cada tarea y el estado actual.

Tabla 23

Sprint Backlog 01

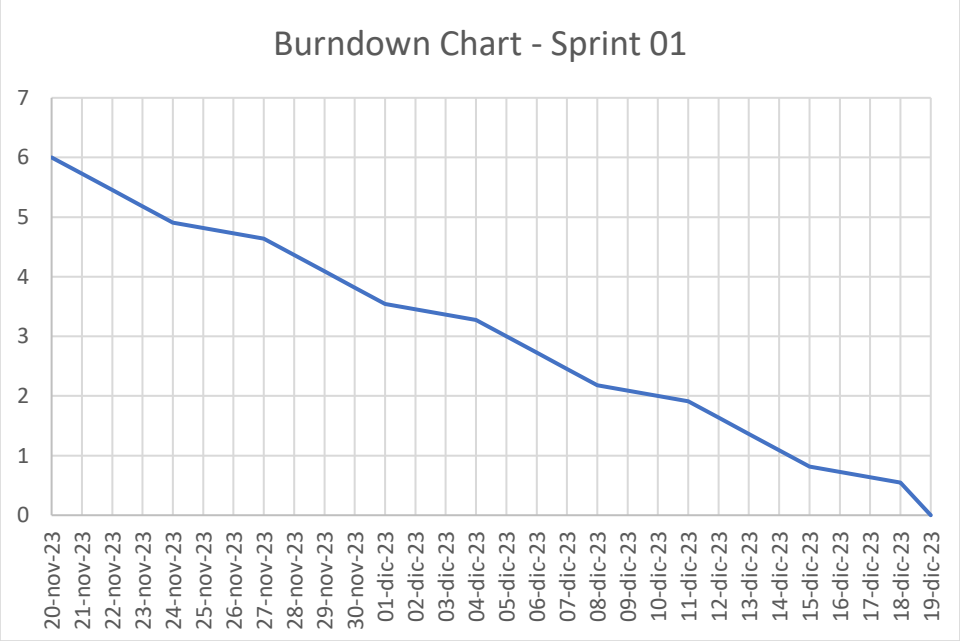
Sprint 01						
Fecha inicio:			***Fecha Fin:***		***Jornada***	
20/11/2023			19/12/2023		*2 horas*	
H. U	***Tareas***	***Horas***	***Fecha Inicio***	***Fecha Fin***	***Responsable***	***Estado***
E. 01	*Generación del dataset*	*14*	*20/11/2023*	*26/11/2023*	*Karen Jumbo Mario García*	*Finalizado*
H.U. 01	*Creación del Modelo*	*10*	*27/11/2023*	*01/12/2023*	*Karen Jumbo Mario García*	*Finalizado*
H.U. 01	*Pruebas del Modelo utilizando los datos de prueba*	*6*	*02/12/2023*	*04/12/2023*	*Karen Jumbo Mario García*	*Finalizado*
H.U. 02	*Diseño del Honeypot*	*4*	*05/12/2023*	*06/12/2023*	*Karen Jumbo Mario García*	*Finalizado*
H.U. 02	*Desarrollo del Honeypot*	*16*	*07/12/2023*	*14/12/2023*	*Karen Jumbo Mario García*	*Finalizado*
H.U. 02	*Configuración del entorno del Honeypot*	*8*	*15/12/2023*	*19/12/2023*	*Karen Jumbo Mario García*	*Finalizado*

Nota. La tabla muestra las actividades que se realizó durante el Sprint 01, el responsable, horas invertidas en cada tarea.

***Burndown chart*.** La figura 4 muestra la tabla de desarrollo del primer sprint, mostrando el progreso del trabajo durante el período asignado para este ciclo inicial del proyecto. Este período abarca del 20 de noviembre al 18 de diciembre de 2023, como se detalla en la Tabla 10. En el eje horizontal (X) del gráfico, se indican las fechas de inicio y finalización de este período, proporcionando una visualización clara del progreso diario dentro del marco de tiempo establecido. El número total de días asignados a este sprint es de 21, con días laborables de 8 horas, obteniendo un total de 168 horas, que se representa en el eje vertical (Y) como el punto máximo de partida para las horas de trabajo. A medida que avanza el tiempo, se espera que las horas se reduzcan constantemente hasta llegar a cero, lo que señala la conclusión exitosa del Sprint 01 de acuerdo con los objetivos previstos.

Figura 4

Burndown chart para el Sprint 01



***Resultados del sprint*.** El enfoque de este sprint estuvo centrado en satisfacer los requerimientos de la épica 1, donde se logra generar un conjunto sólido de datos financieros

simulados y sentar las bases de nuestro modelo para la detección de fraudes. Acompañado del desarrollo de herramientas capaces de recrear historiales de transacciones ficticias en relación a una variedad de usuarios ficticios, cada uno con atributos y comportamientos distintos, en lo que respecta a la generación de datos.

El éxito de este sistema depende de la tabla de características creada para cada usuario. Esta tabla permite una monitorización detallada de las transacciones, lo que facilita la identificación de patrones anómalos que puedan indicar intentos de intrusión o fraude, en relación con transacciones legitimas analizadas. Los honeypots atraen a los atacantes al simular transacciones falsas dentro de un entorno controlado, sin poner en peligro los activos reales de la organización simulada. Este método no solo mejora la capacidad de respuesta ante incidentes de seguridad, sino que también ayuda a crear una base de datos de inteligencia de amenazas, permitiendo una respuesta más rápida y precisa a las actividades sospechosas, lo que hace que las transacciones en línea simuladas sean más precisas en relación de los usuarios. En la Tabla 24 se visualiza las características que el modelo utiliza para la detección de transacciones anómalas

Tabla 24

Características del usuario por transacción

Característica	**Detalle**
Propietario	Identificación del titular de cuenta implicada en la transacción.
Servicio	Especifica el tipo de servicio: Luz, agua, teléfono, internet
Código de pago	Identificador único, asignado a cada transacción para cada pago de servicio realizado

Característica	Detalle
Cédula	Número de identificación del propietario de la cuenta
Fecha / Hora de pago	Marca la actividad temporal de la transacción
Dirección IP	Dirección del dispositivo utilizado en Internet
Monto	La cantidad de dinero involucrada para cada transacción de pago realizado por el usuario
Estado	Detecta el estado de la transacción que puede ser, normal o anómala.

Nota. Se detalla características para la aplicación dentro del sistema simulado, se enfoca en un entorno analizado altamente realista en un contexto controlado.

Con el fin garantizar la replicación de su complejidad en el entorno simulado, se crearon estos perfiles de usuarios analizando datos reales y extrayendo patrones sociodemográficos y conductuales. Como resultado, nuestra selección de simulación crea historiales dinámicos que incluyen una variedad de transacciones que pueden ajustar parámetros como los gastos, los montos detalles de pago y los detalles contextuales.

Nuestro modelo de detección de fraudes se entrena sobre esta base sólida de datos sintéticos. En este sentido, se muestra progresos en la construcción e implementación inicial de una red neuronal que tiene como objetivo detectar intrusiones y comportamientos inusuales en los historiales de transacciones. El entrenamiento se llevó a cabo en dos modelos de Machine Learning: Random Forest y regresión logística con el fin de comparar su eficacia con la red neuronal previamente mencionada. Después de una evaluación exhaustiva del rendimiento que involucra métricas como la precisión, la exactitud y el recall, el modelo Random Forest demostró ser adecuado para el sistema de detección.

El enfoque del sprint 01 también se centró en la implementación de un módulo básico que cumplió las especificaciones descritas en la historia de usuario asignada. El honeypot consta de un controlador que gestiona las interacciones del usuario con un modelo de detección de anomalías. Este modelo compara cada transacción con límites predeterminados basados en el importe, el tipo de operación, la ubicación y otros factores. El sistema registra el evento y notifica al usuario cuando detecta una actividad que excede los parámetros válidos definidos.

Aunque actualmente utiliza la lógica segmentada, se está poniendo las bases para su transición a un sistema más dinámico en futuras iteraciones. Mediante pruebas rigurosas, el módulo se verifica para detectar comportamientos inusuales. Es necesario ampliar el rango de casos, implementar detecciones más inteligentes y contextuales, y mejorar las notificaciones con información para ayudar al usuario a tomar decisiones. Estas mejoras tienen como objetivo mejorar la eficacia y la seguridad de nuestro sistema integral de prevención de intrusiones.

Implementación código

Se creó un código para producir un registro de transacciones simuladas en relación a diferentes servicios como agua, electricidad, teléfono e internet vinculados a los nombres de las personas de 2017 a 2024. Una vez creadas las transacciones, se almacenan en un archivo CSV llamado transacciones.csv. Este archivo permite la personalización simple de los tipos de transacciones, las frecuencias y las cantidades teniendo en cuenta de replicar diversos escenarios.

A efectos de generar las fechas se creó una función que toma un año como entrada y devuelve una lista de fechas aleatorias dentro de ese año, una fecha por cada mes. Implementando una función que genera y devuelve una IP simulada con formato ecuatoriano. Esta implementación se muestra en la Figura 5.

Figura 5

Generación del dataset

```
import random
import pandas as pd
from faker import Faker
from datetime import datetime

fake = Faker()


# Función para generar las IPs ecuatorianas
def generar_ip_ecuatoriana():
    ip_base = '186.105.'
    tercer_octeto = random.randint(0, 255)
    cuarto_octeto = random.randint(0, 255)
    return ip_base + str(tercer_octeto) + '.' + str(cuarto_octeto)


# Función para generar cédulas ecuatorianas válidas
def generar_cedula_ecuatoriana():
    primeros_dos_digitos = random.randint(1, 24)
    cedula = f"{primeros_dos_digitos:02d}"
    for _ in range(7):
        cedula += str(random.randint(0, 9))
    total = sum(int(cedula[x]) * (2 if x % 2 else 1) for x in range(9))
    digito_verificador = (10 - (total % 10)) % 10
    return cedula + str(digito_verificador)


# Función para determinar si una transacción es anómala
def es_anomala(hora, codigo, ip):
    condiciones_anomalas = 0

    if hora < 6 or hora > 22:
        condiciones_anomalas += 1
    if codigo != 'codigo_de_pago_habitual':
        condiciones_anomalas += 1
    if not ip.startswith('186.105.'):
        condiciones_anomalas += 1

    return condiciones_anomalas >= 2


# Generar 500 nombres de personas
personas = [fake.name() for _ in range(500)]

# Generar datos de transacciones para cada persona
```

```
transacciones = []
for persona in personas:
    # Generar una cédula única para cada persona
    cedula = generar_cedula_ecuatoriana()

    # Generar un código de servicio único para cada persona
    codigos_servicio = {
        'agua': fake.uuid4(),
        'luz': fake.uuid4(),
        'telefono': fake.uuid4(),
        'internet': fake.uuid4()
    }

    # Generar transacciones para cada año y mes
    for year in range(2017, 2024):
        for month in range(1, 13):
            # Generar transacciones para cada tipo de servicio
            for servicio, codigo in codigos_servicio.items():
                # Generar el monto basado en el servicio
                if servicio == 'agua':
                    monto = round(random.uniform(5, 20), 2)
                elif servicio == 'luz':
                    monto = round(random.uniform(15, 35), 2)
                elif servicio == 'telefono':
                    monto = round(random.uniform(30, 60), 2)  # Ajusta
el rango según lo requerido
                elif servicio == 'internet':
                    monto = round(random.uniform(10, 50), 2)

                # Generar la IP ecuatoriana
                ip = generar_ip_ecuatoriana()

                # Generar la fecha y hora de la transacción
                fecha_hora = datetime(year, month, random.randint(1,
28), random.randint(0, 23), random.randint(0, 59))

                # Determinar el estado de la transacción (normal o
anómala)
                estado = 'anomalo' if es_anomala(fecha_hora.hour,
codigo, ip) else 'normal'

                # Agregar la transacción
                transaccion = [persona, servicio, codigo, cedula,
                               fecha_hora.strftime("%Y-%m-%d
%H:%M:%S"), ip, estado, monto]
                transacciones.append(transaccion)

# Crear un DataFrame con los datos de transacciones
```

```
df = pd.DataFrame(transacciones, columns=['propietario', 'servicio',
'codigo de pago', 'cedula',
                                          'fecha_hora_de_pago',
'direccion_ip', 'estado', 'monto'])

# Guardar el DataFrame en archivos CSV separados para entrenamiento y
prueba
df_entrenamiento = df.sample(frac=0.8, random_state=42)
df_prueba = df.drop(df_entrenamiento.index)

df_entrenamiento.to_csv('transacciones_entrenamiento.csv', index=False)
df_prueba.to_csv('transacciones_prueba.csv', index=False)
```

