

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Tema:

SISTEMA DE DETECCIÓN DE INTRUSOS (IDS) BASADO EN MACHINE LEARNING PARA EL CONTROL DE LA RED EN LA UNIDAD EDUCATIVA "19 DE SEPTIEMBRE" EN LA CIUDAD DE SALCEDO

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Tecnologías de la Información

ÁREA: Gestión de tecnologías de la Información
LÍNEA DE INVESTIGACIÓN: Tecnologías de la información y sistemas de control

AUTOR: Jairo Guillermo Lascano Banshuy
TUTOR: Ing. Andrea Patricia Sánchez Zumba, Mg.

Ambato - Ecuador
febrero – 2025

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA DE DETECCIÓN DE INSTRUSOS (IDS) BASADO EN MACHINE LEARNING PARA EL CONTROL DE LA RED EN LA UNIDAD EDUCATIVA “19 DE SEPTIEMBRE” EN LA CIUDAD DE SALCEDO, desarrollado bajo la modalidad Proyecto de Investigación por el señor Jairo Guillermo Lascano Banshuy, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2025.

Firmado electrónicamente por:
ANDREA PATRICIA SANCHEZ ZUMBA

Ing. Andrea Patricia Sánchez Zumba, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: SISTEMA DE DETECCIÓN DE INSTRUSOS (IDS) BASADO EN MACHINE LEARNING PARA EL CONTROL DE LA RED EN LA UNIDAD EDUCATIVA "19 DE SEPTIEMBRE" EN LA CIUDAD DE SALCEDO es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2025.



Jairo Guillermo Lascano Banshuy

C.C. 1805730841

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2025.



Jairo Guillermo Lascano Banshuy

C.C. 1805730841

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Jairo Guillermo Lascano Banshuy, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA DE DETECCIÓN DE INSTRUSOS (IDS) BASADO EN MACHINE LEARNING PARA EL CONTROL DE LA RED EN LA UNIDAD EDUCATIVA “19 DE SEPTIEMBRE” EN LA CIUDAD DE SALCEDO, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, febrero 2025.

Firmado electrónicamente por:
FRANKLIN OSWALDO
MAYORGA MAYORGA

Ing. Franklin Oswaldo Mayorga Mayorga, Mg.
PRESIDENTE DEL TRIBUNAL

Firmado electrónicamente por:
DAVID OMAR GUEVARA
AULESTIA

Ing. David Guevara Aulestia, Mg.
PROFESOR CALIFICADOR

Firmado electrónicamente por:
JOSE VICENTE
MORALES LOZADA

Ing. José Morales Lozada, PhD.
PROFESOR CALIFICADOR

DEDICATORIA

A mis padres, Nancy y Guillermo por su apoyo incondicional en cada una de las etapas de mi vida, por inculcarme los valores de la humidad, el trabajo duro y el respeto. Siendo ellos la inspiración y el motivo para superarme día con día.

A mi hermana Karen, parte importante en mi vida, por el amor y apoyo incondicional que demuestra tenerme día a día.

A mi padrino Washington, mi madrina Gladys por ser unos segundos padres. Gracias por sus consejos y motivación para no rendirme y seguir adelante.

A mis primos Bryan y Britney por ser hermanos de toda la vida, por el compañerismo, el apoyo y motivación que me han brindado desde siempre.

A mis amigos Karina, Martín, Matías por estar para mi en los momentos difíciles y en las alegrías a lo largo de la carrera universitaria. Gracias por enseñarme el verdadero significado de la amistad incondicional.

Finalmente, a toda mi familia, abuelos y tíos por siempre brindarme su apoyo, en especial a mi abuela Marieta por ser mi guía moral y por todo su cuidado en mis primeros años de vida.

AGRADECIMIENTO

Primeramente, quiero agradecer a Dios por darme vida, salud, sabiduría y perseverancia para poder culminar mi carrera universitaria.

A mis Padres Nancy y Guillermo por sus palabras de aliento y apoyo en momentos difíciles.

A mi padrino Washington y mi madrina Gladys por ser una brújula moral y por los consejos para seguir adelante.

A mi tutora Ing. Andrea Sánchez por su apoyo, conocimientos y guía para la elaboración de este trabajo de investigación.

A mi tutor empresarial Lcdo. Mauricio Banda por la apertura y facilidades brindadas para que el proyecto se pueda llevar a cabo.

A mis amigos Karina, Martín, Matías por su compañía, la motivación que me han brindado, y sobre todo por mostrarme que la amistad verdadera existe, su amistad es el mejor regalo que puede haber tenido en esta travesía.

A mi primo Bryan por ser mi confidente y compañero de aventuras a lo largo de esta travesía. Gracias por siempre estar en los momentos difíciles.

A toda mi familia en general, por apoyarme siempre y brindarme todo el amor y comprensión.

Finalmente, a todo el personal de la "Unidad Educativa 19 de Septiembre" por su colaboración y disposición.

ÍNDICE GENERAL DE CONTENIDOS

PORTADA i
APROBACIÓN DEL TUTOR ii
AUTORÍA iii
DERECHOS DE AUTOR iv
APROBACIÓN DEL TRIBUNAL DE GRADO v
DEDICATORIA vi
AGRADECIMIENTO vii
ÍNDICE GENERAL DE CONTENIDOS viii
ÍNDICE DE TABLAS x
ÍNDICE DE FIGURAS xii
ÍNDICE DE ANEXOS xv
RESUMEN EJECUTIVO xvi
ABSTRACT xvii
CAPÍTULO I. MARCO TEÓRICO 1
1.1 Tema de investigación 1
1.1.1 Planteamiento del problema 1
1.2 Antecedentes investigativos 2
1.3 Fundamentación teórica 7
1.3.1 Tecnologías de la Información y Comunicación (TIC) 7
1.3.2 Seguridad Informática y Ciberseguridad Avanzada 8
1.3.3 Seguridad en Redes Informáticas mediante Inteligencia Artificial 10
1.3.4 Sistema de detección de intrusos (IDS) basado en machine learning 12
1.3.5 Transformación Digital en el Sector Educativo 13
1.3.6 Gestión de Recursos Tecnológicos en Instituciones Educativas 14
1.3.7 Control de la Red en la Unidad Educativa 16
1.4 Objetivos 17
1.4.1 Objetivo general 17
1.4.2 Objetivos específicos 17
CAPÍTULO II. METODOLOGÍA 18
2.1 Materiales 18

2.1.1 Entrevista al rector y docente encargado de los laboratorios de computación de la “Unidad Educativa 19 de Septiembre” 18
2.1.2 Encuesta dirigida a la población con mayor interacción con la red 19
2.1.3 Observación de la tolerancia de la red al recibir ataques cibernéticos 23
2.2 Métodos 25
2.2.1 Modalidad de la investigación 25
2.2.2 Población y muestra 25
2.2.3 Recolección de información 27
2.2.4 Procesamiento y análisis de datos 54
CAPÍTULO III. RESULTADOS Y DISCUSIÓN 68
3.1 Análisis y discusión de los resultados 68
3.1.1 Análisis de la infraestructura de red inicial 68
3.1.2 Software de detección de intrusiones (IDS) 71
3.1.3 Metodologías para proyectos de ciencia de datos y machine learning. 73
3.2 Desarrollo de la propuesta 78
3.2.1 Fase 1: Comprensión del negocio 79
3.2.2 Fase 2: Comprensión de los datos 84
3.2.3 Fase 3: Preparación de los datos 89
3.2.4 Fase 4: Modelado 99
3.2.5 Fase 5: Evaluación 106
3.2.6 Fase 6: Despliegue 118
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES 147
4.1 Conclusiones 147
4.2 Recomendaciones 148
REFERENCIAS BIBLIOGRÁFICAS 150
ANEXOS 154

ÍNDICE DE TABLAS

Tabla 1. Cadenas de búsqueda realizadas. 3
Tabla 2. Búsqueda en repositorios de instituciones de educación superior. 4
Tabla 3. Criterios de inclusión y exclusión. 4
Tabla 4. Ficha de entrevista 18
Tabla 5. Modelo de ficha de observación. 24
Tabla 6. Población de estudio. 26
Tabla 7. Muestra de estudio. 27
Tabla 8. Escalas de confiabilidad de Alfa de Cronbach. 29
Tabla 9. Encuesta al rector de la Unidad Educativa “19 de Septiembre”. 45
Tabla 10. Encuesta al docente encargado del laboratorio de computación de la Unidad Educativa “19 de Septiembre”. 48
Tabla 11. Ficha de observación acerca de la tolerancia de la red y reacción de usuarios a distintos ataques cibernéticos. 51
Tabla 12. Respuestas más seleccionadas del grupo 1 de preguntas 56
Tabla 13. Respuestas más seleccionadas del segundo grupo de preguntas. 60
Tabla 14. Respuestas más seleccionadas del tercer grupo de preguntas. 64
Tabla 15. Comparativa softwares IDS. 71
Tabla 16. Metodologías para proyectos de ciencias de datos y machine learning. 75
Tabla 17. Requerimientos funcionales del sistema IDS. 80
Tabla 18. Requerimientos no funcionales del sistema IDS. 81
Tabla 19. Fases del proyecto. 83
Tabla 20. Ficha de búsqueda datasets. 84
Tabla 21. Ficha de búsqueda dataset UNSW-NB15. 85
Tabla 22. Ficha de búsqueda dataset CSE-CIC-DDoS 2019. 86
Tabla 23. Ficha de búsqueda de dataset CIC-IoV 2024. 87
Tabla 24. Comparativa entre datasets. 88
Tabla 25. Ficha técnica comparativa para algoritmos de machine learning. 100
Tabla 26. Ficha de aceptación prueba 01 en base a resultados obtenidos en Figura 35. 109
Tabla 27. Ficha de aceptación prueba 02 en base a resultados obtenidos en Figura 35. 112

Tabla 28. Prueba de aceptación 03 en base a resultados obtenidos en Figura 39. ... 117
Tabla 29. Criterios de inclusión y exclusión para software IDS. ... 118
Tabla 30. Ficha comparativa de herramientas IDS. ... 119
Tabla 31. Ficha de aceptación del sistema IDS prueba 01 al encargado del laboratorio de computación en la Unidad Educativa “19 de Septiembre”. ... 132
Tabla 32. Criterios de inclusión y exclusión para softwares de monitoreo. ... 133
Tabla 33. Ficha comparativa de softwares de monitoreo. ... 134
Tabla 34. Ficha de aceptación de sistema IDS prueba 02 al encargado del laboratorio de computación en la Unidad Educativa “19 de Septiembre”. ... 144
Tabla 35. Evaluación final de desempeño del sistema IDS. ... 146

ÍNDICE DE FIGURAS

Figura 1. Algoritmo para el cálculo Alfa de Cronbach. 29
Figura 2. Resultados pregunta 1. 30
Figura 3. Resultados pregunta 2. 31
Figura 4. Resultados pregunta 3. 32
Figura 5. Resultados pregunta 4. 33
Figura 6. Resultados pregunta 5. 34
Figura 7. Resultados pregunta 6. 35
Figura 8. Resultados pregunta 7. 36
Figura 9. Resultados pregunta 8. 37
Figura 10. Resultados pregunta 9. 38
Figura 11. Resultados pregunta 10. 39
Figura 12. Resultado pregunta 11. 40
Figura 13. Resultados pregunta 12. 41
Figura 14. Resultados pregunta 13. 42
Figura 15. Resultados pregunta 14. 43
Figura 16. Resultados pregunta 15. 44
Figura 17. Correlación de preguntas del grupo 1. 55
Figura 18. Correlación de preguntas del grupo 2. 59
Figura 19. Correlación de preguntas del grupo 3. 63
Figura 20. Topología de red inicial de la Unidad Educativa “19 de Septiembre”. 68
Figura 21. Diagrama de metodología CRISP-DM 78
Figura 22. Archivos CSV de dataset CSE-CIC-DDoS 2019. 90
Figura 23. Información general de archivo CSV. 91
Figura 24. Descripción estadística de archivo CSV. 91
Figura 25. Información de filas y columnas de archivo CSV. 92
Figura 26. Primeras cinco filas de archivo CSV. 92
Figura 27. Últimas cinco filas de archivo CSV. 93
Figura 28. Columnas del dataset CSE-CIC-DDoS 2019. 94
Figura 29. Filtrado de valores nulos y repetidos de archivos CSV. 97
Figura 30. Eliminación de outliers en archivos CSV. 98
Figura 31. Archivos CSV depurados. 99

Figura 32. Pasos iniciales de entrenamiento del modelo. 103
Figura 33. Entrenamiento inicial del modelo. 104
Figura 34. Búsqueda de mejores hiperparámetros. 105
Figura 35. Primera evaluación del modelo con los mejores hiperparámetros. 108
Figura 36. Segunda evaluación del modelo con hiperparámetos ajustados. 110
Figura 37. Distribución de clases en el dataset. 112
Figura 38. Unión de información complementaria a dataset principal. 113
Figura 39. Entrenamiento del modelo utilizando la técnica Synthetic Minority Over-sampling Technique (SMOTE). 114
Figura 40. Tercera evaluación con datos complementarios y aplicando la técnica SMOTE. 116
Figura 41. Prueba de validación cruzada. 117
Figura 42. Configuración de rango de análisis de Suricata. 121
Figura 43. Configuración de interface de red en Suricata. 122
Figura 44. Configuración formato de salida de Suricata 123
Figura 45. Configuración de archivo de reglas de Suricata. 124
Figura 46. Reglas complementarias para Suricata. 124
Figura 47. Arquitectura de la red con Suricata y modelo de machine learning en la Unidad Educativa “19 de Septiembre”. 125
Figura 48. Configuración de carpeta compartida en Samba. 126
Figura 49. Configuración de actualización de archivo eve.json. 126
Figura 50. Carpeta compartida con archivo eve.json. 127
Figura 51. Configuración de conexión de modelo con archivo eve.json. 128
Figura 52. Función para procesar archivo eve.json en script. 129
Figura 53. Clase para el monitoreo y detección de cambios en el archivo eve.json. 130
Figura 54. Ejecución de prueba de script de integración de modelo con Suricata... 131
Figura 55. Creación de API a partir del archivo eve.json. 135
Figura 56. Dashboard generado en Grafana a partir de API de Suricata. 136
Figura 57. Configuración de servicio para automatizar script para archivo compartido eve.json 137
Figura 58. Configuración de servicio para automatizar generación de API en Suricata 138
Figura 59. Configuración para almacenar información de intentos de ataque. 139

Figura 60. Función para guardar detalles de intentos de ataques en formato CSV. 140
Figura 61. Modificaciones a la función de análisis de líneas de formato JSON..... 141
Figura 62. Arquitectura final de la red de la Unidad Educativa “19 de Septiembre”.
..... 142
Figura 63. Ejecución de detección automática luego de cambios realizados. 143

ÍNDICE DE ANEXOS

Anexo A. Árbol de problemas 154
Anexo B. Código de limpieza de datos opción 1. 155
Anexo C Código de limpieza de datos opción 2. 158
Anexo D. Código de unión de datasets limpios en dataset universal. 160
Anexo E. Código de generación de API en servidor IDS. 162
Anexo F. Código final de preprocesamiento de dataset universal. 165
Anexo G. Código de generación de modelo de machine learning. 169
Anexo H. Código de ejecución de modelo de machine learning para la detección automática. 172
Anexo I. Manual de usuario de sistema IDS. 177

RESUMEN EJECUTIVO

La creciente sofisticación de las amenazas cibernéticas resalta la necesidad de soluciones robustas para proteger las redes institucionales. El presente proyecto se centra en la implementación de un Sistema de Detección de Intrusos (IDS) basado en machine learning para la Unidad Educativa "19 de Septiembre". Las evaluaciones iniciales de la red revelaron vulnerabilidades críticas de seguridad, incluyendo problemas frecuentes de conectividad, acceso limitado e insatisfacción de los usuarios, principalmente debido a la falta de atención a la ciberseguridad dentro de la institución. Para abordar estos desafíos, se desarrolló un conjunto de datos híbridos al combinar datos en tiempo real recolectados a través del IDS con datasets públicos disponibles. Este enfoque garantizó la relevancia y robustez del dataset, mejorando la precisión del modelo en la clasificación de tráfico benigno y la detección de posibles ataques. Tras analizar varios algoritmos de machine learning, se seleccionó Random Forest debido a su alta adaptabilidad, sólido desempeño y compatibilidad con las limitaciones de recursos del proyecto. El modelo entrenado logró buenos resultados en precisión, recall y confiabilidad general, proporcionando una base sólida para el IDS.

Este enfoque permite a las instituciones educativas y otros entornos organizacionales adaptarse de manera proactiva a las amenazas cibernéticas emergentes, que evolucionan constantemente en complejidad y alcance. Al fortalecer la infraestructura de red, se promueve un entorno seguro y confiable que protege tanto los datos como las operaciones críticas de la institución, además, esta mejora contribuye a garantizar la confianza de los usuarios en la utilización de la infraestructura en general.

Palabras clave: Sistema de detección de intrusiones, IDS, Machine learning, Red educativa, Infraestructura de red, Random Forest, Suricata, Grafana.

ABSTRACT

The increasing sophistication of cyber threats highlights the need for robust solutions to protect institutional networks. This project focuses on the implementation of a Machine Learning-based Intrusion Detection System (IDS) for the "19 de Septiembre" Educational Unit. Initial network assessments revealed critical security vulnerabilities, including frequent connectivity issues, limited access, and user dissatisfaction, primarily caused by the lack of attention to cybersecurity within the institution.

To address these challenges, a hybrid dataset was developed by combining real-time data collected through the IDS with publicly available datasets. This approach ensured the dataset's relevance and robustness, enhancing the model's accuracy in classifying benign traffic and detecting potential attacks. After evaluating various machine learning algorithms, Random Forest was selected due to its high adaptability, strong performance, and compatibility with the project's resource constraints. The trained model achieved exceptional results in terms of accuracy, recall, and overall reliability, providing a solid foundation for the IDS.

This approach allows educational institutions and other organizational environments to proactively adapt to emerging cyber threats, which are constantly evolving in complexity and scope. By strengthening the network infrastructure, a secure and reliable environment is fostered, protecting both data and the institution's critical operations. Furthermore, this improvement contributes to ensuring user trust in the overall use of the infrastructure.

Keywords: Intrusion detection system, IDS, Machine learning, Educational network, Network infrastructure, Random Forest, Suricata, Grafana.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

SISTEMA DE DETECCIÓN DE INSTRUSOS (IDS) BASADO EN MACHINE LEARNING PARA EL CONTROL DE LA RED EN LA UNIDAD EDUCATIVA "19 DE SEPTIEMBRE" EN LA CIUDAD DE SALCEDO

1.1.1 Planteamiento del problema

La globalización actual de las tecnologías de la información ha supuesto un cambio significativo en el manejo de la información confidencial de las personas. A medida que la digitalización de los datos crece anualmente, también aumentan los tipos de ciberataques que buscan acceder a esa información, lo que representa grandes pérdidas económicas. Por ejemplo, en el año 2023, el costo promedio por la filtración de datos llegó a los 4.45 millones de dólares, lo que significó un aumento del 2.3 % con respecto al año 2022. Si se toma en cuenta un periodo corto, como el comprendido entre 2020 y 2023, el aumento del costo ha sido del 15.3 % [1].

En el panorama latinoamericano, con datos analizados entre junio de 2022 y julio de 2023 y de junio de 2021 a julio de 2022, se determinó un incremento del 617 % en ataques a computadoras y dispositivos móviles relacionados con malware, y un crecimiento del 50 % en ataques de phishing y troyanos enfocados en el sector bancario. Según expertos, el retorno a las actividades económicas tras la pandemia fue el principal detonante del aumento de los intentos de ataque en la región. Entre los países más afectados se encuentran Brasil, con 134 millones de intentos de ataque, México (43 millones), Perú (31.5 millones), Colombia (30.9 millones), Ecuador (12.2 millones), Chile (10.5 millones) y Argentina (9.4 millones) [2].

En el caso local de Ecuador, los ciberataques no han pasado desapercibidos. Con el aumento exponencial del uso de redes sociales y varios servicios online, es crucial reforzar la ciberseguridad. Teniendo como antecedentes los ataques a la Corporación Nacional de Telecomunicaciones (CNT), a la Agencia Nacional de Tránsito (ANT), entre otras entidades, han subrayado esta necesidad. En respuesta, el Ministerio de

Telecomunicaciones ecuatoriano ha presentado una estrategia nacional de ciberseguridad, concebida como una herramienta fundamental para enfrentar el actual contexto tecnológico. Esta estrategia se sustenta en seis pilares clave de ciberseguridad [3].

En el contexto educativo, la ciberseguridad no recibe la atención que merece. Por ejemplo, Chuqi [4], al analizar la ciberseguridad en una institución educativa, determinó que esta se encuentra en un nivel insuficiente. Se observó la presencia de accesos no autorizados, pérdida de datos y problemas de confidencialidad. Las infraestructuras de red educativas se pueden considerar como un eslabón débil en las organizaciones gubernamentales, ya que sus sistemas de protección suelen ser precarios, lo que las convierte en un punto vulnerable para la entrada de información valiosa y delicada.

De manera similar, Villamar [5] realizó un análisis en la Unidad Educativa "Luis Alfredo Noboa Icaza" y encontró que los ataques más comunes eran troyanos y ransomware. La información recolectada mostró que una parte significativa de la población de estudio fue víctima de estos ataques. Villamar recomienda centrarse en estrategias de prevención, como el uso de firewalls y software antivirus, así como mantener respaldos constantes de información indispensable. También destaca la importancia de utilizar contraseñas fuertes para disminuir el riesgo de ser víctimas de ataques informáticos.

1.2 Antecedentes investigativos

Para llevar a cabo la investigación, se realizó la respectiva revisión bibliográfica utilizando bases de datos como Scopus e IEEE Xplore. Con el fin de obtener resultados óptimos, se emplearon cadenas de búsqueda que incluían las siguientes palabras clave: Intrusion detection system, IDS, machine learning, intrusion detection system in Ecuador, IDS in educational institutions in Ecuador. Se estableció un rango de periodo de publicación comprendido entre los años 2020 y 2024. Como resultado de este proceso, se generó la Tabla 1, la cual proporciona una descripción detallada de las cadenas de búsqueda utilizadas y el número de artículos encontrados.

Tabla 1. Cadenas de búsqueda realizadas.

	Repositorio	Cadena	Nº Artículos
Cadena 1	Scopus	(TITLE-ABS-KEY (intrusion AND detection AND system) AND TITLE-ABS-KEY (ids) AND TITLE-ABS-KEY (machine AND learning)) AND PUBYEAR > 2019 AND PUBYEAR < 2025 AND (LIMIT-TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA , "ENGI")) AND (LIMIT-TO (DOCTYPE , "ar") OR LIMIT-TO (DOCTYPE , "ch")) AND (LIMIT-TO (EXACTKEYWORD , "Intrusion Detection") OR LIMIT-TO (EXACTKEYWORD , "Network Security") OR LIMIT-TO (EXACTKEYWORD , "Intrusion Detection Systems") OR LIMIT-TO (EXACTKEYWORD , "Machine Learning")) AND (LIMIT-TO (LANGUAGE , "English") OR LIMIT-TO (LANGUAGE , "Spanish"))	103
	IEEE Xplore	("All Metadata":intrusion detection system) AND ("All Metadata":ids) AND ("All Metadata":machine learning) Filters Applied: Intrusion Detection SystemIntrusion DetectionMachine LearningConvolutional Neural NetworkLearning AlgorithmsNeural NetworkAnomaly DetectionArtificial Neural Network2020 - 2024	167
Cadena 2	Scopus	(TITLE-ABS-KEY (intrusion AND detection AND system) AND TITLE-ABS-KEY (educational AND institutions) AND TITLE-ABS-KEY (ids)) AND PUBYEAR > 2019 AND PUBYEAR < 2024 AND (LIMIT-TO (EXACTKEYWORD , "Intrusion Detection")) AND (LIMIT-TO (OA , "all"))	2
	IEEE Xplore	("All Metadata":intrusion detection system) AND ("All Metadata":educational institutions) AND ("All Metadata":IDS) Filters Applied: 2020 - 2024	1
Cadena 3	Scopus	(TITLE-ABS-KEY (intrusion AND detection AND system) AND TITLE-ABS-KEY (ids) AND TITLE-ABS-KEY (ecuador) OR TITLE-ABS-KEY (educational AND institutions) AND TITLE-ABS-KEY (intrusion AND detection AND system)) AND PUBYEAR > 2019 AND PUBYEAR < 2024	3
	IEEE Xplore	("All Metadata":intrusion detection system) AND ("All Metadata":Ecuador) AND ("All Metadata":IDS) OR ("All Metadata":educational institutions) AND ("All Metadata":intrusion detection system) Filters Applied: 2020 - 2024	2
		TOTAL	278

Con el objetivo de ampliar el conjunto de fuentes disponibles, se decidió realizar búsquedas en los repositorios académicos de instituciones de educación superior en Ecuador, tales como la Universidad Técnica de Ambato (UTA) y la Universidad Nacional de Chimborazo (UNACH). Los resultados de estas búsquedas se presentan en la Tabla 2.

Tabla 2. Búsqueda en repositorios de instituciones de educación superior.

Universidad	Filtro de Búsqueda	Nº Artículos
Universidad Técnica de Ambato	Facultad de Ingeniería en Sistemas, Electrónica e Industrial.	7
Universidad Nacional de Chimborazo	Escuela de Ingeniería en Sistemas y Computación	7
	TOTAL	14

Después revisar la literatura pertinente, se procedió al análisis de la documentación que guardara una alta relevancia con respecto al tema propuesto, lo que resultó en la identificación de 292 artículos iniciales. Estos fueron evaluados según los criterios de inclusión y exclusión detallados en la Tabla 3. Como resultado, se seleccionaron 44 trabajos de investigación que se consideraron capaces de aportar de manera significativa al desarrollo del marco teórico de la investigación propuesta.

Tabla 3. Criterios de inclusión y exclusión.

Inclusión	• La información debe abordar específicamente el concepto, implementación o mejora de sistemas de detección de intrusos (IDS) y el uso de machine learning en estos sistemas. • Información sobre experiencias exitosas de implementación de IDS basados en machine learning en entornos educativos o instituciones similares, destacando sus beneficios y desafíos. • Información que incluya detalles técnicos sobre la configuración, mantenimiento y gestión de sistemas IDS basados en machine learning. • Consideraciones sobre la seguridad de la red, específicamente en el contexto de la implementación de IDS basados en machine learning en redes educativas o similares.
Exclusión	• Contenido que no tiene relación directa con la implementación de IDS basados en machine learning. • Contenido sobre tecnologías o soluciones de detección de intrusos que sean desactualizadas o no relevantes para el contexto actual de machine learning y seguridad de la red. • Contenido que no aborda adecuadamente las preocupaciones de seguridad de la red en el contexto educativo y de machine learning. • Contenido publicado hace más de cinco años que puede no reflejar las tendencias y tecnologías actuales en machine learning y seguridad de redes.

La seguridad informática ha cobrado un papel crucial en el mundo actual, ya que gran parte de la población confía ciegamente en los sistemas de protección tradicionales predominantes en la mayoría de las infraestructuras de red. No obstante, muchos de estos sistemas se vuelven obsoletos con el tiempo. Así como la tecnología avanza positivamente, también lo hacen las amenazas digitales. Los virus informáticos, las técnicas de intrusión y los métodos para colapsar redes se refinan continuamente con el objetivo de vulnerar las medidas de seguridad existentes. Por ello, con el auge de la

inteligencia artificial y el machine learning, se ha buscado integrar estas tecnologías en los Sistemas de detección de intrusos (IDS) de diversas maneras. Liu, Gu, y Wang [6], proponen la configuración de un IDS en tres pasos basado en los algoritmos K-means y Random Forest, junto con el uso de una red neuronal convolucional, del inglés convolutional neural network (CNN) y los conjuntos de datos NSL-KDD y CIC-IDS2017. En el primer paso, se realiza el preprocesamiento de la información para limpiar los conjuntos de datos y así mejorar la precisión del modelo en sus predicciones. En el segundo paso, se combinan los algoritmos K-means y Random Forest: K-means agrupa eventos de detección de intrusiones y Random Forest clasifica el resultado de la agrupación. En el tercer paso, se discriminan los resultados obtenidos para transmitir la información al IDS. Los resultados fueron notables, logrando una precisión del 85.24% con los datos NSL-KDD y del 99.91% con el conjunto CIC-IDS2017.

Dado que el campo de la inteligencia artificial y el machine learning es muy amplio, se pueden explorar otros métodos para la configuración de los sistemas de detección de intrusos. Un ejemplo de esto es el estudio realizado por Sstla, Kolli, Voggu, Bhavanam y Vallabhasoyula [7], en el cual se utiliza redes neuronales convolucionales profundas, del inglés deep convolutional neural networks (DCNN), que han tenido grandes logros en el campo de la visión por computadora. En su modelo, utilizan el conjunto de datos NSL-KDD, el algoritmo de clasificación SVM y el kernel RBF. De las 2490 muestras de prueba que consideraron, 1200 se predijeron como verdaderos positivos y solo una muestra tuvo un falso negativo. De las 1280 muestras restantes, 910 se reconocieron como verdaderos negativos y 370 como falsos positivos. El modelo reconoció correctamente 2210 de las 2490 muestras, con una precisión del 85%. Sin embargo, al utilizar el kernel RBF con un valor gamma de “1”, de las 2490 muestras de prueba, solo 1320 se clasificaron correctamente, con una precisión del 54.21%. Esto sugiere que para determinar la configuración óptima que se adapte a las necesidades específicas, se puede recurrir a un proceso de prueba y error.

En comparación, existen modelos de aprendizaje automático que pueden ofrecer mejores resultados y adaptarse de manera más orgánica a las necesidades de la red. Por ejemplo, Prasath, Sethi, Mohanty, Bera, y Samantaray [8] realizaron una comparativa entre modelos tradicionales de redes neuronales profundas, del inglés

deep neural networks (DNN) y modelos de aprendizaje continuo basados en repetición. Para estos últimos, desarrollaron tres enfoques: aprender sin olvidar, repetición de experiencias y repetición de experiencias oscuras. Estos modelos lograron una precisión de hasta el 99 % con tasas de falsos positivos inferiores al 0,5 %, y requerimientos mínimos adicionales en la capacidad de almacenamiento, utilizando los conjuntos de datos NSL-KDD y CICIDS2017. Los autores determinaron que los modelos de aprendizaje continuo basados en repetición superan a las técnicas estadísticas tradicionales y a los modelos DNN, resultando especialmente eficaces en el manejo del problema del cambio de covariables.

En la educación, es crucial relacionarse con la tecnología de manera segura desde una edad temprana, ya que la exposición a la red sin los controles necesarios puede resultar riesgosa. Como destacan Villegas, Jácome, García, Calvache y Sánchez [9], para realizar un cambio tecnológico efectivo, es fundamental considerar la comodidad de los usuarios y su percepción del proceso. Contar con un entorno en el cual la interacción con la tecnología sea natural puede tener un impacto positivo en el desarrollo cognitivo de niños y adolescentes. La familiarización temprana con la tecnología no solo facilita la adaptación a entornos digitales en el futuro, sino que también promueve habilidades de resolución de problemas, creatividad y pensamiento crítico, preparando a los estudiantes para enfrentar los desafíos del mundo moderno. Apoyar esta relación con sistemas de seguridad adecuados, ayuda a asegurar la experiencia de los usuarios y aumenta su confianza en la tecnología.

Tener conectividad en una institución educativa secundaria es fundamental debido al avance tecnológico actual. El internet se convierte en una herramienta esencial que complementa el ámbito educativo. Sin embargo, la creciente dependencia tecnológica también incrementa la vulnerabilidad a ciberataques. Por ello, es crucial implementar sistemas de seguridad robustos. Robalino [10], en la Unidad Educativa “Bolívar” ubicada en la ciudad de Ambato, destaca la importancia de una buena conectividad de red como complemento para una experiencia académica óptima. Para asegurar esta infraestructura, se consideraron parámetros clave como escalabilidad, eficiencia, seguridad, costo, tiempo de implementación y facilidad de administración. Se destacan las metodologías PPDIOO de Cisco y Top-Down Network Design, entre otras. En la Unidad Educativa “Bolívar”, se optó por la metodología PPDIOO de Cisco Inc., que

mejor se adaptó a las necesidades estudiadas, subrayando que un diseño adecuado es fundamental para una implementación exitosa de infraestructura de red y la seguridad de los datos. Complementariamente, ayudándose también de la metodología PPDIOO de Cisco se puede integrar un Sistema de Detección de Intrusos (IDS) basado en machine learning que pueda proporcionar una capa adicional de seguridad, detectando patrones anómalos y potencialmente maliciosos en la red, lo que resulta vital para proteger la integridad y disponibilidad de los recursos educativos en línea.

1.3 Fundamentación teórica

1.3.1 Tecnologías de la Información y Comunicación (TIC)

Se puede considerar como las Tecnologías de la Información a las herramientas que suelen ser utilizadas para la administración y distribución de la información a través de dispositivos IoT [11].

En el contexto del ámbito educativo en Ecuador, las TIC han avanzado de manera exponencial, Bravo y Quezada [12] mencionan la transformación pedagógica en la cual se integran las Tecnologías de la Información y Comunicación como herramientas que facilitan el acceso, producción y difusión de información, enriqueciendo así la experiencia estudiantil. Según la LOEI en el Artículo 6j, es responsabilidad del Estado "garantizar la alfabetización digital y el uso de las tecnologías de la información y comunicación en el proceso educativo". Por lo tanto, la actualización curricular del 2016 reconoce que las TIC son herramientas habituales que facilitan el desarrollo del currículo y deben ser utilizadas por docentes y estudiantes. Esta actualización no considera que el uso de las TIC deba enseñarse como una asignatura independiente, sino que se integra de manera transversal, tratándola como una herramienta que facilita el aprendizaje, apoya la búsqueda crítica de información, el análisis y la comunicación de experiencias y conclusiones sobre los temas estudiados, y contribuye a la construcción del conocimiento y al aprendizaje autónomo.

Por ello, las TIC han perdido relevancia en el entorno educativo ecuatoriano. Al no ser una asignatura obligatoria, muchas instituciones optaron por relegarlas, lo que ha resultado en un analfabetismo tecnológico en niños y adolescentes. La emergencia

provocada por el coronavirus evidenció las consecuencias de descuidar las Tecnologías de la Información y Comunicación, ya que resaltaron las deficiencias tecnológicas de los establecimientos.

1.3.2 Seguridad Informática y Ciberseguridad Avanzada

En la actualidad, interactuar con la red se ha vuelto una tarea cotidiana y sencilla. Grandes empresas almacenan información valiosa en la nube, implementando las medidas de seguridad necesarias. Sin embargo, las personas comunes suelen confiar únicamente en los sistemas de seguridad predeterminados, ignorando la alta probabilidad de vulnerabilidad que enfrentan al estar dentro de una red de bajo nivel de seguridad. Un claro ejemplo son las redes educativas, que son de libre acceso y pueden ser fácilmente comprometidas por atacantes que buscan robar información valiosa de los usuarios. Morales y Zambrano [13] destacan que la información bancaria es una de las categorías más críticas en términos de seguridad de redes. En redes abiertas, las personas pueden realizar transacciones bancarias, pero al no ser redes totalmente confiables, esos datos pueden ser interceptados y utilizados para fines ilegales. Proteger las redes de ataques requiere el uso de sistemas avanzados que integren machine learning e inteligencia artificial. Sin embargo, para defenderse eficazmente, es esencial primero conocer y entender las amenazas a las que se enfrenta:

Malware: El término "malware" proviene del inglés "Malicious Software" y se refiere a cualquier programa o código que, al ejecutarse, causa daños en los sistemas. Para protegerse de este tipo de ataques, se pueden emplear métodos avanzados. Como exponen Rodríguez, Sidorov y Escamilla [14], las técnicas de Inteligencia Artificial (AI, por sus siglas en inglés) se han aplicado a diversas áreas del conocimiento para resolver problemas que no habían podido ser solucionados mediante métodos tradicionales. Gracias a los avances y desarrollos de algoritmos supervisados y no supervisados (Machine Learning), ha sido posible emplear estas técnicas para abordar una variedad de problemas, incluyendo la detección y mitigación de malware.

DDoS: Los ataques de Denegación de Servicio (DoS) tienen como objetivo agotar los recursos de un sistema informático, como el ancho de banda o el tiempo de procesamiento, provocando así la interrupción temporal o permanente de servicios

como internet, bases de datos o sitios web. Estos ataques funcionan mediante la realización de múltiples solicitudes simultáneas a la computadora o al equipo vulnerable, comprometiendo así los atributos de seguridad del sistema. Los protocolos más susceptibles a este tipo de ataques incluyen SMTP, DNS y NTP [15].

Ransomware: El ransomware es un tipo de software malicioso que secuestra, bloquea, elimina y/o modifica los datos de la computadora de la víctima, exigiendo un pago para su liberación, permitiendo así que la víctima pueda retomar sus actividades. Este tipo de ataque tiene casi siempre un motivo monetario. A diferencia de otros ataques, la víctima es notificada a través de algún medio electrónico sobre la intrusión y se le proporciona información detallada sobre los pasos a seguir para recuperarse del ataque [16].

El DNS spoofing: Es un método utilizado para redirigir a los usuarios de un sitio web a otro diferente al registrado para ese nombre de dominio. Este tipo de ataque puede basarse en dos variantes: el envenenamiento de la caché y el ID spoofing [17].

El envenenamiento de la caché es la forma más común de este tipo de suplantación y se lleva a cabo modificando la información contenida en un servidor de nombres de dominio. Cualquier servidor de nombre de domino (DNS, por sus siglas en inglés) que no esté protegido específicamente contra este tipo de ataques es vulnerable a ser comprometido [17].

El ID spoofing, por otro lado, implica hacer que la víctima crea que la máquina del atacante es un servidor DNS legítimo. Este método requiere capturar el número de identificación del datagrama User Datagram Protocol (UDP) que contiene una solicitud DNS inicial. Por ejemplo, un atacante puede hacer que las víctimas de un sitio web crean que están conectadas a otro sitio cuando, en realidad, están siendo redirigidas al sitio controlado por el atacante, atacante.com quisiera que las víctimas de victima1.com creyesen que están conectadas a victima2.com cuando en realidad están conectadas a atacante.com [17].

Phishing: El phishing es una técnica que combina la ingeniería social con exploits técnicos y tiene como objetivo persuadir a una víctima para que proporcione información personal, generalmente con el fin de obtener ganancias monetarias por

parte del atacante. La mayoría de los ataques de phishing se originan a través del envío de correos electrónicos fraudulentos que contienen enlaces de Localizador uniforme de recursos, del inglés Uniform Resource Locator (URL) falsos. Estos enlaces conducen a sitios web fraudulentos diseñados para parecerse a sitios web legítimos. A pesar de la atención significativa que ha recibido a lo largo de los años, todavía no existe una solución definitiva para combatir este tipo de ataque [18].

Para llevar a cabo un ataque de phishing, los atacantes suelen utilizar dos métodos principales que las personas no suelen percibir fácilmente como ataques: páginas web específicamente diseñadas para phishing que imitan a páginas reales, y correos electrónicos falsos. Estos métodos continúan siendo las principales formas de intrusión en las empresas.

Man-in-the-middle (MitM): Un atacante se interpone entre dos partes que intentan comunicarse para interceptar sus mensajes o datos. Como analogía física, sería como alguien abriendo tu correo, viendo tu información y luego cerrando el sobre sin que te des cuenta. Los objetivos principales de estos ataques suelen ser obtener datos bancarios o credenciales de inicio de sesión, ya que representan una oportunidad lucrativa. Dado que estos ataques suelen ocurrir en tiempo real, a menudo pasan desapercibidos hasta que es demasiado tarde. Para ejecutar con éxito este ataque, el atacante debe realizar dos pasos: primero, interceptar la red para obtener los datos, y luego descifrarlos. En la fase de interceptación, el atacante puede interferir con la red, por ejemplo, configurando un falso punto de acceso Wi-Fi o ingresando a la red del sujeto. Una vez entre medio, el atacante actúa como intermediario entre el usuario y la web a la que está tratando de conectarse, permitiendo que el usuario piense que está accediendo al sitio genuino cuando, en realidad, está viendo todos los datos intercambiados en tiempo real. Luego, en la segunda fase del ataque, el atacante procede a desencriptar estos datos, los cuales inicialmente están encriptados [19].

1.3.3 Seguridad en Redes Informáticas mediante Inteligencia Artificial

Los desafíos contemporáneos en la protección de datos son cada vez más complejos, con tácticas de vulneración de seguridad que se perfeccionan constantemente, poniendo en riesgo la información almacenada en la red. Ante estas amenazas en

evolución, la atención se ha volcado hacia la inteligencia artificial, un campo que ha experimentado un crecimiento significativo en los últimos años. Las características inherentes a la inteligencia artificial permiten el desarrollo de sistemas inteligentes que complementan la seguridad de la información. Para comprender mejor la integración de la inteligencia artificial en la seguridad, es fundamental familiarizarse con ciertos conceptos clave:

Inteligencia Artificial (IA): La IA ha transformado el panorama tecnológico actual, llevando a muchas organizaciones a buscar la automatización de sus procesos mediante esta innovadora herramienta. Definida como la rama de la informática que se enfoca en desarrollar algoritmos y sistemas capaces de realizar tareas que normalmente requerirían inteligencia humana, como reconocimiento de patrones, toma de decisiones y resolución de problemas [20].

Seguridad de la Información: La seguridad de la información abarca la protección de los sistemas y los datos de una organización contra amenazas y riesgos de seguridad. En entornos de IoT, donde hay numerosos dispositivos conectados y una gran cantidad de datos generados, la seguridad de la información adquiere una importancia aún mayor. Dentro del ámbito de la detección y clasificación de software malicioso, este concepto implica resguardar los sistemas y la información de una organización ante posibles riesgos y amenazas, considerando el constante incremento de amenazas de malware y la complejidad de los ataques cibernéticos [20].

Impacto de la IA en la Seguridad de la Información: La IA se ha empleado para mejorar la seguridad de la información mediante la detección de amenazas en tiempo real, lo que permite tomar medidas inmediatas ante intrusiones. Además, no solo actúa reactivamente ante las amenazas presentes, sino que también puede anticipar y prevenir futuros ataques mediante el análisis y la predicción de patrones de intrusión. Esta capacidad de aprendizaje continuo es fundamental, aunque su implementación requiere un conocimiento avanzado y puede afectar la privacidad de los datos. A pesar de sus beneficios, la confiabilidad de la IA puede ser cuestionada, ya que interpreta y clasifica las amenazas de forma automática, lo que podría conducir a errores y afectar la percepción de su eficacia por parte de los usuarios [20].

1.3.4 Sistema de detección de intrusos (IDS) basado en machine learning

La importancia de los Sistemas de detección de intrusos (IDS) en la actualidad es extremadamente alta debido a los constantes riesgos de vulnerabilidad en los sistemas. Desarrollar e implementar IDS potentes es crucial para contrarrestar los intentos de intrusión. Una forma eficaz de mejorar la capacidad de estos sistemas es combinarlos con técnicas de machine learning, permitiendo que los IDS detecten anomalías en la red de manera autónoma. Sudyana, Lin, Verkerken, Hwang, Lai, D'hooge, Wauters, Volckaert y De Turck [21] investigan un sistema IDS asistido por machine learning con el objetivo de mejorar la detección de problemas en la red. Su enfoque holístico busca mejorar la generalización del modelo utilizando un conjunto de datos basado en el ciclo de vida.

Investigaciones futuras pueden explorar el uso de múltiples fuentes de datos, como tráfico, contabilidad y Syslog, para la generalización de modelos tanto en clasificación binaria (benigna/maligna) como en multiclase (tipo de ataque). También sugieren investigar el rendimiento de la generalización mediante un proceso de aprendizaje automático en dos etapas: primero detectando la técnica y luego identificando el ciclo de vida en función de las secuencias de técnicas. Otra área potencial de exploración es mejorar la generalización del conjunto de datos CREMEv2 en relación con datos benignos, desarrollando un método integral para generar estos datos. Además, consideran la incorporación de técnicas de ataque y ciclos de vida adicionales del marco MITRE ATT&CK, con el fin de crear conjuntos de datos más grandes y diversos, mejorando la robustez del rendimiento de generalización del modelo de machine learning [21].

Por otro lado, Siddiqi y Pak [22] implementaron un sistema de detección de intrusiones en la red (NIDS, por sus siglas en inglés) que utiliza técnicas de machine learning y deep learning, demostrando que los algoritmos de deep learning son altamente efectivos para identificar patrones de comportamiento normales y anormales en una red. Aprovechando los avances en el procesamiento de imágenes, los expertos en seguridad están explorando la construcción de NIDS eficientes basados en esta tecnología. En su investigación, el marco propuesto sigue un enfoque de tres niveles

para generar una representación refinada y mejorada del conjunto de datos NIDS no basados en imágenes. Este marco reduce la cantidad de funciones necesarias, logrando un bajo nivel computacional con alta precisión. Además, el proceso de selección de características normaliza los datos para mejorar la interpretación de los modelos basados en deep learning. Aunque en el procesamiento de imágenes una mayor resolución puede significar mayor precisión, el marco propuesto optimiza el balance entre eficiencia computacional y precisión en la detección.

Integrar técnicas avanzadas como el machine learning en los sistemas de detección de intrusiones puede proporcionar importantes beneficios para la seguridad de las infraestructuras de red. En lugar de simplemente memorizar patrones, los algoritmos de machine learning aprenden, lo que mejora la precisión en la identificación de comportamientos anómalos. Este enfoque adaptable es crucial para enfrentar de manera efectiva las amenazas cibernéticas en constante cambio. Además, el machine learning facilita la detección de amenazas emergentes y desconocidas mediante el análisis continuo y la actualización de sus modelos de comportamiento. Esto no solo fortalece la capacidad de respuesta inmediata ante posibles ataques, sino que también disminuye el riesgo de falsos positivos, optimizando los recursos de seguridad. La implementación de técnicas de machine learning en los sistemas de detección de intrusiones marca un avance significativo hacia la creación de infraestructuras de red más resilientes y proactivas, capaces de anticipar y neutralizar ataques antes de que causen daños graves. Esta evolución en la seguridad informática es esencial para proteger datos sensibles y mantener la integridad de los sistemas en un entorno digital cada vez más complejo y amenazado.

1.3.5 Transformación Digital en el Sector Educativo

La transformación digital en el sector educativo demandas políticas claras que aborden las tecnologías a adoptar, sus características y objetivos, enfocándose en responder a las demandas tecnológicas y pedagógicas específicas de cada país y región. Esta iniciativa no solo implica una mera adopción de tecnología, sino que requiere una visión integral que tenga en cuenta cómo estas herramientas pueden enriquecer el proceso educativo. Por lo tanto, es esencial que las políticas educativas

contemplen no solo el acceso a la tecnología, sino también su integración efectiva en el aula para mejorar la calidad de la enseñanza y orientarla hacia la formación de ciudadanos del futuro [23].

Para lograr una transformación digital exitosa, es crucial aprovechar el impulso generado por la virtualización de emergencia durante la pandemia y recuperar los aprendizajes obtenidos. Esto implica ir más allá de la mera digitalización de contenidos y mantener una visión pedagógica en todas las dimensiones de la enseñanza, desde la planificación de las clases hasta la evaluación del aprendizaje. Además, se debe invertir en mejorar la infraestructura y el acceso a las tecnologías, reduciendo la brecha digital y garantizando un acceso equitativo a la educación virtual para todos los estudiantes, independientemente de su ubicación geográfica o nivel socioeconómico [23].

Otro aspecto fundamental para una transformación digital efectiva en el sector educativo es la actualización de los currículos de formación docente para integrar conocimientos en el uso didáctico de herramientas digitales y jerarquizar el trabajo docente hacia el rol de facilitador del aprendizaje personalizado. Además, es crucial pensar en las tecnologías a utilizar en función de su rol como futuros ciudadanos y trabajadores, desarrollando habilidades digitales, cognitivas y socioemocionales [23].

1.3.6 Gestión de Recursos Tecnológicos en Instituciones Educativas

La incorporación de las tecnologías digitales en la educación abre nuevas dimensiones y posibilidades en los procesos de enseñanza-aprendizaje. Facilita la individualización y flexibilización de la instrucción, motivando a los estudiantes y superando las limitaciones de tiempo y espacio. Sin embargo, para aprovechar plenamente estas herramientas, es fundamental definir la relación entre educación y gestión del conocimiento. Esto implica ir más allá de la transmisión de información y enfocarse en formar graduados autosuficientes, capaces de gestionar eficazmente la información en un entorno cambiante [24].

La gestión del conocimiento en las instituciones educativas se ve influenciada por el desarrollo tecnológico, especialmente por las TIC. Estas tecnologías transforman el rol

del docente, pasando de ser un transmisor autoritario de conocimiento a ser un facilitador del aprendizaje y la gestión del conocimiento en un campo específico. Además, las TIC ofrecen nuevas oportunidades para mejorar la calidad del trabajo educativo y científico, al permitir una mayor eficiencia en la gestión de la información y el conocimiento. En resumen, la gestión de recursos tecnológicos en las instituciones educativas es fundamental para adaptarse a la sociedad del conocimiento y garantizar una educación de calidad en un mundo cada vez más digitalizado [24].

Optimización del Ancho de Banda en Redes Escolares: En una red de conexión, disponer de una configuración eficiente para optimizar el ancho de banda es fundamental. Esto contribuirá significativamente a mejorar la experiencia del usuario al navegar por la red. Un aspecto crucial para lograr esto es el uso de Quality of Service (QoS) junto con un Sistema de detección de intrusos (IDS) ya que al tener una red configurada que se adapte a las necesidades y se complemente con un sistema de seguridad fuerte, ayudará a mitigar problemas de ralentización en la red que pueden ser provocados por intentos de ataques o intrusiones. Para optimizar el ancho de banda, se pueden considerar diversas técnicas:

Técnica de doble capa de sustrato: Esta técnica implica agregar una segunda capa de sustrato sobre la capa original, creando una configuración de doble capa. La segunda capa se utiliza para ajustar la constante dieléctrica efectiva ($\varepsilon eff = \varepsilon r\varepsilon 0$) de la estructura de la antena y así aumentar su ancho de banda. Una ventaja de esta técnica es su fácil implementación sin necesidad de modificar la geometría de la antena. Sin embargo, puede ser difícil de diseñar correctamente, ya que la capa superior puede afectar la impedancia y el diagrama de radiación de la antena [25].

Técnica de parche con perturbaciones: Esta técnica consiste en añadir pequeñas perturbaciones al parche radiante de la antena, actuando como resonadores adicionales que incrementan su ancho de banda. Estas perturbaciones pueden tener diversas formas, como ranuras, cortes o discontinuidades. Su ubicación y forma pueden optimizarse mediante simulación y pruebas experimentales para obtener los mejores resultados en términos de ancho de banda y eficiencia de la antena. Aunque es fácil de implementar y puede proporcionar un ancho de banda considerablemente mayor que otras técnicas, las perturbaciones pueden afectar negativamente la impedancia y el

diagrama de radiación de la antena, requiriendo una optimización cuidadosa del diseño [25].

Técnica de tapas de resonancia: Esta técnica implica agregar dos o más capas de material dieléctrico sobre el plano de tierra de la antena parche, actuando como resonadores adicionales que aumentan su ancho de banda [25].

Técnica de segmentación: Se basa en dividir el parche radiante de la antena en varias partes, creando así resonadores adicionales que aumentan su ancho de banda. Cada segmento funciona como una antena individual, y su combinación produce un ancho de banda más amplio que una antena parche convencional [25].

1.3.7 Control de la Red en la Unidad Educativa

La implementación del Sistema de Gestión de Seguridad de la Información (SGSI, por sus siglas en inglés) en entornos educativos, específicamente en el control de la red en una unidad educativa, se apoya en metodologías reconocidas como MAGERIT. Esta metodología proporciona un marco estructurado para el análisis y gestión de riesgos de seguridad de la información [26].

Al aplicar MAGERIT, se pueden identificar y evaluar los riesgos asociados con la red de la institución educativa, considerando elementos como el Modelo de Valoración de Activos (MOVA), la Identificación de Amenazas (MARI), el Análisis de Escenarios (DEAP), la Evaluación de Vulnerabilidades (EVSA) y la Evaluación de Riesgos Residuales (ESRI). Estos componentes permiten comprender mejor las amenazas potenciales, evaluar su impacto en la seguridad de la red y tomar medidas adecuadas para mitigar los riesgos [26].

Así, al aplicar el enfoque de MAGERIT en el control de la red en una unidad educativa, se puede establecer un SGSI robusto que proteja los activos de información, garantice la continuidad de las operaciones y promueva un entorno seguro y propicio para el aprendizaje y la administración [26].

Para poder entender mejor la forma del control de la red, hay que conocer un poco más del enfoque MAGERIT. MAGERIT, es un marco común utilizado para analizar y

gestionar los riesgos asociados con las aplicaciones informáticas. Desarrollado y promovido por el gobierno español, MAGERIT es de uso libre y se adapta a estándares internacionales como ISO/IEC 27000/27001 (Sistemas de Gestión de la Seguridad de la Información), ISO/IEC 27005 (Gestión de Riesgos de la Seguridad de la Información) e ISO 31000 (Sistema de Gestión de Riesgos). MAGERIT proporciona un enfoque estructurado para la identificación, análisis y tratamiento de los riesgos de seguridad de la información en los sistemas informáticos. Su aplicación ayuda a las organizaciones a comprender mejor las amenazas potenciales, evaluar su impacto en la seguridad de la información y tomar medidas adecuadas para mitigar los riesgos. Es una herramienta valiosa para instituciones y empresas que buscan proteger sus activos de información y garantizar la continuidad de sus operaciones en un entorno cada vez más digital y conectado [27].

1.4 Objetivos

1.4.1 Objetivo general

Implementar un Sistema de detección de intrusos (IDS) basado en machine learning para el control de la red en la Unidad Educativa “19 de Septiembre” en la ciudad de Salcedo.

1.4.2 Objetivos específicos

- Evaluar riesgos potenciales en la infraestructura de red de la Unidad Educativa “19 de Septiembre” que pueden afectar su control y seguridad.
- Investigar algoritmos de machine learning para la detección de intrusos, considerando su eficacia y adaptabilidad para la red en la Unidad Educativa “19 de Septiembre”.
- Documentar los procedimientos para implementar un IDS basado en machine learning en la red de la Unidad Educativa “19 de Septiembre”, garantizando seguridad y estabilidad.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

Para llevar a cabo la recolección de datos en el presente proyecto, se emplearon diversas herramientas que facilitaron una mejor comprensión del panorama en torno a la seguridad de la red en la "Unidad Educativa 19 de Septiembre". Entre los mecanismos utilizados, se incluyó una entrevista dirigida al rector de la institución y al docente encargado de los laboratorios de computación, así como una encuesta enfocada en la población que tiene mayor interacción con la red. Finalmente, se analizó la tolerancia de la red ante ataques cibernéticos mediante la técnica de observación.

