UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE TELECOMUNICACIONES

Sistema de ciberseguridad para reducir ataques cibernéticos aplicando tecnología blockchain.

Trabajo de Titulación para optar al título de Ingeniero en Telecomunicaciones

Autor:
Andrés Eduardo Cajas Sánchez

Tutor:
Mgs. Luis Gonzalo Santillan Valdiviezo

Riobamba, Ecuador. 2026

DECLARATORIA DE AUTORÍA

Yo, **Andrés Eduardo Cajas Sánchez**, con cédula de ciudadanía **0202500740**, autor del trabajo de investigación titulado: **Sistema de ciberseguridad para reducir ataques cibernéticos aplicando tecnología blockchain**, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 20 de marzo de 2026

Andrés Eduardo Cajas Sánchez

C.I: 0202500740

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

Quien suscribe, **Luis Gonzalo Santillán Valdiviezo** catedrático adscrito a la Facultad de Ingeniería, por medio del presente documento certifico haber asesorado y revisado el desarrollo del trabajo de investigación titulado: **"Sistema de ciberseguridad para reducir ataques cibernéticos aplicando tecnología blockchain"**, bajo la autoría de **Andrés Eduardo Cajas Sánchez**; por lo que se autoriza ejecutar los trámites legales para su sustentación.

Es todo cuanto informar en honor a la verdad; en Riobamba, a los 20 días del mes de febrero de 2026

Ing. Luis Gonzalo Santillán Valdiviezo, Mgs
C.I: 0603225350

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación "**SISTEMA DE CIBERSEGURIDAD PARA REDUCIR ATAQUES CIBERNÉTICOS APLICANDO TECNOLOGÍA BLOCKCHAIN**", presentado por **ANDRÉS EDUARDO CAJAS SÁNCHEZ**, con cédula de identidad número **0202500740**, bajo la tutoría de Mgs. Luis Gonzalo Santillán Valdiviezo; certificamos que recomendamos la **APROBACIÓN** de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba el 20 de marzo de 2026.

Alejandra Del Pilar Pozo Jara, Msc.
PRESIDENTE DEL TRIBUNAL DE GRADO

Marlon Danilo Basantes Valverde, PhD.
MIEMBRO DEL TRIBUNAL DE GRADO

Deysi Vilma Inca Balseca, Mgs.
MIEMBRO DEL TRIBUNAL DE GRADO

CERTIFICACIÓN

Que, **Andrés Eduardo Cajas Sánchez** con CC: **0202500740**, estudiante de la Carrera **TELECOMUNICACIONES**, Facultad de **INGENIERÍA**; ha trabajado bajo mi tutoría el trabajo de investigación titulado **"Sistema de ciberseguridad para reducir ataques cibernéticos aplicando tecnología blockchain.",** cumple con el 5 %, de acuerdo con el reporte del sistema Anti plagio **COMPILATIO**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente, autorizo continuar con el proceso.

Riobamba, 18 de marzo de 2026

PhD. Carlos Ramiro Peñafiel Ojeda
Director de carrera

DEDICATORIA

Este trabajo de tesis dedico en primer lugar a Dios por brindarme vida, salud y fortaleza para poder culminar con esta etapa importante en mi formación profesional.

A mis padres, por su constante apoyo, sacrificio, por motivarme a seguir adelante y no desmayar en momentos difíciles.

A mi familia, por su compresión, optimismo durante este proceso académico, siendo una fuente de inspiración para alcanzar esta meta.

Y a todas las personas que, de una u otra manera, formaron parte de este camino y contribuyeron a la culminación de este logro tan significativo.

AGRADECIMIENTO

Agradezco, en primer lugar, a Dios, por darme la vida, la salud, la fortaleza y la sabiduría necesarias para culminar esta etapa importante de mi formación académica.

Expreso mi más sincero agradecimiento a mis padres y a mi familia, quienes han sido mi apoyo incondicional, brindándome ánimo, comprensión y motivación constante a lo largo de este proceso.

De manera especial, agradezco a mi tutor de tesis, por su guía, orientación, tiempo y conocimientos compartidos, los cuales fueron fundamentales para el desarrollo y culminación de esta investigación.

Asimismo, extiendo mi agradecimiento a los docentes y a la institución, por contribuir a mi formación profesional y por brindarme los conocimientos necesarios para alcanzar esta meta.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

En la actualidad los sistemas informáticos se encuentran expuestos a diversas amenazas cibernéticas, como el aumento de ataques DoS afectando la disponibilidad de plataformas computacionales, los ciber atacantes buscan cambiar o eliminar información de los registros de ataques para evitar ser rastreados. En el presente trabajo tiene como objetivo el diseño de un sistema de ciberseguridad utilizando blockchain.

La investigación fue llevada a cabo empleando un enfoque experimental, se utilizó un entorno de laboratorio controlado. Durante las pruebas realizadas se obtuvo un registro de las direcciones ip, tipo de ataque, hora y fecha previniendo que las máquinas virtuales victimas sufran alguna alteración en su sistema, al activar en modo de monitoreo al Snort se obtuvo información para métricas como disponibilidad, cpu, mem. Para el análisis estadístico se utilizó métodos no paramétricos.

Los resultados que se obtuvieron evidencian el almacenamiento y detección al momento de recibir la presencia de ataques DoS de forma oportuna y confiable. La integración de blockchain ayuda a tener un sistema más confiable en los eventos relacionados a la seguridad informática, almacenamiento de direcciones ip y mecanismos definidos por reglas de Snort.

Palabras claves: Blockchain, Snort, DoS, Docker.

ABSTRACT

Currently, computer systems are exposed to various cyber threats, such as the increase in DoS attacks affecting the availability of computing platforms; cyber attackers seek to change or delete information from attack logs to avoid being tracked. The present work aims to design a cybersecurity system using blockchain. The research was carried out experimentally in a controlled laboratory environment. During the tests, a record was kept of IP addresses, attack type, time, and date, preventing the victim virtual machines from suffering any alteration to their systems. By activating Snort in monitoring mode, metrics such as availability, CPU, and memory were obtained. Non-parametric methods were used for statistical analysis. The results obtained demonstrate the timely and reliable storage and detection of DoS attacks upon receipt. Blockchain integration helps create a more reliable system for events related to cybersecurity, IP address storage, and Snort rule-defined mechanisms.

Keywords: Blockchain, Snort, DoS, Docker.

Reviewed by:
Mgs. Hugo Romero
ENGLISH PROFESSOR
C.C. 0603156258

CAPÍTULO I INTRODUCCIÓN

1.1 Antecedentes

La disponibilidad, confidencialidad e integridad de la información son pilares centrales para la operación segura de organizaciones en la era digital; sin embargo, estos aspectos se ven sistemáticamente amenazados por ataques cibernéticos que aumentan en frecuencia y sofisticación. La madurez en la gestión de ciberseguridad sigue siendo baja en varias instituciones y sectores, lo que incrementa la probabilidad de incidentes críticos si no se adoptan controles técnicos y de gobernanza más robustos [1]. Esto explica por qué los enfoques perimetrales como antivirus, firewalls aislados, ya no son suficientes: la arquitectura de protección debe contemplar trazabilidad, resistencia a la manipulación de evidencias y coordinación entre dominios, requisitos que orientan la búsqueda de soluciones complementarias como blockchain.

La tecnología blockchain introducen propiedades técnicas, descentralización, resistencia a la alteración de información mediante criptografía o hashing, así como trazabilidad de las transacciones las cuales la hacen objeto de interés para mejorar aspectos de la ciberseguridad como por ejemplo: la integridad o el no repudio de los registros de auditoría, en el caso de que se aplique el blockchain de la manera que se considera correcta como registrar sólo metadatos y hashes on-chain y dejar los volúmenes altos off-chain, el blockchain mejora la capacidad de la organización para registrar la manipulación de los datos y reconstruir cadenas de eventos [2]. De este modo, el blockchain puede complementar los sistemas de detección tradicionales (IDS/IPS) al asegurar que los eventos y las evidencias no sean fácilmente alterados, lo que a su vez puede mejorar la respuesta forense; sin embargo, la adopción del blockchain implica decisiones de diseño permisionada vs pública, off-chain para datos pesados y para equilibrar seguridad/rendimiento

Las plataformas permisionadas, como el caso de Hyperledger Fabric otorgan beneficios prácticos para contextos institucionales, control de la identidad, canales privados y mejores métricas de rendimiento en comparación con cadenas públicas cuando se requiere gobernanza restringida. Estas características permiten mitigar riesgos inherentes a los

contextos operacionales (latencia, privacidad, escalado transaccional) y hacen posible el registro distribuido de eventos de seguridad en redes empresariales o gubernamentales [3]. Apostar por una red permisionada provoca menos fricciones legales y técnicas, acceso, control de datos e implica que sea posible diseñar, por ejemplo, mecanismos automáticos smart contracts/chaincode; que orquesten respuestas entre nodos autorizados; en todo caso, su uso requeriría una evaluación empírica de overhead en latencia y consumo de recursos.

En el caso particular de Ecuador y contextos similares de la región, los informes y estrategias nacionales indican un incremento sostenido de incidentes y una necesidad explícita de fortalecer capacidades técnicas y políticas públicas; esto crea un contexto propicio para explorar soluciones híbridas (blockchain + detección tradicional) que mejoren la resiliencia de infraestructuras críticas y la transparencia en la gestión de incidentes. Aplicar y validar arquitecturas basadas en blockchain en un testbed permitirá ofrecer recomendaciones específicas para el despliegue en entidades públicas y privadas, ajustadas a las limitaciones operativas del país. [1]

Los beneficios medibles en trazabilidad y reducción de manipulación de datos cuando blockchain se usa adecuadamente son muy importantes; sin embargo, existen limitaciones prácticas como el riesgo humano, vectores de ataque contra capas periféricas, y costos de replicación, lo que apunta a que blockchain no es una panacea sino un componente más dentro de una arquitectura de ciberseguridad [4]. Desde la perspectiva metodológica, esto obliga a la tesis a diseñar experimentos controlados que cuantifiquen tanto mejoras de seguridad (integridad, detección colaborativa) como los costos (latencia, TPS, consumo de recursos), para ofrecer una evaluación realista y recomendaciones técnicas aplicables.

En un sentido operativo, la forma más racional y reproducible para una evaluación consiste en añadir sensores/IDS como, Suricata o Zeek que emitan un evento firmado hacia una puerta de enlace en la que se registren hashes o metadatos en una blockchain permisionada y que almacene artefactos muy pesados offchain (IPFS o BD cifrado). Los smart contracts muy probablemente pueden encargarse de aspectos como las validaciones y llegar a disparar acciones relacionadas con controladores SDN y/o sistemas de acciones de bloqueo, mejorando el tiempo de respuesta coordinado entre los diferentes dominios. Esta forma arquitectónica de sensores → gateway → hashes on-chain + off-chain datos → smart

contracts → orquestación; expone que se debe evaluar la seguridad, su interoperabilidad y su impacto posterior en el desempeño de red, que será crítico cuando llegue el momento de validar la viabilidad en situaciones más aplicables de telecomunicaciones.

1.2 Planteamiento del problema

El mundo digital ha crecido de forma constante en los últimos años, al mismo tiempo han ido aumentando los ataques cibernéticos, estos ya no son incidentes aislados o exploits simples, sino auténticas campañas que realizan ataques organizados, utilizando técnicas como el phishing masivo, ransomware de doble extorsión, ataques DDoS y explotación de vulnerabilidades conocidas, atacando a empresas, gobiernos y a cualquier usuario en general. El crecimiento sostenido en el número de incidentes de ciberseguridad en América Latina, donde sectores como la educación, el gobierno y las telecomunicaciones llegan a sufrir centenares de ataques semanales, muchos de ellos sofisticados y persistentes [5].

Este problema creciente de las amenazas muestra las limitaciones que presentan los modos tradicionales de abordar la seguridad, los cuales reposan en sistemas centralizados como firewalls o servidores de autenticación que son vulnerables a fallos y manipulaciones. Los estudios realizados sobre blockchain han mostrado que su estructura descentralizada y criptográficamente segura presenta claras ventajas para garantizar la integridad y trazabilidad de los datos, lo que podría ayudar a solventar tales limitaciones. Sin embargo, en la actualidad son muy escasos los trabajos aplicados que muestren empíricamente cómo se puede hacer uso de blockchain para ayudar a reforzar mecanismos de detección, registro y respuesta ante ataques reales en entornos de red, quedando una brecha importante por cubrir en la literatura científica en español sobre este tema [6].

Este aspecto de la ciberseguridad presenta especial importancia por distintas variables de orden estructural, por ejemplo, la escasa inversión en tecnologías defensivas, la inexistencia de protocolos normalizados de respuesta a incidentes o la alta dependencia de sistemas legados inseguros. En la región muestra uno de los mayores aumentos relativos en el volumen de incidentes de ciberataques con incrementos de hasta el 25% por año en algunos países y el impacto creciente del ransomware sobre las instituciones públicas y privadas [7].

En tal sentido el problema central de esta investigación se plantea el siguiente: ¿Cómo puede un sistema de ciberseguridad basado en tecnología blockchain contribuir a la reducción de ataques cibernéticos, garantizando la integridad, trazabilidad y confiabilidad de la información?

1.3 Justificación

La ciberseguridad, hoy en día puede considerarse un tema preocupante para las organizaciones y los usuarios, debido a un aumento desmesurado de ataques que exponen información sensible, comprometen sistemas y que, incluso, generan pérdidas económicas y de reputación. Como señala un reciente informe, América Latina está experimentando de forma sostenida un aumento en la cantidad de ataques cibernéticos en los últimos años que actualmente se ven afectados las empresas, los gobiernos y las personas [8]. Esto hace evidente que los mecanismos de protección tradicionales se están quedando obsoletos en la lucha contra los ataques sofisticados, lo que hace necesaria la exploración de nuevas soluciones tecnológicas que refuercen las defensas y reduzcan la posibilidad de que sucedan violaciones de la seguridad.

Los sistemas de seguridad tradicionales como firewalls, antivirus e IDS/IPS fueron los baluartes de la protección de redes y datos, pero su enfoque centralizado tiene vulnerabilidades especialmente cuando el ataque sólo se encuentra dentro de la misma, al igual que en el caso que un atacante obtenga el acceso al servidor de registro o de autenticación mediante una inyección de código. Recientemente se han indicado que aun con varias capas de seguridad, los métodos tradicionales pueden ser burlados si un atacante es capaz de obtener credenciales privilegiadas o hackear un nodo central [5]. Este tipo de situaciones han puesto en exposición la necesidad de pensar en arquitecturas alternativas ajenas a una autoridad central.

La tecnología blockchain ha surgido como una alternativa prometedora para incrementar la seguridad digital por sus características de descentralización, inmutabilidad y trazabilidad. Estudios han demostrado que la inclusión de la blockchain en los procesos de gestión de registros, autenticación y auditoría de forma separada, incrementa la integridad y fiabilidad de la información, disminuyendo el riesgo de alteraciones o borrados maliciosos

de los datos [6]. Como consecuencia de esto el uso de la blockchain dentro de un sistema de seguridad informática, se podrían diseñar defensas más potentes, en los entornos donde la trasparencia de las evidencias y resistencia a la manipulación sean criterios más importantes.

Además, blockchain permite mejorar la integridad y trazabilidad de los registros y también admite la creación de mecanismos de autenticación distribuidos que no dependan de servidores centrales, disminuyendo así el riesgo de sufrir ataques por suplantación de identidad o robo de credenciales. Un claro ejemplo de ello es la posibilidad de utilizar las claves públicas y privadas dentro de una red blockchain que permita verificar la legitimidad de los usuarios y de los dispositivos sin tener que confiar en un motor de autenticación centrado en un servidor y que suele ser bastante vulnerable [9]. Es entonces, una forma más robusta y resistente de gestionar la identidad digital, alcanzando este objetivo de forma particular en sistemas de telecomunicaciones y servicios críticos.

1.4 Objetivos

1.4.1 General

Implementar un sistema de ciberseguridad aplicando tecnología blockchain para la detección y registro de ataques de denegación de servicio en un entorno simulado.

1.4.2 Específicos

- Investigar sobre los principios fundamentales de la tecnología blockchain en entornos digitales mediante el método analítico y su aplicabilidad en sistemas de ciberseguridad para fortalecer conocimientos teóricos.
- Diseñar un sistema de ciberseguridad basado en tecnología blockchain, orientado a la detección de ataques informáticos, así como el registro y almacenamiento inmutable de la información asociada a dichos incidentes.
- Implementar un prototipo funcional del sistema en entornos simulados, que permita la captura, procesamiento y almacenamiento de eventos de seguridad mediante la simulación de ataques de denegación de servicio (DoS).

- Evaluar el desempeño del sistema propuesto en términos de seguridad, latencia y confiabilidad, así como documentar el proceso de desarrollo y los resultados obtenidos.

CAPÍTULO II MARCO TEÓRICO

2.1 La ciberseguridad

La ciberseguridad surge como respuesta necesaria al creciente uso de tecnologías digitales, redes interconectadas y servicios en línea, donde la información deja de estar en papel y pasa a viajar por redes vulnerables. Garantizar que la información permanezca segura significa mucho más que simplemente proteger un sistema: implica usar un marco conceptual claro que oriente todo diseño de seguridad. Ese marco esencial se conoce como la Tríada CIA que integra la Confidencialidad, Integridad y Disponibilidad, los tres pilares fundamentales sobre los cuales se deben construir políticas, controles y arquitecturas de seguridad. [9]

Figura 1. CIA triada [10]

Confidencialidad: Cuando se habla de la propiedad de la confidencialidad se afirma que los datos y los recursos solo pueden ser accedidos por aquellos que cuentan con autorización, previniendo que una información sensible tenga acceso de una forma no permitida

Integridad: Hace referencia a garantizar que la información no haya sido modificada de una forma no dispuesta y que la información continúa siendo veraz, confiable y consistente a lo largo de su ciclo de vida.