Se implementó un código con Python que realiza un análisis de datos y evaluación de modelos de aprendizaje automático utilizando Pandas, scikit-learn y matplotlib. Se cargó un conjunto de datos de un archivo csv llamado transacciones_entrenamientos.csv en un DataFrame de Pandas. Entonces se inicializa y entrena dos modelos de machine learning: Random Forest y Regresión Logística utilizando los conjuntos de entrenamiento, este código se muestra en la Figura 6.

Figura 6

Modelo de Machine Learning

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score,
recall_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Cargar los datos de entrenamiento
df_entrenamiento = pd.read_csv('transacciones_entrenamiento.csv')

# Convertir la columna 'fecha_hora_de_pago' a tipo datetime
df_entrenamiento['fecha_hora_de_pago'] =
pd.to_datetime(df_entrenamiento['fecha_hora_de_pago'])

# Crear nuevas características para año, mes, día y hora
```

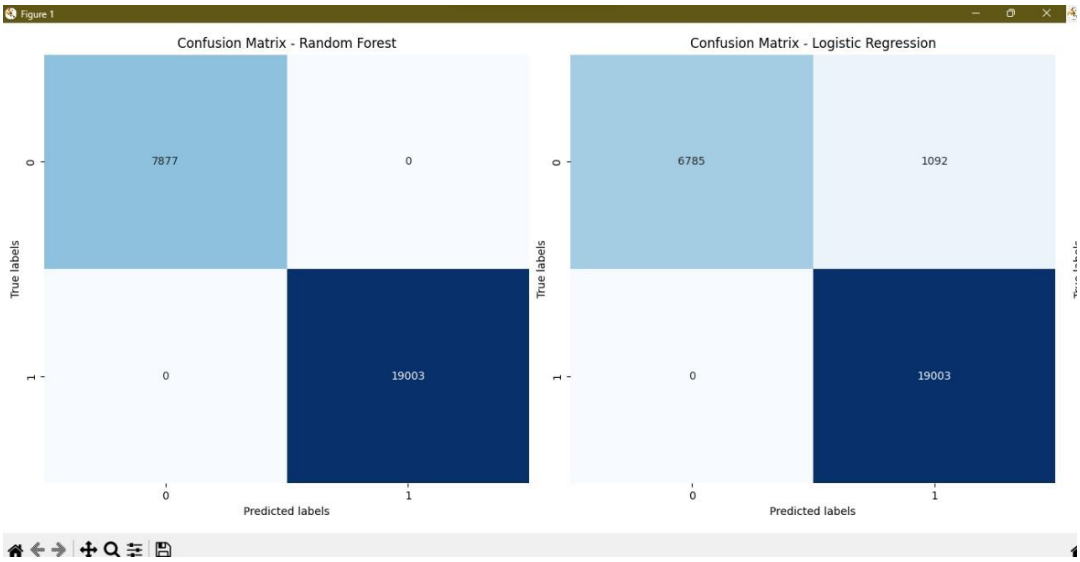
```
df_entrenamiento['año'] =
df_entrenamiento['fecha_hora_de_pago'].dt.year
df_entrenamiento['mes'] =
df_entrenamiento['fecha_hora_de_pago'].dt.month
df_entrenamiento['dia'] = df_entrenamiento['fecha_hora_de_pago'].dt.day
df_entrenamiento['hora'] =
df_entrenamiento['fecha_hora_de_pago'].dt.hour

# Seleccionar características y etiquetas
X = df_entrenamiento[['hora', 'monto']]
y = df_entrenamiento['estado']

# Dividir los datos en conjunto de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Inicializar y entrenar el modelo de Random Forest
random_forest_model = RandomForestClassifier(n_estimators=100,
random_state=42)
random_forest_model.fit(X_train, y_train)

# Inicializar y entrenar el modelo de Regresión Logística
logistic_regression_model = LogisticRegression(random_state=42)
logistic_regression_model.fit(X_train, y_train)

# Predecir las etiquetas en el conjunto de prueba
y_pred_random_forest = random_forest_model.predict(X_test)
y_pred_logistic_regression = logistic_regression_model.predict(X_test)

# Calcular métricas para Random Forest
accuracy_rf = accuracy_score(y_test, y_pred_random_forest)
precision_rf = precision_score(y_test, y_pred_random_forest,
pos_label='anomalo')
recall_rf = recall_score(y_test, y_pred_random_forest,
pos_label='anomalo')
confusion_matrix_rf = confusion_matrix(y_test, y_pred_random_forest)

# Calcular métricas para Regresión Logística
accuracy_lr = accuracy_score(y_test, y_pred_logistic_regression)
precision_lr = precision_score(y_test, y_pred_logistic_regression,
pos_label='anomalo')
recall_lr = recall_score(y_test, y_pred_logistic_regression,
pos_label='anomalo')
confusion_matrix_lr = confusion_matrix(y_test,
y_pred_logistic_regression)

# Mostrar las métricas y la matriz de confusión para Random Forest
print("Random Forest:")
print(f"Accuracy: {accuracy_rf:.4f}")
```

```
print(f"Precision: {precision_rf:.4f}")
print(f"Recall: {recall_rf:.4f}")
print("Confusion Matrix:")
print(confusion_matrix_rf)

# Visualizar la matriz de confusión para Random Forest
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_matrix_rf, annot=True, cmap='Blues', fmt='g',
cbar=False)
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix - Random Forest')
plt.show()

# Mostrar las métricas y la matriz de confusión para Regresión
Logística
print("\nLogistic Regression:")
print(f"Accuracy: {accuracy_lr:.4f}")
print(f"Precision: {precision_lr:.4f}")
print(f"Recall: {recall_lr:.4f}")
print("Confusion Matrix:")
print(confusion_matrix_lr)