2.1.1 Entrevista al rector y docente encargado de los laboratorios de computación de la "Unidad Educativa 19 de Septiembre"

La entrevista realizada al rector de la unidad educativa y al docente encargado de los laboratorios de computación tuvo como objetivo profundizar en la gestión de la seguridad de la información en el establecimiento. Para llevar a cabo la entrevista, se utilizó el formato descrito en la Tabla 4.

Tabla 4. Ficha de entrevista

Datos personales **Institución:** **Apellido y Nombre:** **Cargo:**			
Nº	**Pregunta**	**Respuesta**	**Observación**
1	**¿Qué tan importante considera el uso de la red para las actividades académicas de la institución?**		
2	**¿Qué tan efectivo cree que es el sistema de seguridad actual de la red?**		
3	**¿Qué tan frecuente han sido los incidentes de seguridad en la red en los últimos años?**		
4	**¿Qué tan probable considera que la red sea vulnerada por ataques externos en su estado actual?**		

Nº	Pregunta	Respuesta	Observación
5	¿Qué tan dispuesto/a está la institución para invertir en la mejora de la seguridad de la red?		
6	¿Qué tan necesario cree que es capacitar al personal sobre el uso de nuevas tecnologías de seguridad?		
7	¿Qué tan confiado/a se sentiría si la red estuviera protegida por un sistema IDS?		
8	¿Qué impacto espera que tenga la implementación de un sistema IDS en la reputación de la institución?		
9	¿Qué tan importante cree que es contar con un sistema de monitoreo continuo para detectar amenazas en tiempo real?		
10	¿Qué tan frecuente se revisan y actualizan las políticas de seguridad de la institución?		
11	¿Considera que los estudiantes y profesores están suficientemente informados sobre los riesgos de seguridad en la red?		
12	¿Qué medidas adicionales cree que deberían tomarse para mejorar la seguridad en la red institucional?		
13	¿Qué tan importantes son las alianzas con otras instituciones o empresas para fortalecer la seguridad tecnológica?		
14	¿Qué tan probable es que el presupuesto actual de la institución permita la implementación de un sistema de detección de intrusos?		
15	¿Qué tan probable es que se necesiten ajustes significativos en la red para implementar un sistema de detección de intrusos?		
Conclusiones:			

2.1.2 Encuesta dirigida a la población con mayor interacción con la red

Con el fin de obtener un mayor contexto sobre la percepción de la seguridad de la red, así como de conocer con mayor detalle los incidentes que han afectado a los usuarios frecuentes, se decidió emplear una encuesta compuesta por 15 preguntas, estructurada en formato de escala de Likert.

A continuación, se detallan las preguntas empleadas en el cuestionario:

1. **¿Con qué frecuencia existen fallas en la red durante actividades académicas?**

- 1: Nunca
- 2: Ocasionalmente
- 3: A veces
- 4: Frecuentemente
- 5: Siempre

2. **Según su criterio, ¿Qué tan importante es la red para sus actividades académicas diarias?**

- 1: Nada importante
- 2: Poco importante
- 3: Moderadamente importante
- 4: Importante
- 5: Muy importante

3. **¿Con qué frecuencia experimenta desconexiones inesperadas?**

- 1: Nunca
- 2: Ocasionalmente
- 3: A veces
- 4: Frecuentemente
- 5: Siempre

4. **¿Cuánto le preocupa la posibilidad de que alguien pueda acceder a su información personal a través de la red de la institución?**

- 1: Nada preocupado
- 2: Poco preocupado
- 3: Normal
- 4: Preocupado
- 5: Muy preocupado

5. **¿Qué tan consciente es de los riesgos de seguridad que implica conectarse a la red institucional?**

- 1: Nada consciente
- 2: Poco consciente
- 3: Normal
- 4: Consciente
- 5: Muy consciente

6. **¿Qué tan dispuesto/a estaría a aceptar restricciones de acceso a la red si eso mejora su seguridad?**

- 1: Nada dispuesto
- 2: Poco dispuesto
- 3: Moderadamente dispuesto
- 4: Dispuesto
- 5: Muy dispuesto

7. **¿Qué tan importante cree que es la seguridad de la red para la protección de la información personal?**

- 1: Nada importante
- 2: Poco importante
- 3: Moderadamente importante
- 4: Importante
- 5: Muy importante

8. **¿Qué tan frecuente es que deba buscar alternativas (como usar datos móviles) debido a problemas en la red institucional?**

- 1: Nunca
- 2: Ocasionalmente
- 3: A veces
- 4: Frecuentemente
- 5: Siempre

9. **¿Qué tan grave considera que son los problemas de conexión en la red de la institución?**

- 1: Nada grave
- 2: Poco grave
- 3: Moderadamente grave
- 4: Grave
- 5: Muy grave

10. **¿Qué tan útil considera que sería monitorear la red más estrictamente para evitar intrusiones?**

- 1: Nada útil
- 2: Poco útil
- 3: Moderadamente útil
- 4: Útil
- 5: Muy útil

11. **¿Qué tan necesario considera recibir notificaciones sobre la seguridad de la red y posibles amenazas?**

- 1: Nada necesario
- 2: Poco necesario
- 3: Moderadamente necesario
- 4: Necesario
- 5: Muy necesario

12. **¿Qué tanto ha escuchado sobre los sistemas de detección de intrusos (IDS)?**

- 1: Nada
- 2: Poco
- 3: Moderado
- 4: Muy moderado
- 5: Mucho

13. **¿Qué tan seguro/a se sentiría sabiendo que la red está protegida por un sistema de detección de intrusos (IDS)?**

- 1: Nada seguro
- 2: Poco seguro
- 3: Normal
- 4: Seguro
- 5: Muy seguro

14. **¿Qué tan necesario considera la implementación de un sistema para detectar intrusos para la protección de la red en la institución?**

- 1: Nada necesario
- 2: Poco necesario
- 3: Necesario
- 4: Moderadamente necesario
- 5: Muy necesario

15. **¿Qué tan interesado/a estaría en recibir más información sobre los sistemas de detección de intrusos (IDS) y su funcionamiento?**

- 1: Nada interesado
- 2: Poco interesado
- 3: Moderadamente interesado
- 4: Interesado
- 5: Muy interesado

2.1.3 Observación de la tolerancia de la red al recibir ataques cibernéticos

Para evaluar la tolerancia de la red frente a diversos ataques en una infraestructura de conexión, se empleó la técnica de observación. El objetivo fue analizar el comportamiento de la infraestructura y determinar los límites que puede soportar sin experimentar fallos. Como referencia para este análisis, se utilizó el modelo descrito en la Tabla 5.

Tabla 5. Modelo de ficha de observación.

Fecha:	
Lugar:	
Observador:	
Descripción general del entorno:	
Herramientas utilizadas:	
Instrucción:	En las escalas, resaltar la opción que concuerde con lo observado durante la ejecución de los ataques.
Objetivo de la observación:	Analizar el comportamiento de la red de la Unidad Educativa "19 de Septiembre" ante diferentes ataques controlados, con el fin de identificar vulnerabilidades, tiempos de respuesta, estabilidad y posibles interrupciones que puedan comprometer su seguridad y rendimiento.

Tipo de ataque	**Descripción**	**Escala (1-5)**	**Observaciones**
DDoS (Denegación de servicio distribuida)	Se genera un tráfico masivo de datos hacia la red para sobrecargar los recursos y hacer que los servicios sean inaccesibles.	1 Red muy inestable	
		2 Red inestable	
		3 Red moderadamente estable	
		4 Red estable	
		5 Red muy estable	
Escaneo de puertos	Exploración de los puertos de la red para detectar aquellos abiertos y posibles vulnerabilidades.	1 Ningún puerto vulnerable	
		2 Pocos puertos vulnerables	
		3 Algunos puertos vulnerables	
		4 Varios puertos vulnerables	
		5 Muchos puertos vulnerables	
Spoofing de IP	Ataque en el que el atacante suplanta una dirección IP legítima para engañar a los sistemas de la red y obtener acceso.	1 Ninguna dirección IP sospechosa detectada	
		2 Pocas direcciones IP sospechosas detectadas	
		3 Moderada cantidad de direcciones IP sospechosas detectadas	
		4 Varias direcciones IP sospechosas detectadas	
		5 Muchas direcciones IP sospechosas detectadas	
Ataque de fuerza bruta	Intento repetido de adivinar contraseñas o credenciales para obtener acceso no autorizado.	1 Ningún intento fallido	
		2 Pocos intentos fallidos	
		3 Moderados intentos fallidos	
		4 Varios intentos fallidos	
		5 Muchos intentos fallidos	
Man-in-the-Middle (MITM)	Intercepción de comunicaciones entre dos partes para robar o manipular la información en tránsito.	1 Sin intercepción de datos	
		2 Poca intercepción de datos	
		3 Moderada intercepción de datos	
		4 Considerable intercepción de datos	
		5 Alta intercepción de datos	

Observaciones adicionales	
Conclusiones:	

2.2 Métodos

2.2.1 Modalidad de la investigación

La investigación se centró en un enfoque de investigación de campo y bibliográfico-documental.

a. Investigación de campo

Se realizó una investigación de campo, lo que conllevó al investigador a obtener datos de manera inmediata mediante mecanismos como encuestas y entrevistas. Estas actividades se realizaron dentro del área geográfica previamente delimitada, lo que permitió una recopilación de información directamente relevante para el estudio en cuestión.

b. Investigación bibliográfica – documental

El trabajo de investigación abarcó un enfoque bibliográfico-documental, ya que se contrastó información en revistas, artículos, libros, tesis, entre otros recursos, relacionados con los temas expuestos. Se consideró especialmente la información relevante para la selección de algoritmos de inteligencia artificial eficaces en la implementación del Sistema de detección de intrusos (IDS) basado en machine learning, con el objetivo de alcanzar una mejor alineación con los objetivos establecidos.

2.2.2 Población y muestra

Para la recolección de datos, se trabajó con una muestra compuesta por estudiantes de 15 años en adelante, ya que, según Graciano [28], los adolescentes entre 15 y 17 años suelen contar con un teléfono celular propio, lo que les facilita un mayor acceso a internet y, en consecuencia, los expone a mayores riesgos de seguridad en la red. Pérez [29], en su estudio, identificó que la edad promedio de los estudiantes de décimo año de Educación General Básica oscila entre los 14 y 15 años. Por este motivo, en la presente investigación se incluyó a la población desde décimo año de Educación

General Básica hasta el tercer año de Bachillerato General Unificado y el tercer año de Bachillerato Técnico, ya que el uso frecuente y la mayor facilidad de acceso a internet que se presenta en estos grupos los hace particularmente relevantes para el estudio de vulnerabilidades en la red. Asimismo, se consideró al personal que utiliza regularmente el laboratorio de computación, como docentes, personal de consejería estudiantil y personal administrativo de la Unidad Educativa "19 de Septiembre". En la Tabla 6 se detalla el desglose de la población seleccionada.

Tabla 6. Población de estudio.

Población	Número	Porcentaje
Administrativos	5	0.55%
Consejería Estudiantil	2	0.22%
Docentes	33	3.63%
Estudiantes de 10º año de Educación General Básica a 3º año de Bachillerato	868	95.59%
Total	908	100%

a. Muestra

Dado que la población total excedió significativamente el estándar de 100 individuos para una muestra, se utilizó la formulación de muestra finita para calcular la muestra de estudiantes. Dado que se trata de un muestreo probabilístico, la selección de individuos se realizó de forma aleatoria.

Para el cálculo de la muestra se utilizó la fórmula:

$$n = \frac{NZ^2PQ}{Ne^2 + Z^2PQ}$$

En dónde:

N = Tamaño de la población de estudiantes equivalente a 868.

e = Error estándar 9% equivalente a 0.09.

Z = Nivel de confianza 95% equivalente a 1.96.

P = Probabilidad de éxito equivalente a 0.5.

Q = Probabilidad de fracaso equivalente a 0.5.

n = Tamaño de la muestra.

$$n = \frac{(868)(1.96)^2(0.5)(0.5)}{(868)(0.09)^2 + (1.96)^2(0.5)(0.5)} = 100.10$$

El tamaño ideal de la muestra corresponde a 100 estudiantes.

En cuanto al resto de los integrantes de la población, como el personal directivo, el personal de consejería estudiantil y el personal docente, cada uno cuenta con una población inferior a 100 personas. Por lo tanto, se incluyó a toda la población en el estudio. En total, se consideraron 139 individuos, sobre los cuales se aplicaron los mecanismos de recolección de información, como se indica en Tabla 7.

Tabla 7. Muestra de estudio.

Población	Número	Porcentaje
Administrativos	5	3.59%
Consejería Estudiantil	2	1.43%
Docentes	32	23.02%
Estudiantes	100	71.94%
Total	139	100%

2.2.3 Recolección de información

La recolección de información se realizó mediante diversas técnicas. En primer lugar, se aplicó una encuesta dirigida a la población en general, compuesta por 15 preguntas, cuyo objetivo fue la percepción sobre la conectividad y la seguridad de la red en la Unidad Educativa "19 de Septiembre". Además, se realizaron entrevistas al rector de la institución y al docente encargado del laboratorio de computación, para de conocer la administración de las políticas de seguridad y obtener detalles sobre incidentes previos, así como la forma en que estos son percibidos por los usuarios. Finalmente, se utilizó la técnica de observación para analizar la tolerancia de la red a ciertos ataques y la reacción de los usuarios cuando sus sistemas fueron vulnerados.

Validación del instrumento

Para asegurar una mayor confiabilidad en los datos recolectados, se optó por emplear el coeficiente alfa de Cronbach. Esta medida estadística permite evaluar la consistencia interna de los ítems que componen los instrumentos de recolección de información, como en este caso, la encuesta aplicada a la población con la que se trabajó.

Para el cálculo del alfa de Cronbach se aplica la siguiente fórmula:

$$\propto = \frac{K}{K-1}\left[1 - \frac{\sum Si^2}{St^2}\right]$$

Dónde:

K = Número de ítems del instrumento de recolección de la información

$\sum Si^2$ = Sumatoria de la varianza de los ítems del instrumento de recolección de la información

St^2 = Varianza de la sumatoria de los ítems del instrumento de recolección de la información

Para poder optimizar el proceso del cálculo del coeficiente alfa de Cronbach se recurrió a la ayuda de un algoritmo creado en Python como se puede visualizar en figura 1.



Figura 1. Algoritmo para el cálculo Alfa de Cronbach.

Al identificar individualmente las cantidades que componen la fórmula de Alfa de Cronbach se obtuvo:

$$\propto = \frac{15}{15-1}\left[1 - \frac{18.48}{72.86}\right] = 0.80$$

Tabla 8. Escalas de confiabilidad de Alfa de Cronbach

Ítem	Valor
Mayor 0.90	Excelente
0.80 – 0.89	Buena
0.70 – 0.79	Aceptable
0.60 – 0.69	Deficiente

El coeficiente Alfa de Cronbach obtenido fue de 0.80, lo cual, de acuerdo con la escala presentada en la Tabla 8, se clasifica dentro de un rango considerado como "bueno". Este resultado indica que la información recolectada mediante la herramienta de la entrevista presenta un alto grado de confiabilidad.

c. Resultados de la encuesta aplicada a la población seleccionada

Pregunta 1:

¿Con qué frecuencia existen fallas en la red durante actividades académicas?

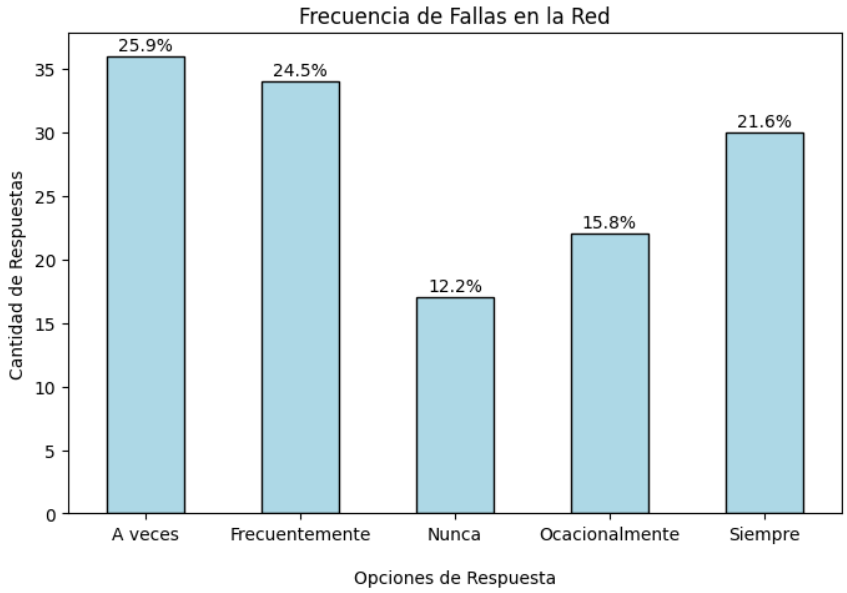


Figura 2. Resultados pregunta 1.

Interpretación de resultados

La información recolectada sugiere que las fallas en la red son un problema que afecta constantemente a la mayoría de los usuarios de la institución. Según los datos, el 25.9 % de los encuestados señaló que los problemas ocurren "A veces", el 24.5 % indicó que suceden "Frecuentemente", y el 21.6 % mencionó que pasan "Siempre". Esto significa que más del 70 % de los usuarios experimentan algún nivel de inestabilidad en la conexión. Solo el 12.2 % dijo que "Nunca" han enfrentado fallas, lo cual resalta la magnitud del problema. Este panorama refleja serias deficiencias en la infraestructura de red, que podrían deberse a configuraciones inadecuadas, como una

sobrecarga de usuarios, o a la falta de monitoreo adecuado. Además, no se puede descartar vulnerabilidades en la seguridad, entre las que se puede incluir posibles ataques como los de denegación de servicio o accesos no autorizados, estén afectando la estabilidad. Este escenario genera incomodidad y descontento entre los usuarios y representa un desafío que necesita atención urgente para garantizar una red estable y confiable, especialmente en un entorno académico que depende tanto de una buena conexión para sus actividades diarias.

Pregunta 2:

Según su criterio, ¿Qué tan importante es la red para sus actividades académicas diarias?

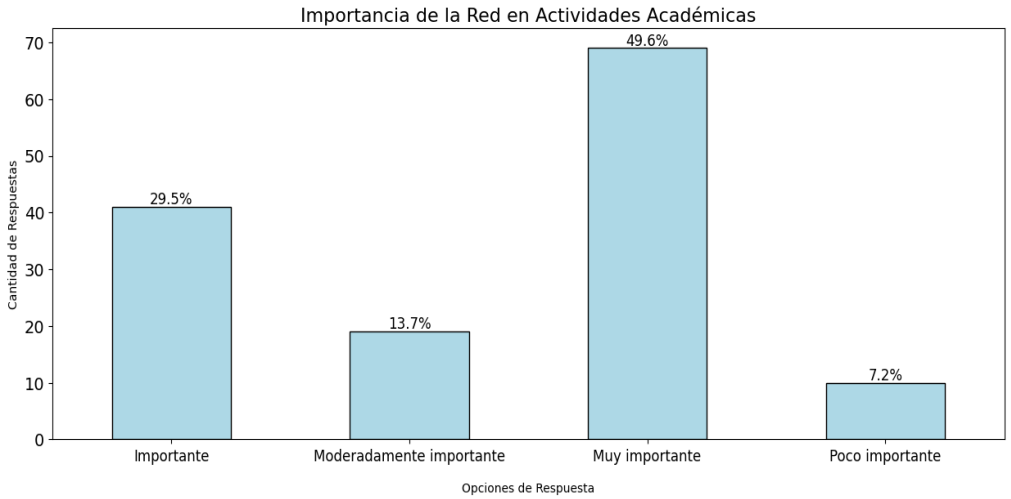


Figura 3. Resultados pregunta 2.

Interpretación de resultados

Los datos muestran que la mayoría de los encuestados consideró que la red es fundamental para sus actividades académicas. El 49.6 % calificó su importancia como "Muy importante", mientras que el 29.5 % la describió como "Importante". Esto significa que casi el 80 % de las personas reconoce la relevancia de la conectividad en el ámbito educativo. Por otro lado, un 13.7 % considera que la red es "Moderadamente importante" y solo el 7.2 % opina que es "Poco importante". Estos resultados evidencian cómo la tecnología se ha convertido en una herramienta indispensable en el aprendizaje diario. Garantizar una red estable y segura no solo facilita el acceso a

recursos en línea, sino que también contribuye a un proceso educativo más dinámico y efectivo. Además, es esencial promover la capacitación en el uso adecuado de estas herramientas tecnológicas para maximizar su impacto positivo en el entorno académico.

Pregunta 3:

¿Con qué frecuencia experimenta desconexiones inesperadas?

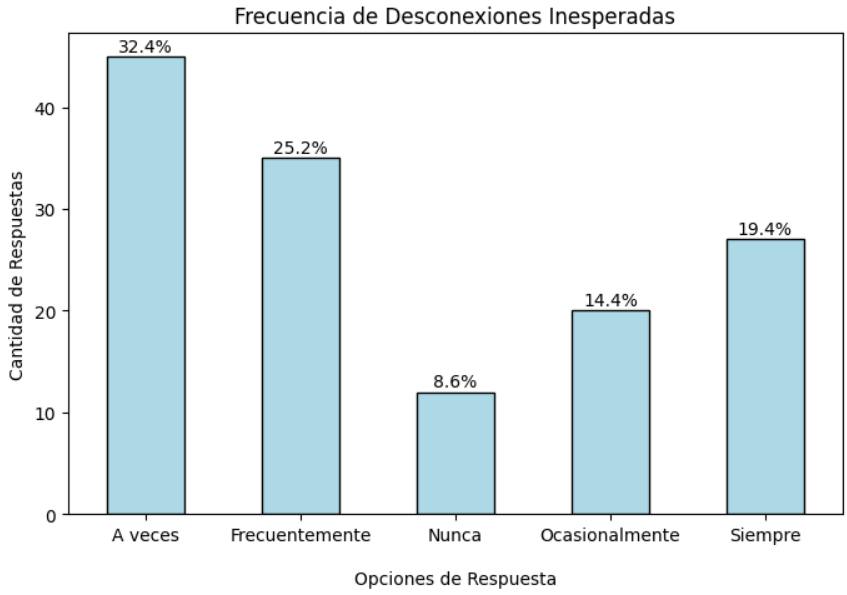


Figura 4. Resultados pregunta 3.

Interpretación de los resultados

Los resultados revelan que las desconexiones inesperadas son un problema frecuente en la institución, ya que el 32.4 % de los encuestados indicó que ocurren "A veces", el 25.2 % señaló que suceden "Frecuentemente" y el 19.4 % afirmó que ocurren "Siempre", lo que suma un preocupante 77 % de respuestas que reflejan recurrencia. Por otro lado, un 14.4 % mencionó que estas desconexiones se presentan "Ocasionalmente", mientras que solo el 8.6 % indicó que "Nunca" han enfrentado este problema. Estos datos reflejan una deficiencia importante en la infraestructura de red, que afecta no solo la conectividad en momentos críticos, sino también la percepción de los usuarios, quienes podrían considerar la tecnología como poco confiable e ineficiente. Este panorama refuerza los hallazgos de la primera pregunta, donde se

destacó la inestabilidad de la red durante actividades académicas. La prevalencia de estas desconexiones subraya la necesidad urgente de implementar mejoras en la estabilidad y confiabilidad de la red, para garantizar un entorno educativo adecuado y alineado con las demandas de la era digital.

Pregunta 4:

¿Cuánto le preocupa la posibilidad de que alguien pueda acceder a su información personal a través de la red de la institución?

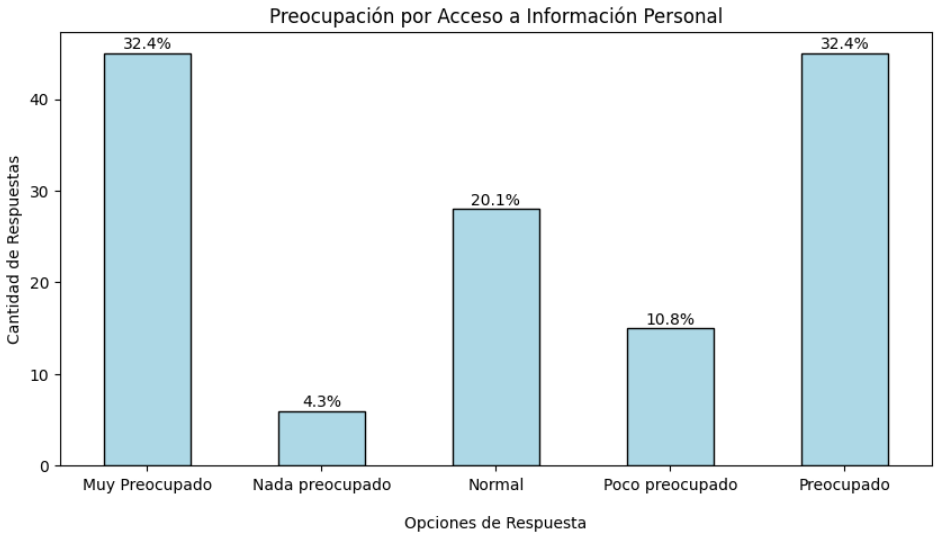


Figura 5. Resultados pregunta 4.

Interpretación de los resultados

Los resultados reflejan una alta preocupación entre los usuarios sobre la posibilidad de que su información personal sea vulnerada al utilizar la red de la institución. En total, el 64.8 % de los encuestados indicó estar "Muy preocupado" (32.4 %) o "Preocupado" (32.4 %), lo que evidencia una percepción generalizada de inseguridad en la red. Un 20.1 % expresó una preocupación "Normal", mientras que un 10.8 % afirmó estar "Poco preocupado". En contraste, solo un 4.3 % de los encuestados dijo no estar preocupado en absoluto, lo que podría deberse al desconocimiento de los riesgos que implica la conexión a internet o a que estos usuarios prefieren no utilizar la red institucional para proteger su información personal. Estos datos subrayan la importancia de mejorar las medidas de seguridad en la red, ya que la percepción de

vulnerabilidad no solo afecta la confianza de los usuarios, sino que también podría limitar su uso de las herramientas digitales disponibles, impactando negativamente en el entorno académico.

Pregunta 5:

¿Qué tan consciente es de los riesgos de seguridad que implica conectarse a la red institucional?

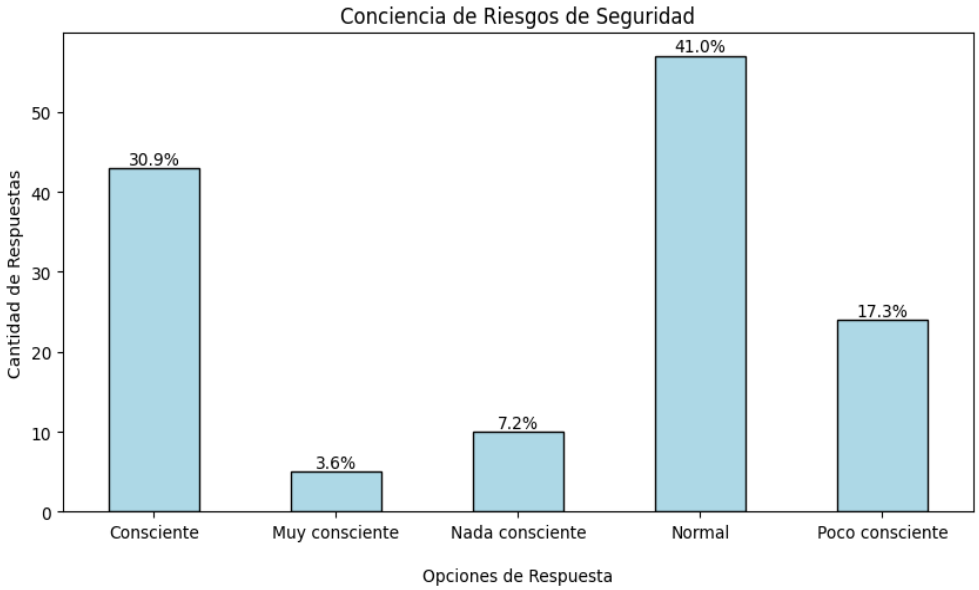


Figura 6. Resultados pregunta 5.

Interpretación de los resultados

La mayoría de los usuarios muestran un nivel moderado de conciencia sobre los riesgos de seguridad al conectarse a la red institucional, con un 41.0 % ubicado en la categoría "Normal" y un 30.9 % en "Consciente", lo que suma un 71.9 % que tiene cierta noción de los riesgos. Sin embargo, aún existe un 17.3 % que se considera "Poco consciente", un 7.2 % "Nada consciente", y solo un 3.6 % que se identifica como "Muy consciente". Esto evidencia la necesidad de fortalecer la educación y sensibilización sobre seguridad informática, ya que la falta de una mayor conciencia puede limitar la confianza en la red institucional y, en consecuencia, afectar la adopción de herramientas tecnológicas esenciales para el aprendizaje. Incrementar el conocimiento de los usuarios sobre estos riesgos no solo mejorará la percepción de seguridad, sino

que también permitirá integrar de manera más efectiva las tecnologías en los procesos educativos, contribuyendo a un entorno académico más seguro y productivo.

Pregunta 6:

¿Qué tan dispuesto/a estaría a aceptar restricciones de acceso a la red si eso mejora su seguridad?

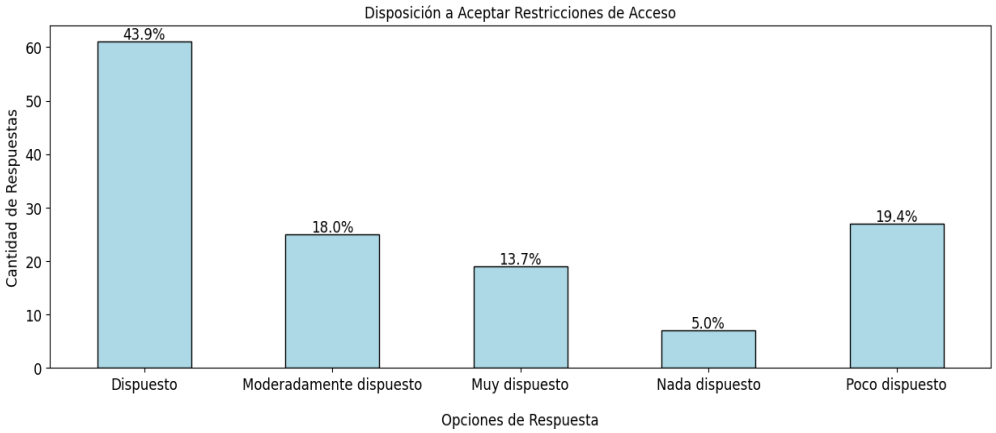


Figura 7. Resultados pregunta 6.

Interpretación de los resultados

La información evidencia que el 43.9 % de los encuestados mostró interés en aceptar restricciones de acceso a la red si esto mejora la seguridad de sus datos personales, mientras que un 18.0 % se muestra "Moderadamente dispuesto" y un 13.7 % "Muy dispuesto", lo que indica que una mayoría significativa está abierta a estas medidas. Sin embargo, un 19.4 % señaló estar "Poco dispuesto" y un 5.0 % "Nada dispuesto", evidenciando cierta resistencia, posiblemente asociada a la percepción de pérdida de control sobre su libertad de navegación. Esto resalta la necesidad de campañas de concientización enfocadas en los beneficios de implementar restricciones como parte de una estrategia de seguridad, mostrando cómo estas pueden reducir riesgos sin comprometer excesivamente la experiencia del usuario. Fomentar esta comprensión podría mejorar la aceptación general y garantizar un equilibrio entre seguridad y accesibilidad en la red institucional.

Pregunta 7:

¿Qué tan importante cree que es la seguridad de la red para la protección de la información personal?

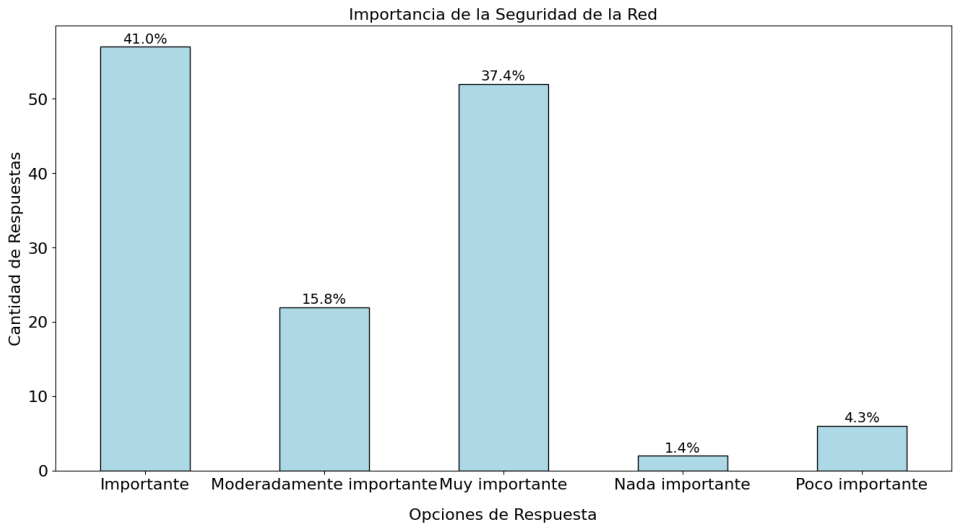


Figura 8. Resultados pregunta 7.

Interpretación de los resultados

Los resultados reflejan que la seguridad de la red es percibida como un aspecto crucial por la mayoría de los usuarios. El 41.0 % de los encuestados considera que es "Importante" y el 37.4 % cree que es "Muy importante", lo que suma un 78.4 % que valora la protección de la información personal como una prioridad. Un 15.8 % la clasifica como "Moderadamente importante", mientras que solo un 4.3 % y un 1.4 % opinan que es "Poco importante" o "Nada importante", respectivamente. Este alto nivel de conciencia demuestra que los usuarios comprenden los riesgos asociados con la falta de medidas de seguridad adecuadas en la red institucional. Estos datos destacan la necesidad de implementar estrategias efectivas para fortalecer la protección de la información, ya que la confianza de los usuarios depende en gran medida de la percepción de seguridad. Además, la falta de atención a esta preocupación podría limitar el uso de la red, impactando negativamente en el desempeño académico y administrativo, y exponiendo a los usuarios a riesgos innecesarios. Es fundamental que la institución priorice la seguridad digital para garantizar un entorno tecnológico seguro y confiable.

Pregunta 8:

¿Qué tan frecuente es que deba buscar alternativas (como usar datos móviles) debido a problemas en la red institucional?

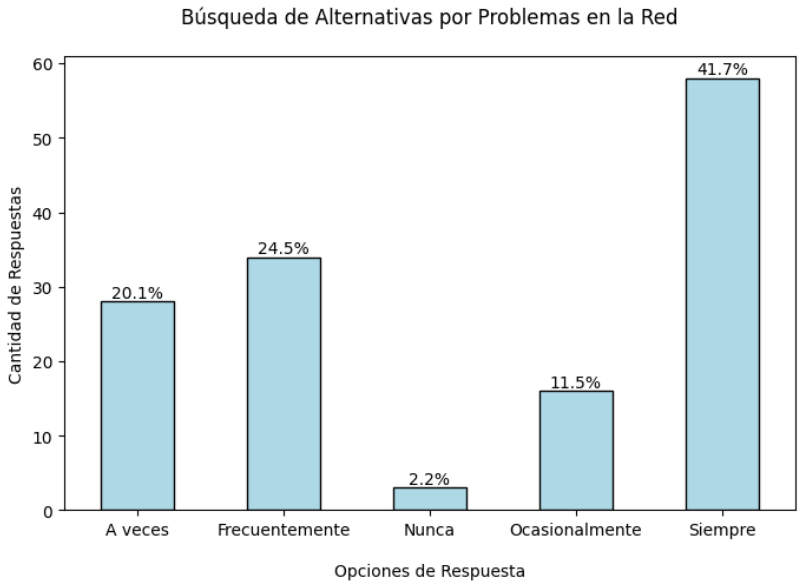


Figura 9. Resultados pregunta 8.

Interpretación de los resultados

La mayoría de los usuarios han tenido que recurrir a alternativas, como el uso de datos móviles, debido a problemas en la red institucional. El 41.7 % indicó que "Siempre" deben buscar alternativas, mientras que un 24.5 % lo hace "Frecuentemente" y un 20.1 % "A veces". Solo un 11.5 % señaló que recurre a estas medidas "Ocasionalmente", y apenas un 2.2 % afirmó que "Nunca" ha tenido esta necesidad. Estos datos reflejan una situación preocupante, ya que la dependencia de soluciones externas como los datos móviles puede implicar costos adicionales para los estudiantes y el personal, generando un impacto económico significativo. Además, esta problemática transforma el acceso a internet en un privilegio más que en una herramienta esencial para el aprendizaje y el desempeño académico. La falta de una red confiable en la institución no solo limita la accesibilidad, sino que también afecta la equidad educativa, al dificultar el acceso pleno a recursos en línea indispensables para estudiantes y docentes. Es crucial implementar mejoras en la infraestructura de la red para garantizar que todos los usuarios puedan disfrutar de un servicio estable y accesible.

Pregunta 9:

¿Qué tan grave considera que son los problemas de conexión en la red de la institución?

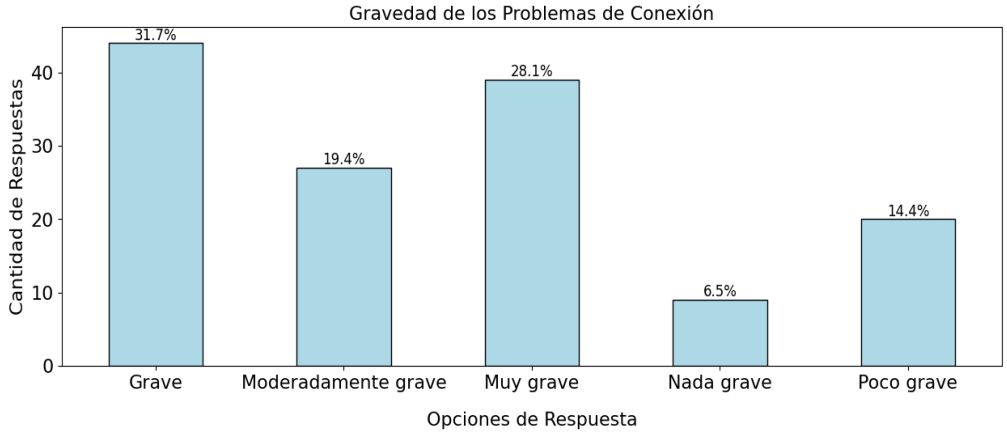


Figura 10. Resultados pregunta 9.

Interpretación de los resultados

Los resultados indican que la mayoría de la población encuestada percibe los problemas de conexión en la red institucional como significativos, ya que un 31.7 % los califica como "Grave" y un 28.1 % como "Muy grave", lo que suma un 59.8 % que considera esta situación de alta relevancia. Un 19.4 % de los participantes los percibe como "Moderadamente grave", mientras que un 14.4 % los considera "Poco grave" y solo un 6.5 % opina que no son "Nada grave". Esta percepción refleja un alto nivel de insatisfacción entre los usuarios, lo cual está alineado con los resultados de preguntas anteriores, donde se destacó la frecuencia de interrupciones y desconexiones. La constante inestabilidad de la conexión no solo afecta el desarrollo de las actividades académicas de estudiantes y docentes, sino que también fomenta una percepción negativa y una desconfianza generalizada hacia la red institucional. Estos datos subrayan la urgencia de abordar los problemas de conectividad, ya que garantizar una red confiable es esencial para el éxito académico y la confianza en las herramientas tecnológicas de la institución.

Pregunta 10:

¿Qué tan útil considera que sería monitorear la red más estrictamente para evitar intrusiones?

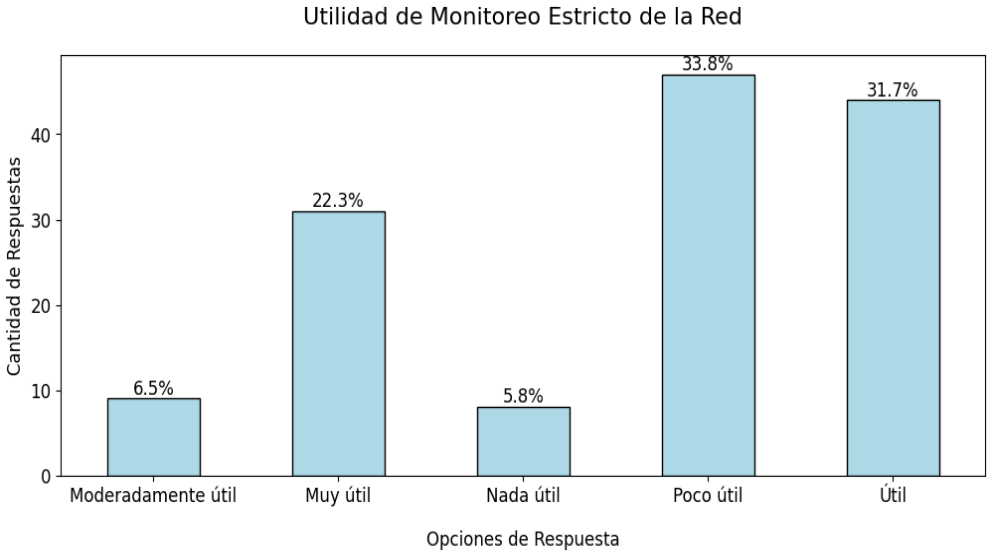


Figura 11. Resultados pregunta 10.

Interpretación de los resultados

La percepción sobre la utilidad de un monitoreo estricto de la red para prevenir intrusiones está claramente dividida entre los encuestados. Un 22.3 % cree que sería "Muy útil", mientras que un 31.7 % lo considera "Útil", lo que refleja un reconocimiento parcial de su importancia. Por otro lado, un 33.8 % lo califica como "Poco útil", seguido de un 6.5 % que lo ve "Moderadamente útil" y un 5.8 % que lo percibe como "Nada útil". Esta diversidad de opiniones podría deberse a la preocupación de algunos usuarios sobre la posible invasión de privacidad que podría implicar un control estricto, lo cual genera cierta resistencia. Para enfrentar esta percepción, sería fundamental aclarar que el objetivo del monitoreo es garantizar la seguridad de la red y proteger los datos de los usuarios sin supervisar su actividad personal. Campañas informativas enfocadas en este aspecto podrían ayudar a reducir los temores y aumentar la aceptación de esta medida como una herramienta esencial para fortalecer la seguridad de la red institucional.

Pregunta 11:

¿Qué tan necesario considera recibir notificaciones sobre la seguridad de la red y posibles amenazas?

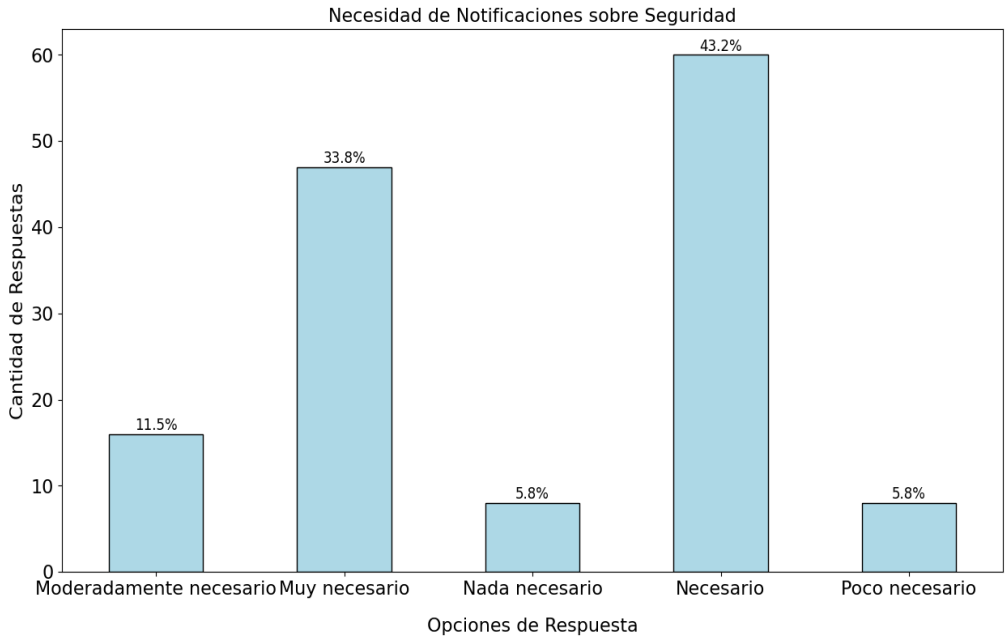


Figura 12. Resultado pregunta 11.

Interpretación de los resultados

La mayoría de los encuestados considera fundamental recibir notificaciones sobre la seguridad de la red y posibles amenazas, ya que un 43.2 % califica esta medida como "Necesaria" y un 33.8 % como "Muy necesaria", lo que suma un 77 % de respuestas que reflejan una fuerte demanda de información en este ámbito. En contraste, un 11.5 % lo considera "Moderadamente necesario", mientras que solo un 5.8 % señala que es "Poco necesario" o "Nada necesario". Este interés por mantenerse informados refleja una preocupación latente por la seguridad personal y la protección de sus datos, así como una percepción de que los mecanismos actuales de la red no son completamente confiables. Al estar al tanto de posibles riesgos, los usuarios podrían tomar medidas preventivas, como desconectarse temporalmente de la red, hasta que las amenazas sean neutralizadas.

Pregunta 12:

¿Qué tanto ha escuchado sobre los sistemas de detección de intrusos (IDS)?

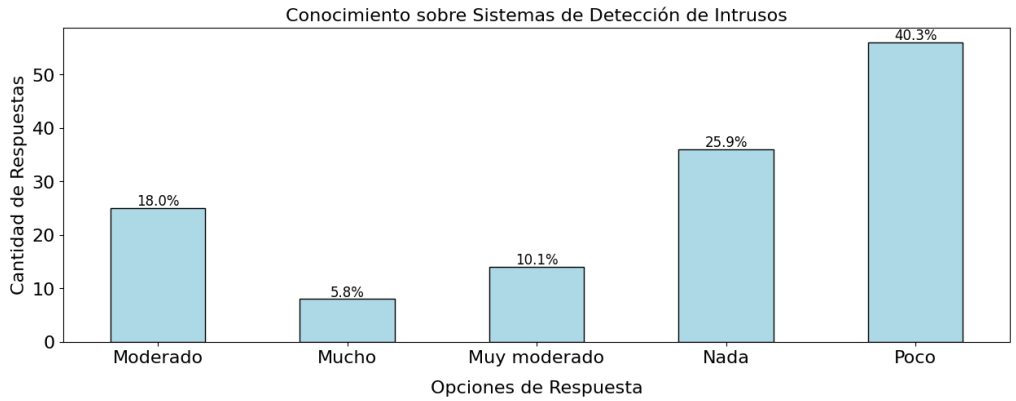


Figura 13. Resultados pregunta 12.

Interpretación de los resultados

La mayoría de los encuestados tiene un conocimiento limitado sobre los sistemas de detección de intrusos (IDS), ya que un 40.3 % señaló haber escuchado "Poco" sobre ellos y un 25.9 % indicó "Nada", lo que suma un 66.2 % que muestra escasa familiaridad con estas tecnologías. Solo un 18.0 % indicó tener un conocimiento "Moderado", un 10.1 % señaló "Muy moderado" y apenas un 5.8 % afirmó tener un conocimiento significativo, según la opción "Mucho". Estos datos evidencian una falta generalizada de conciencia sobre herramientas clave para la seguridad informática. Esta falta de familiaridad no solo subraya la necesidad de capacitar a los usuarios, sino que también puede estar alimentando la percepción de inseguridad en la red, ya que muchos desconocen las medidas que podrían estar protegiendo sus datos. Informar y educar a la comunidad sobre los beneficios y el funcionamiento de los IDS podría ser crucial para aumentar la confianza en la red institucional y fomentar una mayor colaboración en su uso seguro.

Pregunta 13:

¿Qué tan seguro/a se sentiría sabiendo que la red está protegida por un sistema de detección de intrusos (IDS)?

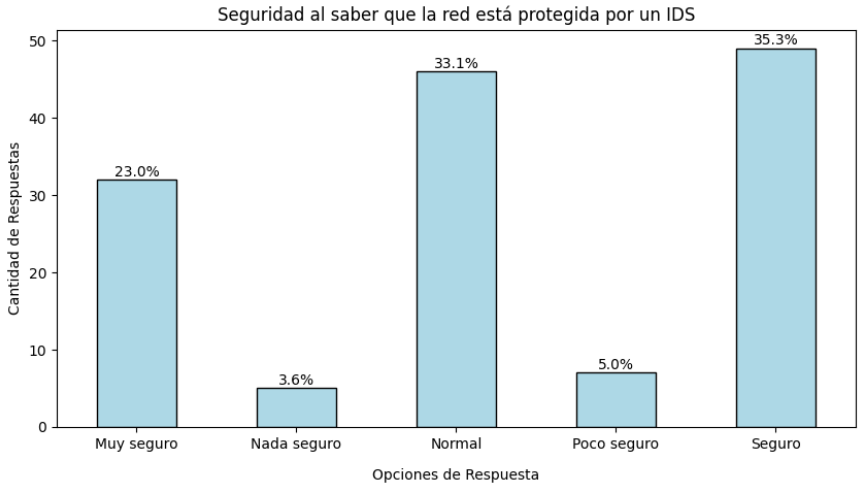


Figura 14. Resultados pregunta 13.

Interpretación de resultados

La percepción de seguridad al saber que la red está protegida por un sistema de detección de intrusos (IDS) muestra opiniones divididas. Un 35.3 % de los encuestados afirmó sentirse "Seguro", mientras que un 33.1 % eligió la opción neutral "Normal". Por otro lado, un 23.0 % expresó que se sentiría "Muy seguro", lo que resalta cierta confianza en este tipo de protección. Sin embargo, un 5.0 % señaló estar "Poco seguro" y un 3.6 % afirmó estar "Nada seguro", evidenciando que aún persiste cierta incertidumbre. Este balance podría estar influenciado por el desconocimiento general sobre los IDS reflejado en la pregunta anterior, lo que dificulta una valoración precisa de su efectividad. Promover la comprensión y el conocimiento sobre estas tecnologías podría ayudar a reducir la sensación de inseguridad y aumentar la confianza de los usuarios, fomentando un entorno digital más protegido y confiable dentro de la institución.

Pregunta 14:

¿Qué tan necesario considera la implementación de un sistema para detectar intrusos para la protección de la red en la institución?

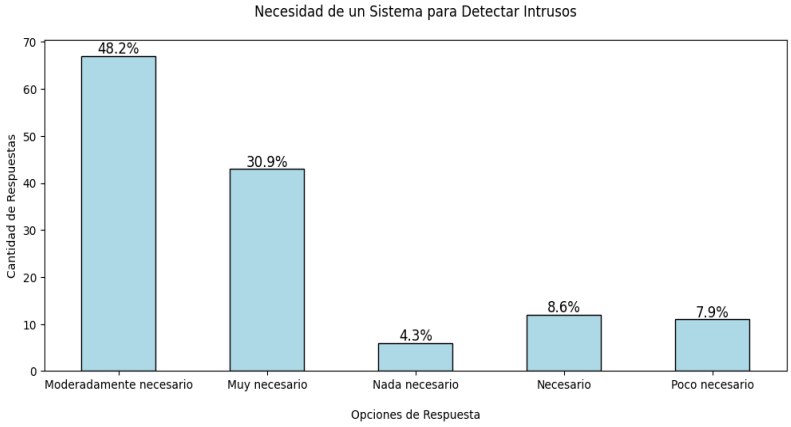


Figura 15. Resultados pregunta 14.

Interpretación de los resultados

Los usuarios muestran un reconocimiento moderado hacia la necesidad de implementar un sistema de detección de intrusos (IDS) para proteger la red de la institución. El 48.2 % de los encuestados calificó esta medida como "Moderadamente necesario", seguido por un 30.9 % que la consideró "Muy necesario". Sin embargo, el 20.8 % restante se distribuye entre las categorías "Necesario" (8.6 %), "Poco necesario" (7.9 %) y "Nada necesario" (4.3 %), lo que refleja que aún existe una proporción significativa de usuarios que no perciben esta tecnología como esencial o prioritaria. Este panorama podría estar influenciado por la falta de conocimiento detallado sobre los beneficios de los sistemas de detección de intrusos, tal como se observó en los resultados de la pregunta 12. Además, algunos usuarios podrían sentirse seguros con las medidas actuales de protección, lo que refuerza la necesidad de campañas informativas que expliquen el valor añadido de un IDS en términos de seguridad y protección de datos. Sensibilizar a los usuarios sobre cómo estas herramientas pueden fortalecer la infraestructura tecnológica de la institución es clave para incrementar la percepción de su importancia y fomentar una aceptación más amplia de su implementación.

Pregunta 15:

¿Qué tan interesado/a estaría en recibir más información sobre los sistemas de detección de intrusos (IDS) y su funcionamiento?

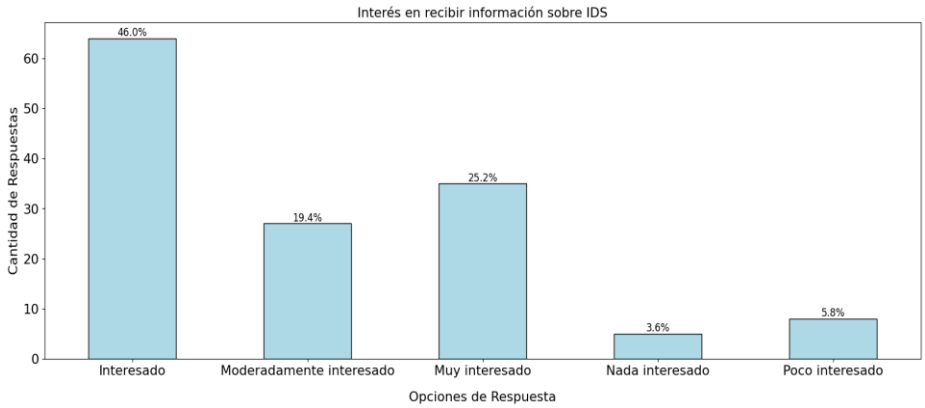


Figura 16. Resultados pregunta 15.