Disponibilidad: Se refiere a garantizar que los sistemas y los datos sean accesibles en todo momento ante la solicitud de aquellos que sean los usuarios habilitados, garantizando

de esta forma que no existan interrupciones posibles y que puedan comprometer de esta forma los distintos servicios u operaciones. [11]

2.2 Amenazas digitales

La realidad hoy en día es que los ataques informáticos ya no son como los "virus de antaño" que simplemente se replicaban o borraban carpetas: el panorama se ha vuelto mucho más sofisticado, variado y peligroso. Ahora, los delincuentes combinan malware, ransomware, herramientas de acceso remoto, phishing dirigido, ataques a la identidad y campañas masivas de denegación de servicio (DoS), además de apuntar a infraestructuras críticas y cadenas de suministro digital. Un informe reciente advierte que en América Latina se detectan más de 3.1 millones de ataques de malware por día, lo que evidencia la magnitud del problema [12]. Este cambio implica que no basta con tener un antivirus únicamente, los mecanismos de defensa deben adaptarse para enfrentar amenazas mucho más dinámicas, persistentes y multifacéticas.

Tabla 1. Principales amenazas digitales

Tipo de ataque	Vector principal/vía de ataque	Impacto típico	Mecanismo tradicional de mitigación
Malware / Ransomware	A través de descarga de archivos, phishing	Perdida o secuestro de datos e información	Antivirus, backups, segmentación de red
DDoS	Mediante el tráfico masico desde varias fuentes	Caída del servicio o indisponibilidad	Firewalls, filtros, ratelimitng, mitigación por terceros
MITM / Intercepción	Por redes inseguras, sniffing, falta de crifrado	Intercepta o manipula datos en tránsito	Cifrado (TLS / VPN), IDS/IPS
Phishing / Ingeniería social	A través de correos engañosos a usuarios	Robo de credencias, suplantación de identidad	Con educación, autentificación multifactor, monitoreo
Manipulación de logs / evidencias	Por acceso interno, ataques post-explotación	Oculta datos, borra evidencia.	Sistemas de logging centralizados auditorías.

Como las amenazas se expanden en técnicas y vectores, las defensas tradicionales resultan insuficientes al compararlas con arquitecturas distribuidas e integradas. Esto impulsa la

necesidad de mecanismos que prevengan ataques y aseguren la integridad de los eventos y la trazabilidad.

2.3 Sistemas tradicionales

En sistemas informáticos convencionales, uno de los primeros escudos construidos en la defensa de infraestructuras frente a ciertos peligros que amenazan es lo que en seguridad llama defensa perimetral. Lo que, en palabras informáticas, significa poner un muro virtual que controle quienes son los que entran y salen, que datos son los que se desplazan dentro de la red. Herramientas como un firewall (o cortafuegos), IDS/IPS, antivirus, VPN son los principales componentes de este muro de defensa., cada uno de ellos con una función en la defensa, pero interrelacionados para conseguir proteger nuestro sistema. Un firewall sería la primera línea de defensa ya que concentra el tráfico de red que puede entrar o salir, bajo unos criterios previamente definidos de direcciones, puertos o de protocolos. De esta forma se puede intentar evitar accesos indeseados, intentos de intrusión, conexiones malintencionadas o tráfico malicioso, además añadir a este control, la inspección profunda de paquetes Deep Packet Inspection, DPI, el filtrado web, el control aplicaciones, el resultado y efectividad se incrementa significativamente. [13]

Pero el firewall por sí solo no alcanza para todo. Por ello se usan IDS (Intrusion Detection System) o IPS (Intrusion Prevention System): el IDS observa el tráfico, detecta patrones o anomalías que coinciden con ataques conocidos o comportamientos sospechosos, y alerta al administrador; mientras que el IPS va más allá: cuando detecta algo malicioso, interviene en tiempo real para bloquear o mitigar la amenaza. Esto es crucial para frenar ataques que logran pasar las reglas de firewall, ya que muchos ataques modernos usan técnicas evasivas, cifrado, variación de puertos o payloads que un simple filtrado por IP/puerto no detecta. [14]

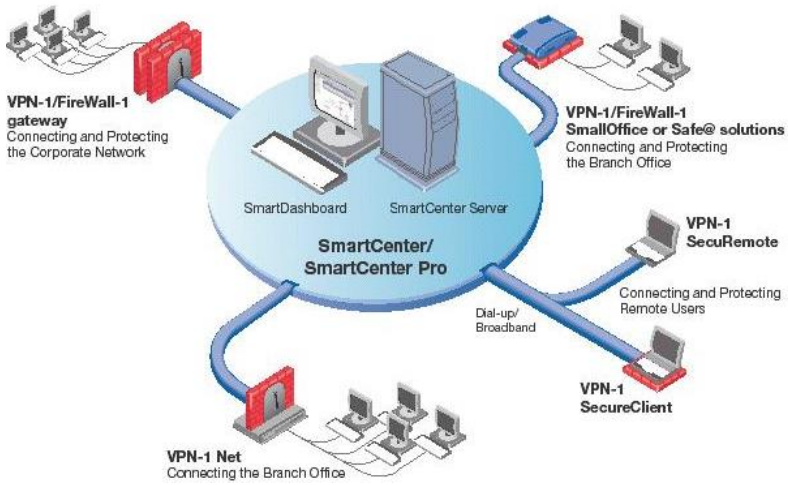


Figura 2. Arquitectura de defensa perimetral [15]

A su vez, el antivirus complementa la protección de una manera distinta, pero a nivel de los dispositivos: analiza archivos, procesos, detecta malware, spyware, troyanos o ransomware, antes de que logren diferenciarse y propagarse. Esto cubre una capa adicional, la del endpoint, en donde muchas veces inicia la amenaza. La VPN (Virtual Private Network) es la herramienta que permite garantizar que la comunicación entre redes, red interna o los usuarios remotos, mantengan un trayecto cifrado, evitando el riesgo de que los atacantes puedan interceptar datos sensibles en tránsito, de esta manera, protegen el nivel de la confidencialidad y la integridad del canal de comunicación. [14]

Pero lo más importante es que estos mecanismos no deben verse como elementos aislados funcionan mejor cuando se integran bajo un enfoque de defensa en profundidad, es decir, varias capas de protección: perímetro (firewall + IPS), red (segmentación, VPN), endpoints (antivirus), monitoreo continuo (IDS), control de acceso, cifrado, backups, etc. De esta forma, si un componente falla o es vulnerado, hay otros que impiden el acceso, la propagación o el daño masivo. [16]

2.4 Arquitectura basad en Blockchain

Las organizaciones en la actualidad, no se conforman solamente con un antivirus, un firewall o un servidor seguro, puesto que los ataques y vulnerabilidades a los que tienen que hacer frente son muy más sofisticados. Por este motivo empieza a interesar arquitecturas de seguridad basadas en blockchain a saber, un sistema que puede prescindir de un único punto de control una solución centralizada e implementar, por un lado, un esquema que permita

que múltiples nodos compartan la responsabilidad de, verificar transacciones, eventos, y garantizar que la información que ha sido registrada no pueda cambiarse mediante un consenso, lo cual permite implementar defensas distribuidas, que cuenten con el soporte criptográfico, así como la trazabilidad, la resiliencia frente a manipulaciones internas como externas; que es importante a la hora de hacer frente a información delicada, logs de auditoría, historiales transaccionales o identidad digital. [17]

Figura 3. Bloques encadenados [18]

Por otro lado, las arquitecturas de seguridad en blockchain permiten revertir el paradigma de confianza: ya no es la autoridad central quien garantiza la integridad, sino la propia red; incluso si un nodo ha sido comprometido, la estructura global siga siendo segura y la intervención de datos implicaría comprometer una mayoría importante de nodos, no solo un nodo, de este modo, se intensifica considerablemente el umbral para que un atacante sea capaz de intervenir datos; también, los sistemas basados en blockchain permiten auditoría distribuida, transparencia, control de versiones, historiales permanentes y automatización de políticas de control de acceso a través de smart contracts, por lo que este tipo de arquitectura es una transformación en la forma de realizar el diseño de seguridad de la información como una nueva capa de seguridad, y tambien es un cambio radical en el diseño de seguridad en la información y de la arquitectura de seguridad de la información, convirtiéndose esta arquitectura en un nuevo pilar de la misma. [19]

2.5 El Blockchain

La tecnología blockchain, en términos simples pero poderosos, se define como un "sistema basado en un libro mayor distribuido", o en un registro compartido por una multiplicidad de dispositivos: las transacciones o transacciones/eventos se agrupan en bloques, enlazando un bloque con el anterior mediante técnicas criptográficas, y todos los nodos validan mediante un consenso. Dicha estructura garantiza que, una vez grabado, algo no puede modificarse sin alterar toda la cadena de bloques, esto es lo que permite la integridad e inmutabilidad. [20]

Figura 4. Funcionamiento del blockchain

Al confiar en un servidor central, el mismo que puede ser atacado, manipulado o corrompido, blockchain establece la confianza en varios participantes, si detecta que alguien intenta cambiar un dato que ya se registró, inmediatamente la red detectará la inconsistencia, puesto que las copias no coinciden [19]. Esto vuelve a la tecnología completamente dinámica y atractiva para sistemas donde la trazabilidad, veracidad y resistencia a fraudes son críticas, como es el caso, en gestión documental, en las auditorías, historial de transacciones, identidad digital, contratos inteligentes, etc.

Atributos técnicos del blockchain

Entre los principales atributos que permite que blockchain sea especialmente útil en aspectos de seguridad, se destacan los siguientes:

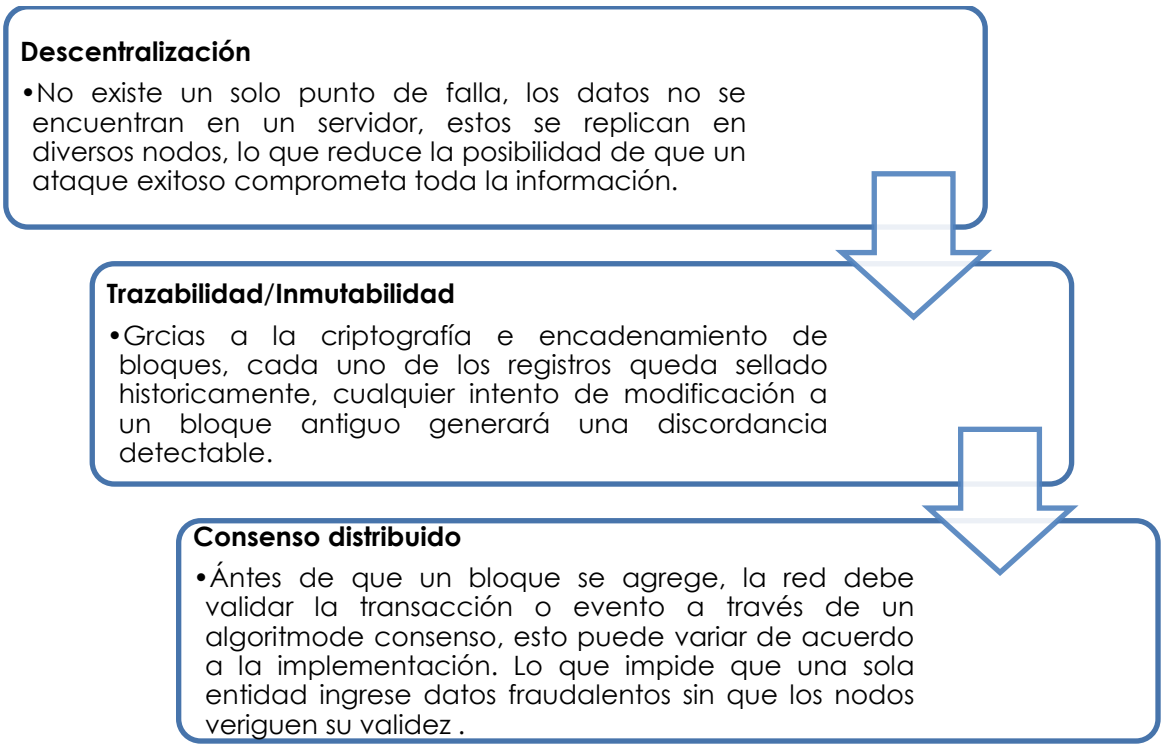


Figura 5. Atributos técnicos del blockchain [17],[19],[21]

En la práctica, blockchain es utilizado para el aseguramiento de gestión documental, procesos administrativos, datos clínicos, identidad digital, etc. Un claro ejemplo se puede evidenciar en el ámbito de la salud, donde se ha propuesto blockchain para asegurar historiales médicos, garantizando privacidad, duración y registro transparente de acceso [22]. Otro caso similar es en las instituciones educativas, donde blockchain se utiliza para garantizar los datos de admisión o diferentes documentos para que no sean manipulados o falsificados [23].

De igual forma, existe limitaciones si blockchain no logra resolver todos los problemas o inconvenientes, en varios casos debe combinarse con almacenamiento off-chain cuando son datos voluminosos, debe diseñarse con cuidado la gobernanza de nodos, mientras que la latencia o costo de consenso resulta un desafío. Para que logre funcionar como un sistema de seguridad real, necesita integrarse de forma adecuada con sistemas de autenticación, políticas de acceso, monitoreo y análisis de amenazas [17].

2.6 Componentes y funcionamiento técnico

Para comprender que blockchain en aspectos de seguridad, es importante y necesario conocer sus diferentes componentes y como estos funcionan internamente.

- Bloque: Este contiene un conjunto de transacciones o eventos, un hash del bloque anterior y un nonce que depende del mecanismo.
- Hash criptográfico: por su parte garantiza la integridad, al cambiar un solo bit de la transacción, el hash completamente cambia.
- Cadena de bloques: cada bloque apunta al anterior a través de su hash, lo que concierne a la inmutabilidad histórica.
- Consenso: estos mecanismos sirven para validar nuevos bloques como son los siguientes (Pow, PoS, pruebas de autoridad y permiso)
- Smart contracts: que se refiere a los contratos, es decir contratos inteligentes, el cual se basa en la lógica automatizada ejecutable al cumplirse condiciones, es necesario para la automatización de políticas de seguridad, y registro de eventos y validaciones. [9]

Blockchain no es magia, pero sí es una nueva y posiblemente interesante infraestructura para la ciberseguridad que, por su diseño descentralizado, inmutable y consensuado, le otorga ventajas frente a diseños tradicionales que basan su funcionamiento en servidores únicos. En un momento en el que las amenazas crecen y los datos se convierten en datos sensibles las arquitecturas basadas en Blockchain suponen una alternativa válida para agrandar la integridad y la trazabilidad y la resistencia para dar respuesta a los nuevos retos. No obstante, su diseño requiere de una construcción adecuada que mezcle lo mejor que ofrece el Blockchain con buenas prácticas de seguridad tradicional que podrían dar lugar a una mezcla lo suficientemente buena como para dar lugar a soluciones modernas, robustas y fiables.

2.7 Tipos de blockchain

La tecnología Blockchain en la actualidad no es la misma para todos los casos de uso, existen diferentes tipos con sus respectivas reglas, beneficios y también desventajas tratando de dar respuesta a dicha utilización. Por un lado, las blockchains públicas son abiertas a cualquiera, que puede leer en la red o escribir en ella, son ideales cuando la intención es buscar la

máxima transparencia, reiteración y descentralización. Por otro lado, las blockchains privadas o permisionadas y que no permiten que cualquiera pueda validar o acceder a la información, esto se traduce en un mayor control, privacidad y velocidad, algo útil para empresas o instituciones que gestionan información que consideran sensible. En medio de ambas se encuentran aquellas híbridas, que constituyen una especie de punto intermedio de pensadas para aunar lo mejor de ambas opciones; permiten la consulta de información pública, mientras que los datos privados o sensibles son controlados de alguna forma. [21]

Tabla 2. Tipos de blockchain

Tipo	Acceso - Participación	Ventajas	Desventajas - riesgos
Pública	Cualquier tipo de nodo puede participar, existen anonimato posible	Descentralización máxima, transparencia	Escalabilidad limitada, privacidad débil, altas latencias
Privada / permisionada	Solo acceden nodos autorizados	Control de acceso. Mejor rendimiento. Privacidad	Centralización parcial, confianza en nodos autorizados
Híbrida / consorcio	Mezcla de nodos autorizados y acceso controlado	Balance entre descentralización y gobernanza	Complejidad en su configuración.

La elección de un tipo de blockchain sea esta pública, privada o híbrida, no es únicamente una cuestión al azar, sino es una decisión estratégica, puesto que cada opción implica un equilibrio diferente ya sea en seguridad, velocidad, privacidad y como su descentralización; si se requiere de amplia transparencia y resistencia a censura, una pública es la más adecuada; mientras si se maneja datos sensibles o se necesita mayor control interno, lo ideal es una privada ya que da mejores garantías; y si se desea lo mejor de las dos partes, la híbrida es la que ofrece un buen servicio. Por consiguiente, comprender las diferencias de los diferentes tipos de blockchain es importante antes del diseño de un sistema, porque de la elección depende los aspectos esenciales como el rendimiento, confiabilidad, escalabilidad y protección de datos.

2.8 Integración de blockchain en sistemas de ciberseguridad

La integración de blockchain dentro de los sistemas de ciberseguridad son una estrategia llamativa para organizaciones que buscan mayor confiabilidad y trazabilidad en el

manejo de datos críticos. Esto se debe a que blockchain, a diferencia de los sistemas tradicionales, permite registrar información sin que pueda ser alterada, ocultada o eliminada sin dejar rastro, lo que lo convierte en un complemento ideal frente a amenazas como manipulación de logs, ataques internos, suplantación de identidad o intrusiones avanzadas. La naturaleza distribuida y criptográficamente protegida de la cadena de bloques funciona como un candado digital permanente, fortaleciendo procesos de auditoría, monitoreo y control frente a atacantes cada vez más sofisticados [24].

2.8.1 Blockchain como sistema de gestión de registros (logs)

El uso de blockchain como sistema de gestión de logs de eventos consiste en la inclusión de cada evento del sistema como bloque cifrado que se relaciona con el anterior de forma que un atacante no puede borrar las evidencias o modificar los logs, aunque haya comprometido el propio servidor. En redes tradicionales, un intruso que logra comprometer un servidor y obtiene privilegios de administrador puede borrar los logs o alterar las fechas de los logs con el fin de ocultar su rastro; sin embargo, en una cadena de bloques, cualquier intento de modificar los logs transgrediría la secuencia criptográfica de la cadena de bloques y se detectaría inmediatamente. Existen investigaciones recientes que avalan que la inmutabilidad y verificación distribuida de una cadena de bloques previenen que un atacante modifique los logs incluso si se compromete un nodo puesto que el resto de la red sigue teniendo la copia intacta [25].

