



**Escola Politècnica Superior  
d'Enginyeria de Vilanova i la Geltrú**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# TRABAJO FINAL DE GRADO

**TÍTULO: ANÁLISIS DE MODELOS DE INTELIGENCIA ARTIFICIAL PARA LA  
DETECCIÓN Y CONTROL DE PATRONES DE CIBERSEGURIDAD**

**AUTOR: MARTÍN AGUILAR, FRANCISCO JOSÉ**

**FECHA DE PRESENTACIÓN: 8 DE JULIO DE 2024**

**APELLIDOS: MARTÍN AGUILAR**

**NOMBRE: FRANCISCO JOSÉ**

**TITULACIÓN: GRADO EN INGENIERÍA INFORMÁTICA**

**PLAN: 2018**

**DIRECTOR: MASIP BRUIN, XAVIER**

**DEPARTAMENTO: DAC-DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES**

**QUALIFICACIÓ DEL TFG**

**TRIBUNAL**

**PRESIDENTE**

**SECRETARIO**

**VOCAL**

**Jordi Garcia Almiñana**

**Eva Marin Tordera**

**Hugo Valls Mancha**

**FECHA DE LECTURA: 8 DE JULIO DE 2024**

Este proyecto tiene en cuenta aspectos medioambientales:  Sí  No

## AGRADECIMIENTOS

Este trabajo no habría sido posible sin el apoyo y la colaboración de muchas personas y organizaciones a las que me gustaría expresar mi más sincero agradecimiento.

En primer lugar, quiero agradecer a mi director de tesis, el Sr. Masip, por su inestimable orientación, paciencia y consejos a lo largo de todo el proceso. Su experiencia y conocimiento en el campo de la ciberseguridad han sido fundamentales para la realización de este proyecto. Su dedicación y apoyo constante me han inspirado a dar lo mejor de mí en cada etapa del trabajo.

Agradezco también a mis profesores y compañeros de la Universidad Politécnica de Catalunya por el ambiente académico enriquecedor y por las discusiones constructivas que me han permitido aprender y crecer tanto personal como profesionalmente. Las clases y laboratorios impartidos por el Departamento de Redes han sido cruciales para adquirir las habilidades y conocimientos necesarios para llevar a cabo este trabajo.

Mi gratitud se extiende a los miembros de mi familia, quienes han sido mi pilar de apoyo incondicional. A mis padres, por su amor y comprensión, y por siempre creer en mí y en mis capacidades. A mis amigos, por su ánimo y por brindarme momentos de distracción y alegría cuando más los necesitaba.

Quiero agradecer especialmente a STP GROUP por facilitarme el acceso a los recursos y herramientas necesarias para la investigación. Su colaboración ha sido esencial para el desarrollo e implementación de las soluciones presentadas en este trabajo.

A todos aquellos que participaron de alguna manera en este proyecto, ya sea ofreciendo su tiempo, conocimientos o apoyo moral, les estoy profundamente agradecido. Sus contribuciones, aunque a veces puedan parecer pequeñas, han sido de gran valor para el logro de este objetivo.

Finalmente, agradezco especialmente a Valentín Palonsky Guitard, un amigo y compañero, por su apoyo técnico en la utilización de diversas plataformas y tecnologías que han sido cruciales para la elaboración de este trabajo. Su disposición para ayudar y resolver mis dudas ha sido invaluable.

A todos ustedes, mi más sincero agradecimiento por estar a mi lado en este camino. Este trabajo es tanto mío como de todos aquellos que me han acompañado y apoyado en esta travesía. ¡Gracias!

## RESUMEN

El mundo digital se ha convertido en un campo de batalla en constante evolución, que, junto a los ataques cibernéticos, cada vez más sofisticados y multifacéticos, acechan a empresas, gobiernos e individuos por igual. Las amenazas se multiplican exponencialmente, creando un panorama de ciberseguridad en constante cambio y desafío. A medida que la digitalización avanza y la generación de datos aumenta exponencialmente en nuestra vida cotidiana, la seguridad de la información emerge como una preocupación crítica. La ineludible necesidad de proteger los activos digitales y robustecer la infraestructura frente a una constante evolución de las amenazas cibernéticas se posiciona como una prioridad irrefutable. En este contexto, el presente Trabajo Final de Grado (TFG) se propone abordar estos desafíos mediante una investigación exhaustiva, un diseño meticuloso y una implementación rigurosa de una solución integral de ciberseguridad enfocada en el ámbito de las redes y de la inteligencia artificial.

Se plantea la creación de un Sistema de Detección de Intrusiones de Red con Inteligencia Artificial (NIDS-AI - Network Intrusion Detection System with Artificial Intelligence), diseñado con el objetivo de fortalecer la seguridad tanto en entornos empresariales como personales. Este sistema se sustenta en modelos de aprendizaje automático, siguiendo un proceso que abarca desde la recolección, análisis y preprocesamiento de datos hasta el entrenamiento, selección, validación y evaluación de modelos.

La implementación de este enfoque implica el uso de diversas técnicas y algoritmos, empleando datos no supervisados proveniente de un entorno local en tiempo real como datos supervisados proveniente de fuentes abiertas, como Kaggle o la Universidad de Nueva Gales del Sur, que incluyen diversas clases de ataques y datos benignos. Los algoritmos con las mejores precisiones en la detección de ataques serán seleccionados para integrarse en el NIDS-AI. Para la representación visual de los datos obtenidos, se desarrollará un panel de control que permitirá la visualización de las diferentes categorías para los distintos flujos de tráfico.

En cuanto a la infraestructura, se optará por la implementación del NIDS-AI a través de servicios en la nube como AWS o Hetzner, o en local, con el propósito de simular un entorno real donde se capture tráfico para su entrenamiento y análisis posteriores.

### Paraules clau (màxim 10):

Inteligencia Artificial	Ciberseguridad	Empresa	NIDS-AI
Análisis de Datos	Aprendizaje automático	AWS	Hetzner
Panel de Visualizacion	Seguridad de Redes		

## RESUM

El món digital s'ha convertit en un camp de batalla en constant evolució, que, juntament amb els atacs cibernetcs, cada cop més sofisticats i multifacetics, aguaiten empreses, governs i individus per igual. Les amenaces es multipliquen exponencialment, creant un panorama de ciberseguretat en constant canvi i desafiat. A mesura que la digitalització avança i la generació de dades augmenta exponencialment a la nostra vida quotidiana, la seguretat de la informació emergeix com una preocupació crítica. La necessitat ineludible de protegir els actius digitals i enfortir la infraestructura davant d'una constant evolució de les amenaces cibernetiques es posicione com una prioritat irrefutabile. En aquest context, aquest Treball Final de Grau (TFG) es proposa abordar aquests desafiaments mitjançant una investigació exhaustiva, un disseny meticolós i una implementació rigorosa d'una solució integral de ciberseguretat enfocada a l'àmbit de les xarxes i de la intel·ligència artificial.

Es planteja crear un Sistema de Detecció d'Intrusions de Xarxa amb Intel·ligència Artificial (NIDS-AI - Network Intrusion Detection System with Artificial Intelligence), dissenyat amb l'objectiu d'enfortir la seguretat tant en entorns empresarials com personals. Aquest sistema se sustenta en models d'aprenentatge automàtic, seguit un procés que abasta des de la recol·lecció, anàlisi i preprocessament de dades fins a l'entrenament, selecció, validació i evaluació de models.

La implementació d'aquest enfocament implica l'ús de diverses tècniques i algorismes, emprant dades no supervisades provinents d'un entorn local en temps real com dades supervisades provinents de fonts obertes, com Kaggle o la Universitat de Nova Gal·les del Sud, que inclouen diverses classes de atacs i dades benignes. Els algoritmes amb les millors precisions en la detecció d'atacs seran seleccionats per integrar-se al NIDS-AI. Per a la representació visual de les dades obtingudes, es desenvoluparà un tauler de control que permetrà la visualització de les diferents categories per als diferents fluxos de trànsit.

Pel que fa a la infraestructura, s'optarà per la implementació del NIDS-AI a través de serveis al núvol com AWS o Hetzner, o en local, amb el propòsit de simular un entorn real on es capturi trànsit per al seu entrenament i anàlisis posteriors.

Anàlisi de Dades

### Paraules clau (màxim 10):

Intel·ligència Artificial	Ciberseguretat	Empresa	NIDS-AI
Anàlisi de Dades	Aprendentatge automàtic	AWS	Hetzner
Panell de Visualització	Seguretat de Xarxes		

## ABSTRACT

The digital world has become a constantly evolving battlefield, which, along with increasingly sophisticated and multifaceted cyber attacks, stalk companies, governments and individuals alike. Threats are multiplying exponentially, creating an ever-changing and challenging cybersecurity landscape. As digitalization advances and data generation increases exponentially in our daily lives, information security emerges as a critical concern. The unavoidable need to protect digital assets and strengthen infrastructure in the face of a constant evolution of cyber threats is positioned as an irrefutable priority. In this context, this Final Degree Project (TFG) aims to address these challenges through exhaustive research, meticulous design and rigorous implementation of a comprehensive cybersecurity solution focused on the field of networks and artificial intelligence.

The creation of a Network Intrusion Detection System with Artificial Intelligence (NIDS-AI - Network Intrusion Detection System with Artificial Intelligence) is proposed, designed with the objective of strengthening security in both business and personal environments. This system is based on machine learning models, following a process that ranges from data collection, analysis and preprocessing to training, selection, validation and evaluation of models.

The implementation of this approach involves the use of various techniques and algorithms, using unsupervised data from a local environment in real time as well as supervised data from open sources, such as Kaggle or the University of New South Wales, which include various kinds of attacks and benign data. The algorithms with the best accuracies in detecting attacks will be selected to be integrated into the NIDS-AI. For the visual representation of the data obtained, a control panel will be developed that will allow the visualization of the different categories for the different traffic flows.

Regarding infrastructure, the implementation of NIDS-AI will be chosen through cloud services such as AWS or Hetzner, or locally, with the purpose of simulating a real environment where traffic is captured for subsequent training and analysis.

**Keywords (10 maximum):**

Artificial intelligence	Cybersecurity	Company	CIDS-AI
Data Analysis	Machine learning	AQS	Hetzner
Dashboard	Network Security		

## ÍNDICE

<b>1. INTRODUCCIÓN</b>	<b>8</b>
1.1 MOTIVACIÓN	8
1.2 OBJETIVO	9
1.3 ALCANCE	9
<b>2. METODOLOGÍA</b>	<b>11</b>
2.1 AGILE Y SCRUM	11
2.1.1 ROLES	11
2.1.2 EVENTOS	11
2.1.3 ARTEFACTOS	11
2.1.4 SCRUM EN ACCION	12
2.1.5 BENEFICIOS	13
2.2 SISTEMA DE CONTROL DE VERSIONES	13
2.2.1 ENTORNOS DE DESARROLLO	13
2.3 PRUEBAS Y CALIDAD DE SOFTWARE	14
<b>3. CIBERSEGURIDAD</b>	<b>15</b>
<b>4. DESCRIPCIÓN DE LAS TAREAS</b>	<b>17</b>
4.1 INTRODUCCIÓN	17
4.2 TAREAS	17
4.2.1 TAREAS DE GESTIÓN Y DOCUMENTACIÓN	17
4.2.2 TAREAS DE DESARROLLO	18
4.3 HERRAMIENTAS Y RECURSOS	19
<b>5. ESTIMACIÓN Y GANTT</b>	<b>21</b>
5.1 DIAGRAMA DE GANTT	22
<b>6. GESTIÓN DE RIESGOS</b>	<b>23</b>
6.1 FALTA DE EXPERIENCIA EN LAS TECNOLOGÍAS	23
6.2 ERRORES Y FALLOS DE EJECUCIÓN	23
6.3 MODIFICACIÓN DEL PLANTEAMIENTO	23
6.4 DESVIACIÓN DE LA PLANIFICACIÓN	24
6.5 FECHA DE ENTREGA	24
6.6 DEPENDENCIAS DE SERVICIOS Y LIBRERÍAS DE TERCEROS	24

<b>7. PRESUPUESTO</b>	<b>25</b>
7.1 IDENTIFICACIÓN Y ESTIMACIÓN DE COSTES	25
7.1.1 EQUIPAMIENTO HARDWARE	26
7.1.2 EQUIPAMIENTO SOFTWARE	27
7.1.3 COSTE GENERAL	28
7.2 CONTROL DE GESTION	30
<b>8. MARCO TEÓRICO</b>	<b>31</b>
8.1 SISTEMA DE DETECCIÓN DE INTRUSIONES	31
8.1.1 COMPONENTES IDS	31
8.1.2 TIPOLOGÍA DE IDS	32
8.1.2.1 IDS BASADO EN RED (NIDS)	32
8.1.2.2 IDS BASADO EN HOST(HIDS)	32
8.1.2.3 CARACTERÍSTICAS HIDS VS NIDS	32
8.1.3 MÉTODOS TRADICIONALES	33
8.1.3.1 DETECCIÓN BASADA EN FIRMAS	33
8.1.3.2 DETECCIÓN BASADA EN ANOMALÍAS	33
8.1.4 APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL	33
8.1.4.1 APRENDIZAJE AUTOMÁTICO (MACHINE LEARNING)	33
8.1.4.2 APRENDIZAJE PROFUNDO (DEEP LEARNING)	33
8.1.4.3 PROCESAMIENTO DEL LENGUAJE NATURAL (NLP)	34
8.2 SISTEMA DE DETECCIÓN DE INTRUSIONES EN RED	34
8.2.1 FASES DEL NIDS	34
8.2.2 CONCLUSIONES DEL NIDS	35
8.2.3 EJEMPLOS DE NIDS	36
8.3 INFRAESTRUCTURA NIDS	36
8.4 ALGORITMOS DE MACHINE LEARNING	36
8.4.1 NO SUPERVISADOS	37
8.4.1.1 CLUSTERING (AGRUPAMIENTO)	37
8.4.1.2 REDUCCIÓN DE DIMENSIONALIDAD	38
8.4.1.3 DETECCIÓN DE ANOMALÍAS	39
8.4.2 SUPERVISADOS	39
8.4.2.1 NAIVE BAYES	40

8.4.2.2 DECISION TREE	42
8.4.2.3 RANDOM FOREST	43
8.4.2.4 GRADIENT BOOSTING MACHINES (GBM)	44
8.4.2.5 REDES NEURONALES	47
8.4.3 REFORZADOS	51
8.4.4 SELECCIÓN DE UN ALGORITMO	52
8.5 ENSAMBLE DE MODELOS	52
8.5.1 MAJORITY VOTING	53
8.5.2 BAGGING	53
8.5.3 BOOSTING (ADABOOST)	54
8.5.4 CONCLUSIONES	56
8.6 MÉTRICAS DE RENDIMIENTO	56
8.6.1 MATRIZ DE CONFUSIÓN	57
8.6.2 ACCURACY	58
8.6.3 RECALL	58
8.6.4 PRECISIÓN	59
8.6.5 CURVA DE PRECISIÓN-RECALL	59
8.6.6 CURVA ROC	60
8.6.7 F1-SCORE	61
8.6.8 CLASSIFICATION REPORT	62
8.6.9 CONCLUSIONES	63
8.7 DATASET UNSW-NB15	63
8.7.1 ORIGEN DEL DATASET UNSW-NB15	64
8.7.2 ANÁLISIS DEL DATASET UNSW-NB15	64
8.8 PREPROCESAMIENTO DE DATOS	66
8.8.1 LIMPIEZA DE DATOS	66
8.8.1.1 ELIMINACIÓN DE DATOS VACÍOS	66
8.8.1.2 IMPUTACIÓN DE DATOS	66
8.8.1.3 ELIMINACIÓN DE DATOS REDUNDANTES	67
8.8.2 TRANSFORMACIÓN DE DATOS	67
8.8.2.1 NORMALIZACIÓN	67
8.8.2.2 ESCALADO	68

8.8.2.3 CODIFICACIÓN DE DATOS CATEGÓRICOS	69
8.8.2.4 TOKENIZACIÓN	70
8.8.3 MANEJO DEL DESEQUILIBRIO DE CLASES	70
8.8.3.1 SUBMUESTREO (UNDERSAMPLING)	70
8.8.3.2 SOBREMUESTREO (OVERSAMPLING)	71
8.8.3.3 METODOS AVANZADOS (SMOTE, ADASYN)	72
8.9 INGENIERÍA DE CARACTERÍSTICAS	73
8.9.1 MÉTODOS DE SELECCIÓN DE CARACTERÍSTICAS	73
8.9.1.1 MÉTODO BASADO EN FILTROS	73
8.9.1.2 MÉTODO BASADO EN ENVOLTURAS	73
8.9.1.3 MÉTODO BASADO EN MODELOS	74
8.10 VALIDACIÓN	74
8.10.1 DIVISIÓN DE DATOS DE TRAINING Y TEST	74
8.10.1.1 HOLD-OUT VALIDATION	75
8.10.1.2 VALIDACIÓN CRUZADA (CROSS-VALIDATION)	76
8.10.1.3 DIVISION ESTRATIFICADA	76
8.11 ZEEK	77
8.11.1 FUNCIONALIDADES	77
8.12 ELASTIC STACK	77
8.12.1 FILEBEAT	78
8.12.2 ELASTICSEARCH	78
8.12.3 KIBANA	79
8.12.4 INTEGRACIÓN	79
<b>9. ESPECIFICACIÓN DEL SISTEMA</b>	<b>80</b>
9.1 REQUISITOS FUNCIONALES	80
9.1.1 DIAGRAMA DE CASOS DE USO	80
9.1.2 CASOS DE USO (BRYEF STYLE)	81
9.1.3 CASOS DE USO DETALLADOS	82
9.2 REQUISITOS NO FUNCIONALES	87
<b>10. SISTEMA DE DETECCIÓN DE INTRUSIONES EN RED (NIDS)</b>	<b>90</b>
10.1 PROCESO DE GENERACIÓN DE DATASETS	90
10.1.1 PREPROCESAMIENTO	91

10.1.1.1 DATASETS POR LABEL	112
10.1.1.2 DATASETS POR TIPO	114
10.1.2 ANÁLISIS DE LOS DATASETS PREPROCESADOS	115
10.2 EJECUCIÓN DE MODELOS	118
10.2.1 ALGORITMOS BINARIOS	119
10.2.2 ALGORITMOS MULTICLASIFICACIÓN	119
10.3 RESULTADOS	120
10.3.1 ANÁLISIS DE RESULTADOS	120
10.3.2 COMPARACIÓN ENTRE DATASETS	126
10.3.3 RENDIMIENTO DE LOS MODELOS	127
10.3.4 ANÁLISIS DE ERRORES	128
10.3.5 MODELO A ESCOGER	133
10.4 DISEÑO DEL NIDS-AI E IMPLEMENTACIÓN EN LOCAL	135
10.4.1 INFRAESTRUCTURA	135
10.5 CAPTURA DE TRÁFICO REAL CON ZEEK	136
10.5.1 SCRIPT LOCAL.ZEEK	139
10.5.2 RESPUESTAS LOG	140
10.6 TKINTER	142
10.7 ELASTIC STACK	144
10.8 EVALUACIÓN DEL NIDS-AI	145
<b>11. LEYES Y REGULARIZACIONES</b>	<b>147</b>
11.1 NORMATIVA DEL TRABAJO FINAL DE CARRERA	147
11.2 LEY DE PROTECCIÓN DE DATOS	147
11.3 AVISO LEGAL DE LA APLICACIÓN	148
11.4 TÉRMINOS Y CONDICIONES DE GOOGLE COLAB	149
11.5 TÉRMINOS Y CONDICIONES DE GIT	149
11.6 TÉRMINOS Y CONDICIONES DE LOS DATASETS	150
11.7 LICENCIA DE LIBRERÍAS Y PROGRAMARIO	150
11.8 PROPIEDAD INTELECTUAL	151
<b>12. SOSTENIBILIDAD</b>	<b>152</b>
12.1 ÁMBITO AMBIENTAL	152
12.2 ÁMBITO SOCIAL	152

12.3 ÁMBITO ECONÓMICO	152
12.4 AUTOEVALUACIÓN	153
<b>13. CONCLUSIONES</b>	<b>154</b>
13.1 ANÁLISIS DE ALTERNATIVAS Y MEJORAS	154
13.2 CAMBIOS Y DESVIACIONES	155
13.3 IMPACTO ECONÓMICO	156
<b>14. INTEGRACIÓN DE CONOCIMIENTOS</b>	<b>158</b>
14.1 COMPETENCIAS TÉCNICAS	159
14.2 CONCLUSIONES PERSONALES	161
<b>15. BIBLIOGRAFÍA</b>	<b>162</b>
<b>16. ÍNDICE DE FIGURAS</b>	<b>166</b>
<b>17. ÍNDICE DE TABLAS</b>	<b>168</b>
<b>ANEXO 1 - LOCAL.ZEEK</b>	<b>169</b>
<b>ANEXO 2 - SOLICITUD DE CÓDIGO ADICIONAL</b>	<b>171</b>

## 1. INTRODUCCIÓN

En el cambiante panorama digital actual, la ciberseguridad se ha convertido en un desafío crítico y en constante evolución para empresas, gobiernos e individuos. Con el incremento exponencial de los ataques cibernéticos y la creciente sofisticación de las amenazas, resulta esencial adoptar enfoques innovadores para proteger los activos digitales y salvaguardar la integridad de la información.

En este contexto, este proyecto se centra en el desarrollo e implementación de un Sistema de Detección de Intrusiones de Red con Inteligencia Artificial (NIDS-AI) [1], diseñado para fortalecer la seguridad tanto en entornos empresariales como personales. Esta solución integral aprovecha técnicas avanzadas de aprendizaje automático (ML) [2] y análisis de datos para detectar y mitigar eficazmente las amenazas cibernéticas en un entorno dinámico y desafiante.

Esta investigación radica en la creciente complejidad de los ataques cibernéticos, que pueden comprometer no solo la confidencialidad, integridad y disponibilidad de los datos, sino también la reputación y la continuidad operativa de las organizaciones. Este trabajo se enmarca dentro de las tendencias y tecnologías más avanzadas en el campo de la ciberseguridad, y busca contribuir al desarrollo de soluciones más inteligentes y adaptables para la protección de redes.

Este proyecto se enfoca en la presentación del NIDS-AI como una solución innovadora y adaptable a las necesidades actuales de protección de redes. Este sistema se inscribe en la línea de investigación que persigue soluciones más inteligentes y efectivas para la protección de redes frente a las crecientes amenazas cibernéticas, proporcionando un enfoque proactivo y adaptable para enfrentar los desafíos de la ciberseguridad en el siglo XXI.

### 1.1 MOTIVACIÓN

La motivación principal detrás del desarrollo de este proyecto radica en la urgencia de mejorar las estrategias de ciberseguridad mediante la implementación de tecnologías innovadoras. En particular, el uso de la inteligencia artificial (IA) [3] y el aprendizaje automático ofrece un potencial significativo para crear sistemas de detección de intrusiones más precisos y eficientes. La capacidad de estos sistemas para aprender y adaptarse a nuevas amenazas en tiempo real representa una ventaja crucial frente a los métodos tradicionales de detección, que a menudo resultan insuficientes ante la evolución constante de los ciberataques.

Asimismo, este proyecto se alinea con la necesidad de proteger no solo a grandes corporaciones y entidades gubernamentales, sino también a pequeñas y medianas empresas, y a usuarios individuales que, con frecuencia, carecen de recursos suficientes para implementar medidas de ciberseguridad robustas. Desarrollar un Sistema de Detección de Intrusiones en red con Inteligencia Artificial (NIDS-AI) accesible y efectivo contribuirá a cerrar esta brecha, proporcionando una capa adicional de protección que es crucial en el panorama digital actual.

Además, este proyecto representa una oportunidad invaluable para contribuir al avance de mi conocimiento en el campo de la ciberseguridad, explorando nuevas metodologías y enfoques que pueden ser aplicados de manera práctica y escalable. Al mismo tiempo, me ha permitido el desarrollo y perfeccionamiento de habilidades técnicas y analíticas en el uso de herramientas avanzadas de inteligencia artificial y aprendizaje automático.

Mi pasión por la inteligencia artificial, junto con la urgente necesidad de abordar los desafíos crecientes en el campo de la ciberseguridad, me impulsan a embarcarme en el desarrollo del NIDS-AI. En este contexto, la aplicación de técnicas de inteligencia artificial y aprendizaje automático para la detección de intrusiones se presenta como solución prometedora. La capacidad de los algoritmos de aprendizaje automático para analizar grandes volúmenes de datos, identificar patrones y anomalías y adaptarse dinámicamente a nuevas amenazas ofrece una oportunidad única para mejorar significativamente la seguridad cibernetica.

## 1.2 OBJETIVO

El objetivo principal de este proyecto es desarrollar y evaluar un Sistema de Detección de Intrusiones en Red con Inteligencia Artificial (NIDS-AI) que pueda identificar y mitigar amenazas ciberneticas en tiempo real, mejorando así la protección de los activos digitales tanto en entornos empresariales como personales. Este sistema integrará técnicas avanzadas de aprendizaje automático y análisis de datos para ofrecer una solución eficiente y adaptable a las demandas actuales de ciberseguridad.

Uno de los objetivos fundamentales del proyecto es profundizar en la comprensión del funcionamiento de los modelos de aprendizaje automático y su capacidad para aprender a partir de grandes volúmenes de datos. La inteligencia artificial aplicada a la ciberseguridad permite una detección más precisa de amenazas ciberneticas, aprovechando el poder de los algoritmos de machine learning que a menudo son subutilizados en este campo.

Otro objetivo clave es realizar un análisis exhaustivo de las necesidades y desafíos específicos de seguridad en entornos empresariales y personales. Esto incluye abordar aspectos cruciales como la protección contra ataques ciberneticos y la implementación de estrategias sólidas de seguridad en la infraestructura. Se propone diseñar una arquitectura de sistema que integre el procesamiento de datos, soluciones de detección de amenazas y respuestas automáticas, utilizando servicios en la nube. Este enfoque busca proporcionar una defensa proactiva contra las amenazas en la red, permitiendo a las empresas detectar y responder rápidamente a incidentes de seguridad.

Además, se establecerá un sistema de monitoreo continuo de eventos de seguridad y del estado de la infraestructura. Se generarán informes y alertas detalladas sobre las amenazas detectadas, proporcionando una visión completa de la postura de seguridad de la empresa.

## 1.3 ALCANCE

El presente proyecto, cuyo objetivo es el desarrollo de un Sistema de Detección de Intrusiones en Red con Inteligencia Artificial (NIDS-AI), está dirigido a una amplia variedad de usuarios y entidades que se beneficiarán significativamente de su implementación. A continuación se detallan los grupos principales a los que está dirigido y el impacto esperado en cada uno de ellos:

*Empresas y Corporaciones:* Las empresas son objetivos frecuentes de ataques ciberneticos que pueden comprometer información confidencial y operaciones críticas. La implementación del NIDS-AI permitirá a las empresas detectar y mitigar amenazas de manera proactiva, reduciendo el riesgo de interrupciones operativas y pérdidas económicas. Además, ayudará a cumplir con normativas de seguridad y protección.

*Instituciones Gubernamentales:* Los organismos gubernamentales manejan una gran cantidad de información sensible y son blanco de ciberataques que buscan comprometer la seguridad nacional y la privacidad de los ciudadanos. El NIDS-AI proporcionará a estas instituciones una herramienta avanzada para proteger infraestructuras críticas, mejorar la seguridad nacional y garantizar la integridad de los servicios públicos.

*Pequeñas y Medianas Empresas (PYMEs):* Las PYMEs [4], a menudo con recursos limitados, pueden ser especialmente vulnerables a los ciberataques. Este sistema ofrecerá una solución asequible y eficiente para que las PYMEs fortalezcan su ciberseguridad sin necesidad de grandes inversiones en tecnología o personal especializado.

*Usuarios Individuales:* Con el aumento de las amenazas cibernéticas, los individuos también se enfrentan a riesgos significativos que pueden afectar su privacidad y seguridad personal. Aunque el enfoque principal del proyecto son las organizaciones, las versiones simplificadas del NIDS-AI pueden adaptarse para su uso personal, ayudando a los usuarios a proteger sus dispositivos y datos personales.

*Profesionales y Equipos de Ciberseguridad:* Estos profesionales necesitan herramientas avanzadas para monitorear, detectar y responder a las amenazas en tiempo real. El NIDS-AI facilitará el trabajo de los equipos de ciberseguridad al proporcionarles una herramienta robusta y automatizada, permitiendo una respuesta más rápida y eficaz ante incidentes de seguridad.

*Investigadores y Académicos:* La investigación en ciberseguridad es crucial para el desarrollo de nuevas tecnologías y la mejora continua de las estrategias de defensa. Este proyecto contribuirá al cuerpo de conocimiento existente, proporcionando una base para futuras investigaciones y desarrollos en el campo de la ciberseguridad y la inteligencia artificial.

El alcance del proyecto abarca desde grandes corporaciones y organismos gubernamentales hasta pequeñas empresas y usuarios individuales, todos los cuales se beneficiarán de una mejor protección contra las amenazas cibernéticas. La adopción del NIDS-AI promete no solo mejorar la seguridad, sino también fomentar una cultura de ciberseguridad más sólida y proactiva en todos los niveles.

## 2. METODOLOGÍA

El desarrollo de un Trabajo de Fin de Grado (TFG) implica un proceso complejo que requiere organización, planificación y colaboración. En esta dinámica, la incorporación de metodologías ágiles como Scrum [5] emerge como una herramienta invaluable para dirigir el proyecto de forma eficaz y adaptable. Además, la implementación de un sólido control de versiones, junto con la creación de entornos distintos para desarrollo, pruebas y producción, potencia aún más la gestión integral del TFG, garantizando una evolución controlada y un producto final de calidad.

### 2.1 AGILE Y SCRUM

La metodología ágil Scrum es un enfoque de gestión de proyectos que se basa en roles definidos, eventos clave y artefactos específicos para fomentar la colaboración, la adaptación al cambio y la mejora continua del producto. A diferencia de los enfoques tradicionales, Scrum no impone una estructura rígida, sino que proporciona un marco flexible que se adapta a las necesidades de cada proyecto. Estos eventos permiten una colaboración estrecha entre los miembros del equipo, una adaptación continua al cambio y una mejora constante en la calidad del producto.

#### 2.1.1 ROLES

*Product Owner:* El representante de las necesidades del cliente o del proyecto, responsable de priorizar y definir las funcionalidades del producto.

*Scrum Master:* El facilitador del proceso Scrum, que guía al equipo y elimina obstáculos para garantizar un desarrollo fluido.

*Equipo de Desarrollo:* Un grupo multidisciplinario responsable de diseñar, desarrollar y probar las funcionalidades del producto.

#### 2.1.2 EVENTOS

*Planificación del Sprint:* Al inicio de cada Sprint [6], el equipo define las tareas que se completarán durante ese ciclo corto de desarrollo.

*Revisión del Sprint:* Al final del Sprint, el equipo presenta el trabajo realizado al Product Owner y a las partes interesadas, recibiendo retroalimentación valiosa.

*Retrospectiva del Sprint:* En una reunión reflexiva, el equipo analiza el Sprint anterior, identificando los aspectos positivos, los desafíos y las áreas de mejora para futuros Sprints.

#### 2.1.3 ARTEFACTOS

*Product Backlog:* Una lista priorizada de todas las funcionalidades que se deben desarrollar para el producto.

*Sprint Backlog:* Un subconjunto del Product Backlog que contiene las tareas que se abordarán durante el Sprint actual.

*Incremento:* El conjunto de funcionalidades completadas y probadas al final de cada Sprint.

## 2.1.4 SCRUM EN ACCION

Scrum se caracteriza por dividir el trabajo en ciclos cortos de desarrollo llamados "sprints". Durante cada sprint, se desarrolla un conjunto de funcionalidades priorizadas, que son determinadas en base a las necesidades del cliente o del proyecto. Los pasos utilizados son:

*Definición del proyecto:* Establece el objetivo y alcance de nuestro TFG, identificando los requisitos y entregables necesarios para su finalización.

*Creación del Product Backlog:* Enumera todas las tareas y actividades requeridas para lograr el objetivo del proyecto, priorizándolas según su importancia o dependencias.

*Planificación del Sprint:* Selecciona un marco de tiempo para cada Sprint y escoge las tareas del Product Backlog que se completarán durante ese período.

*Ejecución del Sprint:* Trabaja en las tareas asignadas durante el Sprint, siguiendo las prioridades establecidas y realizando reuniones diarias de seguimiento para mantener la comunicación y el ritmo de trabajo.

*Revisión del Sprint:* Evalúa el trabajo realizado al final del Sprint, asegurando que las tareas se hayan completado según lo planeado y que se hayan cumplido los objetivos.

*Retrospectiva del Sprint:* Revisa el Sprint en conjunto, identificando los aspectos positivos, los desafíos y las áreas de mejora. Esta retroalimentación servirá para optimizar el proceso en los siguientes Sprints.

*Repetir:* Continúa con el ciclo de Sprints, planificando, ejecutando, revisando y reflexionando sobre el progreso del proyecto hasta finalizar la última tarea del proyecto, por consecuencia finalizar el último Sprint.

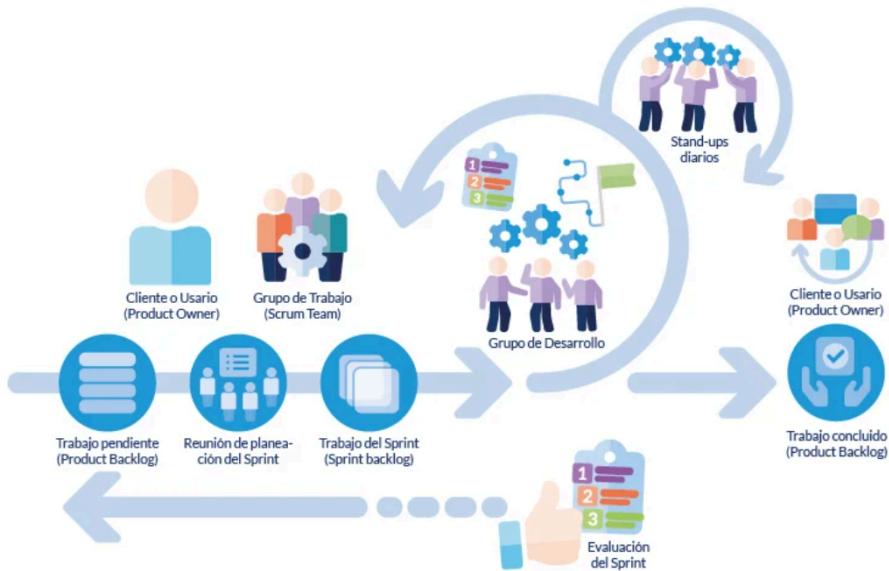


Figura 1: Metodología Agile Scrum. Fuente: Innevo.

## 2.1.5 BENEFICIOS

Los beneficios que nos genera utilizar esta metodología son:

*Flexibilidad:* Scrum permite ajustar el enfoque y los entregables a medida que avanza el proyecto, lo que es fundamental en un entorno académico donde los requisitos pueden cambiar o surgir nuevas ideas.

*Entrega incremental:* Divide el proyecto en ciclos cortos (Sprints) con entregas tangibles al final de cada uno. Esto facilita la revisión y retroalimentación por parte del tutor, permitiendo identificar y corregir desviaciones tempranamente.

*Mejora continua:* Cada Sprint finaliza con una retrospectiva para analizar lo que salió bien y lo que se puede mejorar. Esta reflexión constante fomenta la optimización del proceso y el aprendizaje continuo.

*Transparencia:* la transparencia se logra mediante artefactos como el Product Backlog, que lista todas las funcionalidades pendientes de desarrollo, y el Sprint Backlog, que contiene las tareas específicas que se abordarán durante el sprint en curso.

*Motivación:* Scrum fomenta un ambiente de trabajo colaborativo y motivador, donde cada miembro del equipo se siente valorado y responsable del éxito del proyecto.

## 2.2 SISTEMA DE CONTROL DE VERSIONES

Para controlar los cambios realizados en el código fuente, se utilizará el sistema de control de versiones Git [7]. Git es una herramienta de código abierto utilizada por grandes empresas del sector tecnológico como Google, Microsoft, Meta, entre otros.

Este sistema de control distribuido permitirá mantener un repositorio central con todo el historial de cambios, así como una copia en cada ordenador donde se esté desarrollando el proyecto.

Se seguirá una metodología de ramificación Gitflow, que consta de las siguientes ramas principales:

*Main:* Contendrá las versiones definitivas en el entorno de producción y no se modificará con frecuencia.

*Develop:* Contendrá las versiones en desarrollo donde se incorporarán todos los cambios.

*Release:* Se crearán ramas de versión a partir de la rama develop cuando se consideren suficientes funcionalidades.

*Hotfix:* Se utilizarán estas ramas para solucionar errores que requieran una corrección inmediata en el entorno de producción.

### 2.2.1 ENTORNOS DE DESARROLLO

Durante el desarrollo de la aplicación, se utilizará un entorno local en el ordenador del desarrollador. Además, existirá un entorno de pruebas para realizar pruebas de la aplicación. Finalmente, una vez completada la primera versión, se creará un entorno de producción para que los usuarios puedan utilizar el sistema.

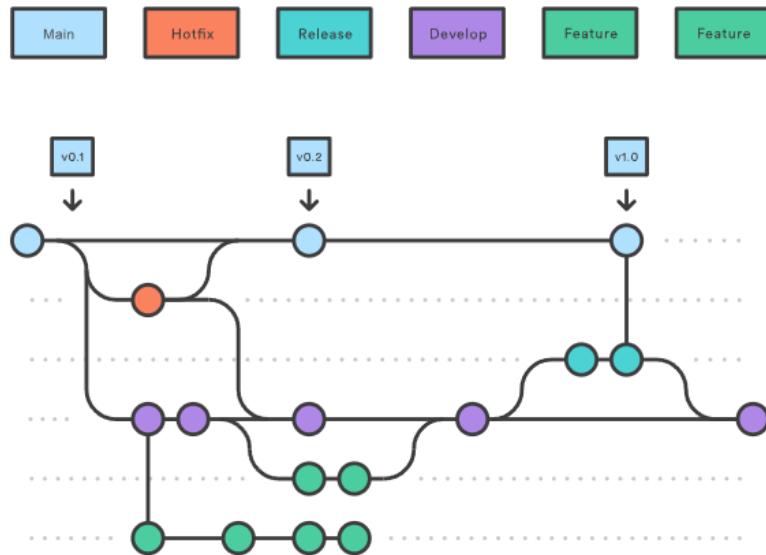


Figura 2: Sistema de control de versiones distribuido. Fuente: GitFlow.