El interés en aprender más sobre los sistemas de detección de intrusos (IDS) es notablemente alto entre los encuestados. Un 46 % manifestó estar "Interesado" en recibir más información, mientras que un 25.2 % se declaró "Muy interesado", lo que suma un 71.2 % de la población con una actitud positiva hacia este tema. Por otro lado, un 19.4 % indicó estar "Moderadamente interesado", y solo un pequeño porcentaje, correspondiente al 5.8 % y 3.6 %, señaló estar "Poco interesado" o "Nada interesado", respectivamente. Estos datos reflejan una creciente conciencia sobre la importancia de la ciberseguridad en el ámbito educativo y una disposición significativa para profundizar en el conocimiento de estas tecnologías. Para aprovechar este interés, sería valioso organizar talleres o sesiones informativas que expliquen el funcionamiento y los beneficios de los IDS, incentivando así una mayor aceptación y comprensión de su rol en la protección de la red institucional. Esto no solo aumentaría el conocimiento técnico, sino que también fortalecería la confianza de los usuarios en las medidas de seguridad implementadas.

d. Resultado de las entrevistas aplicadas al rector y docente encargado del laboratorio de computación de la Unidad Educativa 19 de "Septiembre"

Tabla 9. Encuesta al rector de la Unidad Educativa “19 de Septiembre”.

Datos personales **Institución: Unidad Educativa “19 de Septiembre”** **Apellido y Nombre: Banda Mauricio** **Cargo: Rector**			
Nº	**Pregunta**	**Respuesta**	**Observación**
1	**¿Qué tan importante considera el uso de la red para las actividades académicas de la institución?**	Hoy en día, con la globalización del internet y los sistemas informáticos, la red es una herramienta fundamental para la educación. Considero que el internet es esencial para que la educación avance a pasos agigantados.	
2	**¿Qué tan efectivo cree que es el sistema de seguridad actual de la red?**	Actualmente, el sistema de seguridad no es muy efectivo. Siempre existe la necesidad de ser cautelosos respecto a la infraestructura de la red, pero hasta ahora no se ha implementado un sistema de seguridad adecuado que contribuya a la protección de la red de la institución.	
3	**¿Qué tan frecuente han sido los incidentes de seguridad en la red en los últimos años?**	En los últimos años, hemos experimentado intentos de vulneración de datos de manera recurrente, tanto que hoy en día se ha convertido en un problema significativo.	Como adicional luego de la entrevista el rector de la institución comentó que entre los tipos de altercados están la filtración de los datos.
4	**¿Qué tan probable considera que la red sea vulnerada por ataques externos en su estado actual?**	Siempre estamos expuestos a todo tipo de ataques informáticos, especialmente externos, ya que existen muchas aplicaciones que facilitan este tipo de acciones. Debido a la rapidez con la que ocurren, siempre estamos en una posición vulnerable.	
5	**¿Qué tan dispuesto/a está la institución para invertir en la mejora de la seguridad de la red?**	En cuanto a la mejora de la red, la institución tiene prioridades en distintos sectores, pero considerando la importancia del área informática, existe la posibilidad de destinar recursos para reforzar la seguridad y la tecnología de la institución.	
6	**¿Qué tan necesario cree que es capacitar al personal sobre el uso de nuevas tecnologías de seguridad?**	Es sumamente importante capacitar al personal docente de manera inmediata, ya que hoy en día los docentes deben incluir en su planificación curricular lo que se conoce como horas TIC’s. Es esencial que estén capacitados al 100 % en el manejo de tecnología.	
7	**¿Qué tan confiado/a se sentiría si la red estuviera protegida por un sistema IDS?**	Nos sentiríamos muy confiados, ya que podríamos navegar por la red de manera segura al contar con un sistema formal que respalde la seguridad informática en la institución.	

Nº	Pregunta	Respuesta	Observación
8	**¿Qué impacto espera que tenga la implementación de un sistema IDS en la reputación de la institución?**	Se espera que tenga un impacto muy positivo en la institución, ya que contar con un respaldo de seguridad mejoraría la calidad del servicio de internet, es una solución que se necesita de manera inmediata.	
9	**¿Qué tan importante cree que es contar con un sistema de monitoreo continuo para detectar amenazas en tiempo real?**	Contar con un sistema de monitoreo continuo es crucial, ya que permitiría prevenir intentos de acceso no autorizado a la información de la institución. La monitorización debe llevarse a cabo de manera constante para garantizar la seguridad de la red.	
10	**¿Qué tan frecuente se revisan y actualizan las políticas de seguridad de la institución?**	En el año que llevo desempeñando funciones como rector de la institución, no se han establecido aún políticas de seguridad informática, y tampoco se ha tenido documentación o información de mandatos previos.	
11	**¿Considera que los estudiantes y profesores están suficientemente informados sobre los riesgos de seguridad en la red?**	No están completamente informados. Tanto los docentes como los estudiantes tienen un conocimiento vago sobre los riesgos asociados con la navegación en internet, lo cual es bastante limitado.	
12	**¿Qué medidas adicionales cree que deberían tomarse para mejorar la seguridad en la red institucional?**	La implementación de un sistema IDS sería la medida ideal para mejorar la seguridad de la red en la institución.	
13	**¿Qué tan importantes son las alianzas con otras instituciones o empresas para fortalecer la seguridad tecnológica?**	La seguridad tecnológica en la institución ha sido gestionada principalmente por el sector público, en este caso, CNT, a lo que es a nivel de zonas distritales. La alianza con ellos sería muy importante, ya que al manejar todo en el ámbito de la tecnología, se convertirían en un socio estratégico para fortalecer nuestra seguridad.	La única seguridad informática en la que se confía en la institución es la configuración que deja por defecto el personal de CNT, ya que no existe un verdadero control de la red como por ejemplo la segmentación de la misma, tanto así que existen dispositivos conectados al router principal sin tener por lo menos un etiquetado de dónde pertenecen.

Nº	Pregunta	Respuesta	Observación
14	**¿Qué tan probable es que el presupuesto actual de la institución permita la implementación de un sistema de detección de intrusos?**	La institución cuenta con equipos que pueden servir para la implementación de este tipo de sistemas. Como institución, estamos dispuestos a facilitar el uso de estos equipos para llevar a cabo la implementación del sistema de detección de intrusos.	En la institución educativa existen bienes que pueden ser utilizados para la mejora de la infraestructura de red, los cuales no se han empleado por falta de personal calificado, como por ejemplo servidores funcionales que se encontraban embodegados, así como switches y routers.
15	**¿Qué tan probable es que se necesiten ajustes significativos en la red para implementar un sistema de detección de intrusos?**	Se requiere un ajuste total, ya que la red actual es muy básica y hay aspectos que necesitan ser adaptados o configurados para su correcto funcionamiento.	La configuración de la red debe ser total ya que al solo tener la configuración por defecto por parte de la empresa que brinda el servicio de internet, las brechas en cuanto optimización y seguridad resultan considerables.
Conclusiones: Al entrevistar al rector de la institución, se pone en manifiesto una clara preocupación por la seguridad de la red en la institución educativa. El entrevistado destaca la necesidad urgente de mejorar la infraestructura tecnológica y considera importante la implementación de un sistema de detección de intrusos (IDS) para proteger la información. Además, existe un reconocimiento de la falta de capacitación en el uso de tecnologías de seguridad y un desconocimiento general sobre los riesgos asociados con la navegación en línea. La disposición para invertir en mejoras y establecer alianzas con proveedores tecnológicos es evidente, lo que sugiere un compromiso hacia la mejora continua en la seguridad informática de la institución.			

Tabla 10. Encuesta al docente encargado del laboratorio de computación de la Unidad Educativa “19 de Septiembre”.

Datos personales **Institución: Unidad Educativa “19 de Septiembre”** **Apellido y Nombre: Javier Reinoso** **Cargo: Docente encargado del laboratorio de computación**			
Nº	**Pregunta**	**Respuesta**	**Observación**
1	**¿Qué tan importante considera el uso de la red para las actividades académicas de la institución?**	En un mundo donde la tecnología avanza constantemente, la conexión a internet se vuelve crucial. Permite a los docentes generar nuevas oportunidades para que los estudiantes accedan no solo a plataformas virtuales, sino también a simuladores. La parte operativa se integra con el manejo curricular, lo que hace imprescindible contar con una red interna que facilite a docentes y estudiantes estar a la vanguardia y explorar nuevas metodologías para aplicar la tecnología en función del perfil curricular.	Como uno de los proyectos de integración curricular en la institución, es trabajar con simuladores online para que los estudiantes del área técnica puedan tener un mejor proceso de aprendizaje al practicar en un entorno simulado, actualmente no se lo realiza por distintos factores como: la inestabilidad de la red, mejor equipamiento de los laboratorios, etc.
2	**¿Qué tan efectivo cree que es el sistema de seguridad actual de la red?**	Actualmente, no contamos con un sistema de seguridad estructurado. Tanto docentes como estudiantes tienen libre acceso a la red. Si bien se ha intentado limitar el acceso a ciertas páginas que no contribuyen al desarrollo académico, estas restricciones son solo en casos puntuales. En general, no existe un mecanismo que limite de manera integral los riesgos que pueden afectar la parte operativa de la red institucional.	
3	**¿Qué tan frecuente han sido los incidentes de seguridad en la red en los últimos años?**	En la institución se utilizan dos sistemas: uno centralizado para matrículas y calificaciones, que cuenta con sus propias protecciones gestionadas por el rector. Sin embargo, el sistema interno ha sufrido vulneraciones en el pasado, lo que ha llevado a implementar medidas de respaldo, como la utilización de discos duros y copias físicas. A pesar de estas acciones, la seguridad del sistema interno no ha mejorado de manera significativa, ya que ha habido manipulación por parte de agentes externos, y aún no se cuenta con un plan para solventar ese tipo de casos.	
4	**¿Qué tan probable considera que la red sea vulnerada por ataques externos en su estado actual?**	En las condiciones actuales, la vulnerabilidad de la red supera el 100%. Cualquier persona con acceso, ya sea limitado o total, podría ingresar a la base de datos de estudiantes, calificaciones y portafolios docentes, ya que no se cuenta con un sistema de seguridad estricto ni con protocolos adecuados para controlar el acceso. Además, el error humano de algunos docentes también contribuye a esta situación, ya que, al no tener un buen dominio de la tecnología, a menudo no implementan contraseñas seguras o, en algunos casos, proporcionan acceso total, lo que expone aún más la red a riesgos.	Puede resultar fácil infiltrarse en la red ya que en las computadoras de los laboratorios tampoco se maneja la utilización mediante usuarios con acceso restringido, es decir todas las máquinas se ejecutan con privilegios administrativos, lo que hace fácil el conocer credenciales y acceder a la información en general.

Nº	Pregunta	Respuesta	Observación
5	**¿Qué tan dispuesto/a está la institución para invertir en la mejora de la seguridad de la red?**	La institución no maneja recursos de manera autónoma, ya que estos se gestionan a nivel de distrito. Esto dificulta la asignación de fondos para implementar programas de seguridad de forma general. La limitación económica es un desafío, pero se han hecho esfuerzos significativos a través de propuestas y proyectos de integración curricular. Estos proyectos, impulsados por los estudiantes, han permitido la generación de recursos internos. Por ejemplo, este año se ha propuesto el desarrollo de simuladores, lo cual se considera un primer paso hacia la apertura de nuevos proyectos. Cabe destacar que los padres de familia y los estudiantes han sido fundamentales en el apoyo a estas iniciativas.	Gracias a la gestión autónoma en la institución han logrado adquirir equipo de cómputo que todavía no se encuentra instalado por falta de personal capacitado y por desconocimiento para poder adaptar la topología de red actual.
6	**¿Qué tan necesario cree que es capacitar al personal sobre el uso de nuevas tecnologías de seguridad?**	Es fundamental capacitar al personal, ya que en la actualidad se utilizan herramientas tecnológicas no solo en el ámbito operativo, sino también en el didáctico. Este aspecto no se ha explorado adecuadamente en la institución. Los docentes están adoptando métodos de enseñanza que no se alinean con las posibilidades que ofrece la tecnología actual. Los estudiantes, quienes están familiarizados con el uso de la tecnología, pueden beneficiarse enormemente de métodos de enseñanza que incorporen juegos y simulaciones, lo que permitiría generar nuevos enfoques para el aprendizaje.	
7	**¿Qué tan confiado/a se sentiría si la red estuviera protegida por un sistema IDS?**	Me sentiría muy confiado, ya que esto permitiría iniciar con procesos de digitalización de documentos que facilitarían la agilidad en los trámites dentro de la institución. Además, tendría la certeza de que estos documentos estarían protegidos y no serían susceptibles a modificaciones no autorizadas.	
8	**¿Qué impacto espera que tenga la implementación de un sistema IDS en la reputación de la institución?**	Espero que tenga un impacto positivo, ya que nos permitiría digitalizar documentos de manera segura. Esto generaría confianza tanto en docentes como en estudiantes, asegurándoles que sus documentos no serán modificados. Además, podríamos incluir a los padres de familia en este proceso, para que tengan la certeza de que documentos, calificaciones y otros datos estarán en buenas manos, al igual que los documentos dirigidos a los docentes, como acuerdos y resoluciones.	
9	**¿Qué tan importante cree que es contar con un sistema de monitoreo continuo para detectar amenazas en tiempo real?**	Considero que es muy importante contar con un sistema de monitoreo continuo, especialmente durante la jornada laboral. Esto permitiría gestionar la seguridad de manera más eficaz, ya que el monitoreo podría realizarse las 24 horas del día. De esta forma, obtendríamos notificaciones que nos alertarían sobre posibles accesos no autorizados durante las horas no laborables.	

Nº	Pregunta	Respuesta	Observación
10	**¿Qué tan frecuente se revisan y actualizan las políticas de seguridad de la institución?**	Actualmente, no se cuenta con un manual de seguridad formal. Se ha propuesto generar un manual de buenas prácticas en los talleres, a partir del cual se desarrollarán temas relacionados con la seguridad. Inicialmente, se ha puesto énfasis en la seguridad física, pero aún no se han abordado aspectos de seguridad informática. Sin embargo, estamos trabajando en ello; especialmente en este periodo, se ha hecho hincapié en el desarrollo del manual, en el que se delimitarán los temas de seguridad.	
11	**¿Considera que los estudiantes y profesores están suficientemente informados sobre los riesgos de seguridad en la red?**	No, definitivamente no. Hubo un caso en el que se utilizaron las máquinas del laboratorio para subir un documento con calificaciones. Sin embargo, por descuido, cuando los estudiantes utilizaron esas mismas máquinas para rendir el examen "Ser Bachiller", se modificó este documento. Fue solo cuando la lista de calificaciones llegó al rectorado que se dieron cuenta de que el archivo había sido alterado. El docente que realizó el documento no sabía cómo guardar esta información de forma segura, por lo que la dejó expuesta. Este es solo un ejemplo de lo que ocurre con muchos docentes en la institución. Creo que es fundamental que se fomente el uso adecuado de la tecnología, pero sobre todo que se brinde capacitación para mejorar el desempeño en el ámbito tecnológico.	
12	**¿Qué medidas adicionales cree que deberían tomarse para mejorar la seguridad en la red institucional?**	Es un trabajo articulado. En primera instancia, entendemos que, si bien podemos contar con seguridad cibernética y archivos bien resguardados, es crucial establecer un enfoque integral que brinde seguridad a todos los ámbitos dentro de los laboratorios. Para lograr esto, la colaboración debe comenzar desde el laboratorio, involucrando a los docentes que lo utilizan y a los estudiantes que operan las máquinas. Todo este proceso permitirá centralizar la información; si esta información se gestiona de manera segura, contribuirá a que la institución se mantenga a la vanguardia tecnológica.	
13	**¿Qué tan importantes son las alianzas con otras instituciones o empresas para fortalecer la seguridad tecnológica?**	Las alianzas son fundamentales. Se generan proyectos de vinculación cuando ambas partes abren sus puertas a la colaboración. Trabajar con instituciones que puedan ayudar a abordar nuestras debilidades representa una alternativa muy valiosa. Si a través de convenios logramos dotar de material que permita poner en práctica los conocimientos adquiridos, eso beneficiará a ambas partes y fortalecerá nuestra seguridad tecnológica.	
14	**¿Qué tan probable es que el presupuesto actual de la institución permita la implementación de un sistema de detección de intrusos?**	El presupuesto es un factor limitante, ya que, como se mencionó, el ámbito económico no lo maneja directamente la institución. Sin embargo, existe la posibilidad de autogestión, lo cual nos permite avanzar más allá de los conceptos básicos y empoderarnos para ofrecer una educación activa. Para proyectos de esta naturaleza, se puede gestionar financiamiento a través de iniciativas de vinculación, donde tanto estudiantes como padres de familia jueguen un papel fundamental en el aspecto económico.	

Nº	Pregunta	Respuesta	Observación
15	**¿Qué tan probable es que se necesiten ajustes significativos en la red para implementar un sistema de detección de intrusos?**	Es muy probable que se requieran ajustes significativos, dado que actualmente no se cuenta con un sistema de detección de intrusos. La implementación de un sistema de este tipo permitirá evaluar el nivel de modificaciones necesarias en la red de la institución, asegurando su efectividad y funcionalidad en la protección de la información.	Al tener nuevo equipo listo para ser instalado pero que todavía aún no se lo hace ya que por falta de conocimiento no se ha podido adaptar la topología de red, implica que se genere una nueva configuración total que implique las nuevas máquinas y el servidor IDS.
Conclusiones: La entrevista con el docente encargado de los laboratorios de computación revela una comprensión clara de la importancia de la seguridad en la red y su impacto en las actividades académicas de la institución. A pesar de la dependencia de la tecnología para mejorar la enseñanza y el aprendizaje, se identifican debilidades significativas en la infraestructura de seguridad actual. La falta de un sistema de seguridad formal, junto con la escasa capacitación del personal sobre el uso de nuevas tecnologías de seguridad, pone en riesgo la integridad de la información y la confianza en el uso de la red. El docente enfatiza la necesidad de implementar un sistema de detección de intrusos (IDS) como una medida crucial para garantizar la seguridad de los datos y la privacidad de los estudiantes. Además, se destaca la importancia de realizar ajustes en la red y de establecer alianzas con otras instituciones para fortalecer la seguridad tecnológica. La autogestión y la colaboración con padres y estudiantes son vistas como estrategias clave para enfrentar las limitaciones presupuestarias.		

e. Observación de la tolerancia de la red y reacción de usuarios frente a distintos ataques cibernéticos

Tabla 11. Ficha de observación acerca de la tolerancia de la red y reacción de usuarios a distintos ataques cibernéticos.

Fecha:	01 de octubre de 2024
Lugar:	Laboratorio de computación de la Unidad Educativa “19 de Septiembre”
Observador:	El investigador
Descripción general del entorno:	Laboratorio de informática con conexión a internet, en el cual se encuentra el router principal, al momento de la observación el laboratorio se encontraba con usuarios.
Herramientas utilizadas:	• Computador con acceso a la red • Sistema operativo “Kali Linux” • Herramienta de monitoreo “WireShark”
Instrucción:	En las escalas, resaltar la opción que concuerde con lo observado durante la ejecución de los ataques.
Objetivo de la observación:	Analizar el comportamiento de la red de la Unidad Educativa “19 de Septiembre” ante diferentes ataques controlados, con el fin de identificar vulnerabilidades, tiempos de respuesta, estabilidad y posibles interrupciones que puedan comprometer su seguridad y rendimiento.

Tipo de ataque	Descripción	Escala (1-5)	Observaciones
DDoS (Denegación de servicio distribuida)	Se genera un tráfico masivo de datos hacia la red para sobrecargar los recursos y hacer que los servicios sean inaccesibles.	1 Red muy inestable 2 Red inestable 3 Red moderadamente estable 4 Red estable 5 Red muy estable	Luego de realizar el ataque de denegación de servicios la red colapsó a los 5 minutos, no dejando que dispositivos legítimos se conecten mediante wifi o conexión con cable ethernet, y los usuarios conectados a la red empezaron a sufrir desconexiones o conexión lenta a internet, al ser la inestabilidad de la red un problema recurrente, los usuarios no pudieron identificar que la red se encontraba bajo ataque restándole importancia al evento de conexión lenta.
Escaneo de puertos	Exploración de los puertos de la red para detectar aquellos abiertos y posibles vulnerabilidades.	1 Ningún puerto vulnerable 2 Pocos puertos vulnerables 3 Algunos puertos vulnerables 4 Varios puertos vulnerables 5 Muchos puertos vulnerables	Al realizar el análisis de puertos vulnerables se logró extraer información de host que no presentan medidas de seguridad como un antivirus o firewall, para los hosts que presentaban esta medida de seguridad, el software encargado alerto al usuario de un intento de escaneo de puertos, pero de igual manera a los usuarios desconocer este tipo de conceptos, le restaron importancia y siguieron desarrollando sus actividades con normalidad.
Spoofing de IP	Ataque en el que el atacante suplanta una dirección IP legítima para engañar a los sistemas de la red y obtener acceso.	1 Ninguna dirección IP sospechosa detectada 2 Pocas direcciones IP sospechosas detectadas 3 Moderada cantidad de direcciones IP sospechosas detectadas 4 Varias direcciones IP sospechosas detectadas 5 Muchas direcciones IP sospechosas detectadas	Al realizar el ataque ningún usuario logró detectar que una de las direcciones IP estuvo siendo suplantada, logrando de esa forma una suplantación de dirección sin levantar sospecha alguna, evidenciando de esa forma lo desapercibido que puede resultar un atacante si intentara realizar este tipo de vulnerabilidad en la institución.
Ataque de fuerza bruta	Intento repetido de adivinar contraseñas o credenciales para obtener acceso no autorizado.	1 Ningún intento fallido 2 Pocos intentos fallidos 3 Moderados intentos fallidos 4 Varios intentos fallidos 5 Muchos intentos fallidos	Al realizar un ataque de fuerza bruta para poder lograr una conexión SSH a un servidor local, se registraron pocos intentos fallidos, de igual manera ningún usuario de la red notó que se estuvo realizando este tipo de vulnerabilidad, lo que sugiere que un atacante puede tomarse un muy buen tiempo hasta poder descifrar usuarios y contraseñas utilizando la red institucional ya que el ataque duró 15 minutos, en ese lapso de tiempo no se llamó la atención de ningún usuario.

Tipo de ataque	Descripción	Escala (1-5)	Observaciones
Man-in-the-Middle (MITM)	Intercepción de comunicaciones entre dos partes para robar o manipular la información en tránsito.	1 Sin intercepción de datos 2 Poca intercepción de datos 3 Moderada intercepción de datos 4 Considerable intercepción de datos 5 Alta intercepción de datos	Al lograr la intercepción entre un modem y el router principal se logró obtener información sobre el tráfico DNS que los usuarios estaban generando en ese instante, durante la duración de 10 minutos que tuvo el ataque, no se levantó sospechas en los usuarios, una vez más demostrando en alto nivel de vulnerabilidad que tiene la red de la institución actualmente.
Observaciones adicionales	Los usuarios confían ciegamente en el bajo nivel de seguridad que la red tiene actualmente, por falta de conocimiento de los mismos, no pueden identificar a breves rasgos si sus datos o integridad de la red están siendo vulnerados, creen que es un comportamiento normal de la red debido a la costumbre que existe gracias a la estabilidad de la conexión que resulta muy recurrente.		
Conclusión:	El método de observación mediante ataques controlados reveló serias vulnerabilidades en la red de la institución, tanto a nivel de infraestructura como de falta de preparación de los usuarios. Los ataques de DDoS, escaneo de puertos, spoofing de IP, fuerza bruta y Man-in-the-Middle demostraron que la red no cuenta con medidas de seguridad robustas ni monitoreo en tiempo real, permitiendo que los ataques pasen desapercibidos. Además, la ausencia de conocimiento por parte de los usuarios y la falta de protocolos de respuesta ante incidentes resaltan la necesidad urgente de implementar soluciones de seguridad, como un sistema de detección de intrusos (IDS), junto con la capacitación de estudiantes y docentes para mitigar riesgos futuros.		

2.2.4 Procesamiento y análisis de datos

A partir de las entrevistas realizadas en la Unidad Educativa “19 de Septiembre”, se identificó una preocupación creciente en torno a la ciberseguridad, aunque, por diversas razones, este campo no ha recibido la atención necesaria. A pesar de contar con un equipamiento significativo que podría ser aprovechado para mejorar la infraestructura de red, la falta de personal capacitado ha impedido la implementación de soluciones adecuadas. La preocupación de la institución por mantenerse al día con los avances tecnológicos se refleja en los esfuerzos de autogestión, como la adquisición de nuevos equipos de cómputo para adaptar el sistema educativo a las demandas de la tecnología moderna. Sin embargo, al igual que el equipamiento de red almacenado, estos dispositivos aún no han sido puestos en funcionamiento debido a la ausencia de un estudio básico sobre la topología de red, necesario para integrar y optimizar el uso de estos nuevos recursos.

Los datos obtenidos a través de la ficha de observación sobre la tolerancia de la red y la reacción de los usuarios frente a ataques cibernéticos, presente en la Tabla 11, revela un desconocimiento considerable en la institución respecto a la seguridad informática. Durante los ataques realizados, los usuarios no manifestaron sospechas, incluso atribuyeron la lentitud y fallas en la conexión a problemas recurrentes de la red, lo que demuestra la falta de conciencia sobre posibles intrusiones. Además, la persona encargada de administrar la red no recibió ningún tipo de alerta o notificación, lo que permitió que los ataques se ejecutaran sin obstáculos durante un tiempo prolongado. Esto evidencia una vulnerabilidad crítica, ya que, en un escenario real, el tiempo concedido podría ser suficiente para que un atacante sustraiga información sensible y elimine cualquier rastro de su presencia en la red institucional. Esta situación subraya la urgente necesidad de implementar mecanismos de monitoreo y capacitación en temas de ciberseguridad para los usuarios y el personal encargado.

Para un mejor análisis de la encuesta aplicada, se decidió agrupar las preguntas que mostraron mayor correlación entre sí, con el objetivo de obtener una visión más amplia y generalizada de los resultados, y de cada pregunta se seleccionaron los dos ítems con mayor número de respuestas. Este proceso permitió la formación de tres grupos

principales. El primer grupo se enfocó en temas relacionados con la conectividad y los fallos en la red. El segundo grupo abarcó la temática de los sistemas de detección de intrusos, mientras que el tercer grupo trató la percepción sobre la seguridad de la red institucional y aspectos de la seguridad informática.

Análisis de preguntas del grupo 1.

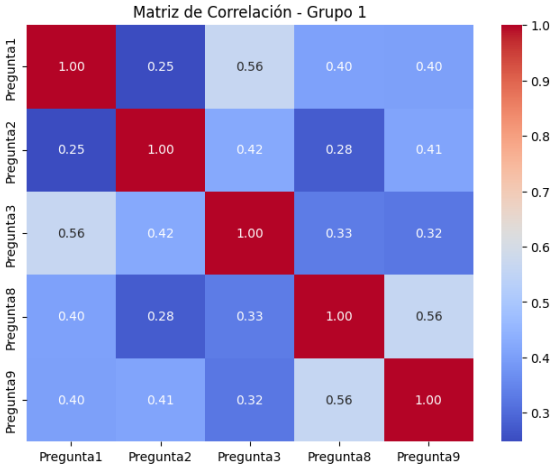


Figura 17. Correlación de preguntas del grupo 1.

Para el análisis de las preguntas del primer grupo se presentan los datos considerados en Tabla 12.

Tabla 12. Respuestas más seleccionadas del grupo 1 de preguntas

Pregunta	Opción 1	Porcentaje	Opción 2	Porcentaje	Porcentaje Total
¿Con qué frecuencia existen fallas en la red durante actividades académicas?	A veces	25.9%	Frecuentemente	24.5%	50.4%
Según su criterio, ¿Qué tan importante es la red para sus actividades académicas diarias?	Muy importante	49.6%	Importante	29.5%	79.1%
¿Con qué frecuencia experimenta desconexiones inesperadas?	A veces	32.4%	Frecuentemente	25.2%	57.6%
¿Qué tan frecuente es que deba buscar alternativas (como usar datos móviles) debido a problemas en la red institucional?	Siempre	41.7%	Frecuentemente	24.5%	66.2%
¿Qué tan grave considera que son los problemas de conexión en la red de la institución?	Grave	31.7%	Muy Grave	28.1%	59.8%

El análisis del grupo 1 de preguntas revela información significativa sobre la percepción de los usuarios respecto a la red institucional y su impacto en las actividades académicas, obteniendo los hallazgos más relevantes:

Fallas en la Red Durante Actividades Académicas:

Opción 1: "A veces" (25.9%)

Opción 2: "Frecuentemente" (24.5%)

Total (50.4%): Más de la mitad de los encuestados reporta que las fallas en la red ocurren al menos ocasionalmente, lo que sugiere que la red no es completamente confiable durante las actividades académicas. Sin embargo, el 49.6% restante indicó que las fallas ocurren "Nunca", "Ocasionalmente", o "Siempre", lo que podría significar una experiencia menos problemática o más grave para algunos usuarios.

Importancia de la Red para Actividades Académicas:

Opción 1: "Muy importante" (49.6%)

Opción 2: "Importante" (29.5%)

Total (79.1%): Una gran mayoría de los encuestados considera que la red es esencial para sus actividades académicas diarias, subrayando la necesidad de mejorar la infraestructura de red para asegurar el desarrollo óptimo del proceso educativo. El 20.9% restante percibe la red como "Moderadamente importante", "Poco importante" o "Nada importante", lo que podría reflejar una menor dependencia de la conectividad en sus actividades diarias.

Desconexiones Inesperadas:

Opción 1: "A veces" (32.4%)

Opción 2: "Frecuentemente" (25.2%)

Total (57.6%): La combinación de respuestas indica que un alto porcentaje de usuarios experimenta desconexiones inesperadas, lo que podría afectar negativamente su desempeño académico. Por otro lado, el 42.4% restante reporta que las

desconexiones ocurren "Nunca", "Ocasionalmente" o "Siempre", lo que sugiere que algunos usuarios enfrentan menos interrupciones o las perciben como algo más extremo.

Búsqueda de Alternativas Debido a Problemas en la Red:

Opción 1: "Siempre" (41.7%)

Opción 2: "Frecuentemente" (24.5%)

Total (66.2%): Este hallazgo sugiere que una parte considerable de los encuestados se ve obligada a buscar soluciones alternativas, como el uso de datos móviles, lo que indica una dependencia crítica de la red institucional. El 33.8% restante optó por "Nunca", "Ocasionalmente" o "A veces", lo que refleja que algunos usuarios no tienen que recurrir con frecuencia a estas medidas.

Gravedad de los Problemas de Conexión:

Opción 1: "Grave" (31.7%)

Opción 2: "Muy grave" (28.1%)

Total (59.8%): Casi el 60% de los encuestados considera que los problemas de conexión son significativos, lo que pone de relieve la urgencia de abordar estas deficiencias para mejorar la experiencia del usuario. El 40.2% restante calificó los problemas como "Poco grave", "Nada grave" o "Moderadamente grave", lo que sugiere que, aunque son perceptibles, para este grupo los inconvenientes son menos severos.

En conjunto, los resultados obtenidos evidencian una insatisfacción general con la calidad de la red institucional, lo cual impacta negativamente en las actividades académicas. La percepción de la red como un recurso fundamental y la alta incidencia de problemas de conectividad subrayan la necesidad de implementar mejoras en la infraestructura y en la gestión de la red. Estos hallazgos son cruciales para guiar futuras decisiones sobre inversiones y políticas en ciberseguridad y conectividad en la Unidad Educativa "19 de Septiembre".

Análisis del grupo 2 de preguntas

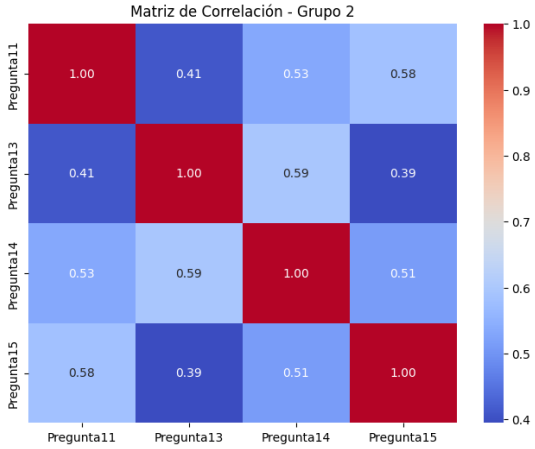


Figura 18. Correlación de preguntas del grupo 2.

En la Tabla 13 se exponen los datos tomados para el segundo grupo de preguntas.

Tabla 13. Respuestas más seleccionadas del segundo grupo de preguntas.

Pregunta	Opción 1	Porcentaje	Opción 2	Porcentaje	Porcentaje Total
¿Qué tan necesario considera recibir notificaciones sobre la seguridad de la red y posibles amenazas?	Necesario	43.2%	Muy necesario	33.8%	77%
¿Qué tan seguro/a se sentiría sabiendo que la red está protegida por un sistema de detección de intrusos (IDS)?	Seguro	35.3%	Normal	33.1%	68.4%
¿Qué tan necesario considera la implementación de un sistema para detectar intrusos para la protección de la red en la institución?	Moderadamente necesario	48.2%	Muy necesario	30.9%	79.1%
¿Qué tan interesado/a estaría en recibir más información sobre los sistemas de detección de intrusos (IDS) y su funcionamiento?	Interesado	46%	Muy interesado	25.2%	71.2%

El segundo grupo de preguntas aborda la percepción de los encuestados sobre la necesidad y efectividad de un sistema de detección de intrusos (IDS) y la importancia de recibir notificaciones sobre la seguridad de la red. A continuación, se presenta un resumen de los resultados obtenidos:

Necesidad de Recibir Notificaciones sobre la Seguridad de la Red:

Opción 1: "Necesario" (43.2%)

Opción 2: "Muy necesario" (33.8%)

Total (77%): Una gran mayoría de los encuestados considera que es fundamental recibir notificaciones sobre la seguridad de la red y posibles amenazas. Esto subraya la importancia de implementar un sistema de alertas que mejore la conciencia sobre los riesgos. Por otro lado, el 23% restante lo clasifica como "Moderadamente necesario", "Poco necesario" o "Nada necesario", lo que podría indicar que algunos usuarios no ven las notificaciones como una prioridad en su experiencia digital.

Sensación de Seguridad con un IDS:

Opción 1: "Seguro" (35.3%)

Opción 2: "Normal" (33.1%)

Total (68.4%): La percepción de seguridad aumenta significativamente al saber que la red podría estar protegida por un sistema de detección de intrusos, lo que indica que un IDS ayudaría a generar confianza entre los usuarios. Sin embargo, el 31.6% restante expresó sentirse "Muy seguro", "Poco seguro" o "Nada seguro", lo que podría estar relacionado con un conocimiento limitado sobre el funcionamiento y beneficios de un IDS.

Necesidad de Implementar un IDS:

Opción 1: "Moderadamente necesario" (48.2%)

Opción 2: "Muy necesario" (30.9%)

Total (79.1%): La mayoría de los encuestados considera que implementar un sistema de detección de intrusos es importante para la protección de la red en la institución, lo que refleja una alta aceptación de medidas proactivas en ciberseguridad. Por otro lado, el 20.9% restante lo calificó como "Poco necesario", "Nada necesario" o "Necesario", lo que indica que una minoría de usuarios percibe menor urgencia para adoptar esta medida.

Interés en Recibir Más Información sobre IDS:

Opción 1: "Interesado" (46%)

Opción 2: "Muy interesado" (25.2%)

Total (71.2%): Un porcentaje considerable de los encuestados muestra interés en recibir más información sobre los sistemas de detección de intrusos y su funcionamiento, lo que refleja una disposición a aprender y participar en iniciativas de seguridad. Sin embargo, el 28.8% restante expresó estar "Moderadamente interesado", "Poco interesado" o "Nada interesado", lo que podría evidenciar desconocimiento o una percepción de menor relevancia del tema.

Los resultados de este grupo de preguntas resaltan una fuerte demanda de información y una alta valoración de las medidas de seguridad, como la implementación de un IDS. Esto indica que los usuarios son conscientes de la importancia de proteger la red y están abiertos a la adopción de soluciones que aumenten la seguridad institucional. La implementación de un sistema de detección de intrusos no solo podría mejorar la protección de la red, sino también fomentar una mayor confianza y conciencia de los usuarios respecto a la ciberseguridad.

Análisis del grupo 3 de preguntas

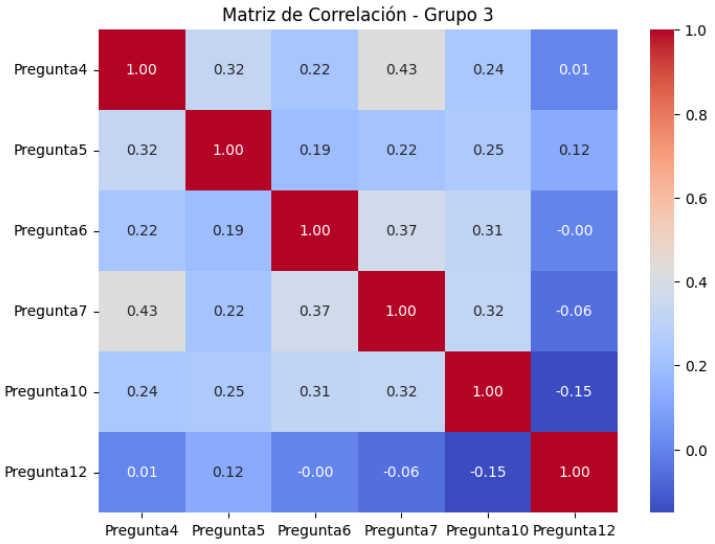


Figura 19. Correlación de preguntas del grupo 3.

Para el respectivo análisis, los datos seleccionados se encuentran en la Tabla 14.

Tabla 14. Respuestas más seleccionadas del tercer grupo de preguntas.

Pregunta	Opción 1	Porcentaje	Opción 2	Porcentaje	Porcentaje Total
¿Cuánto le preocupa la posibilidad de que alguien pueda acceder a su información personal a través de la red de la institución?	Muy preocupado	32.4%	Preocupado	32.4%	64.8%
¿Qué tan consciente es de los riesgos de seguridad que implica conectarse a la red institucional?	Normal	41%	Consciente	30.9%	71.9%
¿Qué tan dispuesto/a estaría a aceptar restricciones de acceso a la red si eso mejora su seguridad?	Dispuesto	43.9%	Poco dispuesto	19.4%	63.3%
¿Qué tan importante cree que es la seguridad de la red para la protección de la información personal?	Importante	41%	Muy importante	37.4%	78.4%
¿Qué tan útil considera que sería monitorear la red más estrictamente para evitar intrusiones?	Poco útil	33.8%	Útil	31.7%	65.5%
¿Qué tanto ha escuchado sobre los sistemas de detección de intrusos (IDS)?	Poco	40.3%	Nada	25.9%	66.2%

Este tercer grupo de preguntas se centra en la preocupación de los encuestados sobre la seguridad de su información personal en la red institucional, así como su conciencia y disposición hacia la implementación de medidas de seguridad, entre los resultados están:

Preocupación por el Acceso a Información Personal:

Opción 1: "Muy preocupado" (32.4%)

Opción 2: "Preocupado" (32.4%)

Total (64.8%): Una gran parte de los encuestados muestra una preocupación significativa por la posibilidad de que alguien acceda a su información personal a través de la red de la institución, lo que resalta la necesidad de medidas de protección robustas. Sin embargo, el 35.2% restante se distribuye entre "Nada preocupado", "Poco preocupado" o "Normal", lo que podría indicar que este grupo percibe menores riesgos o no es plenamente consciente de las posibles amenazas.

Conciencia de los Riesgos de Seguridad:

Opción 1: "Normal" (41%)

Opción 2: "Consciente" (30.9%)

Total (71.9%): La mayoría de los encuestados tiene algún nivel de conciencia sobre los riesgos de seguridad asociados con la conexión a la red institucional, lo que proporciona una base sólida para futuras iniciativas educativas sobre ciberseguridad. Por otro lado, el 28.1% restante indicó "Nada consciente", "Poco consciente" o "Muy consciente", lo que podría reflejar una variabilidad en el entendimiento de los riesgos.

Disposición a Aceptar Restricciones de Acceso:

Opción 1: "Dispuesto" (43.9%)

Opción 2: "Poco dispuesto" (19.4%)

Total (63.3%): Un porcentaje considerable de encuestados está dispuesto a aceptar restricciones de acceso a la red para mejorar su seguridad, lo que refleja un

compromiso con la protección de su información personal. El 36.7% restante señaló estar "Muy dispuesto", "Moderadamente dispuesto" o "Nada dispuesto", lo que podría sugerir una resistencia a las restricciones o falta de comprensión sobre sus beneficios.

Importancia de la Seguridad de la Red:

Opción 1: "Importante" (41%)

Opción 2: "Muy importante" (37.4%)

Total (78.4%): Casi el 80% de los encuestados considera que la seguridad de la red es fundamental para la protección de su información personal. Sin embargo, el 21.6% restante incluye respuestas como "Moderadamente importante", "Poco importante" o "Nada importante", lo que podría reflejar una menor dependencia de la red para actividades sensibles o una percepción de que la seguridad actual es suficiente.

Utilidad del Monitoreo Estricto de la Red:

Opción 1: "Poco útil" (33.8%)

Opción 2: "Útil" (31.7%)

Total (65.5%): Aunque la percepción general apunta a que el monitoreo podría ser útil, un porcentaje significativo lo considera "Poco útil", lo que podría reflejar un desconocimiento sobre cómo el monitoreo contribuye a la seguridad o una preocupación por la invasión de privacidad. El 34.5% restante se distribuye entre "Nada útil", "Moderadamente útil" o "Muy útil", lo que indica una mezcla de opiniones sobre el balance entre seguridad y privacidad.

Conocimiento sobre Sistemas de Detección de Intrusos (IDS):

Opción 1: "Poco" (40.3%)

Opción 2: "Nada" (25.9%)

Total (66.2%): La mayoría de los encuestados ha oído hablar poco o nada sobre los sistemas de detección de intrusos, lo que evidencia una oportunidad significativa para educar a la comunidad educativa sobre esta herramienta crucial para la ciberseguridad.

El 33.8% restante respondió "Moderadamente", "Mucho" o "Muy moderado", lo que sugiere que un segmento más informado podría ser clave para liderar el cambio hacia una mayor adopción de estas tecnologías.

El análisis del tercer grupo de preguntas refleja una considerable preocupación y conciencia sobre la seguridad de la información personal entre los usuarios de la red institucional. La disposición a aceptar restricciones de acceso y la alta valoración de la importancia de la seguridad de la red resaltan la necesidad de implementar políticas y herramientas efectivas de ciberseguridad, así como de realizar campañas de educación que incrementen el conocimiento sobre las amenazas y las soluciones, como los sistemas de detección de intrusos (IDS). Estos hallazgos son esenciales para guiar futuras decisiones sobre inversiones en seguridad y mejoras en la infraestructura de la Unidad Educativa "19 de Septiembre".

En la Unidad Educativa "19 de Septiembre" existe una creciente preocupación por la ciberseguridad. A pesar de contar con un presupuesto limitado, las autoridades de la institución han mostrado disposición para colaborar en la búsqueda de soluciones a este problema. Asimismo, gran parte de la comunidad educativa experimenta malestar al utilizar la red institucional, debido a diversos factores como problemas de conexión y las dificultades para acceder a los laboratorios de computación, lo que ha generado una desconfianza generalizada en la red.

Además, se evidenció que la comunidad educativa posee un conocimiento limitado sobre temas de ciberseguridad, lo que incrementa el riesgo de exposición a amenazas en una red que carece de medidas de protección adecuadas. Paralelamente, la institución está trabajando en reducir la brecha tecnológica entre los estudiantes mediante la construcción de nuevos laboratorios informáticos y la adquisición de equipos modernos. Sin embargo, estos aún no se encuentran en funcionamiento debido a la falta de personal capacitado para su instalación.

Este contexto refuerza el interés en la implementación de un sistema de detección de intrusos (IDS) en la unidad educativa, lo que no solo protegerá la nueva infraestructura, sino que también permitirá a los estudiantes interactuar de manera más fluida y segura con la tecnología.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados

En este apartado se analiza la relación entre los datos recolectados y los resultados obtenidos, discutiendo los hallazgos teóricos que sustentan la implementación de un sistema de detección de intrusos (IDS) basado en machine learning. Además, se evalúa la utilidad de esta solución en el contexto del proyecto, abordando la selección de los datos utilizados, el proceso de entrenamiento del modelo y la elección del algoritmo adecuado para su integración con el IDS.

3.1.1 Análisis de la infraestructura de red inicial

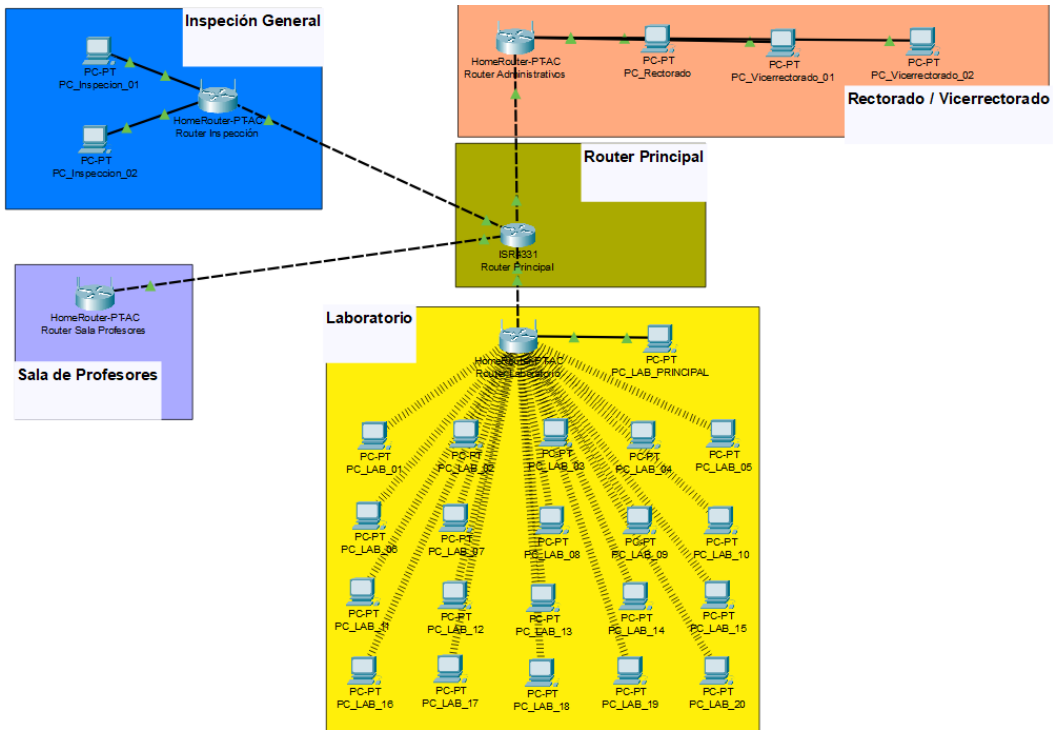


Figura 20. Topología de red inicial de la Unidad Educativa “19 de Septiembre”.

En la Figura 20, se presenta el estado inicial de la infraestructura de red de la Unidad Educativa "19 de Septiembre". Se identificaron cinco secciones principales que conforman su topología, las cuales se detallan a continuación, junto con las vulnerabilidades detectadas en cada una de ellas:

a. Router principal

El router principal de la institución está ubicado en el laboratorio de computación. Se identificaron vulnerabilidades que podrían comprometer la seguridad de la red, siendo la más significativa la falta de seguridad física. El acceso al router es fácil, ya que no cuenta con medidas de protección adecuadas, como candados o llaves en el rack que lo resguarden. Esta falta de control físico permite que cualquier persona pueda manipular el dispositivo, lo que representa un riesgo considerable para la seguridad informática de la red.

b. Laboratorio

En la sección de la red correspondiente a "Laboratorio", se observó que las máquinas acceden a la red de forma inalámbrica, lo que indica una falta de control sobre los dispositivos conectados, ya que no se les han asignado direcciones IP estáticas. Además, los computadores no están configurados con usuarios específicos, permitiendo que cualquier persona acceda con privilegios administrativos. Esto representa un riesgo significativo, ya que facilita la extracción de información sensible, como contraseñas de la red, lo que genera una importante vulnerabilidad en el sistema.

c. Sala de profesores

Al igual que en el laboratorio, en la sala de profesores se observó que los usuarios acceden a la red de manera inalámbrica. Esta práctica representa un riesgo significativo, ya que el tráfico de datos puede ser interceptado si no se cuenta con medidas de seguridad adecuadas, como la encriptación del tráfico o el uso de redes seguras. Además, no se tiene un control o registro de los dispositivos que se conectan a la red, lo que implica que cualquier equipo dentro del rango de la señal inalámbrica puede acceder a la red sin restricciones ni autenticación previa.

d. Inspección general

En el área de inspección general, se observó que dos equipos están conectados a la red mediante cables Ethernet, que provienen de un módem de uso residencial. Este módem, está conectado directamente al router principal de la institución. Esta configuración plantea varios problemas de seguridad, ya que el módem no ha sido

diseñado para entornos corporativos o educativos que requieren un mayor control y protección de la red. Además, al no estar implementadas configuraciones de seguridad avanzadas, como el uso de VLANs (redes locales virtuales) para segmentar el tráfico, toda la comunicación de estos equipos circula sin separación adecuada del resto de la red.

g. Rectorado / Vicerrectorado

En las oficinas de rectorado y vicerrectorado, se utiliza un tipo de conexión similar al que se encuentra en inspección general, es decir, mediante cables Ethernet conectados a un módem de uso residencial, el cual está vinculado directamente al router principal. Esta configuración aumenta las vulnerabilidades previamente identificadas, ya que carece de medidas de seguridad robustas, como la segmentación de la red mediante VLANs o la implementación de mecanismos de autenticación y control de acceso. Este tipo de conexión permite que todo el tráfico de la red circule sin restricciones a través de la misma infraestructura. La falta de segmentación y aislamiento entre las distintas áreas de la red incrementa el riesgo de ataques, como la intercepción de tráfico o el acceso no autorizado a los sistemas de rectorado y vicerrectorado, poniendo en peligro la confidencialidad y seguridad de la información institucional.

Al contrastar la información recopilada en el segundo capítulo con el análisis de la red en su estado actual, se puede observar una clara necesidad de implementar un sistema de detección de intrusiones (IDS). Esto se debe a la evidente carencia de controles de acceso y a la falta de configuraciones básicas de seguridad en la red. La infraestructura actual parece estar diseñada únicamente para asegurar el funcionamiento básico de la red, sin la aplicación de normas de seguridad esenciales. Esta situación no solo expone la red a riesgos significativos, como la filtración de datos, sino que también refuerza la percepción negativa que tienen los usuarios sobre la calidad de la red institucional, manifestada en problemas de conectividad, lentitud al acceder a internet, y la falta de protección ante posibles intrusiones.

3.1.2 Software de detección de intrusiones (IDS)

Para poder implementar un sistema de detección de intrusiones (IDS), existen varias opciones de software diseñado para este fin, para el proyecto se tomó en cuenta opciones con filosofía de código abierto debido al presupuesto limitado de la Unidad Educativa "19 de Septiembre" para poder costear opciones con licencia de pago, entre las opciones consideradas están:

Suricata: Software de código abierto y alto rendimiento para la detección y prevención de amenazas en la red, capaz de operar en tiempo real, identificando patrones maliciosos y respondiendo a las amenazas de manera proactiva [30].

Snort: Herramienta IDS e IPS de código abierto popular. Se basa en la utilización de reglas y comparación de paquetes para definir la actividad maliciosa de la red y generar alertas a los usuarios [31].

Zeek: Framework de código abierto utilizado para el análisis y búsqueda de amenazas en la red. Su principal propósito es filtrar el tráfico en la red y generar logs que describan todo lo que ocurre en ella [32].

Tabla 15. Comparativa softwares IDS [30], [31], [32].

Criterio	Suricata	Snort	Zeek
Enfoque principal	Detección de intrusos, análisis de tráfico en tiempo real, y captura de datos	Detección de intrusos basado en firmas	Análisis profundo del tráfico de red y monitoreo de comportamiento
Detección	Basado en firmas y análisis del comportamiento (detección múltiple)	Basado principalmente en firmas (patrones conocidos de ataques)	Análisis basado en comportamiento y eventos
Rendimiento	Soporta alta velocidad, ideal para grandes redes con tráfico intenso	Adecuado para redes medianas y grandes, pero puede sufrir bajo alta carga	Adecuado para análisis profundo, pero con mayor uso de recursos
Capacidad de inspección	Inspección de tráfico de red a nivel de capa de aplicación, soporte para DPI (Deep Packet Inspection)	Inspección a nivel de paquetes, limitado en cuanto a DPI en capas superiores	Inspección profunda de sesiones y conexiones, con enfoque en análisis a nivel de aplicación

Criterio	Suricata	Snort	Zeek
Velocidad	Muy eficiente en el análisis de tráfico a alta velocidad (10 Gbps o más)	Menor rendimiento que Suricata en entornos de alto tráfico	No está optimizado para alta velocidad, se enfoca en análisis detallado
Actualización de firmas	Soporte para firmas de Snort, más actualizaciones propias	Amplia base de firmas, actualizaciones frecuentes	No depende de firmas, sino de políticas de comportamiento
Compatibilidad con Machine Learning	Compatible con modelos de machine learning, permite la detección basada en comportamiento y la integración con algoritmos de aprendizaje automático	Limitado en integración directa con machine learning; principalmente basado en firmas	Puede ser complementado con machine learning para el análisis de eventos, pero no se integra directamente
Soporte multicore	Optimizado para aprovechar múltiples núcleos de CPU (alto rendimiento)	No está completamente optimizado para multicore	Mayor demanda de CPU por el análisis detallado, pero no optimizado para multicore
Facilidad de uso	Amplias capacidades, pero con curva de aprendizaje moderada debido a sus múltiples funciones	Relativamente fácil de configurar y usar, especialmente para detección basada en firmas	Requiere mayor conocimiento técnico y personalización
Documentación y comunidad	Excelente documentación, activa comunidad de soporte y contribuciones	Documentación sólida, muy popular y bien respaldado por la comunidad	Documentación técnica extensa, comunidad académica y profesional activa

Después de analizar las características de diversos softwares de detección de intrusos, se decidió utilizar Suricata debido a su capacidad para manejar grandes volúmenes de datos, lo que resulta fundamental para la escalabilidad prevista en la Unidad Educativa "19 de Septiembre". Su optimización para arquitecturas multinúcleo de CPU permite un uso más eficiente de los recursos disponibles, garantizando un rendimiento óptimo incluso en entornos de alto tráfico.