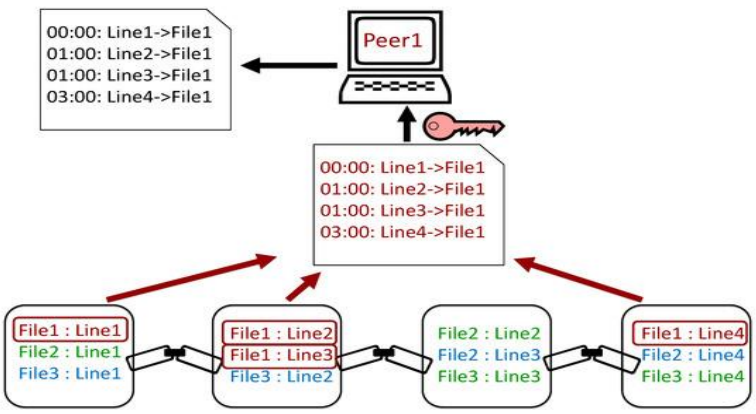


Figura 6. Registro blockchain

Además de permitir una conservación integral de los logs, la tecnología blockchain se considera un método de guardar los registros de los logs que concretamente permite una comprobación fácil y continua de la integridad de los logs y que incluso puede guardar un registro explícito del estado. En este sentido, en situaciones en los que hay que reconstruir la línea de tiempo de un ataque, por ejemplo, para comprender cómo entró un intruso, qué es lo que alteró o qué usuarios se vieron afectados, la cadena de bloques resulta de gran utilidad [26]. En auditorías forenses, la trazabilidad que permite esta tecnología se vuelve esencial para evitar contenciones de carácter interno sobre si la información se ha modificado antes de ser verificada o no. Una reciente línea de literatura sobre esta tecnología apunta que la cadena de bloques se comporta como un "testigo incorruptible" que preserva una historia de operaciones dentro del sistema, conservando la condición en la que se encontraba originalmente.

2.8.2 Manipulación de logs

Blockchain previene la alteración del registro de este programa porque cada registro está cifrado, y enlazado matemáticamente al bloque que antecede mediante funciones hash. Un atacante que quiera modificar un solo evento del log, debe alterar todos los hashes que siguen, y si tuviera éxito, sería necesario convencer a la mayoría de los nodos de aceptar dicha falsificación, lo cual es prácticamente imposible a no ser que tenga el control de la red. Esta propiedad que se alude desde algunos estudios recientes, es impidiendo destruir las pruebas, lo que convierten a blockchain en un depósito perfecto para la auditoria digital [27].

Además, dado que la información está disponible en varios nodos, un atacante no podría comprometer el log total, ni aunque pudiera llegar a comprometer un servidor central. De hecho, incluso si una máquina es atacada, la red tiene versiones verificadas de la historia, evitando alteraciones del tiempo de los registros y la inserción de eventos fraudulentos. En otras palabras, tampoco se puede "reescribir la historia", lo que reduce notablemente los riesgos de encubrimiento de los ataques e intentos maliciosos [27].

2.9 Blockchain para identificar y autenticar

La autenticación basada en blockchain introduce un modelo descentralizado donde las identidades ya no dependen de un servidor central que, si se cae o se vulnera, compromete a todos los usuarios. En su lugar, se usa un sistema distribuido donde cada identidad se valida criptográficamente mediante llaves y validaciones en la red. Esto reduce el riesgo de robo de credenciales o falsificación, muy común en sistemas tradicionales basados en contraseñas o bases de datos centralizadas. Estudios latinoamericanos recientes destacan que este modelo descentralizado ha demostrado mayor resistencia a ataques de suplantación y phishing dirigido[28].

2.9.1 Llaves públicas o privadas

El manejo de llaves públicas y privadas es el núcleo de la autenticación en blockchain; la llave privada únicamente la tiene el usuario y le sirve para firmar digitalmente transacciones o solicitudes; la llave pública, a diferencia de la llave privada, sí la comparte con la red para darse a conocer e identificar su identidad. Así es como se forma un sistema en el que nadie puede hacerse pasar por otra persona, porque en el fondo se hace uso de la criptografía que garantiza que únicamente la persona dueña de las llaves puede generar una firma válida para una transacción. Este modelo elimina uno de los puntos débiles más importantes de los sistemas clásicos: el almacenamiento de contraseñas en bases de datos inseguras [29].

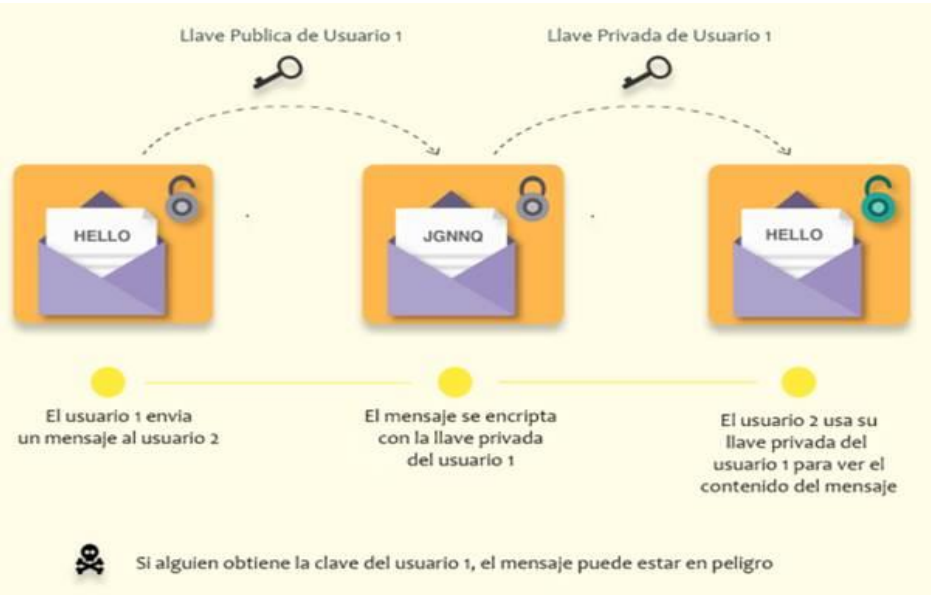


Figura 7. Llaves pública y privada [30]

Por otra parte, la autenticación criptográfica se vuelve una opción muy fuerte ante ataques de fuerza bruta, ataques de phishing o robo de credenciales; ni un atacante que robe la llave pública podrá usarla para suplantar a un usuario, dado que la clave está contenida dentro de la llave privada que jamás entra en circulación [30]. Este mecanismo puede evitar por completo las fugas masivas o los accesos indebidos, siendo una alternativa muy sólida para los servicios que manejan información sensible.

2.10 Blockchain para detección de intrusiones

La detección por medio de blockchain para registrar toda alerta producida por un IDS dentro de una cadena de bloques inmutable hace que los eventos no sean removidos, ni alterados por los atacantes, lo que refuerza por lo tanto la detección ante incidentes como así también el análisis posterior forense, ya que toda la información que constituyó el ataque queda registrada como prueba verificable, realizar la integración entre IDS y blockchain hace aumentar la precisión, así como reducir el riesgo de manipulación en caso de ataques avanzados [31].

Este mecanismo también permite confrontar las diferentes alertas que surgen de nodos dispares para determinar si un sospechoso comportamiento se mantiene o no, o incluso si se trata de un falso positivo. Por tanto, al compartir la información de seguridad sobre el estado de la red con varios participantes se obtiene una imagen más precisa y veraz de la situación, y como consecuencia se hace más eficiente el sistema frente a las amenazas persistentes. La literatura de referencia reciente sugiere que este modelo colaborativo contribuye a incrementar la capacidad de respuesta del sistema ante las amenazas distribuidas [32].

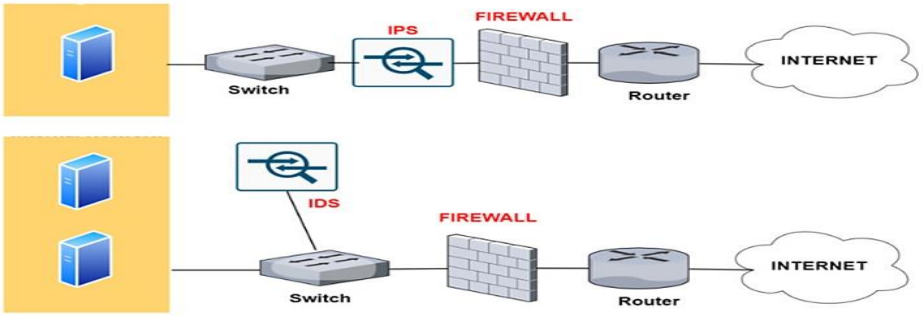


Figura 8. Diagrama de un IPS e IDS [33]

2.10.1 Flujo operativo

El flujo operativo se basa en el siguiente proceso: el IDS identifica un comportamiento anómalo y genera una alerta; un smart contract procesa automáticamente ese evento y lo registra en la blockchain; los nodos validan si la alerta refleja un patrón de comportamiento o si se ha interrumpido el flujo de datos. En todo caso, este flujo permite que el sistema mantenga las propiedades deseadas de integridad, transparencia y trazabilidad [34].

IDS genera alerta	El momento que los IDS identifican tráfico sospechosos, como escaneos de puertos, patrones ançomadalos, este genera una alerta que queda registrado en un servidor local, pero si este servido se compromete la alerta puede ser eliminada. Con blockchain la alerta se respalda por toda la red.
Smarte Contract	Este funciona como intermediario automático que recibe la alerta del IDS y lo convierte en una transacción inmutable, lo que garantiza que el evento quede registrado sin intervención humana, evitando manipulación o pérdida de evidencia.
Nodo	Los nodos de la red verifican que la alerta no haya sido modificada y que corresponda con patrones válido. Si existe inconsistencias el sistema detecta posible manipulación, lo que refuerza la seguridad del proceso.

Figura 9. Flujo operativo [34]

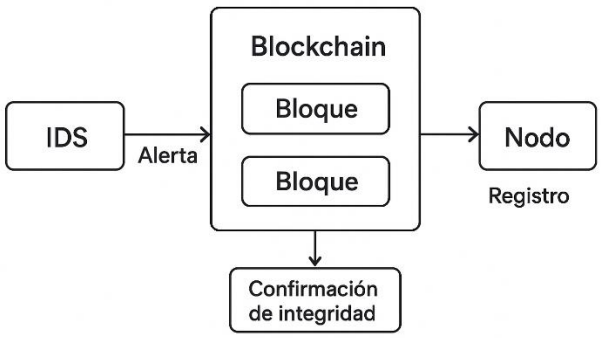


Figura 10. Diagrama de la arquitectura IDS-Blockchain

2.11 Blockchain como sistema anti – ataques

Blockchain funciona como un mecanismo anti – ataques, ya que cada uno de sus componentes como consenso, criptografía y distribución actúan como barreras contra la manipulación, modificación o suplantación de datos o información. Mientras que los sistemas tradicionales se alimentan de la protección perimetral, blockchain se protege desde dentro, garantizando que cada trozo de información que tiene dentro de la red sea inalterable, trazable y verificable. La consecuencia es que el impacto de ataques como MITM, manipulación de logs o alteración de información sensible queda reducido drásticamente [6].

Ofrece además un nivel de resiliencia que los sistemas clásicos ni siquiera pueden emular, en particular cuando se contraataca mediante ataques internos; un empleado malicioso o un atacante con privilegios altos no puede modificar registros sin dejar huella. La cadena de bloques fuerza a que todas las acciones que se han ejecutado sean verificadas de forma distribuida, generando un ambiente más seguro en particular las infraestructuras donde se valora la integridad y confidencialidad [6].

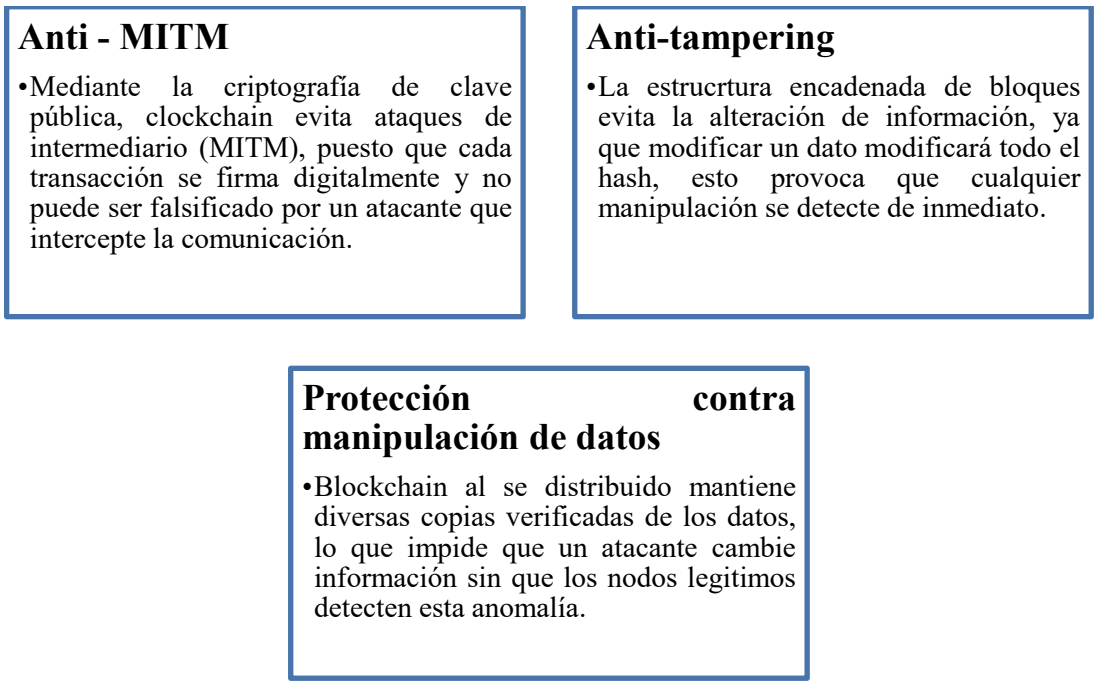


Figura 11. Capacidades destacadas del blockchain

CAPÍTULO III METODOLOGÍA.

La metodología presenta el enfoque general y los diferentes métodos utilizados para el desarrollo de esta investigación, el cual garantiza que los objetivos establecidos se cumplan de manera ordenada y eficaz. El propósito primordial es describir el proceso realizado para la recopilación, análisis e interpretación de los datos, de esta forma se proporciona calidad y solidez científica a este estudio. Al desarrollar el tipo de investigación, enfoques utilizados y herramientas aplicadas, la metodología asegura que los resultados que se han obtenido son veraces, verificables y se alinean con los objetivos propuestos en el proyecto.

3.1 Tipo de Investigación.

El proyecto con el título "Sistema de ciberseguridad para reducir ataques cibernéticos aplicando tecnología blockchain" Se enfoca en investigar y evaluar los datos mediante diferentes pruebas, lo que implica que la investigación contenga diversos métodos como bibliográfica, de campo, experimental y descriptiva.

3.1.1 Investigación bibliográfica

La investigación bibliográfica fue fundamental para la consecución de este proyecto, ya que se pudo obtener información fiable y actualizada respecto a ciberseguridad, blockchain y ataques cibernéticos, para ello se acudió a libros, artículos indexados, revistas académicas, tesis y repositorios nacionales e internacionales. Este proceso contribuyó a reunir antecedentes teóricos, examinar avances recientes en la protección digital y a conocer cómo otras investigaciones que han aplicado tecnologías similares. Una revisión bibliográfica es fundamental para fundamentar un estudio puesto que proporcionará un panorama actualizado del conocimiento existente y permite encontrar vacíos que justifiquen la investigación [35].

3.1.2 Investigación de campo

La investigación de campo es parte esencial del propio proyecto, porque permite la obtención de la información desde el mismo ámbito en que se produce el fenómeno a estudiar. A

diferencia de la investigación documental, en la investigación de campo se recogen datos reales, datos obtenidos mediante observaciones, entrevistas, test y experimentos, ya sea en ambientes naturales o en ambientes simulados. En el caso del sistema de ciberseguridad basado en blockchain, la investigación de campo permitió conocer el funcionamiento real del sistema al evaluar con qué problemas prácticos se encuentra y realizar los ajustes de parámetros según el comportamiento real del sistema, La investigación de campo permite obtener datos verídicos y situados, que son vitales para validar sistemas tecnológicos en condiciones muy parecidas a la realidad [36].

3.1.3 Investigación experimental

En primer lugar, la investigación experimental fue la forma de investigación que se utilizó porque permite manipular las variables de un contexto controlado y observar la influencia que tienen unas, o varias, sobre las otras. Es especialmente adecuada en los proyectos tecnológicos porque permite la realización de pruebas de los sistemas, la medida del rendimiento o la evaluación de la seguridad, al poder variar determinados parámetros y obtener directamente las consecuencias de tal intervención. Por consiguiente, lo experimental consiste en poner a prueba los diferentes componentes del sistema de medida propuesta desde la ciberseguridad basada en blockchain, al igual de simular eventos, realizar ataques y observar cómo reacciona el sistema. La investigación experimental permite establecer relaciones de causa – efecto y poder comprobar si la intervención realizada genera algún tipo de mejora en la acción del sistema [37].

3.1.4 Investigación descriptiva

La investigación descriptiva también se contempla en la investigación, puesto que permite observar y recoger fenómenos, sin manipularlos. La investigación descriptiva es adecuada cuando se quiere saber cómo se comporta un sistema, qué características tiene o qué patrones se repiten. La investigación descriptiva es la parte que ayuda a caracterizar los tipos de ciberataques más comunes, recoger vulnerabilidades presentes o los cambios en el comportamiento de unas variables antes y después de aplicar el sistema propuesto. La investigación descriptiva deja conocer a fondo un problema real a partir de datos

recolectados en la realidad, encuestas, observaciones o análisis de bases de datos, que ayudan a conocer el sistema antes de intervenirlo [35].

3.2 Diseño de Investigación

En la primera etapa se buscará toda la información posible de trabajos realizados con el tema planteado por medio de trabajos investigativos, revistas, tesis y demás recursos bibliográficos y se tomará en consideración los contenidos más relevantes para llevar a cabo la investigación comprendiendo su forma de funcionar para posteriormente realizar simulaciones.

En la segunda etapa se levantará un escenario aplicando la tecnología de cadena de bloques empleando un software informático.

En la tercera etapa se generará ataques simulados para comprobar la robustez del sistema y su capacidad de reacción ante la detección de vulnerabilidad, evaluación continua de la integridad, seguridad y confiabilidad de la información que se encuentra almacenada en la base de datos.