# Visualizar la matriz de confusión para Regresión Logística
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_matrix_lr, annot=True, cmap='Blues', fmt='g',
cbar=False)
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix - Logistic Regression')
plt.show()
```

La Figura 7 muestra la matriz de confusión, clasificando los reales positivos, falsos positivos, reales negativos y falsos negativos como se mencionó en la Tabla 2, usando los datos de los dos escenarios.

Figura 7

Matriz de confusión



Nota. Matriz de confusión realizada con Python del segundo escenario con 40 000 características.

En la Tabla 25, se visualiza los resultados de la ejecución del código que se muestra en la Figura 5, los cuales fueron analizados con las métricas de evaluación mencionadas, estas miden el rendimiento de los algoritmos de Machine Learning según las métricas de: accuracy, precision, recall. Estas pruebas se realizan con el dataset que se genera con el algoritmo de la Figura 6, que contiene un historial de transacciones de diferentes usuarios de una entidad bancaria simulada.

Primero se valida la métrica accuracy para calcular el porcentaje de transacciones bancarias clasificados correctamente como Spoofing, en relación con el total de datos de entrenamiento, en el primer escenario con 25 000 características se obtuvo un valor de 98,68% con el modelo Random Forest. Mientras que en el segundo escenario con 40 000 características se obtuvo 100% como mejor resultado de accuracy del modelo Random Forest.

Con la métrica de precisión se obtuvo el porcentaje de transacciones bancarias anómalas, es decir detectados con spoofing con respecto al resto de transacciones bien

clasificadas como spoofing, se utiliza las mismas características que en la métrica de accuracy, con el primer escenario con 25 000 características se obtuvo la precisión de 99,68% con el algoritmo de Random Forest. Mientras que en el segundo escenario con 40 000 características se obtuvo 100% de precisión con el algoritmo Random Forest. Los resultados obtenidos en los dos escenarios indican que las 40 000 características son las que aportan más en la detección de spoofing en transacciones bancarias.

La métrica Recall determina el porcentaje de transacciones bancarias con spoofing correctamente identificados en relación con el total de ejemplos de entrenamiento de transacciones bancarias, se realizó el mismo procedimiento que con las métricas accuracy y precisión, obteniendo en el primer escenario con 25 000 características de Recall 100% con el modelo Random Forest. Con el segundo escenario con 40 000 características se obtuvo 100% de Recall con el modelo Random Forest.

Tabla 25

Resultados de pruebas del modelo y algoritmo implementado primer escenario

Ord	Características	Algoritmos/Modelos	Accuracy	Recall	Precision
1	25 000	Random Forest	0.98679	1.0000	0.99679
2		Regresión Logística	0.9389	1.0000	0.8476

Nota. Resultados primer escenario con las métricas de evaluación a los modelos de Machine Learning.

Tabla 26

Resultados de pruebas del modelo y algoritmo implementado segundo escenario

Ord	Características	Algoritmos/Modelos	Accuracy	Recall	Precision
1	40 000	Random Forest	1.0000	1.0000	1.0000
2		Regresión Logística	0.9594	1.0000	0.8614

Nota. Resultados segundo escenario con las métricas de evaluación a los modelos de Machine Learning.

Implementación del sistema bancario

Sprint 02. Desarrollo de Sistema bancario simulado

Para el desarrollo del presente Sprint, la historia de usuario 3 especifica los detalles con el propósito de generar las alertas en caso de que el sistema culmine una transacción, en donde se encuentra: el nombre de la historia de usuario, prioridad, los encargados del desarrollo, el riesgo y la iteración asignada. Como se muestra en la Tabla 27.

Tabla 27

Historia de usuario detallada 03

Historias de Usuario	
Número: H.U. 03	**Usuario:** Usuario de la entidad bancaria
Nombre historia: Emisión de alertas por transacciones anómalas	
Prioridad de negocio: Alta	**Riesgo en desarrollo:** Moderado
Puntos estimados (días): 14	**Iteración asignada:** 1
Programadores responsables: Karen Jumbo, Mario García	
Descripción: • Como usuario, quiero que el sistema emita alertas cuando se detecten transacciones anómalas para que pueda tomar medidas rápidas y asegurar la seguridad de mis transacciones financieras.	
Validación (Criterios de aceptación): • El sistema debe generar una alerta visual para notificar al usuario y al equipo de seguridad, cuando se detecte una transacción anómala.	

- Las alertas deben contener información relevante sobre la transacción anómala detectada, como el tipo de transacción, hora, ubicación IP.
- Se debe realizar pruebas exhaustivas para garantizar que las alertas se emitan de manera oportuna y precisa.

Nota. Detalles de la historia de usuario 03

*Sprint backlog***.** El Sprint Backlog 02 detalla las tareas realizadas durante el sprint, la persona responsable de cada una, la información sobre la planificación para el desarrollo y ejecución del sprint, estimación del tiempo invertido y el estado actual de cada tarea.

Tabla 28

Sprint Backlog 02

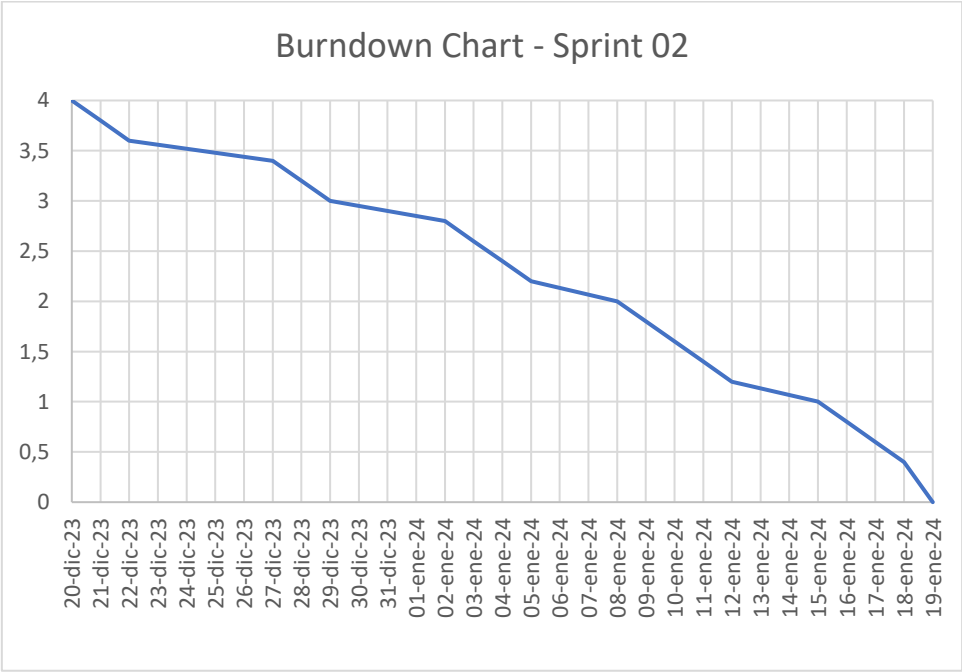
Sprint 02						
Fecha inicio:			***Fecha Fin:***		***Jornada***	
20/12/2023			15/01/2024		*4 Horas*	
H.U	***Tareas***	***Horas***	***Fecha Inicio***	***Fecha Fin***	***Responsable***	***Estado***
H.U. 03	*Desarrollo Back - end*	*32*	*20/12/2023*	*27/12/2023*	*Karen Jumbo Mario García*	*Finalizado*
H.U. 03	*Desarrollo Front - end*	*28*	*02/01/2023*	*08/01/2023*	*Karen Jumbo Mario García*	*Finalizado*
H.U. 03	*Integración de funcionalidades*	*16*	*09/01/2024*	*12/01/2024*	*Karen Jumbo Mario García*	*Finalizado*
H.U. 03	*Pruebas de funcionalidad*	*12*	*13/12/2023*	*16/01/2024*	*Karen Jumbo Mario García*	*Finalizado*

Nota. La tabla muestra las actividades realizadas durante el sprint 02, la persona encargada, las horas gastadas y el esfuerzo para cada tarea.

***Burndown chart*.** La figura 6 muestra el gráfico de los resultados del segundo sprint, que proporciona una visión general detallada de los progresos realizados durante el período de tiempo especificado para la segunda fase de este proyecto. Esta sección se desarrolló el período comprendido entre el 20 de diciembre de 2023 y el 16 de enero de 2024, como se detalla en la Tabla 28. El eje horizontal (X) del gráfico captura el rango de fechas durante esta etapa, lo que facilita el seguimiento del progreso realizado día a día dentro del período definido. La duración de este sprint se estableció en 19 días, con un horario de trabajo de 8 horas por día, obteniendo un volumen total de 152 horas, que se coloca en el eje vertical (Y), representando el total inicial de horas de trabajo del Sprint. Con el tiempo, se espera una disminución gradual de este valor, avanzando hacia el objetivo de completar todas las tareas asignadas hasta llegar a cero. Esto indica la conclusión del segundo sprint, en conformidad con los plazos y objetivos inicialmente propuestos.

Figura 8

Burndown chart para el Sprint 2



***Resultados del sprint*:** El objetivo de este sprint fue crear un sistema bancario simulado con una interfaz vulnerable que permitiera la intrusión de atacantes potenciales. Esto infundado en la arquitectura MVC y tecnologías como MongoDB, MySQL, JavaScript y React, con el propósito de consumir el módulo funcional del honeypot en la interfaz de servicios del sistema bancario.

Figura 9

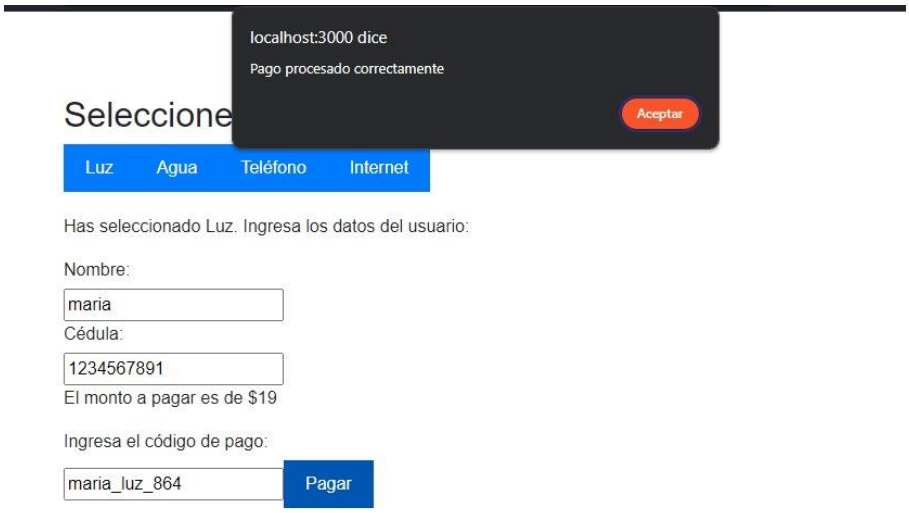
Interfaz de servicios del sistema bancario

La interfaz ofrece funcionalidades para procesar pagos de servicios y crear un entorno de transacciones en línea simulados como se visualiza en la Figura 9. Además, previamente se creó el modelo de detección de anomalías programado a fin de detectar patrones inusuales en las operaciones y enviar alertas tempranas de actividades sospechosas.

Durante la etapa de pruebas, se muestra que el sistema puede distinguir entre transacciones legitimas e intentos fraudulentos de intrusión, además notifica mediante alertas en tiempo real el proceso de pago realizado como se muestra en la Figura 10.

Figura 10

Mensaje del pago ingresado en el sistema bancario



Este sprint estableció bases sólidas en cuanto a la arquitectura e interfaz de usuario para desarrollar un entorno controlado de pruebas de penetración sobre el sistema de detección de intrusiones, aunque todavía hay lugar en relación a mejoras de términos de cobertura de casuística e inteligencia del modelo predictivo.

Implementación del código

Se implementó el back - end del sistema bancario utilizando Flask, donde se consumen las funciones de request, que proporciona acceso a los datos enviados por el cliente al servidor y jsonify que ayuda a crear respuestas JSON de las solicitudes del cliente. En esta implementación se define las rutas para realizar las solicitudes HTTP como POST, GET, DELETE y PUT. Se creó funciones asociadas a estas rutas procesan los datos enviados por el cliente. Este código se muestra en la Figura 11.

Figura 11

Implementación del back – end

```
from flask import Flask,request,jsonify
from flask_pymongo import PyMongo,ObjectId
from flask_bcrypt import Bcrypt
from flask_cors import CORS
from datetime import datetime
import pytz

app = Flask(_name_)
app.config['MONGO_URI']='mongodb://localhost/pythonreactdb'
mongo=PyMongo(app)
db=mongo.db.users
auth_db = mongo.db.auth
tarjeta_db=mongo.db.tarjeta
compras_db = mongo.db.compras
compras_anomalas_db= mongo.db.compras_Anomalas
servicios_db= mongo.db.servicios


#para evitar el cors de node
bcrypt = Bcrypt(app)
#para evitar el cors de node
CORS(app)
# ruta
@app.route('/users', methods=['POST'])
def createUser():
    try:
        # Utiliza insert_one para insertar un solo documento
        result = db.insert_one({
            'name': request.json['name'],
            'email': request.json['email'],
            'password': request.json['password'],
        })
        # Obtén el ID del documento insertado
        return jsonify(str(result.inserted_id))
        #return 'received'
    except KeyError as e:
        return jsonify({'error': f'Missing key: {str(e)}'}), 400
# Función para listar usuarios
@app.route('/users', methods=['GET'])
def getUsers():
```

```
    users = []
    for doc in db.find():
        user_data = {
            '_id': str(ObjectId(doc['_id'])),
            'name': doc.get('name', ''),
            'email': doc.get('email', ''),
            'password': doc.get('password', '')
        }
        users.append(user_data)
    return jsonify(users)
# Función para obtener un usuario por ID
@app.route('/user/<id>', methods=['GET'])
def getUser(id):
    user = db.find_one({'_id': ObjectId(id)})
    if user:
        return jsonify({
            '_id': str(ObjectId(user['_id'])),
            'name': user['name'],
            'email': user['email'],
            'password': user['password']
        })
    else:
        return jsonify({'error': 'User not found'}), 404
# Ruta para eliminar un usuario por ID
@app.route('/users/<id>', methods=['DELETE'])
def deleteUser(id):
    result = db.delete_one({'_id': ObjectId(id)})
    if result.deleted_count > 0:
        return jsonify({'message': 'User deleted successfully'})
    else:
        return jsonify({'error': 'User not found'}), 404
# Ruta para actualizar un usuario por ID
@app.route('/users/<id>', methods=['PUT'])
def updateUser(id):
    # Verifica si el usuario existe
    existing_user = db.find_one({'_id': ObjectId(id)})
    if existing_user:
        # Actualiza los campos con los nuevos valores proporcionados
        existing_user['name'] = request.json.get('name',
existing_user['name'])
        existing_user['email'] = request.json.get('email',
existing_user['email'])
        existing_user['password'] = request.json.get('password',
existing_user['password'])

        # Guarda la actualización en la base de datos
        db.update_one({'_id': ObjectId(id)}, {'$set': existing_user})

        return jsonify({'message': 'User updated successfully'})
```

```
    else:
        return jsonify({'error': 'User not found'}), 404

#NUEVAS RUTAS
# Ruta para el registro de usuario
@app.route('/register', methods=['POST'])
def register():
    try:
        print(f"Request JSON: {request.json}")  # Imprimir el contenido
de la solicitud JSON
        # Verifica si el usuario ya existe por correo electrónico
        existing_user = db.find_one({'email': request.json['email']})
        if existing_user:
            return jsonify({'error': 'User already exists'}), 400