## 2.3 PRUEBAS Y CALIDAD DE SOFTWARE

Las pruebas se ejecutarán en el entorno local de desarrollo. Además, se ejecutará un linter [8] para analizar el código y darle formato de manera consistente. Esta herramienta también detectará posibles errores y estructuras inválidas o inefficientes en el código. Para asegurar la calidad del TFG, se implementarán diferentes estrategias de testing:

*Pruebas automatizadas:* Scripts que ejecutan pruebas de forma automática para verificar el correcto funcionamiento del código y detectar posibles errores.

*Pruebas manuales:* Realizadas por los miembros del equipo para evaluar la usabilidad, la interfaz y la experiencia general del sistema.

### 3. CIBERSEGURIDAD

Las empresas deben estar preparadas contra cualquier tipo de ciberataque. Existen diversas formas de comprometer los sistemas informáticos, pero la mayoría se basa en procedimientos similares para infiltrarse y obtener datos. Por ello, se destinan recursos a la ciberseguridad como una medida preventiva esencial para proteger sus datos y sistemas, y también para estar preparadas ante posibles ataques.

Las razones fundamentales que respaldan esta inversión son varias:

*Preservación de la propiedad intelectual:* Protege las ideas exclusivas y patentes de la empresa.

*Disponibilidad continua de servicios:* Garantiza que los servicios permanezcan operativos, evitando interrupciones que podrían afectar el funcionamiento regular de la organización.

*Protección de accesos:* Asegura la confidencialidad e integridad de los datos almacenados, evitando accesos no autorizados.

*Operatividad ininterrumpida:* Contribuye a mantener un entorno de trabajo estable y seguro.

*Competitividad en el mercado:* Al fortalecer las defensas digitales, la empresa mejora su capacidad competitiva y se posiciona favorablemente en la transformación digital.

*Protección contra manipulaciones de datos:* Evita alteraciones que puedan comprometer la veracidad de la información.

*Facilitación de nuevos modelos de negocio:* Proporciona un entorno digital seguro para la innovación.

*Seguridad de dispositivos personales:* Protege los dispositivos utilizados por el personal, que pueden ser puntos de entrada para amenazas cibernéticas.

*Prevención del robo de información:* Actúa como un escudo contra accesos no autorizados y sustracciones indebidas.

El robo de información, datos e imágenes es una amenaza constante. La ciberseguridad asegura la digitalización de procesos, protegiéndolos contra vulnerabilidades que podrían comprometer su eficiencia y seguridad. [9]

Las empresas se enfrentan a una variedad de ciberataques que amenazan la seguridad de su información y la continuidad de sus operaciones. A continuación, se presentan los ciberataques más comunes según el ranking de OWASP [10] , listados en orden de frecuencia:

- Injection
- Broken Authentication
- Sensitive Data Exposure
- XML External Entities
- Broken Access Control
- Security Misconfiguration

Para protegerse contra estas amenazas, la implementación de un Sistema de Detección de Intrusiones en Red es clave. Este sistema ayuda a monitorizar y analizar el tráfico de red en busca de actividades sospechosas, alertando sobre posibles intentos de intrusión y permitiendo una respuesta rápida para salvaguardar la infraestructura digital de la empresa.

Además, las empresas deben adoptar una estrategia de ciberseguridad integral que incluya:

*Actualización constante de software y sistemas:* Para proteger contra vulnerabilidades conocidas.

*Capacitación continua del personal:* Para reconocer y evitar posibles amenazas.

*Implementación de políticas de seguridad estrictas:* Para gestionar el acceso y uso de datos y sistemas.

*Uso de soluciones de seguridad avanzadas:* Como firewalls [11], sistemas de prevención de intrusiones (IPS) [12] y tecnologías de encriptación.

Invertir en ciberseguridad no solo protege los activos digitales de una empresa, sino que también fortalece su posición en el mercado, fomenta la innovación y asegura la continuidad operativa.

## 4. DESCRIPCIÓN DE LAS TAREAS

### 4.1 INTRODUCCIÓN

El proyecto se realiza de forma personal, tratándose de una propuesta nueva que no se ha trabajado antes y que, además, será realizada únicamente por mí. Las tareas que se abordarán en este documento incluyen todas las actividades del proyecto entre el 26 de febrero y el 8 de julio, fechas de inicio y finalización del propio proyecto. En mi actual puesto de trabajo estoy realizando una jornada laboral de 6 horas diarias, de las cuales me han permitido dedicar unas horas de la jornada a este trabajo, compaginando con otras tareas asignadas dentro de la empresa.

Durante este período, dedicaré un total de 153 horas dentro de la empresa, la mayoría de las cuales se han destinado a reuniones, redacción del documento final, gestión del proyecto y preparación de la presentación final, que se realizará en la primera semana de julio. A estas horas se suman aproximadamente 381 horas de trabajo dedicadas al desarrollo y a la formación previa. En total, se estima que dedicaré 534 horas a la elaboración de este proyecto. Por lo que el papel que juega la empresa en este proyecto es dar soporte tanto en horas de trabajo útiles para el desarrollo del proyecto, como para una reunión semanal de seguimiento.

### 4.2 TAREAS

#### 4.2.1 TAREAS DE GESTIÓN Y DOCUMENTACIÓN

- *TGD1 - Contextualización y alcance (25h)*: Redacción de la primera entrega de la gestión del proyecto, que incluye el contexto, justificación, alcance y metodología. Dependencias: Ninguna.
- *TGD2 - Planificación temporal (8h)*: Revisión de la información relativa a la entrega previa y redacción de la segunda entrega de la gestión del proyecto, basada en la planificación del trabajo. Dependencias: TGD1.
- *TGD3 - Presupuesto y sostenibilidad (10h)*: Revisión de la información relativa a las entregas previas y elaboración del presupuesto del proyecto, así como un informe de sostenibilidad. Dependencias: TGD2.
- *TGD4 - Integración del documento final (20h)*: Unificación en un solo documento de las entregas de las tres tareas anteriores, adaptando su contenido para una mejor interrelación entre los diferentes apartados. Dependencias: TGD3.
- *TGD5 - Documentación del trabajo (90h)*: Redacción de las demás fases y desarrollo del trabajo. Dependencias: TGD4.
- *TGD6 - Preparación de la presentación final (30h)*: Elaboración de las diapositivas y el guión de la presentación final, así como la práctica de la misma. Dependencias: TGD5.
- *TGD7 - Reuniones de seguimiento (15h)*: Aunque habrá un seguimiento continuo, también se ha programado una reunión de una hora cada semana para realizar un seguimiento más exhaustivo y el sprint planning. Dependencias: Ninguna.

## 4.2.2 TAREAS DE DESARROLLO

Todas las tareas descritas a continuación, excepto las de la fase inicial, incluye tanto el tiempo de desarrollo de la propia funcionalidad como el desarrollo y la ejecución de los tests unitarios y de integración.

### Fase Inicial

- *FI1 - Formación (100h)*: Formación en tecnología Machine Learning, Deep Learning, Redes, Zeek, Python, Docker, Firebase y Pipelines. Dependencias: Ninguna.
- *FI2 - Especificación de requisitos (12h)*: Identificación de los diferentes requisitos y funcionalidades del sistema. Dependencias: Ninguna.
- *FI3 - Diseño inicial de la pipeline (8h)*: Se realizará un mockup de la infraestructura de la pipeline con los diferentes flujos y opciones disponibles para el usuario. Dependencias: FI2.
- *FI4 - Preparación del entorno (5h)*: Creación del repositorio de Git, instalación del software necesario y configuración de librerías. Dependencias: FI2.

### Modelado de IA

- *IA1 - Importación de los Datasets (6h)*: Importación del conjunto de datasets para el entrenamiento de los modelos de IA. Dependencias: Ninguna.
- *IA2 - Análisis de los Datasets (25h)*: Análisis del conjunto de dataset para el entrenamiento de los modelos de IA. Dependencias: IA1.
- *IA3 - Preprocesamiento de los Datasets (30h)*: Preparación del conjunto de dataset para el entrenamiento de los modelos de IA. Dependencias: IA2.
- *IA4 - Creación de los Datasets de Entrenamiento (5h)*: Creación de los diferentes datasets de entrenamiento/prueba para el entrenamiento de los modelos de IA. Dependencias: IA3.
- *IA5 - Ejecución de Modelos (57h)*: Ejecución del entrenamiento de los modelos de IA. Dependencias: IA4.
- *IA6 - Análisis de Resultados (7h)*: Análisis de los resultados de todos los entrenamiento de los modelos de IA y guardado en local los mejores modelos entrenados. Dependencias: IA5.

### Servicio Docker

### Zeek

- *ZS1 - Zeek (8h)*: Configuración del servicio Zeek y levantamiento del servicio en Docker. Dependencias: Ninguna.
- *ZS2 - Script configuración Zeek (5h)*: Script de configuración del servicio Zeek para monitoreo de tráfico y conexiones. Dependencias: ZS1.

### Kibana

- *KS1 - Kibana (10h)*: Configuración del servicio Kibana y levantamiento del servicio en Docker para la visualización de datos de Zeek. Dependencias: ZS2.

### Elasticsearch

- *ES1 - Elasticsearch (8h)*: Configuración del servicio Elasticsearch y levantamiento del servicio en Docker para el almacenamiento y búsqueda de datos. Dependencias: KS1.

### Filebeat

- *FS1 - Filebeat (8h)*: Configuración del servicio Filebeat y levantamiento del servicio en Docker para el envío de logs a Elasticsearch. Dependencias: ES1.

### App Python

- *PS1 - App Python (15h)*: Desarrollo de una App que obtenga los datos de Zeek, los envíe al modelo de IA y devuelva un resultado en log. Dependencias: ZS2, IA6.

### Pipeline

- *PL1 - Pipeline entre Zeek y el Script Python (8h)*: Integración del flujo de datos entre Zeek y el script Python para el procesamiento de datos con el modelo de IA. Dependencias: ZS2, PS1.
- *PL2 - Pipeline entre Filebeat y Zeek (7h)*: Integración del flujo de datos entre Filebeat y Zeek para la recolección de logs de red generados por Zeek. Dependencias: ZS2, FS1.
- *PL3 - Pipeline entre Zeek y Elasticsearch (6h)*: Integración del flujo de datos entre Zeek y Elasticsearch para el almacenamiento y búsqueda de logs. Dependencias: ZS2, ES1.
- *PL4 - Pipeline entre Elasticsearch y Kibana (6h)*: Integración del flujo de datos entre Elasticsearch y Kibana para la visualización de datos en tiempo real. Dependencias: ES1, KS1.

## 4.3 HERRAMIENTAS Y RECURSOS

Para un proyecto, los recursos más relevantes son los humanos. En este caso, el equipo estará formado por tres personas. En primer lugar, yo mismo (**FM**) como desarrollador del proyecto. En segundo lugar, necesitamos al director de mi empresa, Elías Cid (**EC**), quien se encargará de realizar un seguimiento exhaustivo y de facilitar un feedback sobre el progreso. Por último, el ponente del proyecto, Valentín Palonsky (**VP**), con quien me reuniré periódicamente para mostrar los avances y quien brindará apoyo cuando sea necesario. También será muy útil su feedback y ayuda durante todo el proceso de elaboración de la documentación del trabajo y preparación de la presentación.

El trabajo se realizará tanto en la oficina como en casa, dado que actualmente en la empresa se trabaja en formato híbrido. En cualquier caso, siempre se contará con el material mínimo necesario para trabajar en condiciones adecuadas.

Estos recursos materiales básicos son una mesa y una silla. Además, para desarrollar las diversas tareas es necesario un ordenador portátil, suministrado por la empresa. En casa, utilizaré un teclado, un ratón y un monitor de 24" para mayor comodidad.

Finalmente, a continuación se enumeran los recursos tecnológicos necesarios para realizar este trabajo de forma satisfactoria:

- **VSC - Visual Studio Code:** Editor de código para el desarrollo del proyecto.
- **GIT - Git:** Herramienta de control de versiones.
- **MO - Microsoft Office:** Herramientas para documentación, cálculos, etc. (Word, Excel), y para la comunicación interna (Outlook, Teams).
- **GC - Google Colab :** Herramienta utilizada para el modelaje de IA, así como para el preprocesamiento de los datos, su entrenamiento y análisis de los resultados.
- **B - Brave:** Navegador utilizado para la búsqueda de información y ejecución de la parte visual del proyecto.
- **Z - Zotero:** Herramienta para gestionar la bibliografía del proyecto.
- **GP - Gantt Project:** Herramienta para crear el diagrama de Gantt del apartado 5.1.
- **F - Figma:** Aplicación para realizar diseños de interfaces.
- **A - Atenea:** Descarga de recursos para la gestión de proyectos y herramienta de entrega de esta gestión.
- **GD - Google Drive :** Herramienta utilizada para el desarrollo de la documentación del proyecto.
- **DK - Docker :** Herramienta utilizada para levantar servicios en local.

## 5. ESTIMACIÓN Y GANTT

En la Tabla 1 se puede visualizar un resumen de todas las tareas mencionadas anteriormente, incluyendo el número de horas estimadas, dependencias y recursos necesarios para llevarlas a cabo.

ID	Tarea	Horas	Dependencias	Recursos
TGD1	Contextualización y alcance	25h	-	FM, EC, VP, MO, B, A, GD
TGD2	Planificación temporal	8h	TGD1	FM, EC, VP, MO, B, A, GP, GD
TGD3	Presupuesto y sostenibilidad	10h	TGD2	FM, EC, VP, MO, B, A, GD
TGD4	Integración del documento final	20h	TGD3	FM, EC, VP, MO, B, A, Z, GP, GD
TGD5	Documentación del trabajo	90h	TGD4	FM, EC, VP, MO, B, A, GP, GD
TGD6	Preparación de la presentación final	30h	TGD5	FM, EC, VP, MO, B, A
TGD7	Reuniones de seguimiento	15h	-	FM, EC, VP, MO, B, A, GP
FI1	Formación	100h	-	FM, VSC, B, GD
FI2	Especificación de requisitos	12h	-	FM, VP, GP, GD
FI3	Diseño inicial de la pipeline	8h	FI2	FM, VSC, B, GD
FI4	Preparación del entorno	5h	FI2	FM, VSC, B, GIT
IA1	Importación de los Datasets	6h	-	FM, VSC, B, GC
IA2	Análisis de los Datasets	25h	IA1	FM, VSC, B, GC
IA3	Preprocesamiento de los Datasets	30h	IA2	FM, VSC, B, GC
IA4	Creación de los Datasets de Entrenamiento	5h	IA3	FM, VSC, B, GC
IA5	Ejecución de Modelos	57h	IA4	FM, VSC, B, GC
IA6	Análisis de Resultados	7h	IA5	FM, VSC, B, GC, GD
ZS1	Zeek	8h	-	FM, VSC, B, DK
ZS2	Script configuración Zeek	5h	ZS1	FM, VSC, B, DK
KS1	Kibana	10h	ZS2	FM, VSC, B, DK
ES1	Elasticsearch	8h	KS1	FM, VSC, B, DK
FS1	Filebeat	8h	ES1	FM, VSC, B, DK
PS1	App Python	15h	ZS2, IA6	FM, VSC, B, DK
PL1	Pipeline entre Zeek y el Script Python	8h	ZS2, PS1	FM, VSC, B, DK
PL2	Pipeline entre Filebeat y Zeek	7h	ZS2, FS1	FM, VSC, B, DK
PL3	Pipeline entre Zeek y Elasticsearch	6h	ZS2, FS1	FM, VSC, B, DK
PL4	Pipeline entre Elasticsearch y Kibana	6h	ES1, KS1	FM, VSC, B, DK
<b>TOTAL</b>		<b>534h</b>		

Tabla 1: Resumen de tareas. Fuente: Elaboración propia.

## 5.1 DIAGRAMA DE GANTT

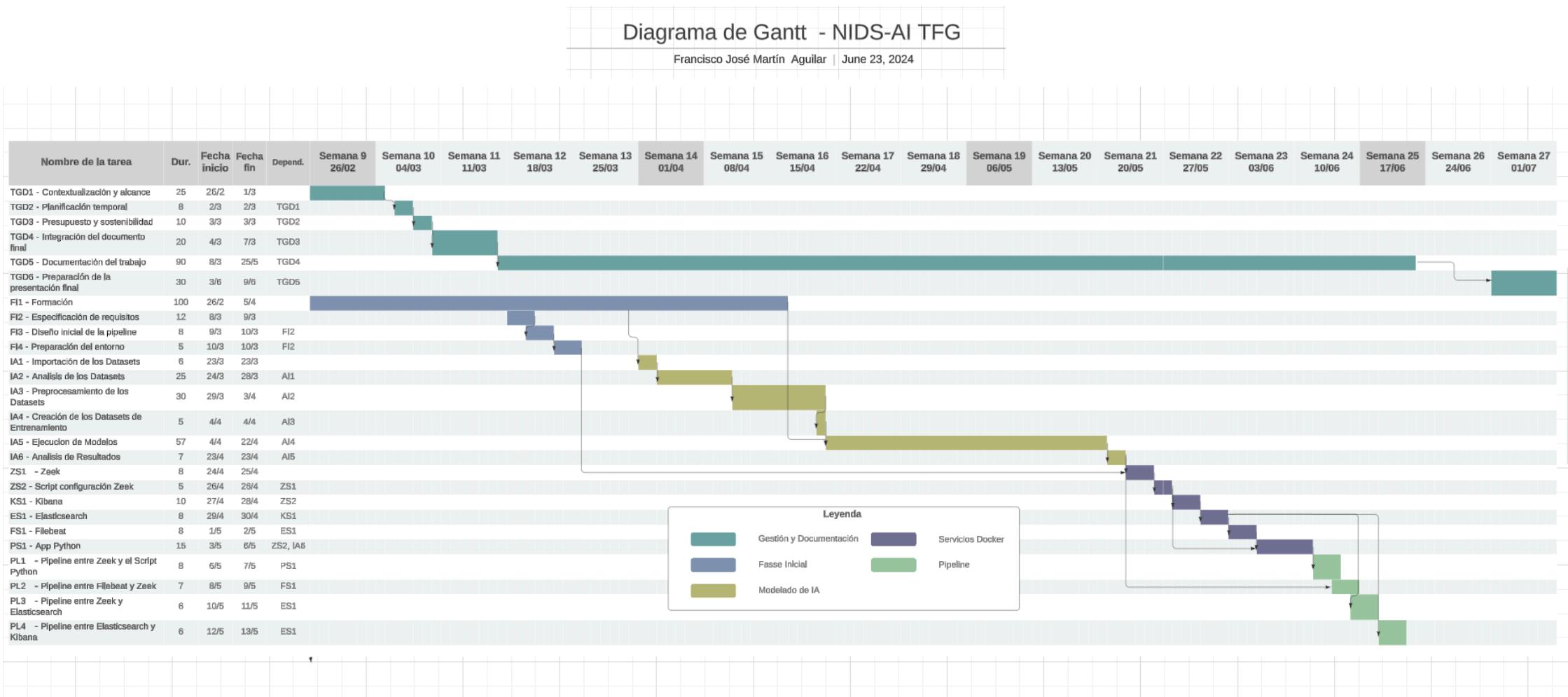


Figura 3: Diagrama de Gantt TFG - NIDS-A. Fuente: Elaboración Propia.

## 6. GESTIÓN DE RIESGOS

### 6.1 FALTA DE EXPERIENCIA EN LAS TECNOLOGÍAS

*Descripción:* Actualmente, ningún desarrollador de la empresa ni yo hemos trabajado con tecnologías de IA, por lo que la ayuda en caso de estancamiento puede ser más difícil de obtener o el propio desarrollo puede llevar más tiempo.

*Probabilidad:* La probabilidad de estancamiento o lentitud en algún momento es alta.

*Severidad:* Media. Esto ya está previsto en la planificación de la duración de las tareas. Aun así, en algunos casos podría aumentar el número de horas de alguna tarea.

*Estrategia de mitigación:* La formación en esta nueva tecnología es muy importante, por lo que al principio se requiere un gran esfuerzo de aprendizaje a través de cursos y búsqueda de información en Internet.

### 6.2 ERRORES Y FALLOS DE EJECUCIÓN

*Descripción:* Nos encontramos con uno de los riesgos que aparecen en todos los proyectos de software, sin importar el nivel de experiencia de sus desarrolladores o el conocimiento del entorno. Siempre aparece algún fallo o error.

*Probabilidad:* Como se ha mencionado, en todo proyecto se encuentra algún bug durante el desarrollo, por lo tanto, la probabilidad de que suceda es muy alta.

*Severidad:* Media. Al igual que en el caso anterior, esto también está previsto en la duración de las tareas en la sección de planificación explicada anteriormente.

*Estrategia de mitigación:* Hay una manera de mitigarlos e identificarlos. Estamos hablando de las pruebas, tanto unitarias como de integración. Estas son claves para detectar errores siempre que se realicen correctamente y la cobertura sea del 100% o casi.

### 6.3 MODIFICACIÓN DEL PLANTEAMIENTO

*Descripción:* Este es un riesgo que puede surgir en medio del desarrollo, cuando después de realizar una tarea o parte de esta, se reconsidera la manera de programarla para un beneficio posterior o por incompatibilidades con otras tareas del proyecto.

*Probabilidad:* Este riesgo se considera que tiene una probabilidad baja, ya que se realiza una buena gestión del proyecto.

*Severidad:* Dado que este riesgo implicaría rehacer parte del desarrollo, la severidad podría llegar a ser moderadamente alta, dependiendo también de la tarea en cuestión.

*Estrategia de mitigación:* La manera de evitar que suceda este riesgo es realizar correctamente las reuniones de seguimiento, tanto con el director como con el ponente del proyecto, para discutir los procesos y así prever con antelación este posible riesgo.

## 6.4 DESVIACIÓN DE LA PLANIFICACIÓN

*Descripción:* El tiempo destinado a este trabajo es aproximadamente de 3 o 4 horas al día. Sin embargo, en función de la semana y la urgencia de otras tareas durante este tiempo, puede afectar la previsión inicial de la planificación.

*Probabilidad:* Basándome en el tiempo trabajado en la empresa y algunas de las asignaturas que he realizado, la probabilidad es entre baja y media. Puede suceder, pero no en gran magnitud.

*Severidad:* Como se ha mencionado, en caso de que aparezca este problema, no sería de gran severidad, ya que no afectaría a muchas horas de diferencia.

*Estrategia de mitigación:* En caso de que una semana dedique mucho tiempo a otra tarea, la semana siguiente la dedicaría más a este trabajo, compensando así el tiempo perdido.

## 6.5 FECHA DE ENTREGA

*Descripción:* La fecha de entrega es fija y puede no ser suficiente para presentar el trabajo con todos los objetivos y criterios de fiabilidad cumplidos.

*Probabilidad:* La probabilidad es muy baja dada la planificación temporal marcada para cada una de las tareas.

*Severidad:* La severidad de no tener el proyecto listo para la fecha fijada por la Universidad es muy alta, ya que puede afectar considerablemente a la nota final.

*Estrategia de mitigación:* Para reducir el riesgo al mínimo o evitarlo completamente, se debe seguir la planificación al máximo posible y evitar dejar tareas para más adelante, como por ejemplo la redacción del documento final, tarea que habitualmente se pospone hasta los últimos días por parte de los estudiantes.

## 6.6 DEPENDENCIAS DE SERVICIOS Y LIBRERÍAS DE TERCEROS

*Descripción:* En la mayoría de proyectos se utilizan librerías de terceros para agilizar el desarrollo. Además, en este proyecto se tiene una gran dependencia de Zeek y librerías como scikit-learn. En caso de que alguna librería o servicio dejase de funcionar, el desarrollo se vería interrumpido.

*Probabilidad:* Baja. Los servicios que se utilizan son desarrollados y mantenidos por empresas de prestigio.

*Severidad:* En caso de que Zeek o algún otro servicio o librería dejase de funcionar, la severidad se considera baja.

*Estrategia de mitigación:* En el caso de que Zeek dejase de funcionar podría montar un pequeño script que capturase los paquetes de red en tiempo real con python, por lo que no afectaría al desarrollo del proyecto. Además, existen diferentes librerías como tensor flow, ... que nos proporcionan las mismas funcionalidades de entrenamiento y evaluación de modelos de inteligencia artificial que scikit-learn. Finalmente, en el caso de que otras librerías dejases de funcionar, se buscarían librerías similares, aunque siempre se revisará la información, como las empresas que las utilizan, métricas, etc.

## 7. PRESUPUESTO

Una vez ya especificadas las tareas y la estimación temporal del proyecto hasta su finalización, debemos realizar el presupuesto del proyecto para ver la viabilidad o no del mismo. Para este presupuesto se debe considerar: recursos humanos necesarios, equipamiento tecnológico (tanto hardware como software), costes generales (Internet, electricidad, material, logística, etc), los impuestos que se aplican, contingencias e imprevistos.

### 7.1 IDENTIFICACIÓN Y ESTIMACIÓN DE COSTES

Como ya se ha comentado en anteriores apartados, este trabajo se realiza en colaboración con una empresa pero, aun así, será realizado por una sola persona. Es por este motivo que, para realizar los cálculos económicos se ha decantado por calcular el precio de diferentes roles dentro del proyecto.

Un equipo de personas de una empresa trabajando en el proyecto con roles diferenciados y, en consecuencia, cada uno de ellos tendrá un salario. Dado el desconocimiento del coste que tiene para la empresa, así como el salario de cada trabajador, se han obtenido los datos de la plataforma Glassdoor [13], de donde obtenemos los salarios medios anuales. Calculamos el salario bruto por hora trabajada dividiendo este salario bruto medio anual por 1.752 (número de horas medias trabajadas en convenios en España en el año 2023), y multiplicando por 1,3 obtenemos el coste total, incluyendo la Seguridad Social a cargo de la empresa. Estos costes, sin embargo, no son los reales para una empresa, ya que se debe incluir el beneficio que debe obtener la empresa.

Además, detallo el costo por hora relacionado con el desarrollo del proyecto cuando lo ejecuto personalmente. Es decir, detallo la tarifa actual dentro de la empresa para cada uno de los roles desempeñados en el proyecto. Además, se indica el costo de cada uno de estos roles aplicando el beneficio que la empresa espera obtener, lo que representa la tarifa que el cliente pagaría a la empresa.

Esto proporciona unos datos de una previsión de costes que se describen en la Tabla 2. Para una mayor precisión, sin tener en cuenta los beneficios de un proyecto, se utilizarán los costes de Glassdoor, dado que, tal y como se puede ver a continuación, los precios con beneficios se encuentran muy inflados.

Perfil	Coste	Coste con beneficio	Coste Glassdoor (x 1,3)
JP (Jefe de Proyecto)	28,78€	60,00€	31,91€
A (Analista programador)	24,65€	45,00€	24,11€
UX (Diseñador de experiencia de usuario)	22,89€	45,00€	26,71€
P (Programador)	18,58€	43,00€	26,34€
PM (Programador ML)	26,65€	43,00€	29,68€

Tabla 2: Recursos humanos del projecte con sus costes. Fuente: Elaboración propia.

Una vez analizados los costes por cada uno de los roles que forman parte de este proyecto, se puede especificar, en la Tabla 3, el coste que tendrá en cada una de las tareas detalladas en el Diagrama de Gantt, separadas por cada rol, y finalmente con el total correspondiente.

ID	Tarea	Horas Total	Horas					Coste	Coste (Glassdoor)
			JP	A	UX	P	PM		
TGD1	Contextualización y alcance	25h	5	20				636,90€	641,75€
TGD2	Planificación temporal	8h	7	1				226,11€	247,48€
TGD3	Presupuesto y sostenibilidad	10h	10					287,80€	319,10€
TGD4	Integración del documento final	20h	17	3				563,21€	614,80€
TGD5	Documentación del trabajo	90h	55	35				2445,65€	2.598,90€
TGD6	Preparación de la presentación final	30h	30					863,40€	957,30€
TGD7	Reuniones de seguimiento*	15h	15	15	5	15	15	431,70€	1.814,15€
FI1	Formación	100h				50	50	2261,50€	2.801,00€
FI2	Especificación de requisitos	12h	6	6				320,58€	336,12€
FI3	Diseño inicial de la pipeline	8h		2		6		160,78€	206,26€
FI4	Preparación del entorno	5h				2,5	2,5	113,08€	140,05€
IA1	Importación de los Datasets	6h				6		111,48€	158,04€
IA2	Análisis de los Datasets	25h					25	666,25€	742,00€
IA3	Preprocesamiento de los Datasets	30h					30	799,50€	890,40€
IA4	Creación de los Datasets de Entrenamiento	5h					5	133,25€	148,40€
IA5	Ejecución de Modelos	57h					57	1519,05€	1.691,76€
IA6	Análisis de Resultados	7h					7	186,55€	207,76€
ZS1	Zeek	8h				4	4	180,92€	224,08€
ZS2	Script configuración Zeek	5h				2,5	2,5	113,08€	140,05€
KS1	Kibana	10h			8	2		220,28€	266,36€
ES1	Elasticsearch	8h				8		148,64€	210,72€
FS1	Filebeat	8h				8		148,64€	210,72€
PS1	App Python	15h				15		278,70€	395,10€
PL1	Pipeline entre Zeek y el Script Python	8h				4	4	180,92€	224,08€
PL2	Pipeline entre Filebeat y Zeek	7h				3,5	3,5	158,31€	196,07€
PL3	Pipeline entre Zeek y Elasticsearch	6h				3	3	135,69€	168,06€
PL4	Pipeline entre Elasticsearch y Kibana	6h				3	3	135,69€	168,06€
<b>TOTAL</b>		<b>534h</b>						<b>13.427,66€</b>	<b>16.718,57€</b>

Tabla 3: Recursos humanos del proyecto con sus costes. Fuente: Elaboración propia.

Nota: \* Solo se tiene en cuenta una vez para el cómputo total.

### 7.1.1 EQUIPAMIENTO HARDWARE

Para desarrollar el software del proyecto, es necesario tener un mínimo de equipamiento físico donde poder trabajar, como un ordenador, un teclado y un ratón, y, en este caso, también un monitor secundario para trabajar con una mayor comodidad.

En la Tabla 4 se observan los costes y amortizaciones de estos elementos. Para el cálculo de las amortizaciones, hemos seguido la tabla de amortizaciones publicadas por Hacienda, según el Real Decreto 1777/2004 [14].

El coste de la amortización se realiza de la siguiente manera:

$$\frac{\text{Coste} (\text{€})}{\text{Vida Útil (años)} * \text{Horas / año (h/año)}} * \text{Duración del Proyecto (h)}$$

Material	Coste	Vida útil	Amortizacion
Portátil Lenovo IdeaPad 530s	799,99€	4	60,96€
Teclado Ozone Strike Battle MX	89,99€	4	6,86 €
Raton Logitech G PRO SUPERLIGHT	99,00	4	7,55€
Monitor AOC 24G2SPAE	139,00€	4	10,60€
<b>TOTAL</b>			<b>85,97€</b>

Tabla 4: Tabla de costes del equipamiento de hardware necesario. Fuente: Elaboración propia.

Para obtener los costes del proyecto, deberemos multiplicar estas cifras por cinco. De esta forma, los cinco integrantes del proyecto dispondrán del equipo necesario para el desarrollo.

### 7.1.2 EQUIPAMIENTO SOFTWARE

En la Tabla 5 se especifican los costes del software necesario para el desarrollo, comunicación y ejecución del proyecto. Como puede verse, la mayoría de los productos son de software gratuito. Por otra parte, la licencia de Microsoft Office es de tipo mensual. En el caso de cualquier programa informático, según Hacienda, se puede amortizar en un período mínimo de 3 años, dado que el porcentaje máximo es del 33%.

Para el caso de la suscripción de Microsoft Office, es de 11,48€ mensual por cada usuario. Por tanto, como en lo que respecta a los equipos informáticos hardware, deberían adquirirse 5 licencias. El servidor utilizado para el proyecto se trata de un Hetzner de nivel medio-alto con un coste mensual de 19,52€ .

Tanto la suscripción como el pago del servidor mensual, que son los dos únicos servicios de pago, son gastos no amortizables, ya que al ser suscripciones no son bienes que posee la empresa.

Software	Coste (Anual)
Visual Studio Code	Gratis
GitHub	Gratis
Docker	Gratis
Brave	Gratis
Google Drive	Gratis
Gantt Project	Gratis
Google Colab	Gratis*
Microsoft Teams	Gratis
Servidor de pruebas	234,24€
Microsoft Office	137,76€
<b>TOTAL</b>	<b>372€</b>

Tabla 5: Tabla de costes del equipamiento de software necesario. Fuente: Elaboración propia.

### 7.1.3 COSTE GENERAL

A continuación debemos considerar los costes generales del proyecto como pueden ser el material de oficina (mesa, silla) u otros elementos como el desplazamiento, electricidad, etc.

Para el primer caso, el material de oficina, la amortización mínima que Hacienda permite aplicar es de 10 años.

Por otro lado, el coste del desplazamiento se hará teniendo en cuenta el precio de la gasolina a fecha de la gestión inicial de este proyecto (febrero 2024), que es de 1,562€ el litro de carburante. Siendo 8 kilómetros de ida y vuelta, y un gasto medio de 6,5 litros cada 100 kilómetros, hace un total de 0,81€ de desplazamiento por cada jornada, que se aplica dos veces por semana. Al ser 19 semanas las valoradas para el desarrollo del proyecto, se han contabilizado 38 desplazamientos.

El coste de la electricidad se calcula a partir de la suma de los elementos que se han comentado anteriormente, que son: el ordenador (65W), el monitor (27W) y un par de luces (18W cada una). En total, 128 W de consumo, tanto si me encuentro en la oficina como teletrabajando en casa durante 19 semanas entre las fechas de inicio y finalización del proyecto, a 0,50368512€/día. El precio a fecha de la gestión del proyecto con el que se realizarán los cálculos es de 0,16396€/kWh de media.

También se ha considerado el coste de internet teniendo este un coste de 42,59€/mes, utilizado durante los 5 meses de duración del proyecto.

Finalmente, dado que trabajo más días en casa que en la oficina, el cálculo se ha realizado valorando los elementos que tengo en casa. Teniendo en cuenta estos aspectos, en la Tabla 6 se pueden consultar los costes generales del proyecto. Esta tabla refleja el caso de una única persona, para obtener los costes en un caso real, debería multiplicarse por cinco.

<b>Costes</b>	<b>Coste</b>	<b>Amortización</b>
Material de oficina	359,00€	10,94€
Desplazamiento	30,78€	No aplica
Electricidad	66,99€	No aplica
Internet	212,95€	No aplica
<b>Total</b>		<b>321,66€</b>

Tabla 6: Tabla de costes de los costes generales del proyecto. Fuente: Elaboración propia.

Por último, debemos considerar los costes de los diversos riesgos e imprevistos mencionados y documentados anteriormente. Estos costes están especificados en la Tabla 7. Cómo estos riesgos e imprevistos afectarán a los desarrolladores (tanto al programador como al programador machine learning), se multiplica el tiempo por la media de los dos salarios en partes iguales (22,62€).

<b>Riesgo</b>	<b>Probabilidad</b>	<b>Tiempo</b>	<b>Coste estimado</b>	<b>Coste</b>
Falta de experiencia en las tecnologías	50%	20h	452,40€	226,20€
Errores y fallos de ejecución	70%	20h	452,40€	316,68€
Modificación del planteamiento	5%	20h	452,40€	22,62€
Desviación de la planificación	25%	5h	113,10 €	28,28€
Fecha de entrega	5%	10h	226,20€	11,31€
Dependencia de servicios y librerías de terceros	5%	5h	113,10 €	5,66€
<b>TOTAL</b>			<b>1.809,60€</b>	<b>610,75€</b>

Tabla 7: Tabla de costes de imprevistos. Fuente: Elaboración propia.

Una vez revisados y desglosados todos los costes del proyecto, podemos calcular el coste total estimado del proyecto. Puede verse en la siguiente tabla 8, el coste de cada punto y el total del proyecto. Para este proyecto, se han establecido unos costes de contingencia del 15%, teniendo en cuenta que los proyectos informáticos oscilan entre 10% y 20%.

	<b>Coste</b>
Coste de las tareas	16.718,57€
Coste de gastos generales	779,63€
Contingencias	2.624,73€
Imprevistos	610,75€
<b>TOTAL</b>	<b>20.733,68€</b>

Tabla 8: Tabla de costes totales. Fuente: Elaboración propia.

## 7.2 CONTROL DE GESTION

Hasta ahora hemos realizado la estimación previa al inicio del proyecto, y por tanto, es una estimación prevista y no la definitiva. Durante el desarrollo hay que tener constancia de los costes y dedicación real para comparar lo estimado con la realidad. A medida que se vayan dedicando horas a una tarea, se irá guardando este valor para conseguir las diversas desviaciones que pueda tener el proyecto. Esto se conseguirá teniendo contabilizados estos datos:

Desviación temporal en precio = (coste estimado - coste real) \* horas dedicadas

Desviación en consumo = (consumo horas estimadas - consumo horas reales) \* coste estimado

Desviación costes de imprevistos = (coste imprevisto estimado - coste imprevisto real)

Desviación total horas = (horas estimado - horas real)

Desviación total costes= (costes totales estimados - coste total real)

Una vez finalizado el desarrollo se realizarán los cálculos, y así se obtendrán los desvíos finales del proyecto, pudiendo desglosar y analizar si se han estimado o no correctamente estos costes y el porqué de las desviaciones.