La cuarta etapa será la evaluación y análisis de los resultados obtenidos para la aplicación de la tecnología.

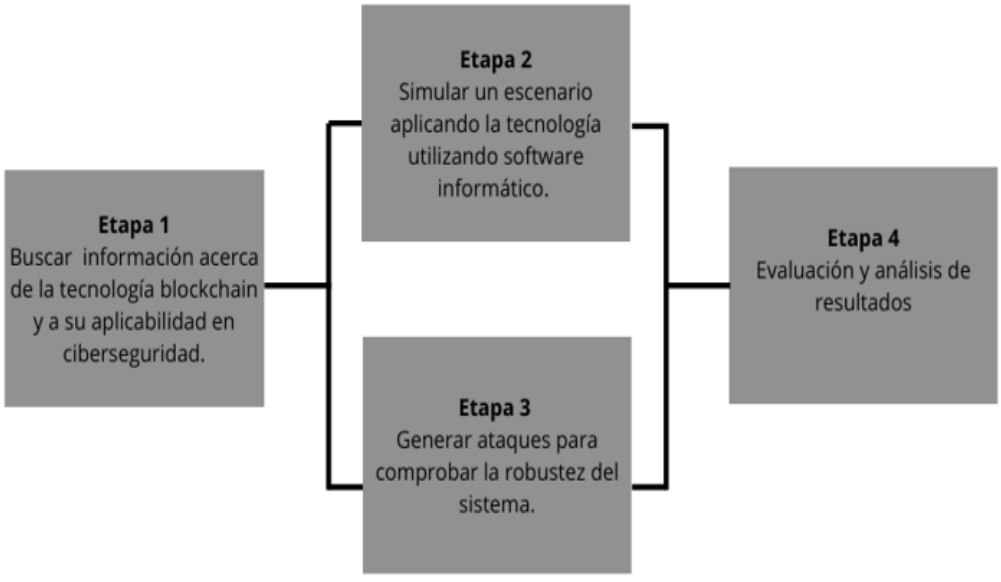


Figura 12. Proceso para la elaboración del proyecto de investigación

3.3 Técnicas de recolección de Datos,

La técnica de investigación fue a través de un experimento controlado donde se manipularán las variables independientes en este caso será el sistema de ciberseguridad basado en blockchain para observar su efecto con respecto a la variable dependiente.

3.4 Población de estudio y tamaño de muestra,

3.4.1 Población

Ataques de denegación de servicio operando en entornos simulados dentro del sistema de ciberseguridad.

3.4.2 Muestra

Un conjunto especifico de datos obtenido durante las simulaciones por cada ataque ICMP Flood, UDP Flood y SYN Flood. Estos sistemas serán evaluados en entornos simulados para medir la eficiencia en la detección de ciberataques tipo DoS.

3.5 Hipótesis

La implementación de un sistema de ciberseguridad basado en tecnología blockchain evita la manipulación de registros, mejora la detección temprana de intrusiones y disminuye la vulnerabilidad ante ataques informáticos.

3.6 Operacionalización de variables

Tabla 3. Operacionalización de variables

Variable	Nombre Variable	Descripción	Indicador	Técnica instrumento
Dependientes				
-Eficiencia del sistema en la detección de ciberataques.	Indicadores de rendimiento del sistema diseñado.	Conjunto de técnicas empleadas para verificar el nivel de afectación de los ataques.	Porcentajes en Latencia (%), CPU (%), MEM (%) y Disponibilidad (%).	Pruebas y simulaciones en entornos simulados. Registro de incidentes.
-Tiempo de respuesta del sistema.	Tiempo de respuesta ante ciberataques.	Tiempo que tarda el sistema en responder a un ciberataque desde su detección hasta la acción tomada en entornos simulados.	Tiempo de respuesta esta medida en segundos.	Registro de tiempo de respuesta durante pruebas y simulaciones en entornos simulados.
Independiente Diseño del sistema basado en blockchain.	Complejidad del sistema de ciberseguridad.	Proceso de creación y desarrollo de un sistema de seguridad cibernética que emplee a la tecnología blockchain, capaz de ser implementado en entornos simulados.	Número de simulaciones realizadas.	Registro de simulaciones realizadas. Análisis de logs del sistema.

3.7 Procedimiento

En este capítulo se detalla el proceso de configuración llevado a cabo para la simulación de un sistema de ciberseguridad utilizando tecnología blockchain, para tener un registro digital debido a que presenta características importantes en seguridad, eficiencia y confianza en los datos almacenados. Se implemento una blockchain privada basada en Hyperledger fabric, permitiendo crear infraestructuras de manera distribuida, certificados CA, red de prueba, nodos, MSP para posteriormente desplegar un chaincode.

Se realiza pruebas de ataques de DOS donde se pone a prueba el sistema, enviando gran cantidad de tráfico con el objetivo de saturar el ancho de banda a través de un ataque ICMP

Flod, la infraestructura blockchain registra información de los equipos atacantes evitando cambios, modificaciones o alteraciones, estos datos son detectados a través de una herramienta que registra el tráfico malicioso en tiempo real.

3.7.1 Preparación del entorno

Para llevar a cabo la implementación de la plataforma de ciberseguridad, como fase inicial se procedió a la descarga e instalación de herramientas necesarias para ejecutar Hiperledger Fabric.

3.7.2 Docker Compose

Es un componente indispensable que permite el aislamiento de los contenedores donde se alojan diferentes elementos de Fabric, entre ellos tenemos peers, orderers, servicios de certificación (CA) y demás herramientas auxiliares, el docker ayuda a tener una automatización al momento del despliegue despliegue de la red, permitiendo entornos reproducibles y asegurando consistencia en los nodos durante diferentes pruebas.

3.7.3 Instalación de Node.js (v18 LTS) y npm

Es necesario la instalación de estas herramientas para llevar a cabo el desarrollo y configuración de una red privada de blockchain implementada en Hypperledger Fabric, facilitando la ejecución de aplicaciones y despliegue del chaincode que son configuradas en el lenguaje JavaScript, la versión elegida debido a su compatibilidad con el framework.

3.7.4 Instalación de Go (Golang)

Para la compilación del chaincode se necesita la instalación del lenguaje de programación Go ya que es un lenguaje nativo que permite la realización de contratos inteligentes en la plataforma Fabric con el comando.

```
sudo apt install golang-go
```

3.7.5 Despliegue de la infraestructura de Hyperledger Fabric

Después de haber preparado el entorno para continuar con el sistema de ciberseguridad se procede a descargar componentes e iniciar con Fabric.

3.7.6 Descarga de binarios y herramientas necesarias

Para la instalación de Hiperledger Fabric y posteriormente ejecución de una red privada se procede a la descarga de binarios oficiales que permitan las diferentes configuraciones, generación de certificados MSP, scripts que automatizan el despliegue de la red, habilitando nodos, canales privados, contribuyendo una base en la cual se realiza diferentes pruebas y herramientas necesarias para manejar y monitorear la red.

```
curl -sSL https://bit.ly/2ysbOFE | bash -s -- 2.5.0 1.5.6
```

3.7.7 Levantar red de prueba

Con los procesos anteriores instalados correctamente sus versiones y demás librerías se levanta la red de prueba dentro del directorio fabric sample/test network, esta incluye ejemplos oficiales (test-network). Es una parte fundamental tener la red de prueba ya que contiene nodos, canales, chaincode y facilita el manejo de un entorno controlado.

Estando dentro de la carpeta test network se levanta los nodos de la red, elijo un nombre al canal de comunicación que voy a crear y ejecuto el script network.sh que contiene parámetros ideales, se crea servicios de certificación que permiten gestionar a los usuarios por medio de certificados digitales.

```
./network.sh up createChannel -c mychannel -ca
```

Después de realizar el levantamiento de prueba los nodos tendrán contenedores docker y servicios auxiliares necesarios, se genera configuraciones para el canal entro ellos el certificado criptográfico (MSP) para cada nodo. La red cumple con los requisitos necesarios para una instalación y despliegue de contratos inteligentes, asegurando, integridad, trazabilidad y seguridad de la información que se encuentra en el canal.

3.7.8 Despliegue y activación de chaincode en la red Hyperledger Fabric

Si la red de prueba está funcional y los nodos están activos el siguiente paso es la habilitación del contrato inteligente, en este se define el proceso que va a llevar la red blockchain por medio de ello se puede consultar y registrar la información que se encuentra en el ledger (registro completo y ordenado de lo que se realiza en la red blockchain).

3.7.9 Selección del contrato inteligente (chaincode)

Existen varios smart contract pero el que se eligió es fabcar, este permite de manera sencilla el registro de recursos, consulta de información, simplicidad y activación dentro de un entorno de Fabric, soporta lenguajes como Go y JavaScript.

3.7.10 Directorio de implementación chaincode/fabcar

Dentro de este directorio se crea un package.json, es un componente importante en proyectos a través de ello gestiona dependencias y scripts, facilitando componentes indispensables para que exista conectividad entre el contrato y el ledger y puedan realizar operaciones de blockchain.

3.7.11 Creación del archivo fabcar.js

Este archivo contiene las definiciones de un smart contract que permite realizar una gestión de una red que se encuentra dentro de blockchain basado en el lenguaje de JavaScript, en su configuración presenta CreateEvent y GetAllEvent.

A través de este contrato se puede crear eventos y obtener la información que se encuentra registrada en el ledger como tipo de evento, descripción, ip, fecha y hora asegurando que los registros sean transparentes y accesibles, utiliza librerías que se encuentran dentro de Hyperledger Fabric permitiendo interactuar con la tecnología blockchain desde un contrato inteligente.

3.7.12 Creación del archivo index.js

Index.js tiene la finalidad de registrar el chaincode que se llama fabcar, para posteriormente pueda ser ejecutado en la red dentro de Hyperledger Fabric.

3.7.13 Instalación y configuración del sistema de detección de intrusiones Snort

El Snort es utilizado para la detección y análisis de tráfico en la red en tiempo real, sus características permiten observar actividades maliciosas, ciberataques, patrones de ataque, el IDS es muy utilizado cuando se trabaja en proyectos basados en ciberseguridad ya que presenta grandes cualidades en rendimiento, flexibilidad, puede trabajar en detección pasiva y detección activa, esta herramienta es muy útil en entornos educativos y redes corporativas. Se basa en la detección y comparación de tráfico en base a las reglas definidas en su configuración.

```
sudo apt install snort -y
```

Al momento de la descarga se configura la interfaz de red donde va a escuchar el Snort para la captura y análisis de tráfico, para conocer la interfaz activa se ejecuta el comando ip a, la interfaz que se conecta a la red es enp0s3.

3.7.14 Configuración de reglas en el IDS/IPS

La regla que se define consta de una alerta que se activa cada vez que detecta un paquete que coincide con la configuración, definimos que detecte y bloquee el tráfico TCP, ICMP y UDP de cualquier maquina atacante con un mensaje que se visualizara en los logs al momento de activar el IDS/IPS, su objetivo principal detectar y registrar toda información relacionada con tráfico malicioso en la red para mitigar los ataques DoS, esto ayuda a la evaluación de patrones maliciosos que actúan dentro del sistema monitoreado.

- drop tcp any any -> any any (msg:"IPS: Bloqueado SYN Flood general"; detection_filter:track by_src, count 100, seconds 10;sid:10000002;rev:1;)
- drop udp any any -> any any (msg:"IPS: Bloqueado UDP Flood general";detection_filter:track by_src, count 200, seconds 10;sid:10000003;rev:1;)

- drop icmp any any -> any any (msg:"IPS: Bloqueado ICMP Flood general";detection_filter:track by_src, count 50, seconds 10;sid:10000004;rev:1;)

Cuando la red recibe ataques se activarán las reglas a través del drop estás son detectadas y bloqueadas solo en modo IPS inline. Para solo detectar se activan las siguientes reglas en el Snort.

- #alert icmp any any -> $HOME_NET any (msg:"ICMP Flood LOW"; itype:8; threshold:type both, track by_dst, count 100, seconds 1; sid:1000001; rev:1;)
- #alert icmp any any -> $HOME_NET any (msg:"ICMP Flood MEDIUM"; itype:8; threshold:type both, track by_dst, count 300, seconds 1; sid:1000002; rev:1;)
- #alert icmp any any -> $HOME_NET any (msg:"ICMP Flood HIGH"; itype:8; threshold:type both, track by_dst, count 1000, seconds 1; sid:1000003; rev:1;)
- #alert udp any any -> $HOME_NET any (msg:"UDP Flood LOW"; threshold:type both, track by_dst, count 500, seconds 1; sid:1000010; rev:1;)
- #alert udp any any -> $HOME_NET any (msg:"UDP Flood MEDIUM"; threshold:type both, track by_dst, count 1500, seconds 1; sid:1000011; rev:1;)
- #alert udp any any -> $HOME_NET any (msg:"UDP Flood HIGH"; threshold:type both, track by_dst, count 5000, seconds 1; sid:1000012; rev:1;)
- #alert tcp any any -> $HOME_NET any (msg:"TCP SYN Flood LOW"; flags:S; threshold:type both, track by_dst, count 100, seconds 1; sid:1000020; rev:1;)
- #alert tcp any any -> $HOME_NET any (msg:"TCP SYN Flood MEDIUM"; flags:S; threshold:type both, track by_dst, count 500, seconds 1; sid:1000021; rev:1;)
- alert tcp any any -> $HOME_NET any (msg:"TCP SYN Flood HIGH"; flags:S; threshold:type both, track by_dst, count 2000, seconds 1; sid:1000022; rev:1;)

3.7.15 Preparación del directorio donde se almacenan los registros

Para llevar a cabo una gestión adecuada y ordenada de los eventos producidos por el Snort, es necesario establecer un directorio que se lo llama snort-logs destinado únicamente a almacenar los archivos de los registros con sus respectivos permisos.

```
sudo mkdir -p /var/log/snort
sudo chown snort:snort /var/log/snort 2>/dev/null || true
sudo chmod 755 /var/log/snort
```

Estos comandos crean el directorio en donde se alojan los registros y alertas, otorgando todos los privilegios administrativos, asignando permisos al propietario y permisos de lectura y ejecución para el grupo.

3.7.16 Ejecuto Snort en modo de alerta rápida

Con la finalidad de comprobar el correcto funcionamiento del Snort se inicia en modo de alerta y bloqueo rápido de tráfico malicioso, de acuerdo con las reglas configuradas cada que exista coincidencias con el paquete se visualiza un mensaje en la consola, esto permite visualizar el tiempo real lo que sucede en la red.

Con el siguiente comando se inicia el snort en modo IPS (inline), para la adquisición de paquetes emplea un mecanismo NFQUEUE.

```
sudo snort -Q --daq nfq --daq-var queue=0 -c /etc/snort/snort.conf -A console
```

Luego de activar el Snort con el comando sudo tail -f /tmp/snort-logs/alert podemos observar en los logs el tipo de ataque que está recibiendo.

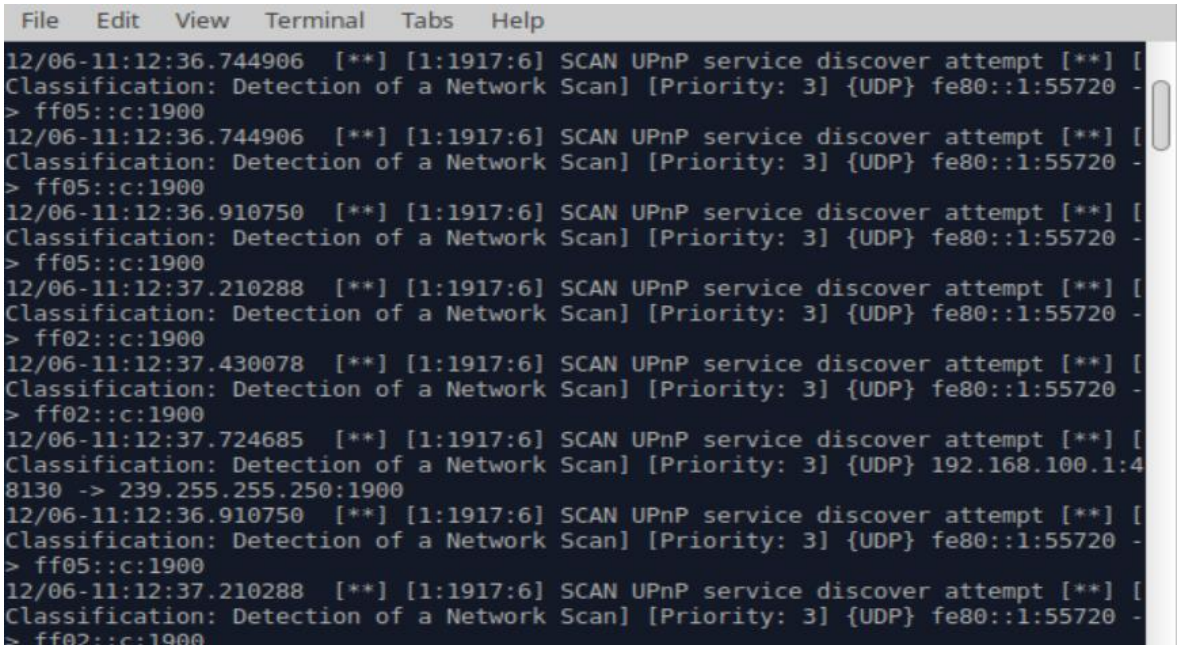


Figura 13. Monitoreo en tiempo real de alertas detectadas por Snort
Elaborado por: Andrés Cajas
Fuente: Simulación blockchain

3.7.17 Integración automática de eventos recibidos en Snort con blockchain

La combinación entre los dos sistemas Snort con tecnología blockchain asegura un registro seguro e inmutable de los datos obtenidos por el IDS/IPS, esto permite la capacidad de visualizar en tiempo real, bloquear y almacenar la información de forma segura por características propias de blockchain a través de cadenas de bloques frente a posibles alteraciones o manipulaciones de intrusos.

3.7.18 Creación de un directorio y archivo .js

Para realizar la integración es indispensable ejecutar un entorno de cliente, se crea un archivo llamado ipDetectadas.js que contienen configuraciones para procesar cada log que genera el Snort donde se extrae las direcciones Ip detectadas por el IDS/IPS, las mismas que serán enviadas a la blockchain para un registro inmutable filtrando solo Ips únicas, el script muestra en la consola la cantidad de direcciones detectadas facilitando una verificación rápida.