        # Crea un nuevo usuario en la colección "users"
        user_result = db.insert_one({
            'name': request.json['name'],
            'email': request.json['email'],
            # Otros campos de información general
        })

        # Hashea la contraseña y crea un registro en la colección
"auth"
        hashed_password =
bcrypt.generate_password_hash(request.json['password']).decode('utf-8')
        auth_result = auth_db.insert_one({
            'user_id': user_result.inserted_id,
            'password': hashed_password,
            # Otros campos relacionados con la autenticación
        })

        return jsonify({'message': 'User registered successfully',
'user_id': str(user_result.inserted_id)})
    except KeyError as e:
        print(f'Missing key: {str(e)}')
        return jsonify({'error': f'Missing key: {str(e)}'}), 400
    except Exception as e:
        print(f'Error during registration: {str(e)}')
        return jsonify({'error': 'Error during registration'}), 400

# Ruta para el inicio de sesiónsss
@app.route('/login', methods=['POST'])
def login():
    try:
        # Busca el usuario por su correo electrónico en la colección
"users"
        user = db.find_one({'email': request.json['email']})
        if user:
```

```
            # Busca la información de autenticación en la colección
"auth"
            auth_info = auth_db.find_one({'user_id': user['_id']})
            if auth_info and
bcrypt.check_password_hash(auth_info['password'],
request.json['password']):
                return jsonify({'message': 'Login successful',
'user_id': str(user['_id'])})
            else:
                return jsonify({'error': 'Invalid credentials'}), 401
        else:
            return jsonify({'error': 'User not found'}), 404
    except KeyError as e:
        return jsonify({'error': f'Missing key: {str(e)}'}), 400

#la parte de las tarjetas
# Ruta para agregar una nueva tarjeta
@app.route('/banco/agregar-tarjeta', methods=['POST'])
def agregar_tarjeta():
    try:
        # Inserta una nueva tarjeta en la colección 'tarjeta_db'
        tarjeta_result = tarjeta_db.insert_one({
            'nombre_propietario': request.json['nombre_propietario'],
            'numero_tarjeta': request.json['numero_tarjeta'],
            'fecha_expiracion': request.json['fecha_expiracion'],
            'cvv': request.json['cvv'],
            'saldo': request.json['saldo']
        })

        return jsonify({'message': 'Tarjeta agregada correctamente',
'tarjeta_id': str(tarjeta_result.inserted_id)})
    except KeyError as e:
        return jsonify({'error': f'Missing key: {str(e)}'}), 400

# Ruta para consultar una tarjeta por número de tarjeta
@app.route('/banco/consultar-tarjeta/<numero_tarjeta>',
methods=['GET'])
def consultar_tarjeta(numero_tarjeta):
    tarjeta = tarjeta_db.find_one({'numero_tarjeta': numero_tarjeta})
    if tarjeta:
        return jsonify({
            'nombre_propietario': tarjeta['nombre_propietario'],
            'numero_tarjeta': tarjeta['numero_tarjeta'],
            'fecha_expiracion': tarjeta['fecha_expiracion'],
            'cvv': tarjeta['cvv'],
            'saldo': tarjeta['saldo']
        })
    else:
        return jsonify({'error': 'Tarjeta no encontrada'}), 404
```

```
@app.route('/compras', methods=['POST'])
def createCompra():
    try:
        data = request.json
        zona_horaria_local = pytz.timezone('America/Mexico_City')  #
Reemplaza con tu zona horaria local
        fecha_hora_pago = datetime.now(zona_horaria_local)
        direccion_ip = request.remote_addr
        compras_anomalas_db = mongo.db.compras_Anomalas
        result = compras_anomalas_db.insert_one({
            'tarjeta': data.get('tarjeta', ''),
            'productos': data.get('productos', []),
            'total': data.get('total', 0),
            'fecha_hora_pago': fecha_hora_pago.strftime("%Y-%m-%d
%H:%M:%S %p"),
            'direccion_ip': direccion_ip,
        })
        return jsonify({'message': 'Compra realizada con éxito',
'compra_id': str(result.inserted_id)})
    except Exception as e:
        return jsonify({'error': str(e)}), 500

@app.route('/servicios/pagar', methods=['POST'])
def procesar_pago_servicio():
    try:
        # Obtener los datos del pago desde la solicitud
        data = request.json

        # Guardar los detalles del pago en la colección "servicios"
        servicios_db = mongo.db.servicios
        result = servicios_db.insert_one({
            'servicio': data['servicio'],
            'monto': data['monto'],
            'codigo_pago': data['codigoPago'],  # Guardar el código de
pago
            'nombre': data['nombre'],
            'cedula': data['cedula'],
            'fecha_pago': data['fechaPago'],
            'ip_pago': data['ipPago']
        })

        # Responder con un mensaje de éxito
        return jsonify({'message': 'Pago procesado correctamente',
'pago_id': str(result.inserted_id)})
    except Exception as e:
        # Manejar errores
        return jsonify({'error': str(e)}), 500
```

```
if _name_ == "_main_":
    app.run(debug=True)
```

Se implemento una API con flask maneja solicitudes POST en la ruta ‘servicios/pagar’ para procesar pagos de servicios, obtiene los datos del pago desde la solicitud JSON. Responde con un mensaje JSON indicando que el pago se procesó correctamente, junto con el ID del pago insertado. Si ocurre alguna excepción durante el proceso, maneja el error y devuelve un mensaje de error. El código se muestra en la Figura 12.

Figura 12

API del backend

```
@app.route('/servicios/pagar', methods=['POST'])
def procesar_pago_servicio():
    try:
        # Obtener los datos del pago desde la solicitud
        data = request.json

        # Guardar los detalles del pago en la colección "servicios"
        servicios_db = mongo.db.servicios
        result = servicios_db.insert_one({
            'servicio': data['servicio'],
            'monto': data['monto'],
            'codigo_pago': data['codigoPago'],  # Guardar el código de
pago
            'nombre': data['nombre'],
            'cedula': data['cedula'],
            'fecha_pago': data['fechaPago'],
            'ip_pago': data['ipPago']
        })

        # Responder con un mensaje de éxito
        return jsonify({'message': 'Pago procesado correctamente',
'pago_id': str(result.inserted_id)})
    except Exception as e:
        # Manejar errores
        return jsonify({'error': str(e)}), 500
```

En cuanto a la implementación del Front – end se importaron los módulos de React y Axios, al igual que una biblioteca de solicitudes HTTP en JavaScript. Con esto se crea la función de axios que renderiza la interfaz de usuario cuando se selecciona un servicio y realizar un pago. Se definieron funciones auxiliares para realizar la solicitud de POST al servidor que procesa el pago, una vez que este es realizado se muestra un mensaje de éxito o de error. Este código se muestra en la Figura 13.

Figura 13

Implementación del Front – end

```
import React, { useState } from 'react';
import axios from 'axios';
import './css/servicios.css';

function Servicios() {

    const [montoPago, setMontoPago] = useState('');

    const [servicioSeleccionado, setServicioSeleccionado] =
useState(null);

    const [datosUsuario, setDatosUsuario] = useState({
        nombre: '',
        cedula: ''
    });

    const [valorAleatorio, setValorAleatorio] = useState(null);

    // Función para generar un valor aleatorio dentro de un rango
específico
    const generarValorAleatorio = (min, max) => {
        return Math.floor(Math.random() * (max - min + 1)) + min;
    };

    const establecerValorAleatorio = (servicio) => {
        switch (servicio) {
            case 'Luz':
                return generarValorAleatorio(15, 30);
            case 'Agua':
                return generarValorAleatorio(5, 25);
            case 'Teléfono':
                return 12;
            case 'Internet':
```

```
                return 44;
            default:
                return null;
        }
    };

  // Función para procesar el pago
const procesarPago = async () => {
    try {
        const response = await
axios.post('http://localhost:5000/servicios/pagar', {
            servicio: servicioSeleccionado,
            monto: valorAleatorio,
            codigoPago: montoPago,
            nombre: datosUsuario.nombre,
            cedula: datosUsuario.cedula,
            fechaPago: new Date().toISOString(),
            ipPago: await obtenerDireccionIP()
        });

        alert(response.data.message);
        setServicioSeleccionado(null);
        setMontoPago('');
        setDatosUsuario({
            nombre: '',
            cedula: ''
        });
        // Generar un nuevo valor aleatorio para el próximo pago
        setValorAleatorio(establecerValorAleatorio(servicioSeleccionado
));
    } catch (error) {

        console.error('Error al procesar el pago:', error);
        alert('Error al procesar el pago. Por favor, inténtalo de
nuevo.');
    }
};


    // Función para obtener la dirección IP del cliente
    const obtenerDireccionIP = async () => {
        const response = await
axios.get('https://api.ipify.org?format=json');
        return response.data.ip;
    };