## 8. MARCO TEÓRICO

### 8.1 SISTEMA DE DETECCIÓN DE INTRUSIONES

Un Sistema de Detección de Intrusiones (IDS) [15] es una herramienta esencial en la seguridad informática diseñada para monitorear y analizar actividades dentro de una red o sistema informático con el fin de identificar comportamientos sospechosos o maliciosos. El IDS desempeña un papel crucial en la identificación y alerta temprana sobre posibles intentos de intrusión, ayudando a proteger los activos y datos de una organización.

#### 8.1.1 COMPONENTES IDS

**Sensores:** Dispositivos o software que capturan datos en tiempo real de la red o el sistema. Estos datos pueden incluir paquetes de red, logs de eventos, archivos de registro, y más.

**Motor de Análisis:** Procesa los datos capturados por los sensores para detectar comportamientos anómalos o patrones que coinciden con firmas conocidas de ataques o mediante modelos de inteligencia artificial.

**Base de Datos de Firmas:** Contiene firmas o patrones de ataques conocidos. Esta base de datos se actualiza regularmente para incluir nuevas amenazas.

**Consola de Gestión:** Interfaz utilizada por los administradores para configurar el IDS, revisar alertas, y generar informes.

**Sistema de Alerta:** Notifica a los administradores sobre actividades sospechosas mediante diferentes métodos como correos electrónicos, mensajes de texto, o notificaciones en tiempo real.

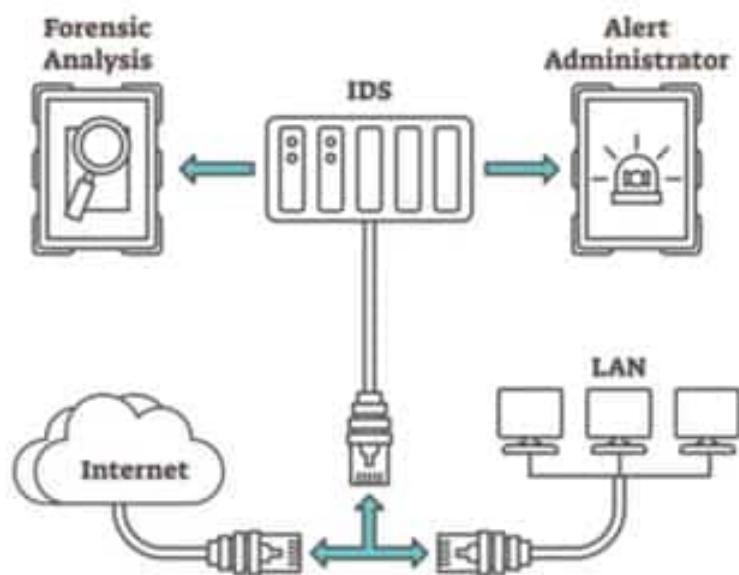


Figura 4: Esquema de componentes IDS. Fuente: Clavei.

## 8.1.2 TIPOLOGÍA DE IDS

Existen diferentes tipos de IDS, cada uno con características específicas y aplicaciones particulares. A continuación, se presentan los principales tipos de IDS: el IDS Basado en Red (NIDS) y el IDS Basado en Host (HIDS).

### 8.1.2.1 IDS BASADO EN RED (NIDS)

El IDS Basado en Red (NIDS) se enfoca en monitorear el tráfico de red en puntos estratégicos, como switches y routers. Su objetivo principal es detectar actividades sospechosas que atraviesan la red. Los NIDS son especialmente útiles para identificar ataques como escaneos de puertos, inundaciones de tráfico (DoS/DDoS) y otras intrusiones en la red. Herramientas populares como Snort y Suricata son ejemplos de NIDS que se utilizan ampliamente en entornos empresariales.

### 8.1.2.2 IDS BASADO EN HOST(HIDS)

El IDS Basado en Host (HIDS), por otro lado, monitorea las actividades de un solo equipo o servidor. Analiza los logs del sistema, registros de archivos y otras actividades internas para detectar comportamientos anómalos. Los HIDS son eficaces para identificar modificaciones no autorizadas de archivos, accesos indebidos y actividades de malware. Ejemplos destacados de HIDS incluyen OSSEC y Tripwire, que proporcionan una monitorización detallada del host y ayudan a mantener la integridad del sistema.

### 8.1.2.3 CARACTERÍSTICAS HIDS VS NIDS

Tanto los NIDS como los HIDS son componentes cruciales en una estrategia de seguridad integral. Mientras que los NIDS ofrecen una visión amplia de la red, los HIDS proporcionan una vigilancia detallada a nivel de host, monitorizando actividades específicas y cambios en los sistemas individuales. Una combinación de ambos sistemas puede ofrecer la mejor protección contra una variedad de amenazas.

Esta tabla la Tabla 9 proporciona una comparación detallada de las principales funcionalidades, ventajas y limitaciones de cada tipo de IDS. Esta información es esencial para determinar cuál tipo de IDS es más adecuado según las necesidades específicas del proyecto que se va a implementar. [16]

Característica	NIDS	HIDS
Alcance de Monitoreo	Tráfico de red en puntos estratégicos	Actividades y archivos en un solo equipo
Tipos de Ataques Detectados	Escaneos de puertos, DoS/DDoS, intrusiones en la red	Modificaciones de archivos, accesos no autorizados, actividades de malware
Ejemplos Comunes	Snort, Suricata	OSSEC, Tripwire
Ventajas	Detecta ataques en toda la red, proporciona visibilidad en tiempo real	Alta precisión en la detección de cambios internos, monitoreo detallado del host
Desventajas	Puede ser ciego a actividades dentro de los hosts, dependiente del tráfico que puede ver	Limitado a un solo equipo, no proporciona una visión completa de la red

Tabla 9: Características HIDS vs NIDS. Fuente: Elaboración propia

### **8.1.3 MÉTODOS TRADICIONALES**

Los métodos tradicionales de detección de intrusiones se basan principalmente en dos enfoques: la detección basada en firmas y la detección basada en anomalías.

#### **8.1.3.1 DETECCIÓN BASADA EN FIRMAS**

El método de detección basada en firmas, también llamado Signature-Based Detection, utiliza una base de datos de firmas, que son patrones predefinidos de ataques conocidos. Cada vez que se recibe un nuevo paquete de datos, se compara con esta base de datos para identificar posibles intrusiones.

Este método contiene una alta precisión para ataques conocidos y una rápida detección de amenazas conocidas. Pero es ineficaz contra ataques desconocidos o nuevos (zero-day) y requiere actualizaciones constantes de la base de datos de firmas.

#### **8.1.3.2 DETECCIÓN BASADA EN ANOMALÍAS**

El método Detección Basada en Anomalías, también conocido como Anomaly-Based Detection, implica la creación de un perfil del comportamiento normal de la red o sistema. Cualquier desviación significativa de este comportamiento normal se considera sospechosa y puede ser señal de una posible intrusión.

Este enfoque es capaz de detectar ataques nuevos o desconocidos y proporciona una visión más amplia de la seguridad del sistema. Pero también conlleva una alta tasa de falsos positivos y requiere tiempo y recursos para establecer un perfil preciso del comportamiento normal. [17]

### **8.1.4 APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL**

La aplicación de la inteligencia artificial (IA) [18] en los sistemas de detección de intrusiones ha revolucionado la forma en que se detectan y responden las amenazas. La IA introduce métodos más avanzados y eficientes para identificar intrusiones en comparación con los métodos tradicionales.

#### **8.1.4.1 APRENDIZAJE AUTOMÁTICO (MACHINE LEARNING)**

Los algoritmos de aprendizaje automático pueden analizar grandes volúmenes de datos y aprender a identificar patrones sospechosos sin necesidad de una programación explícita.

Este método puede mejorar continuamente a medida que se analizan más datos, detecta tanto amenazas conocidas como desconocidas con mayor precisión, pero requiere de grandes volúmenes de datos para su entrenamiento y recursos computacionales significativos. Además los modelos pueden ser susceptibles a sesgos si los datos de entrenamiento no son representativos.

#### **8.1.4.2 APRENDIZAJE PROFUNDO (DEEP LEARNING)**

Utiliza redes neuronales profundas para analizar datos complejos y no estructurados. Estas redes pueden identificar características sutiles y correlaciones que los métodos tradicionales podrían pasar por alto.

Por lo que este método tiene una alta precisión en la detección de patrones complejos y la capacidad para procesar datos en tiempo real y aprender de ellos.

Aunque sea uno de los mejores métodos, este requiere de una infraestructura de hardware potente y especializada, de un tiempo de entrenamiento largo y la necesidad de datos masivos.

### **8.1.4.3 PROCESAMIENTO DEL LENGUAJE NATURAL (NLP)**

Aplicado en la detección de amenazas a través de análisis de textos, como correos electrónicos, mensajes de chat y logs de sistemas, identificando contenido malicioso o comunicaciones sospechosas.

Este método es específico para mejorar la detección de ataques basados en ingeniería social y phishing, el cual permite un análisis más detallado y contextualizado de los datos textuales.

A pesar de ser un método muy eficiente para casos de ingeniería social y de análisis textual, requiere modelos específicos, entrenamiento en idiomas y sobre todo, un contexto de la empresa. Además puede ser un modelo bastante complejo de implementar y de mantener actualizado.

## **8.2 SISTEMA DE DETECCIÓN DE INTRUSIONES EN RED**

Mi Trabajo de Fin de Grado (TFG) se centra en el desarrollo e implementación de un Sistema de Detección de Intrusiones en Red (NIDS). Este sistema cumple varias funciones esenciales para garantizar la seguridad de la red, las cuales se detallan a continuación:

### **8.2.1 FASES DEL NIDS**

*Monitoreo del Tráfico de Red:* El NIDS se coloca en puntos estratégicos de la red, como switches, routers o firewalls, para capturar y analizar el tráfico de datos que la atraviesa. La principal tarea en esta etapa es observar todo el tráfico que fluye a través de la red, captura todos los paquetes de datos que se transmiten para su análisis posterior. Este monitoreo incluye la observación de los protocolos utilizados en la comunicación, como HTTP, FTP, DNS, entre otros.

*Análisis y Detección:* El NIDS tradicionalmente utiliza los métodos principales, de detección basada en firmas o anomalías, para detectar actividades maliciosas. En este trabajo, para la parte de análisis y detección utiliza inteligencia artificial.

Una vez que el NIDS ha recopilado los datos de la red, entra en una fase de análisis y detección asistida por inteligencia artificial (IA). Utiliza avanzados algoritmos de aprendizaje automático para inspeccionar y comprender el tráfico de red en busca de patrones y firmas de ataques conocidos. Además de comparar el tráfico con una base de datos de firmas reconocidas, la IA permite al NIDS identificar comportamientos anómalos que podrían indicar nuevas amenazas no previamente identificadas. Esto implica no solo el perfilado de usuarios y dispositivos para determinar lo que constituye un comportamiento normal, sino también la capacidad de adaptarse y evolucionar para detectar y responder de manera proactiva a las amenazas emergentes en tiempo real.

**Generación de Alertas:** Si el NIDS identifica una actividad sospechosa o un posible ataque, genera inmediatamente una alerta. Estas alertas son notificadas en tiempo real a los administradores de la red, permitiéndoles actuar rápidamente. Además, el sistema registra todos los eventos y alertas para su posterior análisis y referencia. Este registro detallado es crucial para comprender la naturaleza de las amenazas y para llevar a cabo un análisis forense si es necesario.

**Acciones Automáticas:** Algunos NIDS están configurados para tomar acciones automáticas cuando detectan amenazas. Estas acciones pueden incluir bloquear direcciones IP maliciosas, cerrar conexiones sospechosas o modificar reglas del cortafuegos para prevenir ataques adicionales. La capacidad de respuesta automática ayuda a mitigar las amenazas rápidamente, minimizando el daño potencial.

**Generación de Informes:** El NIDS genera informes detallados sobre todos los eventos de seguridad detectados. Estos informes incluyen estadísticas sobre el tráfico de red, los tipos de amenazas detectadas y las acciones tomadas en respuesta a estas amenazas. Los informes pueden ser utilizados por los administradores de red para analizar tendencias, identificar vulnerabilidades y mejorar las estrategias de seguridad a largo plazo.

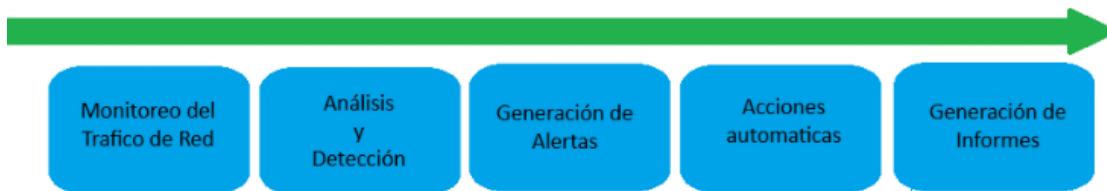


Figura 5: Fases de un NIDS. Fuente: Elaboración propia

### 8.2.2 CONCLUSIONES DEL NIDS

Los NIDS son herramientas altamente efectivas para detectar y prevenir una amplia gama de amenazas cibernéticas en una red, desde ataques conocidos hasta nuevas y emergentes formas de intrusión.

Gracias a su capacidad para monitorear el tráfico de red en tiempo real y generar alertas instantáneas ante actividades sospechosas, los NIDS permiten a los administradores de seguridad tomar medidas rápidas y proactivas para proteger la red y sus activos.

Al proporcionar una visión completa del tráfico de red y generar informes detallados sobre las amenazas detectadas, los NIDS ayudan a los equipos de seguridad a comprender mejor los riesgos y vulnerabilidades de la red, lo que les permite implementar medidas preventivas y correctivas más efectivas.

Los NIDS emplean tecnologías avanzadas, como inteligencia artificial y aprendizaje automático, lo que les permite adaptarse y evolucionar para enfrentar nuevas y cambiantes amenazas cibernéticas. Además, su capacidad para actualizar constantemente las firmas de amenazas y las reglas de detección garantiza una protección continua y actualizada.

A pesar de sus numerosos beneficios, los NIDS también enfrentan desafíos, como la posibilidad de generar falsos positivos o negativos, la necesidad de configuraciones y mantenimiento adecuados, y la dependencia de la visibilidad del tráfico de red.

### 8.2.3 EJEMPLOS DE NIDS

Snort es uno de los NIDS más populares y ampliamente utilizados. Es de código abierto y ofrece capacidades avanzadas de detección de intrusiones. Soporta tanto detección basada en firmas como basada en anomalías y puede ser configurado para una amplia variedad de entornos de red.

Suricata es otra herramienta de código abierto que proporciona capacidades de IDS, IPS (Sistema de Prevención de Intrusiones) y monitoreo de seguridad en tiempo real. Es conocido por su alto rendimiento y su capacidad para manejar grandes volúmenes de tráfico de red a alta velocidad.

## 8.3 INFRAESTRUCTURA NIDS

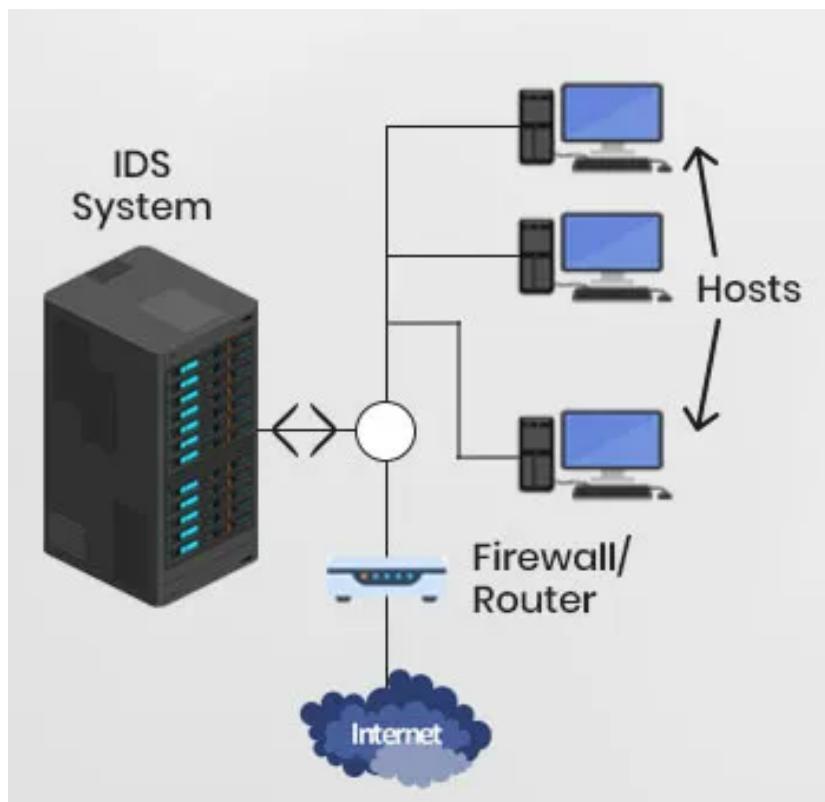


Figura 6: Infraestructura NIDS. Fuente: TeamOK

## 8.4 ALGORITMOS DE MACHINE LEARNING

El Machine Learning, o Aprendizaje Automático en español, es una rama de la Inteligencia Artificial (IA) que dota a los sistemas informáticos de la capacidad de aprender y mejorar su rendimiento de forma autónoma, sin necesidad de ser programados explícitamente. En lugar de seguir instrucciones predefinidas, los algoritmos de machine learning aprenden a partir de datos y experiencias, identificando patrones y relaciones complejas para realizar predicciones o tomar decisiones. Los algoritmos de machine learning se dividen principalmente en tres categorías: no supervisados, supervisados y por refuerzo. [19]

### 8.4.1 NO SUPERVISADOS

Los algoritmos no supervisados se utilizan cuando los datos carecen de etiquetas predefinidas, es decir, no se proporciona una respuesta correcta junto con los datos de entrada. A diferencia del aprendizaje supervisado, estos algoritmos operan de forma independiente, buscando patrones ocultos o estructuras subyacentes en los datos por sí mismos.

Los algoritmos no supervisados son una herramienta poderosa para descubrir conocimiento oculto en grandes conjuntos de datos, especialmente cuando no hay etiquetas disponibles. Su capacidad para identificar patrones complejos y relaciones no obvias los convierte en una herramienta esencial para el análisis de datos en diversos campos.

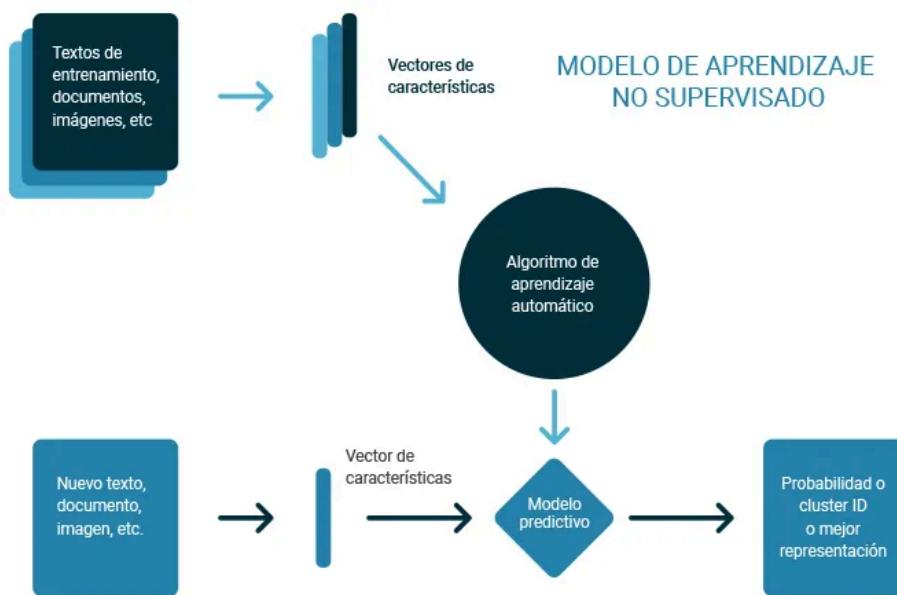


Figura 7: Modelo de aprendizaje no supervisado. Fuente: Medium

#### 8.4.1.1 CLUSTERING (AGRUPAMIENTO)

El clustering, también conocido como agrupamiento o análisis de clusters, es una técnica de aprendizaje no supervisado que se utiliza para agrupar un conjunto de datos en grupos o clusters de forma que los elementos dentro de un mismo cluster sean similares entre sí y diferentes de los elementos de otros clusters.

El objetivo que tiene el clustering es identificar grupos de elementos que comparten características o comportamientos comunes, facilitar la interpretación y el análisis de grandes conjuntos de datos complejos y agrupar elementos con características similares para realizar análisis específicos o tomar decisiones segmentadas. Por ejemplo K-means, un algoritmo de clustering que divide los datos en k clusters, donde cada punto de datos pertenece al cluster con la media más cercana. [20]

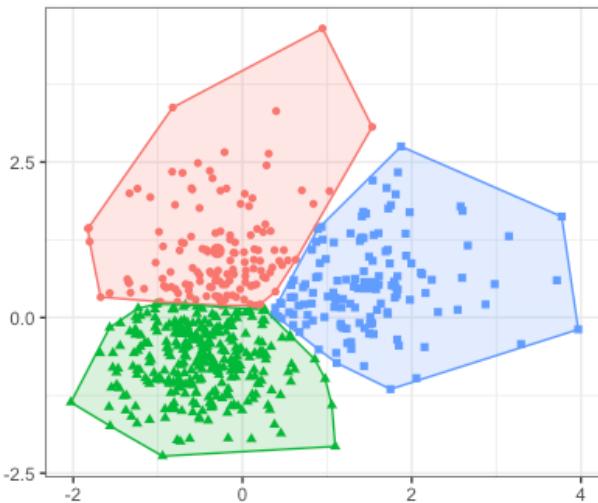


Figura 8: Clustering (Agrupamiento). Fuente: BookDown

#### 8.4.1.2 REDUCCIÓN DE DIMENSIONALIDAD

La reducción de dimensionalidad es una técnica de aprendizaje no supervisado que se utiliza para disminuir el número de variables en un conjunto de datos sin perder información importante. Esto se logra mediante la transformación de las variables originales en un nuevo conjunto de variables con menor dimensionalidad.

Los objetivos de la reducción de dimensionalidad es simplificar el análisis y la visualización de datos de alta dimensionalidad, así como reducir el tiempo de computación y la memoria requerida para el entrenamiento y la ejecución de modelos de machine learning. Sobre todo se encarga de mitigar el problema del sobreajuste en modelos de machine learning, donde el modelo se ajusta demasiado a los datos de entrenamiento y no generaliza bien a nuevos datos. [21]

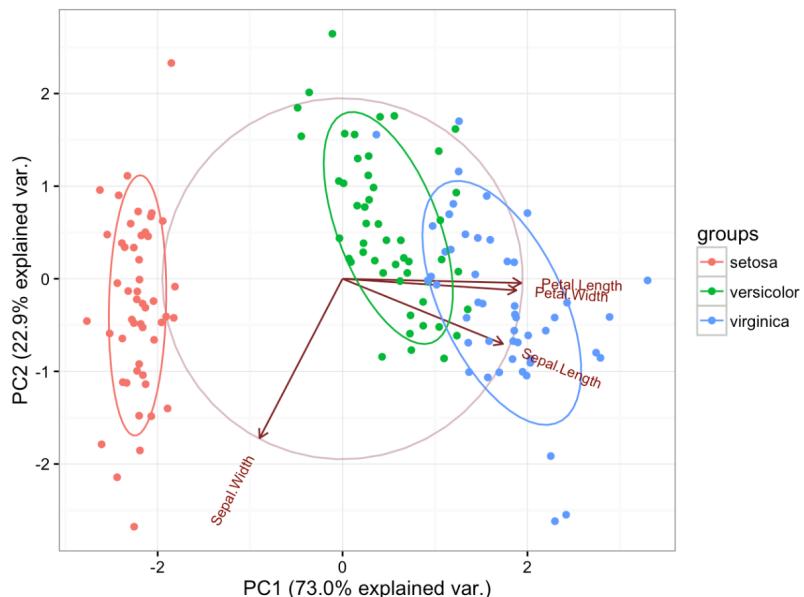


Figura 9: Reducción de dimensionalidad. Fuente: RPubs

### 8.4.1.3 DETECCIÓN DE ANOMALÍAS

La detección de anomalías es una técnica de aprendizaje no supervisado empleada para identificar elementos en un conjunto de datos que se desvían del comportamiento normal o esperado. Estos elementos anómalos pueden indicar errores, fraudes o eventos inusuales que requieren atención.

El objetivo de la detección de anomalías es identificar comportamientos atípicos en sistemas informáticos, redes o procesos industriales, con el fin de prevenir fallos o intrusiones. Al detectar estas irregularidades de manera temprana, se pueden tomar medidas proactivas para mitigar riesgos y proteger la integridad y seguridad de los sistemas. [22]

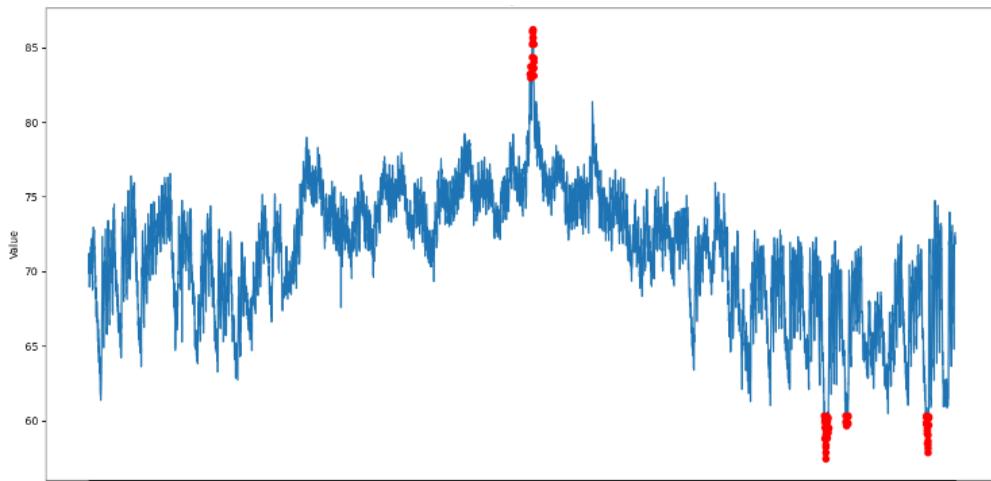


Figura 10: Detección de anomalías en datos. Fuente: GeeksForGeeks

### 8.4.2 SUPERVISADOS

Los algoritmos supervisados se utilizan cuando los datos tienen etiquetas asociadas o clases predefinidas. A diferencia del aprendizaje no supervisado, estos algoritmos aprenden a establecer una relación entre las entradas (datos) y las salidas correctas (etiquetas), lo que permite realizar predicciones o clasificaciones precisas.

Estos algoritmos son herramientas poderosas para obtener predicciones precisas y clasificaciones confiables a partir de datos etiquetados. Su capacidad para aprender de ejemplos y generalizar a nuevos datos los convierte en elementos esenciales para el análisis de datos en diversos campos. El objetivo principal es aprender una función que mapea correctamente las entradas a las salidas correspondientes, optimizando así la precisión y la fiabilidad de las predicciones.

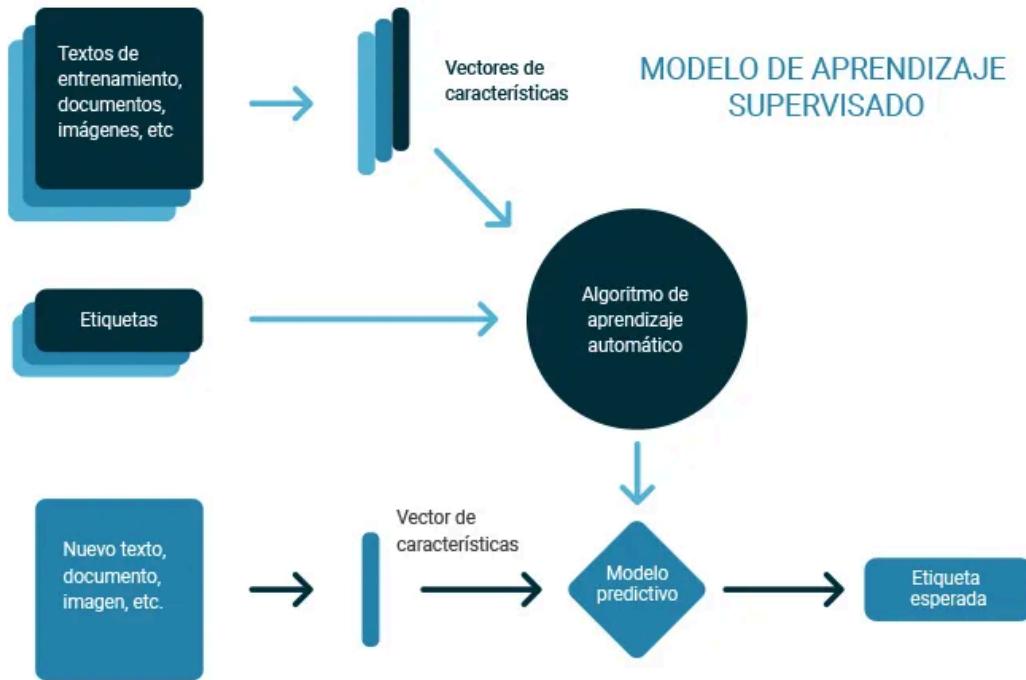


Figura 11: Modelo de aprendizaje supervisado. Fuente: Medium

#### 8.4.2.1 NAIVE BAYES

El algoritmo de Naive Bayes [23] es un clasificador probabilístico basado en el teorema de Bayes, con la suposición de que las características son independientes entre sí, lo cual raramente es cierto en la práctica, pero aún así funciona bien en muchos casos.

El teorema de Bayes establece la relación entre las probabilidades condicionales y marginales de dos eventos A y B.

$$P(A | B) = \frac{P(B | A) * P(A)}{P(B)}$$

Donde:

$P(A | B)$  : es la probabilidad de A dado B (probabilidad posterior).

$P(B | A)$  : es la probabilidad de B dado A (verosimilitud).

$P(A)$  : es la probabilidad de A (probabilidad prior).

$P(B)$  : es la probabilidad de B (evidencia).

En el contexto de clasificación, queremos encontrar la clase más probable  $C_k$  dado un conjunto de características  $x = (x_1, x_2, \dots, x_n)$ . Según el teorema de Bayes:

$$P(C_k | x) = \frac{P(x | C_k) * P(C_k)}{P(x)}$$

Dado que estamos interesados en la clase más probable, podemos ignorar  $P(x)$  porque es constante para todas las clases.

$$P(C_k | x) \approx P(x | C_k) * P(C_k)$$

Aquí es donde entra la suposición de independencia. Suponemos que las características son independientes entre sí dada la clase.

$$P(x | C_k) = P(x_1 | C_k) \cdot P(x_2 | C_k) \cdots P(x_n | C_k)$$

Así, el clasificador Naive Bayes calcula la probabilidad de cada clase  $C_k$  para un vector de características  $x$ .

$$P(C_k | x) \approx P(C_k) \cdot \prod_{i=1}^n P(x_i | C_k)$$

Finalmente, asignamos la clase con la mayor probabilidad posterior.

$$\text{Classe Predicha} = \operatorname{argmax}_k P(C_k) \cdot \prod_{i=1}^n P(x_i | C_k)$$

Del algoritmo de Naive Bayes se pueden extraer variantes como Gaussian Naive Bayes que asume que las características continuas se distribuyen de forma gaussiana (normal). Es útil cuando las características siguen una distribución normal. Aunque también podemos encontrar multinomial Naive Baye utilizado para datos de conteo, como la frecuencia de palabras en un texto. Es común en problemas de clasificación de texto.

Naive Bayes es un algoritmo fácil de implementar y rápido de entrenar que funciona bien con grandes conjuntos de datos y maneja bien la alta dimensionalidad y la colinealidad. A pesar de ello, la suposición de independencia es rara vez cierta en la práctica y la versión básica no maneja bien las características continuas.

Este algoritmo funciona en gran medida en aplicaciones como clasificación de correos electrónicos como spam o no spam, como clasificación de opiniones en positivas o negativas o como predicción de categorías o productos que le pueden interesar a un usuario.

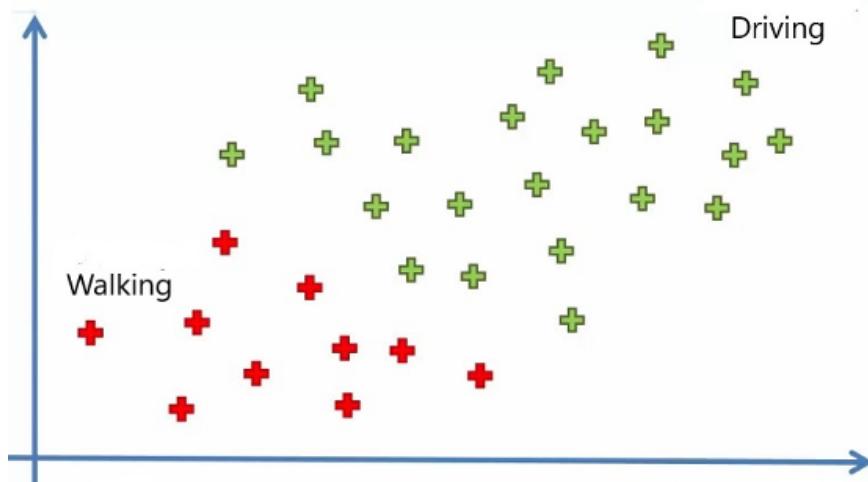


Figura 12: Modelo Naive Bayes. Fuente: Medium

### 8.4.2.2 DECISION TREE

El algoritmo de Árbol de Decisión (Decision Tree) [24] es un método de aprendizaje supervisado que se utiliza tanto para tareas de clasificación como de regresión. Funciona dividiendo el espacio de características en regiones más pequeñas y sencillas, utilizando reglas de decisión que se representan en una estructura de árbol.

El árbol comienza con un nodo raíz que contiene todos los datos de entrenamiento. En cada nodo, el algoritmo selecciona la característica y el punto de división que mejor separa los datos en clases diferentes (para clasificación) o que minimiza el error (para regresión). Esto se hace utilizando criterios como la ganancia de información, el índice Gini, o la varianza. El nodo se divide en dos o más ramas, cada una representando un subconjunto de datos. Este proceso se repite recursivamente para cada rama, creando nuevos nodos y ramas, hasta que se cumplen ciertos criterios de parada (como una profundidad máxima del árbol o un número mínimo de muestras por nodo). Las hojas del árbol (nodos terminales) contienen la predicción final. En clasificación, cada hoja se asigna a la clase más frecuente de los datos en esa hoja. En regresión, cada hoja contiene el valor promedio de los datos en esa hoja.

Para dividir la raíz del árbol, como se menciona anteriormente, se puede utilizar Gini Impurity (Índice Gini) utilizado en clasificación que mide la pureza de una partición ya que una partición pura contiene datos de una sola clase.

$$Gini = 1 - \sum_{i=1}^n p_i^2$$

donde  $p_i^2$  es la proporción de elementos en la clase i

También se puede usar Gain para clasificación, el cual está basado en la entropía, midiendo la cantidad de información que se gana al dividir un nodo.

$$Entropy = - \sum_{i=1}^n p_i \cdot \log_2(p_i)$$

Para regresión se puede usar el criterio de división Mean Squared Error (MSE) que mide la variación de los valores de las características con respecto a la media.

$$MSE = \frac{1}{N} - \sum_{i=1}^N (y_i - \bar{y})^2$$

donde “ $y_i$ ” son los valores reales y “ $\bar{y}$ ” es la media de los valores en el nodo.

Los árboles de decisión son fáciles de entender y visualizar. No necesitan que los datos sean escalados o normalizados. Pueden capturar relaciones no lineales entre las características y la variable objetivo. A pesar de ello los árboles de decisión tienden a sobreajustarse a los datos de entrenamiento si no se podan adecuadamente y unas pequeñas variaciones en los datos pueden dar resultado a árboles muy diferentes.

Este algoritmo funciona en gran medida en aplicaciones de diagnóstico médico clasificando enfermedades en base a síntomas y pruebas, evaluación de riesgos en el crédito de aplicaciones financieras, etc .

## Survival of passengers on the Titanic

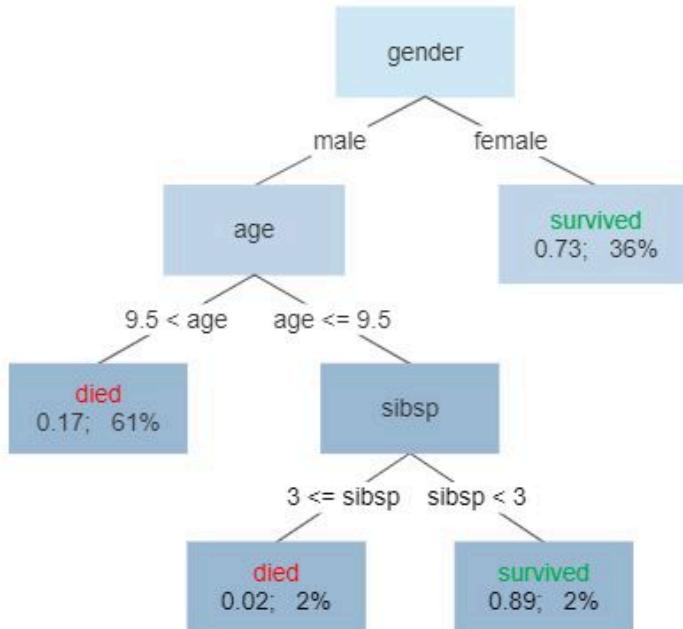


Figura 13: Ejemplo de Árbol de Decisión, Supervivientes del Titanic. Fuente: Wikipedia

### 8.4.2.3 RANDOM FOREST

El algoritmo de Random Forest [25] o Bosque Aleatorio es una extensión del algoritmo de Árbol de Decisión que mejora su precisión y robustez. Random Forest es un método de aprendizaje supervisado que se utiliza tanto para tareas de clasificación como de regresión. Se basa en la idea de crear múltiples árboles de decisión y combinarlos para obtener una predicción más precisa y generalizable.