3.7.19 Instalación de dependencias

Necesita dependencias para una correcta integración entre los sistemas esto garantiza que los scripts cliente puedan cumplir su objetivo de interactuar con la red de forma eficiente y segura, se instala un módulo de fabric-network perteneciente a Node.js con la finalidad de que los clientes se conecten donde puedan realizar transacciones y consultas en el ledger.

```
npm install fabric-network
```

3.7.20 Consulta de eventos almacenados en blockchain

Una vez que alertas generadas por el IDS han sido almacenadas en la blockchain es ideal la implementación de mecanismos para poder observar y verificar los registros que se encuentran almacenados.

3.7.21 Ataques de denegación de servicios

Se realiza una simulación contralada de ciberataques con la finalidad de probar la robustez del sistema antes posibles amenazas, donde se evalúa el comportamiento, las reglas de bloqueo configuradas y rapidez de respuesta ante los ataques SYN flood donde se busca saturar la tabla de conexiones, estos ataques se realizan en máquinas virtuales para evitar impactos en sistemas externos teniendo presente las políticas de ética en seguridad informática.

3.7.22 Prueba de carga (Load Test) en recursos locales

Una prueba de estrés local que se realiza al propio sistema con la finalidad de evaluar el rendimiento solo consume recursos del equipo en el que se está ejecutando, genera carga a los núcleos del procesador con fines de benchmarking y diagnósticos con el comando.

```
sudo apt install stress -y
```

3.7.23 Ataques DoS utilizados

Tabla 4. Ataques DoS

Ataques DoS
sudo hping3 -S -p 80 –flood < ip>
sudo hping3 –udp –icmp –flood < ip>
sudo hping3 –udp -p 80 –flood < ip>

3.7.24 Esquema del sistema de ciberseguridad

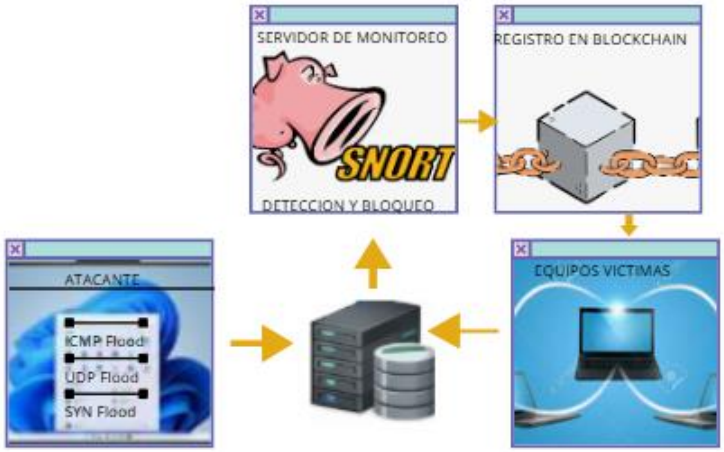


Figura 14. Sistema de ciberseguridad
Elaborado por: Andrés Cajas

El entorno que se utiliza para realizar las diferentes pruebas está constituido por tres equipos virtuales interconectados formando una red controlada, cada uno cumple con una función importante para el análisis y ejecución del sistema de seguridad informática, cuyo objetivo es llevar a cabo la simulación de ataques DOS, detección, bloqueo de tráfico y almacenamiento. La función de atacante cumple PC1 utilizando como sistema operativo Ubuntu, gracias a sus características de compatibilidad y herramientas sofisticadas usadas dentro de ciberseguridad.

PC2 contiene la configuración de Snort para detectar y prevenir ataques de DOS, cuenta con un registro inmutable para almacenar las ip de máquinas atacantes a través de blockchain.

PC3 actúa como equipo victima a quien va dirigido los ataques DOS, ya sea con sistema operativo de Ubuntu Linux o Windows

CAPÍTULO IV RESULTADOS Y DISCUSIÓN

.

4.1 Resultados

Se detalla los resultados obtenidos después de la programación y ejecución del sistema de ciberseguridad con tecnología blockchain, donde se integra un IDS/IPS para control y bloqueo de tráfico, la información obtenida es almacenada en un registro seguro, inmutable y descentralizado. En este registro se puede visualizar información detallada como día, mes, año, hora, segundos, el tipo de ataque y la dirección de la computadora atacante.

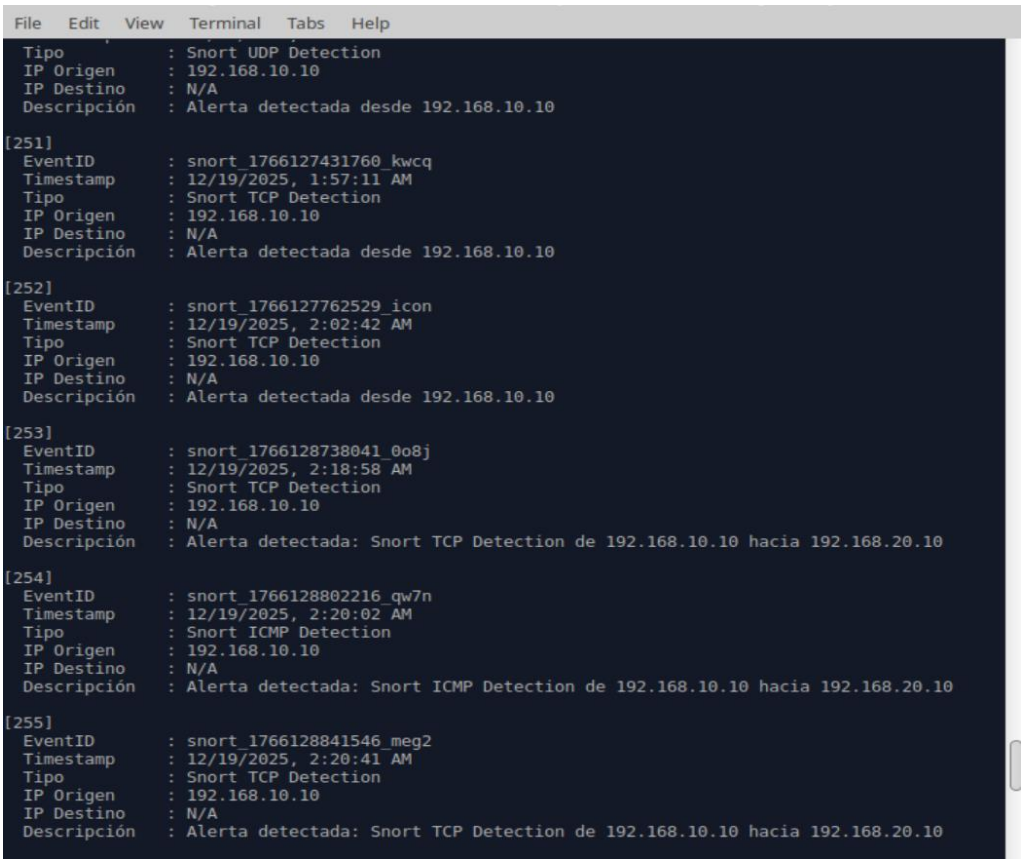


Figura 15. Almacenamiento en blockchain
Elaborado por: Andrés Cajas
Fuente: Simulación blockchain

Como etapa previa al análisis estadístico se realizaron pruebas de normalidad empleando pruebas de Kolmogorov-Smirnov y Shapiro-Wilk, los resultados obtenidos se presentan en la siguiente tabla.

Tabla 5. Pruebas de Normalidad de Tipo de Ataque

Variable	Tipo de Ataque	Shapiro-Wilk(W)	gl	Sig
Porcentaje de paquetes perdidos	ICMP Flood	0.907	60	0.000
	UDP Flood	0.951	60	0.017
	SYN Flood	0.936	60	0.003
Tiempo de ida	ICMP Flood	0.960	60	0.048
	UDP Flood	0.977	60	0.325
	SYN Flood	0.912	60	0.000
Disponibilidad del sistema	ICMP Flood	0.946	60	0.010
	UDP Flood	0.932	60	0.002
	SYN Flood	0.903	60	0.000
CPU libre	ICMP Flood	0.952	60	0.020
	UDP Flood	0.911	60	0.000
	SYN Flood	0.928	60	0.002
Uso de Memoria	ICMP Flood	0.938	60	0.004
	UDP Flood	0.876	60	0.000
	SYN Flood	0.968	60	0.112

4.1.1 Análisis Inferencial Global: Prueba de Kruskal-Wallis

Se utilizo pruebas de Kruskal-Wallis equivalente no paramétrico del ANOVA de un factor, para poder determinar si existen diferencias estadísticamente significativas entre los ataques ICMP Flood, UDP Flood y SYN Flood en las variables de rendimiento de red, esto se puede observar en la tabla de resultados de la prueba Kruskal-Wallis.

Tabla 6, Resultados de la prueba Kruska- Wallis

Variable	H de Kruskal-Wallis	gl	Sig. asintótica
Porcentaje de paquetes perdidos	147.438	2	<0.001
Tiempo de ida	159.116	2	<0.001
Disponibilidad del sistema	147.685	2	<0.001
CPU libre	155.094	2	<0.001
Uso de memoria	32.614	2	<0.001

Los resultados de la prueba Kruskal-Wallis muestro la existencia de diferencias estadísticamente significativas ($p < 0.001$) en todos los parámetros analizados que están expuestos a los tres tipos de ataques DoS. Los altos valores del estadístico H (rango: 32.614 - 159.116) indican que estas diferencias son sustanciales.

Dado que la prueba de Kruskal-Wallis es ómnibus y solo indica que existe al menos una diferencia entre los grupos, se llevó a cabo comparaciones post-hoc por pares a través de la prueba U de Mann-Whitney para conocer como los grupos difieren entre sí.

4.2 Comparaciones Post-Hoc: Prueba U de Mann-Whitney

Se efectuó una prueba estadística de U de Mann-Whitney con la finalidad de comparar el impacto de los ataques ICMP Flood vs UDP Flood, ICMP Flood vs SYN Flood, UDP Flood vs SYN Flood en parámetros de rendimiento de red, la prueba seleccionada debido a que el análisis realizado por Shapiro-Wilk donde los valores no cumplieron con el supuesto de normalidad. Tamaño de muestra: 60 observaciones por cada tipo de ataque (N total = 120) Nivel de significancia ajustado: $\alpha = 0.017$ (corrección de Bonferroni para 3 comparaciones).

4.2.1 Comparación 1: ICMP Flood vs UDP Flood

Tras completar el análisis estadístico, se procedió a calcular el efecto r mediante fórmulas correspondientes, los valores obtenidos se pueden observar a continuación.

Cálculo del tamaño del efecto (r)

Fórmula: $r = |Z| / \sqrt{120} = |Z| / 10.954$

Paquetes perdidos: $r = 9.448 / 10.954 = 0.862$ → Efecto GRANDE

Tiempo de ida: $r = 9.448 / 10.954 = 0.862$ → Efecto GRANDE

Disponibilidad: $r = 9.448 / 10.954 = 0.862$ → Efecto GRANDE

CPU libre: $r = 9.448 / 10.954 = 0.862$ → Efecto GRANDE

Uso de memoria: $r = 3.099 / 10.954 = 0.283$ → Efecto PEQUEÑO

Tabla 7. Resultados de la prueba U Mann-Whitney

Variable	**U de Mann-Whitney**	**Z**	**Sig.(bilateral)**	**r**
Porcentaje de paquetes perdidos	0.000	-9.448	< 0.001*	0.862
Tiempo de ida	0.000	-9.448	< 0.001*	0.862
Disponibilidad del sistema	0.000	-9.448	< 0.001*	0.862
CPU libre	0.000	-9.448	< 0.001*	0.862
Uso de memoria	1209.500	-3.099	0.002*	0.283

Porcentaje de paquetes perdidos: Se observaron diferencias estadísticamente significativas ($U = 0.000$, $p < 0.001$ con un impacto de gran magnitud ($r = 0.862$). El ataque UDP Flood es el que genera una mayor cantidad de pérdida de paquetes 2.84 veces más que el ataque ICMP Flood (21.66% vs 7.61%). Esta diferencia es altamente relevante desde el punto de vista práctico, indicando que UDP Flood tiene un mayor impacto en la confiabilidad de transmisión de datos.

Tiempo de ida (latencia): Se logro identificar diferencias altamente significativas ($U = 0.000$, $p < 0.001$, $r = 0.862$). El ataque UDP Flood produjo una latencia promedio de 74.76 ms, casi el triple que el ataque ICMP Flood (26.72 ms). Este incremento de aproximadamente 48 ms representa un deterioro severo en el tiempo de respuesta de la red, lo cual resulta más crítico para entornos que necesitan mínima latencia (VoIP, videoconferencias, aplicaciones en tiempo real).

Disponibilidad del sistema: Las diferencias fueron estadísticamente significativas ($U = 0.000$, $p < 0.001$, $r = 0.862$), pero con un patrón inverso al observado en las variables anteriores. ICMP Flood presenta 3.64 veces más la disponibilidad del sistema (12.74% vs 3.50%). El resultado obtenido dice que ante el ataque ICMP Flood se presenta una mayor disponibilidad ya que está expuesto a menos perdidas de paquetes y latencia, el ataque UDP presenta una menor disponibilidad sobrecarga debido al flujo de paquetes a nivel del sistema operativo.

CPU libre: Se observaron diferencias significativas ($U = 0.000$, $p < 0.001$, $r = 0.862$) en la utilización de recursos computacionales. UDP Flood tiene solo 9.99% de CPU disponible, mientras que el 21.53% corresponde al ataque ICMP Flood. UDP Flood consume aproximadamente 90% del procesador, mientras que ICMP Flood consume 78.5%, por lo tanto, el ataque más intenso y que afecta al sistema es UDP Flood.

Uso de memoria: Se logro mirar diferencias estadísticamente significativas ($U = 1209.500$, $p = 0.002$), el tamaño del efecto fue pequeño ($r = 0.283$). La diferencia práctica entre UDP Flood (28.02%) e ICMP Flood (27.47%) fue de apenas 0.55 puntos porcentuales, lo que indica que dicho parámetro no es un elemento determinante entre ambos tipos de ataque desde el punto de vista operacional.

4.2.2 Comparación 2: ICMP Flood vs SYN Flood

Tras completar el análisis estadístico, se procedió a calcular el efecto r mediante fórmulas correspondientes, los valores obtenidos se pueden observar a continuación.

Cálculo del tamaño del efecto (r)

Porcentaje de paquetes perdidos:

Z = -9.259

r = |−9.259| / 10.954

r = 9.259/ 10.954

r = 0.845 → Efecto GRANDE

2. Tiempo de ida (Latencia):

Z = -9.448

r = |−9.448| / 10.954

r = 9.448 / 10.954

r = 0.862 → Efecto GRANDE

3. Disponibilidad del sistema:

Z = -9.448

r = |−9.448| / 10.954

r = 9.448 / 10.954

r = 0.862 → Efecto GRANDE

4. CPU libre:

Z = -8.957

r = |−8.957| / 10.954

r = 8.957/ 10.954

r = 0.817 → Efecto GRANDE

5. Uso de memoria:

Z = -3.165

r = |−3.165| / 10.954

r = 3.165 / 10.954

r = 0.289 → Efecto PEQUEÑO

Tabla 8. Resultados de la Prueba U Mann-Whitney: ICMP Flood vs SYN Flood

Variable	U de Mann-Whitney	Z	Sig.(bilateral)	r
Porcentaje de paquetes perdidos	36.000	-9.259	< 0.001*	0.845
Tiempo de ida	0.000	-9.448	< 0.001*	0.862
Disponibilidad del sistema	0.000	-9.448	< 0.001*	0.862
CPU libre	93.500	-8.957	< 0.001*	0.817
Uso de memoria	1197.000	-3.165	0.002*	0.289

Porcentaje de Paquetes Perdidos: después de realizar el análisis se obtuvo que SYN Flood genera una mayor cantidad pérdida de paquetes en relación con ICMP Flood. En el análisis de rangos se observó que SYN Flood obtuvo un rango promedio de 89.90, mientras que ICMP Flood presentó 31.10, existe mayor pérdida de paquetes en SYN Flood. La diferencia estadísticamente significativa ($p<0.001$) con un tamaño del efecto grande ($r=0.845$) confirma que esta diferencia no es solo estadísticamente detectable, sino también es muy relevante desde el punto de vista práctico. SYN Flood generó 1.87 veces más pérdida de paquetes en relación con el ataque ICMP Flood, lo que representa un incremento del 87.5%. Este nivel de pérdida (14.27%) puede causar incremento en retransmisiones, afectación en calidad de servicio o en aplicaciones sensibles a la pérdida de datos como streaming y transferencia de archivos. No obstante, es necesario contextualizar que esta diferencia, aunque sustancial, es inferior a la observada entre UDP Flood e ICMP Flood (2.84 veces), situando al ataque SYN Flood como un ataque de intensidad moderada-alta respecto a la pérdida de paquetes.

Tiempo de Ida (Latencia): En el parámetro latencia SYN Flood presenta un impacto significativamente mayor, al realizar el análisis de rangos se pudo observar que SYN Flood genera un rango promedio de 90.50 frente a 30.50 de ICMP Flood, mostrando mayores niveles de valores de latencias. Los resultados de la prueba estadística evidenciaron que existe diferencias altamente significativas ($p<0.001$) con el tamaño del efecto más alto observado ($r=0.862$), lo que indica una diferencia de mayor relevancia en la parte práctica.

La latencia de SYN Flood es 1.88 veces superior a ICMP Flood, con un aumento de 23.53 ms que representa un incremento del 88%. Una latencia de 50.25 ms, aunque inferior puede llegar afectar al umbral crítico para videoconferencias (150 ms), puede tener afectaciones en el rendimiento de aplicaciones de gaming online, servicios de VoIP y aplicaciones de trading

de alta frecuencia donde cada milisegundo es crítico. Se puede evidenciar que la desviación estándar es muy baja de SYN Flood (1.15 ms) sugiere un patrón de ataque estable y predecible, posiblemente debido a la naturaleza consistente del proceso de handshake TCP incompleto. El incremento de latencia, aunque considerable, fue aproximadamente la mitad del causado por UDP Flood (48 ms), mostrando que el ataque SYN Flood presenta un impacto medio-alto en esta métrica.