A diferencia de Snort, Suricata no se limita únicamente a la detección mediante firmas; también incorpora un enfoque basado en el comportamiento. Esta capacidad es esencial para identificar tráfico sospechoso que no se ajusta a patrones previamente conocidos, lo que permite una integración más fluida con algoritmos de machine learning. En contraste, Snort y Zeek no facilitan esta integración de manera tan orgánica. Suricata es compatible con las firmas de Snort y cuenta con su propio conjunto de firmas optimizadas, lo que proporciona una gran flexibilidad en términos

de actualización y personalización de reglas, adaptándose fácilmente a las necesidades cambiantes de seguridad de la red.

La comunidad de soporte para Suricata es activa y comprometida, lo que asegura actualizaciones regulares y mejoras continuas en el rendimiento. Esto contrasta con Zeek, que se centra más en la investigación de seguridad y el análisis detallado, mientras que Suricata está orientado a un uso práctico y comercial en redes en producción. Suricata ofrece un equilibrio ideal entre rendimiento y análisis profundo: Snort puede resultar limitado en redes de gran tráfico, mientras que Zeek tiende a concentrarse en un análisis más exhaustivo y orientado a eventos.

Por su versatilidad y capacidad de adaptación, Suricata se considera la mejor opción para el desarrollo del proyecto. Su capacidad de escalar se alinea perfectamente con los planes futuros de la institución para la implementación de nuevos laboratorios de computación. Además, al contar con su propia base de firmas y capacidades de detección basadas en comportamiento, Suricata no solo protege la red, sino que también permite la integración efectiva con algoritmos de machine learning, ofreciendo así un sistema de detección de intrusos robusto que realiza detecciones en tiempo real.

3.1.3 Metodologías para proyectos de ciencia de datos y machine learning.

En proyectos que integran ciencia de datos y machine learning, la elección de la metodología adecuada es crucial para gestionar de manera eficiente todas las etapas del ciclo de vida del proyecto. Estas metodologías están diseñadas para abordar desde la recolección y procesamiento de datos hasta el despliegue de modelos de aprendizaje automático en entornos de producción. Bajo este enfoque, se han considerado las siguientes opciones:

CRISP-DM (Cross Industry Standard Process for Data Mining): Una metodología estándar ampliamente utilizada en proyectos de minería de datos. Su estructura se organiza en seis etapas claras: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue. Este enfoque asegura una implementación ordenada y eficiente de soluciones analíticas [33].

KDD (Knowledge Discovery in Databases): Un proceso sistemático diseñado para extraer conocimiento de grandes volúmenes de datos. Sus etapas incluyen selección, preprocesamiento, transformación, minería de datos e interpretación de resultados, con el objetivo de convertir datos sin procesar en información valiosa y conocimiento accionable [34].

Agile Data Science: Un enfoque iterativo y ágil que se adapta a proyectos de ciencia de datos mediante ciclos cortos de desarrollo. Promueve la colaboración constante entre equipos y usuarios, permitiendo generar rápidamente soluciones funcionales de análisis y modelos predictivos que se ajusten a las cambiantes necesidades del negocio [35].

En la Tabla 16, se presenta una comparación de estas metodologías, destacando sus características y capacidades, con el propósito de identificar cuál ofrece la mejor adaptabilidad para el desarrollo del proyecto.

Tabla 16. Metodologías para proyectos de ciencias de datos y machine learning [33], [34], [35].

Criterio	CRISP-DM (Cross Industry Standard Process for Data Mining)	KDD (Knowledge Discovery in Databases)	Agile Data Science
Enfoque general	Proceso estructurado para proyectos de minería de datos, adaptable a diversos problemas.	Centrado en la extracción de conocimiento útil a partir de grandes volúmenes de datos.	Adaptación de los principios de metodología ágil a proyectos de ciencia de datos.
Adecuación a proyectos IDS	Ideal para estructurar un proyecto desde la comprensión del problema hasta el despliegue del modelo IDS, con fases claras de evaluación.	Adecuado para proyectos que requieren grandes volúmenes de datos, pero menos centrado en el contexto del negocio y las necesidades del usuario.	Útil para desarrollos rápidos, pero puede carecer de profundidad analítica en algunos casos complejos como un IDS.
Flexibilidad	Muy adaptable a distintos tipos de problemas y escalas de datos. Se puede ajustar el nivel de detalle en cada fase.	Menos flexible, dado que su foco está en grandes volúmenes de datos y no necesariamente en la aplicabilidad del sistema final.	Alta flexibilidad para cambios rápidos, pero menos formal en la definición de requisitos específicos de seguridad y análisis.
Nivel de documentación	Altamente documentado, lo que permite un seguimiento riguroso en cada fase del proyecto.	Requiere mucha preparación en las primeras fases, pero puede carecer de una documentación tan detallada en fases posteriores.	La documentación es más ligera y se ajusta a cada sprint, pero puede ser difícil de seguir en proyectos complejos.
Complejidad	Complejidad media, cada fase es bien definida y se ajusta a proyectos individuales o en equipo.	Mayor complejidad en la preparación y transformación de datos antes de aplicar modelos.	Baja complejidad, pero requiere más iteraciones para obtener resultados completos y puede perder detalle en proyectos más grandes.
Gestión de recursos	Eficiente en términos de recursos, permite optimizar tiempos en cada fase y evita la redundancia de tareas.	Consumo elevado de recursos en las fases iniciales de selección y procesamiento de datos.	Ágil, pero puede requerir más ajustes en cada iteración, lo que puede llevar a un uso ineficiente de recursos en ciertos puntos.
Soporte para Machine Learning	Altamente compatible con técnicas de machine learning. Tiene fases dedicadas para modelado y evaluación.	Enfocado en minería de datos, pero no específicamente en machine learning. El ajuste del modelo es menos detallado.	Compatible con machine learning, pero la evaluación continua puede ser superficial si no se ajustan bien los sprints.

Tras el respectivo análisis, se opta por la metodología CRISP-DM debido a su estructura clara y adaptable, la cual permite dividir el proyecto en fases bien definidas. Esto facilita un desarrollo ordenado y sistemático, lo que resulta especialmente útil al combinarlo con la implementación de un sistema de detección de intrusos (IDS). Además, CRISP-DM no se limita únicamente al procesamiento de datos, sino que también abarca la comprensión del problema del negocio, que en este caso es la seguridad de la red, asegurando que la solución se ajuste a las necesidades específicas de la institución.

La flexibilidad en cada fase de CRISP-DM permite ajustar la profundidad de cada etapa según los recursos disponibles y el tiempo estipulado. Al incluir fases dedicadas al modelado y la evaluación, la metodología garantiza que el algoritmo de machine learning funcione de manera adecuada al integrarlo con el IDS, asegurando su eficacia. Otro aspecto clave de CRISP-DM es que promueve una documentación detallada en cada fase del proyecto, lo que facilita la elaboración de un informe final claro y detallado, sin omitir aspectos esenciales. Por todas estas características, CRISP-DM se considera la metodología con mejor adaptabilidad y alineación para el desarrollo del presente proyecto.

a. Fases de la metodología CRISP-DM

Una vez seleccionada la metodología adecuada, se procede a adaptar sus fases para aplicarlas al desarrollo del proyecto. Esta adaptación permite contar con una estructura de trabajo organizada, asegurando un flujo de trabajo eficiente que guíe cada etapa hasta el cumplimiento de los objetivos establecidos. A continuación, se describen las fases del proyecto, basadas en la metodología CRISP-DM, adaptadas para la implementación de un sistema de detección de intrusos (IDS) apoyado en machine learning para la Unidad Educativa “19 de Septiembre”.

- Comprensión del negocio
- Comprensión de los datos
- Preparación de los datos
- Modelado
- Evaluación

- Despliegue
- Documentación

Comprensión del negocio: Busca comprender la falta de seguridad y protección contra posibles intrusiones de la red de la Unidad Educativa “19 de Septiembre”. Se lleva a cabo un análisis de las necesidades de la red, las políticas actuales de seguridad, y las vulnerabilidades detectadas. Además, se analizan los posibles impactos que estas deficiencias de seguridad tienen en la comunidad educativa.

Comprensión de los datos: Aquí, se define y selecciona el dataset que se utilizará para entrenar el algoritmo de machine learning. Durante esta fase, se analiza la calidad de los datos disponibles, identificando cualquier problema como datos faltantes o inconsistentes, y se evalúa la adecuación de los mismos para entrenar un modelo efectivo.

Preparación de los datos: Se procede a la preparación del dataset seleccionado. Esta fase asegura que los datos estén en un formato adecuado para el proceso de modelado. Se realizan tareas de limpieza de datos, que incluyen la eliminación de datos redundantes, la imputación de valores faltantes y la normalización de las variables. Además, se pueden aplicar técnicas de selección de características (features) para reducir la dimensionalidad del dataset y optimizar el rendimiento del modelo de machine learning. El objetivo final es disponer de un conjunto de datos estructurado y limpio que refleje adecuadamente el comportamiento de la red.

Modelado: Se lleva a cabo el entrenamiento del modelo de machine learning. Se seleccionan y comparan distintos algoritmos que han sido considerados. Se entrenan utilizando los datos preparados en la fase anterior. Durante este proceso, se ajustan los hiperparámetros de los algoritmos para maximizar su rendimiento. También se realizan pruebas utilizando técnicas de validación cruzada para asegurar que el modelo no solo funcione bien en los datos de entrenamiento, sino que también generalice correctamente en datos nuevos

Evaluación: Una vez entrenado el modelo, se procede a la evaluación de su rendimiento. Se miden indicadores clave como la precisión, el recall, la tasa de falsos positivos y negativos, y el tiempo de respuesta del modelo. Se realiza una comparativa entre los diferentes algoritmos probados, con el fin de determinar cuál se adapta mejor a las necesidades del proyecto. El modelo seleccionado debe ser eficiente en la

detección de intrusiones sin generar demasiadas alertas falsas, y debe ser capaz de identificar con precisión cualquier comportamiento anómalo en la red.

Despliegue: Implementa el sistema de detección de intrusos (IDS) en la red de la Unidad Educativa "19 de Septiembre". Se integran las configuraciones del IDS con el algoritmo de machine learning seleccionado, para que el sistema sea capaz de detectar intrusiones en tiempo real. Esta etapa incluye la configuración del hardware y software necesario, así como la puesta en marcha del sistema en un entorno controlado. También se realizan pruebas para garantizar que el IDS funcione correctamente en el entorno de producción, identificando posibles ajustes necesarios antes de su despliegue completo.

Documentación: Finalmente, se procede a la elaboración de la documentación correspondiente. Esta fase incluye la creación de un informe técnico que detalla cada fase del proyecto, así como los resultados obtenidos. Además, se elabora una guía de usuario para que el personal de la Unidad Educativa pueda administrar y mantener el sistema IDS de manera eficiente. Esta documentación es esencial para garantizar la continuidad del proyecto, facilitando futuras mejoras y adaptaciones, y proporcionando un registro claro de todo el proceso.

3.2 Desarrollo de la propuesta

Para desarrollar el proyecto se analizó las fases que conforman la metodología CRISP-DM, que se encuentran resumidas en el diagrama de la Figura 21, misma que se enfoca en proyectos que utilizan machine learning, adaptando cada una de ellas a las necesidades encontradas para poder cumplir con la propuesta.

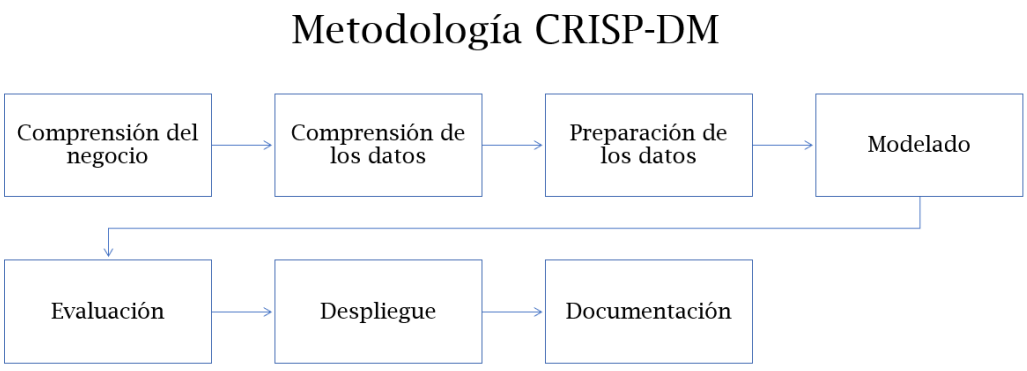


Figura 21. Diagrama de metodología CRISP-DM

3.2.1 Fase 1: Comprensión del negocio

En la fase de comprensión del negocio, se llevó a cabo un análisis inicial para determinar el estado de la red de la Unidad Educativa “19 de Septiembre”. Se emplearon técnicas de recolección de información, para identificar problemas clave como la desconexión repentina de la red, la desconfianza de los usuarios, un nivel considerable de vulnerabilidad, y la ausencia de registros y políticas de seguridad. Este análisis permitió establecer un contexto general que facilitó la definición de segmentos clave para el proyecto, como grupos de interés, requisitos funcionales, requisitos no funcionales, el alcance del proyecto y el establecimiento del mismo.

a. Especificación de los grupos de interés

- **Responsable de TI**
  Es el encargado de implementar es sistema IDS y capacitar al personal correspondiente.
- **Docentes y estudiantes**
  Son los usuarios que se benefician directamente con el desarrollo del proyecto.
- **Personal administrativo**
  Son los que se encargan de gestionar los bienes tecnológicos de la institución, interesados en asegurar la confiabilidad de la red.

b. Requisitos funcionales

Los requerimientos funcionales, detallados en la Tabla 17, especifican las capacidades esenciales que debe implementar el sistema, tales como la detección de intrusiones y la generación de alertas visuales en tiempo real. Estos requisitos se identifican mediante el prefijo "RF" seguido de un número correlativo, como en el caso de RF01, que corresponde al "Requerimiento Funcional 01", relacionado con la detección de intrusiones. Este sistema de codificación fue adoptado para seguir los lineamientos de la metodología CRISP-DM [33].

El orden asignado a los requerimientos funcionales refleja una jerarquía lógica y una priorización fundamentada en su relevancia para cumplir con los objetivos del proyecto. La detección de intrusiones se estableció como el enfoque central del

sistema, seguido de funciones complementarias que refuerzan su desempeño. Los requerimientos funcionales fueron establecidos a partir de las respuestas obtenidas mediante herramientas de recolección de información, las cuales permitieron identificar las necesidades clave del sistema para satisfacer las directrices de la propuesta.

Tabla 17. Requerimientos funcionales del sistema IDS.

Código	Requerimiento	Descripción
RF01	Detección de intrusiones	El sistema debe ser capaz de detectar actividades sospechosas y posibles ataques en tiempo real
RF02	Registro de actividad	Al detectar actividad sospechosa o un posible ataque, se almacena la información de las amenazas en archivos log.
RF03	Compatibilidad con la infraestructura existente	El sistema IDS debe adaptarse a la infraestructura de la institución, además de utilizar los recursos existentes.
RF04	Alerta de riesgos	Mientras se realiza el escaneo de la red en tiempo real, al detectar una amenaza, el sistema debe mostrar de manera visual los inconvenientes encontrados.
RF05	Interfaz de monitoreo	Para facilitar la comprensión de los datos generados mientras el sistema está activo, la información debe mostrarse en dashboards.

c. Requisitos no funcionales

Los requerimientos no funcionales, presentados en la Tabla 18, se centran en aspectos cualitativos que garantizan la eficiencia operativa y la calidad del sistema, tales como su rendimiento, escalabilidad y seguridad. Estos aspectos son fundamentales para asegurar que el sistema no solo cumpla con las funciones esperadas, sino que también se desempeñe de manera óptima y pueda adaptarse a futuras necesidades. Para facilitar la organización y el seguimiento de estos requisitos, se asignó el prefijo "RNF" seguido de un número correlativo, como en RNF01, que hace referencia al "Requerimiento No Funcional 01", relacionado con la eficiencia del sistema. Esta codificación fue adoptada conforme a los principios de la metodología CRISP-DM [33].

La priorización de los requerimientos no funcionales se enfocó inicialmente en el rendimiento del sistema IDS, asegurando que pudiera procesar el tráfico en tiempo real sin comprometer el funcionamiento de la red. Dichos requerimientos fueron establecidos a partir de un análisis general, con el objetivo de garantizar eficiencia, escalabilidad y seguridad.

Tabla 18. Requerimientos no funcionales del sistema IDS.

Código	Requerimiento	Descripción
RNF01	Eficiencia	El sistema IDS debe poder analizar el tráfico en tiempo real sin afectar el rendimiento de la red.
RNF02	Escalabilidad	La implementación del IDS debe permitir la adaptación a un futuro crecimiento de la red.
RNF03	Fiabilidad	La detección debe realizarse de manera continua y no verse afectada por errores en el código del algoritmo de machine learning.
RNF04	Seguridad	El IDS solo deberá ser manipulado por personal autorizado y se debe garantizar la confidencialidad de los datos.
RNF05	Facilidad de mantenimiento	La implementación del sistema debe contar con la documentación respectiva que permita manipular el IDS para trabajo de mantenimiento y actualizaciones.

d. Alcance del proyecto

Para poder definir el alcance del proyecto, hay que conocer las limitaciones que este presenta.

Limitaciones:

- El servidor IDS cuenta con 4 GB de memoria RAM, lo que en ciertas ocasiones puede llegar a limitar la capacidad de procesamiento.
- El entrenamiento inicial del modelo se basa en un dataset existente, que pese a tener datos representativos, puede no cubrir todos los tipos de ataques específicos que puede experimentar una institución educativa.
- Para un correcto funcionamiento tanto el IDS como el modelo de machine learning requieren actualizaciones periódicas para adaptarse a nuevos tipos de ataques.

De igual forma es necesario identificar los supuestos y dependencias del proyecto.

Establecimiento de categoría:

- El sistema IDS brinda una solución adaptada a la protección de una red educativa, con un enfoque en el monitoreo y detección de intrusiones.

- Emplea un modelo de machine learning para lograr la identificación temprana de amenazas en la red.

e. Establecimiento del proyecto

Para poder desarrollar el proyecto se utilizaron las distintas herramientas que se describen a continuación:

- **Ubuntu server**

Sistema operativo basado en Linux diseñado para servidores. Es estable, seguro y muy utilizado para ejecutar aplicaciones y servicios de red.

- **Suricata**

Sistema de detección de intrusos (IDS) de alto rendimiento. Puede analizar el tráfico de red en tiempo real y generar registros detallados en formato JSON.

- **Grafana**

Herramienta para la visualización y monitoreo de datos. Permite crear dashboards dinámicos que sirven para analizar y presentar métricas en tiempo real.

- **Plugin Infinity para Grafana**

Es una extensión de Grafana que ayuda a cargar datos desde APIs Json, CSV o cualquier otro formato personalizado.

- **Flask**

Framework web ligero en Python, ideal para desarrollar APIs y aplicaciones web.

- **Postman**

Herramienta que sirve para probar APIs, ayuda a facilitar la realización de peticiones para poder comprobar el funcionamiento de las mismas.

- **Random Forest**

Algoritmo para el aprendizaje automático basado en los árboles de decisión.

- **Sklearn**

Biblioteca de Python que incluye herramientas para modelado predictivo, procesamiento de datos y evaluación de modelos de machine learning.

- **Pandas, NumPy**

Bibliotecas para la manipulación y análisis de datos estructurados. Utilizada para procesar y limpiar datasets en Python.

- **Matplotlib, Seaborn**

Son bibliotecas que se usan en Python para generar visualizaciones básicas, como histogramas o diagramas de dispersión.

- **CSE-CIC-DDoS 2019**

Es un dataset público que contine datos de tráfico de red etiquetado, incluyendo ataques y tráfico benigno.

- **Visual Studio Code**

Editor de código ligero. Utilizado para escribir, depurar y administrar código.

f. Definición de las fases

Para poder definir una mejor estructura del proyecto en la Tabla 19, se expone el establecimiento de las distintas fases de manera resumida permitiendo la realización progresiva del mismo hasta la finalización.

Tabla 19. Fases del proyecto.

Fase	Iteración	Descripción
Comprensión del negocio	Iteración 0	- Definición de la arquitectura inicial de la red mediante la recolección de datos. - Puntualizar grupos de interés. - Especificación de requerimientos funcionales y no funcionales. - Establecimiento del proyecto. - Fijar las fases del proyecto.
Comprensión de los datos	Iteración 1	- Identificación y evaluación de fuentes de datasets públicos. - Comparación de los datasets considerados. - Selección y descarga del dataset.

Fase	Iteración	Descripción
Preparación de los datos	Iteración 2	- Carga y validación de los datos. - Filtrado de características. - Limpieza de datos. - Normalización y escalado.
Modelado	Iteración 3	- Selección de algoritmo de machine learning. - Configuración de entorno de desarrollo. - Entrenamiento inicial. - Optimización de hiperparámetros.
Evaluación	Iteración 4	- Definición de métricas de evaluación. - Evaluación general del modelo.
Despliegue	Iteración 5	- Preparación del entorno. - Configuración inicial del servidor. - Instalación y configuración de software IDS. - Creación y configuración de API en el servidor IDS.
	Iteración 6	- Instalación y configuración de herramienta de monitoreo. - Integración con modelo de machine learning. - Automatización de procesos
	Iteración 7	- Pruebas de integración.
Documentación	Iteración 8	- Elaboración de manual de usuario. - Elaboración de documento final.

3.2.2 Fase 2: Comprensión de los datos

La fase de comprensión de los datos abarca una serie de pasos para poder tener como resultado final la elección del dataset que servirá para el entrenamiento del modelo de machine learning.

a. Identificación de datasets públicos

Para la respectiva búsqueda de los datasets se cuenta con una ficha de búsqueda expuesta en Tabla 20.

Tabla 20. Ficha de búsqueda datasets.

Ficha de búsqueda Datasets	
Campo	**Descripción**
Nombre del Dataset	
Fuente/Repositorio	
Descripción	
Propósito	
Tamaño del Dataset	
Formato de los Datos	

Campo	Descripción
Etiquetas Disponibles	
Características Clave	
Relevancia	
Limitaciones	

Luego de realizar la respectiva búsqueda de datasets públicos, se seleccionaron los que se consideran relevantes en el contexto del proyecto, mismos que se describen a continuación.

Tabla 21. Ficha de búsqueda dataset UNSW-NB15 [36].

Ficha de búsqueda Datasets	
Campo	**Descripción**
Nombre del Dataset	UNSW-NB15
Fuente/Repositorio	Centro Australiano de Seguridad Cibernética, por sus siglas en inglés, (ACCS)
Descripción	Dataset creado para evaluar sistemas IDS modernos. Incluye tráfico benigno y malicioso generado en un entorno real y virtual.
Propósito	Entrenamiento y evaluación de modelos IDS.
Tamaño del Dataset	700 MB
Formato de los Datos	.CSV
Etiquetas Disponibles	Tráfico benigno, múltiples tipos de ataques modernos.
Características Clave	49 características, incluidas métricas de red, estadísticas de flujo y características específicas de los ataques.
Relevancia	Alta: Representa una variedad de escenarios y es manejable en tamaño.
Limitaciones	Bajo volumen de datos.

UNSW-NB15 es un dataset desarrollado en 2015 por el Instituto Australiano de Seguridad Cibernética (ACCS, por sus siglas en inglés), surgió como una respuesta a la creciente necesidad de contar con datos más recientes y representativos sobre ataques cibernéticos. Según Moustafa [36], este dataset fue diseñado con el objetivo de abordar las limitaciones de conjuntos de datos anteriores que, debido a su antigüedad, no reflejaban adecuadamente los patrones de ataque y las amenazas

emergentes en un entorno digital en constante evolución. Además, el aumento en el uso de sistemas de detección de intrusos basados en red (NIDS) y la necesidad de entrenar modelos automatizados para la detección efectiva de intrusiones requerían datos actualizados y de alta calidad.

Tabla 22. Ficha de búsqueda dataset CSE-CIC-DDoS 2019 [37].

Ficha de búsqueda Datasets	
Campo	**Descripción**
Nombre del Dataset	CSE-CIC-DDoS 2019
Fuente/Repositorio	Instituto Canadiense de Ciberseguridad, por sus siglas en inglés (CIC)
Descripción	Dataset de tráfico de red etiquetado que incluye ataques modernos como DDoS, Inyección SQL, Fuerza bruta, entre otros.
Propósito	Análisis de detección de intrusos y entrenamiento de modelos de machine learning.
Tamaño del Dataset	20.7 GB
Formato de los Datos	CSV, JSON
Etiquetas Disponibles	Tráfico benigno, tipos específicos de ataques como Fuerza bruta, DDoS, Inyección SQL.
Características Clave	88 características, incluidas métricas de red como tiempo activo/inactivo, longitud de paquetes, etc.
Relevancia	Muy alta: Contiene datos recientes y variados.
Limitaciones	Gran tamaño, requiere preprocesamiento extenso para reducir dimensionalidad.

El dataset CSE-CIC-DDoS 2019, desarrollado por el Instituto Canadiense de Ciberseguridad (CIC, por sus siglas en inglés) en el año 2019, fue creado con el objetivo de impulsar el desarrollo de sistemas de detección de intrusos (IDS) inteligentes y enfrentar los desafíos dinámicos en el ámbito de la ciberseguridad. Según el CIC [37], este conjunto de datos incluye información generada a partir de ataques DDoS basados en reflexión, los cuales replican la forma tradicional de estos ataques, pero con el uso de múltiples protocolos de la capa de aplicación, incrementando su complejidad y alcance. Asimismo, incorpora ataques basados en exploración, diseñados para deshabilitar redes mediante la replicación masiva de

solicitudes. Además, el dataset incluye tráfico benigno, ofreciendo una combinación equilibrada que representa tanto escenarios de ataque como tráfico normal. Estas características hacen que el dataset sea una herramienta versátil y actualizada, ideal para entrenar y evaluar sistemas IDS basados en machine learning, especialmente en redes de mediana y gran escala.

Tabla 23. Ficha de búsqueda de dataset CIC-IoV 2024 [38].

Ficha de búsqueda Datasets	
Campo	**Descripción**
Nombre del Dataset	CIC-IoV 2024
Fuente/Repositorio	Instituto Canadiense de Ciberseguridad, por sus siglas en inglés (CIC)
Descripción	Dataset especializado para escenarios de IoT, incluyendo redes vehiculares y tráfico relacionado con dispositivos inteligentes.
Propósito	Evaluar sistemas IDS en entornos IoT y vehículos inteligentes.
Tamaño del Dataset	100 GB
Formato de los Datos	CSV, JSON
Etiquetas Disponibles	Tráfico benigno, ataques específicos a dispositivos IoT como Botnets, Infiltración, Denegación de servicio.
Características Clave	Más de 90 características, con foco en métricas relevantes para dispositivos IoT, como consumo de energía y patrones de comunicación específicos.
Relevancia	Alta: Incluye tráfico especializado para IoT, pero puede no ser directamente aplicable a redes convencionales.
Limitaciones	Requiere hardware especializado para generar resultados significativos en entornos vehiculares.

Con la evolución constante de los entornos IoT y el creciente auge de tecnologías avanzadas como los vehículos inteligentes, el Instituto Canadiense de Ciberseguridad (CIC) decidió en 2024 desarrollar el dataset CIC-IoV 2024, diseñado específicamente para abordar los desafíos emergentes en este tipo de entornos tecnológicos. Según el CIC [38], el objetivo principal al crear este conjunto de datos fue generar información realista y representativa que pueda ser utilizada en escenarios específicos, con el fin de desarrollar nuevas y efectivas soluciones de ciberseguridad. Este dataset no solo se centra en el ámbito general del Internet de las cosas (IoT), sino que pone un énfasis particular en las operaciones y comunicaciones de los vehículos inteligentes, la

interacción con infraestructuras inteligentes y la seguridad de los datos transmitidos. La creación de este dataset responde a la necesidad de contar con herramientas que permitan investigar y diseñar medidas de protección adaptadas a los riesgos únicos que enfrentan los entornos IoT en constante evolución, especialmente en aplicaciones críticas las redes vehiculares inteligentes.

b. Comparación de datasets considerados

Tras el análisis de los datasets considerados, se elaboró la Tabla 24 como una herramienta comparativa que permite evaluar sus características principales y determinar cuál de ellos se ajusta mejor al contexto y los objetivos específicos del proyecto. Esta tabla no solo resume las fortalezas y debilidades de cada conjunto de datos, sino que también destaca aspectos clave como su relevancia, actualidad, tipo de ataques incluidos y aplicabilidad en redes tradicionales y modernas. De esta manera, se facilita una toma de decisiones fundamentada, asegurando que la fuente de datos seleccionada respalde de manera efectiva el desarrollo del sistema de detección de intrusos basado en machine learning y garantice resultados precisos y confiables.

Tabla 24. Comparativa entre datasets [36], [37], [38].

Aspecto	UNSW-NB15	CSE-CIC-DDoS 2019	CIC-IoV 2024
Año de publicación	2015	2019	2024
Propósito principal	Investigación general en seguridad de redes	Desarrollo de IDS para redes tradicionales y modernas	Desarrollo de IDS enfocados en IoT y vehículos inteligentes
Tipo de ataques incluidos	Ataques tradicionales como flooding y exploits	DDoS basados en reflexión y exploración	Ataques avanzados en IoT y entornos vehiculares
Cobertura de tráfico benigno	Incluye tráfico benigno general	Incluye tráfico benigno detallado y representativo	Incluye tráfico benigno adaptado a IoT
Relevancia para el proyecto	Media, demasiado antiguo para los escenarios actuales	Alta, perfectamente alineado con el objetivo del sistema IDS	Baja, diseñado para un entorno diferente (IoT y autos)
Escenarios específicos abordados	Redes tradicionales	Redes tradicionales y modernas	Redes IoT y vehículos inteligentes
Actualización de datos	Fuera de rango con 9 años de antigüedad	Actualizado y dentro del rango aceptable de 5 años	Muy reciente pero no relevante para este proyecto
Flexibilidad de uso	Limitado a contextos tradicionales	Versátil para redes de mediana y gran escala	Adaptado exclusivamente a IoT y movilidad autónoma

c. Selección de dataset

Luego del análisis de los datasets disponibles, se decidió seleccionar CSE-CIC-DDoS 2019 como fuente de datos principal para este proyecto. Aunque su fecha de publicación no es reciente, permanece dentro del límite aceptable de 5 años para ser utilizado como referencia confiable, a diferencia del dataset UNSW-NB15, cuya antigüedad de 9 años lo hace menos relevante para los escenarios actuales. Además, CSE-CIC-DDoS 2019 se alinea de manera más precisa con el contexto y los objetivos del proyecto, ofreciendo datos que representan de manera adecuada los ataques y el tráfico benigno en redes tradicionales y modernas. En contraste, el dataset CIC-IoV 2024 resulta más apropiado para proyectos centrados en entornos IoT o vehículos inteligentes, áreas que no constituyen el foco principal del estudio. Esta elección refuerza la intención de utilizar un conjunto de datos actualizado, relevante y directamente relacionado con las necesidades específicas de la investigación, maximizando así su impacto en el desarrollo e implementación del sistema de detección de intrusos basado en machine learning.

3.2.3 Fase 3: Preparación de los datos

Una vez seleccionado el dataset CSE-CIC-DDoS 2019 como la fuente de datos más adecuada para el proyecto, se procede a realizar un análisis más profundo y detallado de su contenido, prestando especial atención a las características que lo componen. Este análisis resulta fundamental, ya que permite identificar las variables más relevantes, los posibles valores atípicos y las necesidades específicas de limpieza y transformación de los datos para garantizar su calidad.

a. Carga y validación de los datos

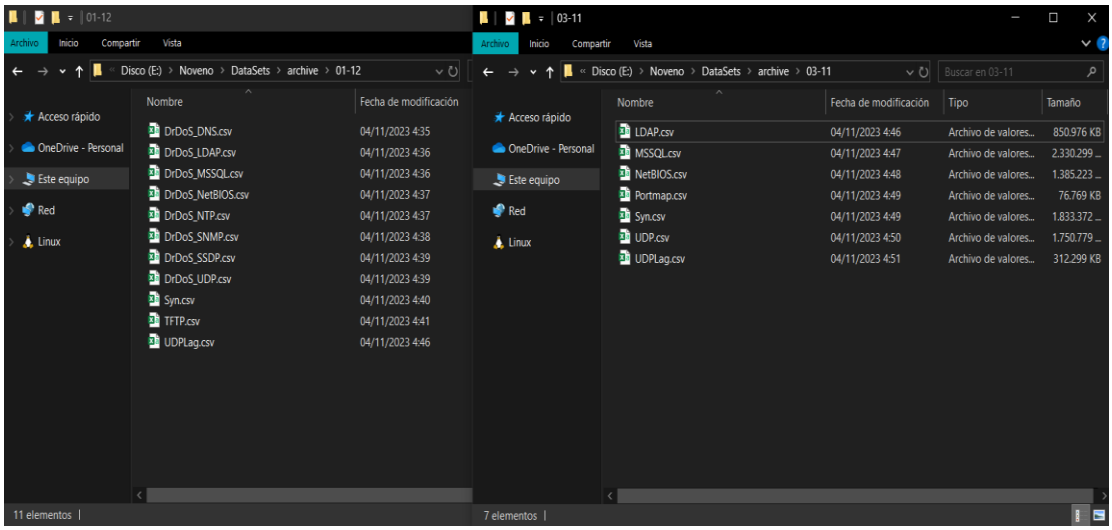


Figura 22. Archivos CSV de dataset CSE-CIC-DDoS 2019.

El dataset CSE-CIC-DDoS 2019 incluye 18 archivos en formato CSV, cada uno representa un tipo específico de ataque dirigido a una red, como se ilustra en la Figura 22. Para entender mejor su estructura y contenido, se realizó un análisis inicial cargando las primeras mil filas de cada archivo. Este enfoque permitió simplificar y agilizar la revisión preliminar, brindando una visión clara y detallada sobre la composición de cada archivo que compone el dataset. Durante este proceso, se recopiló información clave, incluyendo una descripción general de cada archivo, un análisis estadístico detallado, el número de filas y columnas presentes, los tipos de datos asociados a cada columna y una vista preliminar de las primeras y últimas cinco filas de las cargadas. Este análisis permitió sentar las bases para las siguientes etapas del proyecto, asegurando que cada aspecto del dataset sea utilizado de manera eficiente.

```
Información de archivo 1 DrDoS_DNS.csv:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 88 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Unnamed: 0                   1000 non-null   int64
 1   Flow ID                      1000 non-null   object
 2    Source IP                   1000 non-null   object
 3    Source Port                 1000 non-null   int64
 4    Destination IP              1000 non-null   object
 5    Destination Port            1000 non-null   int64
 6    Protocol                    1000 non-null   int64
 7    Timestamp                   1000 non-null   object
 8    Flow Duration               1000 non-null   int64
 9    Total Fwd Packets           1000 non-null   int64
 10   Total Backward Packets      1000 non-null   int64
 11  Total Length of Fwd Packets  1000 non-null   float64
 12   Total Length of Bwd Packets 1000 non-null   float64
 13   Fwd Packet Length Max       1000 non-null   float64
 14   Fwd Packet Length Min       1000 non-null   float64
 15   Fwd Packet Length Mean      1000 non-null   float64
 16   Fwd Packet Length Std       1000 non-null   float64
 17  Bwd Packet Length Max        1000 non-null   float64
 18   Bwd Packet Length Min       1000 non-null   float64
...
 87   Label                       1000 non-null   object
dtypes: float64(45), int64(37), object(6)
memory usage: 687.6+ KB
None
```

Figura 23. Información general de archivo CSV.

```
Descripción estadística de archivo 1 DrDoS_DNS.csv:
        Unnamed: 0   Source Port   Destination Port     Protocol  \
count  1000.000000   1000.000000       1000.000000  1000.000000
mean   1631.263000  20314.686000      21446.700000    14.952000
std    1043.598568  26655.964649      21814.639851     4.331334
min       0.000000      0.000000          0.000000     0.000000
25%     728.750000    634.000000         80.000000    17.000000
50%    1625.500000    634.000000      15534.000000    17.000000
75%    2578.500000  53678.000000      40279.250000    17.000000
max    3430.000000  65378.000000      65506.000000    17.000000

        Flow Duration   Total Fwd Packets   Total Backward Packets  \
count    1.000000e+03         1000.000000              1000.000000
mean     8.553914e+06          115.212000                 3.568000
std      2.880573e+07          101.453743                15.196721
min      0.000000e+00            1.000000                 0.000000
25%      2.643625e+04            2.000000                 0.000000
50%      4.539700e+04          200.000000                 0.000000
75%      7.334475e+04          200.000000                 2.000000
max      1.198196e+08         1014.000000               261.000000

       Total Length of Fwd Packets   Total Length of Bwd Packets  \
count                   1000.000000                   1000.000000
mean                   49090.482000                   1147.782000
...
25%    0.000000e+00     0.000000
50%    0.000000e+00     1.000000
75%    0.000000e+00     1.000000
max    7.976762e+07     1.000000
```

Figura 24. Descripción estadística de archivo CSV.

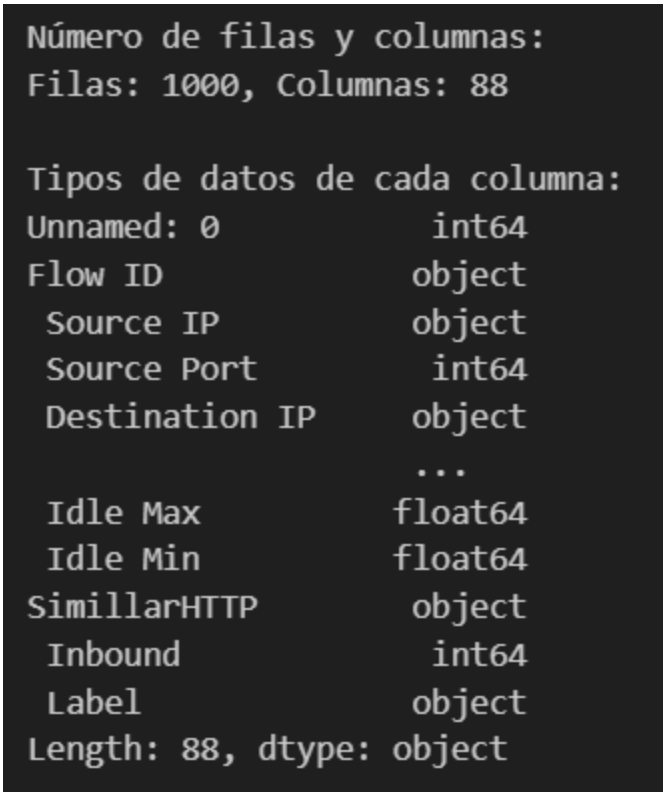


Figura 25. Información de filas y columnas de archivo CSV.

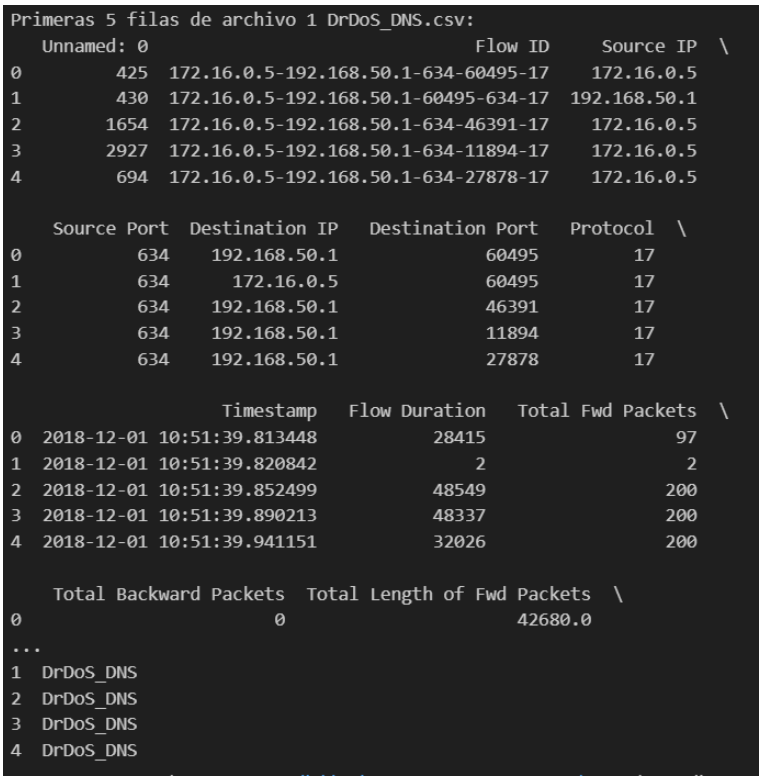


Figura 26. Primeras cinco filas de archivo CSV.

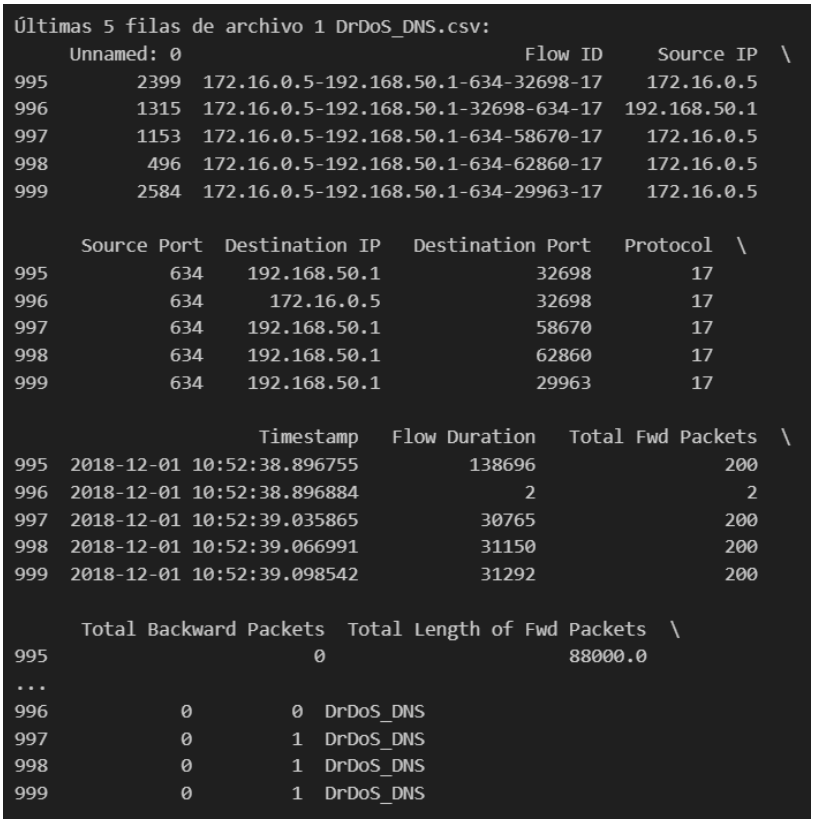


Figura 27. Últimas cinco filas de archivo CSV.

En las Figuras de la 23 a la 27, se presenta un ejemplo del archivo DrDoS_DNS, perteneciente al dataset general CSE-CIC-DDoS 2019, para ilustrar el procedimiento aplicado a cada uno de los archivos que lo conforman. Este método permitió detallar los pasos seguidos para analizar la composición y las características de cada archivo, incluyendo aspectos como la validación de las columnas, la identificación de patrones en los datos y la verificación de posibles inconsistencias. A través de este análisis, se logró una comprensión de la estructura y el contenido del dataset, lo que resultó crucial para definir estrategias de limpieza y transformación adecuadas para etapas posteriores. Además, se pudo identificar que todos los archivos del dataset comparten la misma estructura de datos, lo cual fue un aspecto clave para facilitar el análisis individual y reducir la complejidad del procesamiento. Este hallazgo permitió optimizar el manejo de los datos al garantizar que las mismas metodologías pudieran aplicarse de manera uniforme a todo el conjunto.

b. Filtrado de características

```
Index(['Unnamed: 0', 'Flow ID', ' Source IP', ' Source Port',
       ' Destination IP', ' Destination Port', ' Protocol', ' Timestamp',
       ' Flow Duration', ' Total Fwd Packets', ' Total Backward Packets',
       'Total Length of Fwd Packets', ' Total Length of Bwd Packets',
       ' Fwd Packet Length Max', ' Fwd Packet Length Min',
       ' Fwd Packet Length Mean', ' Fwd Packet Length Std',
       'Bwd Packet Length Max', ' Bwd Packet Length Min',
       ' Bwd Packet Length Mean', ' Bwd Packet Length Std', 'Flow Bytes/s',
       ' Flow Packets/s', ' Flow IAT Mean', ' Flow IAT Std', ' Flow IAT Max',
       ' Flow IAT Min', 'Fwd IAT Total', ' Fwd IAT Mean', ' Fwd IAT Std',
       ' Fwd IAT Max', ' Fwd IAT Min', 'Bwd IAT Total', ' Bwd IAT Mean',
       ' Bwd IAT Std', ' Bwd IAT Max', ' Bwd IAT Min', 'Fwd PSH Flags',
       ' Bwd PSH Flags', ' Fwd URG Flags', ' Bwd URG Flags',
       ' Fwd Header Length', ' Bwd Header Length', 'Fwd Packets/s',
       ' Bwd Packets/s', ' Min Packet Length', ' Max Packet Length',
       ' Packet Length Mean', ' Packet Length Std', ' Packet Length Variance',
       'FIN Flag Count', ' SYN Flag Count', ' RST Flag Count',
       ' PSH Flag Count', ' ACK Flag Count', ' URG Flag Count',
       ' CWE Flag Count', ' ECE Flag Count', ' Down/Up Ratio',
       ' Average Packet Size', ' Avg Fwd Segment Size',
       ' Avg Bwd Segment Size', ' Fwd Header Length.1', 'Fwd Avg Bytes/Bulk',
       ' Fwd Avg Packets/Bulk', ' Fwd Avg Bulk Rate', ' Bwd Avg Bytes/Bulk',
       ' Bwd Avg Packets/Bulk', 'Bwd Avg Bulk Rate', 'Subflow Fwd Packets',
       ' Subflow Fwd Bytes', ' Subflow Bwd Packets', ' Subflow Bwd Bytes',
       'Init_Win_bytes_forward', ' Init_Win_bytes_backward',
       ' act_data_pkt_fwd', ' min_seg_size_forward', 'Active Mean',
       ' Active Std', ' Active Max', ' Active Min', 'Idle Mean', ' Idle Std',
       ' Idle Max', ' Idle Min', 'SimillarHTTP', ' Inbound', ' Label'],
      dtype='object')
```

Figura 28. Columnas del dataset CSE-CIC-DDoS 2019.

Después de realizar un análisis inicial del dataset, se llevó a cabo un estudio más detallado de las características que lo componen. Este paso fue crucial para identificar aquellas que resultaran más relevantes, con el fin de alcanzar un buen equilibrio entre precisión y eficiencia en el modelo. En la Figura 28, se presentan todas las características disponibles en el dataset, de las cuales se seleccionaron:

Flow Duration: Representa la duración total del flujo de datos en milisegundos. Es esencial para entender la temporalidad de las conexiones y detectar comportamientos anómalos como actividades prolongadas o cortas inusuales.

Total Fwd Packets: Número total de paquetes enviados desde el origen al destino. Útil para identificar patrones de comunicación inusuales.

Total Backward Packets: Número total de paquetes enviados desde el destino al origen. Complementa la información de tráfico bidireccional.

Total Length of Fwd Packets: Suma de las longitudes de todos los paquetes enviados en la dirección de origen a destino. Indica la carga transmitida en una dirección.

Total Length of Bwd Packets: Suma de las longitudes de todos los paquetes enviados en la dirección de destino a origen. Ayuda a evaluar la reciprocidad de la comunicación.

Flow Bytes/s: Tasa de bytes transmitidos por segundo en el flujo. Mide la intensidad del tráfico y es clave para detectar posibles ataques de denegación de servicio.

Flow Packets/s: Tasa de paquetes transmitidos por segundo en el flujo. Ayuda a detectar ráfagas de tráfico anómalas.

SYN Flag Count: Cantidad de banderas SYN en el flujo, utilizadas en la inicialización de conexiones TCP. Es fundamental para detectar escaneos de puertos y ataques SYN flood.

ACK Flag Count: Cantidad de banderas ACK en el flujo, que confirman la recepción de paquetes. Su análisis permite identificar actividades irregulares en la comunicación.

RST Flag Count: Cantidad de banderas RST en el flujo, que indican reinicios de conexiones. Es útil para identificar interrupciones anómalas o ataques de reinicio.

Active Mean: Promedio del tiempo durante el cual la conexión está activa. Ayuda a distinguir entre flujos legítimos y comportamientos sospechosos.

Active Std: Desviación estándar del tiempo activo, indicando la variabilidad de la duración de las conexiones activas.

Idle Mean: Promedio del tiempo de inactividad entre paquetes consecutivos en el flujo. Sirve para identificar patrones de pausa inusuales.

Idle Std: Desviación estándar del tiempo de inactividad, para medir la consistencia de los períodos de inactividad.

Min Packet Length: Longitud mínima de los paquetes en el flujo. Útil para identificar tráfico sospechoso con paquetes extremadamente pequeños.

Max Packet Length: Longitud máxima de los paquetes en el flujo. Indica la capacidad máxima de carga por paquete y detecta paquetes irregularmente grandes.

Packet Length Mean: Promedio de la longitud de los paquetes en el flujo. Proporciona un panorama general de la carga promedio.

Label: Indica si el flujo es benigno o malicioso. Es la variable objetivo utilizada para entrenar y evaluar el modelo.

Estas características fueron seleccionadas por su relevancia para identificar y analizar el tráfico de la red, además de su capacidad para distinguir patrones entre actividades normales y maliciosas. La selección se basó en tres aspectos clave:

- **Volumen y temporalidad del tráfico:** Métricas como Flow Duration, Flow Bytes/s y Flow Packets/s permiten comprender la cantidad y velocidad del tráfico en la red. Esto es especialmente útil para detectar ataques como DDoS, que se caracterizan por generar grandes ráfagas de tráfico.

- **Indicadores del protocolo TCP/IP:** Atributos como SYN Flag Count, ACK Flag Count y RST Flag Count son esenciales para identificar comportamientos anómalos en la comunicación de bajo nivel, como escaneos de puertos o interrupciones inesperadas en las conexiones.

- **Longitud y tiempo de los paquetes:** Elementos como Packet Length Mean, Min/Max Packet Length y las métricas de actividad e inactividad (Active/Idle) ofrecen información detallada sobre cómo se comportan los flujos de datos. Esto permite diferenciar entre tráfico legítimo y malicioso.

c. Limpieza de los datos

Después de seleccionar las características más relevantes del dataset, se realizó el proceso de limpieza de los datos. Debido a que el dataset está compuesto por múltiples archivos de gran volumen, fue necesario realizarlo de manera individual para garantizar la consistencia y calidad de la información, cada archivo individual presentó una variación en su tamaño, siendo necesario utilizar dos tipos de algoritmos de limpieza de datos descritos en el Anexo B y Anexo C, para optimizar el proceso.

La limpieza incluyó varias etapas. En primer lugar, se optimizaron los tipos de datos para reducir el uso de memoria, sin comprometer el rendimiento del sistema. Además, se identificaron y trataron valores nulos, duplicados o corruptos, ya que estos pueden afectar negativamente el entrenamiento y la precisión del modelo.

En la Figura 29, se detalla un ejemplo del proceso de limpieza aplicado a uno de los archivos del dataset. Destacan las acciones realizadas para transformar los datos y asegurar que solo se mantuvieran aquellos que aportaran valor al análisis y desarrollo del sistema de detección de intrusos.

```
# Eliminar filas duplicadas
data = data.drop_duplicates()
```

```
# Verificar valores nulos
print("Valores nulos en cada columna:\n", data.isnull().sum())
```

```
Valores nulos en cada columna:
 Flow Duration                0
Total Fwd Packets             0
Total Backward Packets        0
Total Length of Fwd Packets   0
Total Length of Bwd Packets   0
Min Packet Length             0
Max Packet Length             0
Packet Length Mean            0
SYN Flag Count                0
ACK Flag Count                0
RST Flag Count                0
Active Mean                   0
Active Std                    0
Idle Mean                     0
Idle Std                      0
Label                         0
dtype: int64
```

```
# Eliminar filas con valores nulos
data = data.dropna()
```

Figura 29. Filtrado de valores nulos y repetidos de archivos CSV.

d. Normalización y escalado

```
# Definir las columnas en las que se eliminarán outliers, basadas en las columnas seleccionadas
columnas_para_outliers = [
    'Flow Duration', 'Total Fwd Packets', 'Total Backward Packets',
    'Total Length of Fwd Packets', 'Total Length of Bwd Packets',
    'Flow Bytes/s', 'Flow Packets/s'
]

# Aplicar la eliminación de outliers solo en estas columnas
for col in columnas_para_outliers:
    if col in data.columns:  # Verificar que la columna exista en el dataset
        Q1 = data[col].quantile(0.25)
        Q3 = data[col].quantile(0.75)
        IQR = Q3 - Q1
        # Filtrar valores dentro del rango aceptable
        data = data[(data[col] >= Q1 - 1.5 * IQR) & (data[col] <= Q3 + 1.5 * IQR)]

# Imprimir información del dataset después de eliminar los outliers
print(f"Dataset después de eliminar outliers: {data.shape}")
```

```
Dataset después de eliminar outliers: (2329, 16)
```

Figura 30. Eliminación de outliers en archivos CSV.

Para estandarizar el dataset en su conjunto, se aplicó la técnica de eliminación de valores atípicos, conocidos como outliers, en cada uno de los archivos CSV que lo componen. Estos valores, que se desvían significativamente del resto de los datos en la distribución, pueden afectar de manera considerable el rendimiento del modelo. Al eliminarlos, se mejora la generalización del dataset, ya que se suprimen los valores extremos que podrían distorsionar el análisis. Esto permite que el modelo se enfoque en los patrones más representativos, reduciendo el riesgo de sobreajuste y garantizando un mejor desempeño.