Random Forest es un método de ensemble learning, lo que significa que combina las predicciones de múltiples modelos para mejorar la precisión y reducir el riesgo de sobreajuste. Utiliza una técnica llamada bagging para generar múltiples subconjuntos de datos de entrenamiento. Cada subconjunto se crea mediante muestreo aleatorio con reemplazo a partir del conjunto de datos original. Para cada subconjunto de datos, se entrena un árbol de decisión. Durante la construcción de cada árbol, se selecciona aleatoriamente un subconjunto de características en cada división (split), lo que añade un nivel adicional de aleatoriedad y diversidad a los árboles. Finalmente, para clasificar, se combinan las predicciones de todos los árboles utilizando la votación mayoritaria. La clase predicha es la que recibe más votos entre todos los árboles. Y para regresión, se combinan las predicciones de todos los árboles calculando el promedio de las predicciones individuales.

Al combinar múltiples árboles de decisión, Random Forest generalmente produce predicciones más precisas que un solo árbol de decisión. Además, al promediar los resultados de muchos árboles, Random Forest tiende a ser menos propenso al sobreajuste en comparación con un solo árbol de decisión por lo que es menos sensible a los datos ruidosos y a las variaciones en el conjunto de datos de entrenamiento. Puede manejar tanto características continuas como categóricas y puede capturar relaciones no lineales entre las características y la variable objetivo.

Las únicas desventajas que tiene Random Forest es que entrenar múltiples árboles de decisión y combinar sus predicciones requiere más tiempo y recursos computacionales que entrenar un solo árbol, y, aunque Random Forest es más preciso, es menos interpretable que un solo árbol de decisión ya que la combinación de muchos árboles hace que sea más difícil entender cómo se tomó una decisión específica.

Este algoritmo funciona, para todos los casos donde Árbol de Decisión es eficaz y además para clasificación y predicción en aplicaciones de procesamiento de lenguaje natural, ya sea voz o texto.

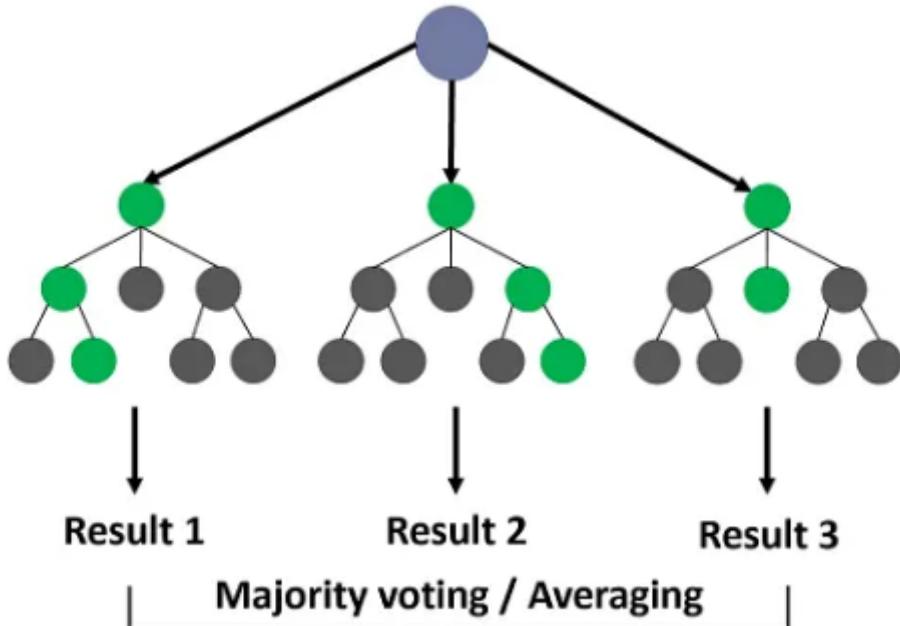


Figura 14: Random Forest. Fuente: Medium

#### 8.4.2.4 GRADIENT BOOSTING MACHINES (GBM)

Las Gradient Boosting Machines (GBM) [26] son una familia de algoritmos de aprendizaje supervisado que se utilizan tanto para clasificación como para regresión. GBM construye modelos de predicción en forma de conjuntos de árboles de decisión, combinando varios modelos débiles para crear un modelo fuerte. Esta técnica es muy eficaz y se ha convertido en una herramienta estándar en competencias de ciencia de datos y aplicaciones del mundo real debido a su capacidad para manejar datos complejos y lograr altas precisiones.

GBM comienza con un modelo inicial, que suele ser un modelo trivial como la media de los valores de la variable objetivo (para regresión) o una predicción uniforme (para clasificación). En cada iteración, se ajusta un nuevo árbol de decisión (o un modelo débil) a los residuos del modelo anterior. Los residuos son las diferencias entre las predicciones del modelo actual y los valores reales. Se ajusta el nuevo modelo a los residuos y se actualiza el modelo total sumando una fracción del nuevo modelo (controlada por un parámetro de aprendizaje). El proceso se repite hasta que se alcanza un número determinado de iteraciones o se minimiza el error ya que el objetivo es reducir el error de predicción de manera iterativa. Finalmente, la predicción es la suma de las predicciones de todos los modelos ajustados.

GBM y sus variantes suelen lograr altas precisiones debido a la capacidad de modelar relaciones complejas. Pueden manejar diferentes tipos de datos y tareas (clasificación y regresión) y las variantes como XGBoost y LightGBM tienen mecanismos integrados para prevenir el sobreajuste.

El entrenamiento de modelos GBM puede ser computacionalmente intensivo y lento y para lograr un buen rendimiento puede requerir un ajuste cuidadoso de los hiperparámetros. A pesar de ser más precisos, los modelos GBM son menos interpretables que los modelos más simples como los árboles de decisión individuales.

Esta familia de algoritmos funciona, para todos los casos anteriormente mencionados pero demostrando una capacidad de predicción mucho mayor y con una mejor robustez en el procesamiento de los datos.

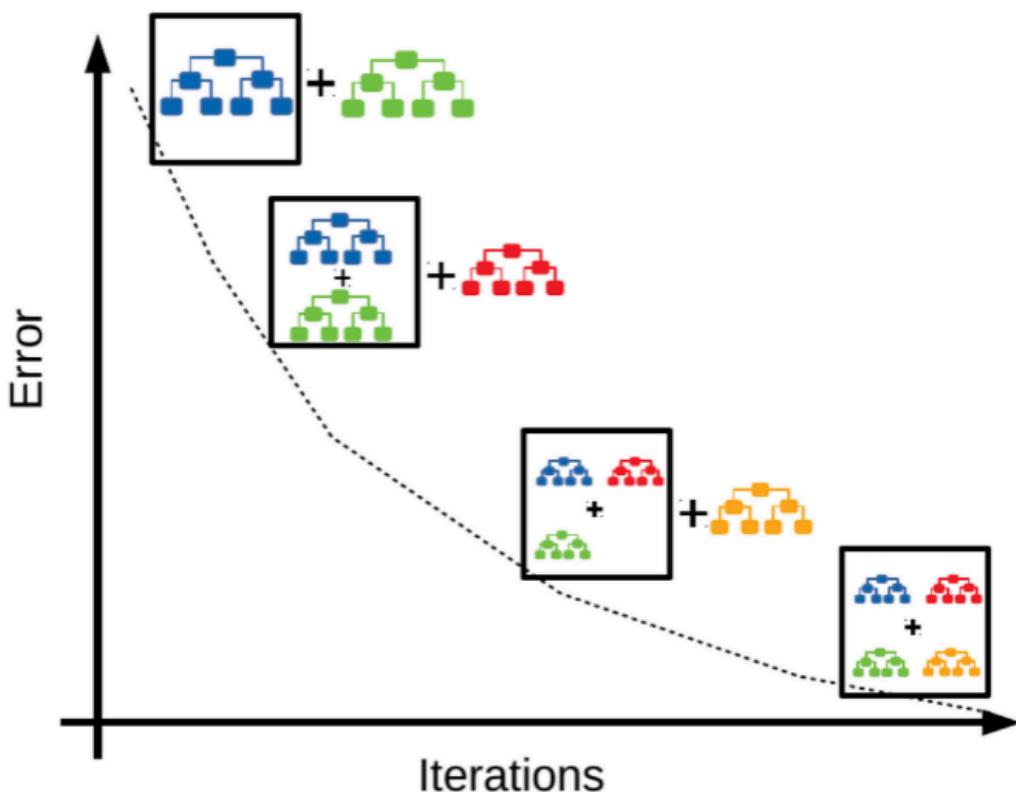


Figura 15: Modelo Gradient Boosting Machine. Fuente: Medium

#### 8.4.2.4.1 XGBOOST

XGBoost es una implementación avanzada de GBM que incluye varias optimizaciones para mejorar la velocidad y el rendimiento.

Esta implementación incluye términos de regularización L1 (Lasso) y L2 (Ridge) para prevenir el sobreajuste.

L1 (Lasso): Agrega una penalización proporcional al valor absoluto de los coeficientes del modelo a la función de pérdida. Esta penalización puede llevar a que algunos coeficientes se vuelvan exactamente cero, lo que resulta en un modelo más simple y en la selección automática de características.

L2 (Ridge): Agrega una penalización proporcional al cuadrado de los coeficientes del modelo a la función de pérdida. Esto no fuerza a los coeficientes a ser exactamente cero, sino que los reduce, distribuyendo la penalización entre todas las características. Mantiene todas las características en el modelo, pero las regula para evitar que cualquier característica tenga una influencia excesiva.

Además XGBoost soporta el procesamiento paralelo, lo que acelera el entrenamiento en comparación con GBM tradicional. Tiene estrategias integradas para manejar datos faltantes y utiliza técnicas para reducir la varianza, como el uso de submuestreo de filas y columnas. Finalmente, construye los árboles en crecimiento nivelado o también conocido como "level-wise growth".

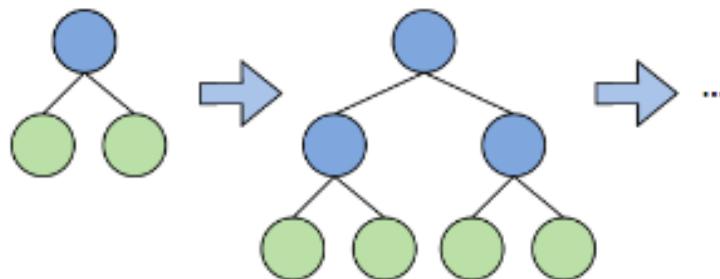


Figura 16: Crecimiento Nivelado. Fuente: Medium

#### 8.4.2.4.2 LIGHTGBM

LightGBM es otra variante de GBM que está optimizada para velocidad y eficiencia en memoria.

Esta implementación utiliza histogramas para discretizar valores continuos, lo que acelera el cálculo de ganancias de división. Construye los árboles en crecimiento foliar o también conocido como "leaf-wise" en lugar de "level-wise", lo que puede mejorar la precisión. Está diseñado para manejar grandes conjuntos de datos y tiene menor uso de memoria.

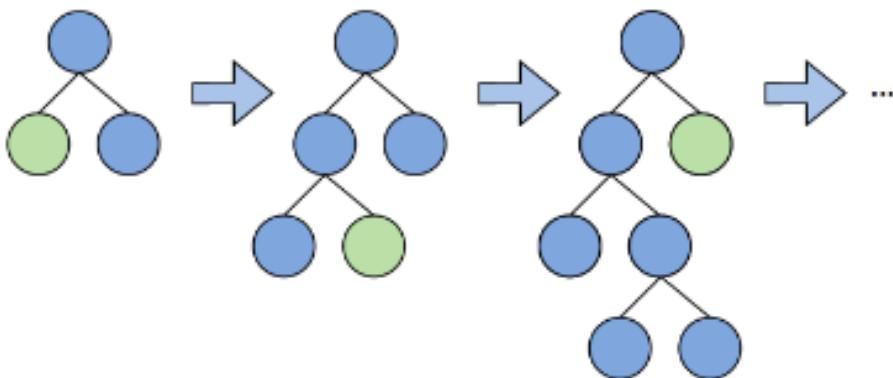


Figura 17: Crecimiento Foliar. Fuente: Medium

#### 8.4.2.4.3 CATBOOST

CatBoost es una variante de GBM especialmente diseñada para manejar características categóricas de manera eficiente.

En muchos algoritmos de aprendizaje automático, las características categóricas deben transformarse antes de poder ser usadas. Esto generalmente se hace mediante técnicas como one-hot encoding, que convierte cada categoría en una columna binaria separada. Sin embargo, esto puede resultar en conjuntos de datos muy grandes y es ineficiente.

Además, es capaz de manejar las características categóricas de manera nativa, sin necesidad de preprocesamiento adicional. Utiliza un método innovador que crea combinaciones de categorías y asigna a cada una un valor basado en la frecuencia de ocurrencia, preservando la información de las categorías de manera más compacta y eficiente.

La mayoría de los algoritmos de GBM construyen árboles de decisión de manera greedy, evaluando todas las posibles divisiones de los datos en cada nodo del árbol. CatBoost introduce un algoritmo de orden para evaluar estas divisiones, lo que mejora la precisión del modelo y reduce el riesgo de sobreajuste. Este método ordena y procesa las características de manera que cada división sea evaluada de forma más robusta y confiable, mejorando la calidad de las predicciones.

En muchos conjuntos de datos, las clases no están distribuidas uniformemente; algunas clases pueden ser mucho más comunes que otras. CatBoost incorpora técnicas para manejar este desbalance de clases de manera efectiva. Esto incluye la asignación de pesos diferentes a las clases menos representadas durante el entrenamiento, asegurando que el modelo preste atención adecuada a todas las clases y no se sesgue hacia las más comunes.

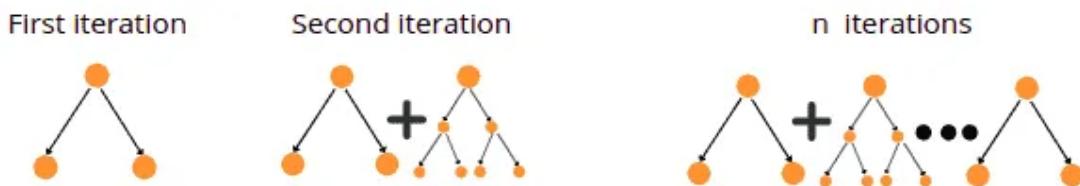


Figura 18: Crecimiento de árbol en CatBoost. Fuente: HandsOnCloud

#### 8.4.2.5 REDES NEURONALES

Las Redes Neuronales son modelos de aprendizaje automático inspirados en la estructura y el funcionamiento del cerebro humano. Están diseñadas para reconocer patrones y aprender de los datos a través de un proceso de entrenamiento. Las redes neuronales se utilizan para una amplia variedad de tareas, desde la clasificación de imágenes y el reconocimiento de voz hasta la predicción de series temporales y la generación de texto.

### 8.4.2.5.1 ARTIFICIALES (ANN)

Las Redes Neuronales Artificiales (ANN) [27] son el tipo más básico de redes neuronales. Cada Red Neuronal ANN contiene:

*Neurona Artificial:* Es la unidad básica de una ANN, similar a una neurona biológica. Cada neurona recibe varias entradas, realiza una operación matemática con ellas y produce una salida.

*Capas de Neuronas:* Las ANN están formadas por capas de neuronas:

- La capa de entrada recibe los datos de entrada.
- Las capas ocultas, puede haber una o más y, procesan la información.
- La capa de salida que produce el resultado final del modelo.

*Pesos y Sesgos:* Cada conexión entre neuronas tiene un peso asociado, que es un valor que se ajusta durante el entrenamiento. Además, las neuronas también tienen un sesgo, que es otro valor ajustable.

*Función de Activación:* Es una función matemática que se aplica a la salida de una neurona para introducir no linealidad, permitiendo que la red aprenda representaciones más complejas. Ejemplos comunes son:

- *ReLU (Rectified Linear Unit):* ReLU es una función de activación ampliamente utilizada en redes neuronales profundas. La función ReLU es definida como:

$$\text{ReLU} = \max(0, x)$$

En otras palabras, si la entrada  $x$  es mayor que 0, la salida es  $x$ ; de lo contrario, la salida es 0.

- *Sigmoide:* La función sigmoide es otra función de activación que mapea cualquier valor de entrada a un rango entre 0 y 1. La función sigmoide es definida como:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Ambas funciones de activación tienen sus propias ventajas y desventajas, y la elección entre ellas depende del problema específico y de la arquitectura de la red neuronal.

Las redes neuronales ANN se inicializan asignando pesos y sesgos iniciales aleatorios a las conexiones entre neuronas. Los datos de entrada se pasan a través de la red capa por capa.

En la capa de entrada, cada neurona recibe un valor de una característica del conjunto de datos. En las capas ocultas, cada neurona realiza una suma ponderada de sus entradas (usando los pesos y el sesgo) y aplica la función de activación para producir una salida. En la capa de salida, el proceso es similar, y la salida de esta capa es la predicción del modelo.

A continuación se compara la predicción de la red con el valor real (la etiqueta en el caso de clasificación o el valor en el caso de regresión) para calcular el error. Una función de pérdida mide este error.

El error se propaga de vuelta a través de la red, ajustando los pesos y sesgos para minimizar el error. Esto se hace usando el descenso de gradiente, una técnica matemática que ajusta los pesos y sesgos para reducir gradualmente el error.

El proceso de propagación hacia adelante y hacia atrás se repite para muchas iteraciones (épocas) hasta que el error sea suficientemente pequeño o hasta que se alcance un número máximo de iteraciones.

Las ANN pueden aprender y modelar relaciones complejas entre entradas y salidas, y pueden ser aplicadas a una amplia variedad de problemas, incluyendo clasificación, regresión, reconocimiento de patrones y más.

A pesar de la eficacia que tiene el uso de ANN, estas requieren de una gran cantidad de datos para entrenar de manera efectiva, el entrenamiento puede ser lento y requerir mucho poder computacional y pueden sobreajustarse a los datos de entrenamiento si no se regulan adecuadamente.

Este tipo de algoritmo de red neuronal es eficaz para casos de clasificación de imágenes, detección de objetos, segmentación de imágenes, reconocimiento de escritura, reconocimiento facial, etc.

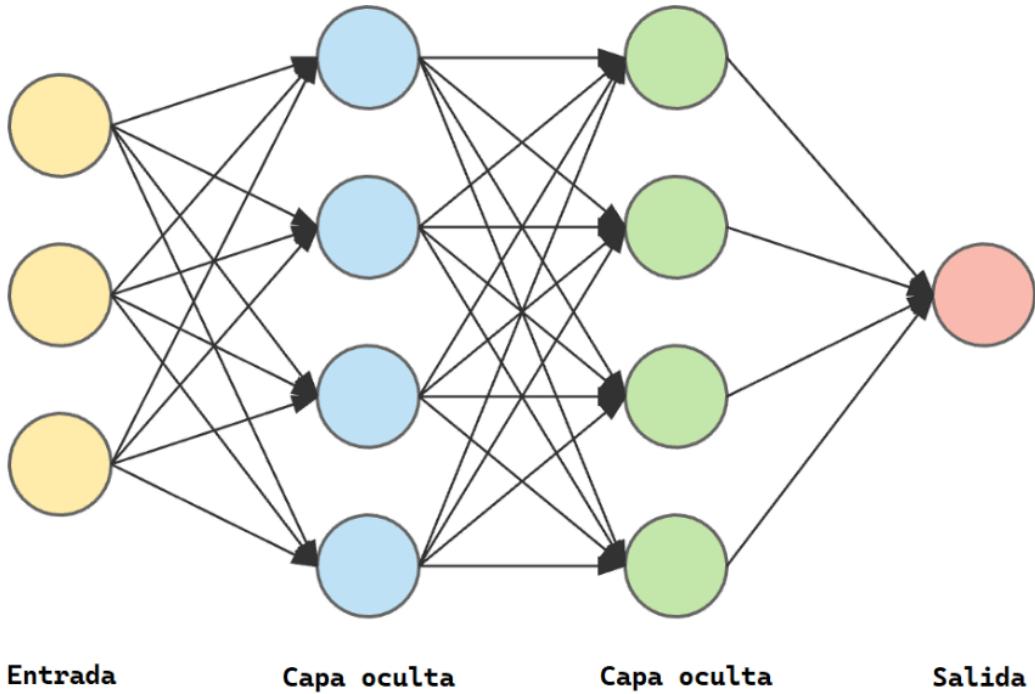


Figura 19: Red Neuronal Artificial. Fuente: OpenWebinars

### 8.4.2.5.2 RECURRENTES (RNN)

Las Redes Neuronales Recurrentes (RNN) [28] son un tipo de red neuronal diseñada específicamente para trabajar con datos secuenciales o temporales. A diferencia de las Redes Neuronales Artificiales (ANN) estándar, las RNN tienen una estructura que permite conservar información a lo largo del tiempo, lo que las hace ideales para tareas donde el contexto temporal es importante, como el procesamiento de lenguaje natural y la predicción de series temporales. Cada Red Neuronal RNN contiene:

*Neurona Recurrente:* En una RNN, cada neurona no solo recibe entradas de la capa anterior, sino que también recibe una entrada de su propio estado anterior. Esto crea un bucle recurrente que permite que la red conserve información de pasos anteriores en el tiempo.

*Estados Ocultos:* Los estados ocultos son vectores que contienen la información acumulada de la secuencia hasta el momento actual. Se actualizan en cada paso temporal.

Las redes neuronales RNN se inicializan asignando los pesos y los estados ocultos iniciales aleatoriamente.

En el paso temporal  $t$ , la RNN recibe una entrada  $x_t$  y produce una salida  $y_t$ . El estado oculto  $h_t$  se calcula usando una función que combina la entrada actual  $x_t$  y el estado oculto anterior  $h_{t-1}$ . La salida  $y_t$  se calcula usando una función de activación aplicada al estado oculto  $h_t$ .

$$h_t = \sigma(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

$$y_t = \phi(W_{hy}h_t + b_y)$$

Donde  $\sigma$  es una función de activación,  $W_{hh}$ ,  $W_{xh}$ , y  $W_{hy}$  son matrices de pesos, y  $b_h$  y  $b_y$  son sesgos.

El error se calcula en cada paso temporal y se propaga de vuelta a través de la secuencia, ajustando los pesos en cada paso temporal. La acción de volver a través de la secuencia maneja las dependencias temporales y ajusta los pesos para minimizar el error global en la secuencia.

Las RNN son muy efectivas para modelar datos secuenciales y temporales y pueden aplicarse a una amplia variedad de tareas que involucran secuencias.

Las RNN pueden sufrir problemas de desvanecimiento y explosión del gradiente, lo que dificulta el entrenamiento y requieren mucho tiempo de entrenamiento especialmente para secuencias largas y complejas.

Este tipo de algoritmo de red neuronal es eficaz para casos de traducción automática, generación de texto, análisis de sentimientos, pronósticos financieros, meteorológicos, conversión de voz a texto, asistentes virtuales, etc.

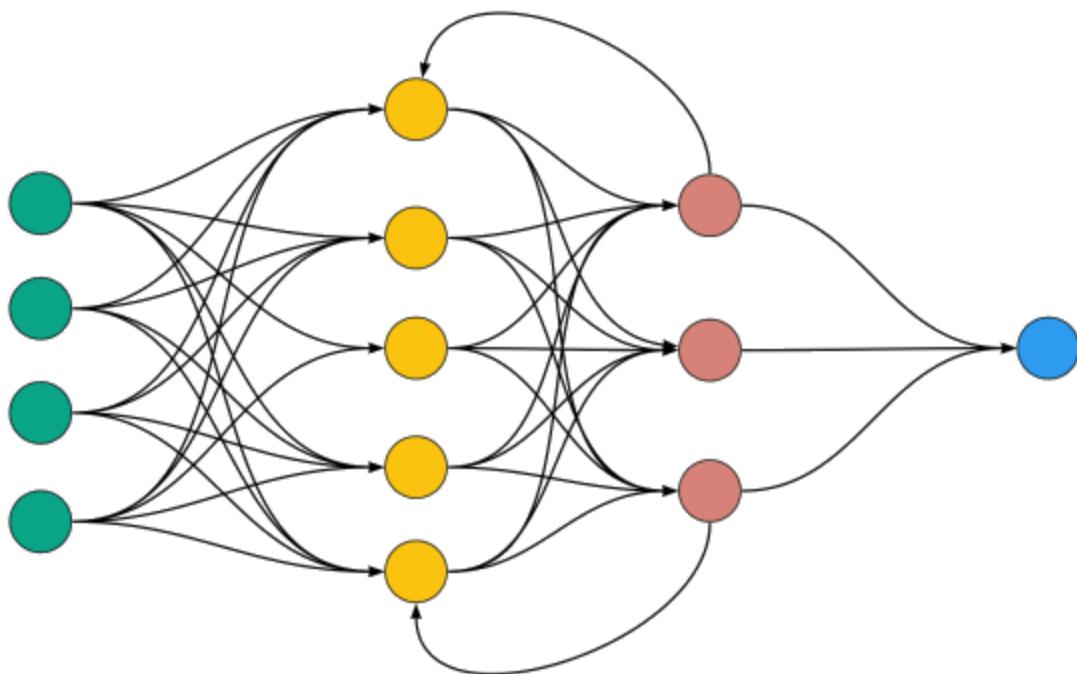


Figura 20: Red Neuronal Recurrente. Fuente: OpenWebinars

#### 8.4.3 REFORZADOS

Los algoritmos de aprendizaje por refuerzo son un tipo de modelo de inteligencia artificial que permite a los modelos aprender a tomar decisiones óptimas en un entorno complejo a través de la interacción y ensayo-error. A diferencia del aprendizaje supervisado, donde se le proporciona al agente un conjunto de datos de entrada y salida, en el aprendizaje por refuerzo el agente no tiene conocimiento previo de la acción óptima. En cambio, aprende a través de la experiencia, explorando el entorno y recibiendo recompensas o penalizaciones por sus acciones.



Figura 21: Modelo de aprendizaje reforzado. Fuente: Medium

#### 8.4.4 SELECCIÓN DE UN ALGORITMO

Elegir el algoritmo de aprendizaje automático adecuado para el proyecto es crucial para obtener resultados exitosos. Diversos factores deben ser evaluados cuidadosamente para seleccionar el algoritmo que mejor se adapte a las necesidades específicas del proyecto.

En función de la tarea que deba realizar el modelo de inteligencia artificial, se pueden elegir algoritmos de clasificación si el objetivo es predecir una categoría, o de regresión si el objetivo es predecir un valor numérico continuo. Si el objetivo es agrupar datos sin etiquetas, podemos escoger modelos de agrupación, o si queremos identificar patrones inusuales, utilizar un modelo de detección de anomalías.

Un modelo de machine learning también se puede escoger en función del tipo, volumen y calidad de los datos. Los algoritmos de ML pueden trabajar con diferentes tipos de datos, como números enteros, números reales, texto o imágenes. Es importante elegir un algoritmo compatible con el tipo de datos del proyecto. Algunos algoritmos, como las redes neuronales profundas, requieren grandes conjuntos de datos para entrenar de manera efectiva, mientras que otros, como los árboles de decisión, pueden funcionar bien con conjuntos de datos más pequeños.

Se deben considerar los recursos computacionales a la hora de elegir un modelo, puesto que algunos algoritmos, como las redes neuronales profundas, requieren un poder de cómputo significativo para entrenar. El tiempo que se tarda en entrenar un modelo puede variar significativamente entre algoritmos.

Características	Aprendizaje supervisado	Aprendizaje no supervisado	Aprendizaje reforzado
Datos de entrenamiento	Etiquetados	No etiquetados	Interacción con el entorno
Objetivo	Predecir una salida basada en ejemplos	Encontrar estructuras/patrones	Maximizar recompensas acumuladas
Ejemplos de algoritmos	Regresión Lineal, SVM, Redes Neuronales	k-Means, PCA, SOM	Q-Learning, SARSA
Aplicaciones	Clasificación, Regresión	Agrupación, Reducción de Dimensionalidad	Juegos, Control Robótico

Tabla 10: Características de los tipos de aprendizaje. Fuente: Elaboración propia

#### 8.5 ENSAMBLE DE MODELOS

Los ensambles de modelos son técnicas poderosas en aprendizaje automático que combinan múltiples modelos base para mejorar el rendimiento predictivo y la generalización del modelo. Aquí vamos a explorar tres enfoques de ensamble comunes: Majority Voting, Bagging y Boosting (Adaboost), centrándonos en su aplicación en los modelos de árboles de decisión y en particular en el algoritmo de Random Forest.

### 8.5.1 MAJORITY VOTING

La técnica de Majority Voting se basa en el principio de la sabiduría de la multitud o sabiduría colectiva, que sugiere que la opinión de un grupo de individuos es más confiable que la de un solo individuo. En el contexto del aprendizaje automático y la clasificación, este principio se aplica al combinar las predicciones de múltiples modelos independientes para tomar una decisión final.

En el aprendizaje automático, el objetivo es construir modelos que puedan generalizar patrones a partir de datos y hacer predicciones precisas sobre datos nuevos y no vistos. Sin embargo, ningún modelo es perfecto y puede cometer errores debido a limitaciones en los datos de entrenamiento, la complejidad del problema o el diseño del modelo.

Aquí es donde entran en juego las técnicas de ensamble como voto por mayoría o Majority Voting. Al combinar múltiples modelos independientes, cada uno con sus propias fortalezas y debilidades, el ensamble puede capitalizar la diversidad de opiniones y suavizar los errores individuales. Esta estrategia se alinea con el principio de que la combinación de múltiples opiniones independientes tiende a ser más precisa y confiable que la opinión de un solo modelo.

En el caso de Majority Voting, se lleva a cabo una "votación" entre los clasificadores base para determinar la clase predicha para una instancia dada. La clase más común entre los clasificadores base se selecciona como la predicción final del ensamble. Esta técnica es simple pero efectiva y se puede aplicar con facilidad a una amplia gama de problemas de clasificación.

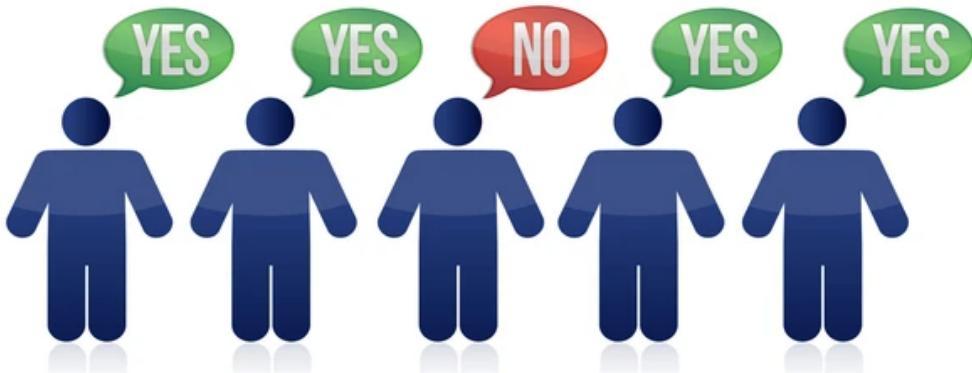


Figura 22: Ejemplo votación por mayoría. Fuente: NSOSSR

### 8.5.2 BAGGING

Esta técnica se basa en la idea de que la combinación de múltiples modelos débiles puede producir un modelo más fuerte y robusto.

El proceso de Bagging [29] comienza dividiendo el conjunto de datos original en múltiples subconjuntos, cada uno generado mediante el muestreo aleatorio con reemplazo, lo que significa que algunas instancias pueden aparecer en varios subconjuntos mientras que otras pueden quedar excluidas. Este método de muestreo permite capturar diferentes perspectivas de los datos en cada subconjunto y, al mismo tiempo, introduce variabilidad en los modelos resultantes.

Una vez que se han creado los subconjuntos, se entrena un modelo base en cada uno de ellos de forma independiente. Estos modelos pueden ser del mismo tipo o de diferentes tipos, lo que agrega diversidad al ensamble. En el caso de la regresión, las predicciones de todos los modelos base se promedian para producir la predicción final del ensamble, mientras que en la clasificación se utiliza una votación por mayoría para determinar la clase final.

La ventaja principal de Bagging radica en su capacidad para reducir la varianza del modelo, lo que ayuda a prevenir el sobreajuste. Al promediar las predicciones de múltiples modelos entrenados en diferentes subconjuntos de datos, Bagging suaviza los errores individuales de cada modelo y produce una predicción más robusta y generalizable.

Además, Bagging es una técnica relativamente sencilla de implementar y puede utilizarse con una variedad de algoritmos de aprendizaje automático. Es especialmente efectiva cuando se aplica a modelos propensos al sobreajuste, como los árboles de decisión, ya que ayuda a mitigar este problema y mejorar el rendimiento del modelo en datos nuevos y no vistos.

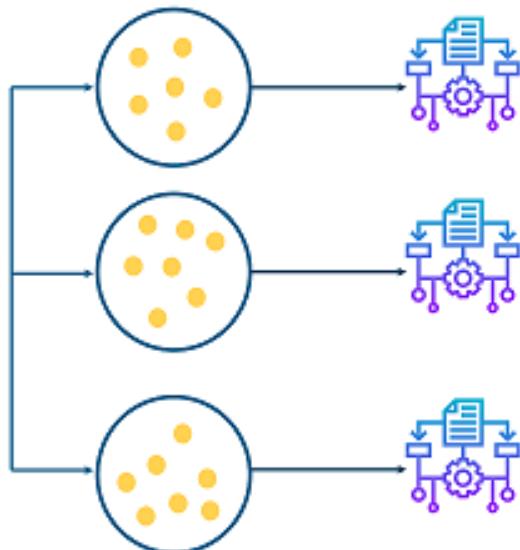


Figura 23: Bagging. Fuente: Machine learning para todos

### 8.5.3 BOOSTING (ADABOOST)

Boosting es una técnica avanzada de ensamble en el campo del aprendizaje automático que busca mejorar la precisión y robustez de los modelos predictivos mediante la construcción secuencial de una serie de modelos base. A diferencia de otros métodos de ensamble como Bagging, donde los modelos base se entranan de forma independiente, en Boosting, los modelos se construyen de manera iterativa, con cada nuevo modelo enfocado en corregir los errores de sus predecesores. Uno de los algoritmos de Boosting más conocidos y ampliamente utilizados es Adaboost (Adaptive Boosting).

La esencia de Boosting radica en su capacidad para mejorar progresivamente el rendimiento del modelo, centrándose especialmente en las instancias más difíciles de clasificar. Esto se logra mediante el ajuste de los pesos de las instancias del conjunto de datos en cada iteración del proceso de entrenamiento, dando más importancia a las instancias que han sido mal clasificadas por los modelos anteriores.

Esta estrategia permite que los modelos subsiguientes se centren más en las instancias difíciles de clasificar, mejorando así gradualmente el rendimiento general del ensamble.

El proceso de Boosting comienza con la asignación de pesos iniciales a cada instancia del conjunto de datos. Estos pesos pueden ser iguales para todas las instancias o pueden inicializarse de manera diferente, dependiendo de la implementación específica del algoritmo de Boosting. Luego, se entrena un modelo base inicial utilizando todo el conjunto de datos. Después de cada iteración subsiguiente, se entrena un nuevo modelo base utilizando el mismo conjunto de datos, pero con un enfoque diferente, dando más peso a las instancias mal clasificadas por los modelos anteriores.

Después de entrenar cada modelo base, se ajustan los pesos de las instancias del conjunto de datos para dar más peso a las instancias mal clasificadas y menos peso a las instancias correctamente clasificadas. Este proceso de ajuste de pesos se repite en cada iteración del proceso de Boosting, lo que permite que los modelos subsiguientes se concentren más en las instancias difíciles de clasificar.

Una vez que se han entrenado todos los modelos base, las predicciones de cada modelo se combinan utilizando algún método de agregación, como una votación ponderada. Esto produce la predicción final del ensamble, que es una combinación de las predicciones de todos los modelos base.

Adaboost, en particular, es conocido por su efectividad para mejorar la precisión del modelo y puede utilizarse con una variedad de algoritmos base. Sin embargo, es especialmente efectivo cuando se combina con árboles de decisión débiles, lo que ayuda a compensar su debilidad inherente al enfocarse en las instancias más difíciles de clasificar.

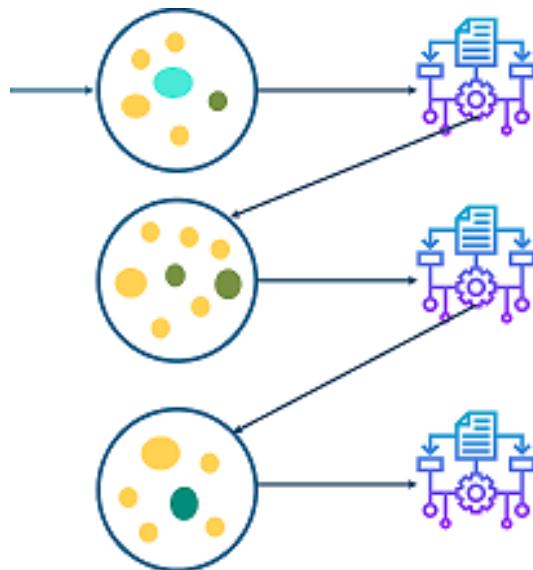


Figura 24: Boosting. Fuente: Machine learning para todos

#### 8.5.4 CONCLUSIONES

Las técnicas de ensamble de modelos, como Bagging, Boosting y Majority Voting, se han consolidado como estrategias fundamentales en el ámbito del aprendizaje automático debido a su capacidad para mejorar significativamente el rendimiento de los modelos predictivos. Estas técnicas representan un enfoque poderoso y efectivo que capitaliza la diversidad y la sabiduría colectiva de múltiples modelos base para producir modelos finales que son más precisos, robustos y generalizables.

En un mundo cada vez más complejo y dinámico, donde los conjuntos de datos pueden ser enormes y las relaciones subyacentes entre las variables pueden ser sútiles y multifacéticas, los modelos individuales pueden no capturar todas las complejidades de los datos. Aquí es donde entran en juego las técnicas de ensamble. Al combinar múltiples modelos, estas técnicas pueden compensar las limitaciones individuales de cada modelo base y producir una solución final que sea más poderosa que la suma de sus partes.