Disponibilidad del Sistema: ICMP Flood es el que produjo un impacto más significativo. ICMP Flood obtuvo un rango promedio de 90.50, a diferencia de ello SYN Flood que tuvo 30.50, indicando valores consistentemente más altos de afectación para ICMP Flood. Esta diferencia fue estadísticamente significativa ($p<0.001$) con un tamaño del efecto grande ($r=0.862$). El ataque ICMP Flood presenta una disponibilidad reducida del sistema de 2.43 veces más que SYN Flood (12.74% vs 5.25%), existiendo un incremento del 143% en la afectación. ICMP Flood ejerce una afectación severa en la capacidad del sistema para mantener servicios activos.

Una afectación del 12.74% representa una pérdida crítica en escenarios que requieren alta disponibilidad (99.9% de uptime), mientras que el 5.25% causado por SYN Flood, aunque moderado, es manejable con mecanismos de mitigación apropiados. El impacto causado por el ataque ICMP en la disponibilidad puede ser provocado debido a la sobrecarga del stack de red a nivel de kernel al procesar solicitudes ICMP, el agotamiento de buffers para respuestas ICMP ECHO, el consumo intensivo de recursos del sistema operativo para generar respuestas, y la interrupción de procesos críticos debido al manejo prioritario de paquetes ICMP. Los resultados indican que ICMP Flood afectara la continuidad operacional del servicio.

CPU: el ataque SYN Flood consume significativamente más recursos de procesamiento que ICMP Flood. Es importante tener en cuenta para el análisis de este parámetro ya que es una interpretación inversa: valores más altos indican mejor rendimiento (menor consumo). El ataque ICMP Flood presentó un rango promedio de 88.94, mientras que SYN Flood obtuvo 32.06, indicando que ICMP Flood deja consistentemente más CPU disponible. Esta diferencia fue estadísticamente significativa ($p<0.001$) con un tamaño del efecto grande ($r=0.817$). Los resultados indican que ICMP Flood deja 21.53% de CPU libre (consumiendo

aproximadamente 78.5%), mientras que SYN Flood deja solo 18.02% libre (consumiendo aproximadamente 82%). Aunque la diferencia absoluta de 3.5 puntos porcentuales puede parecer modesta, representa un incremento del 4.46% en consumo relativo, lo cual es significativo en un contexto de ataque DoS donde cada punto porcentual de CPU es crítico.

Con un consumo del 82%, SYN Flood deja escaso margen para otras operaciones del sistema, pudiendo causar lentitud generalizada, tiempos de respuesta prolongados, dificultad para ejecutar tareas administrativas y posible incapacidad para procesar tráfico legítimo. El alto consumo de CPU de SYN Flood se explica por varios factores técnicos inherentes a este tipo de ataque. Primero, el mantenimiento de la tabla de conexiones semi-abiertas, donde cada SYN no completado ocupa una entrada que el sistema debe gestionar. Segundo, las verificaciones continuas que el kernel realiza constantemente sobre el estado de las conexiones y la limpieza periódica de conexiones expiradas. Tercero, el procesamiento intensivo de respuestas SYN-ACK para cada SYN recibido. En un análisis comparativo, SYN Flood se posiciona entre ICMP Flood (78.5% de consumo) y UDP Flood (90% de consumo), afirmando su perfil como un ataque de intensidad media-alta en consumo de procesador.

Uso de Memoria: Se encontraron diferencias estadísticamente significativas ($p=0.002<0.017$), el tamaño del efecto fue pequeño ($r=0.289$), indicando que la diferencia carece de relevancia práctica. ICMP Flood consumió 27.47% de memoria, ante SYN Flood que consumió 27.03%, una diferencia de apenas 0.44 puntos porcentuales que representa solo un 1.6% de diferencia relativa. La interpretación del tamaño del efecto pequeño es crucial para entender la relevancia práctica de estos resultados. Según Cohen (1988), un valor de $r<0.3$ indica un efecto pequeño, es decir, una diferencia estadísticamente detectable, pero con poca relevancia práctica. Con $r=0.289$, la diferencia se encuentra en el límite superior del rango pequeño, lo que significa que, aunque SPSS detectó la diferencia gracias al tamaño de muestra de 120 observaciones, en la práctica ambos ataques consumen cantidades virtualmente idénticas de memoria. En términos concretos, en un sistema con 16 GB de RAM, la diferencia de 0.44% representa aproximadamente 70 MB, una cantidad prácticamente irrelevante en un entorno de red bajo ataque DoS, especialmente considerando que los buffers de red suelen ser de pocos megabytes.

Este hallazgo es consistente con lo observado en la comparación ICMP vs UDP (r=0.283), confirmando un patrón: el uso de memoria no es una métrica distintiva entre los tres tipos de ataque DoS. Ambos ataques generan tráfico de red similar en volumen, requieren buffers de recepción comparables, mantienen estructuras de datos similares y no almacenan payloads grandes en memoria. Aunque SYN Flood consume ligeramente menos memoria porque los paquetes SYN son más pequeños que ICMP ECHO y las entradas en la tabla de conexiones son menores que los buffers ICMP, estas diferencias son mínimas en la práctica. Por tanto, el uso de memoria no debe considerarse un factor determinante para entre estos tipos de ataque para fines de detección, clasificación o mitigación.

4.2.3 Comparación 3: UDP Flood vs SYN Flood

Tras completar el análisis estadístico, se procedió a calcular el efecto r mediante fórmulas correspondientes, los valores obtenidos se pueden observar a continuación.

porcentaje de paquetes perdidos $r = |Z| / \sqrt{N}$

$r = |-8.175| / 10.954$

$r = 8.175 / 10.954 \rightarrow$ Bajo

$r = 0.746$ Tiempo de ida (Latencia): $r = |Z| / \sqrt{N}$

$r = |-9.448| / 10.954$

$r = 9.448 / 10.954 \rightarrow$ Bajo

$r = 0.862$ disponibilidad del sistema $r = |Z| / \sqrt{N}$

$r = |-7.975| / 10.954$

$r = 7.975 / 10.954 \rightarrow$Alto

$r = 0.728$ CPU libre: $r = |Z| / \sqrt{N}$

$r = |-9.448| / 10.954$

$r = 9.448 / 10.954 \rightarrow$Alto

$r = 0.862$ Uso de memoria:$r = |Z| / \sqrt{N}$

$r = |-5.422| / 10.954$

$r = 5.422 / 10.954$

$r = 0.495 \rightarrow$ Bajo

Tabla 9. Resultados de la Prueba U de Mann-Whitney: UDP Flood vs SYN Flood

Variable	U de Mann-Whitney	Z	Sig.(bilateral)	r
Porcentaje de paquetes perdidos	242.500	-8.175	< 0.001*	0.746
Tiempo de ida	0.000	-9.448	< 0.001*	0.862
Disponibilidad del sistema	280.500	-7.975	< 0.001*	0.728
CPU libre	93.500	-9.448	< 0.001*	0.862
Uso de memoria	767.000	-5.422	< 0.001*	0.495

Porcentaje de Paquetes Perdidos: Existe una mayor pérdida de paquetes en el ataque UDP Flood. El rango promedio de UDP Flood es de 86.46, mientras que SYN Flood presentó un rango promedio 34.54, la diferencia estadísticamente significativa ($p<0.001$) con un tamaño del efecto grande ($r=0.746$) confirma que esta diferencia es altamente relevante desde el punto de vista práctico. UDP Flood presento 1.52 veces más pérdida de paquetes que SYN Flood (21.66% vs 14.27%), con un incremento existente de 51.8%. Este nivel de pérdida de 21.66% es crítico, alcanzando más de un quinto de todos los paquetes transmitidos, esto representa una degradación severa del servicio, retransmisiones masivas y en muchos casos hacer prácticamente inutilizable la red para aplicaciones sensibles. En contraste, aunque SYN Flood con 14.27% de pérdida también representa un nivel alto de degradación, tiene un impacto mucho menor. Por lo tanto, UDP Flood es el ataque más intenso de los tres tipos evaluados en términos de integridad de transmisión de datos, respaldando su caracterización como un ataque enfocado a la sobrecarga masiva de ancho de banda a través del envío indiscriminado de datagramas UDP.

Tiempo de Ida (Latencia): UDP Flood presentó un rango que se encuentra en un promedio de 90.50 frente a 30.50 de SYN Flood, reflejando valores consistentemente más elevados de latencia. La prueba estadística confirmó diferencias altamente significativas ($p<0.001$) con el tamaño del efecto más alto observado ($r=0.862$. UDP Flood produjo una latencia 1.49 veces más alta a SYN Flood (74.76 ms vs 50.25 ms), con un incremento de 24.51 ms que representa un aumento del 48.8%. Una latencia de 74.76 ms se encuentra peligrosamente cerca al umbral crítico para gran cantidad de aplicaciones en tiempo real. A pesar de que está por debajo del límite aceptable para videoconferencias (150 ms), esta latencia genera un deterioro sustancial en el gaming online (donde el umbral óptimo es inferior a 50 ms), deteriora significativamente los servicios de VoIP, y prácticamente imposibilita aplicaciones

de trading de alta frecuencia o control industrial en tiempo real donde cada milisegundo es crítico. La latencia generada por el ataque SYN Flood (50.25 ms), aunque también elevada, se mantiene en un nivel que permite el funcionamiento, aunque afectado, de la mayoría de las aplicaciones. La diferencia de 24.51 ms entre ambos ataques, con un tamaño del efecto de 0.862, representa la mayor diferencia en términos de relevancia práctica observada entre todos los pares de ataques evaluados en esta variable, UDP Flood es el ataque que más afecta al rendimiento temporal de la red.

Disponibilidad del Sistema: Es una de las pocas variables donde SYN Flood mostró una mayor afectación en relación con UDP Flood. Se pudo observar que SYN Flood obtuvo un rango promedio de 85.83, mientras que UDP Flood presentó 35.18, indicando valores consistentemente más altos de afectación para SYN Flood. Esta diferencia fue estadísticamente significativa ($p<0.001$) con un tamaño del efecto grande ($r=0.728$), confirmando su alta relevancia práctica. SYN Flood presenta una afectación a la disponibilidad del sistema de 1.50 veces más que UDP Flood (5.25% vs 3.50%), lo que indica un incremento del 50% en la afectación. Aunque UDP Flood es el ataque más intenso en términos de saturación de red (pérdida de paquetes y latencia), su impacto en la disponibilidad del sistema es relativamente menor. Estos resultados evidencian una congestión masiva de ancho de banda, característica de UDP Flood, no se traduce necesariamente en una reducción proporcional de la disponibilidad del servicio.

Por el contrario, SYN Flood, con su mecanismo específico de agotamiento de la tabla de estados TCP mediante conexiones semi-abiertas, impone una carga más directa sobre la capacidad del sistema para sostener continuidad en los servicios, aunque genere menor degradación en las métricas de red. Esto es debido a las características del ataque: UDP Flood satura el ancho de banda, aunque o implica una afectación severa al consumo de recursos del sistema operativo, mientras que SYN Flood, al mantener múltiples conexiones en estados de espera, consume recursos del kernel y reduce la capacidad del sistema para aceptar nuevas conexiones legítimas. Es importante contextualizar que, aunque SYN Flood muestra mayor impacto que UDP Flood en disponibilidad, ambos ataques presentan valores significativamente menores que ICMP Flood (12.74%).

CPU Libre: En este parámetro el ataque UDP Flood consume más recursos de procesamiento que SYN Flood. Es importante considerar mientras más CPU libre = mejor rendimiento, SYN Flood presentó un rango promedio de 90.50, mientras que UDP Flood obtuvo 30.50, indicando que UDP Flood afecta más al CPU. Esta diferencia fue estadísticamente significativa ($p<0.001$) con el tamaño del efecto más alto posible ($r=0.862$), confirmando su máxima relevancia práctica. Los resultados indican que UDP Flood deja apenas 9.99% de CPU libre, consume el 99% del procesador, mientras que SYN Flood deja 18.02% libre, consumiendo aproximadamente el 82%. La diferencia de 8.03 puntos porcentuales representa un incremento relativo del 80.4% en el consumo de CPU por parte de UDP Flood respecto a SYN Flood.

Los resultados obtenidos son críticos con solo 10% de CPU disponible, el sistema está por colapsar, casi con capacidad nula para procesar el tráfico legítimo y mantener un correcto funcionamiento. Al estar expuesto a este ataque el sistema suele ser lento causando la imposibilidad de implementar contramedidas o incluso de acceder remotamente al sistema para bloquear el ataque. La alta cantidad de consumo de CPU causado por UDP Flood puede ser por: primero, el procesamiento de una gran cantidad de datagramas UDP entrantes, donde cada paquete debe cumplir con ser validado, enrutado y procesado por el stack de red. Segundo, no existe una correcta generación de respuestas ICMP "port unreachable" para puertos cerrados, lo cual tiene un mayor consumo del kernel. Tercero, la presencia de gran cantidad de tráfico y la constante actualización de tablas de enrutamiento. Cuarto, los intentos del sistema por mantener buffers y colas de procesamiento ante la saturación. Aunque SYN Flood también consume recursos significativos (82%), su consumo está más enfocado en el mantenimiento de la tabla de estados TCP, lo cual, aunque agresivo, UDP es el que mayor afectación genera en términos de agotamiento de recursos computacionales, considerando como uno de los ataques más intensos.

Uso de Memoria: Los resultados evidenciaron diferencias estadísticamente significativas ($p<0.001$) con un tamaño del efecto mediano ($r=0.495$), indicando que la diferencia tiene relevancia práctica moderada. El ataque UDP Flood consumió 28.02% de memoria, mientras que el ataque SYN Flood consumió 27.03%, una diferencia de 0.99 puntos porcentuales que representa un 3.7% de diferencia relativa. Esto contrasta con las comparaciones previas

(ICMP vs UDP: r=0.283; ICMP vs SYN: r=0.289), donde el uso de memoria mostró tamaños del efecto pequeños. Con r=0.495, el efecto se encuentra en el rango mediano.

Según Cohen (1988), lo cual indica que, aunque la diferencia absoluta sigue siendo relativamente pequeña (0.99 puntos porcentuales), es más pronunciada y prácticamente relevante que en las otras comparaciones. UDP Flood, al producir gran cantidad de datagramas, requiere buffers de recepción más grandes y mantiene más datos en tránsito en memoria. Adicionalmente, el sistema debe mantener estructuras de datos para tracking de conexiones UDP, tablas de enrutamiento más extensas por la llegada de un alto volumen de tráfico, y estadísticas de red más complejas. Por otro lado, SYN Flood, aunque mantiene entradas en la tabla de estados TCP, estas estructuras son relativamente compactas y el consumo de memoria está más controlado. Los parámetros como pérdida de paquetes, latencia y consumo de CPU muestran diferencias más considerables y con mayor impacto operativo.

4.3 Discusión

4.3.1 ICMP Flood vs UDP Flood

La comparación entre ICMP Flood y UDP Flood evidenció diferencias estadísticamente significativas en todas las variables evaluadas ($p < 0.001$), con tamaños del efecto grandes ($r = 0.862$) en cuatro de los 5 parámetros analizados. Los resultados evidencian diferencias considerables de relevancia práctica y operacional entre los dos tipos de ataque.

En algunos parámetros UDP mostro ser el que el mayor daño causaba, en pérdida de paquetes, generó 2.84 veces más afectación (21.66% vs 7.61%), alcanzando niveles que afectan significativamente la integridad de la transmisión de datos. Este porcentaje implica retransmisiones frecuentes y una afectación considerable en el servicio, especialmente en aplicaciones sensibles a la pérdida de paquetes.

En latencia, el ataque UDP Flood genero un retardo promedio de 74.76 ms frente a 26.72 ms producido por ICMP Flood, esto suele tener afectaciones en aplicaciones que son en tiempo real y servicios interactivos donde la latencia óptima es inferior a 50 ms.

ICMP Flood tuvo un impacto mayor en la disponibilidad del sistema, afectándola 3.64 veces más que UDP Flood (12.74% vs 3.50%). Este resultado sugiere que, aunque ICMP Flood genera menor saturación de red, compromete de manera más directa la continuidad del servicio, posiblemente debido al incremento de la carga en el procesamiento de solicitudes ICMP dentro del sistema operativo.

En consumo de CPU, UDP Flood mantiene disponible solo el 9.99% del procesador, mientras que ICMP Flood presento el 21.53%. En consecuencia, UDP Flood tiene un aproximado de consumo de CPU del 90%, esto afecta al sistema de una manera crítica involucrando la estabilidad y capacidad de respuesta ante tráfico legítimo o mecanismos de mitigación.

En cuanto al uso de memoria, la variación fue estadísticamente significativa, el tamaño del efecto pequeño ($r = 0.283$) señala que esta variable no constituye un factor que marque mucha diferencia entre ambos ataques.

En conjunto, la evidencia de los resultados permite concluir que UDP Flood es el ataque que afecta más en saturación de red y consumo de recursos de procesamiento del sistema, por su parte ICMP Flood presenta una amenaza más crítica a la disponibilidad.

4.3.2 ICMP Flood y SYN Flood

La comparación entre los ataques ICMP Flood y SYN Flood mostro comportamientos distintivos en los dos ataques. Todos los parámetros analizados revelaron diferencias estadísticamente significativas ($p<0.017$), aunque con relevancia práctica variable. Cuatro parámetros presentaron tamaños del efecto grandes ($r>0.8$), indicando diferencias altamente relevantes desde el punto de vista operacional, mientras que una variable (uso de memoria) mostró un efecto no muy relevante en la práctica.

SYN Flood demostró ser el que tiene mayor afectación en relación con ICMP Flood en tres parámetros de rendimiento de red. En pérdida de paquetes, SYN Flood generó 1.87 veces más pérdida con un efecto grande ($r=0.845$). En latencia, se produjo 1.88 veces más retardo

con el tamaño del efecto más alto observado (r=0.862). En consumo de CPU, SYN Flood demandó 3.5 puntos porcentuales adicionales con un efecto grande (r=0.817). SYN Flood es un ataque orientado al consumo de recursos de conexión y procesamiento, que explota el handshake TCP de tres vías dejando conexiones semi-abiertas, generando una afectación moderada-alta en las métricas de red y posicionándose con una severidad media-alta, intermedia entre ICMP Flood y UDP Flood.

Por el contrario, ICMP Flood tiene un mayor impacto en una variable como es la disponibilidad del sistema. ICMP Flood afectó la disponibilidad 2.43 veces más que SYN Flood con un efecto grande (r=0.862), confirmando el patrón presentado en comparaciones anteriores. Por ende, ICMP Flood es un ataque orientado específicamente a afectar la disponibilidad del servicio mediante la saturación con solicitudes ICMP ECHO, buscando causar sobrecarga en el stack de red a nivel de kernel. Aunque ICMP Flood genera menos pérdida de paquetes y latencia que SYN Flood, representa la mayor amenaza para el correcto funcionamiento operacional y el uptime del sistema entre los tres tipos de ataques DoS.