export default Servicios;
```

Resumen del desarrollo del sistema de detección de intrusiones

Sprint 01: Se generó un dataset considerando las columnas de ip de ubicación, nombre, fecha, cantidad, identificador del servicio. Al igual que se creó el modelo de Machine Learning para la detección de intrusos, en este caso con el fin de detectar transacciones anómalas. El modelo se desarrolló utilizando las tecnologías de Flask y Python. Se realizo pruebas del modelo y se corrigió los errores.

Sprint 02: El sistema bancario implementado con el honeypot fue creado empleando tecnologías de React, JavaScript, Flask y Python. Se implemento una alarma en la que nos muestra cuando se realiza un pago anómalo, es decir cuando detecta el ataque de spoofing, se realizó pruebas en el código y se corrigió los errores en el sistema.

Capítulo IV

Validación del honeypot y del sistema

En este capítulo se realizan pruebas a la funcionalidad del honeypot implementado con el sistema bancario, esto se llevó a cabo un riguroso proceso de pruebas. Estas pruebas fueron cruciales para identificar con precisión el procesamiento de pagos en entornos de transacción en línea. La metodología elegida en esta fase fue la validación cruzada, conocida por su capacidad en relación a evaluar la generalización de un modelo a un conjunto de datos independiente.

Se implementó una validación cruzada para proporcionar una estimación fiable del desempeño del modelo en la práctica. Este método implica dividir el conjunto total de datos en particiones k o "folds" de tamaño similar. Luego, iterativamente, se utiliza un fold como conjunto de pruebas y los restantes folds, k-1 se usan como equipo de entrenamiento. Este proceso se repite k veces, con cada plegado usado exactamente una vez como el conjunto de prueba. La ventaja de este enfoque reside en su capacidad de utilizar todos los datos disponibles tanto en la capacitación como en las pruebas, minimizando así los prejuicios y maximizando la precisión en la evaluación del modelo (Sudharshan et al., 2023).

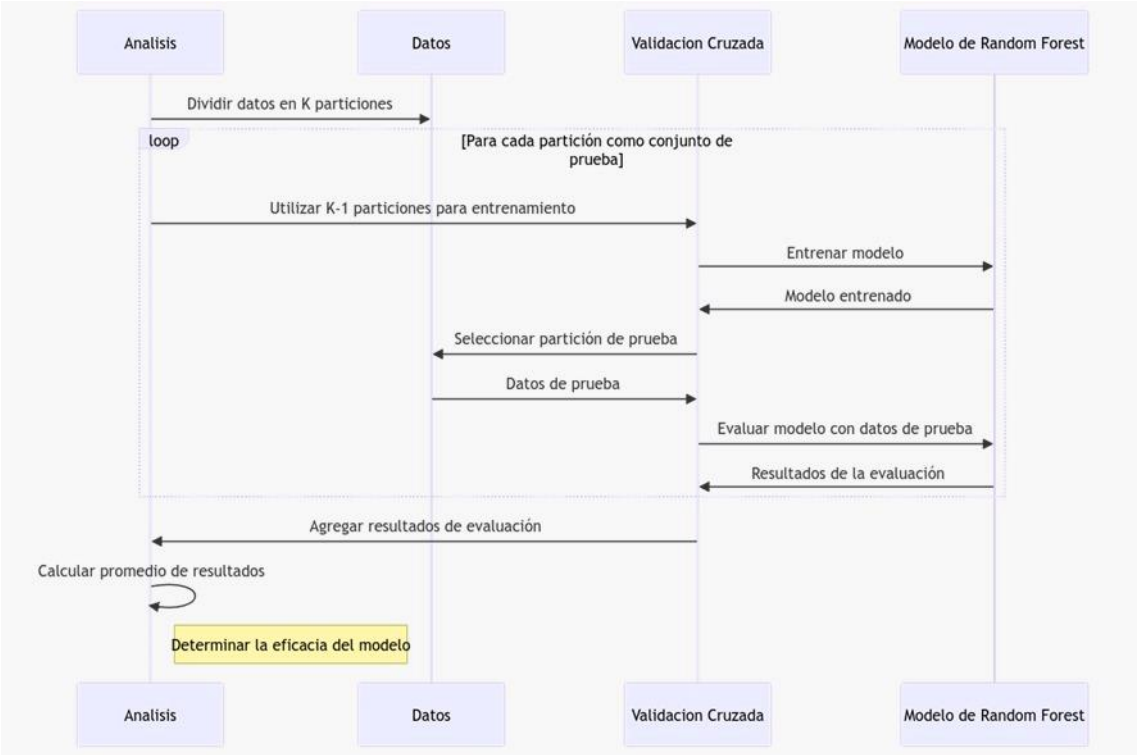
El primer escenario de prueba se centró en evaluar el rendimiento del sistema utilizando un conjunto de datos generados con 25.000 características distintas. Esta información provino del algoritmo que fue hecho con el objetivo de simular el historial de transacciones de los usuarios en un banco ficticio, incluyendo tanto transacciones legítimas y anómalas que se muestra en la Figura 5. La selección de 25.000 características se basó en la hipótesis de que aportarían granularidad suficiente para permitir al modelo identificar patrones y anomalías significativos relacionados con comportamientos fraudulentos.

Con intención de asegurar el aislamiento de cada prueba y evitar interferencias que pudieran afectar a la fiabilidad de los resultados, el sistema se estableció en diferentes puertos

en cada escenario de prueba. Esta estrategia permitió una evaluación exacta de la capacidad del honeypot para detectar y registrar intentos de intrusión bajo diversas condiciones, asegurando que los eventos de una prueba no contaminaron los datos de otra durante el proceso. La Figura 14 muestra visualmente el proceso de ejecución de pruebas.

Figura 14

Proceso de validación cruzada



Definición y Aplicación de Métricas de Pruebas

Con el fin de garantizar la eficacia del sistema Secure Pay Guard, se definieron métricas de evaluación precisas, esenciales para medir el rendimiento y la fiabilidad del honeypot en entornos de transacciones en línea. Estas métricas seleccionadas se muestran en la Tabla 14, incluyendo acurracy, precision, y recall, cada una con fórmulas específicas de cálculo basado en los valores de los positivos verdaderos (TP), negativos veraces (TN), positivos falsos (FP) y negativos falsos (FN). Estas métricas se derivan directamente de la matriz de confusión que se

muestra en la Tabla 15, lo que permite una evaluación cuantitativa del rendimiento del sistema.

Aplicación de las Pruebas

Para la aplicación de pruebas se considera que el dataset utilizado en este escenario fue creado a partir de un algoritmo que simula un historial de transacciones de usuarios de una entidad bancaria ficticia. Este algoritmo fue creado con el objetivo de incorporar una variedad de patrones de transacciones, tanto legítimos como anómalas, con el fin de establecer un entorno de prueba realista en el sistema de detección de intrusiones. Las 25.000 características cubrieron diversos aspectos de las transacciones, incluyendo cantidades, frecuencias, y tiempos pertinentes en relación a detectar comportamientos anormales.

La validación cruzada se utiliza como técnica de evaluación de modelos destinada a mejorar la estimación del rendimiento del modelo sobre datos granulares. Este proceso divide el conjunto de datos en k partes igualmente distribuidas o "folds". Durante cada iteración, un fold diferente se pone a un lado como el conjunto de datos de prueba, mientras que los folds k-1 restantes se utilizan como la formación de datos. Este proceso se repite k veces, con cada plegado sirviendo exactamente una vez en el conjunto de prueba. En última instancia, el rendimiento del modelo se media sobre los k resultados obtenidos, proporcionando una estimación robusta de su eficacia.

En el proceso de pruebas se compara los modelos de regresión logística y Random Forest bajo condiciones de ensayo idénticas, proporcionando una evaluación imparcial de ambos enfoques. El modelo de regresión logística, conocido por su simplicidad y transparencia, a pesar de su estructura lineal, es capaz de proporcionar una base sólida teniendo en cuenta una clasificación eficiente de las transacciones. Al emplear este enfoque con datos específicos, puede evaluar la eficacia de regresión logística en la detección entre transacciones genuinas y anómalas a través de métricas como la acurracy, la precisión y el recall. Sin embargo, la

naturaleza lineal del modelo puede ser una restricción en casos donde las interacciones entre las características de los datos son complicadas o no lineales, lo cual puede afectar la capacidad del modelo con el fin de detectar todas las transacciones fraudulentas, posiblemente influyendo en el recall.

Por otro lado, el modelo Random Forest, que es un conjunto de árboles de decisiones múltiples, muestra una capacidad superior para adaptarse a la complejidad y la variedad inherentes a los datos de transacciones bancarias. La validación cruzada muestra que Random Forest no sólo gestiona de manera eficiente la diversidad de datos, sino que también demuestra una mejora significativa en la identificación de transacciones anómalas, como indican las métricas de recall y precision en la Tabla 25. La utilización de los árboles de decisión del conjunto reduce el riesgo de sobreajuste y mejora las capacidades de generalización del modelo. Esta mejora de rendimiento se produce a expensas de una mayor complejidad computacional y de una disminución de la interpretabilidad de los modelos, aspectos que deben considerarse cuidadosamente.

Identificación de Errores

La falta de columnas (características) en el conjunto de datos empleado para entrenar los modelos de Machine Learning, en particular el modelo de regresión logística y el modelo de Random Forest, fue uno de los primeros y más significativos problemas identificados. La limitación se debió principalmente a la naturaleza simulada del entorno de prueba, esta no ofrecía una representación completa y variada de los patrones de comportamiento del usuario en escenarios de transacciones en el mundo real.

Se busca que el modelo pueda identificar entre comportamientos legítimos y anómalos sin éxito debido a la falta de características discriminativas en el conjunto de datos. Esto es específicamente relevante en el contexto de sistemas de detección de intrusiones, donde la

riqueza y calidad del conjunto de datos de entrenamiento tienen una gran influencia en la precisión de la identificación de actividades sospechosas.

La dificultad encontrada en predecir con exactitud el comportamiento de los usuarios basándose en los datos disponibles fue otro desafío significativo. El comportamiento del usuario en entornos de transacciones en línea puede ser muy diverso y dinámico, influenciado por una multitud de factores psicológicos, contextuales y tecnológicos. Esta diversidad y complejidad hacen que sea especialmente difícil para los modelos de Machine Learning, capacitados en un entorno simulado con un conjunto limitado de características, capturar y predecir con precisión todas las posibles variaciones de comportamiento legítimo y malicioso.

Durante el proceso de validación, se observaron limitaciones significativas en el escenario inicial de 25.000 características, en particular en lo que respecta a la capacidad del sistema para manejar transacciones de gran complejidad y volumen. Esta situación puso de relieve la necesidad de explorar escenarios con un mayor número de características, lo que llevó a la aplicación del segundo escenario de prueba con 45.000 características. El objetivo de este ajuste no sólo era corregir errores previamente identificados, sino también evaluar la robustez del sistema en condiciones más exigentes, simulando un entorno real de transacciones en línea con una carga significativamente mayor.

Corrección de Errores y Ajuste de Modelos

Este proceso implica mejorar los modelos de Machine Learning para aumentar su precisión y eficacia en la identificación de comportamientos maliciosos. Sé explica el método empleado con el propósito de lidiar con los retos previamente identificados, enfatizándose en particular en la segmentación de datos según la cantidad de recursos de los usuarios y en la optimización de la base de datos examinada por el honeypot.

Con la intención de mejorar la precisión del modelo en la detección de anomalías y comportamientos maliciosos, se llevó a cabo un análisis detallado que incluyó la segmentación

de los grupos de simulación según la escala de recursos de los usuarios. Esta estrategia se basó en la premisa de que los patrones de gasto y las transacciones varían significativamente entre los usuarios de diferentes niveles socioeconómicos. La tabla 29 proporcionada, que categoriza los pagos mensuales por servicios básicos en tres segmentos (bajos, medios y altos recursos), sirvió de marco para esta segmentación.

Tabla 29

Segmentación de grupos simulados

Pagos Mensuales Aproximados	Bajos Recursos (USD)	Medios Recursos (USD)	Altos Recursos (USD)
Luz	$20 - $40	$30 - $50	$50 - $80
Agua	$10 - $20	$15 - $30	$30 - $50
Teléfono	$10 - $30	$20 - $40	$30 - $60
Internet	$20 - $50	$30 - $70	$50 - $100

Nota. Pagos Mensuales Aproximados analizados de informes oficiales del INEC para el año 2023

La segmentación permitió al modelo explicar la variabilidad en los patrones de transacción que pueden ser normales para un segmento, pero anormales en otro. Esta diferenciación ayudó a reducir los falsos positivos y a perfeccionar la detección de actividades sospechosas basándose en el contexto socioeconómico del usuario.