Bagging, por ejemplo, se destaca por su capacidad para reducir la varianza del modelo al promediar las predicciones de múltiples modelos entrenados en diferentes subconjuntos de datos. Esto no solo ayuda a prevenir el sobreajuste, sino que también produce modelos más estables y generalizables que pueden hacer frente a una variedad de escenarios y condiciones.

Por otro lado, Boosting se enfoca en mejorar progresivamente el rendimiento del modelo mediante la construcción secuencial de una serie de modelos base. Al ajustar los pesos de las instancias del conjunto de datos en cada iteración del proceso de entrenamiento, Boosting se centra en corregir los errores de los modelos anteriores, especialmente en las instancias más difíciles de clasificar. Esto resulta en modelos finales que son altamente adaptables y precisos, particularmente cuando se combinan con algoritmos base débiles, como los árboles de decisión.

Además, Majority Voting aprovecha la sabiduría colectiva de múltiples modelos independientes para mejorar la precisión y la robustez del ensamble. Al combinar las predicciones de varios modelos base, Majority Voting puede suavizar los errores individuales y producir predicciones más precisas y confiables en general. Esto es especialmente útil cuando los clasificadores base tienen diferentes sesgos y errores, ya que la diversidad de opiniones puede ayudar a compensar estas discrepancias y mejorar la precisión del ensamble.

Al aprovechar la diversidad y la sabiduría colectiva de múltiples modelos base, estas técnicas pueden producir modelos finales que son más robustos, precisos y generalizables, lo que los convierte en herramientas esenciales para una amplia gama de aplicaciones en ciencia de datos, inteligencia artificial y más allá.

#### 8.6 MÉTRICAS DE RENDIMIENTO

Las métricas de rendimiento son fundamentales en la evaluación de modelos de clasificación, ya que ofrecen una comprensión detallada y cuantitativa de su eficacia en la tarea de clasificación. Estas métricas proporcionan información crucial sobre la calidad de las predicciones del modelo, su capacidad para distinguir entre diferentes clases y su generalización a nuevos datos.

Al comprender y analizar estas métricas, podemos tomar decisiones fundamentadas sobre qué modelo es más adecuado para una tarea específica, así como identificar áreas de mejora y optimización.

Además, estas métricas son esenciales para la comparación y la evaluación objetiva de diferentes enfoques de modelado, lo que contribuye al avance y la innovación en el campo del aprendizaje automático y la inteligencia artificial.

En este apartado, exploramos diversas métricas que son comúnmente utilizadas para medir el rendimiento de los modelos de clasificación. Desde la clásica matriz de confusión, que desglosa las predicciones del modelo en verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, hasta medidas más avanzadas como la precisión, el recall, el F1-Score y la curva ROC. [30]

### 8.6.1 MATRIZ DE CONFUSIÓN

La matriz de confusión es una herramienta fundamental en la evaluación de modelos de clasificación, ya que proporciona una visión detallada y desglosada del rendimiento del modelo en la tarea de clasificación. Consiste en una tabla que resume los resultados de las predicciones del modelo.

Esta matriz se organiza en filas y columnas. Las filas representan las clases reales o verdaderas, es decir, las categorías a las que pertenecen las instancias en el conjunto de datos. Mientras tanto, las columnas representan las clases predichas por el modelo, es decir, las categorías a las que el modelo asigna las instancias después de realizar las predicciones.

**Verdaderos Positivos:** Representa las instancias que fueron clasificadas correctamente como positivas por el modelo. Es decir, son las instancias que pertenecen a una clase determinada y fueron correctamente identificadas como tal por el modelo.

**Verdaderos Negativos:** Representa las instancias que fueron clasificadas correctamente como negativas por el modelo. Es decir, son las instancias que no pertenecen a una clase determinada y fueron correctamente identificadas como tal por el modelo.

**Falsos Positivos:** Representa las instancias que fueron clasificadas incorrectamente como positivas por el modelo. Es decir, son las instancias que no pertenecen a una clase determinada, pero fueron incorrectamente identificadas como pertenecientes a esa clase por el modelo.

**Falsos Negativos:** Representa las instancias que fueron clasificadas incorrectamente como negativas por el modelo. Es decir, son las instancias que pertenecen a una clase determinada, pero fueron incorrectamente identificadas como no pertenecientes a esa clase por el modelo.

		Predicción		
Actual			Positivo	Negativo
	Positivo	Verdaderos Positivos		Falsos Negativos
	Negativo	Falsos Positivos		Verdaderos Negativos
		$\text{dato real} = 1$ $\text{dato predicho} = 1$	$\text{dato real} = 0$ $\text{dato predicho} = 1$	$\text{dato real} = 1$ $\text{dato predicho} = 0$
				$\text{dato real} = 0$ $\text{dato predicho} = 0$

Figura 25: Matriz de Confusión. Fuente: Aprendeia

## 8.6.2 ACCURACY

La exactitud, también conocida como "accuracy" en inglés, es una métrica fundamental en la evaluación de modelos de clasificación, ya que proporciona una medida general de qué tan bien el modelo está realizando sus predicciones. Esta métrica se calcula dividiendo el número total de predicciones correctas (verdaderos positivos más verdaderos negativos) por el número total de predicciones realizadas.

Matemáticamente, la fórmula para calcular la exactitud es la siguiente:

$$ACCURACY = \frac{VERDADEROS POSITIVOS + VERDADEROS NEGATIVOS}{TOTAL DE PREDICCIONES}$$

En otras palabras, la exactitud nos dice cuántas de todas las instancias fueron clasificadas correctamente por el modelo, independientemente de si fueron clasificadas como positivas o negativas. Es una medida simple y directa que proporciona una visión general del rendimiento del modelo.

Por ejemplo, si tenemos un modelo de clasificación que ha realizado 100 predicciones, de las cuales 85 son correctas (50 verdaderos positivos y 35 verdaderos negativos), la exactitud del modelo sería:

$$ACCURACY = \frac{50 + 35}{100} = \frac{85}{100} = 0,85$$

Esto significa que el modelo tiene una exactitud del 85%, lo que indica que el 85% de todas las predicciones realizadas por el modelo son correctas.

Sin embargo, es importante tener en cuenta que la exactitud puede no ser la métrica más adecuada en todos los casos, especialmente cuando las clases están desbalanceadas o cuando hay costos asimétricos asociados a los errores de clasificación. En tales casos, puede ser necesario considerar otras métricas de rendimiento además de la exactitud.

## 8.6.3 RECALL

El recall, también conocido como "sensibilidad" o "recuperación", es una métrica crucial en la evaluación de modelos de clasificación, especialmente en situaciones donde es importante identificar correctamente todas las instancias positivas en el conjunto de datos. El recall mide la proporción de verdaderos positivos que fueron correctamente identificados por el modelo, respecto al total de verdaderos positivos reales en el conjunto de datos.

Matemáticamente, el recall se calcula utilizando la siguiente fórmula:

$$RECALL = \frac{VERDADEROS POSITIVOS}{VERDADEROS POSITIVOS + FALSOS NEGATIVOS}$$

En otras palabras, el recall nos dice qué tan bien el modelo está identificando correctamente todas las instancias que son realmente positivas en el conjunto de datos. Es decir, cuántas de las instancias positivas reales logró identificar correctamente el modelo.

Por ejemplo, si tenemos un modelo de detección de spam que identifica 90 correos electrónicos como spam de un total de 100 correos electrónicos que son realmente spam, entonces el recall del modelo sería:

$$RECALL = \frac{90}{90+10} = \frac{90}{100} = 0,9$$

Esto indica que el modelo tiene un recall del 90%, lo que significa que es capaz de identificar correctamente el 90% de todos los correos electrónicos que son realmente spam en el conjunto de datos.

El recall es una medida importante para evaluar la capacidad del modelo para detectar correctamente las instancias positivas en el conjunto de datos y es especialmente útil en casos donde la omisión de instancias positivas puede tener consecuencias significativas.

#### 8.6.4 PRECISIÓN

La precisión es una métrica crucial en la evaluación de modelos de clasificación, ya que proporciona información sobre la calidad de las predicciones positivas realizadas por el modelo. Esta métrica mide la proporción de verdaderos positivos que fueron correctamente identificados por el modelo sobre el total de predicciones positivas realizadas por el modelo.

Matemáticamente, la precisión se calcula utilizando la siguiente fórmula:

$$PRECISION = \frac{VERDADEROS POSITIVOS}{VERDADEROS POSITIVOS + FALSOS POSITIVOS}$$

En otras palabras, la precisión nos dice cuántas de las instancias predichas como positivas por el modelo realmente son positivas. Es una medida de la precisión de las predicciones positivas del modelo, es decir, qué tan precisas son las predicciones positivas en relación con todas las predicciones positivas realizadas por el modelo.

Por ejemplo, si tenemos un modelo de detección de enfermedades que predice que 80 pacientes tienen una enfermedad, de los cuales 70 realmente tienen la enfermedad y 10 no la tienen, entonces la precisión del modelo sería:

$$PRECISION = \frac{70}{70+10} = \frac{70}{80} = 0,875$$

Esto significa que el modelo tiene una precisión del 87.5%, lo que indica que el 87.5% de todas las predicciones positivas realizadas por el modelo son realmente correctas.

La precisión es una medida importante para evaluar la calidad de las predicciones positivas del modelo y es especialmente útil cuando se desea minimizar los falsos positivos.

#### 8.6.5 CURVA DE PRECISIÓN-RECALL

La relación entre precisión y recall se puede visualizar a menudo en un gráfico llamado "curva de precisión-recall". Esta curva traza la precisión en el eje y (vertical) y el recall en el eje x (horizontal). Cada punto en la curva representa un umbral de clasificación diferente utilizado por el modelo para hacer predicciones.

En general, a medida que aumentamos el umbral de clasificación (es decir, hacemos que el modelo sea más selectivo al hacer predicciones positivas), la precisión tiende a aumentar mientras que el recall tiende a disminuir. Esto se debe a que al establecer un umbral más alto, el modelo se vuelve más exigente y solo predice como positivos aquellos casos en los que está más seguro, lo que reduce el número de falsos positivos pero también puede reducir el número de verdaderos positivos.

Por otro lado, si disminuimos el umbral de clasificación (es decir, hacemos que el modelo sea menos selectivo al hacer predicciones positivas), la precisión tiende a disminuir mientras que el recall tiende a aumentar. Esto se debe a que al establecer un umbral más bajo, el modelo predice más casos como positivos, lo que aumenta tanto los verdaderos positivos como los falsos positivos.

En la curva de precisión-recall, idealmente queremos un punto que esté lo más cerca posible de la esquina superior derecha, lo que indica alta precisión y alto recall. Sin embargo, en la práctica, existe un compromiso entre precisión y recall: mejorar uno puede implicar empeorar el otro. Por lo tanto, es importante comprender cómo varían entre sí y encontrar un equilibrio que sea óptimo para la aplicación específica.

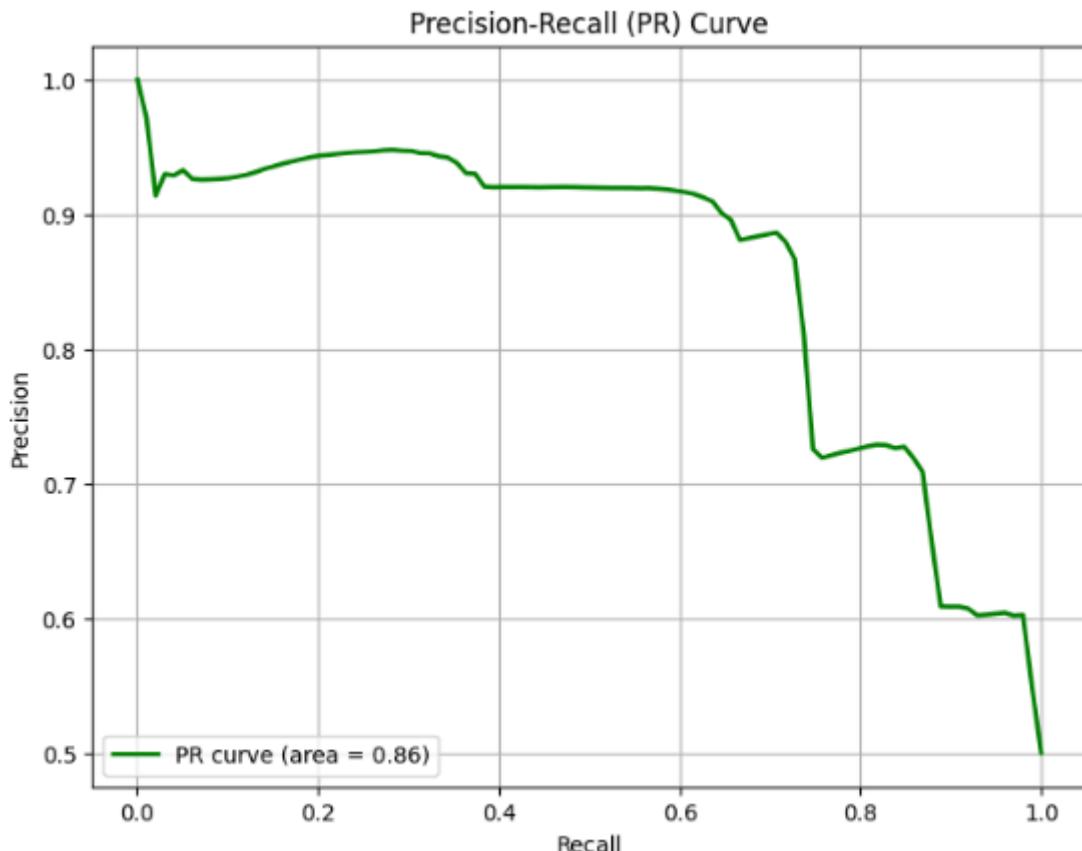


Figura 26: Curva de Precision-Recall. Fuente: Elaboración propia.

### 8.6.6 CURVA ROC

La curva ROC (Receiver Operating Characteristic) es una herramienta fundamental en la evaluación de modelos de clasificación, especialmente cuando se trata de problemas de clasificación binaria. Esta curva es una representación gráfica del rendimiento del modelo en diferentes umbrales de discriminación.

La curva ROC traza la tasa de verdaderos positivos (recall) en el eje y (vertical) y la tasa de falsos positivos en el eje x (horizontal). Cada punto en la curva representa un umbral de clasificación diferente utilizado por el modelo para hacer predicciones.

En la curva ROC, un modelo perfecto que clasifica todas las instancias correctamente tendría un área bajo la curva (AUC) de 1. Por otro lado, un modelo que clasifica aleatoriamente tendría un AUC de 0.5, lo que se traduce en una curva diagonal desde el origen.

La curva ROC ayuda a comprender cómo cambia la tasa de verdaderos positivos en función de la tasa de falsos positivos, lo que permite evaluar la capacidad de discriminación del modelo. En general, queremos que la curva ROC esté lo más cerca posible del rincón superior izquierdo del gráfico, lo que indica una alta tasa de verdaderos positivos y una baja tasa de falsos positivos, lo que refleja un mejor rendimiento del modelo.

La curva ROC es una herramienta poderosa para evaluar y comparar modelos de clasificación, ya que proporciona una representación visual intuitiva de su rendimiento en diferentes umbrales de clasificación y permite tomar decisiones informadas sobre qué modelo es más adecuado para una tarea específica

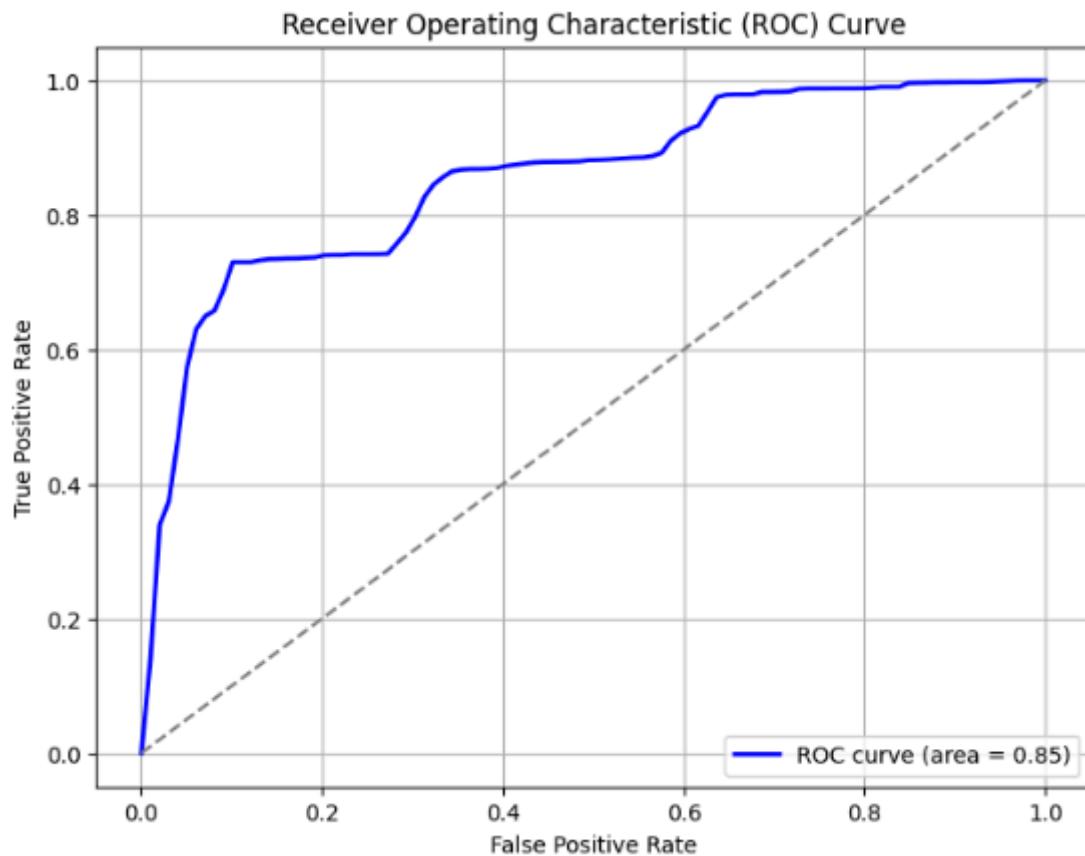


Figura 27: Curva ROC. Fuente: Elaboración propia.

### 8.6.7 F1-SCORE

El F1-Score es una métrica importante en la evaluación de modelos de clasificación, ya que proporciona una medida única que combina tanto la precisión como el recall en una sola métrica.

Esta medida es especialmente útil cuando se desea tener en cuenta tanto la calidad de las predicciones positivas como la capacidad del modelo para identificar correctamente todas las instancias positivas en el conjunto de datos.

El F1-Score se calcula como la media armónica de la precisión y el recall, y se expresa matemáticamente de la siguiente manera:

$$F1 - SCORE = 2 * \frac{PRECISION * RECALL}{PRECISION + RECALL}$$

La media armónica es una forma de calcular la media que da más peso a los valores más bajos. Por lo tanto, el F1-Score penaliza los casos en los que tanto la precisión como el recall son bajos, lo que lo hace útil para situaciones en las que deseamos un equilibrio entre la precisión y el recall.

Por ejemplo, si tenemos un modelo de clasificación con una precisión del 80% y un recall del 75%, el cálculo del F1-Score sería:

$$F1 - SCORE = 2 * \frac{0,8 * 0,75}{0,8 * 0,75}$$

$$F1 - SCORE = 2 * \frac{0,6}{1,55}$$

$$F1 - SCORE \approx 0,774$$

Esto indica que el modelo tiene un F1-Score de aproximadamente 0.774, lo que proporciona una medida combinada de la precisión y el recall del modelo. El F1-Score es una métrica útil para evaluar el equilibrio entre la precisión y el recall en un modelo de clasificación.

### 8.6.8 CLASSIFICATION REPORT

El informe de clasificación es una herramienta esencial en la evaluación detallada de modelos de clasificación, ya que proporciona un resumen exhaustivo de las métricas de rendimiento más importantes. Este informe incluye la precisión, el recall, el F1-Score y otras métricas relevantes para cada clase en el conjunto de datos.

En un informe de clasificación típico, se presentan las métricas de rendimiento para cada clase por separado, lo que permite evaluar cómo se desempeña el modelo en la predicción de cada categoría específica. Esto es especialmente útil en conjuntos de datos donde las clases están desbalanceadas o tienen importancia diferencial en la aplicación.

El informe de clasificación también puede incluir otras métricas como el soporte, que indica el número de instancias de cada clase en el conjunto de datos, así como métricas específicas para problemas multiclase como la precisión macro, la precisión ponderada y el recall macro.

Por lo que el informe de clasificación proporciona una visión completa y detallada del rendimiento del modelo en la tarea de clasificación, lo que ayuda a los investigadores y profesionales a comprender mejor las fortalezas y debilidades del modelo y a tomar decisiones informadas sobre su uso y mejora.

	precision	recall	f1-score	support
0	0.92	0.93	0.92	12500
1	0.93	0.92	0.92	12500
micro avg	0.92	0.92	0.92	25000
macro avg	0.92	0.92	0.92	25000
weighted avg	0.92	0.92	0.92	25000

Figura 28: Informe de Clasificación (Classification Report). Fuente: Elaboración propia.

### 8.6.9 CONCLUSIONES

Sin duda, las métricas de rendimiento son elementos esenciales en la evaluación de modelos de clasificación, ya que proporcionan una comprensión cuantitativa del desempeño del modelo y permiten comparar diferentes enfoques de manera objetiva. Al analizar estas métricas, podemos tomar decisiones informadas sobre qué modelo es más adecuado para una tarea de clasificación específica.

La matriz de confusión, por ejemplo, ofrece una visión detallada de cómo el modelo clasifica las instancias en cada clase, lo que ayuda a identificar posibles áreas de mejora. La precisión y el recall proporcionan información sobre la calidad y la exhaustividad de las predicciones del modelo, respectivamente, mientras que la curva ROC y el área bajo la curva (AUC) ofrecen una perspectiva más global del rendimiento del modelo en diferentes umbrales de clasificación.

El F1-Score, al ser una combinación de precisión y recall, equilibra estas dos métricas y proporciona una medida única del rendimiento del modelo. Además, el informe de clasificación detalla las métricas de rendimiento para cada clase en el conjunto de datos, lo que permite una comprensión más completa del desempeño del modelo en cada categoría.

Estas métricas de rendimiento son herramientas críticas para evaluar la efectividad de los modelos de clasificación y para tomar decisiones informadas sobre su implementación y mejora. Proporcionan una base sólida para la comparación y la optimización de modelos, lo que contribuye a avanzar en el campo de la inteligencia artificial y el aprendizaje automático.

## 8.7 DATASET UNSW-NB15

En el ámbito del aprendizaje automático, los conjuntos de datos juegan un papel crucial. Son fundamentales para la construcción, entrenamiento y evaluación de modelos, permitiendo que los algoritmos aprendan patrones a partir de datos históricos y realicen predicciones precisas sobre datos nuevos. La calidad y la relevancia de los datasets utilizados tienen un impacto directo en la efectividad y precisión de los modelos desarrollados. En la investigación de seguridad cibernética, donde las amenazas son complejas y evolucionan constantemente, disponer de datasets de alta calidad es aún más crítico. Un buen dataset en este campo debe reflejar la diversidad y la dinámica de las amenazas reales, proporcionando una base sólida para la identificación y mitigación de riesgos.

### **8.7.1 ORIGEN DEL DATASET UNSW-NB15**

El dataset UNSW-NB15 [31] fue desarrollado por el Australian Centre for Cyber Security (ACCS) en la Universidad de Nueva Gales del Sur (UNSW), Australia. Su creación tuvo lugar en 2015, con el objetivo de ofrecer un conjunto de datos realista y representativo de los entornos de red modernos. A pesar de haberse originado en 2015, el dataset ha sido actualizado continuamente, con la última actualización realizada el 8 de febrero de 2024.

La generación del dataset UNSW-NB15 involucró la captura de tráfico de red en un entorno controlado que replicaba una red empresarial moderna. El tráfico fue generado utilizando la herramienta IXIA PerfectStorm, que permite simular una amplia variedad de actividades de red, incluyendo tráfico benigno y malicioso.

Posteriormente, los datos capturados fueron etiquetados manualmente por expertos en seguridad cibernética, asegurando la precisión y la relevancia de las etiquetas.

El UNSW-NB15 es un conjunto de datos ampliamente utilizado en la investigación de seguridad cibernética y el aprendizaje automático, específicamente en la detección de intrusiones y análisis de amenazas. Este dataset fue creado para superar las limitaciones de conjuntos de datos anteriores y proporcionar un recurso más completo y actualizado para el desarrollo y evaluación de sistemas de detección de intrusiones.

### **8.7.2 ANÁLISIS DEL DATASET UNSW-NB15**

El UNSW-NB15 fue diseñado para reflejar de manera realista los entornos de red modernos. Incluye una variedad de tráficos, tanto benignos como maliciosos, capturados en un entorno de red simulado que imita las condiciones de una red empresarial típica.

Este dataset abarca múltiples clases de ataques cibernéticos, como análisis, explotación, infiltración y denegación de servicio (DoS). Esta diversidad permite probar y validar la eficacia de los modelos contra una amplia gama de amenazas.

Los datos en el UNSW-NB15 fueron etiquetados manualmente por expertos en seguridad cibernética, lo que garantiza la precisión y fiabilidad de las etiquetas. Esta precisión es crucial para el entrenamiento efectivo de los modelos de aprendizaje automático.

El dataset UNSW-NB15 es de acceso abierto y viene acompañado de una documentación detallada que describe su estructura, características y metodología de generación. Esto me ha facilitado su uso en el desarrollo del proyecto.

El conjunto de datasets UNSW-NB15 contiene un total de 22.339.021 registros, distribuidos en veintitrés archivos CSV. Cada registro está compuesto por 46 características que proporcionan una visión detallada del tráfico de red y son esenciales para el análisis profundo de los patrones de tráfico.

La Tabla 11 presenta de manera exhaustiva la información sobre las características que contiene cada registro de cada conjunto de datos utilizado en este proyecto. En esta tabla, se detallan las distintas variables recopiladas, proporcionando una descripción completa de cada una de ellas. Esto incluye no solo los nombres de las características, sino también su tipo (por ejemplo, numérica, categórica, numerico decimales,...).

Características	Tipo	Descripción
ts	int64	Marca de tiempo que indica el momento en que se capturó la conexión.
src_ip	string	Dirección IP de origen de la conexión.
src_port	int64	Puerto de origen donde se estableció la conexión.
dst_ip	string	Dirección IP de destino de la conexión.
dst_port	int64	Puerto de destino al cual se estableció la conexión.
proto	string	Protocolo de red utilizado (e.g., TCP, UDP).
service	string	Tipo de servicio al que está asociado el tráfico (e.g., HTTP, FTP)
duration	float64	Duración de la conexión / sesión en segundos
src_bytes	string	Número de bytes enviados desde la fuente
dst_bytes	int64	Número de bytes recibidos por el destino
conn_state	string	Estado de la conexión (e.g., ESTABLISHED, CLOSED)
missed_bytes	int64	Número de bytes perdidos en la conexión
src_pkts	int64	Número de paquetes enviados desde la fuente
src_ip_bytes	int64	Número total de bytes enviados desde la fuente.
dst_pkts	int64	Número de paquetes recibidos por el destino.
dst_ip_bytes	int64	Número total de bytes recibidos por el destino.
dns_query	string	Consulta DNS realizada.
dns_qclass	int64	Clase de la consulta DNS.
dns_qtype	int64	Tipo de la consulta DNS.
dns_rcode	int64	Código de respuesta DNS.
dns_AA	string	Indicador de si la respuesta DNS es autoritativa.
dns_RD	string	Indicador de si se solicitó resolución recursiva.
dns_RA	string	Indicador de si la resolución recursiva está disponible.
dns_rejected	string	Indicador de si la consulta DNS fue rechazada.
ssl_version	string	Versión del protocolo SSL utilizado.
ssl_cipher	string	Cifrado SSL utilizado en la conexión
ssl_resumed	string	Indicador de si la sesión SSL fue reanudada
ssl_established	string	Indicador de si la conexión SSL fue establecida.
ssl_subject	string	Asunto del certificado SSL.
ssl_issuer	string	Emisor del certificado SSL.
http_trans_depth	string	Profundidad de la transacción HTTP.
http_method	string	Método HTTP utilizado (e.g., GET, POST).
http_uri	string	URI de la solicitud HTTP.
http_referrer	string	Referrer de la solicitud HTTP.
http_version	string	Versión del protocolo HTTP utilizado.
http_request_body_len	int64	Longitud del cuerpo de la solicitud HTTP.
http_response_body_len	int64	Longitud del cuerpo de la respuesta HTTP.
http_status_code	int64	Código de estado de la respuesta HTTP.
http_user_agent	string	User-Agent de la solicitud HTTP.
http_orig_mime_types	string	Tipos MIME originales de la solicitud HTTP.
http_resp_mime_types	string	Tipos MIME de la respuesta HTTP.
weird_name	string	Nombre del evento extraño o anómalo.
weird_addl	string	Información adicional sobre el evento extraño o anómalo.
weird_notice	string	Indicador de si se generó una notificación para el evento extraño o anómalo.
label	int64	Etiqueta que indica si el tráfico es benigno o malicioso.
type	string	Tipo de ataque malicioso o si es una conexión benigna.

Tabla 11: Características de los conjuntos UNSW-NB15. Fuente: Elaboración propia.

Cada dataset contiene un total de 29 categorías en formato string, 16 en formato biginteger (int64) y un campo bigfloat (float64).

El dataset ha sido actualizado regularmente para mantenerse al día con las nuevas amenazas y técnicas de ataque emergentes. La última actualización, realizada el 8 de febrero de 2024, asegura que los datos sigan siendo relevantes y útiles para la investigación contemporánea.

## 8.8 PREPROCESAMIENTO DE DATOS

El preprocessamiento de datos es una etapa crucial en el análisis y modelado de datos. Consiste en una serie de pasos destinados a preparar los datos para ser usados en modelos de machine learning o análisis estadísticos. Un preprocessamiento adecuado es fundamental para obtener resultados precisos y confiables, ya que permite:

Mejorar la calidad de los datos, eliminando errores, incoherencias y valores atípicos que puedan afectar negativamente al análisis. Facilitar la comprensión de los datos, transformando los datos en un formato más intuitivo y visualizable. Finalmente, preparar los datos para el modelado adaptando los datos a los requisitos específicos de los algoritmos de machine learning. [32]

### 8.8.1 LIMPIEZA DE DATOS

La limpieza de datos consiste en identificar y corregir errores, inconsistencias y valores atípicos que puedan afectar negativamente al análisis o al entrenamiento de los modelos. Un conjunto de datos limpio y de alta calidad es esencial para obtener resultados precisos y confiables.

#### 8.8.1.1 ELIMINACIÓN DE DATOS VACÍOS

Los datos vacíos son aquellos que no tienen ningún valor. Estos datos pueden provenir de diferentes causas, como errores de entrada, registros incompletos o valores no aplicables. La eliminación de datos vacíos es una tarea importante en la limpieza de datos debido a que la eliminación de estos, puede mejorar la calidad de los datos eliminando los registros que no contienen información útil. Además, facilita el análisis simplificando el procesamiento de los datos y evitando errores en los modelos de machine learning ya que los algoritmos pueden interpretar los datos vacíos de manera incorrecta, lo que puede llevar a resultados erróneos.

Sin embargo, es importante tener en cuenta que la eliminación de datos vacíos no siempre es la mejor solución. Es recomendable eliminar datos vacíos cuando representan un porcentaje significativo del conjunto de datos, no aportan información útil para el análisis y su eliminación no afecta significativamente al tamaño del conjunto de datos.

#### 8.8.1.2 IMPUTACIÓN DE DATOS

La imputación de datos consiste en rellenar los valores vacíos con un valor estimado. Existen diferentes métodos de imputación, cada uno con sus propias ventajas y desventajas. Es recomendable imputar datos vacíos cuando representan un pequeño porcentaje del conjunto de datos, aportan información importante y su eliminación afectaría significativamente al tamaño del conjunto de datos.

La elección del método de imputación depende de las características del conjunto de datos y del análisis que se vaya a realizar. En general, se recomienda utilizar un método que tenga en cuenta la distribución de la variable y que no introduzca sesgos en los datos.

Existen diferentes técnicas para llenar esos datos vacíos, alguna de las más comunes son:

*Imputación por la media:* Consiste en llenar los valores vacíos con la media de la variable correspondiente.

*Imputación por la mediana:* Consiste en llenar los valores vacíos con la mediana de la variable correspondiente.

*Imputación por el modo:* Consiste en llenar los valores vacíos con el modo de la variable correspondiente.

*Imputación por regresión:* Consiste en utilizar un modelo de regresión para predecir los valores vacíos en función de las demás variables del conjunto de datos.

### 8.8.1.3 ELIMINACIÓN DE DATOS REDUNDANTES

Los datos redundantes son aquellos que se repiten o que no aportan información adicional al análisis. Eliminar estos datos permite reducir el tamaño del conjunto de datos y mejorar la eficiencia del proceso de análisis. Consiste en identificar y eliminar registros duplicados o que contienen información irrelevante.

Este proceso tiene en cuenta los registros duplicados, los datos que no son relevantes para el análisis, como comentarios o identificadores únicos y los datos inconsistentes, que no tienen sentido o que no coinciden con otros datos del conjunto de datos.

### 8.8.2 TRANSFORMACIÓN DE DATOS

La transformación de datos [33] consiste en aplicar una serie de técnicas a los datos para modificar su estructura o formato, con el objetivo de mejorar su calidad y facilitar su procesamiento por parte de los algoritmos de los modelos de machine learning. Algunas de las técnicas de transformación de datos más comunes se detallan a continuación:

#### 8.8.2.1 NORMALIZACIÓN

La normalización consiste en transformar los valores de las variables a un rango común, generalmente entre 0 y 1. Esto es útil cuando las variables tienen diferentes unidades de medida o escalas. Existen diferentes métodos de normalización como:

*Escala Min-Max:* Este método linealmente escala los datos de tal manera que los valores mínimos y máximos de las variables se ajustan a un rango predefinido, típicamente entre 0 y 1. La fórmula utilizada es:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

donde  $X_{norm}$  es el valor normalizado,  $X$  es el valor original,  $X_{min}$  es el valor mínimo de la variable, y  $X_{max}$  es el valor máximo de la variable.

**Z-Score (Estandarización):** Este método transforma los datos para que tengan una media de 0 y una desviación estándar de 1. Es particularmente útil cuando se desea comparar datos que siguen una distribución normal. La fórmula utilizada es:

$$Z = \frac{X - \mu}{\sigma}$$

donde  $Z$  es el valor normalizado,  $X$  es el valor original,  $\mu$  es la media de la variable, y  $\sigma$  es la desviación estándar de la variable.

La normalización es útil cuando las variables tienen diferentes unidades de medida o escalas, cuando los algoritmos de machine learning son sensibles a la escala de los datos o cuando se desea mejorar la interpretabilidad de los resultados del modelo.

Un inconveniente que genera normalizar los datos de un dataset es que podemos perder información sobre la distribución original de los datos y puede ser sensible a valores atípicos.

### 8.8.2.2 ESCALADO

El escalado consiste en ajustar la distribución de las variables para que se aproxime a una distribución normal. Esto es útil para algunos algoritmos de machine learning que son sensibles a la distribución de los datos. Este proceso se utiliza para homogeneizar las escalas de las distintas variables, facilitando así la convergencia de los algoritmos de aprendizaje y mejorando la interpretabilidad de los resultados.

El escalado es crucial cuando se trabaja con algoritmos de machine learning que asumen que los datos están normalmente distribuidos o que son sensibles a la escala de las variables, como los algoritmos basados en distancias (por ejemplo, k-Nearest Neighbors, Support Vector Machines) y aquellos que utilizan gradientes para la optimización (por ejemplo, redes neuronales, regresión logística). Además, el escalado es útil para mitigar el impacto de los valores atípicos, que pueden distorsionar los resultados de algunos modelos.

Al escalar los datos, se puede perder información sobre la distribución original, lo que podría afectar la interpretación de los resultados. Además, este proceso puede aumentar la complejidad del modelo, especialmente si no se elige el método adecuado para las características específicas del dataset. Por lo tanto, es fundamental seleccionar el método de escalado que mejor se adapte a la naturaleza de los datos y al algoritmo que se va a utilizar.

Existen diversos métodos, cada uno con sus propias características y aplicaciones:

**Escalado Estándar:** Este método implica aplicar la normalización Z a cada variable, transformándola para que tenga una media de 0 y una desviación estándar de 1. Este enfoque es útil cuando los datos no contienen muchos valores atípicos y se requiere una distribución normalizada.

**Escalado Robusto:** Este método en lugar de utilizar la media y la desviación estándar de la variable, emplea la mediana y la desviación absoluta mediana (MAD). La MAD es menos sensible a los valores atípicos que la desviación estándar, lo que hace que el escalado robusto sea más eficiente para datasets con valores extremos.

Ambos métodos de escalado tienen como objetivo estandarizar las variables, pero eligen diferentes estadísticos para lograrlo. La elección entre estos métodos depende de la estructura y las peculiaridades del conjunto de datos.

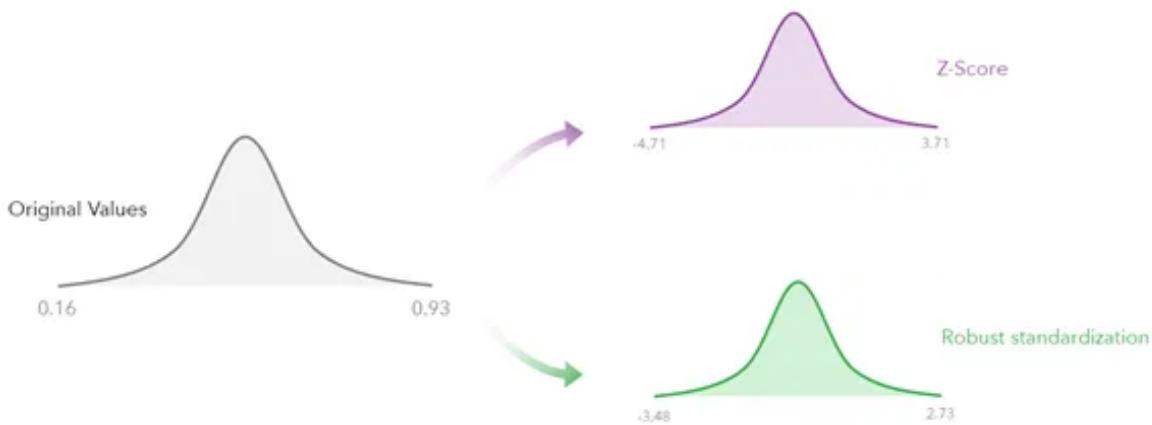


Figura 29: Escalado Estándar vs Escalado Robusto. Fuente: Medium.