Las columnas mostradas en la Figura 30, fueron seleccionadas porque representan aspectos fundamentales del tráfico de red y tienen un impacto significativo tanto en el análisis de los datos como en el entrenamiento del modelo. Estas columnas incluyen métricas relacionadas con la duración de las conexiones, la cantidad de paquetes transmitidos y las tasas de transferencia de datos, aspectos esenciales para comprender y analizar el comportamiento de la red. Estas métricas permiten identificar patrones de tráfico normales, así como detectar anomalías que podrían indicar actividades maliciosas, lo que las convierte en componentes clave.

Para limpiar los datos de valores atípicos, se empleó el método del rango intercuartílico (IQR, por sus siglas en inglés). Este método fue elegido por su robustez frente a outliers extremos, a diferencia de otras técnicas que pueden ser más sensibles a estos valores. El IQR establece un rango basado en los percentiles de los datos, delimitando claramente qué valores se consideran atípicos.

Esto no solo facilita la identificación de outliers, sino que también ofrece la ventaja de ser eficiente en términos computacionales, lo que lo hace especialmente adecuado para datasets de gran volumen. El cálculo de los cuartiles y el posterior filtrado de los datos aseguran que la limpieza sea rápida y efectiva, permitiendo trabajar con un conjunto de datos más representativo y confiable para el entrenamiento del modelo.

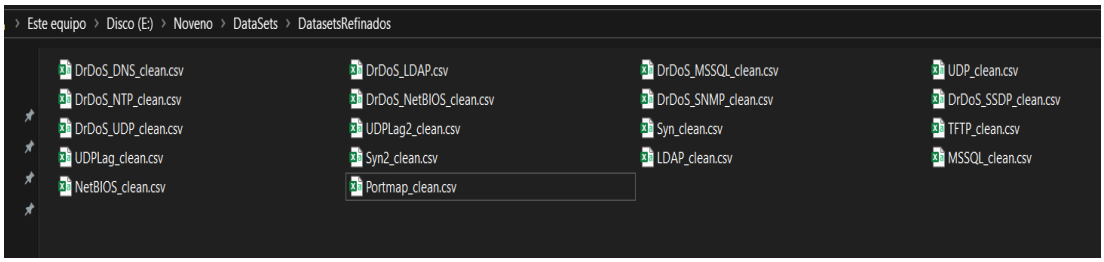


Figura 31. Archivos CSV depurados.

Una vez completado el proceso de filtrado y limpieza de los datos, se procedió a guardar cada uno de los archivos ya depurados. Este paso fue fundamental para garantizar que los datos estuvieran listos para las siguientes etapas del proyecto, incluyendo el tratamiento específico de los datos y el entrenamiento del modelo. En la Figura 31 se muestra un ejemplo del formato y organización final de los archivos limpios, que servirán como base para continuar con las fases posteriores del desarrollo del proyecto.

3.2.4 Fase 4: Modelado

a. Selección de algoritmo de machine learning

La fase de modelado comienza con la selección del algoritmo de machine learning que mejor se adapte a las características y limitaciones del proyecto. Para ello, se realizó una evaluación basada en una ficha técnica presentada en la Tabla 25. Como punto de

partida, se tomó en cuenta el estudio realizado por Malele y Mathonsi [39], en el que se evaluó el desempeño de diversos algoritmos de machine learning en entornos que emplean servidores IDS. En dicho estudio, se concluyó que el algoritmo con el mejor rendimiento fue Random Forest, seguido de XGBoost (XGB) y el algoritmo K-Nearest Neighbors (KNN). Con base en estos resultados, se decidió trabajar inicialmente con estos algoritmos, evaluando cuál de ellos se ajusta de manera más efectiva al contexto y limitaciones del proyecto.

Tabla 25. Ficha técnica comparativa para algoritmos de machine learning [39].

Atributo	XGBoost (eXtreme Gradient Boosting)	Random Forest	K-Nearest Neighbors (KNN)
Tipo de algoritmo	Boosting basado en Árboles de Decisión.	Ensemble Learning (Bagging) basado en Árboles de Decisión.	Basado en proximidad; método no paramétrico.
Hiperparámetros clave	Tasa de aprendizaje (learning_rate). Número de árboles (n_estimators). Regularización (lambda, alpha).	Número de árboles (n_estimators). Máxima profundidad (max_depth). Muestras mínimas para dividir (min_samples_split).	Número de vecinos (kkk). Métrica de distancia (euclidiana, Manhattan, Minkowski).
Manejo de overfitting	Eficaz gracias a la regularización (L1L1L1 y L2L2L2) y la poda de árboles, pero depende de una configuración precisa.	Muy robusto: el promedio de múltiples árboles reduce la varianza y evita el sobreajuste, incluso en datasets complejos.	Sensible al tamaño de kkk: valores pequeños pueden llevar a sobreajuste, valores grandes a subajuste.
Escalabilidad	Muy alta: escalable, pero requiere mayor capacidad de hardware debido a su complejidad iterativa y consumo de memoria.	Alta: paralelizable y eficiente en hardware limitado, lo que lo hace ideal para grandes datasets en sistemas con recursos moderados.	Baja en datasets grandes, ya que necesita calcular distancias para cada muestra, lo que afecta su escalabilidad.
Requerimiento de datos	Requiere preprocesamiento cuidadoso, como manejo de valores nulos y, en algunos casos, normalización.	Funciona bien con datos categóricos y continuos. No requiere normalización previa, lo que simplifica el preprocesamiento.	Necesita datos normalizados o estandarizados para evitar sesgos en la métrica de distancia.
Tiempo de entrenamiento	Alto, especialmente en datasets grandes debido a su naturaleza iterativa y la necesidad de ajustar muchos hiperparámetros.	Moderado: más rápido que XGBoost en sistemas con hardware limitado, manteniendo un buen balance entre velocidad y precisión.	Muy bajo, ya que no construye un modelo explícito (lazy learning).
Tiempo de predicción	Moderado, porque los árboles se evalúan en serie, lo que puede ralentizar predicciones en grandes volúmenes de datos.	Rápido: cada árbol se evalúa de forma independiente, lo que lo hace eficiente incluso en escenarios con alta demanda de predicciones.	Alto en datasets grandes, debido al cálculo intensivo de distancias para cada predicción.

Atributo	XGBoost (eXtreme Gradient Boosting)	Random Forest	K-Nearest Neighbors (KNN)
Tolerancia al ruido	Alta, pero depende de la correcta selección de hiperparámetros.	Alta: el uso de múltiples árboles suaviza el impacto del ruido en los datos, permitiendo generalizaciones más confiables.	Baja: altamente sensible a datos ruidosos y desbalanceados.
Ventajas principales	Alta precisión en competiciones de datos. Optimización en memoria y paralelismo, pero requiere hardware robusto.	Rápido y eficiente en hardware limitado. Robusto frente a ruido y outliers. Ideal para datasets grandes. Balancea velocidad y precisión sin sacrificar desempeño.	Intuitivo y fácil de implementar. Útil en problemas no paramétricos y en datasets pequeños.
Desventajas principales	Altamente dependiente de hiperparámetros. Entrenamiento lento en hardware limitado.	Puede consumir más memoria en datasets muy grandes. Menos interpretable debido al uso de múltiples árboles.	Sensible al ruido y desbalanceo. Escalabilidad limitada en datasets grandes.

Tras analizar las características de los algoritmos, se optó por utilizar Random Forest debido a su capacidad para equilibrar precisión, robustez y eficiencia, lo que lo convierte en una opción ideal para su integración con un servidor IDS, especialmente en entornos con limitaciones de hardware. Este algoritmo destaca por su capacidad para manejar grandes volúmenes de datos, permitiendo un entrenamiento y predicción a buena velocidad, incluso con recursos computacionales moderados.

Además, su robustez frente al ruido y los outliers complementa perfectamente la etapa de limpieza de datos realizada previamente, garantizando que el modelo pueda generalizar mejor y evitar sesgos causados por valores atípicos. Otra ventaja significativa de Random Forest es su facilidad de implementación, ya que no requiere un preprocesamiento complejo ni ajustes complicados de los datos, lo que simplifica el flujo de trabajo. Estas características lo posicionan como una solución confiable y eficiente que se adapta al proyecto, maximizando el rendimiento del sistema sin comprometer la viabilidad técnica.

b. Configuración del entorno de desarrollo

Después de seleccionar el dataset CSE-CIC-DDoS 2019 y determinar que el algoritmo de machine learning a emplear sería Random Forest, se procedió a establecer las herramientas necesarias para el entorno de desarrollo. Como entorno principal se optó

por Jupyter Notebook, debido a su flexibilidad para integrar la implementación del código, realizar pruebas interactivas y documentar los procesos de forma clara y estructurada. Esta elección permitió un flujo de trabajo más dinámico y eficiente, especialmente en las etapas de preprocesamiento y modelado.

Para el desarrollo del modelo, los archivos CSV que conforman el dataset se unieron en un único conjunto de datos general utilizando el algoritmo expuesto en el Anexo D, consolidando toda la información tras un proceso inicial de limpieza individual. Esta fuente de datos general fue sometida a un último preprocesamiento, siguiendo las técnicas descritas en la fase de preparación de los datos, asegurando que estuviera completamente listo para ser utilizado la fase de modelado.

Python fue elegido para implementar el algoritmo de machine learning ya que es un lenguaje ampliamente adoptado en el ámbito de la ciencia de datos y aprendizaje automático. Su extensa colección de bibliotecas especializadas, como Scikit-learn para modelos de aprendizaje y Pandas para manipulación de datos. Además, su simplicidad y versatilidad lo convierten en una herramienta ideal para proyectos de machine learning, especialmente en entornos educativos y profesionales.

En cuanto a las herramientas para poder complementar el flujo de trabajo en la implementación del algoritmo, se optó por usar las bibliotecas:

- **Pandas:** Para la manipulación y análisis del dataset.
- **Scikit-learn:** Para la implementación del modelo Random Forest y la preparación de datos.
- **Imbalanced-learn:** Para técnicas de balanceo de clases.
- **Joblib:** Para guardar y cargar modelos.

c. Entrenamiento inicial y optimización de hiperparámetros

Para el entrenamiento inicial del modelo, se definieron ciertas características como la variable objetivo, encargada de clasificar el tráfico como benigno o malicioso. Además el dataset se dividió en datos de entrenamiento (70%) y datos de prueba (30%), como se indica en la Figura 32.

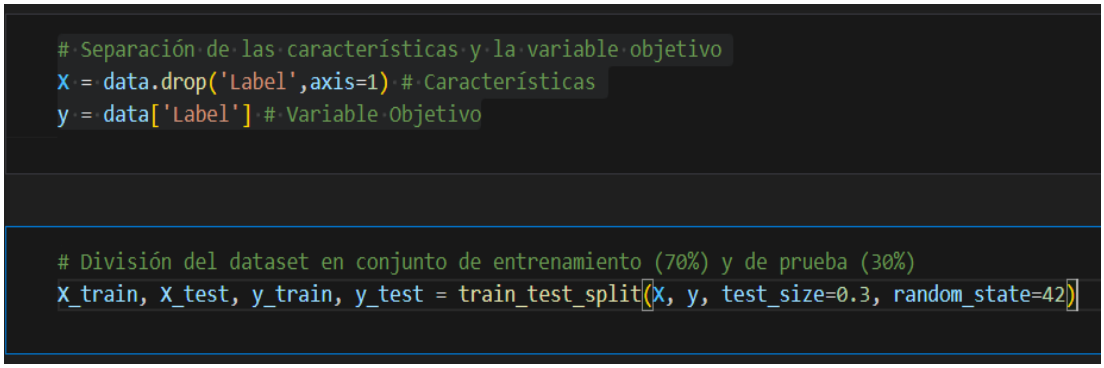


Figura 32. Pasos iniciales de entrenamiento del modelo.

Después de completar las configuraciones iniciales, se procedió con el entrenamiento del modelo, comenzando con una configuración básica que permitió establecer una línea base de rendimiento. Esta etapa inicial es fundamental para obtener un primer resultado del modelo y evaluar su comportamiento con los parámetros predeterminados. Durante este proceso, se busca identificar posibles áreas de mejora y comprender cómo responde el modelo a los datos proporcionados.

Aunque estas configuraciones iniciales no están optimizadas, ofrecen un punto de partida sólido que sirve como referencia para comparar los resultados tras el ajuste de hiperparámetros. Posteriormente, se realizaron ajustes más avanzados para perfeccionar el modelo y maximizar su capacidad de generalización y precisión. Este enfoque progresivo asegura un desarrollo más estructurado y eficiente del modelo.

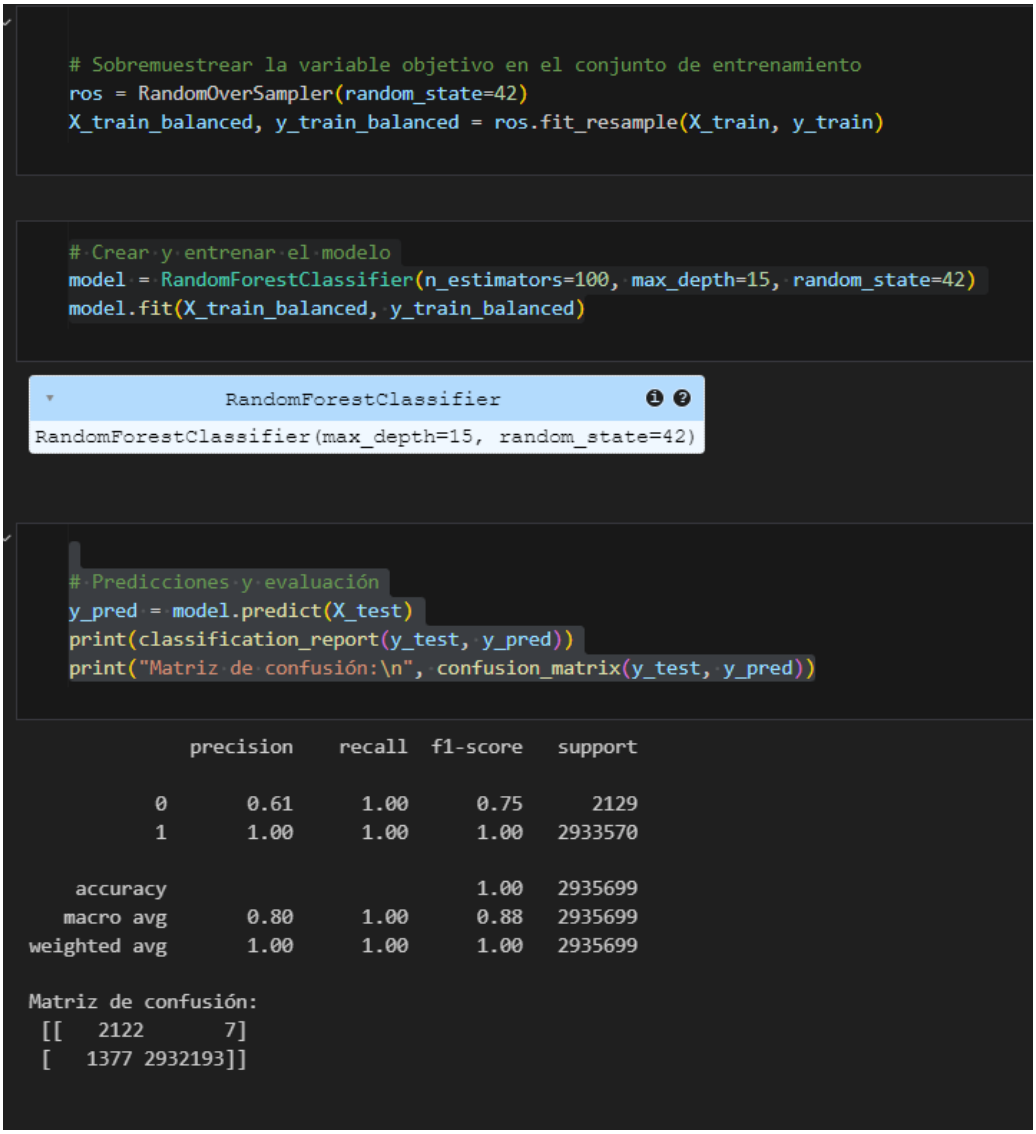


Figura 33. Entrenamiento inicial del modelo.

Como se muestra en la Figura 33, el entrenamiento inicial del modelo se realizó utilizando el algoritmo Random Forest con una configuración básica de hiperparámetros, específicamente 100 árboles (n_estimators=100) y una profundidad máxima de 15 (max_depth=15). Para abordar el desbalance en la variable objetivo, se aplicó la técnica de sobremuestreo aleatorio (RandomOverSampler), equilibrando las clases en el conjunto de entrenamiento, lo que permitió al modelo entrenarse de manera efectiva tratando de evitar el sesgo hacia la clase mayoritaria.

Tras entrenar el modelo con el conjunto balanceado, se evaluó su desempeño en el conjunto de prueba, obteniendo métricas significativas. Los resultados muestran un

F1-Score ponderado de 1.00 y una precisión general del 100%, lo que indica que el modelo logró clasificar casi perfectamente las instancias en el conjunto de prueba. Sin embargo, al observar la matriz de confusión, se puede notar que, aunque la mayoría de las instancias fueron clasificadas correctamente, hubo 7 falsos positivos (clase maliciosa clasificada como benigna) y 1377 falsos negativos (clase benigna clasificada como maliciosa). Esto evidencia que, aunque el modelo tiene un rendimiento sobresaliente, su capacidad para manejar clases minoritarias podría beneficiarse de un ajuste más preciso de los hiperparámetros, considerando la primera prueba del modelo como fallida.

```python
# Optimización del modelo con búsqueda de hiperparámetros
# Definir parámetros de búsqueda
param_grid = {
    'n_estimators': [100, 150],
    'max_depth': [10, None],
    'min_samples_split': [2, 5]
}

random_search = RandomizedSearchCV(
    RandomForestClassifier(random_state=42),
    param_distributions=param_grid,
    n_iter=8,  # Número de combinaciones a probar
    cv=3,
    scoring='f1_weighted',
    n_jobs=1,
    random_state=42
)

random_search.fit(X_train_balanced, y_train_balanced)

# Obtener el mejor modelo y parámetros
best_model = random_search.best_estimator_
print("Mejores hiperparámetros:", random_search.best_params_)
```

```
Mejores hiperparámetros: {'n_estimators': 100, 'min_samples_split': 2, 'max_depth': None}
```

Figura 34. Búsqueda de mejores hiperparámetros.

Para optimizar el desempeño del modelo y ajustarlo a las necesidades específicas del proyecto, se implementó una búsqueda de hiperparámetros utilizando el método RandomizedSearchCV, como se muestra en la Figura 34. Este enfoque, ampliamente empleado en machine learning, permite explorar combinaciones aleatorias de

hiperparámetros dentro de un rango predefinido, reduciendo significativamente el tiempo de procesamiento en comparación con una búsqueda extendida. Además, este método optimiza el uso de recursos computacionales, haciendo que el proceso sea más eficiente y manejable.

En el caso del algoritmo Random Forest, se seleccionaron tres hiperparámetros para su ajuste: el número de árboles en el bosque (n_estimators), la profundidad máxima de los árboles (max_depth) y el número mínimo de muestras necesarias para dividir un nodo (min_samples_split). Al finalizar el proceso, se identificaron los hiperparámetros óptimos: 100 árboles (n_estimators=100), sin límite en la profundidad máxima (max_depth=None) y un mínimo de 2 muestras para dividir un nodo (min_samples_split=2). Estos parámetros reflejan un modelo bien ajustado, capaz de capturar patrones complejos en los datos sin comprometer su capacidad de generalización. Este garantiza un buen desempeño del modelo para las etapas posteriores.

3.2.5 Fase 5: Evaluación

a. Definición de las métricas de evaluación

La selección de métricas de evaluación como precisión, F1-Score y recall responde a la necesidad de analizar a fondo el desempeño del modelo en un contexto donde el tráfico malicioso pertenece a la Clase 1 y el tráfico benigno a la Clase 0. Estas métricas fueron escogidas para proporcionar una evaluación integral y equilibrada del modelo:

Precisión (Precision): Evalúa la proporción de predicciones positivas que son correctas. Es una métrica crucial para minimizar los falsos positivos, especialmente en la Clase 1 (tráfico malicioso). Un alto número de falsos positivos puede generar alertas innecesarias, reduciendo la eficiencia y confiabilidad del sistema de detección.

Recall (Sensibilidad): Mide qué proporción de las instancias positivas reales fueron correctamente identificadas por el modelo. Es particularmente relevante para la Clase 0 (tráfico benigno), ya que un recall bajo significaría un número elevado de falsos negativos, clasificando tráfico benigno como malicioso. Esto podría impactar negativamente la experiencia del usuario, al generar un exceso de alertas incorrectas.

F1-Score: Combina precisión y recall en una única métrica que equilibra ambas dimensiones. Es especialmente útil en datasets desbalanceados, donde priorizar una métrica sobre la otra podría conducir a conclusiones sesgadas. El F1-Score proporciona una evaluación más completa y justa del modelo.

Matriz de confusión: Complementa estas métricas al ofrecer una representación visual clara del desempeño del modelo, mostrando las predicciones correctas y los errores (falsos positivos y falsos negativos) para ambas clases. Esto permite un análisis detallado de las áreas en las que el modelo puede mejorar.

Después de evaluar el modelo mediante una ficha de aceptación y confirmar que cumple con un desempeño superior al 85% en todas sus métricas clave, se procederá a realizar una validación cruzada. Este umbral del 85% se fundamenta en lo señalado por Rababah y Srivastava [40], quienes demostraron, en un caso exitoso relacionado con modelos de machine learning, que un rendimiento igual o superior a este porcentaje es suficiente para garantizar una operación efectiva en aplicaciones críticas como los sistemas de detección de intrusos (IDS).

Este nivel asegura un balance adecuado entre precisión, sensibilidad y F1-Score, minimizando falsos positivos y negativos, aspectos cruciales para la confiabilidad del sistema. Además, superar este estándar permite al IDS operar de manera eficiente en entornos reales, donde una detección oportuna y precisa es esencial para mitigar riesgos de seguridad. Por lo tanto, no alcanzar este umbral comprometería tanto la capacidad del sistema para detectar amenazas como su funcionalidad en escenarios prácticos.

La validación cruzada, por su parte, será esencial para evaluar la estabilidad del modelo. Este proceso permite probar el modelo en particiones diferentes del dataset, confirmando que los resultados no dependen de una única muestra de datos y garantizando su capacidad de generalización a nuevos escenarios. Además, se establece que el desempeño promedio del modelo durante la validación cruzada también debe superar el umbral del 85%. Esto asegura que el modelo no solo cumpla con los estándares iniciales de aceptación, sino que sea lo suficientemente robusto y confiable para su implementación práctica en un sistema de detección de intrusos.

b. Evaluación general del modelo

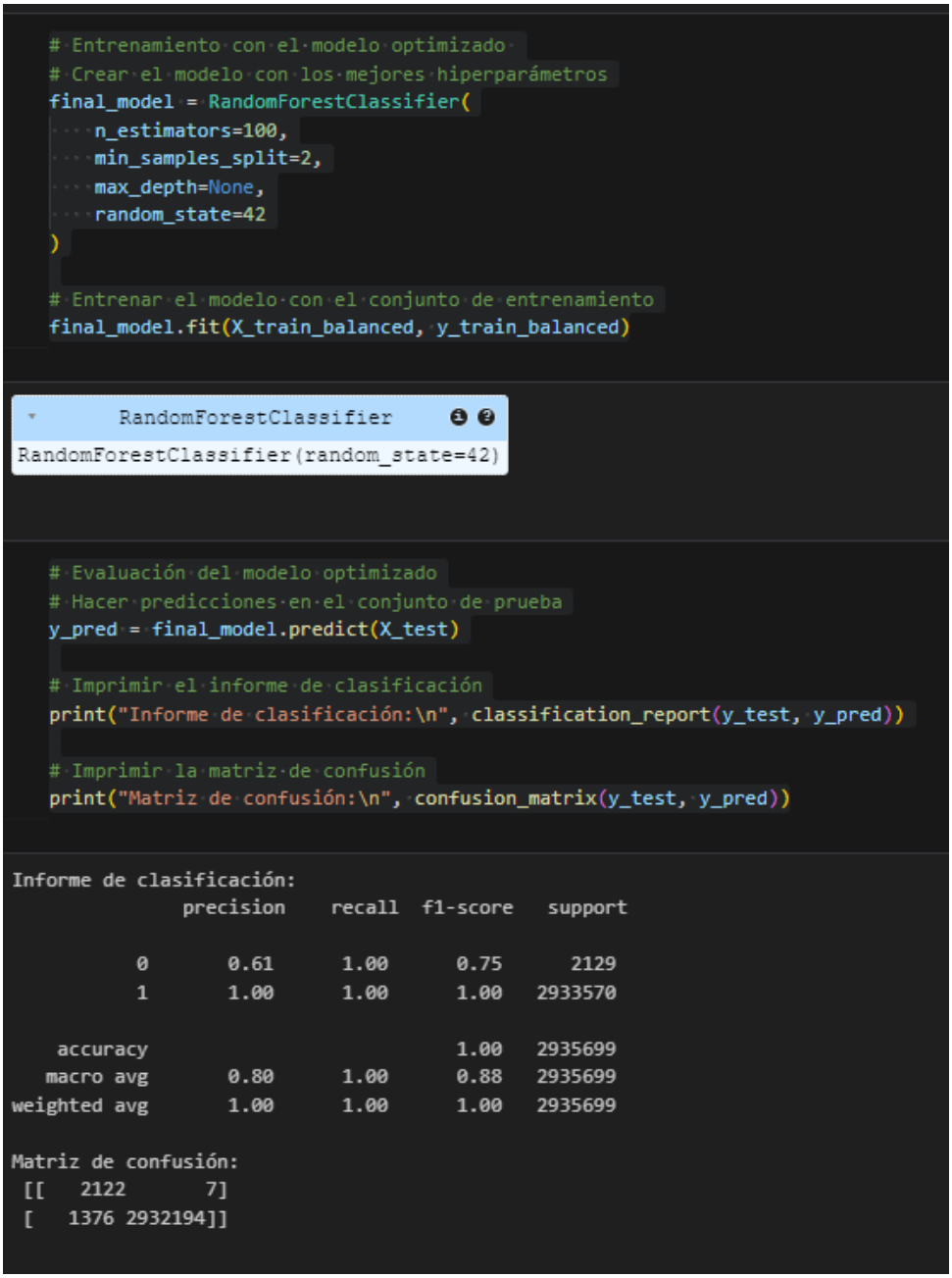


Figura 35. Primera evaluación del modelo con los mejores hiperparámetros.

Una vez identificados los mejores hiperparámetros para el modelo mediante el proceso de optimización y definidas las métricas de evaluación, se procedió a realizar un nuevo entrenamiento, esta vez incorporando los valores ajustados que maximizan el desempeño del algoritmo como se indica en la Figura 35. El objetivo principal de este

entrenamiento fue obtener una primera prueba de aceptación, evaluando métricas clave establecidas como precisión, sensibilidad y F1-Score, lo que permitió verificar si el modelo cumplía con las expectativas establecidas para el sistema de detección de intrusos (IDS).

Tabla 26. Ficha de aceptación prueba 01 en base a resultados obtenidos en Figura 35.

Aspecto evaluado	Valor obtenido	Criterio de aceptación	Resultado
Precisión (Clase 0 - Benigno)	61%	≥ 85%	No aceptado
Precisión (Clase 1 - Malicioso)	100%	≥ 85%	Aceptado
F1-Score (Clase 0 - Benigno)	75%	≥ 85%	No aceptado
F1-Score (Clase 1 - Malicioso)	100%	≥ 85%	Aceptado
Sensibilidad (Recall - Clase 0)	100%	≥ 85%	Aceptado
Sensibilidad (Recall - Clase 1)	100%	≥ 85%	Aceptado
Precisión general (Accuracy)	100%	≥ 85%	Aceptado
Matriz de confusión	[[2122, 7], [1376, 2932194]]	Mínimos falsos positivos y negativos	No aceptado
Resultado final de la prueba		No aprobado	

A pesar de que el modelo mostró un desempeño sobresaliente en la clasificación de tráfico malicioso (Clase 1), alcanzando una precisión, sensibilidad y F1-Score alto, su capacidad para detectar tráfico benigno (Clase 0) fue considerablemente menor, como se observa en la Tabla 26. En particular, la precisión para la Clase 0 fue del 61%, muy por debajo del umbral mínimo de aceptación del 85%. Esto significa que una proporción significativa de las predicciones para el tráfico benigno son incorrectas, lo que compromete la efectividad del modelo en escenarios reales. Este bajo desempeño pone en evidencia una debilidad importante, ya que la detección de tráfico benigno es crucial para evitar alarmas innecesarias en un sistema de detección de intrusos (IDS).

El análisis de la matriz de confusión confirma este problema, mostrando un total de 1376 falsos negativos, lo que equivale a clasificar tráfico benigno como malicioso. Este tipo de error es especialmente crítico, ya que puede generar un volumen elevado de alertas falsas, afectando tanto la confiabilidad como la utilidad práctica del sistema. Por estas razones, esta prueba no puede considerarse exitosa, ya que el modelo no cumple con los requisitos mínimos establecidos para la detección de tráfico benigno,

resaltando la necesidad de realizar ajustes adicionales para equilibrar su desempeño entre ambas clases.

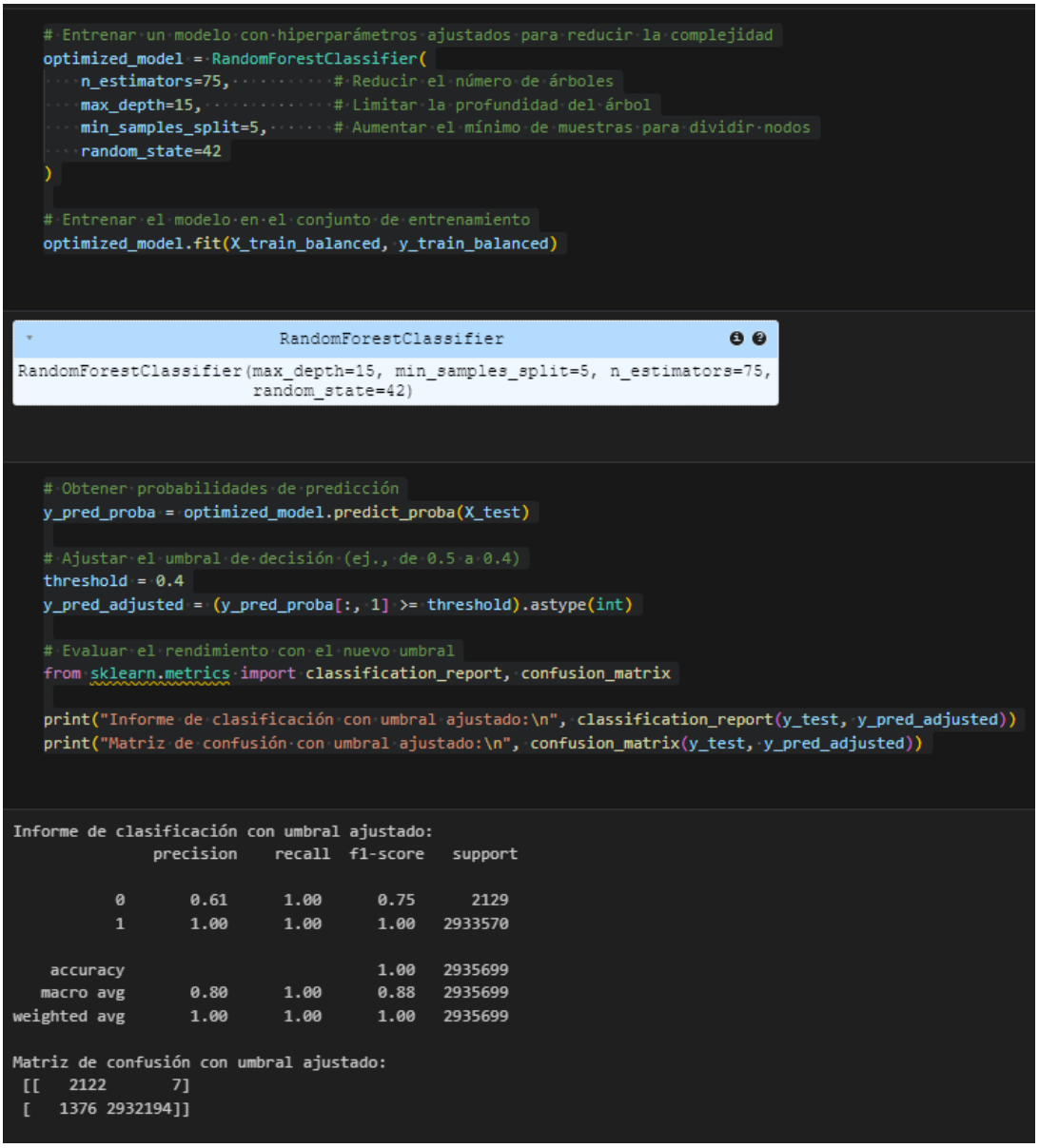


Figura 36. Segunda evaluación del modelo con hiperparámetos ajustados.

La Figura 36 muestra la ejecución de una nueva prueba, donde se ajustaron los hiperparámetros del modelo Random Forest para reducir su complejidad y optimizar el uso de memoria. Estos ajustes fueron realizados con el objetivo de mejorar el desempeño general del modelo y lograr un equilibrio adecuado entre precisión y capacidad de generalización. Las modificaciones realizadas incluyen:

Número de árboles (n_estimators=75): Se redujo el número de árboles en el bosque a 75, tomando como referencia el estudio de Bartz, Zaeffer, Mersmann y Bartz-Beielstein [40], que destaca la importancia de ajustar los hiperparámetros para equilibrar precisión y eficiencia computacional. Este valor fue seleccionado porque, aunque es menor que configuraciones más altas (como 100 o 150 árboles), 75 es suficiente para capturar patrones relevantes en los datos. Además, esta configuración optimiza el consumo de memoria y acelera el tiempo de entrenamiento, aspectos críticos al trabajar con recursos computacionales limitados.

Profundidad máxima (max_depth=15): Limitar la profundidad de los árboles a 15 se fundamenta en el análisis de estrategias de ajuste de hiperparámetros discutidas por Bartz, Zaeffer, Mersmann y Bartz-Beielstein [40]. Este valor representa un equilibrio entre complejidad y generalización, lo que permite al modelo capturar patrones significativos sin caer en el sobreajuste. Según el estudio, controlar la profundidad de los árboles es esencial para evitar que el modelo se adapte excesivamente a los datos de entrenamiento, favoreciendo su capacidad para generalizar a nuevos datos, una característica indispensable para sistemas de detección de intrusos.

Muestras mínimas para dividir nodos (min_samples_split=5): El ajuste del número mínimo de muestras necesarias para dividir un nodo a 5 se basó en las recomendaciones de Bartz, Zaeffer, Mersmann y Bartz-Beielstein [40], quienes enfatizan la importancia de configurar este hiperparámetro para reducir la complejidad del modelo y prevenir divisiones innecesarias. Este valor asegura que cada división esté respaldada por una cantidad suficiente de datos, minimizando el riesgo de que el modelo aprenda patrones erróneos o ruido presente en los datos. Este enfoque también mejora la eficiencia computacional y la estabilidad del modelo.

La configuración de estos hiperparámetros no fue arbitraria, sino que siguió las estrategias propuestas por Bartz, Zaeffer, Mersmann y Bartz-Beielstein [40], analizan cómo ajustar los hiperparámetros en modelos de árboles de decisión como Random Forest. Cada valor seleccionado busca lograr un balance entre precisión, generalización y eficiencia computacional, adaptándose a las características específicas del problema y a las limitaciones de los recursos disponibles.

Tabla 27. Ficha de aceptación prueba 02 en base a resultados obtenidos en Figura 35.

Aspecto evaluado	Valor obtenido	Criterio de aceptación	Resultado
Precisión (Clase 0 - Benigno)	61%	≥ 85%	No aceptado
Precisión (Clase 1 - Malicioso)	100%	≥ 85%	Aceptado
F1-Score (Clase 0 - Benigno)	75%	≥ 85%	No aceptado
F1-Score (Clase 1 - Malicioso)	100%	≥ 85%	Aceptado
Sensibilidad (Recall - Clase 0)	100%	≥ 85%	Aceptado
Sensibilidad (Recall - Clase 1)	100%	≥ 85%	Aceptado
Precisión general (Accuracy)	100%	≥ 85%	Aceptado
Matriz de confusión	[[2122, 7], [1376, 2932194]]	Mínimos falsos positivos y negativos	No aceptado
Resultado final de la prueba		No aprobado	

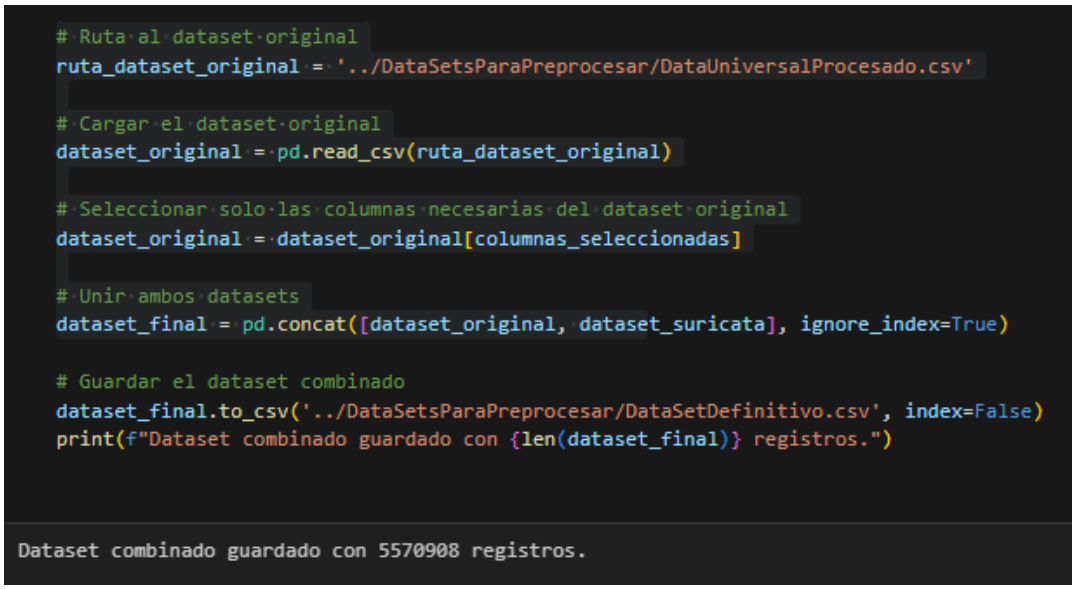
La ficha de aceptación mostrada en la Tabla 27 indica que, aunque el modelo ajustado mostró mejoras significativas en términos de eficiencia computacional, su desempeño en la detección de tráfico benigno (Clase 0) sigue siendo insuficiente. La precisión para la Clase 0 permanece en un 61%, muy por debajo del criterio mínimo de aceptación del 85%. Esto indica que el modelo aún clasifica una proporción significativa de tráfico benigno como malicioso, lo que podría generar un alto número de falsos positivos en un entorno real.

La matriz de confusión respalda esta observación, ya que aún se presentan 1376 falsos negativos, lo que refleja la persistencia del problema. Este resultado evidencia que los ajustes realizados, aunque efectivos en términos de eficiencia y complejidad del modelo, no solucionaron las limitaciones fundamentales en el manejo de la clase minoritaria. Por tanto, esta prueba no puede considerarse exitosa y se requiere un análisis adicional para equilibrar el desempeño entre ambas clases.

```
Distribución de clases en el conjunto de entrenamiento:
Label
1    6845342
0       4622
Name: count, dtype: int64
Distribución de clases en el conjunto de prueba:
Label
1    2933570
0       2129
Name: count, dtype: int64
```

Figura 37. Distribución de clases en el dataset.

Tras obtener resultados fallidos en dos pruebas, incluso después de realizar ajustes en los hiperparámetros, se decidió llevar a cabo un análisis de la fuente de datos utilizada. Como se muestra en la Figura 37, el dataset presenta un desbalance significativo entre las clases. La cantidad de instancias correspondientes a la Clase 1 (tráfico malicioso) es considerablemente mayor, con 6,845,342 registros en el conjunto de entrenamiento, frente a solo 4,622 registros de la Clase 0 (tráfico benigno). Este desbalance se mantiene en el conjunto de prueba, donde la Clase 1 cuenta con 2,933,570 registros, mientras que la Clase 0 tiene apenas 2,129, como se puede apreciar en la Figura 37.



```
# Ruta al dataset original
ruta_dataset_original = '../DataSetsParaPreprocesar/DataUniversalProcesado.csv'

# Cargar el dataset original
dataset_original = pd.read_csv(ruta_dataset_original)

# Seleccionar solo las columnas necesarias del dataset original
dataset_original = dataset_original[columnas_seleccionadas]

# Unir ambos datasets
dataset_final = pd.concat([dataset_original, dataset_suricata], ignore_index=True)

# Guardar el dataset combinado
dataset_final.to_csv('../DataSetsParaPreprocesar/DataSetDefinitivo.csv', index=False)
print(f"Dataset combinado guardado con {len(dataset_final)} registros.")
```

```
Dataset combinado guardado con 5570908 registros.
```

Figura 38. Unión de información complementaria a dataset principal.

Para abordar el desbalance de clases presente en el dataset, se decidió recopilar datos adicionales de la clase minoritaria (tráfico benigno) durante un período de una semana, utilizando un software de monitoreo de tráfico para capturar información real de la red. Este software permitió registrar patrones diversos y representativos del tráfico benigno, asegurando la calidad y relevancia de los datos recolectados.

Posteriormente, estos datos se integraron al dataset general con el objetivo de complementar la información existente y equilibrar la proporción entre las clases como se indica en la Figura 38, también como filtro se estableció un algoritmo de preprocesamiento final descrito en Anexo F. Este enfoque no solo incrementó la representación de la clase minoritaria, sino que también mejoró la diversidad y calidad del dataset, permitiendo al modelo aprender patrones más representativos. Al

enriquecer los datos con información real y actualizada, se busca optimizar la capacidad del modelo para generalizar y minimizar los problemas derivados del desbalance inicial, fortaleciendo así su desempeño en la detección tanto de tráfico benigno como malicioso en escenarios reales.

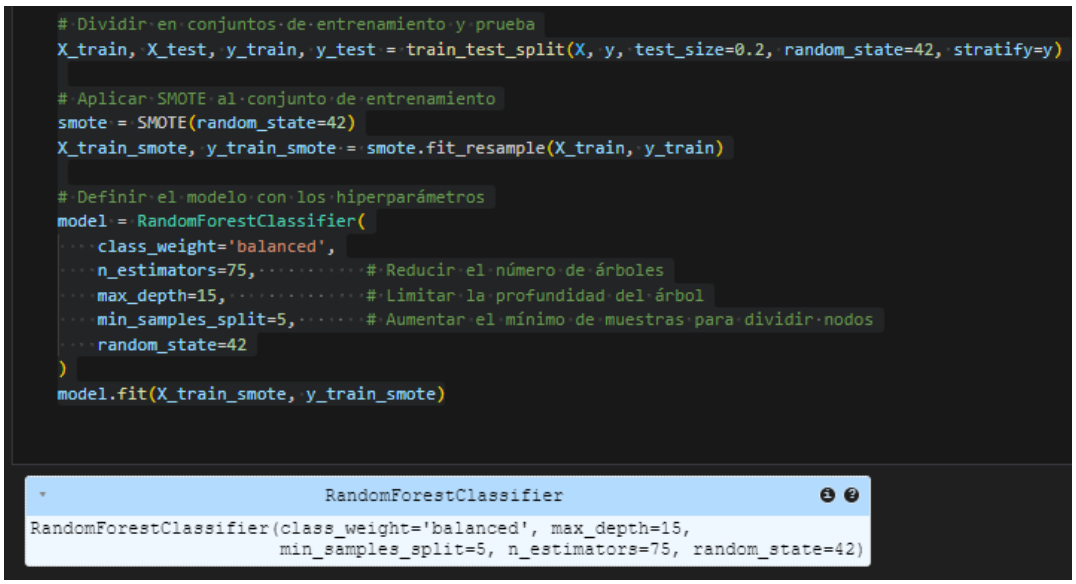


Figura 39. Entrenamiento del modelo utilizando la técnica Synthetic Minority Over-sampling Technique (SMOTE).

Como complemento a la recolección de datos realizada para abordar el desbalance de clases, se implementó la técnica SMOTE (Synthetic Minority Over-sampling Technique), como se muestra en la Figura 39 y en el código descrito en Anexo G. SMOTE es una herramienta ampliamente utilizada en machine learning para equilibrar datasets desbalanceados mediante la generación de ejemplos sintéticos de la clase minoritaria.

A diferencia de estrategias más tradicionales, como la duplicación de datos, SMOTE crea nuevas instancias al interpolar entre puntos existentes de la clase minoritaria. Esto no solo incrementa la cantidad de datos de la clase minoritaria, sino que también mejora su diversidad, permitiendo al modelo aprender patrones más representativos y relevantes.

Esta técnica es especialmente valiosa porque mitiga el sesgo hacia la clase mayoritaria, mejora la precisión en la detección de la clase minoritaria y reduce el riesgo de sobreajuste, al evitar la redundancia que podría surgir al duplicar registros. Además,

SMOTE potencia el rendimiento de algoritmos como Random Forest, ayudando a que estos generalicen mejor en datos no vistos, lo que resulta esencial para garantizar la eficacia del sistema de detección.

En la tercera prueba del modelo, como se detalla en la Figura 38, se optó por dividir los datos en un 80% para entrenamiento y un 20% para evaluación. Esta proporción no solo es comúnmente aceptada en el ámbito de machine learning, sino que también ha sido respaldada por investigaciones como la de Joseph y Vakayil [41], quienes destacaron que una división de este tipo permite evaluar el rendimiento del modelo de manera eficiente, asegurando una estimación confiable de los errores de generalización.

Al reservar un 20% de los datos para pruebas, se garantiza que el conjunto de evaluación refleje adecuadamente la distribución del conjunto de datos completo, mientras que el 80% destinado al entrenamiento maximiza la cantidad de información disponible para que el modelo aprenda patrones relevantes. Además, en esta prueba se reutilizaron los hiperparámetros definidos anteriormente y se integró el parámetro class_weight='balanced', que ajusta automáticamente los pesos de las clases para mitigar el impacto de cualquier desbalance residual en los datos. Para el entrenamiento, se utilizó el conjunto de datos balanceado generado por SMOTE (X_train_smote y y_train_smote), combinando tanto datos originales como sintéticos previamente generados.

La orientación combinada de recolección de datos adicionales y el uso de SMOTE permitió mitigar significativamente el desbalance del dataset general, enriqueciendo la calidad del entrenamiento del modelo. Al hacerlo, se busca que el modelo sea capaz de identificar de manera más precisa tanto tráfico benigno como malicioso, mejorando su desempeño en escenarios reales y fortaleciendo su confiabilidad.

```python
# Predicciones en el conjunto de prueba
y_pred = model.predict(X_test)

# Informe de clasificación
print("Informe de clasificación:")
print(classification_report(y_test, y_pred))

# Matriz de confusión
print("Matriz de confusión:")
print(confusion_matrix(y_test, y_pred))
```

```
Informe de clasificación:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97       616
           1       1.00      1.00      1.00    209953

    accuracy                           1.00    210569
   macro avg       0.97      1.00      0.98    210569
weighted avg       1.00      1.00      1.00    210569

Matriz de confusión:
[[   613      3]
 [    36 209917]]
```

Figura 40. Tercera evaluación con datos complementarios y aplicando la técnica SMOTE.

Gracias a los ajustes realizados en el modelo y al uso de datos balanceados generados mediante la técnica SMOTE, se logró un desempeño significativamente mejorado en esta nueva prueba de evaluación. Como se muestra en la Figura 40, todos los valores reportados fueron obtenidos directamente de la ejecución del código presentado. El modelo generó un informe de clasificación que indica una precisión del 94% para la Clase 0 (tráfico benigno) y del 100% para la Clase 1 (tráfico malicioso).

Se calculó un F1-Score ponderado de 99.8%, combinando precisión y sensibilidad para reflejar el balance general del modelo. La matriz de confusión generada por el mismo código muestra 613 verdaderos positivos para la Clase 0 y 209,917 verdaderos positivos para la Clase 1. Estos resultados, derivados exclusivamente de la ejecución presentada en la Figura 40, evidencian un desempeño casi perfecto y balanceado en ambas clases.

Tabla 28. Prueba de aceptación 03 en base a resultados obtenidos en Figura 39.

Aspecto evaluado	Valor obtenido	Criterio de aceptación	Resultado
Precisión (Clase 0 - Benigno)	94%	≥ 85%	Aceptado
Precisión (Clase 1 - Malicioso)	100%	≥ 85%	Aceptado
F1-Score (Clase 0 - Benigno)	97%	≥ 85%	Aceptado
F1-Score (Clase 1 - Malicioso)	100%	≥ 85%	Aceptado
Sensibilidad (Recall - Clase 0)	100%	≥ 85%	Aceptado
Sensibilidad (Recall - Clase 1)	100%	≥ 85%	Aceptado
Precisión general (Accuracy)	100%	≥ 85%	Aceptado
Matriz de confusión	[[613, 3], [36, 209971]]	Mínimos falsos positivos y negativos	Aceptado
Resultado final de la prueba		Aprobado	

La prueba presentada en la Tabla 28 se considera exitosa, ya que el modelo cumple con todos los criterios de aceptación previamente establecidos. Los resultados evidencian un desempeño sobresaliente en ambas clases, destacando una precisión del 94% para la Clase 0 (tráfico benigno), lo que soluciona las deficiencias identificadas en pruebas anteriores. Además, la matriz de confusión muestra un número mínimo de errores, con solo 3 falsos negativos en la Clase 0 y 36 falsos positivos en la Clase 1 (tráfico malicioso). Estos resultados reflejan un excelente equilibrio entre sensibilidad y precisión, confirmando que el modelo ahora está bien ajustado para identificar de manera confiable ambos tipos de tráfico.

```
# Validación cruzada (F1-Score ponderado)
cv_scores = cross_val_score(model, X_train, y_train, cv=5, scoring='f1_weighted')
print("F1-Score promedio en validación cruzada:", cv_scores.mean())
```

```
F1-Score promedio en validación cruzada: 0.9998398555678912
```

Figura 41. Prueba de validación cruzada.

Tras obtener una prueba de aceptación aprobada y confirmar que el modelo cumplía con todos los criterios establecidos, se procedió a realizar la validación cruzada, como

se detalla en la Figura 41. Esta técnica consiste en dividir el conjunto de datos de entrenamiento en múltiples particiones, entrenando y evaluando el modelo en diferentes combinaciones de estos subconjuntos. El código mostrado en la figura utiliza la función cross_val_score de scikit-learn con el modelo ajustado y el conjunto de entrenamiento (X_train y y_train), configurado para calcular el F1-Score ponderado mediante el parámetro scoring='f1_weighted'.

El resultado de esta ejecución, calculado como el promedio de los F1-Scores obtenidos en las cinco iteraciones, es un valor de 99.98%. Este resultado confirma la capacidad del modelo para generalizar correctamente, manteniendo un buen desempeño en todas las particiones del conjunto de entrenamiento. La validación cruzada asegura que el rendimiento del modelo no depende de una única división de los datos, validando así su estabilidad y consistencia. Con este respaldo, se concluye la etapa de ajustes y evaluaciones, estableciendo este modelo como el definitivo para las etapas siguientes.

3.2.6 Fase 6: Despliegue

a. Preparación del entorno y configuración inicial del servidor.

Luego de tener preparado el modelo de machine learning, se pasa a la fase de despliegue, en esta fase se determina como primer punto el software IDS que se utilizará, para poder seleccionar los mejores candidatos que se puedan adaptar al contexto del proyecto, se establecieron criterios de inclusión y exclusión descritos en la Tabla 29.

Tabla 29. Criterios de inclusión y exclusión para software IDS.

Criterios	Inclusión	Exclusión
Facilidad de configuración	Soluciones con documentación extensa y configuraciones intuitivas.	Herramientas con difícil acceso a las herramientas de configuración o configuraciones extremadamente complicadas.
Escalabilidad	IDS que pueda manejar tráfico de red de alta densidad sin degradar el rendimiento.	Sistemas que presenten limitaciones significativas al aumentar el volumen de tráfico.

Criterios	Inclusión	Exclusión
Actualizaciones y soporte	Software con actualizaciones regulares y una comunidad activa o soporte oficial.	Herramientas sin mantenimiento o con ciclos de actualización irregulares.
Análisis en tiempo real	Soluciones que permitan la detección de intrusiones en tiempo real y generen alertas de manera inmediata.	Sistemas que no incluyan capacidades de detección en tiempo real.
Integración con herramientas existentes	IDS que se integre fácilmente con herramientas de monitoreo y análisis ya implementadas.	Soluciones que requieran modificaciones extensivas para su integración.
Eficiencia en el uso de recursos	Software optimizado para entornos con recursos limitados, especialmente CPU y memoria.	Soluciones que requieran hardware de alta gama para su funcionamiento básico.
Flexibilidad de reglas	IDS que permita personalizar reglas para la detección de tráfico malicioso según las necesidades específicas del proyecto.	Herramientas con reglas estáticas o difíciles de modificar.
Licencia de código abierto	Soluciones que sean open source, permitiendo personalización y libre acceso.	Software propietario que limite la personalización o requiera licencias costosas.

Considerando los criterios de inclusión y exclusión establecidos, se realizó una preselección de tres herramientas IDS: Zeek, Suricata y Snort, ya que estas opciones se alinearon mejor con los parámetros definidos. Posteriormente, se procedió a realizar un análisis de estas herramientas utilizando una ficha comparativa, como se presenta en la Tabla 30, con el propósito de identificar la solución más adecuada para garantizar un rendimiento óptimo y una integración eficiente en el entorno de red del sistema.

Tabla 30. Ficha comparativa de herramientas IDS [30], [31], [32], [42].