La base de datos analizada por el honeypot incluye columnas críticas como IP, hora, ubicación y código de pago, que son esenciales para la detección de anomalías. Sin embargo, el análisis mostró que el modelo se beneficiaría significativamente de una revisión y optimización de estas columnas con el propósito de incluir datos más detallados y contextuales. Se hicieron los siguientes ajustes en el análisis de la base de datos y el modelo:

Se añadieron metadatos adicionales a cada transacción para proporcionar un contexto más rico al modelo. Esto incluyó detalles específicos como la frecuencia de las transacciones

por usuario. Estos datos permiten al modelo identificar patrones de comportamiento más complejos y anomalías sutiles.

Además, se identifica limitaciones en el primer escenario con 25.000 características, y se realizan correcciones de errores y ajustes de modelo, centrándose en el desarrollo y evaluación de un segundo escenario de 40.000 características. Esta decisión fue impulsada por la necesidad de abordar las deficiencias analizadas, especialmente en la capacidad del sistema de procesar y analizar con precisión un mayor volumen y complejidad de las transacciones. Se considera un aumento en el número de características destinadas a evaluar la capacidad del sistema para operar bajo condiciones de alta carga y estrés, simulando un entorno operativo en línea realista.

El modelo se ha mejorado con el fin de evaluar las transacciones no sólo sobre la base de las columnas existentes, sino también teniendo en cuenta el contexto temporal (por ejemplo, tiempos de transacción inusuales) y el contexto geográfico (transacciones realizadas desde ubicaciones atípicas del usuario). Este enfoque integral para la detección de anomalías mejora la precisión de la identificación de posibles amenazas a la seguridad y actividades anómalas.

Aplicación de Métricas de Evaluación del Modelo Ajustado

Las métricas de evaluación elegidas para este análisis incluyen la accuracy, el recall, la precisión. Estas métricas permiten una evaluación completa del desempeño del modelo en términos de su capacidad como propósito de clasificar correctamente las transacciones como legítimas o anómalas. A continuación, se describe la aplicación de cada métrica.

Accuracy: Mide la proporción de predicciones correctas (tanto positivas como negativas) en relación con el número total de casos. Es útil para obtener una visión general del rendimiento del modelo, especialmente en conjuntos de datos equilibrados.

Recall: Calcula la proporción de verdaderos positivos correctamente identificados por el modelo. Esta métrica es crucial en cuanto a el sistema de detección de intrusiones, donde es preferible capturar tantos ataques como sea posible, incluso a expensas del aumento de falsos positivos.

Precision: Estima la proporción de predicciones positivas que fueron correctas. La alta precisión indica un menor número de falsos positivos, lo que es crucial para evitar la interrupción de las transacciones legítimas.

La aplicación de estas métricas se llevó a cabo mediante una serie de pruebas de validación cruzada, utilizando un conjunto de datos enriquecido y cuidadosamente segmentado con el fin simular diversos escenarios y tipos de comportamientos de los usuarios. Cada transacción fue evaluada por el refinado modelo Random Forest, y los resultados fueron meticulosamente registrados y analizados.

La elección del modelo Random Forest fue ampliamente justificada por su superior rendimiento en las pruebas, como se demuestra en la Tabla 26. Este modelo demostró una capacidad excepcional para distinguir entre conductas legítimas y maliciosas, incluso en los casos más sutiles de anomalías, donde la distinción entre transacciones legales y mínimamente anómalas se vuelve particularmente difícil.

El análisis de la Figura 6 mostró que, a pesar del reto planteado por el aumento de los falsos positivos en relación a mantener un alto nivel de recuperación, Random Forest logró un equilibrio eficaz entre capturar una amplia gama de ataques y minimizar las interrupciones a las transacciones legítimas. Este equilibrio entre precision y recall, junto con la alta accuracy, destaca la idoneidad de Random Forest para aplicaciones críticas en el campo de la ciberseguridad, donde la detección precisa es primordial.

La capacidad de Random Forest en relación manejar una amplia gama de características y su robustez contra las variaciones en las escalas de características de los

usuarios lo posicionó como el modelo óptimo para este sistema de detección de intrusiones. Se prestó especial atención a las áreas en las que el modelo tenía debilidades, como la detección de anomalías sutiles y la distinción entre comportamientos legítimos y maliciosos en diferentes escalas de recursos de los usuarios.

Análisis de Resultados

Durante la fase de validación del sistema Secure Pay Guard, las métricas de evaluación mostraron resultados prometedores . Resultados obtenidos después de un análisis comparativo profundo a fin de determinar la eficiencia y adaptabilidad de dos modelos analíticos fundamentales: Random Forest y Regresión Logística. Esta evaluación se diseñó con dos escenarios de prueba distintos, configurados para reflejar una variedad de entornos operacionales de creciente complejidad, representados por conjuntos de datos de 25.000 y 40.000 características, respectivamente. Los resultados de estos escenarios se registraron meticulosamente en las tablas 25 y 26, ofreciendo una base cuantitativa con la intención de analizar el desempeño de cada modelo basado en métricas críticas como la accuracy, el recall y la precisión.

El ajuste del modelo no fue simplemente una recalibración basada en los resultados de las pruebas, sino una revisión exhaustiva que se consideró de la necesidad de un sistema capaz de aprender y adaptarse con el tiempo. Es por esto que la integración de más contexto en la evaluación de transacciones fue crucial en este proceso.

Con el fin evaluar la robustez del sistema en diversas condiciones, se realizaron pruebas en dos escenarios simulados. En cuanto el primer escenario, caracterizado por 25.000 atributos, los resultados visualizados en la Tabla 25 mostraron que el modelo de Random Forest tuvo un rendimiento superior, logrando una precisión de 98.679%, un recall de 100%, y una acurracy de 99.679%. Estos indicadores no sólo muestran una notable capacidad para clasificar las transacciones legítimas y fraudulentas con una precisión casi perfecta, sino

también una reducción insignificante de los falsos positivos. Por otra parte, el modelo de Regresión Logística, a pesar de alcanzar un recall del 100%, mostró una disminución en la Precisión al 84,76%, lo que indica una mayor tendencia a identificar incorrectamente las transacciones legítimas como fraudulentas.

La extensión del análisis al segundo escenario, con 40.000 características, tuvo como objetivo evaluar la escalabilidad de los modelos frente a un mayor volumen de datos. Como se muestra en la Tabla 26, Random Forest mantuvo su rendimiento excepcional, mejorando todas las métricas evaluadas a una puntuación perfecta, demostrando su robustez y eficacia en el manejo de la creciente complejidad de los datos. Por otra parte, la regresión logística, a pesar de mejorar en la precisión al 95,94%, todavía se enfrentaba a limitaciones como lo demuestra su precisión del 86,14%, lo que indica desafíos para manejar eficazmente el aumento de la complejidad. Esta variación se puede atribuir a la complejidad y el dinamismo de los patrones de transacciones, que plantean desafíos únicos en comparación con los entornos simulados.

Estos ajustes no sólo demuestran la adaptabilidad y evolución del sistema en respuesta a los desafíos planteados por el entorno simulado, sino que también hacen hincapié en la importancia de un enfoque iterativo en el desarrollo de sistemas de detección de transacciones anómalas, asegurando su eficacia y relevancia en la protección contra actividades fraudulentas en las transacciones en línea.

Además, la matriz de confusión, detallada en la Figura 6, proporcionó información sobre el rendimiento de los modelos en el escenario más desafiante. Random Forest logró una distinción perfecta sin errores, mientras que la Regresión Logística tuvo varios falsos positivos, lo que indica dificultades en distinguir con precisión entre transacciones legítimas y fraudulentas en condiciones de prueba complejas.

Este análisis comparativo subraya la importancia crítica de la selección de modelos en sistemas de detección de intrusiones en entornos de transacciones en línea, mostrando que

Random Forest es particularmente adecuado para aplicaciones que requieren alta precisión, adaptabilidad y manejo eficaz de datos complejos. Las conclusiones, documentadas en las Tablas 25 y 26, junto con la matriz de confusión en la Figura 7, ponen de relieve la necesidad de un enfoque iterativo y contextual en los sistemas de detección de intrusiones. Este enfoque asegura no sólo la eficacia inicial del sistema, sino también su capacidad de adaptarse y responder eficazmente a los cambios en los patrones de transacciones.

Identificación de Áreas de Mejora

El análisis de los resultados también ha identificado áreas en las que el sistema podría refinarse aún más. Por ejemplo, si bien la segmentación basada en los recursos ha mejorado la precisión, existe el potencial de incluir más variables contextuales que podrían enriquecer aún más el modelo. Estos factores podrían incluir el tipo de dispositivo utilizado en la transacción o el historial de transacciones del usuario, proporcionando un contexto adicional que podría ser útil a identificar comportamientos anormales. Además, se observó que, en los escenarios de alta carga de transacciones, el sistema experimentó un ligero aumento en el tiempo de respuesta. Esto sugiere que la escalabilidad y la optimización del rendimiento bajo altas cargas son áreas que podrían beneficiarse de la investigación y el desarrollo adicionales.

Los hallazgos de este análisis tienen implicaciones significativas en pro de futuras investigaciones y desarrollos en el campo de los sistemas de detección de intrusiones. En primer lugar, enfatizan la importancia de un enfoque iterativo en el diseño y ajuste de modelos de Machine Learning, donde la evaluación y adaptación continuas son cruciales para mantener la relevancia y eficacia del sistema contra las tácticas de ataque en evolución. Los resultados indican que la integración de métodos sofisticados de análisis de datos con nuevas fuentes de datos contextuales puede ofrecer nuevas oportunidades con el fin de mejorar la precisión y la eficiencia de los sistemas de detección de intrusiones.

Capítulo V

Conclusiones y Recomendaciones

Conclusiones

- Se ha logrado una profunda comprensión de las definiciones, la terminología y el funcionamiento de los honeypots, cumpliendo el primer objetivo específico. Este estudio ha examinado varios tipos de honeypots, ofreciendo una comprensión detallada del proceso de desarrollo. La investigación sobre técnicas computacionales avanzadas ha permitido la implementación de un honeypots especializado, destacando la importancia de una base teórica sólida encaminando a el éxito práctico del sistema.