Los resultados obtenidos aportan lineamientos para el diseño de estrategias de defensa, para bloquear SYN Flood, la prioridad debe centrarse en salvaguardar la tabla de estados TCP por medio de técnicas como SYN cookies, reducción de timeouts de conexiones semi-abiertas, rate limiting por IP de origen y filtrado de paquetes con flags anormales. Para ICMP Flood, se debe enfocar en mantener la disponibilidad del servicio por medio de la limitación de respuestas ICMP, rate limiting de ICMP ECHO, priorización de tráfico no-ICMP y bloqueo de ICMP innecesario en el perímetro de red. Las diferencias en los perfiles de ataque evidencian la necesidad de diseñar e implementar estrategias de defensa específicas y adecuarlas a las diferentes amenazas. SYN Flood se posiciona como un ataque de intensidad media-alta se centra en afectar métricas de rendimiento de red y el consumo de recursos computacionales, mientras que ICMP Flood tiene una mayor afectación a la disponibilidad del sistema. El uso de memoria n representa un factor distintivo entre los dos ataques, estos resultados contribuyen con un fundamento empírico para llevar a cabo el desarrollo de mecanismos de detección y mitigación.

4.3.3 UDP Flood vs SYN Flood

Al realizar el análisis entre los ataques UDP Flood y SYN Flood se obtuvo que UDP Flood es más agresivo en relación con los otros ataques que se estudiaron. Los parámetros analizados mostraron diferencias estadísticamente significativas ($p<0.001$), y lo más relevante es que cuatro de las cinco tienen tamaños altos ($r>0.7$), con dos de ellas llegando al máximo valor ($r=0.862$), mostrado diferencias de máxima relevancia práctica y operacional.

UDP Flood resulto ser tener un alto nivel perjudicial en comparación con SYN Flood en cuatro variables críticas. En pérdida de paquetes, UDP Flood generó 1.52 veces más pérdida con un efecto grande ($r=0.746$), llegando alcanzar niveles del 21.66% que afectan a gran escala la integridad de la transmisión. En latencia, produjo 1.49 veces más retardo con el efecto más grande posible ($r=0.862$), llegando alcanzar 74.76 ms que se aproximan peligrosamente a umbrales críticos para mantenerse funcionando correctamente aplicaciones en tiempo real. En consumo de CPU, UDP Flood demandó 8.03 puntos porcentuales adicionales con el efecto máximo ($r=0.862$), teniendo un consumo del 90% del procesador y dejando el sistema cerca de colapsar. En uso de memoria, UDP Flood consumió 0.99 puntos porcentuales más con un efecto mediano ($r=0.495$), representando en este parámetro una diferencia mayor entre todos los pares de ataques evaluados.

Por el contrario, se pudo observar que el ataque SYN Flood tiene un impacto significativamente alto en una variable específica: la disponibilidad del sistema. SYN Flood afectó la disponibilidad 1.50 veces más que UDP Flood con un efecto grande ($r=0.728$). Aunque UDP Flood es el ataque que implica más consumo en términos de saturación de red y recursos computacionales, en disponibilidad del sistema tiene un impacto relativamente menor. Esto se debe a que presenta una mayor saturación de ancho de banda característica de UDP Flood, SYN Flood produce un mayor agotamiento de la tabla de estados TCP a través de conexiones semi-abiertas ejerce una presión más directa sobre la capacidad del sistema para mantener servicios funcionando correctamente.

Los resultados obtenidos muestran que el ataque UDP Flood es el más destructivo y agresivo de los tres tipos evaluados. Por tener mayor saturación en ancho de banda mediante el flujo

de envío de datagramas UDP a múltiples puertos, produciendo una gran perdida en paquetes, latencia y consumo de CPU. Con un 90% de consumo de procesador, UDP Flood deja el sistema al borde de ser saturado. Mientras que SYN Flood se mantiene como un ataque de intensidad media-alta, orientado específicamente al consumo de recursos de conexión mediante la explotación del handshake TCP de tres vías, con impacto moderado en red, pero más pronunciado en disponibilidad que UDP Flood.

Las implicaciones para el diseño de estrategias de defensa son directas y críticas. Para tener un bloqueo de UDP Flood, como prioridad se debe considerar la protección del ancho de banda y recursos de procesamiento mediante mecanismos como rate limiting agresivo, filtrado de tráfico UDP no esencial, implementación de sistemas de scrubbing en el perímetro de red, y sobre-aprovisionamiento de capacidad de procesamiento y ancho de banda. Para el ataque de tipo SYN Flood, las estrategias previamente mencionadas (SYN cookies, reducción de timeouts, rate limiting) siguen siendo una opción válida, los resultados muestran una amenaza mucho menor a la producida por UDP Flood en casi todas las métricas. Ante esto los sistemas se ven en la necesidad de ser capaces de clasificar el tipo de ataque y priorizar las respuestas defensivas según el nivel de amenaza existente.

El ataque UDP Flood es considerado como el ataque DoS más severo en relación con los otros tipos que han sido evaluados, SYN Flood representa una amenaza seria, particularmente para la disponibilidad del sistema, este ataque es menos agresivo que UDP Flood. Estos resultados muestran una evidencia empírica robusta para lograr establecer jerarquías de prioridad en estrategias para mitigar ataques DoS, donde UDP Flood se tiene que considerar como la amenaza más crítica.

CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

La tecnología blockchain presenta bases teóricas adecuadas y acertadas para fortalecer el conocimiento de la seguridad de la información. Mediante el análisis de conceptos como la descentralización, inmutabilidad, trazabilidad y consenso, se evidencia que blockchain no se limita al ámbito financiero, esta presenta un alto potencial para resolver problemas relacionados con la integridad y confiabilidad de los datos en entornos digitales. La utilización del método analítico permitió comprender de forma progresiva de cómo estas características se pueden integrar en sistemas de ciberseguridad, que a través de una fundamentación teórica que mejora los conocimientos teóricos necesarios para el diseño de soluciones tecnológicas seguras y fiables.

El sistema diseñado cumple adecuadamente con la finalidad de garantizar la integridad de la información. La implementación del blockchain privada basada en Hyperledger Fabric ofrece una infraestructura distribuida capaza de almacenar registros de eventos de seguridad de forma inmutable y verificable, este diseño ha demostrado ser funcional, escalable y adaptable, puesto que integra componentes reales como contratos inteligentes, certificados digitales y nodos distribuidos, facilitando su futura implementación en entornos reales, sean estos entornos educativos o corporativos donde la protección de la información es un aspecto crítico.

Se logro desarrollar la implementación de un prototipo funcional de un sistema de ciberseguridad capaz de detectar y mitigar tipos de ataques ICMP Flood, UDP Flood y SYN Flood, los resultados obtenidos demuestran la operatividad del sistema, el prototipo responde de manera adecuada ante los diferentes escenarios de pruebas de ataques, demostrando que el sistema puede responder a tiempo frente a intentos de saturar recursos provocados por el tráfico malicioso. Mientras que el uso de máquinas virtuales garantizo la ejecución de las pruebas siguiendo los principios éticos de la seguridad informática, evitando posibles impactos en redes externas y asegurando la fiabilidad de los resultados obtenidos.

Se llevo a cabo pruebas que permitieron la evaluación del desempeño del sistema propuesto, la integración del sistema IDS/IPS Snort mediante la tecnología blockchain tuvo resultados

eficientes, puesto que el sistema pudo detectar, bloquear y registrar en tiempo real aquellos eventos de seguridad que se generaron por los ataques simulados, almacenando información importante como las direcciones IP, la fecha y hora y tipo de ataque de forma segura e inalterable, de igual manera se pudo evaluar como son afectadas las máquinas virtuales victimas a través de mediciones de parámetros de rendimiento del sistema.

5.2 Recomendaciones

Es recomendable implementar el sistema de ciberseguridad basado en blockchain en entornos reales como en instituciones educativas, redes corporativas pequeñas, con la finalidad de evaluar su desempeño en condiciones reales de tráfico de datos. Esto permitirá validar la escalabilidad, estabilidad y eficiencia del sistema fuera de un entorno simulado, además de identificar posibles mejoras en la configuración de la red blockchain, los contratos inteligentes y la integración con herramientas de detección de intrusos y proteger los datos almacenados.

Se recomienda ampliar las pruebas de seguridad con la incorporación de nuevos tipos de ataques, sean estos internos o externos, donde se incluya ataques distribuidos más complejos, técnicas de evasión y escenarios de múltiples ataques simultáneos. Esta ampliación permite evaluar de forma completa la capacidad de respuesta del sistema ante amenazas avanzadas y de esta forma optimizar las reglas del IDS/IPS y los mecanismos de registro en blockchain, con el objetivo de mejorar la solidez del sistema y su capacidad de adaptación a escenarios reales de ciberataques en constante evolución.

Además, se recomienda prestar atención a las versiones de las herramientas informáticas que se utilizan durante la configuración del sistema, ya que con versiones que no son compatibles suelen existir errores, inconsistencias evitando llevar a cabo un correcto diseño del sistema.

BIBLIOGRÁFIA

[1] Ministerio de Telecomunicaciones y de la Sociedad de la Información, «Diagnóstico de las capacidades de Ciberseguridad del Ecuador», p. 104, 2022, Accedido: 6 de diciembre de 2025. [En línea]. Disponible en: https://www.telecomunicaciones.gob.ec/wp-content/uploads/2023/05/DIAGNO%CC%81STICO-DE-LAS-CAPACIDADES-DE-CIBERSEGURIDAD-Ecuador-Diciembre-2022_compressed.pdf?utm_source=chatgpt.com

[2] A. G. Antoniucci, M. R. S. Sierra, J. Báez, y M. Crisafulli, «Blockchain y la seguridad de la información en América y Europa», *Informática y Derecho. Revista Iberoamericana de Derecho Informático (2.ª época)*, vol. 1, n.º 15, pp. 137-153, jul. 2024, Accedido: 6 de diciembre de 2025. [En línea]. Disponible en: https://revistas.fcu.edu.uy/index.php/informaticayderecho/article/view/4740

[3] M. Guamán, «Hyperledger Blockchain para la seguridad en bases de datos un mapeo sistemático», Universidad Politécncia Salesiana de Guayaquil, Guayaquil, 2022. [En línea]. Disponible en: https://dspace.ups.edu.ec/bitstream/123456789/20320/1/UPS-GT003221.pdf

[4] S. G. del Campillo, «Ciberseguridad y Blockchain», *Revista Blockchain e Inteligencia Artificial*, vol. 2, n.º 3, 2021, doi: 10.22529/rbia.2021(3)05.

[5] Y. Lucio, D. Siler, y K. Márceles, «Arquitectura de un Framework de ciberseguridad inteligente basado en tecnología Blockchain para IIoT», *Ingeniería y competitividad: revista científica y tecnológica*, vol. 24, n.º 2, pp. 1-13, 2022, Accedido: 16 de diciembre de 2025. [En línea]. Disponible en: https://dialnet.unirioja.es/servlet/articulo?codigo=8637554

[6] N. Salgado, «La tecnología Blockchain y su potencial para revolucionar la gestión de datos y la seguridad de las transacciones», vol. 8, n.º 2, pp. 546-562, 2023, doi: https://doi.org/10.23857/fipcaec.v8i2.

[7] AON, «El ciberriesgo es un riesgo corporativo – América Latina responde – Aon Cyber 2025 Español (LATAM)». Accedido: 17 de diciembre de 2025. [En línea]. Disponible en: https://www.aon.com/cyber-risk-report/es-la/cyber-risk-is-a-corporate-risk-latin-america-responds?utm_source=chatgpt.com

[8] B. Saavedra, «Ciberseguridad en América Latina: Retos, preocupaciones y oportunidades», p. 27, 2023, [En línea]. Disponible en: https://ceeep.mil.pe/wp-content/uploads/2023/03/ciber-seguridad-america-latina.pdf

[9] J. Rodríguez, K. Rivas, y A. Mendoza, «Blockchain para Mejorar la Seguridad y el Acceso Confiable a Datos: Aplicaciones en la Protección de Información», *REVISTA MULTIDISCIPLINARIA DE DESARROLLO AGROPECUARIO, TECNOLÓGICO, EMPRESARIAL Y HUMANISTA.*, vol. 7, n.º 2, pp. 10-10, dic. 2025, Accedido: 17 de diciembre de 2025. [En línea]. Disponible en: https://investigacion.utc.edu.ec/index.php/dateh/article/view/1208

[10] Zaintza, «Seguridad informática», Cyber Zaintza. Accedido: 6 de diciembre de 2025. [En línea]. Disponible en: https://ciberseguridad.euskadi.eus/ciberglosario/seguridad-informatica/webcyb00-contcibglos/es/?utm_source=chatgpt.com

[11] M. Beschokov, «Tríada de la CIA: definición y ejemplos», Wallarm. Accedido: 6 de diciembre de 2025. [En línea]. Disponible en: https://lab.wallarm.com/what/definicion-

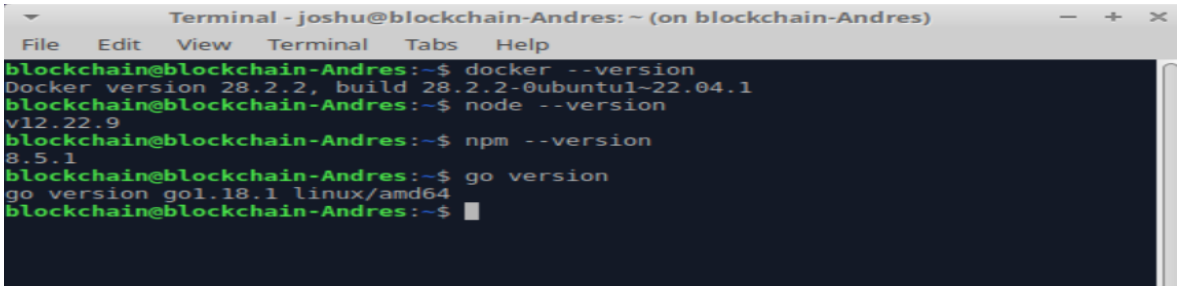
de-la-triada-de-la-cia-ejemplos-de-confidencialidad-integridad-y-disponibilidad/?lang=es
[12] J. T. L. López, J. Aguirre, y P. Navarrete, «Ciberseguridad para todos: orígenes, concepto y alcance», *ININEE CIENCIA*, vol. 2, n.º 4, pp. 39-48, 2024, doi: 10.33110/inineeciencia.v2i4.48.
[13] Kaspersky, «América Latina enfrenta más de 3.1 millones de ataques de malware por día, alerta Kaspersky», /. Accedido: 6 de diciembre de 2025. [En línea]. Disponible en: https://latam.kaspersky.com/about/press-releases/america-latina-enfrenta-mas-de-31-millones-de-ataques-de-malware-por-dia-alerta-kaspersky
[14] MACVerify, «Network Security Fundamentals - Firewall, IDS/IPS Guide | MACVerify». Accedido: 6 de diciembre de 2025. [En línea]. Disponible en: https://macverify.com/blog/network-security-basics.php?utm_source=chatgpt.com
[15] Paloalto, «IPS. vs. IDS vs. Firewall: What Are the Differences?», Palo Alto Networks. Accedido: 6 de diciembre de 2025. [En línea]. Disponible en: https://www.paloaltonetworks.com/cyberpedia/firewall-vs-ids-vs-ips
[16] S. Vergara, «Arquitectura y Diseño de Seguridad de Red Perimetral», Serg Vergara y la informática. Accedido: 6 de diciembre de 2025. [En línea]. Disponible en: https://sergvergara.wordpress.com/2011/03/14/arquitectura-y-diseno-de-seguridad-de-red-perimetral/
[17] J. Jarauta, R. Palacios, y J. Sierra, «Seguridad Informática - Seguridad Perimetral», p. 45, 2022, Accedido: 6 de diciembre de 2025. [En línea]. Disponible en: https://pascua.iit.comillas.edu/palacios/seguridad/cap10.pdf
[18] J. Cárdenas y A. Mendoza, «Atributos de blockchain para la seguridad de datos dentro de las empresas: una revisión sistémica», *Revista de Ciencia, Tecnología e Innovación*, vol. 22, n.º 32, pp. 413-424, dic. 2024, doi: 10.56469/rcti.v22i32.1122.
[19] J. Pastor, «Qué es blockchain: la explicación definitiva para la tecnología más de moda», Xataka. Accedido: 6 de diciembre de 2025. [En línea]. Disponible en: https://www.xataka.com/especiales/que-es-blockchain-la-explicacion-definitiva-para-la-tecnologia-mas-de-moda
[20] L. Vásquez, R. Burciaga, Á. Zárate, y M. Vega, «Blockchain: Una tecnología que revoluciona la seguridad y la transparencia en la era digital.», *UCE Ciencia. Revista de postgrado*, vol. 11, n.º 3, oct. 2023, Accedido: 6 de diciembre de 2025. [En línea]. Disponible en: https://uceciencia.edu.do/index.php/OJS/article/view/341
[21] J. Peñataro, V. Santamaría, y E. Carreón, «Blockchain el lenguaje del futuro», p. 21, 2024, Accedido: 6 de diciembre de 2025. [En línea]. Disponible en: https://campusblockchain.org/wp-content/uploads/2024/03/Blockchain-2024-1.pdf
[22] J. Rezabala y R. Alcivar, «Blockchain aplicado a servicios gubernamentales: revisión sistemática de la literatura», *Revista Científica de Informática ENCRIPTAR*, vol. 8, n.º 15, pp. 217-244, feb. 2025, doi: 10.56124/encriptar.v8i15.012.
[23] M. Albiol y I. Alarcón, «Blockchain en salud: transformando la seguridad y la gestión de datos clínicos», *Atención Primaria*, vol. 56, n.º 5, p. 102848, may 2024, doi: 10.1016/j.aprim.2023.102848.
[24] R. Quiroz, F. Colina, C. Zúñiga, D. Ovalle, y J. Arana, «Modelo de arquitectura de seguridad de datos aplicando Blockchain en los procesos de admisión de una universidad Peruana»:, *Proceedings of the 21st LACCEI International Multi-Conference for Engineering, Education and Technology*, 2023, Accedido: 7 de diciembre de 2025. [En línea]. Disponible en: https://www.scopus.com/pages/publications/85172396223