Criterio	Zeek	Suricata	Snort
Compatibilidad de protocolos	Amplia (TCP, UDP, HTTP, DNS, SMB, etc.).	Amplia y detallada (TCP, UDP, HTTP, DNS, TLS, FTP, etc.).	Amplia, pero más limitada en protocolos avanzados.
Facilidad de configuración	Requiere experiencia avanzada; curva de aprendizaje elevada.	Configuración intuitiva con extensa documentación y ejemplos claros.	Moderada; configuración manual requiere ajustes adicionales.
Escalabilidad	Alta, adecuado para redes grandes con alta densidad de tráfico.	Muy alta, diseñado para soportar múltiples hilos y tráfico intenso.	Moderada, eficiente pero menos robusto en redes de gran volumen.

Criterio	Zeek	Suricata	Snort
Actualizaciones y soporte	Comunidad activa y actualizaciones frecuentes.	Actualizaciones regulares y una comunidad global muy activa.	Soporte activo, aunque con menor frecuencia.
Análisis en tiempo real	Análisis profundo, pero menos eficiente en alertas en tiempo real.	Detección rápida y precisa con alertas en tiempo real.	Muy eficiente en detección en tiempo real, aunque menos detallado que Suricata.
Integración	Compatible con muchas herramientas, pero requiere configuraciones adicionales.	Altamente compatible con sistemas de monitoreo ya existentes.	Compatible, pero con una curva de integración más compleja.
Uso de recursos	Eficiente, pero puede ser demandante en hardware limitado.	Altamente optimizado para entornos con recursos limitados.	Moderado, requiere recursos razonables.
Flexibilidad de reglas	Enfoque en análisis general, menos centrado en reglas personalizables.	Gran flexibilidad para personalizar reglas según las necesidades específicas.	Basado en reglas predefinidas, con opciones de personalización moderadas.
Licencia de código abierto	Sí	Sí	Sí
Diferenciador principal	Ideal para análisis forense y de eventos en profundidad.	Balance óptimo entre rendimiento, personalización y escalabilidad.	Eficaz para detección rápida con enfoque en seguridad tradicional.

Después de analizar las opciones disponibles, Suricata fue elegida como la solución ideal para el proyecto, ya que cumple de manera integral con los criterios de inclusión y exclusión establecidos. Suricata sobresale por su capacidad de equilibrar rendimiento, flexibilidad y escalabilidad, ofreciendo soporte avanzado para una amplia variedad de protocolos (como TCP, UDP, HTTP, DNS, TLS, FTP, entre otros) y destacando por su capacidad para manejar tráfico de alta densidad gracias a su diseño multi-hilo. Además, su configuración intuitiva, respaldada por una documentación extensa y una comunidad activa, facilita tanto su implementación como su personalización, algo especialmente importante en entornos con recursos limitados.

Otra de las fortalezas de Suricata es su flexibilidad para crear reglas personalizadas que se adaptan a las necesidades específicas del proyecto, así como su integración sencilla y eficiente con sistemas de monitoreo existentes. En comparación con otras herramientas como Zeek y Snort, Suricata ofrece un balance entre facilidad de uso, análisis en tiempo real y eficiencia en el uso de recursos.

Para seleccionar la distribución de sistema operativo adecuada para la instalación de Suricata, se realizó una revisión de la guía oficial proporcionada por la Open Information Security Foundation (OISF) [30]. En esta guía, se describen los pasos necesarios para la instalación de Suricata en diferentes sistemas operativos, destacándose Ubuntu Server como una de las opciones más sencillas y directas para implementar. Esta distribución no solo presenta instrucciones claras y detalladas en la documentación oficial, sino que también cuenta con una amplia comunidad de usuarios y desarrolladores, lo que facilita la resolución de problemas y el acceso a recursos adicionales durante la configuración.

Además, Ubuntu Server es reconocido por su estabilidad, eficiencia y compatibilidad con herramientas de monitoreo y detección de intrusos, lo que lo convierte en una opción confiable para proyectos que requieren un alto nivel de desempeño. Estos factores, combinados con su facilidad de instalación y configuración de Suricata, hicieron que Ubuntu Server fuera seleccionado como la distribución adecuada.

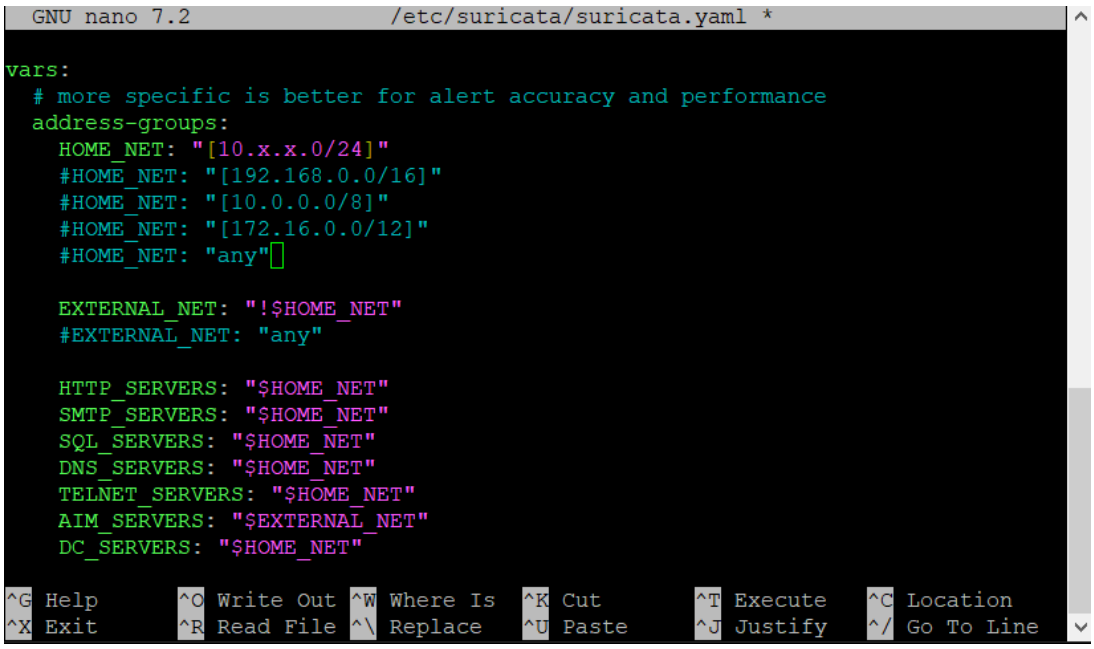


Figura 42. Configuración de rango de análisis de Suricata.

Tras completar la instalación de Suricata, siguiendo las instrucciones proporcionadas en la guía oficial de la Open Information Security Foundation (OISF) [30], se llevó a cabo la configuración necesaria para que el sistema pudiera operar correctamente en la red de la Unidad Educativa “19 de Septiembre”. Como se ilustra en la Figura 42, se

accedió al archivo de configuración sugerido en la guía de instalación, ubicado en la ruta indicada por el instructivo. Dentro de este archivo, específicamente en el apartado address-groups de la sección HOME_NET, se procedió a definir el rango de red en el cual Suricata monitorizará y analizará el tráfico. En este caso, se configuró un rango de red con la notación 10.x.x.0/24, omitiendo la dirección IP exacta por motivos de seguridad.

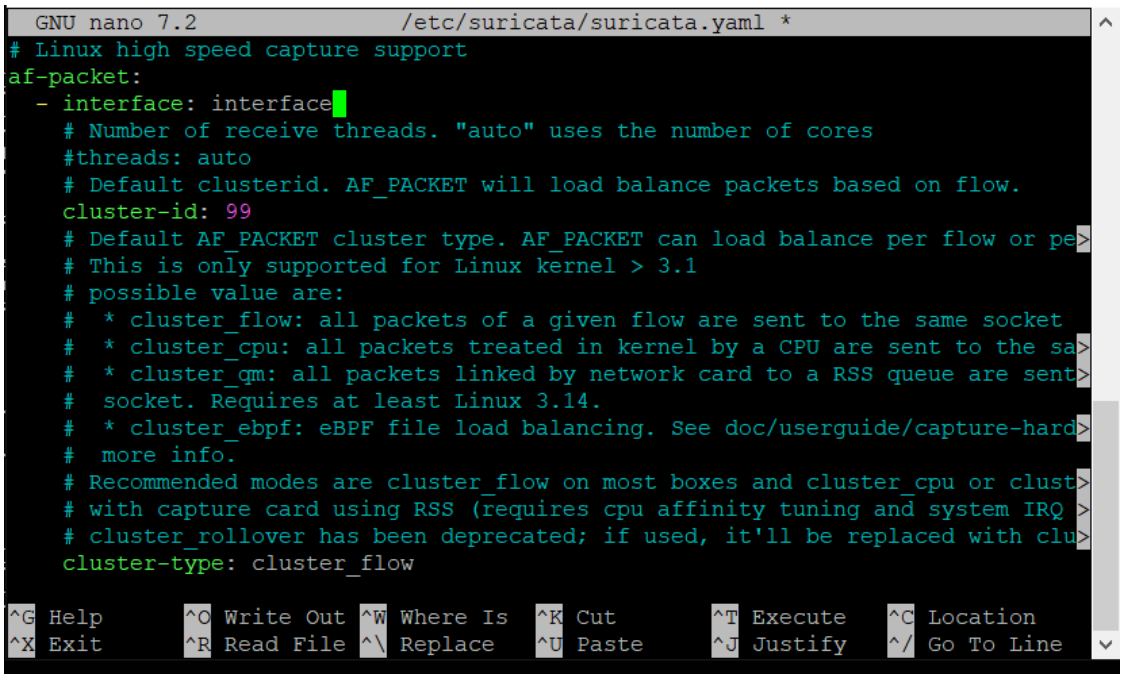


Figura 43. Configuración de interface de red en Suricata.

Otra configuración inicial realizada en Suricata para garantizar su correcto funcionamiento fue la definición de la interfaz de red que se utilizará para monitorear la actividad en la red. Como se observa en la Figura 43, esta configuración se lleva a cabo en la sección af-packet, dentro del apartado interface en el archivo de configuración. Por motivos de seguridad, la interfaz específica configurada no se muestra en la imagen, pero se utilizó la interfaz asignada al servidor de la Unidad Educativa “19 de Septiembre”.

b. Integración con modelo de machine learning

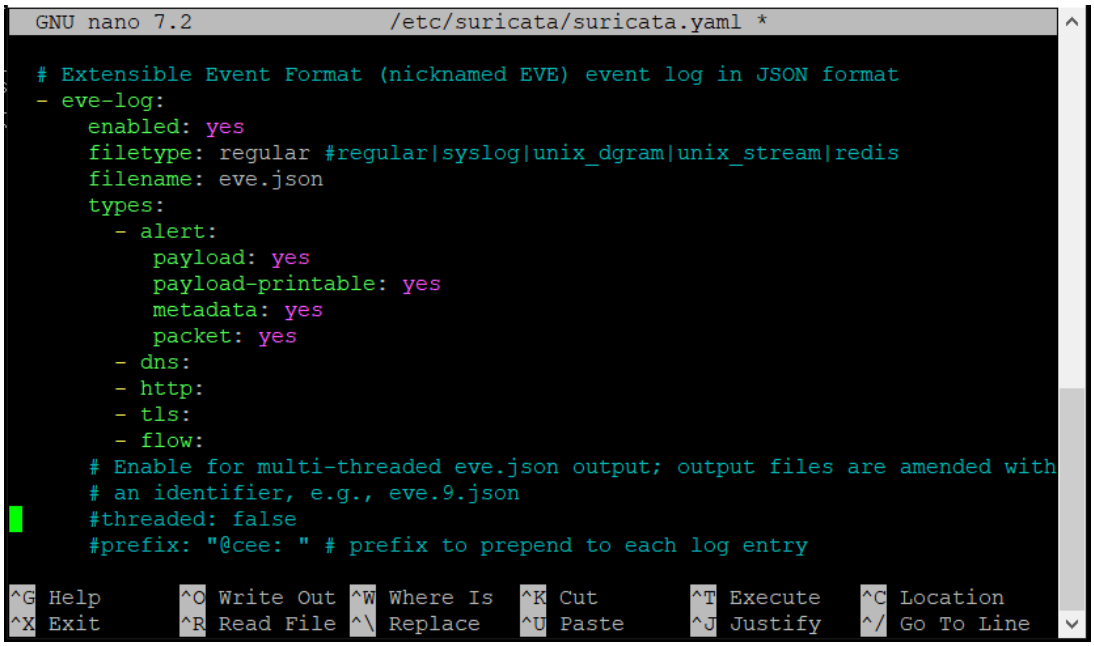


Figura 44. Configuración formato de salida de Suricata.

Para integrar las lecturas realizadas por Suricata con el modelo de machine learning, se aprovechó una de las funcionalidades más destacadas del software: la capacidad de exportar los registros en formato JSON. Este formato es ampliamente utilizado en aplicaciones de análisis debido a su estructura legible y su fácil manejo mediante scripts, lo que permite una interacción fluida entre Suricata y el modelo. De esta manera, el modelo puede acceder en tiempo real a los datos generados por Suricata, analizar el tráfico de red y determinar si se trata de tráfico benigno o de un posible intento de ataque.

Como se muestra en la Figura 44, para habilitar esta funcionalidad, se realizó una configuración específica en el archivo de configuración de Suricata. En este archivo, se activó la opción de salida en formato JSON dentro del apartado eve-log, y se definieron los tipos de eventos que Suricata debe registrar, como alert, dns, http, tls y flow. Estos tipos de datos fueron seleccionados para proporcionar información relevante y completa sobre el tráfico de red, maximizando la capacidad del modelo de machine learning para detectar patrones maliciosos.

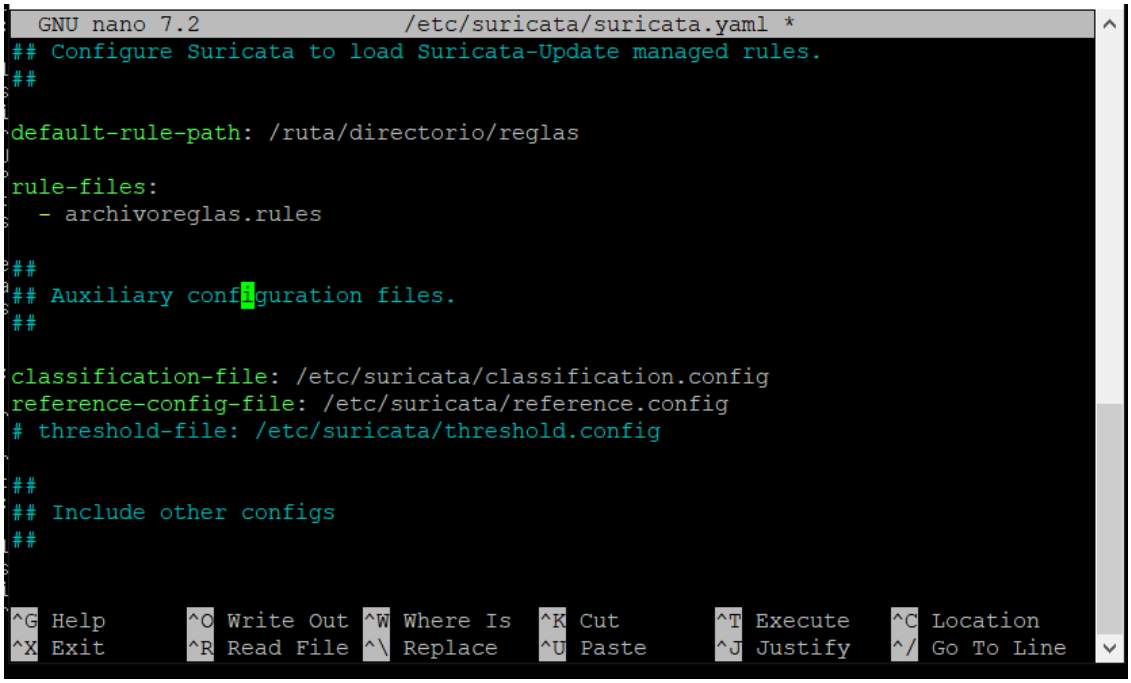


Figura 45. Configuración de archivo de reglas de Suricata.

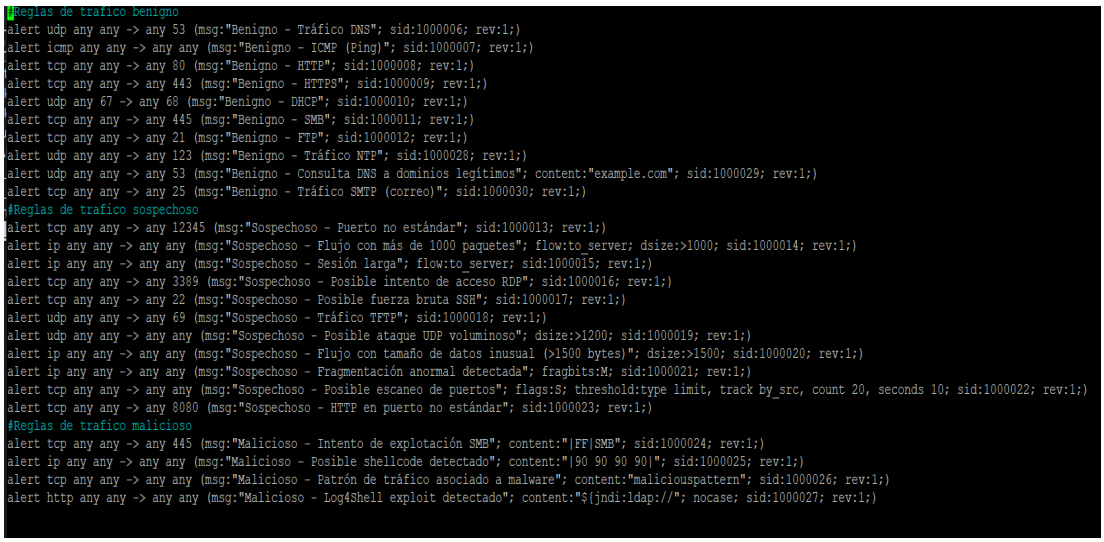


Figura 46. Reglas complementarias para Suricata.

Para complementar la detección automatizada del sistema, se configuraron reglas específicas en Suricata que refuerzan su capacidad de monitoreo y análisis del tráfico de red. Inicialmente, se creó un directorio y un archivo dedicado exclusivamente a las reglas, como se muestra en la Figura 45. En esta configuración, se utilizó la sección default-rule-path para establecer la ruta donde se almacenarán las reglas y el apartado rule-files para especificar el archivo de reglas que será cargado por Suricata. Es

importante señalar que, por razones de seguridad, tanto la ruta como el nombre del archivo mostrados en la Figura 45 son ficticios, pero reflejan la estructura y el proceso general empleado en esta configuración.

En la Figura 46, se detallan las reglas complementarias creadas para el sistema, las cuales se organizaron en tres secciones principales: tráfico benigno, tráfico sospechoso y tráfico malicioso. Estas divisiones tienen como finalidad categorizar el tráfico de red y proporcionar una base sólida para la detección y clasificación de eventos.

Tráfico benigno: Incluye reglas diseñadas para identificar tráfico legítimo y común dentro de la red, como solicitudes DNS, pings ICMP, conexiones HTTP y otros protocolos habituales.

Tráfico sospechoso: Este grupo abarca reglas que detectan comportamientos anómalos o patrones inusuales que podrían ser indicativos de un posible intento de ataque. Por ejemplo, flujos de datos con tamaños inusuales, intentos de acceso a puertos no estándar o conexiones con características atípicas.

Tráfico malicioso: Contiene reglas específicas para identificar actividades confirmadas como maliciosas, como intentos de explotación de vulnerabilidades conocidas, patrones asociados a malware o ataques específicos.

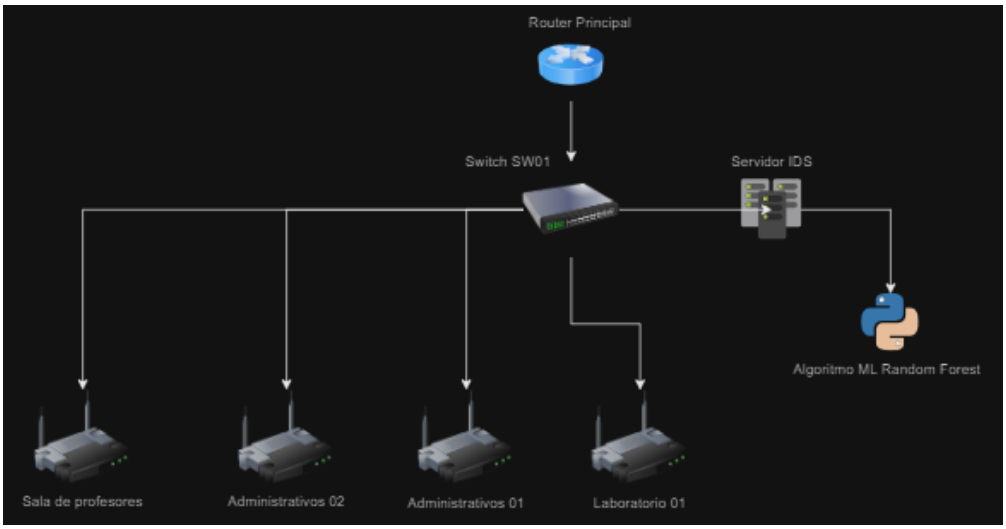


Figura 47. Arquitectura de la red con Suricata y modelo de machine learning en la Unidad Educativa “19 de Septiembre”.

Una vez que el modelo de machine learning fue entrenado y el servidor IDS configurado para trabajar en conjunto con este, se procedió a integrar todo el sistema para que pudiera operar de manera eficiente. Para llevar a cabo esta integración, fue fundamental comprender la arquitectura de la red tras la implementación del servidor IDS y el modelo de machine learning. Como se ilustra en la Figura 47, la arquitectura de la red consta de un router principal conectado a un switch principal, del cual se derivan cuatro subredes y una conexión directa al servidor IDS.

El modelo de machine learning fue alojado en una máquina específica, dedicada exclusivamente a la ejecución y análisis de los datos registrados por Suricata en tiempo real. Esta decisión permitió optimizar el uso de recursos del servidor IDS, enfocándolo únicamente en las funciones de monitoreo y detección. Al delegar la ejecución del modelo a una máquina secundaria, se maximizó la capacidad de procesamiento, asegurando que ambas herramientas, Suricata y el modelo de machine learning, operaran de manera eficiente y sin interrupciones.

```
[SuricataLogs]
path = /ruta_carpeta_compartida/suricata_logs
browseable = yes
writable = yes
guest ok = yes
create mask = 0777
directory mask = 0777
```

Figura 48. Configuración de carpeta compartida en Samba.

```
#!/bin/bash

# Rutas
EVE_FILE="/var/log/suricata/eve.json"
COPY_FILE="/carpeta/compartida/eve_copy.json"

# Monitorear y sincronizar
inotifywait -m -e modify "$EVE_FILE" |
while read path action file; do
    cp "$EVE_FILE" "$COPY_FILE"
    echo "$(date '+%Y-%m-%d %H:%M:%S') - Archivo eve.json actualizado y copiado." >> /var/log/eve_sync.log
done
```

Figura 49. Configuración de actualización de archivo eve.json.

Para garantizar una integración eficiente entre Suricata y el modelo de machine learning, se utilizó Samba en Ubuntu Server, un protocolo ampliamente reconocido por facilitar la compartición de archivos entre diferentes sistemas operativos. En este caso, Samba fue configurado para compartir el archivo de registro generado por Suricata en formato JSON, el cual, como se mostró en la Figura 44, corresponde al archivo eve.json. Este archivo contiene los logs de actividad en tiempo real, esenciales para que el modelo de machine learning pueda analizar y clasificar el tráfico de red.

Además, para preservar la integridad y seguridad del archivo eve.json, se implementó un script automatizado, como se muestra en la Figura 49. Este script monitorea constantemente el archivo original y, cada vez que se actualiza, genera una copia en la carpeta compartida configurada con Samba. Esto garantiza que el modelo de machine learning siempre acceda a una versión actualizada de los registros sin comprometer el archivo original. La sincronización automática también minimiza los riesgos de pérdida de datos y asegura un flujo continuo de información entre Suricata y el modelo.

La configuración de Samba, como se detalla en la Figura 48, incluyó la definición de parámetros clave como la ruta del directorio compartido, permisos de escritura y lectura, así como ajustes para garantizar su accesibilidad. Es importante destacar que, por motivos de seguridad, la ruta mostrada en la figura es ficticia, pero refleja la estructura real utilizada en el proyecto. Esta solución combinada de Samba y el script de sincronización asegura una comunicación fluida y confiable entre las herramientas, optimizando el desempeño del sistema en su conjunto.

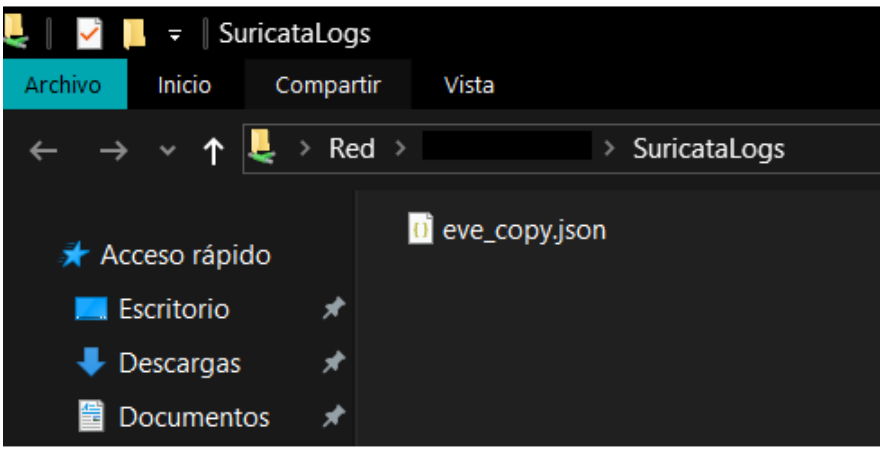


Figura 50. Carpeta compartida con archivo eve.json.

Para verificar que la configuración de Samba se realizó correctamente y que el script de automatización para la copia del archivo eve.json funciona como se esperaba, se comprobó que la carpeta compartida fuera accesible desde un sistema operativo Windows. Además, se verificó que dentro de la carpeta se pudiera visualizar la copia actualizada del archivo eve.json. Como se ilustra en la Figura 50, esta prueba confirma que la integración entre el servidor y el entorno compartido opera de manera fluida, garantizando el acceso a los datos necesarios para el análisis por parte del modelo de machine learning.

```
# **3. Cargar el modelo entrenado**
modelo_ruta = r'c:\ruta\modelo\machinelearning\modelo_final_optimizado.pkl'
try:
    model = joblib.load(modelo_ruta)
    print("✅ Modelo cargado exitosamente.")
except FileNotFoundError:
    print(f"❌ Error: El modelo {modelo_ruta} no se encontró.")
    logging.error(f"Error: El modelo {modelo_ruta} no se encontró.")
    exit()

# **4. Ruta al archivo Suricata (eve.json)**
filepath = '\\\\ruta\\carpeta_compartida\\eve_copy.json'

# **5. Verificar si el archivo eve.json se actualiza**
def verificar_actividad_archivo(filepath, intervalo=60):
    try:
        ultima_modificacion = os.path.getmtime(filepath)
        ahora = time.time()
        if (ahora - ultima_modificacion) > intervalo:
            mensaje = f"⚠️ El archivo {filepath} no ha sido modificado en los últimos {intervalo} segundos."
            print(mensaje)
            logging.warning(mensaje)
            return False
        return True
    except FileNotFoundError:
        mensaje = f"❌ Error: No se puede acceder al archivo {filepath}."
        print(mensaje)
        logging.error(mensaje)
        return False

```

Figura 51. Configuración de conexión de modelo con archivo eve.json.

Para establecer la conexión entre el modelo de machine learning y los datos registrados en el archivo eve.json generado por Suricata, se desarrolló un script que desempeña varias funciones clave, como se muestra en la Figura 51. En primer lugar, el script se

encarga de cargar el modelo de machine learning previamente entrenado desde una ubicación definida. Luego, accede a la ruta donde se encuentra el archivo eve.json dentro de la carpeta compartida para verificar su disponibilidad y asegurarse de que esté accesible para su análisis.

En caso de que ocurra algún problema, como la falla al cargar el modelo o al acceder al archivo compartido, el script está programado para indicar sobre el error y facilitar una intervención oportuna. Cabe destacar que, por motivos de seguridad, las rutas mostradas en la Figura 51 son ficticias y no reflejan la configuración real del sistema, con el fin de proteger la integridad del entorno implementado.

```python
def process_line(line):
    try:
        json_data = json.loads(line)
        if json_data.get('event_type') == 'flow':
            input_data = preprocess_data(json_data)
            if input_data is not None:
                prediction = model.predict(input_data)
                if prediction[0] == 1:
                    print("⚠️ Alerta: Posible ataque detectado.")
                    logging.warning("Alerta: Posible ataque detectado.")
                else:
                    print("✅ Tráfico benigno detectado.")
    except json.JSONDecodeError:
        print("Error al decodificar JSON.")
        logging.error("Error al decodificar JSON.")
    except Exception as e:
        print(f"Error al procesar la línea: {e}")
        logging.error(f"Error al procesar la línea: {e}")
```

Figura 52. Función para procesar archivo eve.json en script.

En el script, se desarrolló una función para analizar y procesar las líneas del archivo eve.json, como se muestra en la Figura 52. Este código permite que el sistema lea cada línea del archivo JSON y evalúe si corresponde a un evento de tipo flow, indicando un flujo de red que debe ser analizado. Si la línea cumple con este criterio, los datos son preprocesados y enviados al modelo de machine learning para su predicción. En función del resultado, el sistema clasifica el tráfico como benigno o como un posible ataque.

Cuando el modelo detecta un posible ataque (Clase 1), se genera una alerta para que la persona encargada tome las medidas necesarias, pero si se detecta tráfico benigno (Clase 0), el sistema confirma que la actividad no representa una amenaza. Además, el script incluye manejo de errores, como la detección de líneas con formato JSON incorrecto, las cuales son ignoradas, y errores generales durante el procesamiento de cada línea, asegurando que el sistema opere de manera continua.

```
class FileChangeHandler(FileSystemEventHandler):
    def __init__(self, filepath):
        self.filepath = filepath
        self.last_position = 0

    def on_modified(self, event):
        if event.src_path == self.filepath:
            self.process_new_lines()

    def process_new_lines(self):
        try:
            with open(self.filepath, 'r') as file:
                if self.last_position > os.path.getsize(self.filepath):
                    self.last_position = 0
                file.seek(self.last_position)
                new_lines = file.readlines()
                self.last_position = file.tell()
                for line in new_lines:
                    process_line(line)
        except Exception as e:
            logging.error(f"Error al procesar nuevas líneas: {e}")
```

Figura 53. Clase para el monitoreo y detección de cambios en el archivo eve.json.

Otra parte importante del script es la implementación de una clase personalizada, como se muestra en la Figura 53, que permite detectar y procesar cambios en el archivo eve.json generado por Suricata. Para ello, se utiliza la clase FileChangeHandler, que hereda de FileSystemEventHandler. Este componente monitorea continuamente el archivo y responde de manera automática cada vez que se modifica. La función principal, on_modified, verifica si el evento de modificación corresponde al archivo monitoreado (self.filepath) y, si es así, llama al método process_new_lines para procesar las nuevas líneas agregadas al archivo desde la última actualización.

El método process_new_lines asegura que solo se lean las líneas añadidas recientemente, comenzando desde la última posición registrada en el archivo

(self.last_position). Esto no solo optimiza el rendimiento al evitar relecturas innecesarias, sino que también garantiza que ningún dato sea omitido.

Esta técnica permite que el sistema procese los datos en tiempo real, asegurando que cada nuevo evento capturado por Suricata sea analizado por el modelo de machine learning de forma inmediata. Además, la implementación basada en eventos optimiza el uso de recursos al procesar solo los cambios necesarios, en lugar de monitorear continuamente todo el archivo desde el inicio. Esto refuerza la eficiencia y la capacidad de respuesta del sistema frente a posibles amenazas.

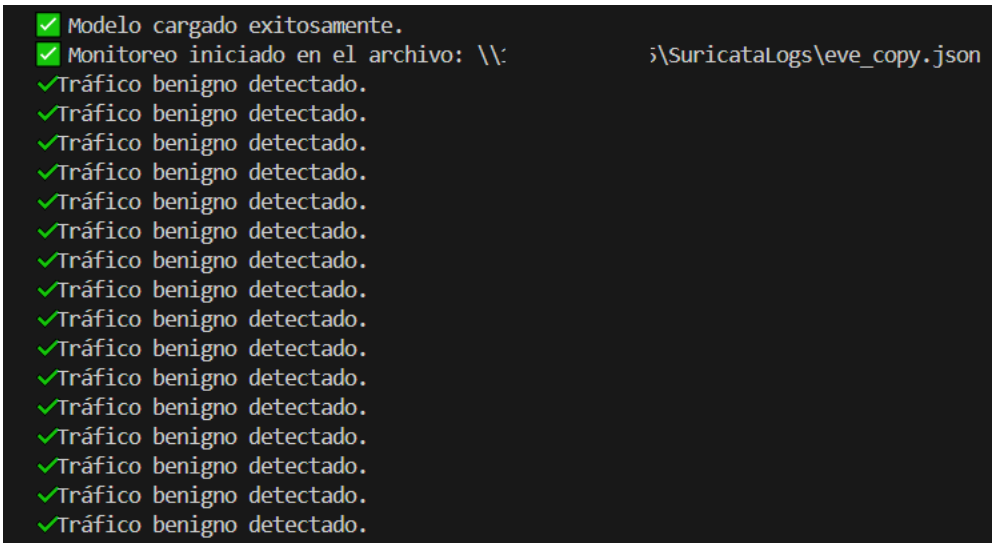


Figura 54. Ejecución de prueba de script de integración de modelo con Suricata.

Una vez configurado el servidor IDS e integrado con el modelo de machine learning, se llevó a cabo una ejecución de prueba con el propósito de verificar que todos los componentes del sistema estuvieran correctamente conectados y funcionando según lo esperado. Como resultado se obtuvo una ejecución exitosa, misma que validan la interacción entre el servidor IDS y el modelo, como se representa en la Figura 54.

c. Pruebas de integración y automatización

Una vez que todos los componentes del sistema IDS estuvieron completamente integrados y operativos, se procedió a realizar la prueba de aceptación descrita en la Tabla 31. Esta prueba fue llevada a cabo por el encargado del laboratorio de la Unidad Educativa “19 de Septiembre” con el objetivo de evaluar el desempeño general del

sistema. En particular, se buscó identificar posibles falencias relacionadas con la usabilidad y la eficiencia del sistema, permitiendo recopilar observaciones y recomendaciones clave para realizar ajustes y optimizar su funcionamiento.

Tabla 31. Ficha de aceptación del sistema IDS prueba 01 al encargado del laboratorio de computación en la Unidad Educativa “19 de Septiembre”.

Ficha de aceptación		
Persona aplicada	Responsable del laboratorio de computación	
Categoría	**Descripción**	**Calificación**
Funcionamiento General	Evaluar si el sistema IDS se ejecuta correctamente, asegurando la interacción adecuada entre Suricata, el modelo de machine learning y los scripts de integración.	Aprobado
Clasificación de Tráfico	Verificar que el sistema clasifique correctamente el tráfico benigno y malicioso, asegurando precisión en ambos casos.	Aprobado
Usabilidad del Sistema	Evaluar la facilidad de uso para el personal encargado, considerando la interpretación de alertas y la navegación en las funciones principales del sistema.	Reprobado
Integración General	Revisar si todos los componentes del sistema (Suricata, modelo de machine learning y scripts) trabajan de manera conjunta y sin errores.	Aprobado
Automatización General	Evaluar el nivel de automatización del sistema, incluyendo la sincronización de archivos, monitoreo en tiempo real y ejecución autónoma de procesos clave.	Reprobado
Observaciones	- Para mejorar la usabilidad del sistema se recomienda la integración con alguna herramienta que permita visualizar de mejor manera la monitorización en tiempo real. - Se sugiere que el sistema sea capaz de generar un archivo log que almacene información detallada sobre las alertas de intento de ataque para poder analizar mejor este tipo de eventos. - Es necesario automatizar la ejecución del script que se encarga de copiar el archivo eve.json en la carpeta compartida.	
Resultado de la prueba	Reprobado	

Tras una primera prueba de aceptación reprobada, se tomaron en cuenta las observaciones proporcionadas para realizar mejoras en el sistema. Se trabajó específicamente en fortalecer los puntos débiles identificados durante la evaluación, que incluían la mejora en la usabilidad para facilitar la interacción del personal con el sistema, la automatización de procesos clave para optimizar su operación, y la generación de archivos que almacenen información más detallada sobre los intentos

de ataque. Estas ayudarán a que el sistema tenga un desempeño más robusto y eficiente.

Para abordar los inconvenientes identificados, el primer punto a tratar fue la usabilidad del sistema. Una de las principales sugerencias fue integrar el sistema IDS con un software de monitoreo que facilite la visualización del funcionamiento y la detección de ataques en tiempo real. Para seleccionar el software de monitoreo más adecuado al contexto del proyecto, se definieron criterios de inclusión y exclusión, los cuales se detallan en la Tabla 32. Estos criterios permitieron analizar diversas opciones y, posteriormente, determinar cuál se adapta mejor a las necesidades específicas del proyecto, garantizando una experiencia de usuario más intuitiva y eficiente.

Tabla 32. Criterios de inclusión y exclusión para softwares de monitoreo.

Criterios	Incluir	Excluir
Soporte para gráficos y paneles	Soluciones con capacidad para crear dashboards personalizables y gráficos interactivos.	Software con visualizaciones limitadas o sin posibilidad de personalización.
Facilidad de configuración	Herramientas con documentación amplia, tutoriales y una interfaz amigable para configuración rápida.	Programas con configuraciones complejas y sin guías claras para nuevos usuarios.
Integración con sistemas existentes	Soluciones que se integren fácilmente con Ubuntu Server y el ecosistema actual de la red.	Herramientas que requieran ajustes extensivos para la integración con el sistema.
Licencia de código abierto	Software open source con comunidad activa y soporte regular.	Soluciones propietarias o con licencias costosas que limiten su accesibilidad.
Eficiencia en el uso de recursos	Programas optimizados para funcionar en entornos con recursos limitados, como CPU y memoria.	Software que consuma recursos excesivos o requiera hardware especializado.
Monitoreo en tiempo real	Soluciones que permitan monitoreo en tiempo real de eventos.	Herramientas con retrasos significativos en la generación de eventos.

Tras analizar los criterios de inclusión y exclusión establecidos, se seleccionaron Kibana, Grafana y Zabbix como las herramientas de monitoreo que mejor se alinearon con las necesidades del proyecto.

Kibana: Herramienta de visualización de datos de código abierto diseñada para trabajar con Elasticsearch. Permite explorar, analizar y visualizar grandes volúmenes de datos a través de gráficos, tablas y paneles interactivos [42].

Grafana: Software de análisis y monitoreo de código abierto que soporta múltiples fuentes de datos, como Prometheus, InfluxDB y Elasticsearch. Ofrece dashboards dinámicos y personalizables que permiten visualizar métricas clave y establecer alertas basadas en umbrales definidos [43].

Zabbix: Plataforma integral de monitoreo de código abierto que proporciona capacidades para rastrear el estado y el rendimiento de redes, servidores, aplicaciones y servicios. Incluye funcionalidades avanzadas de recopilación de datos, notificaciones y generación de alertas [44].

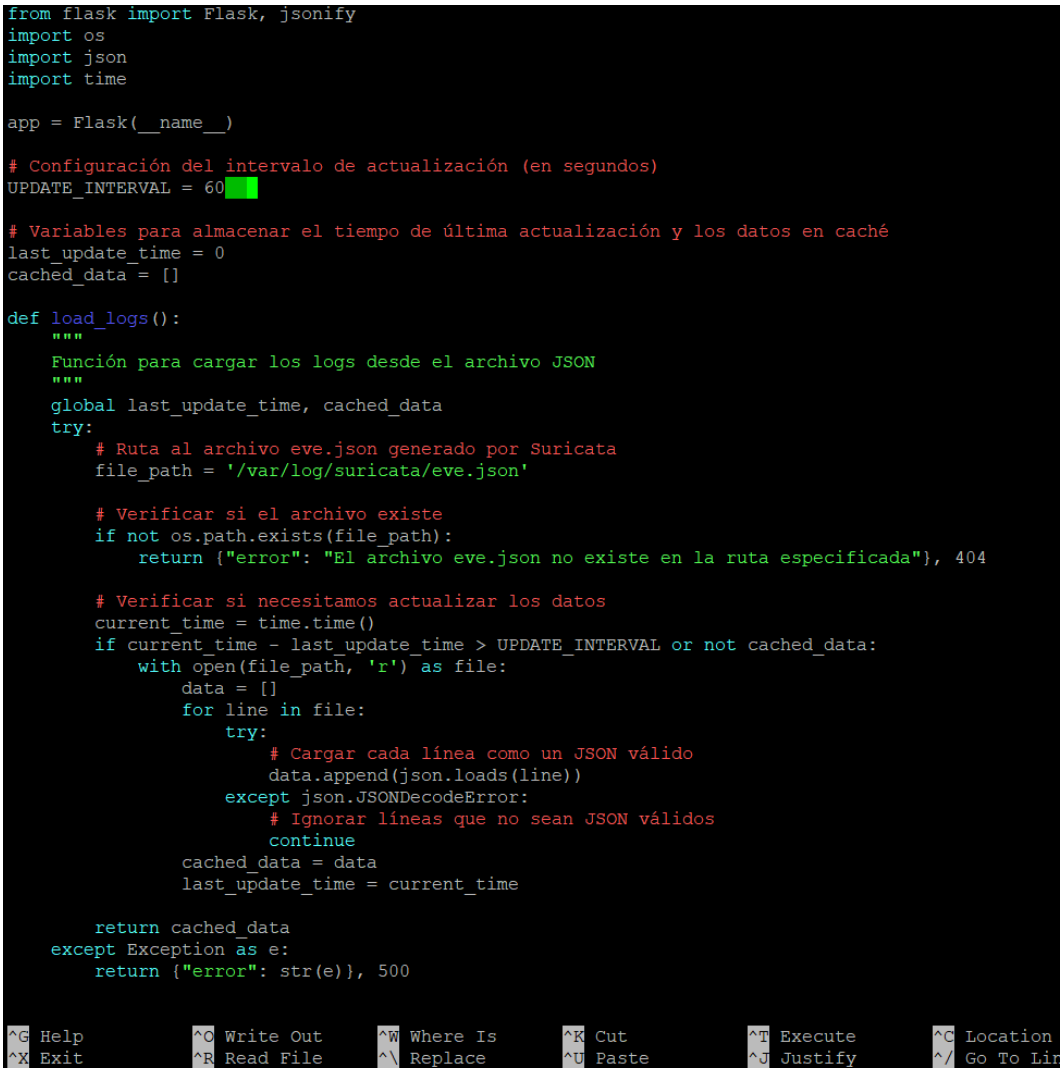
Para determinar cuál de estas herramientas es la más adecuada, se utilizó la ficha comparativa descrita en la Tabla 33, que permitió evaluar sus características en detalle y seleccionar la solución que mejor se adapta al contexto del sistema IDS implementado.

Tabla 33. Ficha comparativa de softwares de monitoreo [42], [43], [44].

Criterio	Kibana	Grafana	Zabbix
Compatibilidad con JSON	Soporte nativo para datos en formato JSON.	Soporte nativo y flexibilidad para trabajar con datos JSON.	Requiere configuraciones adicionales para integrar datos JSON.
Visualización de datos	Dashboards avanzados con opciones interactivas.	Excelentes dashboards altamente personalizables con gráficos interactivos de alta calidad.	Visualizaciones funcionales, pero menos flexibles y estéticas.
Facilidad de configuración	Interfaz amigable, pero curva de aprendizaje media.	Configuración intuitiva y documentación extensa.	Requiere experiencia previa y configuración técnica más compleja.
Integración con Suricata	Compatible, pero requiere ajustes específicos.	Integración directa y flexible, optimizada para múltiples fuentes de datos.	Integración posible, pero requiere configuraciones avanzadas.
Eficiencia en recursos	Optimizado para uso moderado de recursos.	Excelente optimización, ideal para entornos con recursos limitados.	Consume más recursos en comparación con Kibana y Grafana.
Licencia	Código abierto.	Código abierto con comunidad activa y soporte regular.	Código abierto, pero con opciones propietarias para funciones avanzadas.

Tras realizar el análisis utilizando la ficha comparativa expuesta en la Tabla 33, Grafana fue seleccionada como el software de monitoreo más adecuado para el sistema IDS. Su elección se fundamenta en su equilibrio entre flexibilidad, rendimiento y facilidad de integración. Grafana permite crear dashboards altamente personalizables, lo que facilita la visualización clara y precisa de los datos provenientes de Suricata,

mejorando la interpretación de alertas y el análisis del tráfico en tiempo real. Además, su compatibilidad nativa con múltiples formatos de datos, incluido JSON, asegura una integración fluida sin necesidad de configuraciones complejas.



```
from flask import Flask, jsonify
import os
import json
import time

app = Flask(__name__)

# Configuración del intervalo de actualización (en segundos)
UPDATE_INTERVAL = 60

# Variables para almacenar el tiempo de última actualización y los datos en caché
last_update_time = 0
cached_data = []

def load_logs():
    """
    Función para cargar los logs desde el archivo JSON
    """
    global last_update_time, cached_data
    try:
        # Ruta al archivo eve.json generado por Suricata
        file_path = '/var/log/suricata/eve.json'

        # Verificar si el archivo existe
        if not os.path.exists(file_path):
            return {"error": "El archivo eve.json no existe en la ruta especificada"}, 404

        # Verificar si necesitamos actualizar los datos
        current_time = time.time()
        if current_time - last_update_time > UPDATE_INTERVAL or not cached_data:
            with open(file_path, 'r') as file:
                data = []
                for line in file:
                    try:
                        # Cargar cada línea como un JSON válido
                        data.append(json.loads(line))
                    except json.JSONDecodeError:
                        # Ignorar líneas que no sean JSON válidos
                        continue
                cached_data = data
                last_update_time = current_time

        return cached_data
    except Exception as e:
        return {"error": str(e)}, 500

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Lin
```

Figura 55. Creación de API a partir del archivo eve.json.

Para integrar Grafana al sistema IDS, fue necesario crear una API que sirviera como intermediaria para exponer los datos generados en el archivo eve.json por Suricata, permitiendo que Grafana los consuma en tiempo real. La Figura 55 ilustra el script desarrollado para cumplir con este propósito, también el Anexo E presenta de forma extendida el código utilizado. Este script utiliza Flask, un framework de Python que facilita la creación de aplicaciones web ligeras, para implementar la API que proporciona acceso a los datos del archivo eve.json.

Además de exponer los datos, se incluyó una funcionalidad clave que asegura que la API se mantenga actualizada periódicamente. Esto se logra mediante la configuración de un intervalo de actualización definido en segundos, como se observa en la línea que establece UPDATE_INTERVAL = 60. El script verifica constantemente si ha pasado el tiempo establecido desde la última actualización o si los datos en la memoria caché están vacíos. En caso de cumplirse estas condiciones, el archivo eve.json es leído nuevamente, y los datos válidos son cargados y almacenados, garantizando así que Grafana siempre acceda a información actualizada.

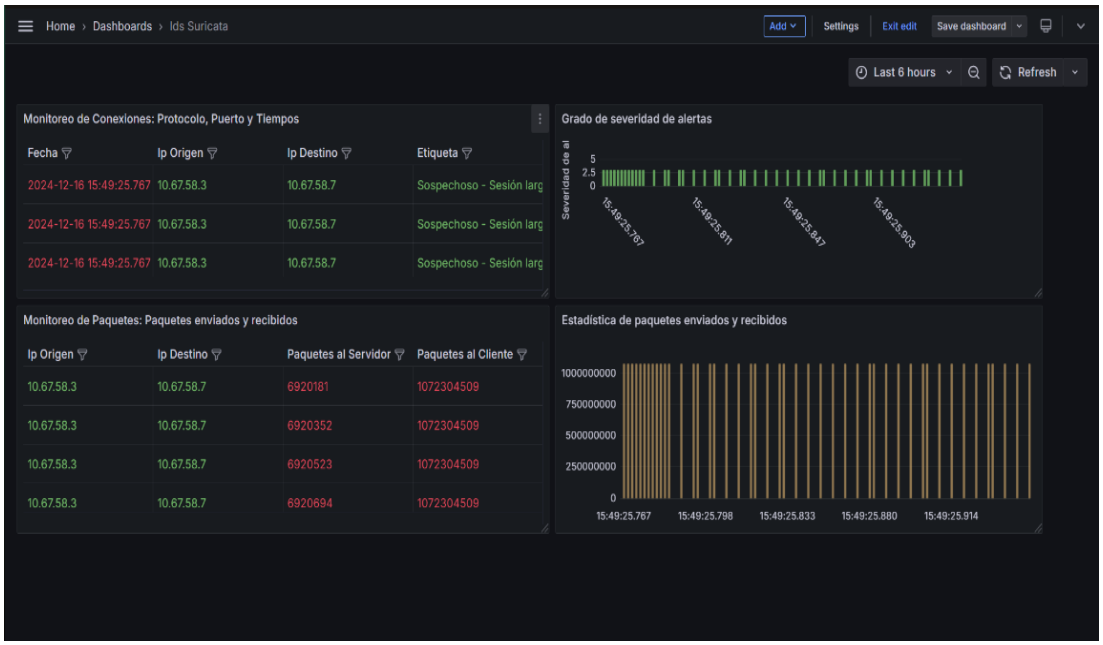


Figura 56. Dashboard generado en Grafana a partir de API de Suricata.

La Figura 56 muestra un dashboard interactivo desarrollado en Grafana, utilizando los datos proporcionados por la API generada a partir del archivo eve.json de Suricata. Este dashboard se estableció durante una simulación de ataque, permitiendo monitorear en tiempo real el comportamiento del tráfico de red y la generación de alertas. La visualización confirma que la integración entre Suricata, la API y Grafana fue exitosa, ya que los datos fluyen correctamente desde el sistema hasta el dashboard, habilitando una supervisión efectiva y dinámica.

El panel proporciona información clave en varias secciones. En la parte superior izquierda, se encuentra el monitoreo de conexiones, que detalla aspectos como protocolo, puerto, tiempos y etiquetas que clasifican el tráfico detectado. Estas

etiquetas, como "sospechoso", destacan patrones que podrían indicar comportamientos anómalos. En el gráfico de severidad, ubicado en la parte superior derecha, se representa la distribución de alertas generadas, clasificadas según su nivel de importancia. Por último, en la parte inferior, se incluyen estadísticas detalladas sobre los paquetes enviados y recibidos, lo que permite analizar con precisión la cantidad de datos procesados entre clientes y servidores.

Para abordar el segundo problema identificado tras la evaluación de la ficha de aceptación, relacionado con la necesidad de automatizar procesos clave del sistema, se implementaron soluciones que optimizan significativamente la operación del sistema IDS. Una de las primeras acciones fue automatizar el proceso de copiado del archivo eve.json, lo que asegura que los datos generados por Suricata se sincronicen de manera constante con el directorio compartido. Esto garantiza que la información siempre esté actualizada y lista para su análisis sin necesidad de intervención manual.

Además, se automatizó la ejecución del script encargado de generar la API, lo que permite mantener una comunicación fluida y en tiempo real entre Suricata y Grafana. Estas medidas no solo simplifican el funcionamiento del sistema, sino que también mejoran su eficiencia y confiabilidad, asegurando que todos los procesos críticos operen de manera continua y autónoma.

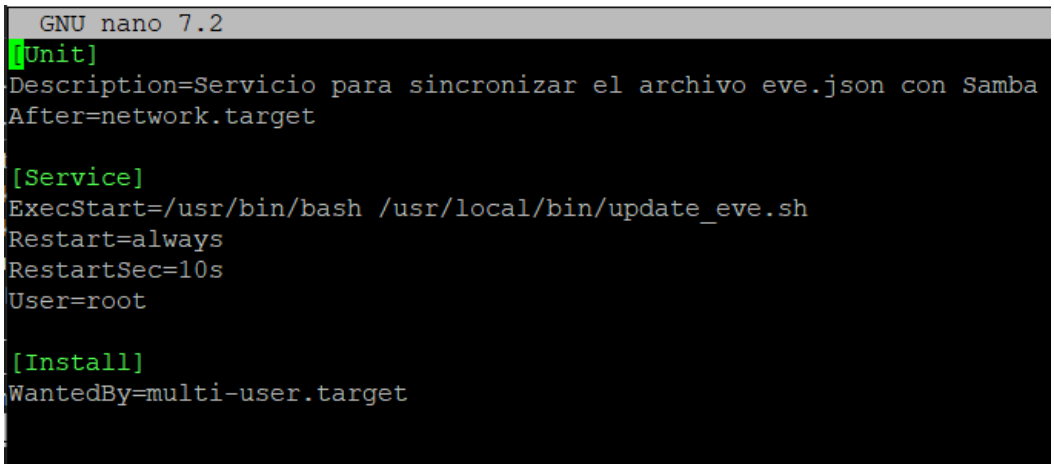


Figura 57. Configuración de servicio para automatizar script para archivo compartido eve.json.

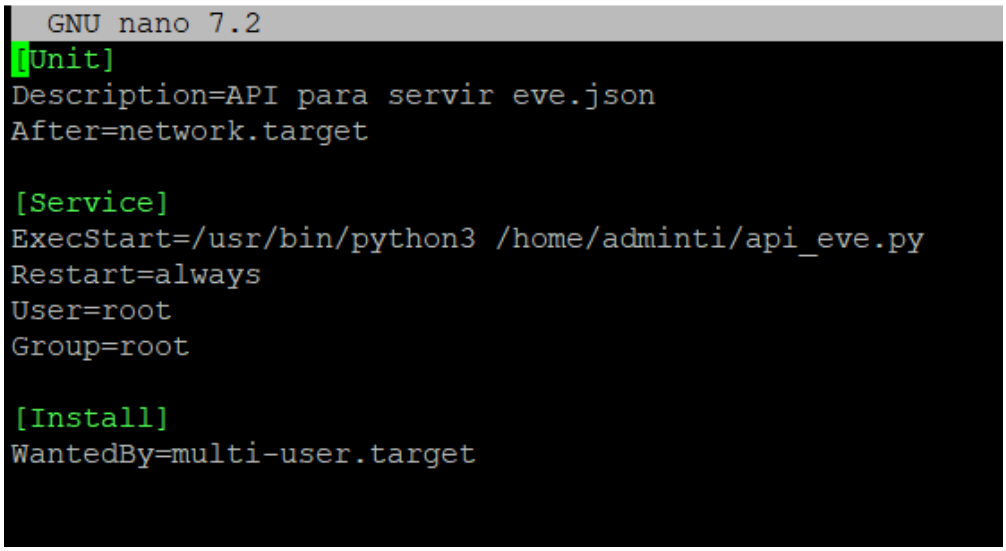


Figura 58. Configuración de servicio para automatizar generación de API en Suricata.