- La adopción de una arquitectura en capas y el patrón MVC se convierten en elementos indispensables relacionados a el diseño y la gestión de sistemas seguros y escalables, como lo requiere el sistema de detección de intrusos en entornos transaccionales.
- La puesta en práctica de un honeypot especializado en el procesamiento de pagos y la generación de transacciones falsas, se ha cumplido con éxito. Este avance ha permitido la implementación directa del sistema en las transacciones en línea, destacando su capacidad con el fin simular vulnerabilidades, facilitando así su detección y análisis.
- La eficacia de este enfoque especializado en el entorno de las transacciones en línea subraya la importancia de adaptar las herramientas de seguridad a las necesidades específicas del contexto en el que se aplicarán.
- Las pruebas realizadas en entornos simulados han confirmado la capacidad del sistema para detectar y registrar con precisión intentos de intrusión. Esta validación no sólo demuestra la funcionalidad del sistema en diversas condiciones, sino que también asegura su aplicabilidad y fiabilidad en entornos reales de transacciones en línea.

Recomendaciones

- Es importante buscar palabras claves que se relacionen con el tema de investigación, con el fin encontrar información más relevante y útil de cara a empezar tu investigación.
- Es esencial ampliar el conjunto de datos utilizado para entrenar modelos de Machine Learning, incluyendo una mayor variedad de características que reflejan de manera más precisa y diversa los patrones de comportamiento del usuario en escenarios de transacciones reales. Esta ampliación debería centrarse en incorporar datos que capten la diversidad y complejidad del comportamiento del usuario, teniendo en cuenta factores psicológicos, contextuales y tecnológicos que pueden influir en las transacciones en línea.
- Es crucial implementar un proceso de evaluación y ajuste continuos de los modelos de aprendizaje automático utilizados. Este proceso debería implicar la validación periódica de la eficacia del modelo con nuevos conjuntos de datos, la reevaluación de las características utilizadas y la exploración de nuevos modelos o técnicas de aprendizaje automático que puedan proporcionar mejores resultados en la detección de intrusiones.
- La adaptabilidad y la mejora continua son esenciales hacia mantener la eficacia del sistema frente a la evolución de las tácticas de intrusión y los patrones de comportamiento del usuario.

Bibliografía

A, D. (2020, febrero 4). Qué es React: Definición, características y funcionamiento. *Tutoriales Hostinger*. https://www.hostinger.es/tutoriales/que-es-react

Aguilar, J. M. (2019, octubre 15). *¿Qué es el patrón MVC en programación y por qué es útil?* campusMVP.es. https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx

Alba Vega, D. A., & Calle Jara, J. F. (2020). *Aplicación de técnicas de Machine Learning basado en información sísmica para profundizar la probabilidad de terremotos mediante el uso de regresión logística y redes neuronales.* http://repositorio.ug.edu.ec/handle/redug/48862

Alqudhaibi, A., Deshpande, S., Jagtap, S., & Salonitis, K. (2023). Towards a sustainable future: Developing a cybersecurity framework for manufacturing. *Technological Sustainability*, *2*(4), 372-387. https://doi.org/10.1108/TECHS-05-2023-0022

Alsaeed, N., Nadeem, F., & Albalwy, F. (2024). A scalable and lightweight group authentication framework for Internet of Medical Things using integrated blockchain and fog computing. *Future Generation Computer Systems*, *151*, 162-181. https://doi.org/10.1016/j.future.2023.09.032

Alvarez, F. (2021). Machine Learning en la detección de fraudes de comercio electrónico aplicado a los servicios bancarios. *Ciencia y Tecnología*, 81-95. https://doi.org/10.18682/cyt.vi0.4310

Argumanis Escalante, D. (2021). *Propuesta de marco de trabajo basado en la integración de Scrum y el diseño centrado en el usuario para el proceso de desarrollo de software*. https://tesis.pucp.edu.pe/repositorio//handle/20.500.12404/19645

Atico. (2020, septiembre 23). Honeypots o sistemas trampa. Definición y funciones. *Grupo Atico34*. https://protecciondatos-lopd.com/empresas/honeypots-sistemas-trampa/

Atlassian. (2023). *Historias de usuario | Ejemplos y plantilla*. Atlassian. https://www.atlassian.com/es/agile/project-management/user-stories

Attia, O., Khoufi, I., Laouiti, A., & Adjih, C. (2019). An IoT-Blockchain Architecture Based on Hyperledger Framework for Healthcare Monitoring Application. *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 1-5. https://doi.org/10.1109/NTMS.2019.8763849

Avilés Bajaña, Y. L. (2017). *Implementación de una herramienta honeypot para detección y respuesta a ataques* [masterThesis, Espol]. http://www.dspace.espol.edu.ec/handle/123456789/37409

Ayuda Ley. (2020). *Transferencias bancarias fraudulentas ¿Cómo funcionan estas estafas?* https://ayudaleyprotecciondatos.es/2020/11/19/transferencia-bancaria-fraudulenta/

Balfaqih, M., Balfagih, Z., Lytras, M. D., Alfawaz, K. M., Alshdadi, A. A., & Alsolami, E. (2023). A Blockchain-Enabled IoT Logistics System for Efficient Tracking and Management of High-Price Shipments: A Resilient, Scalable and Sustainable Approach to Smart Cities. *Sustainability*, *15*(18), Article 18. https://doi.org/10.3390/su151813971

Campoverde Armijos, J. I. (2018). *Honeypot como herramienta de prevencion de ciberataques*. http://bibliotecadigital.econ.uba.ar/download/tpos/1502-1212_CampoverdeArmijosJI.pdf

Campus Stellae. (2023, marzo 6). ¿Qué son los falsos positivos? *Instituto Europeo Campus Stellae*. https://campus-stellae.com/que-son-los-falsos-positivos/

Chitarroni Horacio. (2002). *La regresion logistica*. https://racimo.usal.edu.ar/83/1/Chitarroni17.pdf

Codeicus. (2023, marzo 31). Sobre Épicas e Historias de Usuarios. *Medium*. https://codeicussoftware.medium.com/sobre-%C3%A9picas-e-historias-de-usuarios-9f8ff42a3e3d

Coppola, M. (2023). *Qué es JavaScript, para qué sirve y cómo funciona*. https://blog.hubspot.es/website/que-es-javascript

Crowd. (2022). *What is a Honeypot in Cybersecurity? - CrowdStrike*. https://www.crowdstrike.com/cybersecurity-101/honeypots-in-cybersecurity-explained/

Datos. (2021). *¿Cómo sé si mi modelo de predicción es realmente bueno? | datos.gob.es*. https://datos.gob.es/es/blog/como-se-si-mi-modelo-de-prediccion-es-realmente-bueno

desarrollo Web. (2023). *Qué es MVC*. DesarrolloWeb.com. https://desarrolloweb.com/articulos/que-es-mvc.html

Desarrollo Web. (2023, marzo 1). *¿Qué es Flask Python? Un breve tutorial sobre este microframework*. IONOS Digital Guide. https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/flask/

Díaz, R. (2020, mayo 8). ▷ Métricas de Clasificación—Aprende a EVALUAR tu modelo. *The Machine Learners*. https://www.themachinelearners.com/metricas-de-clasificacion/

Digital, R. (2021, julio 1). ¿Qué es el procesamiento de datos? *reputación digital*. https://reputacion.digital/que-es-el-procesamiento-de-datos/

Drew. (2019). *Ventajas y desventajas de la metodología Scrum*. https://blog.wearedrew.co/productividad/-ventajas-y-desventajas-de-la-metodologia-scrum

Durán, M. (2023). *Qué es la arquitectura en capas, ventajas y ejemplos*. https://blog.hubspot.es/website/que-es-arquitectura-en-capas

Eduard, A., & Daniel, L. (2013). *Honeypot: Ventajas y Desventajas como Mecanismo para la Prevención de Intrusos Informáticos*.

El Brujo. (2021). Los mejores HoneyPots: Ejemplos, tipos, características y configuración. *Blog elhacker.NET*. https://blog.elhacker.net/2021/01/los-mejores-honeypots-ejemplos-y-tipos-trampas-rdp-ssh-cowrie-docker-rdpy.html

Elizabeth, S. (2023). *Los cuatro tipos de fraude en el ecommerce*. https://es.clear.sale/blog/los-cuatro-tipos-de-fraude-en-el-ecommerce

Eslabon. (2011). *Eslabón | Ventajas de trabajar con el método SCRUM*. https://www.eslabon.com.mx/es/articulo/32-ventajas-de-trabajar-con-el-metodo-scrum

Espí Luis, S. (2017). *Desarrollo de un honeypot para la monitorización y prevención de ataques* [Proyecto/Trabajo fin de carrera/grado, Universitat Politècnica de València]. https://riunet.upv.es/handle/10251/91797

Esteban, E., & Ignacio, C. (2013). *Honeynets como herramienta de prevención e investigación de ciberataques*.

Estrada Velasco, M. V., Núñez Villacis, J. A., Saltos Chávez, P. R., & Cunuhay Cuchipe, W. C. (2021). Revisión Sistemática de la Metodología Scrum para el Desarrollo de Software. *Dominio de las Ciencias*, *7*(Extra 4), 54.

Fernández, Y. (2024, enero 12). *Spoofing: Qué es este tipo de ataque y cómo protegerte de él*. Xataka. https://www.xataka.com/basics/spoofing-que-este-tipo-ataque-como-protegerte

Fortinet. (2024). *What Is a Honeypot? Meaning, Types, Benefits, and More | Fortinet*. https://www.fortinet.com/resources/cyberglossary/what-is-honeypot

Frikha, T., Ktari, J., Zalila, B., Ghorbel, O., & Amor, N. B. (2023). Integrating blockchain and deep learning for intelligent greenhouse control and traceability. *Alexandria Engineering Journal*, *79*, 259-273. https://doi.org/10.1016/j.aej.2023.08.027

Garg, S., Puliafito, A., Telek, M., & Trivedi, K. (1998). Analysis of preventive maintenance in transactions based software systems. *IEEE Transactions on Computers*, *47*(1), 96-107. https://doi.org/10.1109/12.656092

Gómez López, J. (2009). *Optimización de sistemas de detección de intrusos en red utilizando técnicas computacionales avanzadas* [Http://purl.org/dc/dcmitype/Text, Universidad de Almería]. https://dialnet.unirioja.es/servlet/tesis?codigo=22175

Gonzales Gomez, D. (2003). *Sistemas de Deteccion de Intrusiones*. https://dgonzalez.net/papers/ids/ids_v1.0.pdf

Gonzales Kevin. (2013, febrero 1). *Técnicas de Procesamiento y Representación de Datos by Kevin Gonzalez—Issuu*. https://issuu.com/kvin92/docs/analisis_de_sistema

Hans710. (2009). *Honeypots: Herramienta de Seguridad de la Informacion*. Honeypots : Herramienta de Seguridad de la Informacion. https://honeypots.wordpress.com/

Harrison, O. (2019, julio 14). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. Medium. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Hernandez Rafael. (2021, junio 28). *El patrón modelo-vista-controlador: Arquitectura y frameworks explicados*. freeCodeCamp.org. https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/

IBM. (2020). *¿Qué es OLTP? | IBM*. https://www.ibm.com/mx-es/topics/oltp

Integra. (2019). *Conoce los principales roles de Scrum y sus responsabilidades*. https://integrait.com.mx/blog/roles-de-scrum/

Internacional. (2021, agosto 17). ¿Cómo realizar una transferencia bancaria? *Banco Internacional*. https://www.bancointernacional.com.ec/como-realizar-una-transferencia-bancaria/

Jaramillo Ramos, C. C., & Ospina Beltrán, M. I. (2019). *Arquitectura de integración basada en tecnología Blockchain para sistemas transaccionales con bases de datos distribuidas. Caso de estudio: Facturación agencia de viaje*. http://repository.udistrital.edu.co/handle/11349/22898