### 8.8.2.3 CODIFICACIÓN DE DATOS CATEGÓRICOS

En el mundo del machine learning, los datos categóricos son aquellos que no se pueden representar numéricamente, sino que se expresan mediante etiquetas o categorías. Por ejemplo, el género (hombre, mujer), el color (rojo, azul, verde) o el tipo de animal (perro, gato, pájaro).

Sin embargo, para que los algoritmos de machine learning puedan procesar y aprender de este tipo de datos, es necesario convertirlos en un formato numérico. Este proceso se conoce como codificación de datos categóricos. Existen diferentes métodos para codificar datos categóricos, cada uno con sus propias ventajas y desventajas. A continuación, se describen dos de los métodos más comunes:

#### Codificación de Etiquetas:

La codificación de etiquetas, también conocida como "label encoding", consiste en asignar un valor numérico (0, 1, 2, etc.) a cada categoría de una variable categórica. Por ejemplo, si tenemos una variable que representa el género (hombre, mujer), podemos codificarla de la siguiente manera: Hombre = 0, Mujer = 1.

Este método es simple y fácil de implementar, pero tiene la desventaja de que no conserva la información sobre el orden de las categorías. Por ejemplo, si codificamos la variable "color" (rojo, azul, verde) de la misma manera, no sabremos qué color es más "cercano" a otro, ya que no considera ninguna relación de proximidad.

#### Codificación One-Hot Encoding:

La codificación one-hot consiste en crear una nueva variable para cada categoría de una variable categórica. Cada nueva variable tendrá un valor de 1 para la categoría correspondiente y un valor de 0 para todas las demás. Por ejemplo, si tenemos una variable que representa el género (hombre, mujer), podemos codificarla de la siguiente manera: Hombre = [1, 0], Mujer = [0, 1].

Este método es eficaz para conservar la información sobre las categorías, ya que cada categoría se representa de manera única y no hay implicación de orden entre ellas. Sin embargo, puede generar un gran número de variables, especialmente si la variable categórica tiene muchas categorías.

La elección del método de codificación de datos categóricos depende de las características del conjunto de datos y del análisis que se vaya a realizar. En general, la codificación binaria es adecuada para variables categóricas con pocas categorías, mientras que la codificación one-hot es más útil para variables con muchas categorías o cuando es importante no implicar un orden entre las categorías.

Es crucial tener en cuenta que no existe un método único que sea el mejor en todos los casos. Es importante probar diferentes métodos y elegir el que mejor funcione para el caso particular. [34]

#### **8.8.2.4 TOKENIZACIÓN**

La tokenización es el proceso de dividir un texto en unidades más pequeñas, llamadas tokens, que pueden ser palabras, frases o incluso caracteres. Este paso es esencial en el procesamiento de lenguaje natural (NLP). La tokenización facilita la manipulación y el análisis del texto, permitiendo a los modelos de NLP trabajar con datos textuales de manera más efectiva.

Existen diferentes métodos de tokenización, pero algunos de los más comunes son:

*Tokenización basada en espacios en blanco:* Este método divide el texto en palabras en función de los espacios en blanco.

*Tokenización basada en puntuación:* Este método divide el texto en palabras y frases en función de los signos de puntuación.

*Tokenización basada en expresiones regulares:* Este método utiliza expresiones regulares para dividir el texto en tokens.

La elección del método de tokenización depende del tipo de texto que se esté procesando y de la tarea que se esté realizando.

#### **8.8.3 MANEJO DEL DESEQUILIBRIO DE CLASES**

El desequilibrio de clases se produce cuando una de las clases en un conjunto de datos de clasificación está representada por un número significativamente menor de registros que la otra clase. Esto puede afectar negativamente al rendimiento del modelo, ya que los algoritmos de machine learning tienden a favorecer a la clase mayoritaria, lo que puede sesgar los resultados de los modelos de machine learning.

#### **8.8.3.1 SUBMUESTREO (UNDERSAMPLING)**

El submuestreo, también conocido como "undersampling", es una técnica para manejar el desequilibrio de clases en conjuntos de datos de clasificación. Consiste en eliminar aleatoriamente registros de la clase mayoritaria hasta que el número de registros de ambas clases sea igual. Esto se hace con el objetivo de igualar la representación de las clases en el dataset y prevenir que los algoritmos de machine learning se sesguen hacia la clase predominante.

El submuestreo es una buena opción cuando los datos de la clase minoritaria son valiosos y costosos de obtener o cuando la clase minoritaria contiene información crucial para el análisis. Eliminar registros de la clase mayoritaria puede ser una forma eficaz de reducir el tamaño del conjunto de datos sin perder información importante y puede ayudar a evitar que el modelo se sesgue hacia la clase mayoritaria.

La técnica de submuestreo puede favorecer a la pérdida de información de la clase mayoritaria, ya que si se eliminan demasiados registros de la clase mayoritaria, el modelo puede no tener suficiente información para aprender correctamente esta clase. Además, si el conjunto de datos de la clase mayoritaria es pequeño, eliminar registros puede aumentar la varianza del modelo y hacerlo más susceptible al sobreajuste.

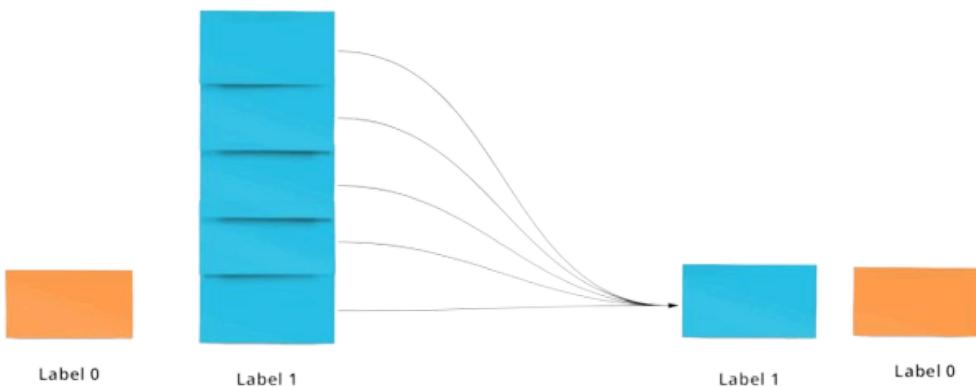


Figura 30: Submuestreo (UnderSampling). Fuente: Neptune.ai

#### 8.8.3.2 SOBREMUESTREO (OVERSAMPLING)

El sobremuestreo, también conocido como "oversampling", es otra técnica utilizada para equilibrar un conjunto de datos desbalanceado al aumentar el número de ejemplos en la clase minoritaria. Esto se logra duplicando ejemplos existentes o generando nuevos ejemplos sintéticos, con el objetivo de igualar la representación de las clases en el dataset.

El sobremuestreo es una buena opción cuando los datos de la clase minoritaria son relativamente fáciles de obtener, ya que crear copias de los registros de la clase minoritaria puede ser una forma eficaz de aumentar el tamaño del conjunto de datos sin perder información importante. También, si la clase minoritaria es muy importante para el análisis, sobremuestrear la clase minoritaria puede ayudar a garantizar que el modelo le preste la atención adecuada.

La técnica de sobremuestreo puede favorecer a un aumento del sesgo del modelo, ya que si se crean demasiadas copias de los registros de la clase minoritaria, el modelo puede sesgar hacia esta clase. Además, si el conjunto de datos de la clase minoritaria es pequeño, crear copias puede aumentar la varianza del modelo y hacerlo más susceptible al sobreajuste.

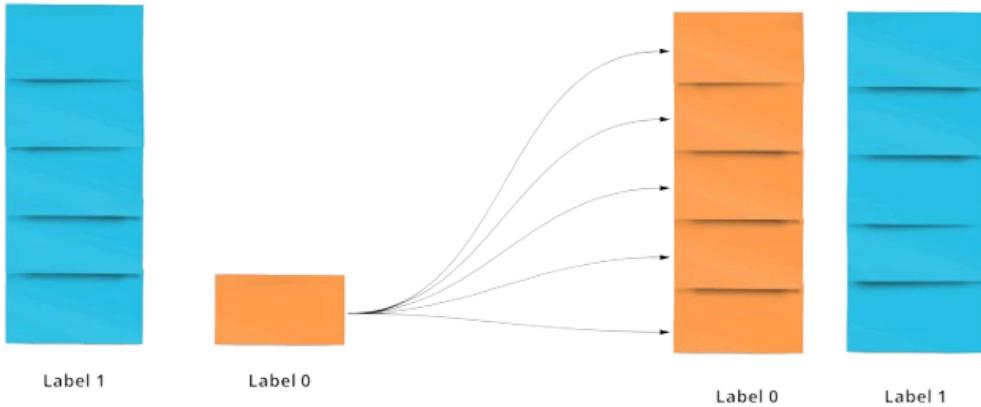


Figura 31: Sobremuestreo (OverSampling). Fuente: Neptune.ai

### 8.8.3.3 METODOS AVANZADOS (SMOTE, ADASYN)

Los métodos avanzados de manejo del desequilibrio de clases buscan generar ejemplos sintéticos de una manera más sofisticada, mejorando la calidad del conjunto de datos balanceado sin simplemente duplicar ejemplos existentes.

SMOTE (Synthetic Minority Over-sampling Technique) y ADASYN (Adaptive Synthetic) son dos métodos avanzados de sobremuestreo que se han demostrado más efectivos que el sobremuestreo aleatorio simple. Estos métodos crean nuevos registros de la clase minoritaria mediante la interpolación entre registros existentes. Esto ayuda a garantizar que los nuevos registros sean similares a los registros originales y que no aumenten el sesgo del modelo.

SMOTE y ADASYN son una buena opción porque pueden crear nuevos registros, de la clase minoritaria, que sean más representativos y ayuda a garantizar que el modelo le preste la atención adecuada a la clase minoritaria. A pesar de ello, pueden ser computacionalmente más costosos que el sobremuestreo aleatorio simple y son más sensibles al ruido en los datos. [36]

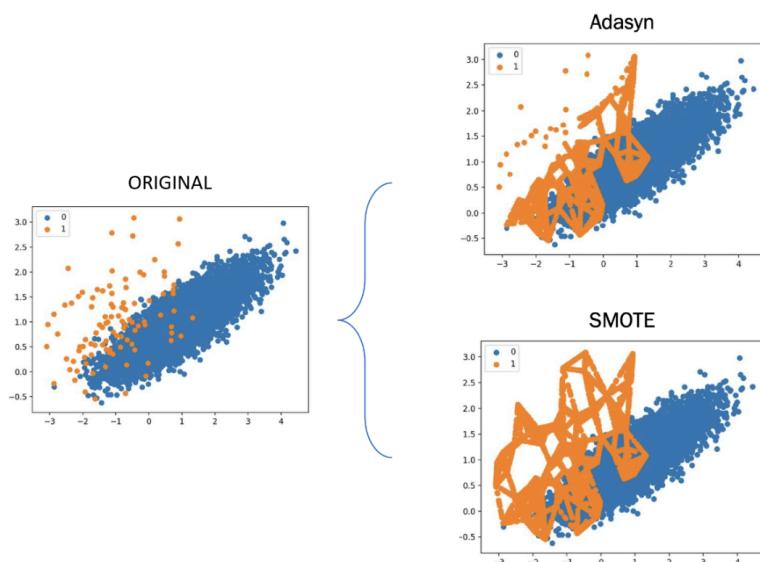


Figura 32: Gráfico de equilibrio de clases SMOTE y ADASYN. Fuente: RPubs

## 8.9 INGENIERÍA DE CARACTERÍSTICAS

La ingeniería de características es una parte fundamental del proceso de aprendizaje automático y análisis de datos, que consiste en seleccionar características (o variables) que pueden mejorar el rendimiento de los modelos predictivos. La selección de características es un paso crítico dentro de este proceso, ya que puede reducir la complejidad del modelo, mejorar su rendimiento y evitar el sobreajuste.

### 8.9.1 MÉTODOS DE SELECCIÓN DE CARACTERÍSTICAS

Existen varios métodos para la selección de características, los cuales se pueden clasificar en tres categorías principales: métodos basados en filtros, métodos basados en envoltorios y métodos basados en modelos. [37]

#### 8.9.1.1 MÉTODO BASADO EN FILTROS

Estos métodos seleccionan características basándose en estadísticas de los datos sin involucrar ningún modelo de aprendizaje automático. Son rápidos y sencillos de implementar. Algunos de los métodos basados en filtros más comunes incluyen:

*Chi-cuadrado*: Mide la independencia de dos eventos, en este caso, la característica y la etiqueta.

*Análisis de varianza (ANOVA)*: Utiliza la prueba F para comparar las variaciones entre grupos.

*Correlación*: Calcula la correlación entre las características y la etiqueta. Se pueden utilizar coeficientes de correlación como Pearson, Spearman o Kendall.

*Información Mutua*: Mide la dependencia mutua entre las características y la etiqueta.

*Varianza*: Elimina características con baja varianza, ya que aportan poca información.

#### 8.9.1.2 MÉTODO BASADO EN ENVOLTURAS

Estos métodos utilizan un modelo predictivo para evaluar la importancia de las características. La selección se realiza mediante la optimización del rendimiento del modelo. Son más precisos que los métodos basados en filtros, pero también más costosos computacionalmente. Algunos ejemplos incluyen:

*Selección hacia adelante (Forward Selection)*: Comienza con un modelo vacío y agrega características una por una, seleccionando en cada paso la característica que mejor mejora el rendimiento del modelo.

*Selección hacia atrás (Backward Elimination)*: Comienza con todas las características y las elimina una por una, en cada paso quitando la característica que menos afecta el rendimiento del modelo.

*Selección recursiva de características (Recursive Feature Elimination, RFE)*: Utiliza un modelo (por ejemplo, una regresión logística) para evaluar la importancia de las características y elimina recursivamente las menos importantes.

### 8.9.1.3 MÉTODO BASADO EN MODELOS

Estos métodos utilizan modelos específicos que tienen incorporada la capacidad de seleccionar características. Algunos ejemplos son:

*Lasso (Least Absolute Shrinkage and Selection Operator)*: Utiliza regularización L1 que puede forzar algunos coeficientes a ser exactamente cero, eliminando efectivamente algunas características.

*Árboles de Decisión y Bosques Aleatorios*: Los árboles de decisión y los bosques aleatorios pueden medir la importancia de las características basado en el número de veces que se usan para dividir los nodos y cómo afectan la pureza de las divisiones.

*Gradient Boosting Machines (GBM)*: Similar a los bosques aleatorios, GBM puede proporcionar una medida de la importancia de las características.

## 8.10 VALIDACIÓN

Este procesos permiten evaluar la eficacia y robustez de un modelo predictivo antes de su implementación en un entorno de producción, asegurando que el modelo no solo se desempeñe bien en los datos de entrenamiento, sino que también generalice adecuadamente a nuevos datos no vistos.

La validación es el proceso de utilizar un conjunto de datos independiente del conjunto de datos de entrenamiento para ajustar los hiperparámetros del modelo y evaluar su rendimiento de manera objetiva.

### 8.10.1 DIVISIÓN DE DATOS DE TRAINING Y TEST

Dividir adecuadamente los datos en conjuntos de entrenamiento (training) y prueba (test) es fundamental para el desarrollo de modelos de aprendizaje automático. Esta división permite evaluar de manera objetiva el rendimiento del modelo y su capacidad de generalización.

Las proporciones de división pueden variar dependiendo del tamaño del conjunto de datos y del contexto del problema.

- **70/30**: Una división común donde el 70% de los datos se utiliza para el entrenamiento y el 30% para la prueba.
- **80/20**: Similar a la anterior pero con más datos dedicados al entrenamiento.
- **60/20/20**: Para divisiones que incluyen conjuntos de validación, con el 60% para entrenamiento, 20% para validación y 20% para prueba.

Además, existen diversas técnicas para dividir el conjunto de datos, cada una con sus propias ventajas y aplicaciones. A continuación, se describen dos de las técnicas más comunes en detalle:

### 8.10.1.1 HOLD-OUT VALIDATION

Consiste en dividir el conjunto de datos original en dos partes: un conjunto de entrenamiento y un conjunto de validación. El modelo se entrena con el conjunto de entrenamiento y se valida con el conjunto de validación. Este método es simple pero puede ser sensible a cómo se realiza la división de los datos.

Generalmente el conjunto de entrenamiento contiene entre el 70% y el 80% de los datos y el otro 20% - 30% para el conjunto de validación/prueba para evaluar el rendimiento final del modelo una vez que se ha entrenado.

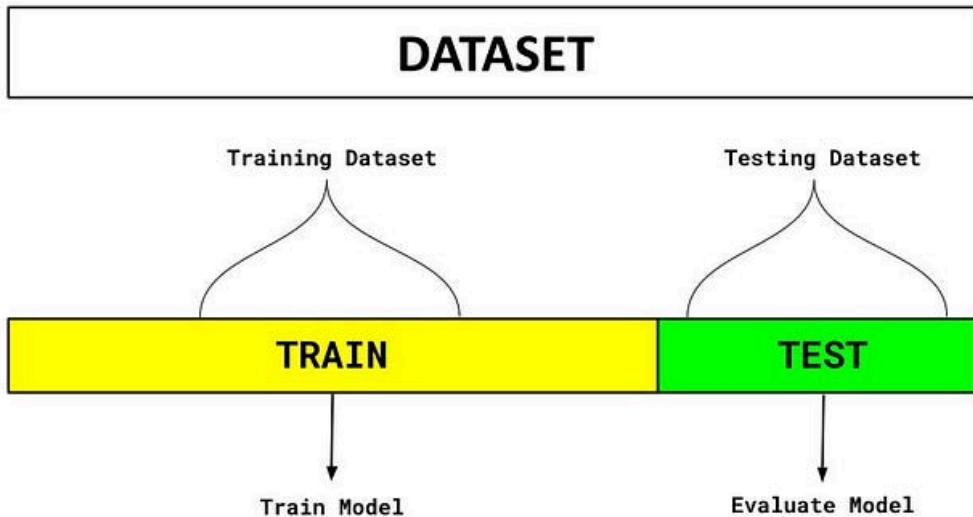


Figura 33: Validacion Hold-Out. Fuente: Comet.

En proyectos más complejos o donde la evaluación del modelo necesita una validación más rigurosa, se puede introducir un tercer conjunto de datos conocido como conjunto de validación. Este conjunto de validación se utiliza junto con los conjuntos de entrenamiento y prueba para ajustar los hiperparámetros del modelo y realizar ajustes finos en la arquitectura o configuración del algoritmo.

Mientras que el conjunto de entrenamiento se utiliza exclusivamente para entrenar el modelo y el conjunto de prueba se reserva para evaluar su rendimiento final después del entrenamiento, el conjunto de validación se utiliza de manera intermedia.

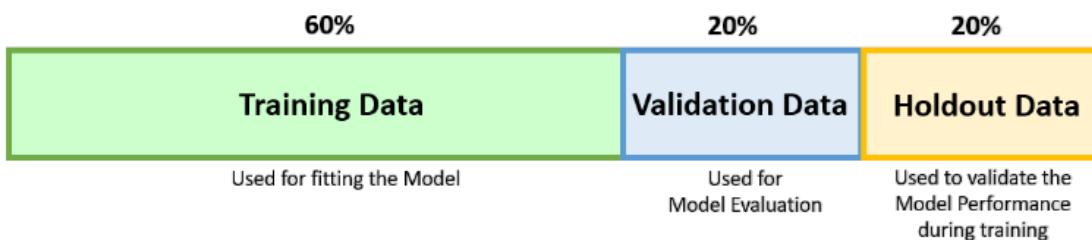


Figura 34: Validacion Hold-Out. Fuente: Comet. [38]

### 8.10.1.2 VALIDACIÓN CRUZADA (CROSS-VALIDATION)

La validación cruzada es una técnica más robusta que la división simple. Se divide el conjunto de datos en "k" subconjuntos (folds) de igual tamaño. El modelo se entrena "k" veces utilizando diferentes combinaciones de estos subconjuntos y se evalúa en el subconjunto restante. El rendimiento final del modelo se promedia a partir de todas las iteraciones. La validación cruzada más común es la "k-fold cross-validation", con "k" generalmente fijado en 5 o 10. Este proceso garantiza que cada observación se utilice tanto para el entrenamiento como para la validación.

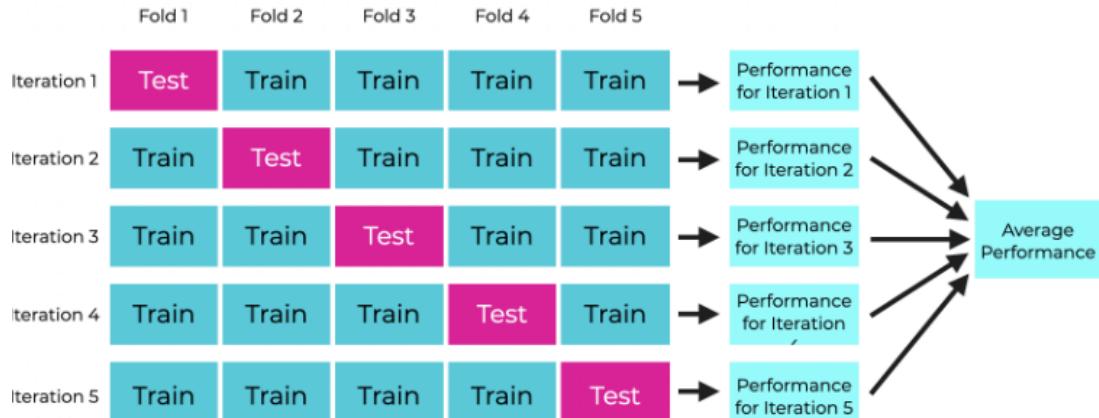


Figura 35: Validación Cruzada K-Fold donde K=5. Fuente: Sharp Sight. [39]

### 8.10.1.3 DIVISION ESTRATIFICADA

Para problemas de clasificación, especialmente aquellos con clases desbalanceadas, es esencial realizar una división estratificada. Esto asegura que cada conjunto (training y test) tenga aproximadamente la misma proporción de clases, lo que permite una evaluación más precisa del modelo.

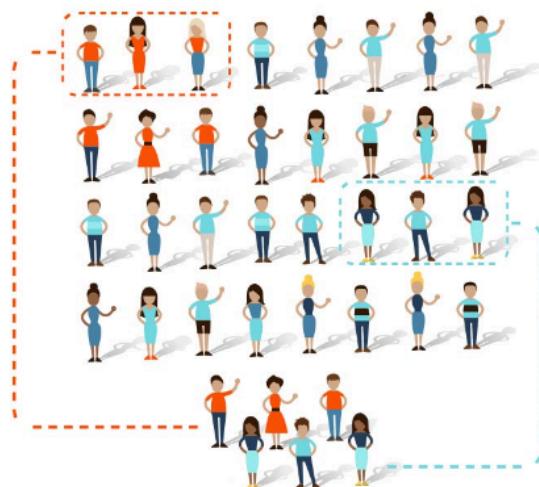


Figura 36: Validación Estratificada. Fuente: QuestionPro. [40]

## 8.11 ZEEK

Zeek [41] (anteriormente conocido como Bro) es un potente sistema de análisis de red de código abierto que se utiliza ampliamente para la detección de amenazas, el monitoreo de seguridad y la generación de informes en entornos de redes de computadoras. Zeek está diseñado para analizar tráfico de red en tiempo real y extraer información útil sobre la actividad de la red, identificando patrones de tráfico sospechoso, anomalías y posibles intrusiones. Zeek es una herramienta esencial para la monitorización y seguridad de redes, proporcionando una visión detallada y contextualizada de la actividad de la red y ayudando a detectar y responder a posibles amenazas y vulnerabilidades de seguridad.

### 8.11.1 FUNCIONALIDADES

*Análisis de Protocolos de Red:* Zeek es capaz de analizar una amplia variedad de protocolos de red, incluyendo TCP, UDP, ICMP, HTTP, DNS, FTP, SSL/TLS, entre otros. Esto le permite entender y examinar el tráfico en la red a un nivel detallado.

*Registro de Eventos:* Zeek genera registros detallados de eventos que ocurren en la red, proporcionando información valiosa sobre la actividad de la red, como conexiones establecidas, transferencias de archivos, consultas DNS, solicitudes HTTP, entre otros. Estos registros pueden ser analizados posteriormente para identificar patrones de comportamiento anómalos o sospechosos.

*Personalización y Extensibilidad:* Zeek es altamente configurable y extensible, lo que permite a los usuarios adaptarlo a sus necesidades específicas y desarrollar nuevos scripts y plugins para ampliar su funcionalidad. Esto lo convierte en una herramienta flexible que puede adaptarse a una amplia variedad de entornos y casos de uso.

*Integración con Otros Sistemas de Seguridad:* Zeek se integra fácilmente con otros sistemas y herramientas de seguridad, como sistemas de gestión de eventos e información de seguridad (SIEM), sistemas de prevención de intrusiones (IPS), firewalls, y sistemas de detección de intrusiones (IDS), lo que permite una mejor coordinación y respuesta a las amenazas.

*Código Abierto y Comunidad Activa:* Zeek es un proyecto de código abierto con una comunidad activa de desarrolladores y usuarios que contribuyen al desarrollo y mejora continua del software. Esto garantiza que Zeek esté constantemente actualizado y pueda adaptarse a los desafíos y cambios en el panorama de seguridad cibernética.

## 8.12 ELASTIC STACK

Elastic Stack [42] es una conjunto de herramientas de software de código abierto desarrolladas por Elastic, una empresa de tecnología de la información con sede en los Estados Unidos. Originalmente conocido como ELK Stack debido a sus tres componentes principales (Elasticsearch, Logstash y Kibana), la suite ha evolucionado y se ha expandido para incluir más herramientas y funcionalidades, lo que llevó a la adopción del término "Elastic Stack".

Aunque sus raíces están en la gestión de registros y análisis de datos, Elastic Stack ha crecido para abarcar una variedad más amplia de casos de uso y soluciones en el ámbito de la búsqueda, análisis y visualización de datos.

### **8.12.1 FILEBEAT**

Filebeat, conocido también como Beat, es un agente ligero de envío de datos desarrollado por Elastic como parte fundamental del Elastic Stack. Este agente está diseñado para recopilar, enviar y centralizar logs y otros datos desde diversas fuentes hacia Elasticsearch, Logstash y otros destinos de almacenamiento y análisis. Es una herramienta esencial en entornos de infraestructura distribuida, donde se emplea para la monitorización de logs, análisis de registros de aplicaciones, seguridad, y otras tareas de recopilación de datos.

Filebeat está especialmente diseñado para recopilar logs de diversas fuentes, como archivos de log de aplicaciones, logs del sistema, logs de contenedores y eventos de red, de manera eficiente y escalable. Utiliza un mecanismo de "tail" para monitorizar activamente los cambios en los archivos de log, enviando únicamente las nuevas líneas o cambios a medida que se producen.

Además, garantiza la seguridad en la transferencia de datos mediante el soporte de envío seguro a través de TLS/SSL, lo que asegura la confidencialidad e integridad de los datos durante la comunicación. También ofrece la autenticación basada en certificados para reforzar la seguridad de las conexiones.

Esta herramienta es altamente escalable y puede distribuirse horizontalmente para manejar grandes volúmenes de logs en entornos distribuidos. La configuración de múltiples instancias de Filebeat permite la recopilación paralela y distribuida de logs, lo que mejora la eficiencia y el rendimiento del proceso de recopilación de datos.

### **8.12.2 ELASTICSEARCH**

Elasticsearch es un motor de búsqueda y análisis distribuido. Es el componente central de la estructura y proporciona capacidades avanzadas para el almacenamiento, búsqueda y análisis de datos en tiempo real.

Elasticsearch almacena y organiza datos en forma de documentos JSON, que se indexan y almacenan de manera distribuida en clusters de nodos. Los datos indexados se pueden buscar, recuperar y analizar de forma eficiente utilizando consultas de texto completo y consultas estructuradas.

Además proporciona capacidades avanzadas de búsqueda, que permiten realizar búsquedas rápidas y precisas en grandes volúmenes de datos, utilizando una variedad de operadores, filtros y funciones para encontrar información relevante.

Los datos en tiempo real pueden ser analizados, utilizando operaciones de agregación como sumas, promedios, máximos, mínimos y conteos en datos indexados, de diferentes formas.

Los clústeres de Elasticsearch pueden crecer o reducirse dinámicamente agregando o eliminando nodos según sea necesario por lo que puede manejar grandes volúmenes de datos y mantener un alto rendimiento y disponibilidad.

Además proporciona funcionalidades de seguridad integradas que permiten controlar el acceso a los datos y proteger los clústeres contra accesos no autorizados. Además, permite configurar roles y permisos para restringir el acceso a los datos y proteger la integridad y confidencialidad de la información.

### 8.12.3 KIBANA

Kibana es una plataforma de visualización y análisis de datos, diseñada para explorar, visualizar y compartir datos almacenados en Elasticsearch. Es una parte esencial del Elastic Stack y se utiliza comúnmente en combinación con Elasticsearch para crear dashboards interactivos, realizar análisis de datos en tiempo real y compartirlo con otros usuarios, facilitando la toma de decisiones informadas y la comprensión de los datos en entornos empresariales y de infraestructura.

Kibana permite crear dashboards interactivos que combinan múltiples visualizaciones personalizadas, como gráficos de barras, gráficos de líneas, mapas geoespaciales, tablas y diagramas de dispersión, personalizando cada visualización según sus necesidades específicas, ajustando la configuración de colores, escalas, etiquetas y más.

Kibana ofrece herramientas y funciones avanzadas para realizar análisis de datos en tiempo real, como agregaciones, métricas, y cálculos de estadísticas descriptivas. Los usuarios pueden utilizar la funcionalidad de agregación de Kibana para resumir y visualizar datos mediante operaciones como sumas, promedios, mínimos y máximos. También incluye capacidades para la visualización de datos geoespaciales, permitiendo mostrar datos en mapas interactivos.

Kibana facilita la colaboración y compartición con otros usuarios, permitiendo guardar y compartir dashboards y visualizaciones con enlaces públicos o privados. Además permite exportar dashboards y visualizaciones en diferentes formatos, como imágenes, PDF o HTML, para su uso fuera de la plataforma.

### 8.12.4 INTEGRACIÓN

La integración nativa de Filebeat con Elasticsearch y Kibana facilita la recopilación, almacenamiento, procesamiento, análisis de los logs recopilados, búsqueda y visualización de datos. Los datos recolectados por Filebeat pueden ser indexados en Elasticsearch para su búsqueda y análisis, y visualizados en tiempo real utilizando Kibana.

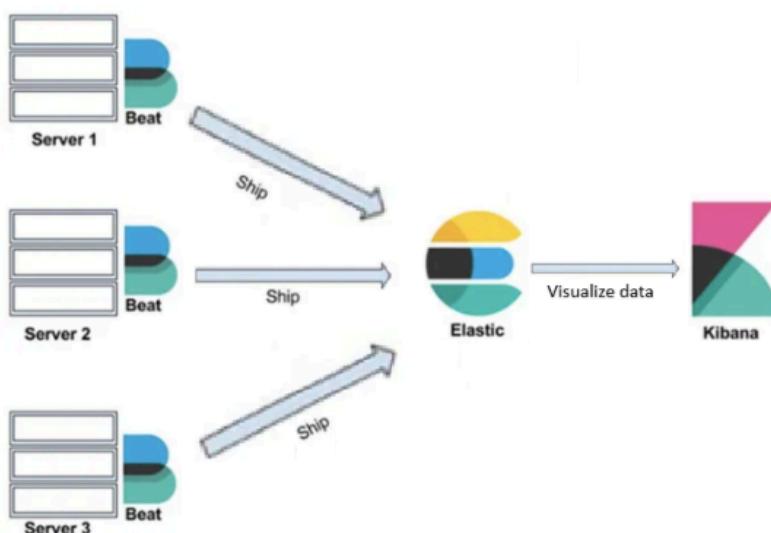


Figura 37: Esquema Elastic Stack. Fuente: Medium.

## 9. ESPECIFICACIÓN DEL SISTEMA

### 9.1 REQUISITOS FUNCIONALES

Los requisitos funcionales son especificaciones detalladas de las funciones y capacidades que un sistema debe tener para cumplir con sus objetivos y satisfacer las necesidades de los usuarios y otras partes interesadas. Estos requisitos describen qué debe hacer el sistema, sin entrar en detalles sobre cómo debe hacerlo.

#### 9.1.1 DIAGRAMA DE CASOS DE USO

Los diagramas de casos de uso son una herramienta fundamental en la ingeniería de software, utilizados para capturar y representar las interacciones entre los usuarios (actores) y el sistema en desarrollo. Estos diagramas se encuentran dentro del conjunto de técnicas de modelado proporcionadas por el Lenguaje Unificado de Modelado (UML), que es un estándar en la industria para la visualización, especificación, construcción y documentación de los artefactos del sistema.

El principal propósito de los diagramas de casos de uso es capturar los requisitos funcionales del sistema desde la perspectiva de los usuarios finales. Esto significa que se centran en qué funcionalidades y acciones pueden realizar los usuarios con el sistema, y cómo el sistema responde a estas interacciones. Esta representación visual ayuda a los analistas, diseñadores y desarrolladores a comprender claramente las expectativas y necesidades de los usuarios, así como a asegurar que el sistema desarrollado cumpla con estos requisitos. En este caso, el usuario tiene unas funcionalidades limitadas, puesto que la gran mayoría del trabajo lo realiza el propio sistema.

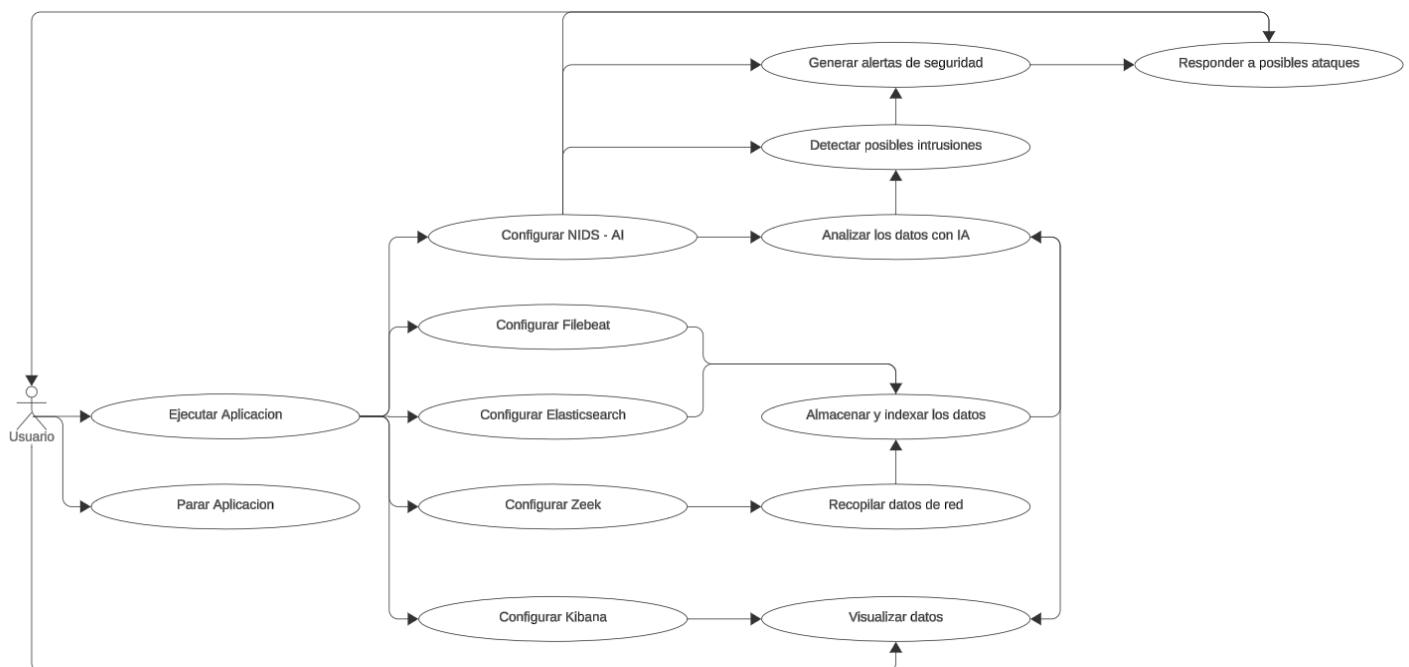


Figura 38: Diagrama de casos de uso del NIDS-AI. Fuente: Elaboración propia

## 9.1.2 CASOS DE USO (BRYEF STYLE)

En esta sección se especifican las funcionalidades del sistema indicadas en el diagrama de casos de uso en formato brief style:

UC01 - Ejecutar Aplicación: El usuario puede iniciar la aplicación, lo que automáticamente configura las diversas funcionalidades del sistema, incluyendo la recopilación, almacenamiento, análisis y visualización de datos de red, para comenzar a utilizar las diferentes funcionalidades del sistema.

UC02 - Configurar NIDS con IA: Al ejecutar la aplicación, el sistema configura automáticamente el sistema de detección de intrusiones en la red que utiliza inteligencia artificial (NIDS-AI). Esta configuración permite al sistema analizar el tráfico de red y detectar posibles amenazas de manera automática.