[25] J. Huaman, G. Haro, y A. Mendoza, «Identificando Tecnologías Blockchain para la Protección de Información Sensible en Redes Sociales: Una Revisión Sistemática», *Innovation and Software*, vol. 6, n.º 1, pp. 6-23, mar. 2025, doi: 10.48168/innosoft.s23.a197.
[26] V. Burgos, «Cifrado de datos usando Cadena de Bloques (BlockChain) como tecnología de convergencia para dispositivos móviles asociados con IoT (Internet of Things), en la capa de aplicación del modelo de capas IoT"», Universidad Técnica del Norte, Ibarra, 2021. [En línea]. Disponible en: https://repositorio.utn.edu.ec/bitstream/123456789/11459/2/PG%20865%20TRABAJO%20GRADO.pdf
[27] J. D. Morillo y M. Sanguino, «Solución de blockchain descentralizada y segura para eventos de registro a prueba de manipulaciones», *Future Internet*, vol. 17, n.º 3, p. 108, mar. 2025, doi: 10.3390/fi17030108.
[28] N. E. Villa, I. Y. Tostado, y I. Y. Tostado Cortés, «Uso de Blockchain en la auditoría: ventajas y desafíos en la verificación de transacciones», *Vértice universitario*, vol. 27, n.º 96, dic. 2025, doi: 10.36792/rvu.v27i96.485.
[29] Kosmos, «Go Passwordless with Identity Based Authentication from», 1Kosmos. Accedido: 7 de diciembre de 2025. [En línea]. Disponible en: https://www.1kosmos.com
[30] M. Willson, «Private Key Vs Public Key – How They Work? - Blockchain Council». Accedido: 7 de diciembre de 2025. [En línea]. Disponible en: https://www.blockchain-council.org/blockchain/private-key-vs-public-key/
[31] C. Moreano, T. Escobar, V. Mena, y L. Herrera, «Tecnología Blockchain y su Implementación en los Sistemas Contables: Efectos en la Eficiencia y Transparencia», 2023, [En línea]. Disponible en: https://ciencialatina.org/index.php/cienciala/article/download/7578/11485?inline=1
[32] T. Shetty y A. Kulshestha, «Blockchain para sistemas de detección de intrusiones», en *Blockchain Technology for Emerging Applications*, Academic Press, 2022, pp. 107-136. doi: 10.1016/B978-0-323-90193-2.00003-X.
[33] J. Huang, Y. Chen, X. Wang, Z. Ouyang, y N. Du, «Esquema de optimización del sistema colaborativo de detección de intrusiones basado en tecnología blockchain», *Electronics*, vol. 14, n.º 2, p. 261, ene. 2025, doi: 10.3390/electronics14020261.
[34] Paloalto, «¿Qué es un sistema de detección de intrusiones?», Palo Alto Networks. Accedido: 7 de diciembre de 2025. [En línea]. Disponible en: https://www.paloaltonetworks.lat/cyberpedia/what-is-an-intrusion-detection-system-ids
[35] K. Hernández, «Tecnología Blockchain y sus aplicaciones potenciales en la gestión de la cadena de suministro en Guayaquil, Ecuador», Universidad Politécnica Salesiana, Guayaquil, 2023. Accedido: 7 de diciembre de 2025. [En línea]. Disponible en: https://dspace.ups.edu.ec/bitstream/123456789/25940/1/UPS-GT004569.pdf
[36] R. Hernández, C. Fernández, y M. Baptista, «Metodología de la investigación Sexta Edición», 2018, Accedido: 7 de diciembre de 2025. [En línea]. Disponible en: https://apiperiodico.jalisco.gob.mx/api/sites/periodicooficial.jalisco.gob.mx/files/metodologia_de_la_investigacion_-_roberto_hernandez_sampieri.pdf
[37] H. Cuadra, J. Balmaceda, y K. Gutiérrez, «Técnicas y herramientas para la gestión y control de la calidad en los procesos de gestión de proyecto en las organizaciones», Universidad Nacional Autónoma de Nicaragua, Managua, 2022. Accedido: 7 de diciembre de 2025. [En línea]. Disponible en: https://repositorio.unan.edu.ni/id/eprint/17192/1/17192-pdf.pdf

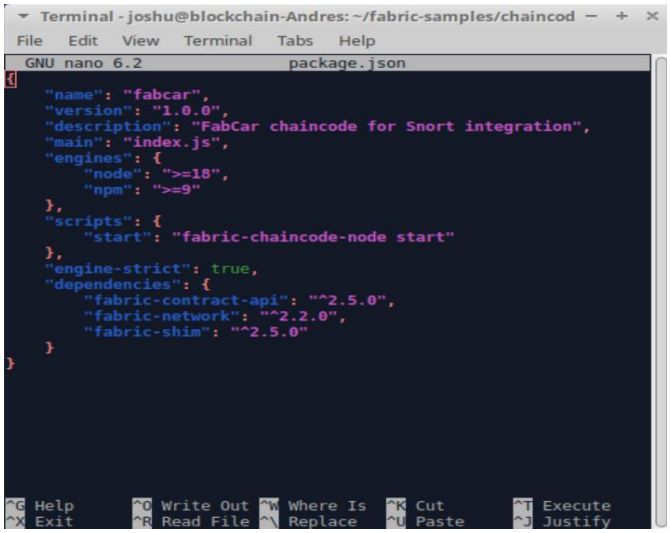
[38] J. Salazar, «La metodología y planteamiento del problema», 2023, Accedido: 7 de diciembre de 2025. [En línea]. Disponible en: https://sga.unemi.edu.ec/media/archivomateria/2023/09/19/archivomaterial_2023919 94936.pdf?utm_source=chatgpt.com

ANEXOS

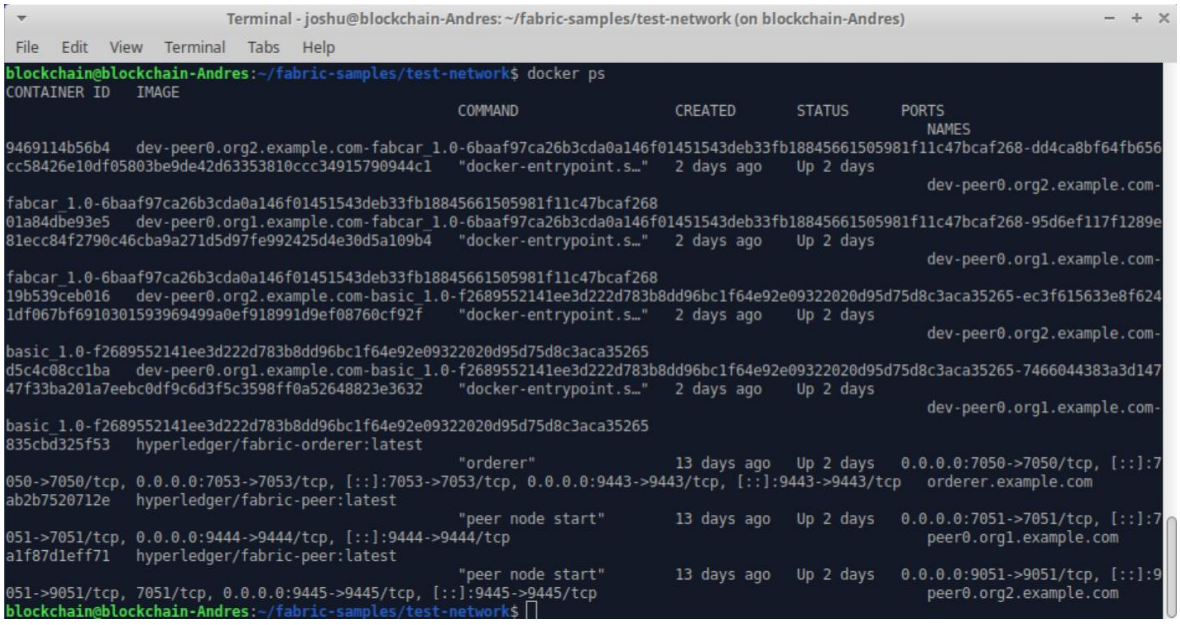
Anexo I. Instalación de Go



Anexo II. Configuración package.json y dependencias



Anexo III. Verifico contenedores activos de nodos y servicios auxiliares de red Hyperledger Fabric.



ANEXO IV. Log ante ataques UDP Flood

```
01/04-21:03:00.942434  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:10471 -> 192.168.20.16:80
01/04-21:03:00.942448  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:10480 -> 192.168.20.16:80
01/04-21:03:00.942461  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:10842 -> 192.168.20.16:80
01/04-21:03:00.942475  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:11184 -> 192.168.20.16:80
01/04-21:03:00.942488  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:11523 -> 192.168.20.16:80
01/04-21:03:00.942502  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:11733 -> 192.168.20.16:80
01/04-21:03:00.942518  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:11744 -> 192.168.20.16:80
01/04-21:03:00.942537  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:11763 -> 192.168.20.16:80
01/04-21:03:00.942552  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:11769 -> 192.168.20.16:80
01/04-21:03:00.942568  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:11775 -> 192.168.20.16:80
01/04-21:03:00.942582  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12343 -> 192.168.20.16:80
01/04-21:03:00.942597  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12348 -> 192.168.20.16:80
01/04-21:03:00.942610  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12353 -> 192.168.20.16:80
01/04-21:03:00.942623  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12361 -> 192.168.20.16:80
01/04-21:03:00.942637  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12410 -> 192.168.20.16:80
01/04-21:03:00.942651  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12414 -> 192.168.20.16:80
01/04-21:03:00.942664  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12418 -> 192.168.20.16:80
01/04-21:03:00.942678  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12519 -> 192.168.20.16:80
01/04-21:03:00.942692  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12523 -> 192.168.20.16:80
01/04-21:03:00.942705  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12524 -> 192.168.20.16:80
01/04-21:03:00.942718  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12525 -> 192.168.20.16:80
01/04-21:03:00.942732  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12528 -> 192.168.20.16:80
01/04-21:03:00.942745  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12531 -> 192.168.20.16:80
01/04-21:03:00.945311  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12695 -> 192.168.20.16:80
01/04-21:03:00.945361  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12790 -> 192.168.20.16:80
01/04-21:03:00.945377  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12796 -> 192.168.20.16:80
01/04-21:03:00.945392  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12801 -> 192.168.20.16:80
01/04-21:03:00.945407  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12807 -> 192.168.20.16:80
01/04-21:03:00.945422  [Drop] [**] [1:1000004:1] UDP Flood BLOQUEADO (hping3) [**] [Priority: 0] {UDP} 192.168.10.18:12808 -> 192.168.20.16:80
```

Anexo V. Log ante ataque ICMP_Flood

```
01/04-21:59:32.230208  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.230328  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.230448  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.230584  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.231565  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.231781  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.231902  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.232024  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.233540  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.233629  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.233683  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.233711  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.233737  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.233822  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.233851  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.233878  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.233958  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.234082  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.234206  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.234326  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.234447  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.234532  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.235339  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.235493  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.235614  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.235737  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
01/04-21:59:32.235837  [Drop] [**] [1:1000011:1] ICMP Echo Flood BLOQUEADO (hping3) [**] [Priority: 0] {ICMP} 192.168.10.18 -> 192.168.20.16
```

Anexo VI. log ante ataques SYN Flood

```
01/04-20:37:22.760261  [Drop] [**] [1:10000003:1] IPS: Bloqueado SYN Flood hacia HTTP [**] [Priority: 0] {TCP} 192.168.10.18:21160 -> 192.168.20.16:80
01/04-20:37:22.760261  [Drop] [**] [1:10000002:1] IPS: Bloqueado SYN Flood general [**] [Priority: 0] {TCP} 192.168.10.18:21160 -> 192.168.20.16:80
01/04-20:37:22.760286  [Drop] [**] [1:10000003:1] IPS: Bloqueado SYN Flood hacia HTTP [**] [Priority: 0] {TCP} 192.168.10.18:21161 -> 192.168.20.16:80
01/04-20:37:22.760286  [Drop] [**] [1:10000002:1] IPS: Bloqueado SYN Flood general [**] [Priority: 0] {TCP} 192.168.10.18:21161 -> 192.168.20.16:80
01/04-20:37:22.760318  [Drop] [**] [1:10000003:1] IPS: Bloqueado SYN Flood hacia HTTP [**] [Priority: 0] {TCP} 192.168.10.18:22014 -> 192.168.20.16:80
01/04-20:37:22.760318  [Drop] [**] [1:10000002:1] IPS: Bloqueado SYN Flood general [**] [Priority: 0] {TCP} 192.168.10.18:22014 -> 192.168.20.16:80
01/04-20:37:22.760359  [Drop] [**] [1:10000003:1] IPS: Bloqueado SYN Flood hacia HTTP [**] [Priority: 0] {TCP} 192.168.10.18:22042 -> 192.168.20.16:80
01/04-20:37:22.760359  [Drop] [**] [1:10000002:1] IPS: Bloqueado SYN Flood general [**] [Priority: 0] {TCP} 192.168.10.18:22042 -> 192.168.20.16:80
01/04-20:37:22.760388  [Drop] [**] [1:10000003:1] IPS: Bloqueado SYN Flood hacia HTTP [**] [Priority: 0] {TCP} 192.168.10.18:22045 -> 192.168.20.16:80
01/04-20:37:22.760388  [Drop] [**] [1:10000002:1] IPS: Bloqueado SYN Flood general [**] [Priority: 0] {TCP} 192.168.10.18:22045 -> 192.168.20.16:80
01/04-20:37:22.760459  [Drop] [**] [1:10000003:1] IPS: Bloqueado SYN Flood hacia HTTP [**] [Priority: 0] {TCP} 192.168.10.18:22209 -> 192.168.20.16:80
01/04-20:37:22.760459  [Drop] [**] [1:10000002:1] IPS: Bloqueado SYN Flood general [**] [Priority: 0] {TCP} 192.168.10.18:22209 -> 192.168.20.16:80
01/04-20:37:22.762760  [Drop] [**] [1:10000003:1] IPS: Bloqueado SYN Flood hacia HTTP [**] [Priority: 0] {TCP} 192.168.10.18:22331 -> 192.168.20.16:80
01/04-20:37:22.762760  [Drop] [**] [1:10000002:1] IPS: Bloqueado SYN Flood general [**] [Priority: 0] {TCP} 192.168.10.18:22331 -> 192.168.20.16:80
01/04-20:37:22.762941  [Drop] [**] [1:10000003:1] IPS: Bloqueado SYN Flood hacia HTTP [**] [Priority: 0] {TCP} 192.168.10.18:22332 -> 192.168.20.16:80
01/04-20:37:22.762941  [Drop] [**] [1:10000002:1] IPS: Bloqueado SYN Flood general [**] [Priority: 0] {TCP} 192.168.10.18:22332 -> 192.168.20.16:80
01/04-20:37:22.763011  [Drop] [**] [1:10000003:1] IPS: Bloqueado SYN Flood hacia HTTP [**] [Priority: 0] {TCP} 192.168.10.18:22339 -> 192.168.20.16:80
01/04-20:37:22.763011  [Drop] [**] [1:10000002:1] IPS: Bloqueado SYN Flood general [**] [Priority: 0] {TCP} 192.168.10.18:22339 -> 192.168.20.16:80
01/04-20:37:22.763104  [Drop] [**] [1:10000003:1] IPS: Bloqueado SYN Flood hacia HTTP [**] [Priority: 0] {TCP} 192.168.10.18:22862 -> 192.168.20.16:80
01/04-20:37:22.763104  [Drop] [**] [1:10000002:1] IPS: Bloqueado SYN Flood general [**] [Priority: 0] {TCP} 192.168.10.18:22862 -> 192.168.20.16:80
01/04-20:37:22.763161  [Drop] [**] [1:10000003:1] IPS: Bloqueado SYN Flood hacia HTTP [**] [Priority: 0] {TCP} 192.168.10.18:22871 -> 192.168.20.16:80
01/04-20:37:22.763161  [Drop] [**] [1:10000002:1] IPS: Bloqueado SYN Flood general [**] [Priority: 0] {TCP} 192.168.10.18:22871 -> 192.168.20.16:80
01/04-20:37:22.763363  [Drop] [**] [1:10000003:1] IPS: Bloqueado SYN Flood hacia HTTP [**] [Priority: 0] {TCP} 192.168.10.18:23873 -> 192.168.20.16:80
01/04-20:37:22.763363  [Drop] [**] [1:10000002:1] IPS: Bloqueado SYN Flood general [**] [Priority: 0] {TCP} 192.168.10.18:23873 -> 192.168.20.16:80
01/04-20:37:22.766757  [Drop] [**] [1:10000003:1] IPS: Bloqueado SYN Flood hacia HTTP [**] [Priority: 0] {TCP} 192.168.10.18:23882 -> 192.168.20.16:80
01/04-20:37:22.766757  [Drop] [**] [1:10000002:1] IPS: Bloqueado SYN Flood general [**] [Priority: 0] {TCP} 192.168.10.18:23882 -> 192.168.20.16:80
```

Anexo VIII. Resumen de casos

Resumen de procesamiento de casos

		Casos					
		Válido		Perdidos		Total	
	Tipo de ataque	N	Porcentaje	N	Porcentaje	N	Porcentaje
Porcentaje de paquetes perdidos (%)	ICMP Flood	60	100,0%	0	0,0%	60	100,0%
	UDP Flood	60	100,0%	0	0,0%	60	100,0%
	SYN Flood	60	100,0%	0	0,0%	60	100,0%
Tiempo de ida	ICMP Flood	60	100,0%	0	0,0%	60	100,0%
	UDP Flood	60	100,0%	0	0,0%	60	100,0%
	SYN Flood	60	100,0%	0	0,0%	60	100,0%
Disponibilidad del sistema (%)	ICMP Flood	60	100,0%	0	0,0%	60	100,0%
	UDP Flood	60	100,0%	0	0,0%	60	100,0%
	SYN Flood	60	100,0%	0	0,0%	60	100,0%
CPU libre	ICMP Flood	60	100,0%	0	0,0%	60	100,0%
	UDP Flood	60	100,0%	0	0,0%	60	100,0%
	SYN Flood	60	100,0%	0	0,0%	60	100,0%
Uso de memoria	ICMP Flood	60	100,0%	0	0,0%	60	100,0%
	UDP Flood	60	100,0%	0	0,0%	60	100,0%
	SYN Flood	60	100,0%	0	0,0%	60	100,0%