Jesus. (2019, noviembre 16). ¿Qué es el Accuracy? *DataSmarts Español*. https://datasmarts.net/es/que-es-el-accuracy/

kaskpersky. (2023, abril 19). *¿Qué es un honeypot?* latam.kaspersky.com. https://latam.kaspersky.com/resource-center/threats/what-is-a-honeypot

Kazemian, H., & Shrestha, S. (2023). Comparisons of machine learning techniques for detecting fraudulent criminal identities. *Expert Systems with Applications*, *229*, 120591. https://doi.org/10.1016/j.eswa.2023.120591

KeepCoding, R. (2022, abril 1). *¿Qué es una Arquitectura de Software?* https://keepcoding.io/blog/arquitectura-de-software/

La Hora. (2023). *Falsas transferencias, una nueva modalidad de estafa*. https://www.lahora.com.ec/tungurahua/falsas-transferencias-nueva-modalidad-estafa/

Lee, J., Azamfar, M., & Singh, J. (2019). A blockchain enabled Cyber-Physical System architecture for Industry 4.0 manufacturing systems. *Manufacturing Letters*, *20*, 34-39. https://doi.org/10.1016/j.mfglet.2019.05.003

Li, Y., Yang, J., Zhang, Z., Wen, J., & Kumar, P. (2023). Healthcare Data Quality Assessment for Cybersecurity Intelligence. *IEEE Transactions on Industrial Informatics*, *19*(1), 841-848. https://doi.org/10.1109/TII.2022.3190405

Liu, Y., Wang, Y., & Zhang, J. (2012). New Machine Learning Algorithm: Random Forest. En B. Liu, M. Ma, & J. Chang (Eds.), *Information Computing and Applications* (pp. 246-252). Springer. https://doi.org/10.1007/978-3-642-34062-8_32

Londoño, P. (2023). *Qué es Python, para qué sirve y cómo se usa (+ recursos para aprender)*. https://blog.hubspot.es/website/que-es-python

López, J. G. (2009). *Optimización de sistemas de detección de intrusos en red utilizando técnicas computacionales avanzadas*. Universidad Almería.

Luisquintanilla. (2023, marzo 13). *Métricas de ML.NET - ML.NET*. https://learn.microsoft.com/es-es/dotnet/machine-learning/resources/metrics

Martin, G., Martin, P., Hankin, C., Darzi, A., & Kinross, J. (2017). Cybersecurity and healthcare: How safe are we? *BMJ*, *358*, j3179. https://doi.org/10.1136/bmj.j3179

Martinekuan. (2023, julio 11). *Procesamiento de transacciones en línea (OLTP)—Azure Architecture Center*. https://learn.microsoft.com/es-es/azure/architecture/data-guide/relational-data/online-transaction-processing

Martos, A. J. O., Valdivia, M. T. M., López, L. A. U., & Cumbreras, M. Á. G. (2017). *Detección automática de Spam utilizando Regresión Logística Bayesiana*.

Maseleno, A. (2021). Design of Optimal Machine Learning based Cybersecurity Intrusion Detection Systems. *Journal of Cybersecurity and Information Management*, *Volume 0*(Issue 1), 32-43. https://doi.org/10.54216/JCIM.000103

Medina Garzón, M. A., & Vásquez Rodríguez, Y. (2020). *Diseño de un modelo y sistema que permite determinar el aseguramiento de la información bajo el estándar ISO 27000 en empresas financieras*. http://repository.udistrital.edu.co/handle/11349/29845

Menon, S., Anand, D., Kavita, Verma, S., Kaur, M., Jhanjhi, N. Z., Ghoniem, R. M., & Ray, S. K. (2023). Blockchain and Machine Learning Inspired Secure Smart Home Communication Network. *Sensors*, *23*(13), Article 13. https://doi.org/10.3390/s23136132

Mira Alfaro, E. J. (2003). *Implantacion de un Sistema de Deteccion de Intrusos en la Universidad de Valencia* [Ingenieria Informatica, Universidad de Valencia]. https://www.rediris.es/cert/doc/pdf/ids-uv.pdf

Miro. (2020). *Sprint backlog: Qué es, cómo organizarlo, ejemplos y plantillas*. https://miro.com/. https://miro.com/es/agile/que-es-sprint-backlog/

Mokube, I., & Adams, M. (2007). Honeypots: Concepts, approaches, and challenges. *Proceedings of the 45th annual southeast regional conference*, 321-326. https://doi.org/10.1145/1233341.1233399

Morales Luiz. (2018). *BIGDATA: 5 métodos para detectar posibles fraudes | Grupo Novatech*. https://www.grupo-novatech.com/bigdata-5-metodos-para-detectar-posibles-fraudes/

Murugappan, M., Nair, R., & Krishnan, S. (2023). Global Market Perceptions of Cryptocurrency and the Use of Cryptocurrency by Consumers: A Pilot Study. *Journal of Theoretical and Applied Electronic Commerce Research*, *18*(4), 1955-1970. Scopus. https://doi.org/10.3390/jtaer18040098

Nagaesh Singh Chauhan. (2023, septiembre 25). *Métricas De Evaluación De Modelos En El Aprendizaje Automático*. DataSource.ai. https://www.datasource.ai/es/data-science-articles/view-source:https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico

Nomas ISO. (2018). ISO 27001 - Seguridad de la información: Norma ISO IEC 27001/27002. *Normas ISO*. https://www.normas-iso.com/iso-27001/

Oracle. (2023). *¿Qué es el spoofing?* Oracle España. https://www.oracle.com/es/database/security/que-es-el-spoofing.html

Ortegón Cortázar, G. (2015). Optimización de sistemas de gestión académica. Una propuesta de gestión, medición y procesamiento de datos en un entorno virtual de aprendizaje para la toma de decisiones en instituciones educativas. *Revista EAN*, *79*, 80-97.

Palacios Jeronimo. (2021). *Scrum: Roles y responsabilidades | Deloitte España*. https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html

Perez Carlos. (2005). *Aplicacion de redes neuronales para la deteccion de intrusos en redes y sistemas de informacion*. https://www.redalyc.org/pdf/849/84911698042.pdf

Pichincha. (2021). *Spoofing aprende cómo reconocerlo y proteger tu información sensible*. https://www.pichincha.com/blog/spoofing-que-es-como-protegerte

Platzi. (2021). *Qué es el Backlog, las Épicas y las Historias de Usuario*. /clases/1750-scrum/24284-las-epicas-y-el-backlog-del-producto/. https://platzi.com/clases/1750-scrum/24284-las-epicas-y-el-backlog-del-producto/

Popovsky, D. B. E. (2015). *ICCSM2015-3rd International Conference on Cloud Security and Management: ICCSM2015*. Academic Conferences and publishing limited.

Priya, V. S. D., & Chakkaravarthy, S. S. (2023). Containerized cloud-based honeypot deception for tracking attackers. *Scientific Reports*, *13*(1), 1437. https://doi.org/10.1038/s41598-023-28613-0

Proofpoint. (2024). *¿Qué es el robo de identidad? - Definición, ejemplos y tipos | Proofpoint ES*. https://www.proofpoint.com/es/threat-reference/identity-theft

Quintero, F. R. C., Rojas, L. A. C., Bautista, D. R., Avendaño, E. B., Vergel, O. J. M., & Noriega, C. M. P. (2019). SISTEMA DE DETECCIÓN DE INTRUSOS A TRAVÉS DE UNA RED HONEYNET PARA ENTORNOS DE RED CABLEADA SOBRE IPV6. *REVISTA COLOMBIANA DE TECNOLOGIAS DE AVANZADA (RCTA)*, *1*(33), Article 33. https://doi.org/10.24054/rcta.v1i33.92

Rasheed, R., Al-Shqeerat, K., Ghorab, A., Abuowaimer, F., & Abusamra, A. (2023). Blockchain Mobile Wallet with Secure Offline Transactions. *Computers, Materials and Continua*, *75*, 2905-2919. https://doi.org/10.32604/cmc.2023.036691

Rigatti, S. J. (2017). Random Forest. *Journal of Insurance Medicine*, *47*(1), 31-39. https://doi.org/10.17849/insm-47-01-31-39.1

Robledano Angel. (2019, septiembre 24). *Qué es MySQL: Características y ventajas*. OpenWebinars.net. https://openwebinars.net/blog/que-es-mysql/

Rodrigues, N. (2021). *Sistemas OLTP: Características, ventajas y desventajas*. https://blog.hubspot.es/sales/que-es-oltp

Sanni, M., Akinyemi, B., Olalere, D., Olajubu, E., & Aderounmu, G. (2023). A Predictive Cyber Threat Model for Mobile Money Services. *Annals of Emerging Technologies in Computing*, *7*, 40-60. https://doi.org/10.33166/AETiC.2023.01.004

Santos Marco. (2020, julio 8). *Falsos Positivos Vs. Falsos Negativos*. DataSource.ai. https://www.datasource.ai/es/data-science-articles/view-source:https://www.datasource.ai/es/data-science-articles/falsos-positivos-vs-falsos-negativos

Scrum Manager. (2022). *Historias de Usuario*.

Spitzner, L. (2003). The Honeynet Project: Trapping the hackers. *IEEE Security & Privacy*, *1*(2), 15-23. https://doi.org/10.1109/MSECP.2003.1193207

Suarez Restrepo, S. (2021). *Indicadores de compromiso basados en ataques a servicios web*. https://repositorio.itm.edu.co/handle/20.500.12622/5601

Sudharshan, R., Sinha, K., Pragya, Manohari Balachander, G., & Agastinose Ronickom, J. F. (2023). Comparing Pertubagens from Differential Gene Expression Data Analysis of ASD using Random Forest and Statistical Test. *Current Directions in Biomedical Engineering*, *9*(1), 682-685. Scopus. https://doi.org/10.1515/cdbme-2023-1171

Takhion. (2018). *How to Use the Cowrie SSH Honeypot to Catch Attackers on Your Network « Null Byte: WonderHowTo*. https://null-byte.wonderhowto.com/how-to/use-cowrie-ssh-honeypot-catch-attackers-your-network-0181600/

Tariq, E., Akour, I., Al-shanableh, N., Alquqa, E., Alzboun, N., Ibra-Heem, S., Al-Hawary, S., & Alshurideh, M. (2023). How cybersecurity influences fraud prevention: An empirical study on Jordanian commercial banks. *International Journal of Data and Network Science*, *8*. https://doi.org/10.5267/j.ijdns.2023.10.016

Teng, D. (2022). Industrial Internet of Things Anti-Intrusion Detection System by Neural Network in the Context of Internet of Things for Privacy Law Security Protection. *Wireless Communications and Mobile Computing*, *2022*, 1-17. https://doi.org/10.1155/2022/7182989

Tian, W., Ji, X., Liu, W., Liu, G., Zhai, J., Dai, Y., & Huang, S. (2020). Prospect Theoretic Study of Honeypot Defense Against Advanced Persistent Threats in Power Grid. *IEEE Access*, *8*, 64075-64085. https://doi.org/10.1109/ACCESS.2020.2984795

Tovar Sergio. (2018). *Glastopf: Honeypot de aplicaciones web – I | Revista .Seguridad*. https://revista.seguridad.unam.mx/numero25/glastopf-honeypot-de-aplicaciones-web-i

Verdejo Alvarez, G. (2004). *HoneypotsyHoneynets.pdf*. https://www.cs.upc.edu/~gabriel/files/DEA-es-4HoneypotsyHoneynets.pdf

Vetrivendan, L., & Kumar, G. (2023). CCNN: An Artificial Intelligent based Classifier to Credit Card Fraud Detection System with Optimized Cognitive Learning Model. *International Journal on Recent and Innovation Trends in Computing and Communication*, *11*(5s), Article 5s. https://doi.org/10.17762/ijritcc.v11i5s.6640

Walsh, D. (2023). *Qué es spoofing, ejemplos y cómo evitarlo*. https://blog.hubspot.es/website/que-es-spoofing

Wang, C., Chai, S., Zhu, H., & Jiang, C. (2023). CAeSaR: An Online Payment Anti-Fraud Integration System With Decision Explainability. *IEEE Transactions on Dependable and Secure Computing*, *20*(3), 2565-2577. https://doi.org/10.1109/TDSC.2022.3186733

web development. (2022). *MongoDB: Qué es, características y para qué sirve*. Inesdi. https://www.inesdi.com/blog/mongodb-que-es-caracteristicas-para-que-sirve/

webDocs. (2023, noviembre 13). *MVC - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web | MDN*. https://developer.mozilla.org/es/docs/Glossary/MVC

West Dave. (2022). *Planificación de sprints | Atlassian*. https://www.atlassian.com/es/agile/scrum/sprint-planning

Yamin, M. M., Katt, B., & Nowostawski, M. (2021). Serious games as a tool to model attack and defense scenarios for cyber-security exercises. *Computers & Security*, *110*, 102450. https://doi.org/10.1016/j.cose.2021.102450

Zheng, Z., Xie, S., Dai, H.-N., Chen, X., & Wang, H. (2018). *Blockchain challenges and opportunities: A survey*.

Anexos