UC03 - Configurar Zeek: Al ejecutar la aplicación, el sistema configura automáticamente Zeek para monitorizar el tráfico de red y generar registros detallados de la actividad en la red.

UC04 - Recopilar Datos de Red: Zeek recopila datos del tráfico de red en tiempo real y genera registros detallados de las actividades observadas..

UC05 - Configurar Elasticsearch: Al ejecutar la aplicación, el sistema configura automáticamente Elasticsearch para almacenar e indexar los datos recopilados por Filebeat. Esto permite realizar búsquedas y análisis eficientes de grandes volúmenes de datos.

UC06 - Configurar Filebeat: Al ejecutar la aplicación, el sistema configura automáticamente Filebeat para recopilar y enviar datos de registros de diferentes fuentes a Elasticsearch. Esta configuración asegura que los datos de eventos se recopilen adecuadamente.

UC07 - Configurar Kibana: Al ejecutar la aplicación, el sistema configura automáticamente Kibana para visualizar los datos almacenados en Elasticsearch. Kibana ofrece herramientas de visualización y dashboards que facilitan la interpretación de los datos de seguridad y el monitoreo de eventos en tiempo real.

UC08 - Almacenar e Indexar los Datos: Filebeat envía los datos recopilados a Elasticsearch, donde son almacenados e indexados. Este proceso asegura que los datos estén disponibles para consultas y análisis.

UC09 - Analizar los Datos con IA: El sistema NIDS utiliza inteligencia artificial para analizar los datos de tráfico de red almacenados en Elasticsearch. Este análisis permite detectar patrones anómalos y posibles intrusiones.

UC10 - Detectar Posibles Intrusiones: Basado en el análisis realizado con IA, el sistema NIDS detecta posibles intrusiones en la red y genera alertas de seguridad.

UC11 - Generar Alertas de Seguridad: Cuando se detectan posibles intrusiones, el sistema NIDS genera alertas de seguridad para notificar a los usuarios sobre la amenaza potencial.

UC12 - Responder a Posibles Ataques: El usuario y el propio NIDS-AI puede responder a las alertas de seguridad generadas por el sistema, tomando medidas para mitigar las amenazas detectadas y proteger la red.

**UC13 - Visualizar Datos:** Los usuarios pueden utilizar Kibana para visualizar los datos almacenados en Elasticsearch. Esto incluye la creación de dashboards personalizados y la exploración de datos para el análisis de seguridad.

**UC14 - Detener la Aplicación:** El usuario puede detener la aplicación, lo que desactiva las funciones automáticas de recopilación, almacenamiento, análisis y visualización de datos.

### 9.1.3 CASOS DE USO DETALLADOS

<b>Accion</b>	UC01 - Ejecutar Aplicación
<b>Actor Principal</b>	Usuario
<b>Precodicion</b>	El usuario tiene acceso al sistema.
<b>Objetivo</b>	El usuario quiere iniciar la aplicación.
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"> <li>1. El usuario accede a la interfaz de la aplicación.</li> <li>2. El usuario selecciona la opción de "Ejecutar Aplicación".</li> <li>3. La aplicación se inicia y comienza a configurar automáticamente las diversas funcionalidades.</li> <li>4. El usuario recibe una notificación de que la aplicación se ha iniciado correctamente.</li> </ol>	
<b>Extensiones</b>	
<ol style="list-style-type: none"> <li>4a. La aplicación no puede iniciar debido a un error.</li> <li>4a.1. Se muestra un mensaje de error indicando el problema.</li> <li>4a.2. El usuario debe contactar con el soporte técnico o con el administrador.</li> </ol>	

Tabla 12: UC01 - Ejecutar Aplicación. Fuente: Elaboración propia

<b>Accion</b>	UC02 - Configurar NIDS con IA
<b>Actor Principal</b>	Sistema
<b>Precodicion</b>	El sistema tiene los componentes necesarios .
<b>Objetivo</b>	Configurar automáticamente el NIDS-AI al ejecutar la aplicación.
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"> <li>1. El sistema inicia la configuración automática del NIDS-AI.</li> <li>2. El sistema carga los parámetros de configuración predeterminados.</li> <li>3. El sistema NIDS-AI se inicia y comienza a analizar el tráfico de red.</li> <li>4. El sistema NIDS-AI está listo para detectar posibles amenazas.</li> </ol>	
<b>Extensiones</b>	
<ol style="list-style-type: none"> <li>3a. El sistema no puede cargar los parámetros de configuración predeterminados.</li> <li>3a.1. Se muestra un mensaje de error indicando el problema.</li> <li>3a.2. El sistema intenta cargar nuevamente los parámetros de configuración.</li> </ol>	

Tabla 13: UC02 - Configurar NIDS con IA. Fuente: Elaboración propia

<b>Accion</b>	UC03 - Configurar Zeek
<b>Actor Principal</b>	Sistema
<b>Precodicion</b>	El sistema tiene los componentes necesarios para configurar Zeek.
<b>Objetivo</b>	Configurar automáticamente Zeek para monitorizar el tráfico de red.
<b>Escenario principal de éxito</b>	
1. El sistema inicia la configuración automática de Zeek. 2. El sistema carga los parámetros de configuración predeterminados para Zeek. 3. El sistema confirma que Zeek está operativo.	
<b>Extensiones</b>	
3a. El sistema no puede cargar los parámetros de configuración predeterminados. 3a.1. Se muestra un mensaje de error indicando el problema. 3a.2. El sistema intenta cargar nuevamente los parámetros de configuración.	

Tabla 14: UC03 - Configurar Zeek. Fuente: Elaboración propia

<b>Accion</b>	UC04 - Recopilar Datos de Red
<b>Actor Principal</b>	Zeek
<b>Precodicion</b>	Zeek está configurado y operativo.
<b>Objetivo</b>	Recopilar datos del tráfico de red en tiempo real.
<b>Escenario principal de éxito</b>	
1. Zeek comienza a monitorizar el tráfico de red. 2. Zeek genera registros detallados de las actividades de red. 3. Zeek envía los datos a una carpeta. 4. Los datos están disponibles para análisis.	
<b>Extensiones</b>	
3a. Zeek no puede monitorizar el tráfico de red debido a un error. 3a.1. Se muestra un mensaje de error indicando el problema. 3a.2. El sistema intenta reiniciar Zeek para solucionar el problema.	

Tabla 15: UC04 - Recopilar Datos de Red. Fuente: Elaboración propia

<b>Accion</b>	UC05 - Configurar Elasticsearch
<b>Actor Principal</b>	Sistema
<b>Precodicion</b>	El sistema tiene los componentes necesarios para configurar Elasticsearch.
<b>Objetivo</b>	Configurar automáticamente Elasticsearch para almacenar e indexar datos.
<b>Escenario principal de éxito</b>	
1. El sistema inicia la configuración automática de Elasticsearch. 2. El sistema carga los parámetros de configuración predeterminados para Elasticsearch. 3. Elasticsearch se inicializa y está listo para almacenar e indexar datos. 4. El sistema confirma que Elasticsearch está operativo.	

<b>Extensiones</b>	
3a. El sistema no puede cargar los parámetros de configuración predeterminados.	
3a.1. Se muestra un mensaje de error indicando el problema.	
3a.2. El sistema intenta cargar nuevamente los parámetros de configuración.	

Tabla 16: UC05 - Configurar Elasticsearch. Fuente: Elaboración propia

<b>Accion</b>	UC06 - Configurar Filebeat
<b>Actor Principal</b>	Sistema
<b>Precodicion</b>	El sistema tiene los componentes necesarios para configurar Filebeat.
<b>Objetivo</b>	Configurar automáticamente Filebeat para recopilar y enviar datos a Elasticsearch.
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"> <li>1. El sistema inicia la configuración automática de Filebeat.</li> <li>2. El sistema carga los parámetros de configuración predeterminados para Filebeat.</li> <li>3. El sistema confirma que Elasticsearch está operativo.</li> </ol>	
<b>Extensiones</b>	
3a. El sistema no puede cargar los parámetros de configuración predeterminados.	
3a.1. Se muestra un mensaje de error indicando el problema.	
3a.2. El sistema intenta cargar nuevamente los parámetros de configuración.	

Tabla 17: UC06 - Configurar Filebeat. Fuente: Elaboración propia

<b>Accion</b>	UC07 - Configurar Kibana
<b>Actor Principal</b>	Sistema
<b>Precodicion</b>	El sistema tiene los componentes necesarios para configurar Kibana
<b>Objetivo</b>	Configurar automáticamente Kibana para visualizar los datos almacenados en Elasticsearch.
<b>Escenario principal de éxito</b>	
<ol style="list-style-type: none"> <li>1. El sistema inicia la configuración automática de Kibana.</li> <li>2. El sistema carga los parámetros de configuración predeterminados para Kibana.</li> <li>3. Kibana se inicializa y está listo para visualizar datos.</li> <li>4. El sistema confirma que Kibana está operativo.</li> </ol>	
<b>Extensiones</b>	
3a. El sistema no puede cargar los parámetros de configuración predeterminados.	
3a.1. Se muestra un mensaje de error indicando el problema.	
3a.2. El sistema intenta cargar nuevamente los parámetros de configuración.	

Tabla 18: UC07 - Configurar Kibana. Fuente: Elaboración propia

<b>Accion</b>	UC08 - Almacenar e Indexar los Datos
<b>Actor Principal</b>	Filebeat, Elasticsearch
<b>Precodicion</b>	Filebeat y Elasticsearch están configurados y operativos.
<b>Objetivo</b>	Almacenar e indexar los datos recopilados para su análisis.
<b>Escenario principal de éxito</b>	
1. Filebeat recopila datos de registros generados por Zeek. 2. Filebeat envía los datos recopilados a Elasticsearch. 3. Elasticsearch almacena e indexa los datos recibidos. 4. Los datos están disponibles para consultas y análisis.	
<b>Extensiones</b>	
3a. Filebeat no puede enviar datos a Elasticsearch debido a un error. 3a.1. Se muestra un mensaje de error indicando el problema. 3a.2. El sistema intenta reiniciar Filebeat para solucionar el problema.	

Tabla 19: UC08 - Almacenar e Indexar los Datos. Fuente: Elaboración propia

<b>Accion</b>	UC09 - Analizar los Datos con IA
<b>Actor Principal</b>	NIDS - AI
<b>Precodicion</b>	El sistema NIDS-AI está configurado y operativo.
<b>Objetivo</b>	Analizar los datos de tráfico de red para detectar posibles intrusiones.
<b>Escenario principal de éxito</b>	
1. El sistema NIDS-AI accede a los datos almacenados en Elasticsearch. 2. El sistema NIDS-AI analiza los datos utilizando algoritmos de IA.	
<b>Extensiones</b>	
3a. El sistema NIDS-AI encuentra datos incompletos o corruptos. 3a.1. Se muestra un mensaje de error indicando el problema. 3a.2. El sistema intenta acceder nuevamente a los datos para completar el análisis.	

Tabla 20: UC09 - Analizar los Datos con IA. Fuente: Elaboración propia

<b>Accion</b>	UC10 - Detectar Posibles Intrusiones
<b>Actor Principal</b>	NIDS - AI
<b>Precodicion</b>	El sistema NIDS-AI está configurado y operativo.
<b>Objetivo</b>	Detectar posibles intrusiones en la red basadas en el análisis de datos.
<b>Escenario principal de éxito</b>	
1. El sistema NIDS-AI analiza continuamente los datos de tráfico de red. 2. El sistema detecta una posible intrusión basada en patrones anómalos.	
<b>Extensiones</b>	

- |   |
|---|
| 3a. El sistema NIDS-AI detecta un falso positivo.   |
| 3a.1. El usuario revisa la alerta de seguridad y determina que no es una amenaza.               |
| 3a.2. El sistema ajusta los algoritmos de detección para reducir falsos positivos en el futuro. |

Tabla 21: UC10 - Detectar Posibles Intrusiones. Fuente: Elaboración propia

<b>Accion</b>	UC11 - Generar Alertas de Seguridad
<b>Actor Principal</b>	NIDS - AI
<b>Precodicion</b>	El sistema NIDS-AI está configurado y operativo.
<b>Objetivo</b>	Notificar al usuario sobre posibles intrusiones.
<b>Escenario principal de éxito</b>	
1. El sistema NIDS-AI detecta una posible intrusión. 2. El sistema genera una alerta de seguridad. 3. El sistema envía la alerta al usuario.	
<b>Extensiones</b>	
3a. El sistema no puede enviar la alerta debido a un problema de conectividad. 3a.1. Se muestra un mensaje de error indicando el problema. 3a.2. El sistema reintenta enviar la alerta.	

Tabla 22: UC11 - Generar Alertas de Seguridad. Fuente: Elaboración propia

<b>Accion</b>	UC12 - Responder a Posibles Ataques
<b>Actor Principal</b>	Usuario, NIDS - AI
<b>Precodicion</b>	El sistema ha generado una alerta de seguridad.
<b>Objetivo</b>	Mitigar las amenazas detectadas y proteger la red.
<b>Escenario principal de éxito</b>	
1. El usuario recibe una alerta de seguridad. 2. El usuario revisa los detalles de la alerta. 3. El usuario toma medidas para mitigar la amenaza (p. e. bloquear una IP, ...). 4. El sistema NIDS-AI también puede tomar medidas automáticas para mitigar la amenaza. 5. La amenaza es neutralizada y la red permanece segura.	
<b>Extensiones</b>	
3a. El usuario no está disponible para responder a la alerta. 3a.1. El sistema NIDS-AI toma medidas automáticas basadas en la configuración predeterminada. 3a.2. El usuario recibe un informe detallado de las acciones tomadas por el sistema.	

Tabla 23: UC12 - Responder a Posibles Ataques. Fuente: Elaboración propia

<b>Accion</b>	UC13 - Visualizar Datos
<b>Actor Principal</b>	Usuario
<b>Precodicion</b>	Kibana está configurado y operativo.
<b>Objetivo</b>	Visualizar y analizar los datos de seguridad almacenados en Elasticsearch.
<b>Escenario principal de éxito</b>	
1. El usuario accede a la interfaz de Kibana. 2. El usuario selecciona los dashboards y visualizaciones predefinidas o crea nuevas. 3. Kibana muestra los datos almacenados en Elasticsearch en forma de gráficos y tablas. 4. El usuario analiza los datos y obtiene insights sobre la seguridad de la red.	
<b>Extensiones</b>	
3a. Kibana no puede acceder a los datos debido a un problema de conectividad. 3a.1. Se muestra un mensaje de error indicando el problema. 3a.2. El usuario intenta reconectar Kibana con Elasticsearch.	

Tabla 24: UC13 - Visualizar Datos. Fuente: Elaboración propia

<b>Accion</b>	UC14 - Detener la Aplicación
<b>Actor Principal</b>	Usuario
<b>Precodicion</b>	El usuario tiene acceso a la aplicación en ejecución.
<b>Objetivo</b>	Detener la aplicación y desactivar las funciones automáticas.
<b>Escenario principal de éxito</b>	
1. El usuario accede a la interfaz de la aplicación. 2. El usuario selecciona la opción de "Detener Aplicación". 3. La aplicación comienza el proceso de cierre. 4. El sistema detiene la recopilación, almacenamiento, análisis y visualización de datos. 5. El sistema confirma que la aplicación se ha detenido correctamente.	
<b>Extensiones</b>	
3a. La aplicación no puede detenerse debido a un error. 3a.1. Se muestra un mensaje de error indicando el problema. 3a.2. El usuario puede optar por intentar nuevamente o contactar con el soporte técnico.	

Tabla 25: UC14 - Detener la Aplicación. Fuente: Elaboración propia

## 9.2 REQUISITOS NO FUNCIONALES

Para garantizar el éxito del proyecto, es crucial no solo cumplir con los requisitos funcionales, sino también satisfacer los requisitos no funcionales. Estos últimos abarcan aspectos críticos que van más allá de las funcionalidades directas del sistema y están diseñados para asegurar su desempeño, seguridad, escalabilidad y usabilidad. En el contexto específico de este proyecto, los requisitos no funcionales que se han identificado como prioritarios incluyen:

<b>Requisito</b>	RNF01 - Fàcil de utilitzar
<b>Descripción</b>	La aplicación debe ser fácil de usar para aquellos que no tienen conocimientos técnicos.
<b>Justificación</b>	La aplicación es muy compleja para que un usuario sin conocimientos sepa ejecutar y configurar cada servicio.
<b>Criterios de aceptación</b>	
	El sistema debe garantizar que el 90% de los usuarios, pasado una semana de pruebas, sepan utilizar el sistema sin dificultad.

Tabla 26: RNF01 - Fàcil de utilitzar. Fuente: Elaboración propia

<b>Requisito</b>	RNF02 - Estilo
<b>Descripción</b>	La aplicación debe producir sensación de seguridad
<b>Justificación</b>	La aplicación monitorea todo el tráfico de red y tiene como finalidad gestionar intrusiones en la red.
<b>Criterios de aceptación</b>	
	Mínimo el 75% de los usuarios de la aplicación deben tener esas sensación de seguridad y confianza en la aplicación.

Tabla 27: RNF02 - Estilo. Fuente: Elaboración propia

<b>Requisito</b>	RNF03 - Comprensión y cortesía
<b>Descripción</b>	El sistema debe utilizar símbolos y palabras comprensibles por el usuario.
<b>Justificación</b>	Al sistema pueden acceder personas con conocimientos técnicos y personas sin conocimientos.
<b>Criterios de aceptación</b>	
	El 95% dels usuaris han d'entendre tots els textos i icones de l'aplicació.

Tabla 28: RNF03 - Comprensión y cortesía. Fuente: Elaboración propia

<b>Requisito</b>	RNF04 - Precisión
<b>Descripción</b>	La aplicación muestra con precisión las posibles intrusiones de red.
<b>Justificación</b>	El sistema trabaja detectando intrusiones, es por eso que debe ser preciso a la hora de detectarlas para que no se cometa ningún falso positivo.
<b>Criterios de aceptación</b>	
	El sistema deberá tener mínimo un 90% de probabilidad de detectar un verdadero positivo y/o un verdadero negativo.

Tabla 29: RNF04 - Precisión. Fuente: Elaboración propia

<b>Requisito</b>	RNF05 - Compatibilidad
<b>Descripcion</b>	La aplicación debe poder ser usada en cualquier sistema.
<b>Justificacion</b>	La aplicación al ser tan compleja, debe poder ser ejecutada en cualquier sistema operativo.
<b>Criterios de aceptación</b>	
El 100% del sistema debe poder ser ejecutado tanto en windows como en linux.	

Tabla 30: RNF05 - Compatibilidad. Fuente: Elaboración propia

<b>Requisito</b>	RNF06 - Integridad
<b>Descripcion</b>	La aplicación debe garantizar que el usuario no puede manipularla ni introducir datos erróneos.
<b>Justificacion</b>	La aplicación al ser tan compleja, debe poder ser ejecutada en cualquier sistema operativo.
<b>Criterios de aceptación</b>	
El sistema debe garantizar en un 90% que los usuarios no introduzcan datos erróneos.	

Tabla 31: RNF06 - Integridad. Fuente: Elaboración propia

<b>Requisito</b>	RNF10 - Privacidad
<b>Descripcion</b>	La aplicación debe garantizar la privacidad de los datos que obtiene del usuario.
<b>Justificacion</b>	El sistema debe cumplir la ley de protección de datos
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>- El producto debe informar al usuario sobre el uso de los datos obtenidos.</li> <li>- El sistema no puede compartir los datos privados de los usuarios</li> <li>- El sistema notificará al usuario en caso de cambiar la política.</li> </ul>	

Tabla 32: RNF10 - Privacidad. Fuente: Elaboración propia

<b>Requisito</b>	RNF10 - Legalidad
<b>Descripcion</b>	La aplicación debe garantizar el cumplimiento de las leyes.
<b>Justificacion</b>	El sistema debe cumplir las leyes del país o comunidad donde se encuentra su mercado.
<b>Criterios de aceptación</b>	
- El sistema debe garantizar el cumplimiento de la RGPD.	

Tabla 33: RNF10 - Legalidad. Fuente: Elaboración propia

## 10. SISTEMA DE DETECCIÓN DE INTRUSIONES EN RED (NIDS)

Para el desarrollo de un Sistema de Detección de Intrusiones en Red (NIDS), se ha seguido un procedimiento detallado destinado a reforzar la seguridad en un entorno local. Este NIDS se basa en modelos de aprendizaje automático, por lo que es esencial contar con datos adecuados para entrenar dichos modelos. Como se explicó en la sección teórica del informe, se emplea el conjunto de datos UNSW-NB15 .

La primera etapa del desarrollo consiste en el preprocesamiento de los datos, asegurando su calidad y pertinencia para el entrenamiento de los modelos. Esto implica la limpieza de datos para eliminar registros irrelevantes o erróneos, el equilibrio de datos para asegurar la uniformidad en las escalas del dataset,y la selección de características relevantes que influirán directamente en la eficacia de los modelos de aprendizaje automático.

En la siguiente etapa, se procede a la ejecución de diversos modelos de inteligencia artificial y al análisis de sus resultados. Esto permite identificar el modelo que ofrece el mejor rendimiento en términos de precisión, sensibilidad y especificidad.

Finalmente, se implementa el modelo seleccionado en una aplicación visual. Esta aplicación, junto con la recolección de datos en tiempo real proporcionada por Zeek, permite visualizar los resultados tanto en un dashboard como en un historial. De esta manera, se pueden evaluar los datos recogidos en tiempo real, proporcionando una herramienta eficaz para la detección y monitoreo de intrusiones en la red.

Tanto la primera como la segunda etapa se han llevado a cabo en Google Colab, utilizando un notebook para toda la preparación previa de los datos y el entrenamiento de los modelos. Esta plataforma ha permitido realizar el preprocesamiento de datos y la ejecución de diversos modelos de aprendizaje automático de manera eficiente y colaborativa.

Para las etapas posteriores, el trabajo se ha realizado de forma local, utilizando un conjunto de herramientas robustas y versátiles. Entre estas herramientas se incluyen Docker para la gestión y despliegue de contenedores, Visual Studio Code para el desarrollo y depuración del código, Python para la implementación de los modelos y scripts, Zeek para la recolección de datos en tiempo real, y Elastic Stack para la visualización y análisis de los resultados. Este enfoque ha permitido una integración fluida y una implementación eficiente del Sistema de Detección de Intrusiones en Red (NIDS) en un entorno real.

### 10.1 PROCESO DE GENERACIÓN DE DATASETS

Durante el preprocesamiento de los datos, se realizarán varias tareas críticas, incluyendo la limpieza de los datos para eliminar registros irrelevantes o erróneos, el equilibrio de datos para asegurar la uniformidad en las escalas del dataset, y la selección de características relevantes que influirán directamente en la eficacia de los modelos de aprendizaje automático. Esta fase es fundamental para garantizar que los modelos reciban información precisa y útil, lo que maximizará su capacidad para detectar intrusiones con alta precisión.

Posteriormente, los datos preprocesados se dividirán en conjuntos de entrenamiento y prueba, permitiendo evaluar el rendimiento de los modelos desarrollados. Se utilizará la técnicas de validación cruzada para optimizar los modelos y evitar el sobreajuste, asegurando que el NIDS tenga un rendimiento robusto y generalizable a diferentes escenarios de red.

### 10.1.1 PREPROCESAMIENTO

El conjunto de datos UNSW-NB15 se distribuye en 23 archivos en formato CSV. Cada archivo contiene datos de conexiones de red malignos y benignos. La tabla 34 describe metadatos del contenido de estos archivos.

Nombre del archivo	Tamaño	Registros
Network_dataset_1.csv	139,9 MB	1.000.000
Network_dataset_2.csv	138,8 MB	1.000.000
Network_dataset_3.csv	138,9 MB	1.000.000
Network_dataset_4.csv	138,5 MB	1.000.000
Network_dataset_5.csv	138,5 MB	1.000.000
Network_dataset_6.csv	158,1 MB	1.000.000
Network_dataset_7.csv	140,4 MB	1.000.000
Network_dataset_8.csv	139,8 MB	1.000.000
Network_dataset_9.csv	139,7 MB	1.000.000
Network_dataset_10.csv	139,7 MB	1.000.000
Network_dataset_11.csv	141,6 MB	1.000.000
Network_dataset_12.csv	146,5 MB	1.000.000
Network_dataset_13.csv	144 MB	1.000.000
Network_dataset_14.csv	146,2 MB	1.000.000
Network_dataset_15.csv	146,1 MB	1.000.000
Network_dataset_16.csv	145,9 MB	1.000.000
Network_dataset_17.csv	142,9 MB	1.000.000
Network_dataset_18.csv	143,8 MB	1.000.000
Network_dataset_19.csv	149,7 MB	1.000.000
Network_dataset_20.csv	150,3 MB	1.000.000
Network_dataset_21.csv	151,3 MB	1.000.000
Network_dataset_22.csv	146,2 MB	1.000.000
Network_dataset_23.csv	49,3 MB	339.021
<b>TOTAL</b>	<b>3216,1 MB</b>	<b>22.339.021</b>

Tabla 34: Descripción de los datasets. Fuente: Elaboración propia

Para cada uno de los conjuntos de datos, se verifica que no contengan valores vacíos. Además, se contabiliza la cantidad de filas que contienen datos benignos y malignos, así como la cantidad total de filas presentes en cada conjunto de datos y en el conjunto total de datasets. Estos datos se pueden observar en la Figura 39.

Sistema de Detección de Intrusiones en Ciberseguridad con Inteligencia Artificial  
 Francisco Jose Martin Aguilar

DF 1 :	DF 7 :	DF 13 :	DF 19 :
Existen NaN : False Cantidad de rows con label 0: 208679 Cantidad de rows con label 1: 791321 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 7499 Cantidad de rows con label 1: 992501 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 917 Cantidad de rows con label 1: 999083 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 12708 Cantidad de rows con label 1: 987292 Cantidad de rows totales: 1000000
DF 2 :	DF 8 :	DF 14 :	DF 20 :
Existen NaN : False Cantidad de rows con label 0: 5717 Cantidad de rows con label 1: 994283 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 21306 Cantidad de rows con label 1: 978694 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 583 Cantidad de rows con label 1: 999417 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 21082 Cantidad de rows con label 1: 978918 Cantidad de rows totales: 1000000
DF 3 :	DF 9 :	DF 15 :	DF 21 :
Existen NaN : False Cantidad de rows con label 0: 2820 Cantidad de rows con label 1: 997180 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 24739 Cantidad de rows con label 1: 975261 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 1060 Cantidad de rows con label 1: 998940 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 14407 Cantidad de rows con label 1: 985593 Cantidad de rows totales: 1000000
DF 4 :	DF 10 :	DF 16 :	DF 22 :
Existen NaN : False Cantidad de rows con label 0: 6256 Cantidad de rows con label 1: 993744 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 30002 Cantidad de rows con label 1: 969998 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 1891 Cantidad de rows con label 1: 998109 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 234988 Cantidad de rows con label 1: 765012 Cantidad de rows totales: 1000000
DF 5 :	DF 11 :	DF 17 :	DF 23 :
Existen NaN : False Cantidad de rows con label 0: 3657 Cantidad de rows con label 1: 996343 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 35168 Cantidad de rows con label 1: 964832 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 33711 Cantidad de rows con label 1: 966289 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 33475 Cantidad de rows con label 1: 305546 Cantidad de rows totales: 339021
DF 6 :	DF 12 :	DF 18 :	Total Rows in all DFs : 22339021
Existen NaN : False Cantidad de rows con label 0: 13473 Cantidad de rows con label 1: 986527 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 32806 Cantidad de rows con label 1: 967194 Cantidad de rows totales: 1000000	Existen NaN : False Cantidad de rows con label 0: 49436 Cantidad de rows con label 1: 950564 Cantidad de rows totales: 1000000	

Figura 39: Datos vacíos e información de los Datasets. Fuente: Elaboración propia

Para garantizar un entrenamiento correcto de los modelos de inteligencia artificial, se ha verificado que los conjuntos de datos no contengan valores duplicados. Los datos duplicados pueden sesgar el entrenamiento de los modelos, haciendo que estos favorezcan ciertos patrones sobre otros de manera indebida.

Aunque se reconoce que los duplicados existen y que el proyecto requiere mantenerlos debido a la alta probabilidad de que los datos en tiempo real se dupliquen con frecuencia, se ha decidido eliminar los duplicados para equilibrar la representación de las clases benignas y malignas. Este desequilibrio podría causar que el modelo aprenda mejor a identificar ataques, pero no a reconocer adecuadamente los comportamientos benignos ya que en su totalidad existen más datos de entrenamientos malignos que benignos.

Por esta razón, se ha optado por eliminar los datos duplicados con el objetivo de lograr un entrenamiento más equilibrado y preciso del modelo. Este enfoque busca mejorar la capacidad del modelo para identificar tanto las intrusiones como el tráfico benigno, obteniendo un sistema de detección de intrusiones más robusto y confiable. En la Figura 40, se pueden observar los datos duplicados presentes en los datasets, junto con la categoría de las filas duplicadas. Los datasets que contienen datos duplicados se han procesado para eliminar estas filas. El resto de los datasets no presentan duplicados.

DF : 1	DF : 10
Existen duplicados : True	Existen duplicados : True
Cantidad de rows duplicadas: 6425	Cantidad de rows duplicadas: 446731
Cantidad de rows duplicadas con label 0: 6425	Cantidad de rows duplicadas con label 0: 0
Cantidad de rows duplicadas con label 1: 0	Cantidad de rows duplicadas con label 1: 446731
DF : 2	DF : 11
Existen duplicados : True	Existen duplicados : True
Cantidad de rows duplicadas: 1	Cantidad de rows duplicadas: 379868
Cantidad de rows duplicadas con label 0: 0	Cantidad de rows duplicadas con label 0: 0
Cantidad de rows duplicadas con label 1: 1	Cantidad de rows duplicadas con label 1: 379868
DF : 4	DF : 13
Existen duplicados : True	Existen duplicados : True
Cantidad de rows duplicadas: 2	Cantidad de rows duplicadas: 2
Cantidad de rows duplicadas con label 0: 0	Cantidad de rows duplicadas con label 0: 0
Cantidad de rows duplicadas con label 1: 2	Cantidad de rows duplicadas con label 1: 2
DF : 7	DF : 18
Existen duplicados : True	Existen duplicados : True
Cantidad de rows duplicadas: 1	Cantidad de rows duplicadas: 82113
Cantidad de rows duplicadas con label 0: 1	Cantidad de rows duplicadas con label 0: 0
Cantidad de rows duplicadas con label 1: 0	Cantidad de rows duplicadas con label 1: 82113
DF : 8	DF : 22
Existen duplicados : True	Existen duplicados : True
Cantidad de rows duplicadas: 278957	Cantidad de rows duplicadas: 141289
Cantidad de rows duplicadas con label 0: 0	Cantidad de rows duplicadas con label 0: 99654
Cantidad de rows duplicadas con label 1: 278957	Cantidad de rows duplicadas con label 1: 41635
DF : 9	DF : 23
Existen duplicados : True	Existen duplicados : True
Cantidad de rows duplicadas: 453801	Cantidad de rows duplicadas: 2717
Cantidad de rows duplicadas con label 0: 0	Cantidad de rows duplicadas con label 0: 1021
Cantidad de rows duplicadas con label 1: 453801	Cantidad de rows duplicadas con label 1: 1696

Figura 40: Comprobación de duplicados en los Datasets. Fuente: Elaboración propia

Además de verificar los duplicados dentro de cada dataset, también se han tenido en cuenta los duplicados que pueden existir entre diferentes datasets. Dado que los datos se recogen de manera temporal, es posible que haya duplicados entre un dataset y el siguiente.

Para abordar esto, se han buscado e identificado los últimos 100 elementos de cada dataset y se han comparado con los primeros 100 elementos del dataset siguiente, para detectar las filas que podrían estar duplicadas entre ellos. Se ha elegido este valor de 100 porque temporalmente los 100 últimos elementos de un dataset y los 100 primeros del siguiente tienden a tener el mismo timestamp de recolección de datos, mientras que a partir de estas filas los tiempos de recolección difieren.

En la Figura 41, se muestran los datasets que contienen filas duplicadas entre ellos, así como la cantidad de duplicados identificados.

```
DF_START: 8; DF_END: 9
Existen duplicados : True
Cantidad de rows duplicadas: 18
Cantidad de rows duplicadas con label 0: 0
Cantidad de rows duplicadas con label 1: 18

DF_START: 9; DF_END: 10
Existen duplicados : True
Cantidad de rows duplicadas: 33
Cantidad de rows duplicadas con label 0: 0
Cantidad de rows duplicadas con label 1: 33

DF_START: 10; DF_END: 11
Existen duplicados : True
Cantidad de rows duplicadas: 10
Cantidad de rows duplicadas con label 0: 0
Cantidad de rows duplicadas con label 1: 10
```

Figura 41: Comprobación de duplicados entre Datasets contiguos. Fuente: Elaboración propia

Para cada uno de estos datos duplicados, se ha procedido a eliminar las filas duplicadas únicamente de un dataset. Para mantener la integridad y consistencia de los datos, estas filas duplicadas se han eliminado del primer dataset correspondiente.

Una vez comprobado que no haya datos vacíos y eliminado los duplicados de los conjuntos de datos, se procedió a verificar la consistencia de los tipos de datos en cada columna. Es decir, se comprobó que cada columna mantuviera el tipo de dato adecuado entre todos los datasets. Por ejemplo, si una columna debía contener enteros, se verificó que en todos los datasets esa columna efectivamente contuviera sólo enteros.

Durante esta verificación, se identificó que la columna "src\_bytes" presentaba inconsistencias en algunos casos donde la conexión utilizaba el protocolo ICMP. En estos casos, el campo contenía cadenas de texto con la dirección IP de destino, mientras que en los casos donde el tipo de conexión no era ICMP, el campo "src\_bytes" contenía valores del tipo int64.

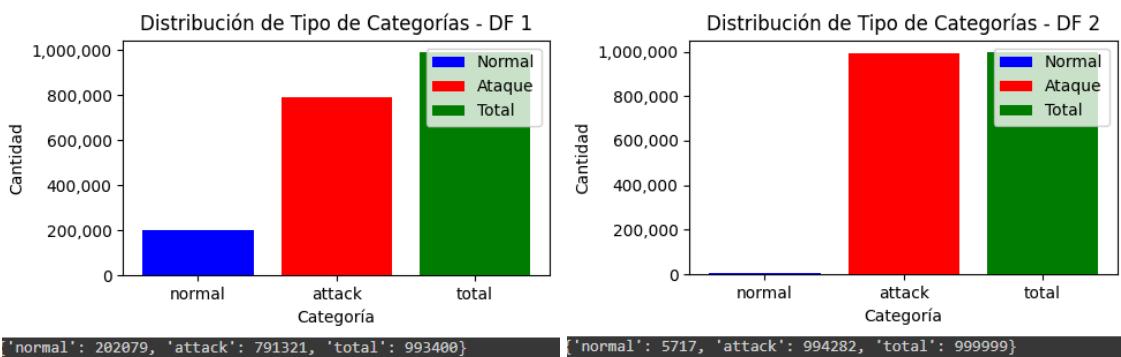
```
DataFrame 1: Type of 'src_bytes' column: object
DataFrame 2: Type of 'src_bytes' column: int64
DataFrame 3: Type of 'src_bytes' column: int64
DataFrame 4: Type of 'src_bytes' column: int64
DataFrame 5: Type of 'src_bytes' column: int64
DataFrame 6: Type of 'src_bytes' column: int64
DataFrame 7: Type of 'src_bytes' column: int64
DataFrame 8: Type of 'src_bytes' column: int64
DataFrame 9: Type of 'src_bytes' column: int64
DataFrame 10: Type of 'src_bytes' column: int64
DataFrame 11: Type of 'src_bytes' column: int64
DataFrame 12: Type of 'src_bytes' column: int64
DataFrame 13: Type of 'src_bytes' column: int64
DataFrame 14: Type of 'src_bytes' column: int64
DataFrame 15: Type of 'src_bytes' column: int64
DataFrame 16: Type of 'src_bytes' column: int64
DataFrame 17: Type of 'src_bytes' column: int64
DataFrame 18: Type of 'src_bytes' column: int64
DataFrame 19: Type of 'src_bytes' column: int64
DataFrame 20: Type of 'src_bytes' column: int64
DataFrame 21: Type of 'src_bytes' column: int64
DataFrame 22: Type of 'src_bytes' column: object
DataFrame 23: Type of 'src_bytes' column: object
```

Figura 42: Inconsistencias en tipo de la columna src\_bytes. Fuente: Elaboración propia

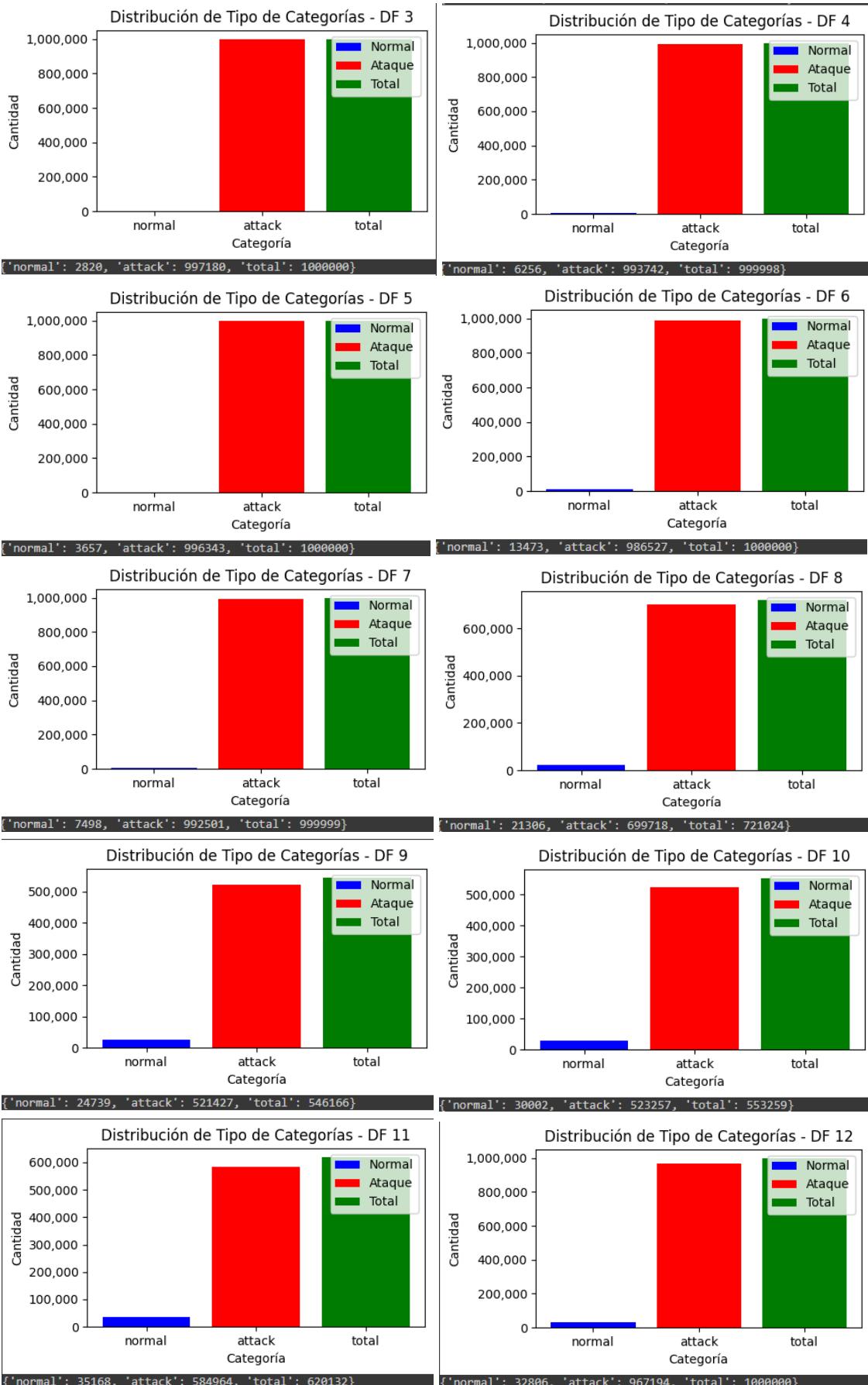
Para mantener la coherencia y precisión de los datos, se decidió depurar las filas afectadas por estas inconsistencias, eliminándolas de los datasets.