Para automatizar los procesos clave del sistema, como la copia del archivo eve.json en la carpeta compartida y la generación de la API necesaria para la comunicación con Grafana, se configuraron servicios en el servidor. La configuración del servicio encargado de sincronizar el archivo eve.json con la carpeta compartida se muestra en la Figura 57, mientras que la automatización de la generación de la API se detalla en la Figura 58. Estos servicios fueron diseñados para activarse automáticamente tan pronto como el servidor se inicie, garantizando que ambos procesos estén operativos sin intervención manual. Esta implementación no solo mejora la eficiencia del sistema, sino que también asegura que los componentes esenciales del sistema IDS estén siempre en funcionamiento, contribuyendo a la estabilidad y confiabilidad del entorno.

Para abordar la tercera inquietud identificada en la ficha de aceptación, relacionada con la necesidad de registrar información detallada sobre los intentos de ataque, se realizaron modificaciones en el algoritmo encargado de ejecutar el modelo de machine learning. Para poder agregar la funcionalidad de generar un archivo de log que almacena detalles específicos de los eventos detectados como intentos de ataque. El propósito de almacenar ese tipo de información es para proporcionar al personal encargado una herramienta que facilite el análisis posterior, permitiendo investigar cada incidente de manera más minuciosa y con información estructurada para tomar decisiones informadas.

```
# **1. Configuración de logging**
logging.basicConfig(
    filename='detector.log',
    level=logging.WARNING,
    format='%(asctime)s - %(levelname)s - %(message)s'
)

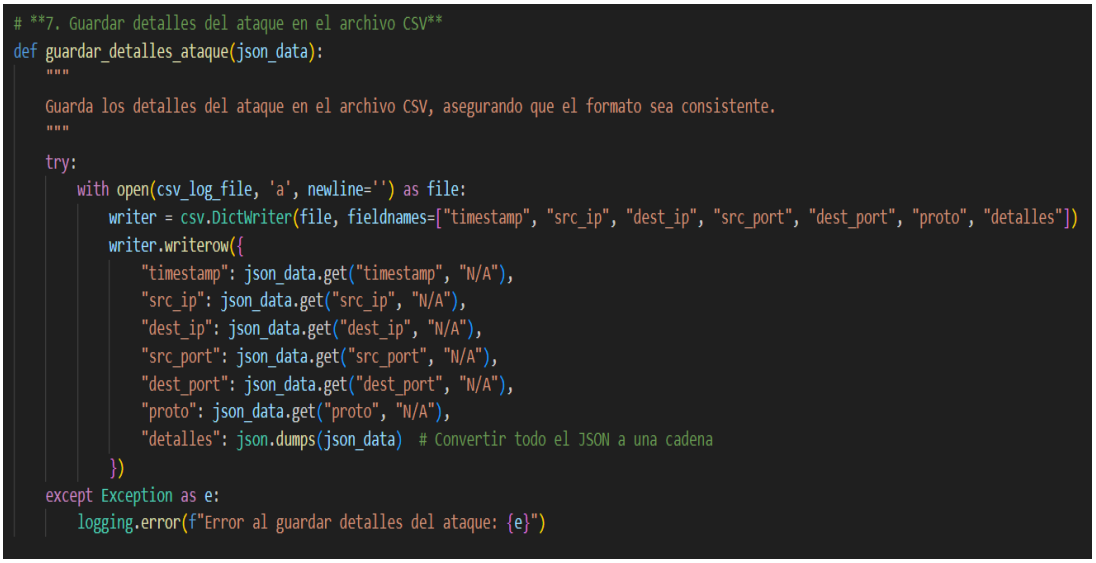
# **2. Archivo para guardar los detalles del ataque**
csv_log_file = 'detalles_ataques.csv'

# Crear el archivo CSV si no existe, con encabezados consistentes
if not os.path.exists(csv_log_file):
    with open(csv_log_file, 'w', newline='') as file:
        writer = csv.DictWriter(file, fieldnames=["timestamp", "src_ip", "dest_ip", "src_port", "dest_port", "proto", "detalles"])
        writer.writeheader()
```

Figura 59. Configuración para almacenar información de intentos de ataque.

Para implementar la funcionalidad de registro detallado de los intentos de ataque, se realizaron ajustes adicionales en el algoritmo, como se muestra en la Figura 59. En primer lugar, se habilitó un sistema de logging que genera un archivo llamado detector.log, diseñado para registrar información básica pero crucial sobre los eventos detectados. Este archivo utiliza un formato que incluye la fecha, el nivel de severidad y un mensaje descriptivo del evento, lo que permite una identificación rápida de los incidentes recientes. Esta funcionalidad es especialmente útil para monitorear el comportamiento general del sistema y responder de manera inmediata a cualquier actividad sospechosa.

Por otro lado, para un análisis más profundo, se configuró un archivo CSV denominado detalles_ataques.csv, que almacena información más completa y estructurada sobre los intentos de ataque detectados. Este archivo incluye encabezados claros como la marca de tiempo, las direcciones IP de origen y destino, los puertos involucrados, el protocolo utilizado y una descripción detallada del evento. En caso de que el archivo no exista, el script lo crea automáticamente e incluye los encabezados necesarios para garantizar la consistencia de los datos.



```python
# **7. Guardar detalles del ataque en el archivo CSV**
def guardar_detalles_ataque(json_data):
    """
    Guarda los detalles del ataque en el archivo CSV, asegurando que el formato sea consistente.
    """
    try:
        with open(csv_log_file, 'a', newline='') as file:
            writer = csv.DictWriter(file, fieldnames=["timestamp", "src_ip", "dest_ip", "src_port", "dest_port", "proto", "detalles"])
            writer.writerow({
                "timestamp": json_data.get("timestamp", "N/A"),
                "src_ip": json_data.get("src_ip", "N/A"),
                "dest_ip": json_data.get("dest_ip", "N/A"),
                "src_port": json_data.get("src_port", "N/A"),
                "dest_port": json_data.get("dest_port", "N/A"),
                "proto": json_data.get("proto", "N/A"),
                "detalles": json.dumps(json_data)  # Convertir todo el JSON a una cadena
            })
    except Exception as e:
        logging.error(f"Error al guardar detalles del ataque: {e}")
```

Figura 60. Función para guardar detalles de intentos de ataques en formato CSV.

La Figura 60 ilustra cómo se implementa la función encargada de registrar, de manera detallada, los intentos de ataque detectados en el archivo detalles_ataques.csv. Esta función juega un papel crucial al garantizar que cada evento identificado se almacene de forma ordenada y consistente, facilitando así el análisis posterior.

El proceso comienza abriendo el archivo CSV en un modo que permite añadir nuevos registros sin eliminar los existentes, asegurando que la información acumulada se conserve intacta. Luego, se define un diccionario con campos clave como la marca de tiempo (timestamp), las direcciones IP de origen y destino (src_ip y dest_ip), los puertos involucrados (src_port y dest_port), el protocolo (proto) y una descripción detallada del evento (detalles). Si alguno de estos datos no está disponible, el sistema completa el campo con "N/A" para garantizar que el formato se mantenga consistente en todos los registros.

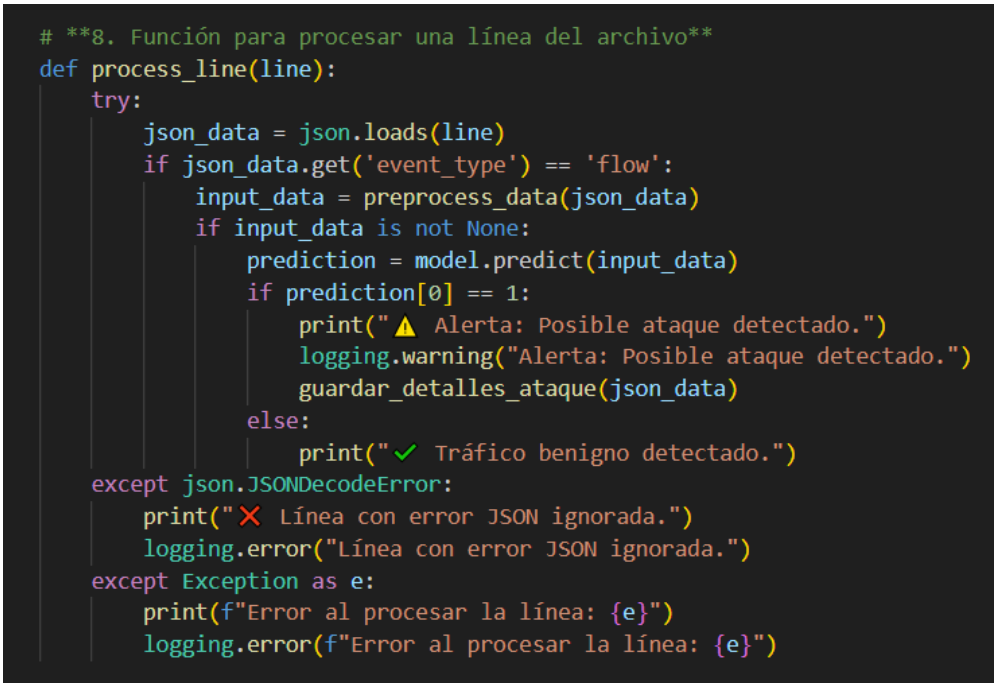


Figura 61. Modificaciones a la función de análisis de líneas de formato JSON.

La Figura 61 muestra una actualización importante en la función diseñada para analizar las líneas del archivo JSON generado por Suricata. Esta función no solo clasifica el tráfico como benigno o malicioso utilizando el modelo de machine learning, sino que ahora incluye una mejora clave: la capacidad de registrar los detalles de los intentos de ataque detectados.

El flujo de trabajo comienza al procesar cada línea del archivo JSON, transformándola en un formato estructurado. Si el tipo de evento corresponde a flow, los datos se preprocesan y se envían al modelo para su análisis. Cuando el modelo identifica un posible ataque (es decir, una predicción igual a 1), la función no solo genera una alerta en la consola y en los registros de logging, sino que también utiliza la función guardar_detalles_ataque para almacenar toda la información relevante en el archivo CSV detalles_ataques.csv. Esto asegura que cada intento de ataque quede documentado de manera detallada y organizada, permitiendo un análisis más profundo.

Por otro lado, si el tráfico es clasificado como benigno, el sistema lo registra como tal en la consola sin generar alertas innecesarias. Adicionalmente, se han implementado mecanismos de manejo de errores, como la capacidad de ignorar líneas JSON mal

formateadas o gestionar excepciones inesperadas, garantizando que el sistema siga operando sin interrupciones incluso en situaciones imprevistas.

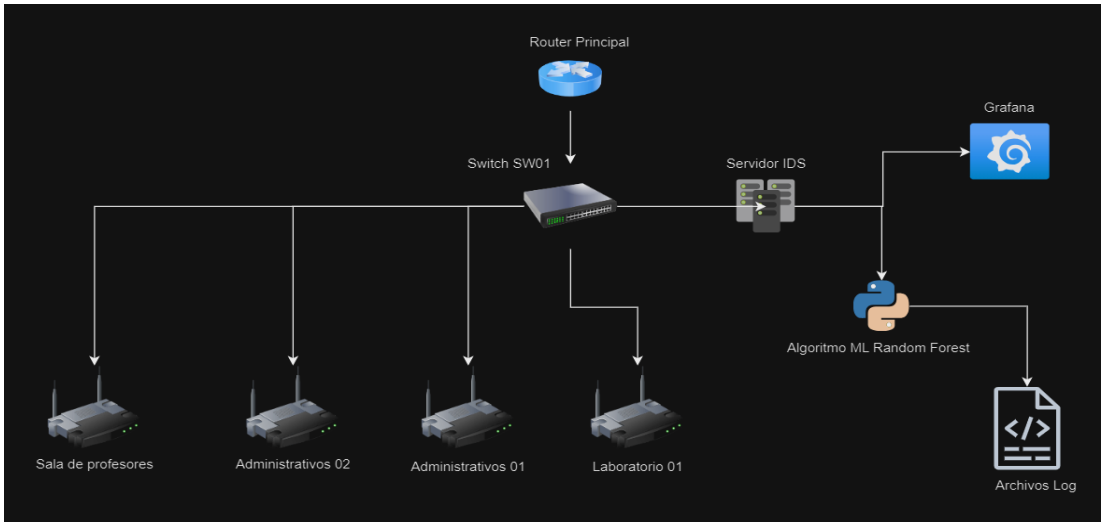


Figura 62. Arquitectura final de la red de la Unidad Educativa “19 de Septiembre”.

Tras implementar las modificaciones sugeridas al sistema IDS, se procedió a analizar y documentar la arquitectura final de la red de la Unidad Educativa "19 de Septiembre", como se ilustra en la Figura 62. La red está compuesta por un router principal que gestiona la conexión general, un switch principal (SW01) desde donde se distribuyen las conexiones hacia cuatro subredes: la sala de profesores, dos áreas administrativas y un laboratorio de informática. Además, el switch está directamente conectado al servidor IDS, que actúa como el núcleo del sistema de detección.

El servidor IDS no solo monitorea y analiza el tráfico en tiempo real, sino que también está vinculado al algoritmo de machine learning y al software de monitoreo Grafana. Esta integración permite detectar posibles intentos de ataque y generar visualizaciones detalladas del estado de la red. Adicionalmente, el algoritmo de machine learning registra información extraída de los eventos sospechosos en archivos log, brindando un respaldo para un análisis posterior más profundo por parte del personal encargado. Esta configuración asegura un balance eficiente entre monitoreo, análisis y documentación, optimizando la seguridad de la red de la institución.

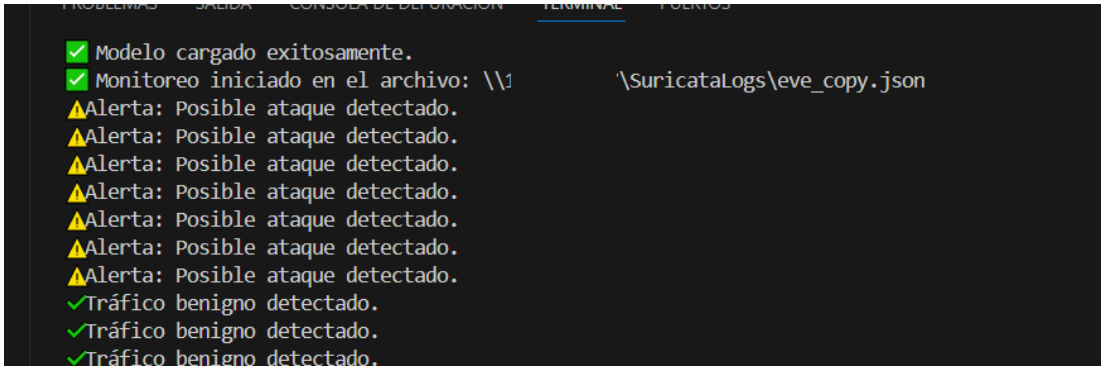


Figura 63. Ejecución de detección automática luego de cambios realizados.

Para garantizar que todos los componentes del sistema IDS integrado funcionen de manera óptima, se llevó a cabo una prueba de ejecución, utilizando el algoritmo presentado en Anexo H, durante un ataque simulado. Esta prueba permitió comprobar que el sistema opera correctamente en tiempo real, registrando tanto tráfico benigno como posibles intentos de ataque. Como se observa en la Figura 63, el sistema no solo identifica y clasifica el tráfico, sino que también genera alertas en la terminal para los eventos detectados como sospechosos o maliciosos.

El proceso comenzó con la carga exitosa del modelo de machine learning, seguido de la activación del monitoreo en el archivo eve_copy.json, que es el registro generado por Suricata. Durante la simulación del ataque, el sistema emitió alertas claras indicando "Posible ataque detectado" para los eventos maliciosos, mientras que el tráfico benigno fue identificado y clasificado correctamente, las alertas se encuentran explicadas a mayor detalle en Anexo I. Esta prueba no solo confirma que el sistema es capaz de integrar y coordinar todos sus componentes, sino que también valida su efectividad para detectar amenazas y garantizar la seguridad de la red en un entorno controlado.

Tabla 34. Ficha de aceptación de sistema IDS prueba 02 al encargado del laboratorio de computación en la Unidad Educativa “19 de Septiembre”.

Ficha de aceptación		
Persona aplicada	Responsable del laboratorio de computación	
Categoría	**Descripción**	**Calificación**
Funcionamiento General	Evaluar si el sistema IDS se ejecuta correctamente, asegurando la interacción adecuada entre Suricata, el modelo de machine learning y los scripts de integración.	Aprobado
Clasificación de Tráfico	Verificar que el sistema clasifique correctamente el tráfico benigno y malicioso, asegurando precisión en ambos casos.	Aprobado
Usabilidad del Sistema	Evaluar la facilidad de uso para el personal encargado, considerando la interpretación de alertas y la navegación en las funciones principales del sistema.	Aprobado
Integración General	Revisar si todos los componentes del sistema (Suricata, modelo de machine learning y scripts) trabajan de manera conjunta y sin errores.	Aprobado
Automatización General	Evaluar el nivel de automatización del sistema, incluyendo la sincronización de archivos, monitoreo en tiempo real y ejecución autónoma de procesos clave.	Aprobado
Observaciones		
Resultado de la prueba	Aprobado	

Finalmente, para validar la funcionalidad del sistema IDS y asegurar su implementación exitosa en la red de la Unidad Educativa “19 de Septiembre”, se aplicó una ficha de aceptación final, cuyos resultados se detallan en la Tabla 34. Esta evaluación fue llevada a cabo por el responsable del laboratorio de computación, quien examinó cada aspecto del sistema. Los criterios evaluados incluyeron el funcionamiento general del IDS, la precisión en la clasificación del tráfico, la usabilidad para el personal encargado, la integración de todos los componentes, y el nivel de automatización de los procesos clave.

El resultado obtenido fue "Aprobado" en todas las categorías, lo que confirma que el sistema cumple con los objetivos planteados y funciona de manera óptima en el entorno real. Esta aprobación valida no solo la interacción fluida entre Suricata, el modelo de machine learning y los scripts de integración, sino también la capacidad del sistema para operar de forma autónoma y eficiente, garantizando la seguridad de la red. Con este paso final, se cierra con éxito la fase de implementación del proyecto,

asegurando que el sistema IDS esté completamente preparado para proteger la infraestructura de red de la institución.

El modelo fue entrenado utilizando el conjunto de datos previamente procesado y balanceado con SMOTE, siguiendo la configuración descrita en la sección "Fase 5: Modelado". Random Forest, como lo menciona Fernández [45], combina los resultados de múltiples árboles de decisión para mejorar la precisión y robustez de las predicciones. En el caso de problemas de clasificación, la predicción final del modelo se determina mediante voto mayoritario, siguiendo la fórmula:

$$H(x) = \ moda\{h_1(x), h_2(x), \ldots . h_t(x)\}$$

Donde $H(x)$ representa la clase predicha para una instancia x, y $h_t(x)$ es la predicción individual del árbol t. Este enfoque asegura una mayor estabilidad en las predicciones al combinar decisiones individuales.

Además, el tiempo requerido para entrenar el modelo está influenciado por múltiples factores, como el número de árboles (T), la profundidad máxima de los árboles (D_t), el tamaño del conjunto de datos (N_d), y el número de características evaluadas en cada nodo (F). De manera aproximada, el tiempo de entrenamiento (T_{train}) puede expresarse como:

$$T_{train}\ \alpha\ T \cdot N_d \cdot\ D_t \cdot\ F$$

El modelo fue entrenado en un equipo con un procesador Intel Core i7 de novena generación y 16 GB de RAM, y requirió aproximadamente 18 horas para completar el proceso de ajuste con los hiperparámetros configurados. Este resultado es consistente con la complejidad de la tarea y la estructura del modelo, destacando la importancia de ajustar los parámetros como el número de árboles y la profundidad para optimizar el rendimiento del sistema.

El desempeño del sistema IDS se evaluó no solo en términos de su capacidad para clasificar correctamente el tráfico, sino también en cuanto a su tiempo de respuesta en tiempo real. El sistema está diseñado para operar en tiempo real, lo que significa que debe procesar las muestras de tráfico y generar alertas dentro de un plazo muy breve. En Tabla 35, se detallan las mediciones clave evaluadas durante la implementación del

sistema, las cuales aseguran que este cumpla con los estándares de desempeño esperado.

Tabla 35. Evaluación final de desempeño del sistema IDS.

Medición	Descripción	Valor Esperado	Referencia
Tiempo de Respuesta (En Tiempo Real)	Tiempo que tarda el sistema en procesar y clasificar una muestra de tráfico de red.	Menor a 1 segundo	Evaluación del sistema en tiempo real
Tiempo de Respuesta Promedio	Promedio del tiempo de respuesta bajo diferentes cargas de tráfico.	0.8 segundos	Medición en condiciones de carga normal de manera cronometrada.
Controles de Errores Activos	Métodos implementados para detectar y manejar errores durante la ejecución del sistema IDS.	Validación continua y manejo de excepciones	Detallado en el Anexo I
Precisión del Algoritmo (Tráfico Benigno)	Porcentaje de precisión del algoritmo al clasificar tráfico benigno.	94%	Evaluación realizada en la Fase 5: Evaluación
Precisión del Algoritmo (Tráfico Maligno)	Porcentaje de precisión del algoritmo al clasificar tráfico maligno.	100%	Evaluación realizada en la Fase 5: Evaluación

La precisión alcanzada en la clasificación de tráfico benigno y maligno, junto con los tiempos de respuesta optimizados, reflejan el cumplimiento de la integración del modelo de machine learning con Suricata. Además, la implementación de controles de errores activos ha asegurado la estabilidad operativa del sistema, garantizando su capacidad para mantener un funcionamiento constante y confiable. Con base en los resultados obtenidos durante las pruebas y la evaluación final, el sistema IDS está completamente acondicionado para ofrecer protección continua a la infraestructura de red de la Unidad Educativa "19 de Septiembre".

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El análisis de los resultados obtenidos mediante las herramientas de recolección de información identificó puntos críticos en la infraestructura tecnológica de la institución, como la insatisfacción de la población por frecuentes interrupciones en la conectividad, la ausencia de políticas formales de seguridad tecnológica y el bajo nivel de conocimiento en ciberseguridad entre los usuarios. Además, la red se encontraba en un estado significativo de vulnerabilidad, exponiéndola a posibles ataques, comprometiendo la seguridad de la información de la institución.

- Inicialmente, los resultados obtenidos con el dataset CSE-CIC-DoS 2019 fueron insatisfactorios. En la primera prueba, el conjunto incluía 2,931,699 instancias de tráfico malicioso y 2,129 de tráfico benigno, logrando una precisión del 61% para la Clase 0 (tráfico benigno) y un F1-Score general de 75%. Tras ajustar los hiperparámetros en una segunda prueba, los resultados no mejoraron significativamente. Para una tercera prueba, se complementaron los datos originales con información real recolectada en la institución y se aplicó SMOTE para balancear las clases, ampliando las instancias de tráfico benigno. El modelo final alcanzó una precisión del 94% para la Clase 0, 100% para la Clase 1 y un F1-Score ponderado de 99.98% en la validación cruzada.

- Tras evaluar diferentes algoritmos de machine learning, se seleccionó Random Forest, esta elección se fundamentó en un estudio comparativo que destacó su robustez, precisión y eficiencia en entornos IDS con limitaciones de hardware. Random Forest es capaz de manejar grandes volúmenes de datos de manera eficiente, equilibrando velocidad de entrenamiento y predicción con un uso moderado de recursos computacionales. Su tolerancia al ruido y a los outliers resultó complementaria al preprocesamiento aplicado al dataset, asegurando un mejor desempeño del modelo.

- La implementación de un IDS basado en machine learning en la Unidad Educativa “19 de Septiembre” ha sido documentada, proporcionando un marco para su instauración, configuración y evaluación. Se registraron los procedimientos desde la selección del modelo hasta su integración con Suricata, garantizando seguridad y estabilidad en la red. El modelo alcanzó un F1-Score de 99.98%, identificando el 100% del tráfico malicioso con una precisión del 94% para el tráfico benigno, optimizando la detección de intrusiones sin afectar la operatividad. Además, el sistema cumple con la tríada CIA: Confidencialidad, al registrar accesos no autorizados; Integridad, al analizar el tráfico en busca de manipulaciones; y Disponibilidad, al generar alertas oportunas para una respuesta inmediata.

4.2 Recomendaciones

- Es recomendable actualizar el algoritmo de detección cada seis meses utilizando los datos recolectados por el software IDS, lo que permitirá mantener su precisión frente a nuevas amenazas sin generar cambios innecesarios debido a un volumen bajo de datos. Para reentrenar el modelo, es prudente utilizar un equipo con procesador de la generación actual o una generación anterior y al menos 16 GB de memoria RAM, lo que asegurará un proceso eficiente y ágil, optimizando los recursos y tiempo de entrenamiento, asegurando que el modelo permanezca actualizado para detectar patrones recientes.

- Realizar mantenimientos trimestrales del servidor IDS es esencial para garantizar su funcionamiento óptimo. Estos mantenimientos deben incluir la actualización del sistema operativo a su última versión estable, asegurando que las vulnerabilidades conocidas sean corregidas. También se debe revisar y optimizar las configuraciones clave, como reglas de detección y políticas de acceso, mediante una auditoría detallada realizada por el personal de TI capacitado. Es fundamental verificar los logs generados por el IDS, identificando patrones anómalos o alertas recurrentes que puedan indicar problemas subyacentes.

- Organizar charlas educativas semestrales dirigidas a la comunidad educativa, así como capacitaciones trimestrales específicas para el personal encargado del área tecnológica, es fundamental para fortalecer la ciberseguridad de la institución. Las charlas deben enfocarse en concientizar sobre los riesgos asociados a la exposición de datos en la red y promover prácticas responsables para proteger la información institucional. Por otro lado, las capacitaciones para el personal técnico deben abordar temas avanzados como estrategias de mitigación de amenazas, manejo de incidentes y actualización en herramientas de seguridad, asegurando que estén preparados para proteger la red frente a amenazas emergentes.

- Fortalecer el sistema de detección de intrusiones mediante la integración de módulos adicionales resulta pertinente para mejorar su funcionalidad. Entre los módulos prioritarios se sugieren: módulos de correlación de eventos, que permitan analizar datos históricos y correlacionar alertas para identificar patrones complejos de ataques, módulos de integración con sistemas de respuesta automatizada, que faciliten la toma de decisiones y la ejecución de medidas preventivas en tiempo real. Estos módulos deben enfocarse en complementar las capacidades de detección del modelo actual, optimizar el flujo de trabajo del personal técnico y anticiparse a nuevas amenazas, fortaleciendo así la seguridad de la red.

REFERENCIAS BIBLIOGRÁFICAS

[1] IBM, “Cost of a Data Breach Report 2023”, 2023.

[2] Kaspersky Team, “Nueva epidemia: el phishing se sextuplicó en América Latina con el reinicio de la actividad económica y el apoyo de la IA”.

[3] L. Loja *et al.*, “Modelo de madurez de ciberseguridad para infraestructuras críticas caso de estudio: Ecuador”, 2023, doi: 10.29018/issn.2588-1000vol7iss48.

[4] J. D. Chuqui Sulca, “Ciberseguridad en la gestión de riesgos en una institución educativa, Callao 2023”.

[5] D. A. Villamar Arellano, “ESTRATEGIAS DE PREVENCIÓN FRENTE A LOS CIBERATAQUES EN LA UNIDAD EDUCATIVA FISCAL LUIS ALFREDO NOBOA ICAZA”, 2022.

[6] C. Liu, Z. Gu, y J. Wang, “A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning”, *IEEE Access*, vol. 9, pp. 75729–75740, 2021, doi: 10.1109/ACCESS.2021.3082147.

[7] V. Sstla, V. K. K. Kolli, L. K. Voggu, R. Bhavanam, y S. Vallabhasoyula, “Predictive model for network intrusion detection system using deep learning”, *Revue d’Intelligence Artificielle*, vol. 34, núm. 3, pp. 323–330, jun. 2020, doi: 10.18280/ria.340310.

[8] S. Prasath, K. Sethi, D. Mohanty, P. Bera, y S. R. Samantaray, “Analysis of Continual Learning Models for Intrusion Detection System”, *IEEE Access*, vol. 10, pp. 121444–121464, 2022, doi: 10.1109/ACCESS.2022.3222715.

[9] W. Villegas Ch, S. Jácome Vásconez, J. García Ortiz, J. Calvache Sánchez, y S. Sánchez Viteri, “Proposal of a Model for the Analysis of the State of the Use of ICT in Education Applied to Technological Institutes of Higher Education”, *Computers*, vol. 11, núm. 7, jul. 2022, doi: 10.3390/computers11070112.

[10] A. M. Robalino Tubón, “OPTIMIZACIÓN DE LA INFRAESTRUCTURA DE RED LAN MEDIANTE LA APLICACIÓN DE UNA METODOLOGÍA DE DISEÑO DE REDES EN LA UNIDAD EDUCATIVA BOLÍVAR”, 2024.

[11] Universidad Latina de Costa Rica, “¿Qué son las TIC y para qué sirven?”, Ulatina.ac.cr.

[12] F. E. Bravo Guerrero y T. G. Quezada Matute, “Uso de Tecnologías de la Información y Comunicación en el Bachillerato”, *Revista Electrónica Cooperación Universidad Sociedad (RECUS)*, vol. 6, pp. 19–27, 2021.

[13] F. X. Morales García y J. L. Zambrano Martínez, “Revisión de la literatura de las metodologías de ciberseguridad en plataformas bancarias”, 2023.

[14] H. Rodriguez Bazan, G. Sidorov, y J. Escamilla Ambrosio, “State of the Art Review of Natural Language Processing Techniques for Malware Analysis”, 2020.

[15] E. Ariel, C. Rivera, M. Andrés, C. Murillo, I. J. Bladimir, y R. Peñaherrera, “EVALUACIÓN DE ATAQUES DDOS A UN SISTEMA DE RED Y SUS DIFERENTES FORMAS DE PROTECCIÓN”, 2022.

[16] F. Narváez, “SISTEMAS DE SEGURIDAD DE LA INFORMACIÓN ‘EMPRESA ASIC’ – INVESTIGACIÓN”, 2022.

[17] R. Iglesias, “Análisis actual de los Ataques Man-In-The-Middle por DNS Spoofing”, 2022.

[18] E. Benavides, W. Fuertes, y S. Sanchez, “Caracterización de los ataques de phishing y técnicas para mitigarlos. Ataques: una revisión sistemática de la literatura”, *Ciencia y Tecnología*, vol. 13, núm. 1, pp. 97–104, jun. 2020, doi: 10.18779/cyt.v13i1.357.

[19] F. J. Martínez Pérez, “ANÁLISIS DE LOS CIBERATAQUES. EL ATAQUE MAN-IN-THE-MIDDLE Y EL SSLSTRIP”, 2023.

[20] J. E. D. Chavez Flores, J. C. J. Pacheco Guzmán, y A. C. Mendoza de los Santos, “El papel de la inteligencia artificial en la seguridad de la información: una revisión de su aplicación en la industria cibernética”, *Revista de investigación de Sistemas e Informática*, vol. 16, núm. 1, pp. 71–80, ago. 2023, doi: 10.15381/risi.v16i1.25390.

[21] D. Sudyana *et al.*, “Improving Generalization of ML-based IDS with Lifecycle-based Dataset, Auto-Learning Features, and Deep Learning”, *IEEE Transactions on Machine Learning in Communications and Networking*, pp. 1–1, may 2024, doi: 10.1109/tmlcn.2024.3402158.

[22] M. A. Siddiqi y W. Pak, “Tier-Based Optimization for Synthesized Network Intrusion Detection System”, *IEEE Access*, vol. 10, pp. 108530–108544, 2022, doi: 10.1109/ACCESS.2022.3213937.

[23] Y. Cruz-Aguayo *et al.*, “Hacia una transformación digital del sector educativo: aprendizajes de la virtualización de emergencia”, 2022. [En línea]. Disponible en: http://www.iadb.org

[24] U. Y. Sociedad *et al.*, “Las tecnologías de la información y comunicación en la gestión del conocimiento”, 2021.

[25] K. J. Ramírez González, “TÉCNICAS PARA MEJORAR EL ANCHO DE BANDA EN ANTENAS TIPO PARCHE ANTENA TIPO PARCHE EN DOBLE SUSTRATO PARA MEJORAR EL ANCHO DE BANDA”, 2023.

[26] E. Alfonso, R. Montalbán, R. J. Martelo Gómez, D. Antonio, y F. Borré, “DESIGN OF AN INFORMATION SECURITY MANAGEMENT SYSTEM FOR THE PROCESS OF MANAGING THE TECHNOLOGICAL INFRASTRUCTURE OF ACADEMIC INSTITUTIONS BASED ON MAGERIT”, 2019.

[27] R. M. Ampuero Herrera, “Gestión de riesgos de la información basado en la metodología Magerit para una Notaría de la Región Lima”, 2021.

[28] C. Graciano Santelises, “Adolescentes y el uso de Internet: Resumen del estudio uso y riesgos en el uso de internet de adolescentes escolarizados entre 12 y 17 años con enfoque de explotación sexual en línea”, sep. 2021.

[29] J. L. PÉREZ SALAGATA, “APLICACIONES MÓVILES PARA ANALIZAR EL ÍNDICE DE MASA CORPORAL DE ESTUDIANTES DE EDUCACIÓN GENERAL BÁSICA SUPERIOR”, 2024.

[30] Open Information Security Foundation, “Suricata Documentation”. Consultado: el 22 de diciembre de 2024. [En línea]. Disponible en: https://docs.suricata.io/en/latest/index.html

[31] K. Ishaq y H. Ahsan Javed, “Implementing Snort Intrusion Prevention System (IPS) for Network Forensic Analysis”, ago. 2023.

[32] I. G. Vitas, B. Guaita Pérez, y H. R. Pous, “Desplegar la herramienta ‘Zeek IDS’ y su posterior explotación para el análisis de actividades sospechosas en la red”, jun. 2020.

[33] J. J. Espinosa Zúñiga, “Aplicación de metodología CRISP-DM para segmentación geográfica de una base de datos pública”, *Ingeniería Investigación y Tecnología*, vol. 21, núm. 1, pp. 1–13, ene. 2020, doi: 10.22201/fi.25940732e.2020.21n1.008.

[34] B. Molina-Coronado, U. Mori, A. Mendiburu, y J. Miguel-Alonso, “Survey of Network Intrusion Detection Methods from the Perspective of the Knowledge Discovery in Databases Process”, ene. 2020.

[35] B. Ferreira, S. Marques, M. Kalinowski, H. Lopes, y S. D. J. Barbosa, “Lessons learned to improve the UX practices in agile projects involving data science and process automation”, *Inf Softw Technol*, vol. 155, mar. 2023, doi: 10.1016/j.infsof.2022.107106.

[36] N. Moustafa y J. Slay, “UNSW-NB15: A Comprehensive Data set for Network Intrusion Detection systems (UNSW-NB15 Network Data Set)”. [En línea]. Disponible en: https://cve.mitre.org/

[37] Canadian Institute for Cybersecurity, “DDoS evaluation dataset (CIC-DDoS2019)”. Consultado: el 18 de diciembre de 2024. [En línea]. Disponible en: https://www.unb.ca/cic/datasets/ddos-2019.html

[38] Canadian Institute for Cybersecurity, “CIC IoV dataset 2024 Advancing Realistic IDS Approaches against DoS and Spoofing Attack in IoV CAN bus”. Consultado: el 18 de diciembre de 2024. [En línea]. Disponible en: https://www.unb.ca/cic/datasets/iov-dataset-2024.html

[39] V. Malele y T. E. Mathonsi, “Testing the performance of Multi-class IDS public dataset using Supervised Machine Learning Algorithms”, mar. 2023.

[40] E. Bartz, M. Zaefferer, O. Mersmann, y T. Bartz-Beielstein, “Experimental Investigation and Evaluation of Model-based Hyperparameter Optimization”, jul. 2021, [En línea]. Disponible en: http://arxiv.org/abs/2107.08761

[41] V. R. Joseph y A. Vakayil, “SPlit: An Optimal Method for Data Splitting”, dic. 2020, doi: 10.1080/00401706.2021.1921037.

[42] Elastic, “Características de Kibana”. Consultado: el 2 de enero de 2025. [En línea]. Disponible en: https://www.elastic.co/es/kibana/features?utm_source=chatgpt.com

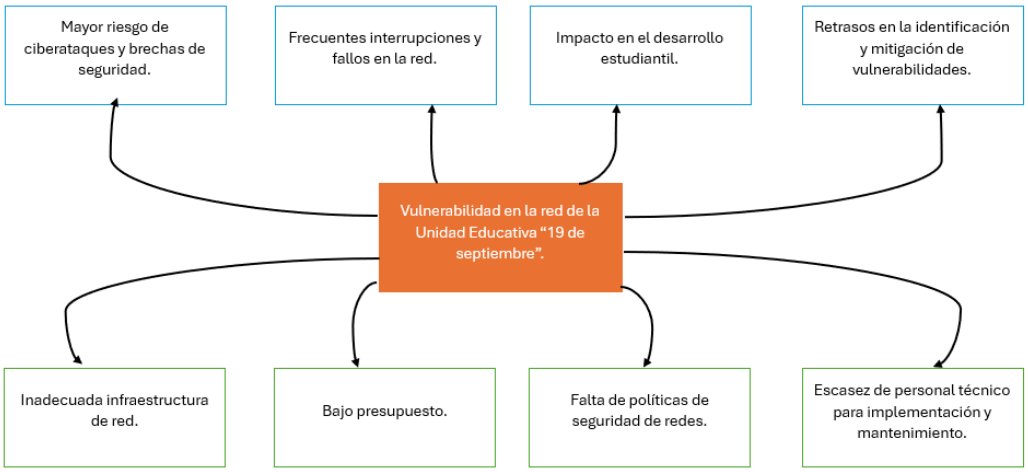
[43] Grafana Labs, “Grafana fundamentals”. Consultado: el 2 de enero de 2025. [En línea]. Disponible en: https://grafana.com/tutorials/grafana-fundamentals/#introduction

[44] Zabbix LLC, “Zabbix Manual”. Consultado: el 2 de enero de 2025. [En línea]. Disponible en: https://www.zabbix.com/documentation/current/es/manual

[45] A. Fernández, *Árboles de decisión en R con Random Forest*. 2023.

ANEXOS

Anexo A. Árbol de problemas



Anexo B. Código de limpieza de datos opción 1.

```
# Importación de bibliotecas necesarias
import pandas as pd

# Configuración y parámetros
chunk_size = 100000  # Tamaño del chunk para lectura en trozos del
archivo
columnas_utilizadas = [
    'Flow Duration', 'Total Fwd Packets', 'Total Backward Packets',
    'Total Length of Fwd Packets', 'Total Length of Bwd Packets',
    'Flow Bytes/s', 'Flow Packets/s',
    'SYN Flag Count', 'ACK Flag Count', 'RST Flag Count',
    'Active Mean', 'Active Std', 'Idle Mean', 'Idle Std',
    'Min Packet Length', 'Max Packet Length', 'Packet Length Mean',
    'Label'  # Variable objetivo
]
columnas_para_outliers = [
    'Flow Duration', 'Total Fwd Packets', 'Total Backward Packets',
    'Total Length of Fwd Packets', 'Total Length of Bwd Packets',
    'Flow Bytes/s', 'Flow Packets/s'
]

# Lista para almacenar los chunks procesados
processed_chunks = []

# Lectura del archivo en chunks
data_chunks = pd.read_csv(
    '../DataSets/DrDoS_DNS.csv',  # Ruta del archivo de datos
    chunksize=chunk_size,        # Tamaño del chunk
    on_bad_lines='skip',         # Ignorar líneas problemáticas
    low_memory=False             # Evitar advertencias de bajo uso
de memoria
)

# Funciones
def procesar_chunk(chunk):
    """
    Procesa un chunk de datos, aplicando varias transformaciones y
limpieza.

    Parámetros:
    ----------
    chunk : pd.DataFrame
        Chunk de datos leído desde el archivo CSV.

    Retorna:
    -------
    pd.DataFrame
        Chunk procesado y limpio.
```

```
    """
    # Limpieza de nombres de columnas
    chunk.columns = chunk.columns.str.strip()

    # Filtrado de columnas relevantes
    chunk = chunk[columnas_utilizadas]

    # Eliminación de duplicados
    chunk = chunk.drop_duplicates()

    # Eliminación de filas con valores nulos
    chunk = chunk.dropna()

    # Conversión de columnas int64 a int32 para optimizar memoria
    int_columns = chunk.select_dtypes(include='int64').columns
    chunk[int_columns] = chunk[int_columns].astype('int32')

    # Conversión de columnas float64 a float32 para optimizar
memoria
    float_columns = chunk.select_dtypes(include='float64').columns
    chunk[float_columns] = chunk[float_columns].astype('float32')

    # Eliminación de outliers en las columnas seleccionadas
    for col in columnas_para_outliers:
        Q1 = chunk[col].quantile(0.25)  # Primer cuartil
        Q3 = chunk[col].quantile(0.75)  # Tercer cuartil
        IQR = Q3 - Q1  # Rango intercuartílico
        # Filtrado de valores dentro de 1.5 veces el rango
intercuartílico
        chunk = chunk[(chunk[col] >= Q1 - 1.5 * IQR) & (chunk[col]
<= Q3 + 1.5 * IQR)]

    return chunk

# Procesamiento
for chunk in data_chunks:
    """
    Itera sobre los chunks del archivo, procesándolos uno a uno
    y almacenando los resultados en una lista.
    """
    chunk_procesado = procesar_chunk(chunk)
    processed_chunks.append(chunk_procesado)

# Combinación de chunks procesados
data_final = pd.concat(processed_chunks, ignore_index=True)

# Verificación de uso de memoria y tipos de datos
print(data_final.info(memory_usage='deep'))

# Guardado del dataset limpio
data_final.to_csv(
```

```
    '../DataSets/DatasetsRefinados/DrDoS_DNS_clean.csv',  # Ruta de
salida
    index=False                                           # Sin
índices en el archivo de salida
)
print("Dataset limpio guardado con éxito.")
```

Anexo C Código de limpieza de datos opción 2.

```
# Importación de bibliotecas necesarias
import pandas as pd

# Lectura y configuración inicial
data = pd.read_csv(
    '../DataSets/DrDoS_LDAP.csv',  # Ruta del archivo de entrada
    low_memory=False               # Evitar advertencias
relacionadas con el uso de memoria
)
print("Primeras 5 filas del dataset:")
print(data.head(5))

# Información sobre las columnas del dataset
print("Columnas presentes en el dataset:", data.columns)

# Limpieza inicial del dataset
# Eliminar espacios adicionales en los nombres de las columnas
data.columns = data.columns.str.strip()

# Definición de columnas relevantes para detección de intrusos
columnas_utilizadas = [
    'Flow Duration', 'Total Fwd Packets', 'Total Backward Packets',
    'Total Length of Fwd Packets', 'Total Length of Bwd Packets',
    'Min Packet Length', 'Max Packet Length', 'Packet Length Mean',
    'SYN Flag Count', 'ACK Flag Count', 'RST Flag Count',
    'Active Mean', 'Active Std', 'Idle Mean', 'Idle Std', 'Label'
]

# Filtrado para conservar solo las columnas relevantes
data = data[columnas_utilizadas]

# Eliminación de duplicados
data = data.drop_duplicates()

# Verificación y manejo de valores nulos
print("Valores nulos por columna antes de limpieza:\n",
data.isnull().sum())
data = data.dropna()  # Eliminación de filas con valores nulos

# Optimización de tipos de datos
# Verificar los tipos de datos actuales
print("Tipos de datos iniciales:\n", data.dtypes)

# Convertir columnas int64 a int32
int_columns = data.select_dtypes(include='int64').columns
data[int_columns] = data[int_columns].astype('int32')

# Convertir columnas float64 a float32
```

```
float_columns = data.select_dtypes(include='float64').columns
data[float_columns] = data[float_columns].astype('float32')

# Revisar el uso de memoria después de la optimización
print("Uso de memoria después de la optimización:")
print(data.info(memory_usage='deep'))

# Eliminación de outliers
# Definición de columnas donde se eliminarán outliers
columnas_para_outliers = [
    'Flow Duration', 'Total Fwd Packets', 'Total Backward Packets',
    'Total Length of Fwd Packets', 'Total Length of Bwd Packets',
    'Flow Bytes/s', 'Flow Packets/s'
]

# Aplicar eliminación de outliers
for col in columnas_para_outliers:
    if col in data.columns:  # Verificar que la columna exista
        Q1 = data[col].quantile(0.25)  # Primer cuartil
        Q3 = data[col].quantile(0.75)  # Tercer cuartil
        IQR = Q3 - Q1  # Rango intercuartílico
        # Filtrar valores dentro de 1.5 veces el rango
intercuartílico
        data = data[(data[col] >= Q1 - 1.5 * IQR) & (data[col] <= Q3
+ 1.5 * IQR)]

# Información después de eliminar outliers
print(f"Dimensiones del dataset después de eliminar outliers:
{data.shape}")

# Verificación final de tipos de datos y uso de memoria
print("Tipos de datos después de la limpieza:\n", data.dtypes)
print("Uso de memoria final:")
print(data.info(memory_usage='deep'))

# Guardado del dataset limpio
data.to_csv(
    '../DataSets/DatasetsRefinados/DrDoS_LDAP.csv',  # Ruta del
archivo limpio
    index=False                                      # Sin incluir
el índice en el archivo de salida
)
print("Dataset limpio guardado con éxito.")
```

Anexo D. Código de unión de datasets limpios en dataset universal.

```
# Importación de bibliotecas necesarias
import pandas as pd
import glob

# Rutas de los archivos filtrados
rutas = [
    '../DataSets/DatasetsRefinados/DrDoS_DNS_clean.csv',
    '../DataSets/DatasetsRefinados/DrDoS_LDAP.csv',
    '../DataSets/DatasetsRefinados/DrDoS_MSSQL_clean.csv',
    '../DataSets/DatasetsRefinados/DrDoS_NTP_clean.csv',
    '../DataSets/DatasetsRefinados/DrDoS_NetBIOS_clean.csv',
    '../DataSets/DatasetsRefinados/DrDoS_SNMP_clean.csv',
    '../DataSets/DatasetsRefinados/DrDoS_SSDP_clean.csv',
    '../DataSets/DatasetsRefinados/DrDoS_UDP_clean.csv',
    '../DataSets/DatasetsRefinados/LDAP_clean.csv',
    '../DataSets/DatasetsRefinados/MSSQL_clean.csv',
    '../DataSets/DatasetsRefinados/NetBIOS_clean.csv',
    '../DataSets/DatasetsRefinados/Portmap_clean.csv',
    '../DataSets/DatasetsRefinados/Syn2_clean.csv',
    '../DataSets/DatasetsRefinados/Syn_clean.csv',
    '../DataSets/DatasetsRefinados/TFTP_clean.csv',
    '../DataSets/DatasetsRefinados/UDPLag_clean.csv',
    '../DataSets/DatasetsRefinados/UDP_clean.csv',
    '../DataSets/DatasetsRefinados/UDPLag2_clean.csv'
]

# Lectura y concatenación de archivos
# Se leen los archivos especificados en la lista "rutas" y se
combinan en un único DataFrame
dataframes = [pd.read_csv(file, low_memory=False) for file in
rutas]  # Leer los archivos
datasetUniversal = pd.concat(dataframes,
ignore_index=True)          # Combinar en un único DataFrame

# Verificar información básica del dataset combinado
print("Tamaño del dataset combinado:",
datasetUniversal.shape)      # Dimensiones del dataset
print("Columnas del dataset combinado:",
datasetUniversal.columns)  # Nombres de las columnas

# Uso de memoria antes de la optimización
print("Uso de memoria antes de la optimización:")
print(datasetUniversal.info(memory_usage='deep'))

# Optimización de tipos de datos
# Convertir columnas int64 a int32 para reducir el uso de memoria
int_columns =
datasetUniversal.select_dtypes(include='int64').columns
```

```
datasetUniversal[int_columns] =
datasetUniversal[int_columns].astype('int32')

# Convertir columnas float64 a float32 para reducir el uso de
memoria
float_columns =
datasetUniversal.select_dtypes(include='float64').columns
datasetUniversal[float_columns] =
datasetUniversal[float_columns].astype('float32')

# Verificar el uso de memoria después de la optimización
print("Uso de memoria después de la optimización:")
print(datasetUniversal.info(memory_usage='deep'))

# Guardar el dataset combinado en un archivo CSV
datasetUniversal.to_csv(
    '../DataSetsParaPreprocesar/DataUniversal.csv',  # Ruta del
archivo de salida
    index=False                                      # No incluir
índices en el archivo
)
print("Dataset combinado y limpio guardado con éxito.")
```

```
# importación de librerías necesarias
from flask import Flask, jsonify
import os
import json
import time

# Inicialización de la aplicación Flask
app = Flask(__name__)

# Configuración global: Intervalo de actualización en segundos
UPDATE_INTERVAL = 60  # Tiempo entre actualizaciones de caché
(ajustable según necesidades)

# Variables globales para almacenar el estado del sistema
last_update_time = 0  # Almacena el tiempo de la última
actualización
cached_data = []  # Almacena los datos en caché para optimizar las
consultas


def load_logs():
    """
    Carga y procesa los logs desde el archivo eve.json generado por
Suricata.
    Utiliza una estrategia de caché para reducir la frecuencia de
lectura del archivo.

    Returns:
        list: Lista de logs extraídos del archivo eve.json.
        tuple: En caso de error, retorna un diccionario con el
mensaje de error y el código HTTP.
    """
    global last_update_time, cached_data
    try:
        # Ruta predeterminada al archivo de logs eve.json
        file_path = '/var/log/suricata/eve.json'

        # Verificar si el archivo existe
        if not os.path.exists(file_path):
            return {"error": "El archivo eve.json no existe en la
ruta especificada"}, 404

        # Verificar si los datos deben actualizarse según el
intervalo definido
        current_time = time.time()
        if current_time - last_update_time > UPDATE_INTERVAL or not
cached_data:
            with open(file_path, 'r') as file:
```

```
                data = []
                for line in file:
                    try:
                        # Intentar cargar cada línea como JSON
                        data.append(json.loads(line))
                    except json.JSONDecodeError:
                        # Ignorar líneas no válidas
                        continue

                # Actualizar el caché y el tiempo de la última
actualización
                cached_data = data
                last_update_time = current_time

        # Retornar los datos en caché
        return cached_data
    except Exception as e:
        # Manejar errores genéricos y retornar como respuesta de
error
        return {"error": str(e)}, 500


@app.route('/', methods=['GET'])
def home():
    """
    Verifica que la API está funcionando correctamente.

    Returns:
        JSON: Mensaje indicando que la API está activa.
    """
    return jsonify({"message": "API de Suricata funcionando
correctamente"})


@app.route('/logs', methods=['GET'])
def get_logs():
    """
    Proporciona acceso a los datos del archivo eve.json.
    Utiliza la caché para optimizar el rendimiento y reduce la
sobrecarga
    al limitar la cantidad de datos retornados.

    Returns:
        JSON: Los primeros 100 registros de logs desde eve.json o un
mensaje de error.
    """
    try:
        # Cargar los datos desde el archivo utilizando la caché
        data = load_logs()

        # Verificar si hubo un error al cargar los datos
```

```
        if isinstance(data, tuple):
            return jsonify(data[0]), data[1]

        # Retornar los primeros 100 registros para evitar sobrecarga
en la respuesta
        return jsonify({"logs": data[:100]})
    except Exception as e:
        # Manejar cualquier error no controlado y retornar como
respuesta
        return jsonify({"error": str(e)}), 500


if __name__ == '__main__':
    """
    Punto de entrada de la aplicación.
    Configura la API para ejecutarse en el puerto 5000 y ser
accesible desde cualquier dirección.
    """
    app.run(host='0.0.0.0', port=5000)
```

Anexo F. Código final de preprocesamiento de dataset universal.

```
# Importacion de librerías necesarias
import pandas as pd
from sklearn.preprocessing import MinMaxScaler  # Utilizado para
normalización
import seaborn as sns
import matplotlib.pyplot as plt
import json
from datetime import datetime

  # ---------------------- SECCIÓN 1: PREPROCESAMIENTO DEL DATASET
                  ORIGINAL ----------------------

# Cargar el dataset original
data = pd.read_csv('../DataSetsParaPreprocesar/DataUniversal.csv')

# Verificar la distribución de la variable objetivo
print("Distribución inicial de la variable Label:")
print(data['Label'].value_counts())

# Transformar la columna 'Label' a valores binarios (0: BENIGN, 1:
MALICIOUS)
data['Label'] = data['Label'].apply(lambda x: 0 if x == 'BENIGN'
else 1)
print("Distribución de la variable Label después de
transformación:")
print(data['Label'].value_counts())

# Generar y visualizar la matriz de correlación
correlation_matrix = data.corr()
plt.figure(figsize=(16, 12))
sns.heatmap(
    correlation_matrix,
    annot=True,
    fmt=".2f",
    cmap="coolwarm",
    annot_kws={"size": 8},
    cbar_kws={"shrink": 0.8}
)
plt.xticks(rotation=45, ha="right", fontsize=10)
plt.yticks(rotation=0, fontsize=10)
plt.tight_layout()
plt.show()

# Guardar el dataset preprocesado
data.to_csv('../DataSetsParaPreprocesar/DataUniversalProcesado.csv',
index=False)
print("Dataset procesado guardado exitosamente.")
```

```
# ---------------------- SECCIÓN 2: PROCESAMIENTO DE LOGS SURICATA -
                        --------------------

# Columnas seleccionadas del dataset
columnas_seleccionadas = [
    'Flow Duration', 'Total Fwd Packets', 'Total Backward Packets',
    'Total Length of Fwd Packets', 'Total Length of Bwd Packets',
    'Flow Bytes/s', 'Flow Packets/s', 'SYN Flag Count', 'ACK Flag
Count',
    'RST Flag Count', 'Active Mean', 'Active Std', 'Idle Mean',
'Idle Std',
    'Min Packet Length', 'Max Packet Length', 'Packet Length Mean',
'Label'
]

# Ruta al archivo eve.json generado por Suricata
ruta_eve_json = '../eve_copy.json'

def procesar_entrada_suricata(entry):
    """
    Procesa una entrada individual del archivo eve.json.
    Args:
        entry (dict): Registro individual del archivo eve.json.
    Returns:
        dict: Diccionario con las columnas seleccionadas.
    """
    try:
        flow = entry.get('flow', {})
        start_time = pd.to_datetime(flow.get('start'))
        end_time = pd.to_datetime(flow.get('end'))
        flow_duration = (end_time - start_time).total_seconds() *
1000
    except Exception as e:
        print(f"Error calculando Flow Duration: {e}")
        flow_duration = 0

    return {
        'Flow Duration': flow_duration,
        'Total Fwd Packets': flow.get('pkts_toserver', 0),
        'Total Backward Packets': flow.get('pkts_toclient', 0),
        'Total Length of Fwd Packets': flow.get('bytes_toserver',
0),
        'Total Length of Bwd Packets': flow.get('bytes_toclient',
0),
        'Label': 0  # Asume tráfico benigno por defecto
    }

def leer_y_procesar_eve_json(ruta):
    """
    Procesa los registros del archivo eve.json.
    Args:
```

```
        ruta (str): Ruta al archivo eve.json.
    Returns:
        DataFrame: DataFrame con los registros procesados.
    """
    registros = []
    with open(ruta, 'r') as archivo:
        for linea in archivo:
            try:
                entrada = json.loads(linea)
                if entrada.get('event_type') == 'flow':
                    registros.append(procesar_entrada_suricata(entra
da))
            except json.JSONDecodeError as e:
                print(f"Error procesando línea: {e}")
    return pd.DataFrame(registros)

# Procesar datos de Suricata
dataset_suricata = leer_y_procesar_eve_json(ruta_eve_json)
print(f"Registros procesados de Suricata: {len(dataset_suricata)}")

# ---------------------- SECCIÓN 3: UNIFICACIÓN Y LIMPIEZA FINAL ---
                        -------------------

# Cargar el dataset original preprocesado
dataset_original =
pd.read_csv('../DataSetsParaPreprocesar/DataUniversalProcesado.csv')

# Seleccionar columnas necesarias
dataset_original = dataset_original[columnas_seleccionadas]

# Combinar datasets
dataset_final = pd.concat([dataset_original, dataset_suricata],
ignore_index=True)

# Guardar el dataset combinado
dataset_final.to_csv('../DataSetsParaPreprocesar/DataSetDefinitivo.c
sv', index=False)
print(f"Dataset combinado guardado exitosamente con
{len(dataset_final)} registros.")

    # ---------------------- SECCIÓN 4: ELIMINACIÓN DE OUTLIERS Y
                VALORES NULOS ----------------------

# Eliminar duplicados
dataset_final = dataset_final.drop_duplicates()

# Identificar y reemplazar valores nulos
columnas_con_nulos =
dataset_final.columns[dataset_final.isnull().sum() > 0]
for col in columnas_con_nulos:
    if dataset_final[col].dtype in ['float64', 'int64']:
```

```
        dataset_final[col] =
dataset_final[col].fillna(dataset_final[col].mean())
    else:
        dataset_final[col] = dataset_final[col].fillna(0)

# Eliminar outliers en columnas seleccionadas
columnas_para_outliers = [
    'Flow Duration', 'Total Fwd Packets', 'Total Backward Packets',
    'Total Length of Fwd Packets', 'Total Length of Bwd Packets'
]
for col in columnas_para_outliers:
    Q1 = dataset_final[col].quantile(0.25)
    Q3 = dataset_final[col].quantile(0.75)
    IQR = Q3 - Q1
    dataset_final = dataset_final[(dataset_final[col] >= Q1 - 1.5 *
IQR) & (dataset_final[col] <= Q3 + 1.5 * IQR)]

# Guardar el dataset limpio final
ruta_dataset_definitivo =
'../DataSetsParaPreprocesar/DataDefinitivoLimpio.csv'
dataset_final.to_csv(ruta_dataset_definitivo, index=False)
print(f"Dataset definitivo guardado en {ruta_dataset_definitivo}.")
```

Anexo G. Código de generación de modelo de machine learning.

```
# Importaciones necesarias
import pandas as pd
from sklearn.model_selection import train_test_split,
RandomizedSearchCV, cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
from imblearn.over_sampling import RandomOverSampler
import joblib

   # ---------------------- SECCIÓN 1: CARGA Y PREPROCESAMIENTO DE
                    DATOS ----------------------

# Cargar el dataset preprocesado
data =
pd.read_csv('../DataSetsParaPreprocesar/DataUniversalProcesado.csv')

# Mostrar una muestra del dataset
print("Muestra del dataset:")
print(data.head(2))

# Verificar la distribución inicial de la variable objetivo
print("Distribución inicial de la variable objetivo (Label):")
print(data['Label'].value_counts())

# Separación de características (X) y la variable objetivo (y)
X = data.drop('Label', axis=1)
y = data['Label']

# División del dataset en conjunto de entrenamiento (70%) y prueba
(30%)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

# Mostrar la distribución de clases en los conjuntos de
entrenamiento y prueba
print("Distribución de clases en el conjunto de entrenamiento:")
print(y_train.value_counts())
print("Distribución de clases en el conjunto de prueba:")
print(y_test.value_counts())

# ---------------------- SECCIÓN 2: BALANCEO DE DATOS --------------
                              --------

# Balancear el conjunto de entrenamiento utilizando
RandomOverSampler
```

```
ros = RandomOverSampler(random_state=42)
X_train_balanced, y_train_balanced = ros.fit_resample(X_train,
y_train)

# ---------------------- SECCIÓN 3: ENTRENAMIENTO DEL MODELO BASE --
                        --------------------

# Crear y entrenar un modelo Random Forest inicial
model = RandomForestClassifier(n_estimators=100, max_depth=15,
random_state=42)
model.fit(X_train_balanced, y_train_balanced)

# Evaluar el modelo inicial
y_pred = model.predict(X_test)
print("Informe de clasificación del modelo base:\n",
classification_report(y_test, y_pred))
print("Matriz de confusión del modelo base:\n",
confusion_matrix(y_test, y_pred))

 # ---------------------- SECCIÓN 4: OPTIMIZACIÓN DE HIPERPARÁMETROS
                       ----------------------

# Definir los parámetros de búsqueda
param_grid = {
    'n_estimators': [100, 150],
    'max_depth': [10, None],
    'min_samples_split': [2, 5]
}

# Realizar una búsqueda aleatoria de hiperparámetros
random_search = RandomizedSearchCV(
    RandomForestClassifier(random_state=42),
    param_distributions=param_grid,
    n_iter=8,
    cv=3,
    scoring='f1_weighted',
    random_state=42
)
random_search.fit(X_train_balanced, y_train_balanced)

# Obtener el mejor modelo y parámetros
best_model = random_search.best_estimator_
print("Mejores hiperparámetros encontrados:",
random_search.best_params_)

 # ---------------------- SECCIÓN 5: ENTRENAMIENTO Y EVALUACIÓN DEL
              MODELO OPTIMIZADO ----------------------

# Entrenar el modelo optimizado con los mejores hiperparámetros
best_model.fit(X_train_balanced, y_train_balanced)
```

```
# Evaluar el modelo optimizado
y_pred = best_model.predict(X_test)
print("Informe de clasificación del modelo optimizado:\n",
classification_report(y_test, y_pred))
print("Matriz de confusión del modelo optimizado:\n",
confusion_matrix(y_test, y_pred))

# ---------------------- SECCIÓN 6: AJUSTE DEL UMBRAL --------------
                              --------

# Obtener las probabilidades de predicción
y_pred_proba = best_model.predict_proba(X_test)

# Ajustar el umbral de decisión
threshold = 0.4
y_pred_adjusted = (y_pred_proba[:, 1] >= threshold).astype(int)

# Evaluar el modelo con el nuevo umbral
print("Informe de clasificación con umbral ajustado:\n",
classification_report(y_test, y_pred_adjusted))
print("Matriz de confusión con umbral ajustado:\n",
confusion_matrix(y_test, y_pred_adjusted))

# ---------------------- SECCIÓN 7: VALIDACIÓN CRUZADA -------------
                              ---------

# Validación cruzada para evaluar la consistencia del modelo
optimizado
cv_scores = cross_val_score(best_model, X_train_balanced,
y_train_balanced, cv=5, scoring='f1_weighted')
print("F1-Score promedio en validación cruzada:", cv_scores.mean())

    # ---------------------- SECCIÓN 8: EXPORTACIÓN Y ANÁLISIS DE
                  RESULTADOS ----------------------

# Guardar el modelo optimizado
joblib.dump(best_model, 'modelo_final_optimizado.pkl')
print("Modelo optimizado guardado exitosamente.")
```

Anexo H. Código de ejecución de modelo de machine learning para la detección automática.

```
# Importaciones necesarias
import os
import time
import json
import pandas as pd
import joblib
import logging
import csv
from datetime import datetime
from watchdog.observers import Observer
from watchdog.events import FileSystemEventHandler

# ---------------------- SECCIÓN 1: CONFIGURACIÓN INICIAL ----------
                            ------------

# Configuración de logging para registrar eventos
logging.basicConfig(
    filename='detector.log',
    level=logging.WARNING,
    format='%(asctime)s - %(levelname)s - %(message)s'
)

# Archivo CSV para guardar los detalles de ataques detectados
csv_log_file = 'detalles_ataques.csv'

# Crear el archivo CSV si no existe, con encabezados predefinidos
if not os.path.exists(csv_log_file):
    with open(csv_log_file, 'w', newline='') as file:
        writer = csv.DictWriter(file, fieldnames=["timestamp",
"src_ip", "dest_ip", "src_port", "dest_port", "proto", "detalles"])
        writer.writeheader()

# Ruta al modelo entrenado
modelo_ruta =
r'e:\ruta\modelo\machinelearning\modelo_final_optimizado.pkl'

# Cargar el modelo de machine learning
try:
    model = joblib.load(modelo_ruta)
    print("✅ Modelo cargado exitosamente.")
except FileNotFoundError:
    print(f"❌ Error: El modelo {modelo_ruta} no se encontró.")
    logging.error(f"Error: El modelo {modelo_ruta} no se encontró.")
```

```
    exit()

# Ruta al archivo eve.json generado por Suricata
filepath = '\\\\ipservidorids\\SuricataLogs\\eve_copy.json'

# ---------------------- SECCIÓN 2: FUNCIONES DE MONITOREO ---------
                            -------------

def verificar_actividad_archivo(filepath, intervalo=60):
    """
    Verifica si el archivo especificado ha sido modificado dentro de
un intervalo de tiempo dado.
    :param filepath: Ruta al archivo a verificar.
    :param intervalo: Intervalo de tiempo en segundos para verificar
la actividad.
    :return: True si el archivo ha sido modificado recientemente,
False de lo contrario.
    """
    try:
        ultima_modificacion = os.path.getmtime(filepath)
        ahora = time.time()
        if (ahora - ultima_modificacion) > intervalo:
            mensaje = f"⚠️ El archivo {filepath} no ha sido
modificado en los últimos {intervalo} segundos."
            print(mensaje)
            logging.warning(mensaje)
            return False
        return True
    except FileNotFoundError:
        mensaje = f"❌ Error: No se puede acceder al archivo
{filepath}."
        print(mensaje)
        logging.error(mensaje)
        return False

def preprocess_data(json_data):
    """
    Procesa un registro JSON de Suricata para convertirlo en un
formato adecuado para el modelo de machine learning.
    :param json_data: Registro JSON de Suricata.
    :return: DataFrame con los datos procesados o None si no es
válido.
    """
    flow = json_data.get('flow', {})
    if not flow:
        return None  # Ignorar registros sin datos de flujo

    # Calcular Flow Duration
    try:
        start_time = datetime.fromisoformat(flow.get('start', '')[:-
6])
```

```
        end_time = datetime.fromisoformat(flow.get('end', '')[:-6])
        flow_duration = (end_time - start_time).total_seconds() *
1000
    except Exception:
        flow_duration = 0

    # Crear diccionario con las columnas seleccionadas
    data = {
        'Flow Duration': flow_duration,
        'Total Fwd Packets': flow.get('pkts_toserver', 0),
        'Total Backward Packets': flow.get('pkts_toclient', 0),
        'Total Length of Fwd Packets': flow.get('bytes_toserver',
0),
        'Total Length of Bwd Packets': flow.get('bytes_toclient',
0),
        'Flow Bytes/s': 0,
        'Flow Packets/s': 0,
        'SYN Flag Count': 0,
        'ACK Flag Count': 0,
        'RST Flag Count': 0,
        'Active Mean': 0,
        'Active Std': 0,
        'Idle Mean': 0,
        'Idle Std': 0,
        'Min Packet Length': 0,
        'Max Packet Length': 0,
        'Packet Length Mean': 0
    }
    return pd.DataFrame([data])

def guardar_detalles_ataque(json_data):
    """
    Guarda los detalles del ataque detectado en un archivo CSV.
    :param json_data: Registro JSON con los datos del ataque.
    """
    try:
        with open(csv_log_file, 'a', newline='') as file:
            writer = csv.DictWriter(file, fieldnames=["timestamp",
"src_ip", "dest_ip", "src_port", "dest_port", "proto", "detalles"])
            writer.writerow({
                "timestamp": json_data.get("timestamp", "N/A"),
                "src_ip": json_data.get("src_ip", "N/A"),
                "dest_ip": json_data.get("dest_ip", "N/A"),
                "src_port": json_data.get("src_port", "N/A"),
                "dest_port": json_data.get("dest_port", "N/A"),
                "proto": json_data.get("proto", "N/A"),
                "detalles": json.dumps(json_data)  # Convertir todo
el JSON a cadena
            })
    except Exception as e:
        logging.error(f"Error al guardar detalles del ataque: {e}")
```

```
def process_line(line):
    """
    Procesa cada línea del archivo eve.json y realiza predicciones
utilizando el modelo.
    :param line: Línea del archivo JSON.
    """
    try:
        json_data = json.loads(line)
        if json_data.get('event_type') == 'flow':
            input_data = preprocess_data(json_data)
            if input_data is not None:
                prediction = model.predict(input_data)
                if prediction[0] == 1:
                    print("⚠️ Alerta: Posible ataque detectado.")
                    logging.warning("Alerta: Posible ataque
detectado.")
                    guardar_detalles_ataque(json_data)
                else:
                    print("✅ Tráfico benigno detectado.")
    except json.JSONDecodeError:
        print("❌ Línea con error JSON ignorada.")
        logging.error("Línea con error JSON ignorada.")
    except Exception as e:
        print(f"Error al procesar la línea: {e}")
        logging.error(f"Error al procesar la línea: {e}")

# ---------------------- SECCIÓN 3: OBSERVADOR DE CAMBIOS ----------
                            ------------

class FileChangeHandler(FileSystemEventHandler):
    """
    Clase para manejar eventos de cambio en el archivo monitoreado.
    """
    def __init__(self, filepath):
        self.filepath = filepath
        self.last_position = 0

    def on_modified(self, event):
        if event.src_path == self.filepath:
            self.process_new_lines()

    def process_new_lines(self):
        """
        Procesa las nuevas líneas añadidas al archivo monitoreado.
        """
        try:
            with open(self.filepath, 'r') as file:
                if self.last_position >
os.path.getsize(self.filepath):
                    self.last_position = 0
```

```
                file.seek(self.last_position)
                new_lines = file.readlines()
                self.last_position = file.tell()
                for line in new_lines:
                    process_line(line)
        except Exception as e:
            logging.error(f"Error al procesar nuevas líneas: {e}")

# ---------------------- SECCIÓN 4: INICIO DEL MONITOREO -----------
                            -----------

def esperar_archivo_disponible(filepath):
    """
    Espera hasta que el archivo especificado esté disponible.
    :param filepath: Ruta al archivo.
    """
    while not os.path.exists(filepath):
        print(f"🔄 Esperando que el archivo {filepath} esté
disponible...")
        time.sleep(10)

esperar_archivo_disponible(filepath)

event_handler = FileChangeHandler(filepath)
observer = Observer()
observer.schedule(event_handler, path=os.path.dirname(filepath),
recursive=False)
observer.start()
print(f"✅ Monitoreo iniciado en el archivo: {filepath}")

try:
    while True:
        if not verificar_actividad_archivo(filepath, intervalo=60):
            print("🔄 Relectura forzada del archivo por
inactividad.")
        time.sleep(10)
except KeyboardInterrupt:
    print("🚨 Monitoreo detenido.")
    observer.stop()
observer.join()
```

Anexo I. Manual de usuario de sistema IDS.

UNIVERSIDAD TÉCNICA DE AMBATO

MANUAL DE USUARIO

Sistema IDS con Machine learning Unidad Educativa "19 de Septiembre"



FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

Introducción

El sistema de detección de intrusos (IDS) basado en machine learning, implementado en la Unidad Educativa “19 de Septiembre”, tiene como objetivo principal proteger la seguridad de la red institucional de manera integral. Este sistema está diseñado para ayudar a identificar posibles ataques o accesos no autorizados, garantizando un entorno digital seguro para toda la comunidad educativa. La solución combina el software Suricata con un modelo de machine learning entrenado con Random Forest, lo que permite detectar tráfico malicioso.

Entre las principales características que ofrece este sistema, destacan:

Mayor seguridad en la red: El IDS analiza continuamente el tráfico de red en busca de patrones anómalos o intentos de intrusión, contribuyendo a prevenir ataques informáticos que puedan poner en riesgo la información o los recursos tecnológicos de la institución.

Automatización y monitoreo continuo: La integración con herramientas de monitoreo, como Grafana, permite visualizar y supervisar la actividad de la red en tiempo real. Además, el sistema genera alertas inmediatas para que el personal encargado pueda actuar rápidamente ante cualquier incidencia.

Optimización de recursos: Gracias al uso de machine learning, el sistema logra un análisis eficaz sin sobrecargar los recursos del servidor. Esto garantiza un equilibrio entre el rendimiento y la precisión, maximizando su efectividad.

Facilidad de uso y personalización: Diseñado pensando en el personal técnico, el sistema es intuitivo y adaptable. Su configuración es sencilla, lo que facilita la integración en el entorno de la Unidad Educativa y asegura que pueda ajustarse a necesidades específicas.

Instrucciones de uso

Grafana

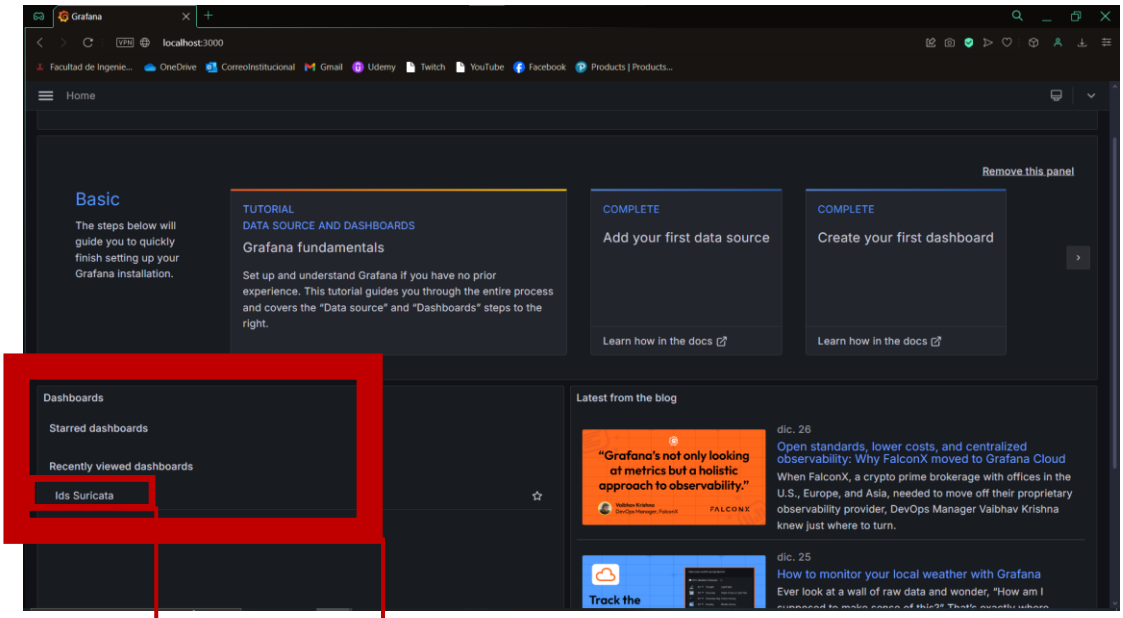
Para el sistema IDS, se seleccionó Grafana como el software de monitoreo en tiempo real debido a su capacidad para ofrecer visualizaciones claras y personalizables. Para acceder a Grafana, el usuario debe abrir su navegador preferido e ingresar la dirección localhost:3000/login, que corresponde a la dirección local donde Grafana opera de manera predeterminada.



En la pantalla de inicio, introduzca las credenciales proporcionadas para acceder al panel de visualización en Grafana y comenzar a monitorear la actividad de la red en tiempo real.



En la interfaz principal de Grafana, podrá explorar una variedad de opciones y herramientas diseñadas para facilitar la personalización y gestión del monitoreo de la red. Para acceder a los dashboards previamente configurados, diríjase a la esquina inferior izquierda de la pantalla, donde encontrará una lista de los dashboards disponibles. Desde ahí, podrá seleccionar el dashboard de su interés para visualizar los datos relevantes de la red y supervisar la actividad en tiempo real.



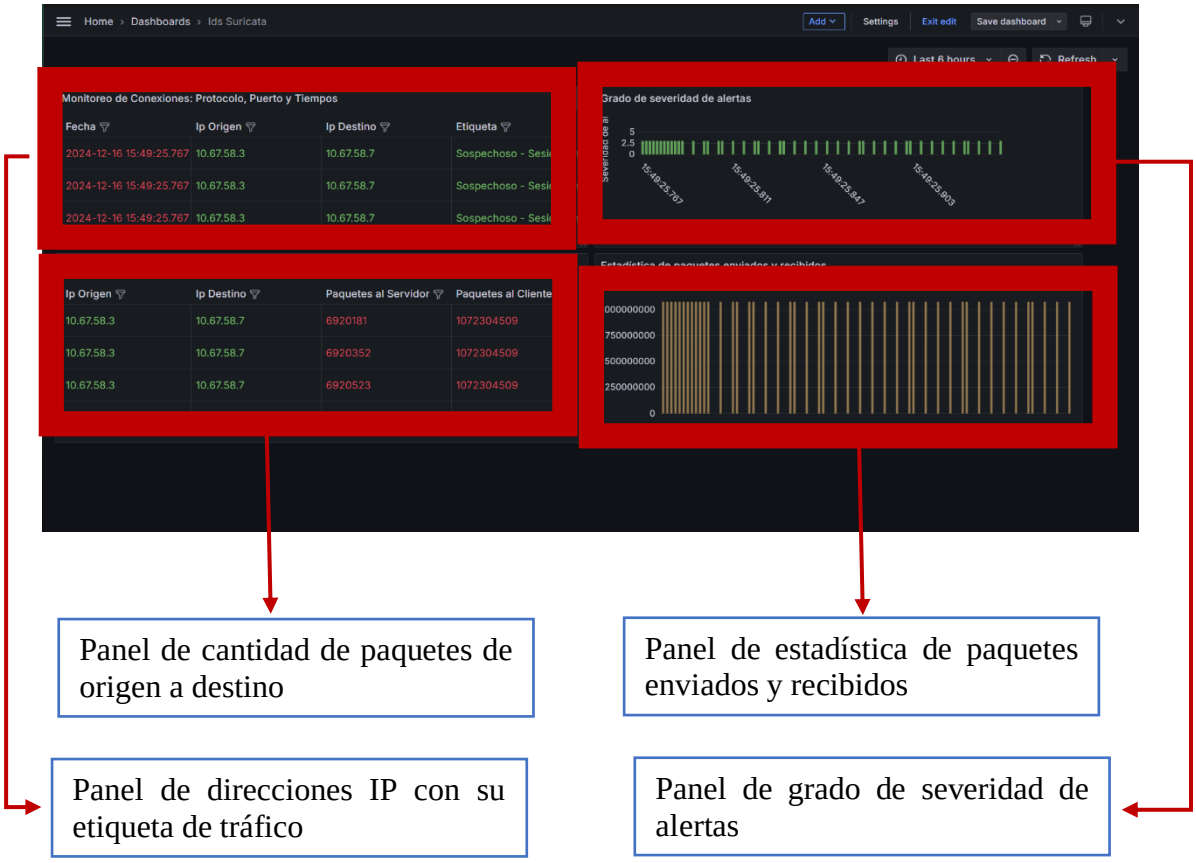
Zona de dashboards disponibles

Dashboard establecido para la monitorización del sistema IDS con machine learning

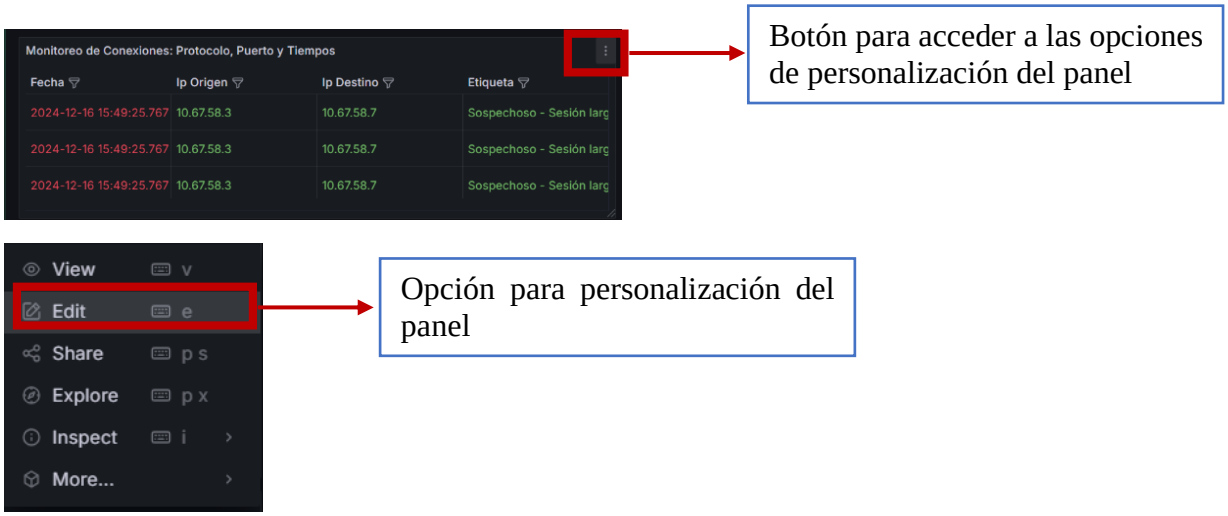
En el dashboard preestablecido para el sistema IDS, se presenta una visión general de la actividad de la red mediante diversos paneles interactivos y visualizaciones en tiempo real. Entre los datos destacados, se incluyen el monitoreo de conexiones con información detallada como fecha, IP de origen y destino, junto con etiquetas que describen el tipo de tráfico detectado. Además, se dispone de un panel que muestra el

grado de severidad de las alertas generadas, permitiendo al personal encargado identificar rápidamente los eventos más críticos.

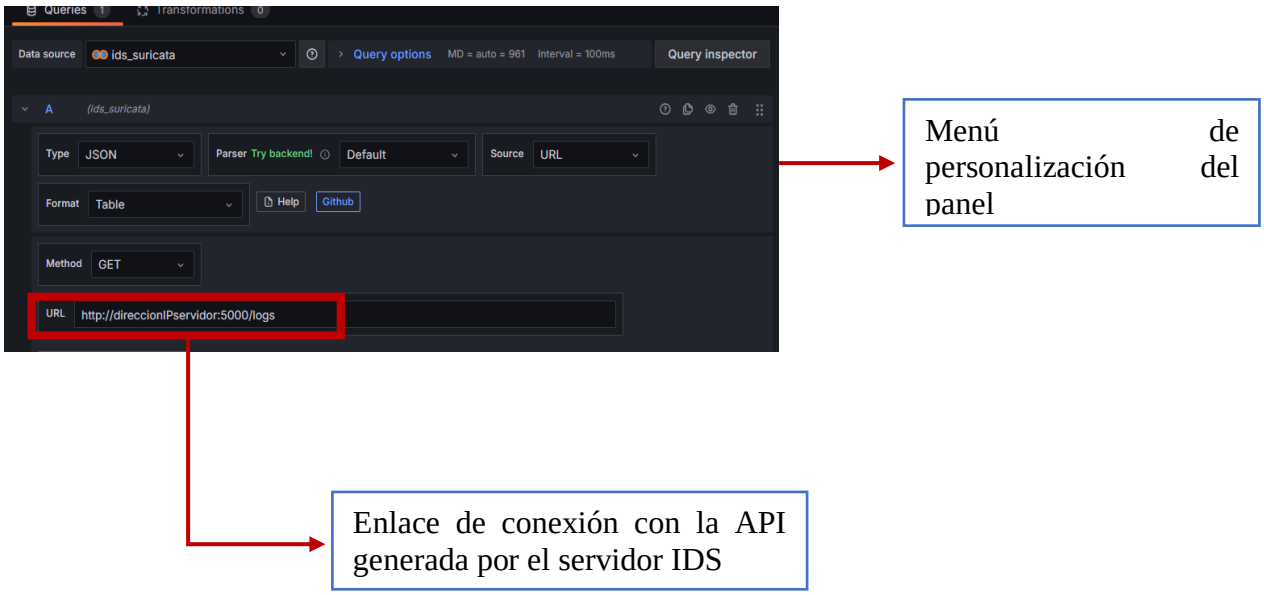
Otro panel relevante es el monitoreo de paquetes, donde se puede observar la cantidad de paquetes enviados de origen y al cliente, lo que proporciona una visión más granular del flujo de datos en la red. Por último, se incluye una gráfica que muestra la estadística de paquetes enviados y recibidos, facilitando el análisis del comportamiento general de la red.



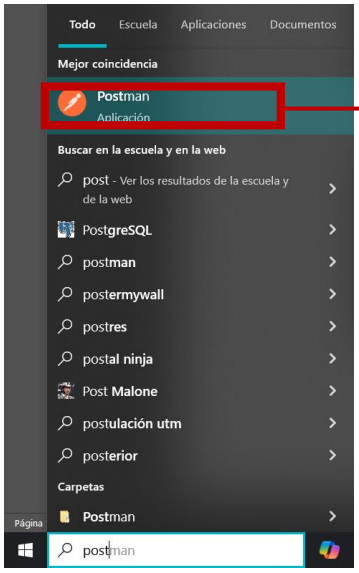
Los paneles mostrados representan solo una parte de las posibilidades que ofrece el sistema, ya que el usuario tiene la flexibilidad de crear y personalizar nuevos paneles. Esta funcionalidad permite adaptar el monitoreo a los requerimientos particulares de la red, añadiendo métricas, gráficos o alertas adicionales que faciliten un análisis más detallado y orientado a los objetivos del usuario.



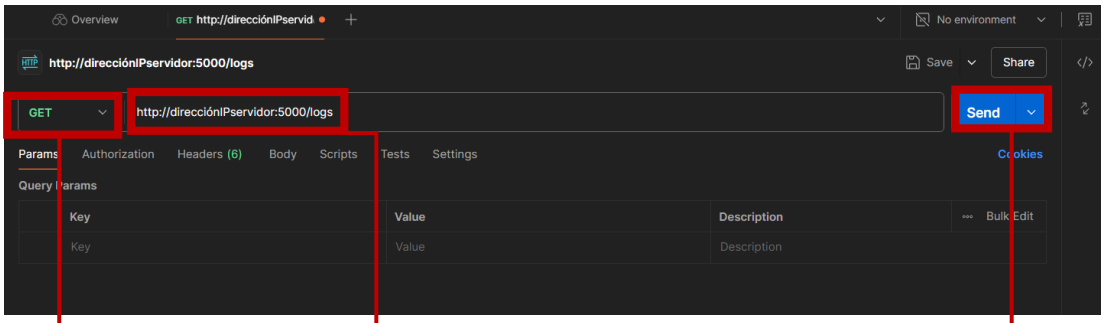
Cada panel se puede personalizar según las necesidades o la información que el usuario desee visualizar, accediendo al menú de configuración correspondiente.



El servidor IDS genera una API accesible a través de la dirección IP del servidor en el puerto 5000/logs, lo que permite obtener los datos de monitoreo y reflejarlos en los paneles. Para comprender mejor la información disponible a través de la API, se puede utilizar la acción GET mediante Postman, ingresando el enlace http://direccionIPservidor:5000/logs.



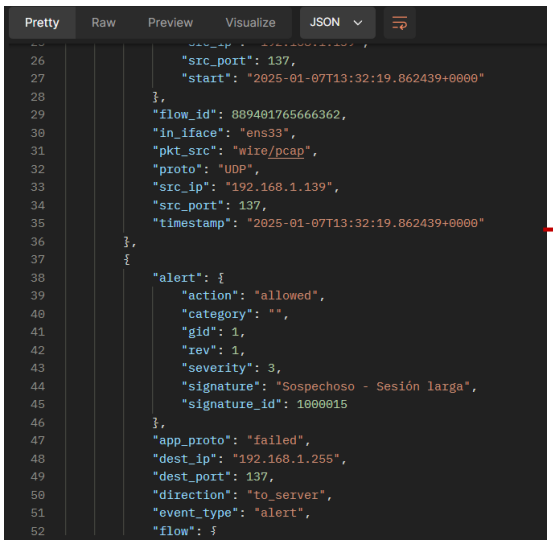
Aplicación Postman para consultar información de API



Acción GET para consultar con la API

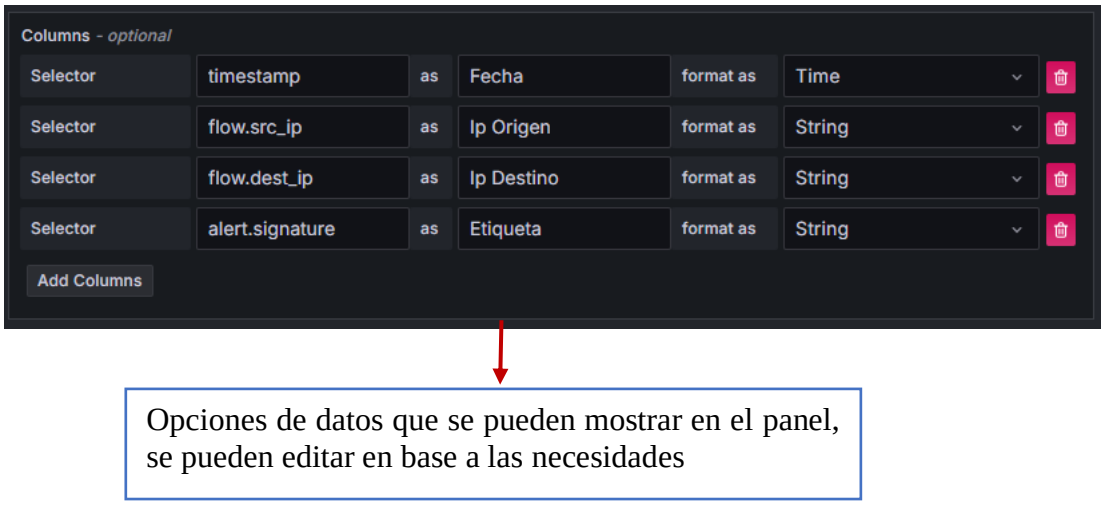
Enlace de conexión con la API

Botón para ejecutar la consulta con la API



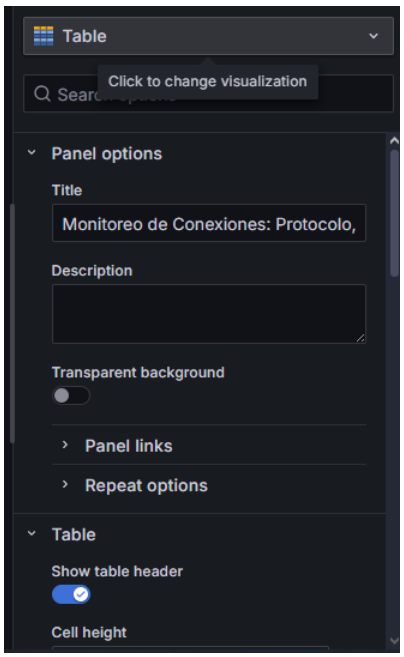
Información que se puede extraer de la API

En el menú de personalización del panel, es posible ajustar la información que se desea mostrar seleccionando las columnas específicas provenientes de la consulta a la API. Se puede configurar cada columna indicando su selector, nombre personalizado y formato de datos (por ejemplo, tiempo, texto, o direcciones IP). Esta flexibilidad permite al usuario adaptar la visualización según las necesidades del monitoreo.



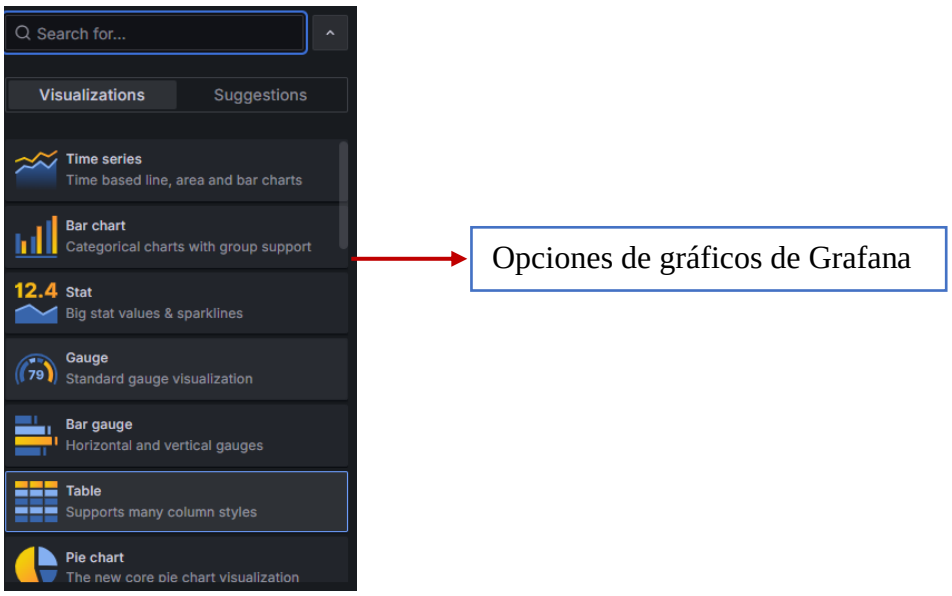
Opciones de datos que se pueden mostrar en el panel, se pueden editar en base a las necesidades

Grafana también ofrece personalización visual sobre los gráficos estadísticos que se desee mostrar en los paneles, permitiendo selección de color, tipo y tamaño de letra, etc, accediendo mediante el menú que se encuentra a lado derecho de la pantalla de edición del panel.



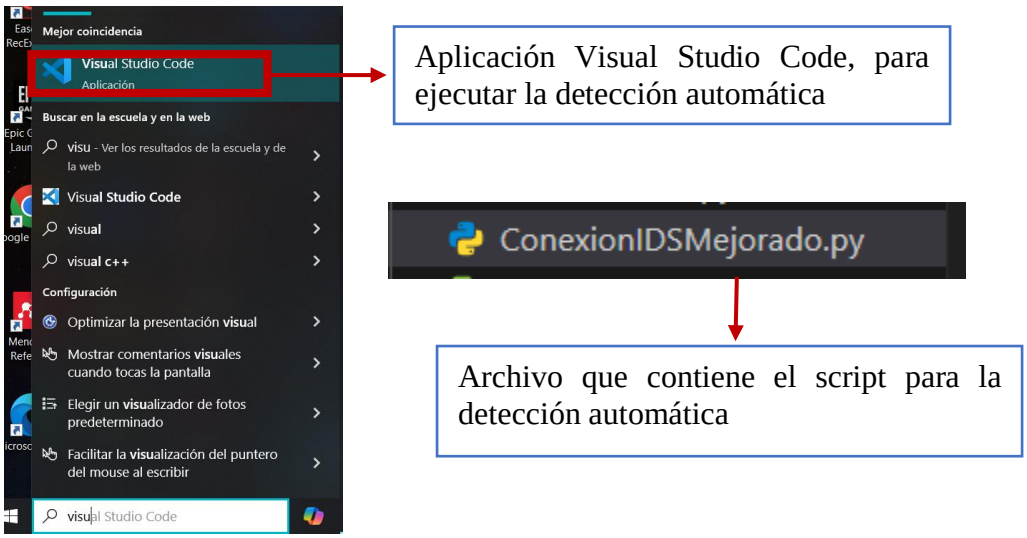
Menú de personalización de los gráficos de Grafana

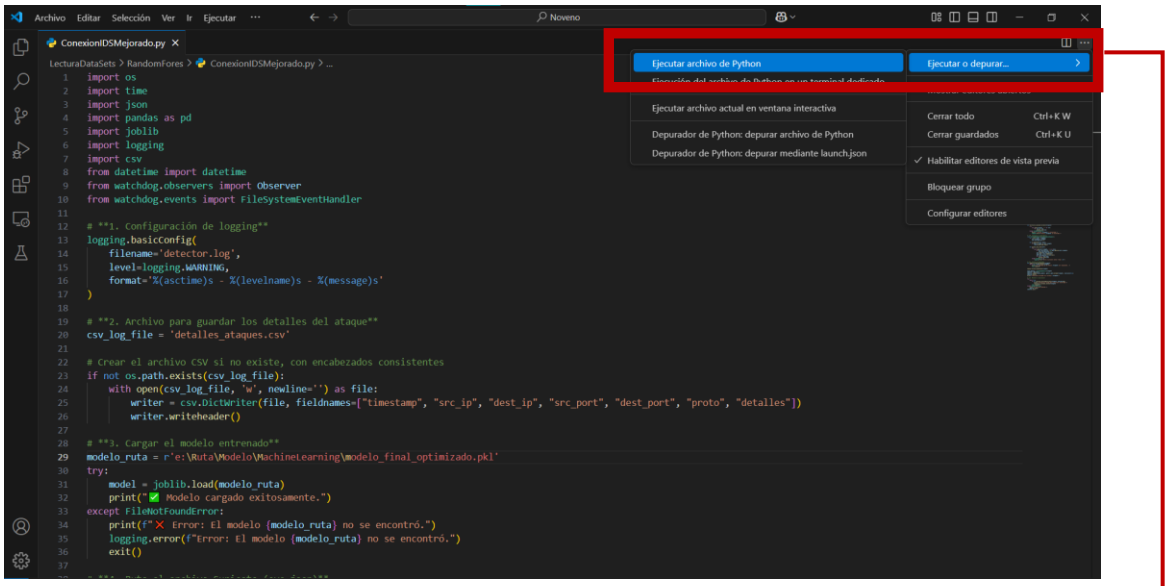
Grafana también ofrece una amplia gama de gráficos que se pueden seleccionar para la presentación de la información.



Modelo de Machine Learning

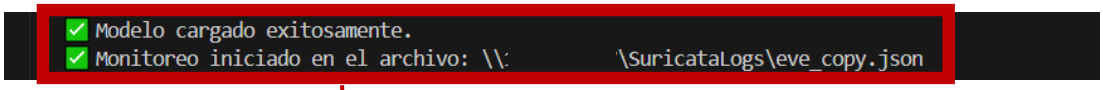
Antes de iniciar el escaneo de la red con el modelo de machine learning, es esencial asegurarse de que el servidor IDS esté completamente operativo. Una vez confirmado que todo funciona correctamente, acceda a la aplicación Visual Studio Code, instalada previamente en el equipo designado para ejecutar el modelo. Desde esta herramienta, navegue hasta el directorio donde se encuentra alojado el algoritmo y proceda a ejecutarlo para comenzar la detección.





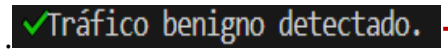
Para ejecutar el modelo, seleccionar los tres puntos y luego dar clic en la opción ejecutar archivo de Python

Para verificar que el modelo se ha ejecutado correctamente, el sistema genera un mensaje de confirmación que indica el éxito del proceso.

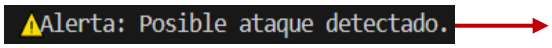


Mensajes que indican que el modelo se cargó correctamente y que se puedo encontrar el archivo compartido eve.json con el servidor IDS

Además de los mensajes que confirman la ejecución exitosa, el sistema también dispone de otras alertas diseñadas para informar sobre diferentes eventos o situaciones relevantes.



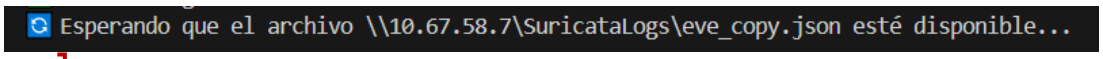
La alerta “Tráfico benigno” indica que en la red el flujo de paquetes es normal y no existe ninguna anomalía



La notificación “Posible ataque detectado”, indica que existe un comportamiento extraño en la red, en base a lo aprendido por el modelo, cuando se visualiza este tipo de alerta, sus datos se guardan en los registros log correspondientes para poder analizarla más a detalle.



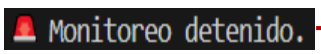
El aviso de “Línea con error JSON ignorada”, se refiere a que el archivo eve.json generado por el servidor IDS no se encuentra con el formato adecuado, el sistema ignora esta línea y pasa a la siguiente para seguir con la detección.



El mensaje “Esperando archivo” indica que cuando el archivo eve.json no está disponible, el modelo no se detiene y constantemente está intentando acceder a este para poder seguir con la detección.

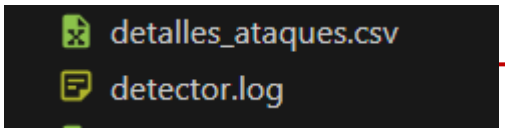


El mensaje “El archivo eve.json no ha sido modificado en los últimos 60 segundos” se genera cuando el archivo no registra actualizaciones durante un minuto. En respuesta, el sistema emite la alerta “Relectura forzada”, lo que indica que, ante la falta de cambios, el archivo será analizado nuevamente de manera forzada hasta identificar líneas actualizadas.



La alerta “Monitoreo detenido” se activa cuando el análisis realizado por el modelo se interrumpe de manera intencional. Esta notificación está diseñada para registrarse específicamente cuando la ejecución del sistema se detiene manualmente mediante la combinación de teclas "Control + C".

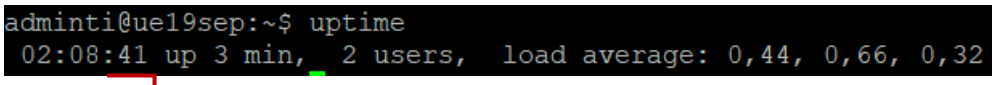
En caso de detectar un posible ataque, el sistema registra la información en dos archivos distintos, cada uno con un propósito específico. El archivo detector.log almacena datos básicos, proporcionando un resumen rápido de los eventos detectados. Por otro lado, el archivo detalles_ataques.csv guarda información más detallada y estructurada sobre los ataques, incluyendo datos clave como direcciones IP, puertos, protocolos y descripciones completas.



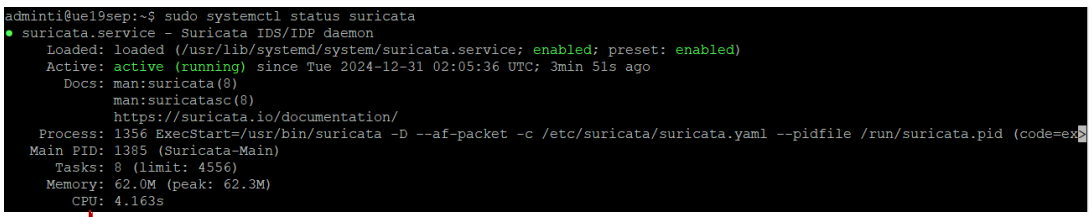
Son los archivos logs que almacenan información sobre los posibles ataques detectados, se generan automáticamente en el mismo directorio donde se encuentra alojado el archivo del modelo de machine learning.

Servidor IDS

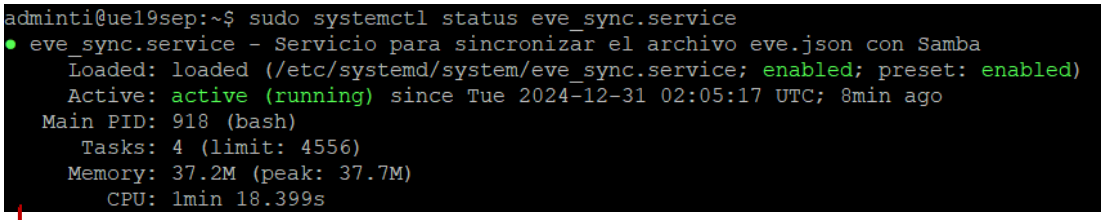
El servidor IDS está configurado específicamente para realizar tareas esenciales relacionadas con el monitoreo del estado de los servicios y, en caso necesario, el reinicio manual de estos. Dado que todos los servicios clave para el funcionamiento del sistema están automatizados, se inician automáticamente tan pronto como el servidor entra en operación.



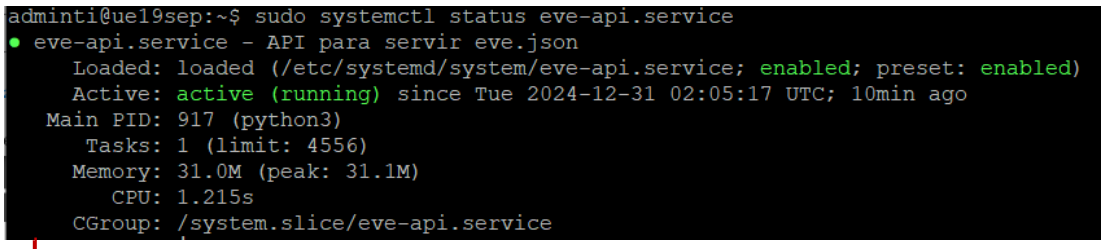
Con el comando uptime se puede verificar el estado del servidor.



Para comprobar el estado de Suricata utilice el comando “sudo systemctl status suricata”. Este comando mostrará el estado actual del servicio. Si Suricata está funcionando correctamente, debería aparecer con un estado de “active (running)” y “enabled”. Si no es así, podría indicar algún tipo de falla que debe investigarse. En ese caso, se recomienda revisar los logs del servicio o documentación de Sutricata para identificar el problema y tomar las medidas necesarias para solucionarlo.



Para comprobar el estado del servicio encargado de sincronizar el archivo eve.json con la carpeta compartida, utilice el comando "sudo systemctl status eve_sync.service". Este comando mostrará información detallada sobre el estado del servicio. Si el servicio está funcionando correctamente, debería aparecer con un estado de "active (running)". En caso contrario, revise los logs del servicio para identificar posibles errores y reinícielo si es necesario para garantizar la sincronización continua del archivo.



Para verificar el estado del servicio encargado de sincronizar y mantener actualizada la API, utilice el comando "sudo systemctl status eve-api.service". Este comando mostrará el estado actual del servicio. Si el servicio está funcionando correctamente, debería aparecer con un estado de "active (running)". Si el estado no es el esperado, revise los logs del servicio para diagnosticar posibles problemas y proceda a reiniciarlo para restaurar su funcionalidad.

Otros comandos útiles para gestionar el estado de los servicios son los siguientes:

Comandos para Reiniciar Servicios

Comando	Descripción
`sudo systemctl restart suricata`	Reinicia el servicio de Suricata. Útil para aplicar cambios en su configuración o reactivar el monitoreo si falla.
`sudo systemctl restart eve_sync.service`	Reinicia el servicio que sincroniza el archivo eve.json con la carpeta compartida, asegurando la integridad de los datos.
`sudo systemctl restart eve-api.service`	Reinicia el servicio de la API, manteniendo la comunicación entre la API y Grafana para la visualización en tiempo real.

Comandos para Detener Servicios

Comando	Descripción
`sudo systemctl stop suricata`	Detiene el servicio de Suricata, interrumpiendo temporalmente la monitorización del tráfico de red.
`sudo systemctl stop eve_sync.service`	Detiene el servicio encargado de sincronizar el archivo eve.json con la carpeta compartida.
`sudo systemctl stop eve-api.service`	Interrumpe el servicio de la API, deteniendo la comunicación entre esta y Grafana.

Comandos para Iniciar Servicios

Comando	Descripción
`sudo systemctl start suricata`	Activa el servicio de Suricata, permitiendo la monitorización del tráfico de red en tiempo real.
`sudo systemctl start eve_sync.service`	Inicia el servicio que sincroniza el archivo eve.json con la carpeta compartida.
`sudo systemctl start eve-api.service`	Arranca el servicio que asegura la comunicación entre la API y Grafana para mostrar datos actualizados.

Notas Importantes

- Frecuencia de Uso: Utilice los comandos de reinicio si se realizan cambios en la configuración, o si alguno de los servicios deja de funcionar.

- Monitoreo de Servicios: Antes de usar los comandos de reinicio, puede verificar el estado de los servicios con ***"sudo systemctl status [nombre del servicio]"***

- Precaución al Detener Servicios: Detener servicios como suricata o eve-api interrumpirá temporalmente la recopilación de datos y su visualización en Grafana.

Preguntas Frecuentes

¿Qué hago si no veo actualizaciones en el dashboard de Grafana? Verifique que el servicio API esté ejecutándose correctamente en el servidor. Para ello, acceda al terminal y asegúrese de que el servicio relacionado con la API esté activo. Si no lo está, reinicie el servicio.

¿Cómo puedo reiniciar el sistema IDS en caso de falla? Asegúrese de que todos los servicios relacionados con Suricata, la API y el modelo de machine learning estén detenidos. Luego, reinicie cada componente siguiendo el orden: Suricata, la API y, finalmente, el modelo de machine learning. Puede verificar su funcionamiento accediendo al dashboard de Grafana.

¿Qué debo hacer si no puedo acceder a los logs generados? Compruebe que tiene permisos adecuados para acceder al directorio donde se generan los logs. Si el problema persiste, asegúrese de que el script encargado de generar los logs está funcionando correctamente.

¿Qué hago si recibo una alerta de que el archivo eve.json no ha sido actualizado? Esto podría indicar que Suricata no está generando nuevos registros. Verifique que el servicio de Suricata esté activo y funcionando correctamente.