Eliminar datos vacíos, duplicados e inconsistencias en los datasets ofrece numerosos beneficios cruciales. En primer lugar, mejora la precisión y fiabilidad de los modelos de inteligencia artificial, ya que elimina el ruido y los sesgos introducidos por valores nulos o duplicados. Además, reduce el tamaño del dataset, lo que optimiza el uso de recursos computacionales y acelera el procesamiento. Los datos limpios también aseguran que los análisis y predicciones sean más coherentes y fiables, facilitando la interpretación de patrones y tendencias. Finalmente, mantener la integridad y consistencia de los datos aumenta la confianza en los resultados, reduciendo la probabilidad de errores en la implementación y mejorando la toma de decisiones estratégicas y operativas. En resumen, la limpieza de datos es esencial para garantizar la calidad y eficiencia en el análisis de datos y el entrenamiento de modelos de inteligencia artificial.

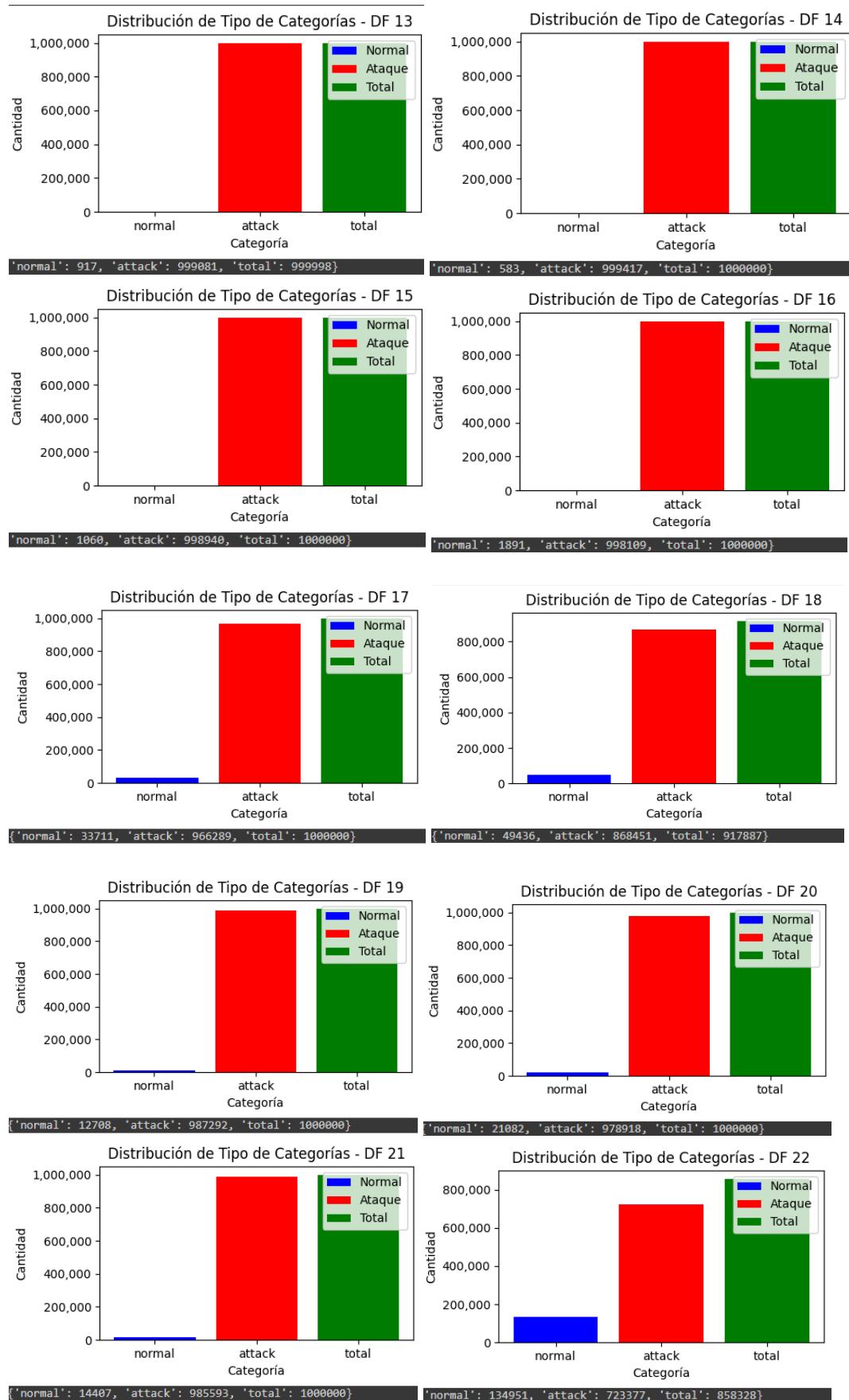
A partir de este punto, se procedió a analizar la distribución de las diferentes categorías dentro de cada dataset para evaluar su equilibrio. Mediante los gráficos presentados en la Figura 43, se observó que existen una gran cantidad de categorías de intrusión y muy pocas de conexiones benignas.



Sistema de Detección de Intrusiones en Ciberseguridad con Inteligencia Artificial  
 Francisco Jose Martin Aguilar



Sistema de Detección de Intrusiones en Ciberseguridad con Inteligencia Artificial  
 Francisco Jose Martin Aguilar



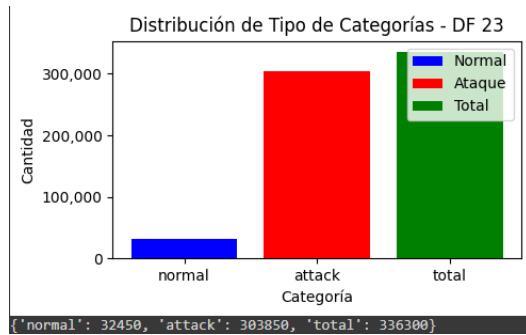


Figura 43: Distribución en cada dataset del campo label. Fuente: Elaboración propia

Como se puede observar en la Figura 43, los valores están totalmente desequilibrados en cada uno de los datasets, lo cual afecta negativamente el entrenamiento de los modelos de inteligencia artificial. En la Figura 44, se muestra de manera general la distribución de clases en el conjunto completo de datasets, evidenciando el desbalance entre las categorías de intrusión y las conexiones benignas.

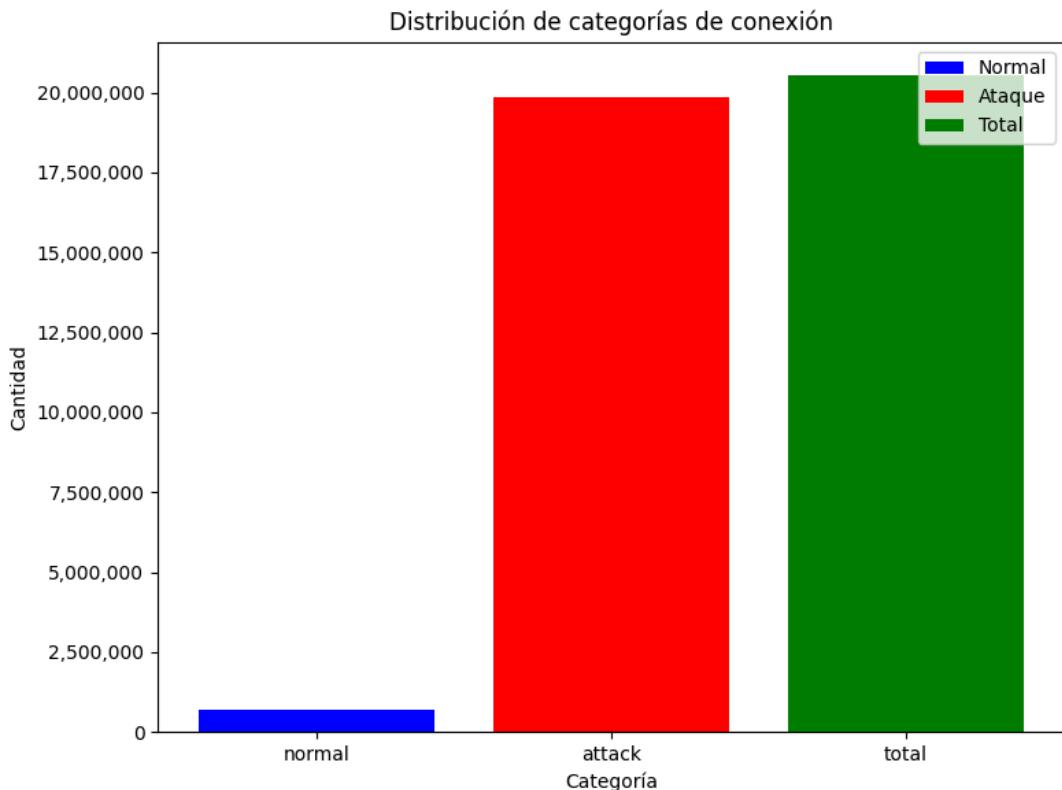


Figura 44: Distribución del conjunto de datasets del campo label. Fuente: Elaboración propia

Categoría	Registros
Datos normales	688.717
Datos de ataque	19.857.773
Total de registros	20.546.490

Tabla 35: Detalle del conjunto de datasets por label. Fuente: Elaboración propia

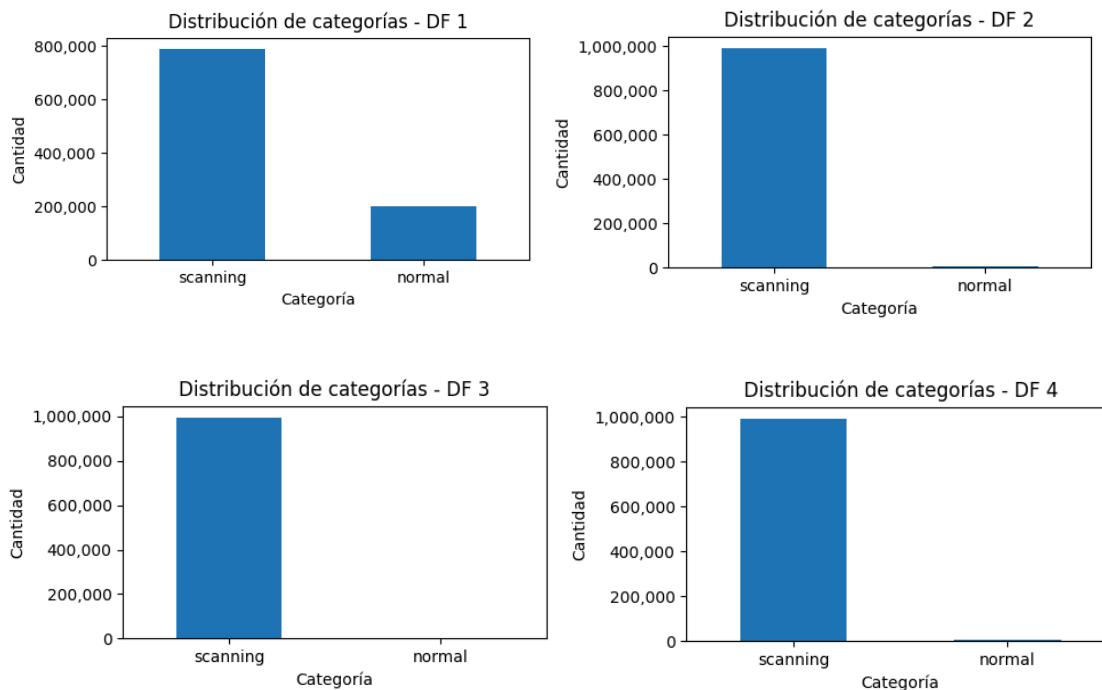
Este desbalance significativo, donde los datos de ataque superan ampliamente a los datos normales, presenta varios desafíos para el entrenamiento de modelos de inteligencia artificial.

En primer lugar, un modelo entrenado en un dataset tan desequilibrado puede desarrollar un sesgo hacia la detección de ataques, ya que estos representan la mayoría de los datos. Esto puede llevar a una alta tasa de falsos positivos, donde el modelo clasifica incorrectamente datos normales como ataques. Esta situación es indeseable, especialmente en un entorno de producción, ya que podría causar alarmas innecesarias y una carga de trabajo adicional para los administradores de red.

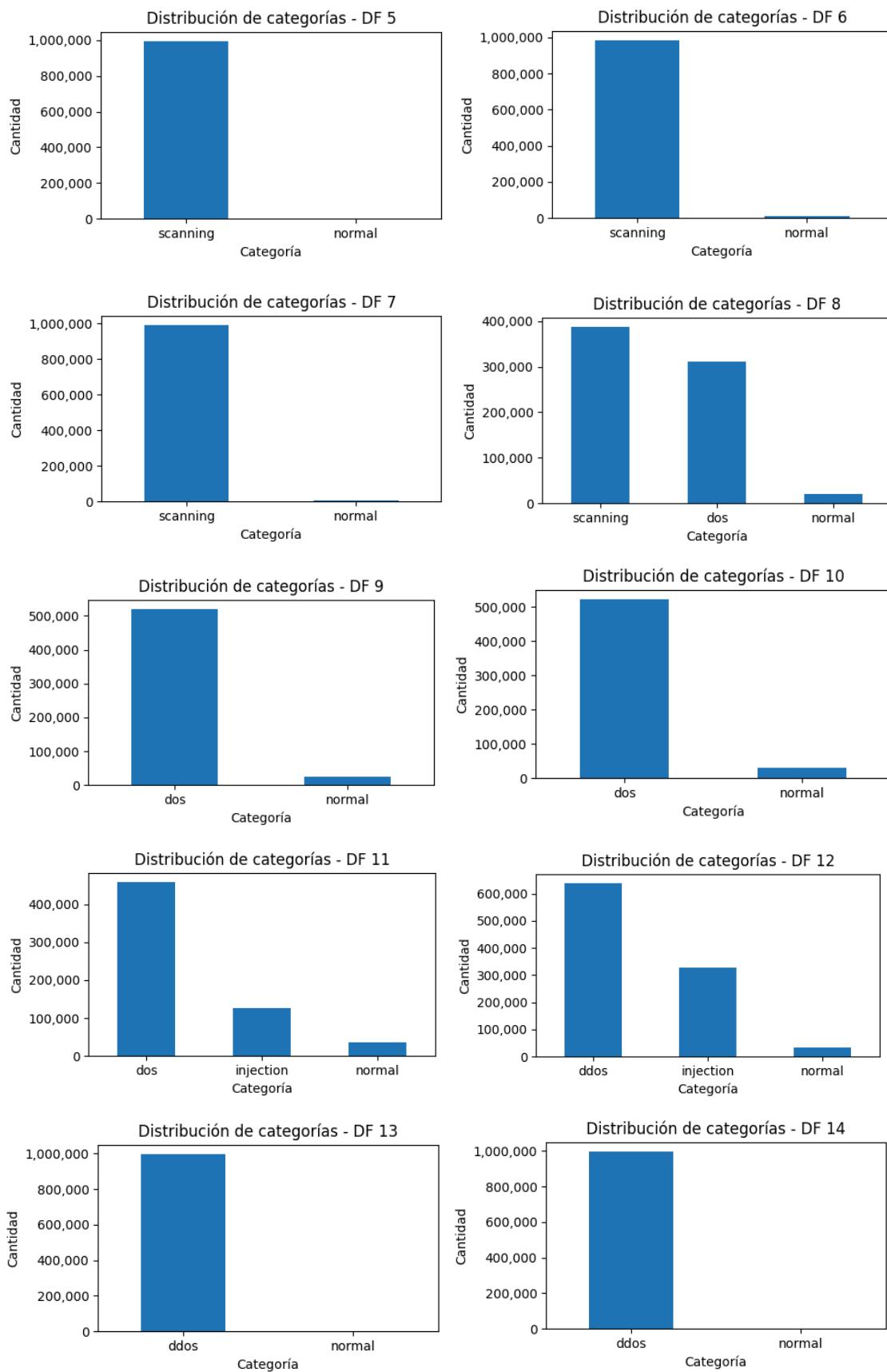
En segundo lugar, la escasez de datos normales limita la capacidad del modelo para aprender y reconocer patrones de tráfico benigno. Un modelo efectivo debe ser capaz de distinguir con precisión entre tráfico normal y malicioso. Con tan pocos ejemplos de tráfico normal, el modelo puede tener dificultades para generalizar adecuadamente y reconocer comportamientos benignos en situaciones reales.

A continuación, se ha llevado a cabo una evaluación detallada de la distribución de los diferentes tipos de categorías presentes en los datasets. Estas categorías incluyen tanto las diversas formas de ataques maliciosos, como scanning, fuzzing, entre otros, así como la categoría benigna denominada "normal".

Esta evaluación detallada de los tipos de categorías presentes en los datasets se pueden ver reflejadas en la figura 45. Los resultados de este análisis permiten entender mejor el desequilibrio existente entre las clases y cómo puede afectar el entrenamiento y rendimiento de los modelos de inteligencia artificial.



Sistema de Detección de Intrusiones en Ciberseguridad con Inteligencia Artificial  
Francisco Jose Martin Aguilar



Sistema de Detección de Intrusiones en Ciberseguridad con Inteligencia Artificial  
Francisco Jose Martin Aguilar

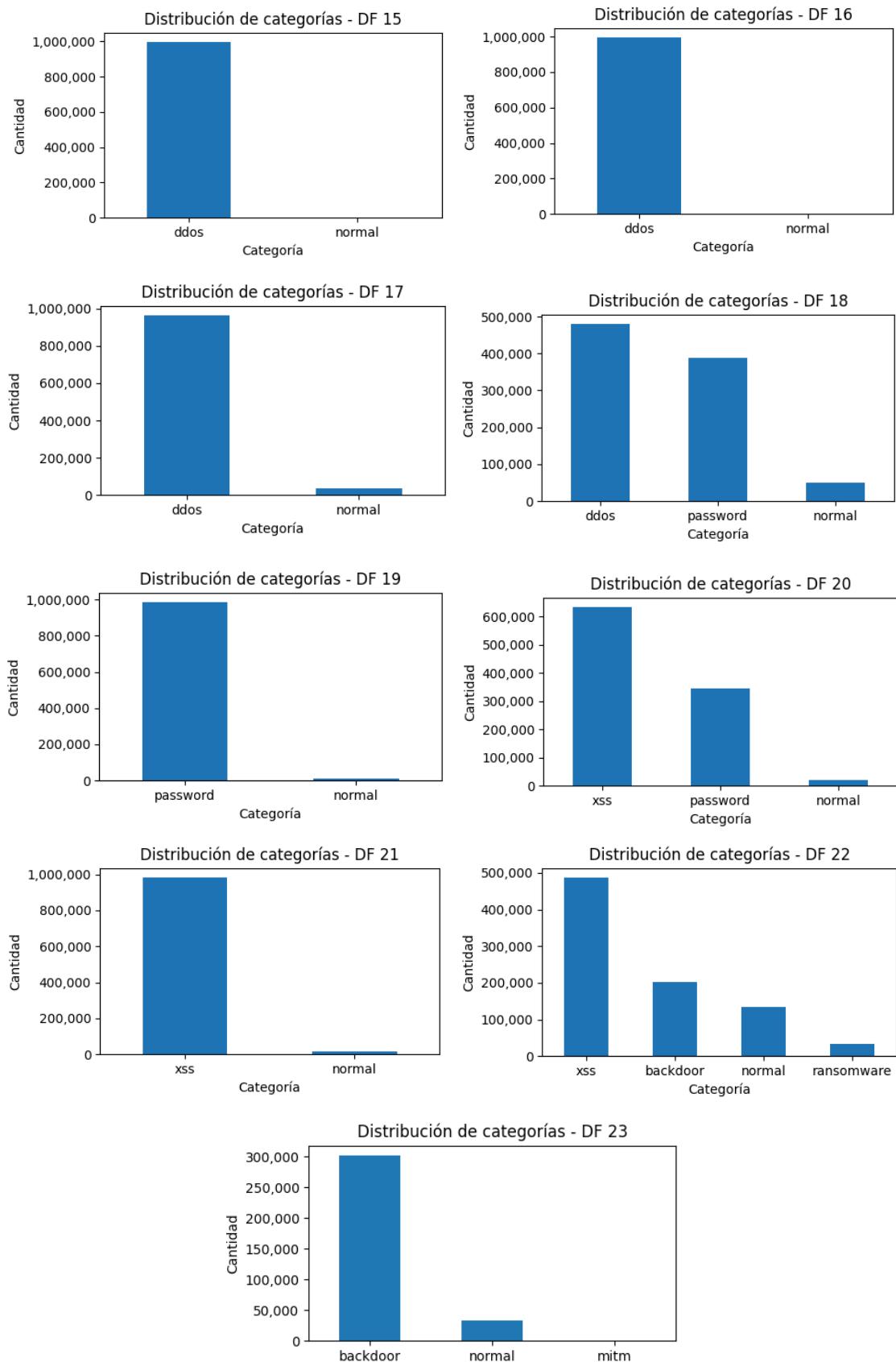


Figura 45: Distribución de los tipos de categoría en cada dataset. Fuente: Elaboración propia

Como se puede observar en la Figura 45, los valores están totalmente desequilibrados en cada uno de los datasets, lo cual afecta negativamente el entrenamiento de los modelos de inteligencia artificial. En la Figura 46, se muestra de manera general la distribución de clases en el conjunto completo de datasets, evidenciando el desbalance entre las categorías de intrusión y las conexiones benignas.

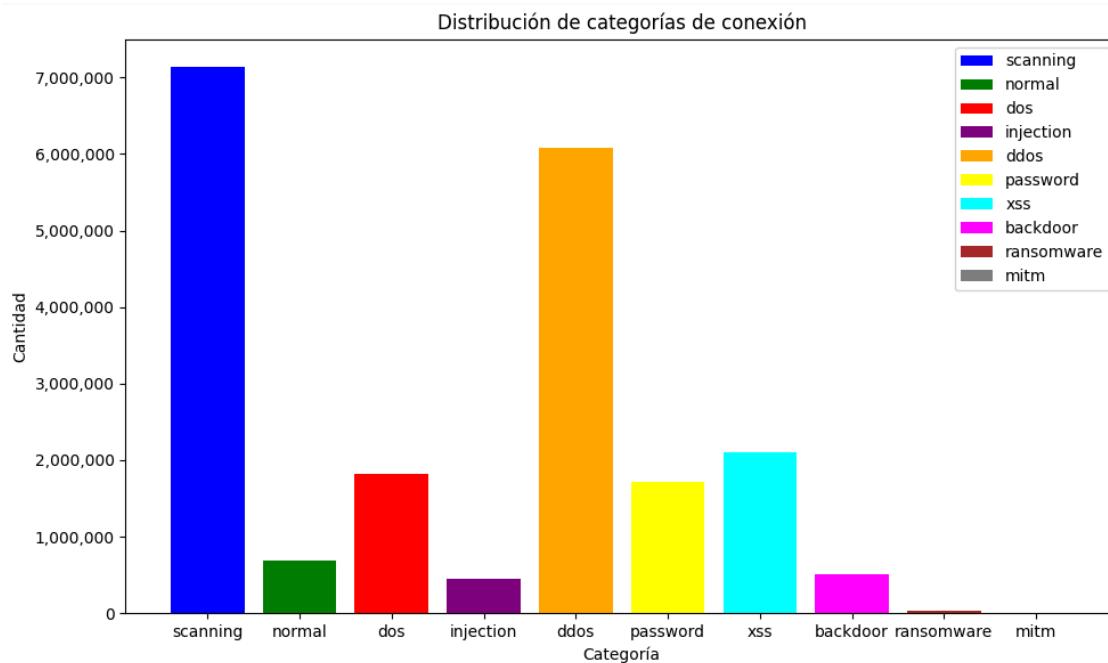


Figura 46: Distribución de los tipos de categoría en los datasets. Fuente: Elaboración propia

Al analizar la distribución de los tipos de categorías dentro de los datasets, se observa un marcado desequilibrio entre las clases de datos. Específicamente, la distribución general de los datos es la siguiente:

Categoría	Registros
Datos normales	688.717
Scanning	7.140.158
Dos	1.815.909
Injection	452.659
DDOS	6.082.893
Password	1.718.568
XSS	2.108.944
Backdoor	505.385
ransomware	32.214
Mitm	1.043
<b>Total</b>	<b>20.546.490</b>

Tabla 36: Detalles de los tipos de categoría en los datasets. Fuente: Elaboración propia

Este desbalance significativo, donde los datos de ataque predominan notablemente sobre los datos normales, presenta varios desafíos para el entrenamiento de modelos de inteligencia artificial.

Como ya se ha comentado previamente en el análisis del desequilibrio por categorías, un modelo entrenado en un dataset tan desequilibrado puede desarrollar un sesgo por lo que podría resultar en una alta tasa de falsos positivos, donde el modelo clasifica incorrectamente datos. Además de que al tener una escasez de datos normales limita la capacidad del modelo para aprender y reconocer patrones de tráfico benigno.

Este análisis realizado subraya la importancia de abordar el desequilibrio existente entre las clases para mejorar el entrenamiento y el rendimiento de los modelos.

Para obtener un conocimiento exhaustivo de las diferentes distribuciones de datos contenidas en los datasets, se ha analizado la distribución de los protocolos utilizados en las conexiones (Figura 47) y de los servicios que emplean dichas conexiones.

De manera general, se observa que la gran mayoría de las conexiones se realizan mediante el protocolo TCP, una pequeña proporción utiliza UDP, y una ínfima parte emplea el protocolo ICMP.

Esto implica que, inevitablemente, el modelo será más capaz de predecir datos relacionados con el protocolo TCP, dado que la cantidad de datos para TCP es significativamente mayor en comparación con los otros protocolos. La distribución de protocolos incluye las siguientes cantidades de datos:

Protocolo	Registros
TCP	18.864.383
UDP	1.666.797
ICMP	15.310

Tabla 37: Detalle de los protocolos en los datasets. Fuente: Elaboración propia

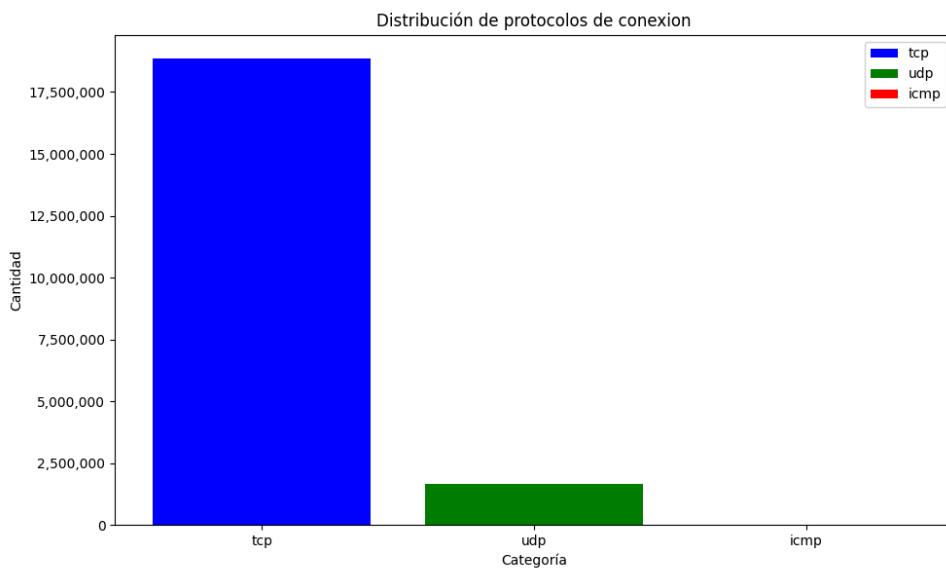


Figura 47: Distribución de los protocolos en los datasets. Fuente: Elaboración propia

A continuación, en la Tabla 38, se observa en detalle la distribución de los diferentes servicios de cada conexión en los datasets y vemos que la gran mayoría de las conexiones no se identifica el servicio al que pertenece. A pesar de ello se obtienen una cantidad detallada de conexión bajo el servicio HTTP, DNS y SSL.

Servicio	Registros
-	15.123.096
dns	1,565,378
ssl	379,987
http	3,440,723
dhcp	123
gssapi;smb;ntlm	170
gssapi;smb	24
ftp	15,038
dce_rpc	279
gssapi	383
radius	41
smb;ntlm;gssapi	7
gssapi;ntlm	5
dce_rpc;ntlm	15
smb	128
socks	1
gssapi;dce_rpc;smb	1
imap	73
krb_tcp	26
smtp;ssl	7,245
imap;ssl	6,031
smtp	470
ftp-data	84
irc	16
ssh	23
pop3	8
rfb	4
pop3;ssl	3
ssl;imap	144
ntlm;gssapi;smb	3
ssl;smtp	6,747
smb;gssapi;ntlm	153
gssapi;ntlm;smb	28
smb;dce_rpc;gssapi;ntlm	10
dce_rpc;gssapi;smb;ntlm	5
ntlm;dce_rpc	4
ntlm;smb;gssapi	5
smb;dce_rpc;gssapi	1
gssapi;ntlm;smb	7
<b>Total</b>	<b>20.546.490</b>

Tabla 38: Detalle de los servicios de las conexiones en los datasets. Fuente: Elaboración propia

Durante el proceso de análisis de los datos que contienen los datasets para su procesamiento, se ha tratado de buscar valores atípicos para validar el comportamiento que tendrían estos valores en los diferentes modelos. Estos valores atípicos se muestran en la Figura 48.

Los valores atípicos pueden distorsionar las medidas estadísticas como la media, desviación estándar, y otros estadísticos descriptivos, proporcionando una visión incorrecta de los datos. Además que pueden afectar negativamente el rendimiento de los modelos de aprendizaje automático, especialmente aquellos sensibles a los valores extremos, como las redes neuronales.

Para la detección de anomalías, es fundamental identificar valores atípicos, ya que pueden ser indicativos de eventos importantes como fraudes, errores de entrada de datos o comportamientos inusuales en el sistema. En este contexto, se ha optado por validar los valores de las columnas “src\_bytes” y “dst\_bytes”, que representan el tamaño de datos enviados por el origen y el destino, respectivamente, así como los valores de duración.

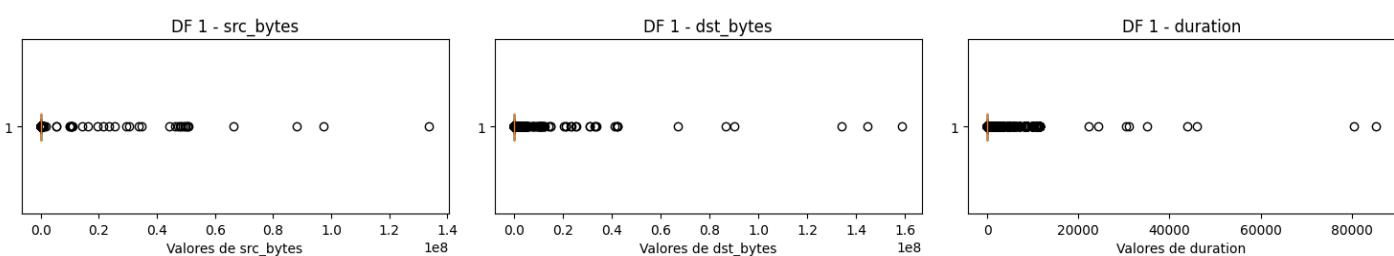
Los campos “src\_bytes” y “dst\_bytes” son esenciales para analizar el volumen de datos transferidos, lo cual puede indicar comportamientos anómalos cuando los valores son excesivamente altos o bajos.

La duración de una conexión es un indicador clave. Conexiones extremadamente cortas o largas pueden ser consideradas outliers y potencialmente fraudulentas.

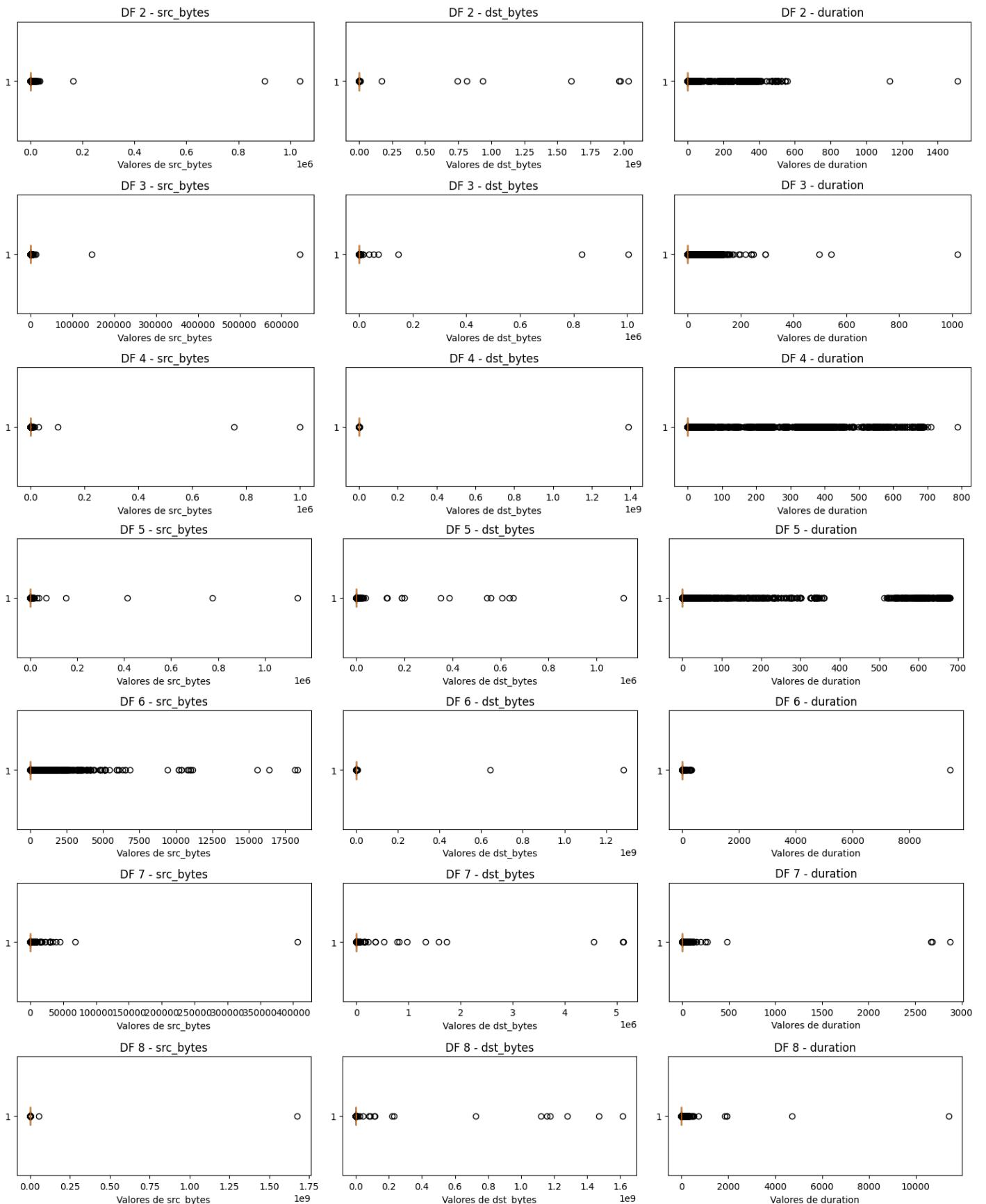
Valores extremos en los tamaños de datos o en la duración de las conexiones pueden señalar actividades sospechosas, como intentos de intrusión o ataques de denegación de servicio.

Al observar los campos “src\_bytes”, “dst\_bytes” y “duración”, se puede ver que la cantidad de outliers es considerable. Este ruido no representativo del comportamiento normal de las conexiones nos indica que estos outliers son cruciales para identificar posibles casos de ataque. La alta incidencia de outliers en estos campos no solo resalta su importancia, sino que también confirma que son indicadores valiosos en la identificación de comportamientos fraudulentos y anómalos en el sistema.

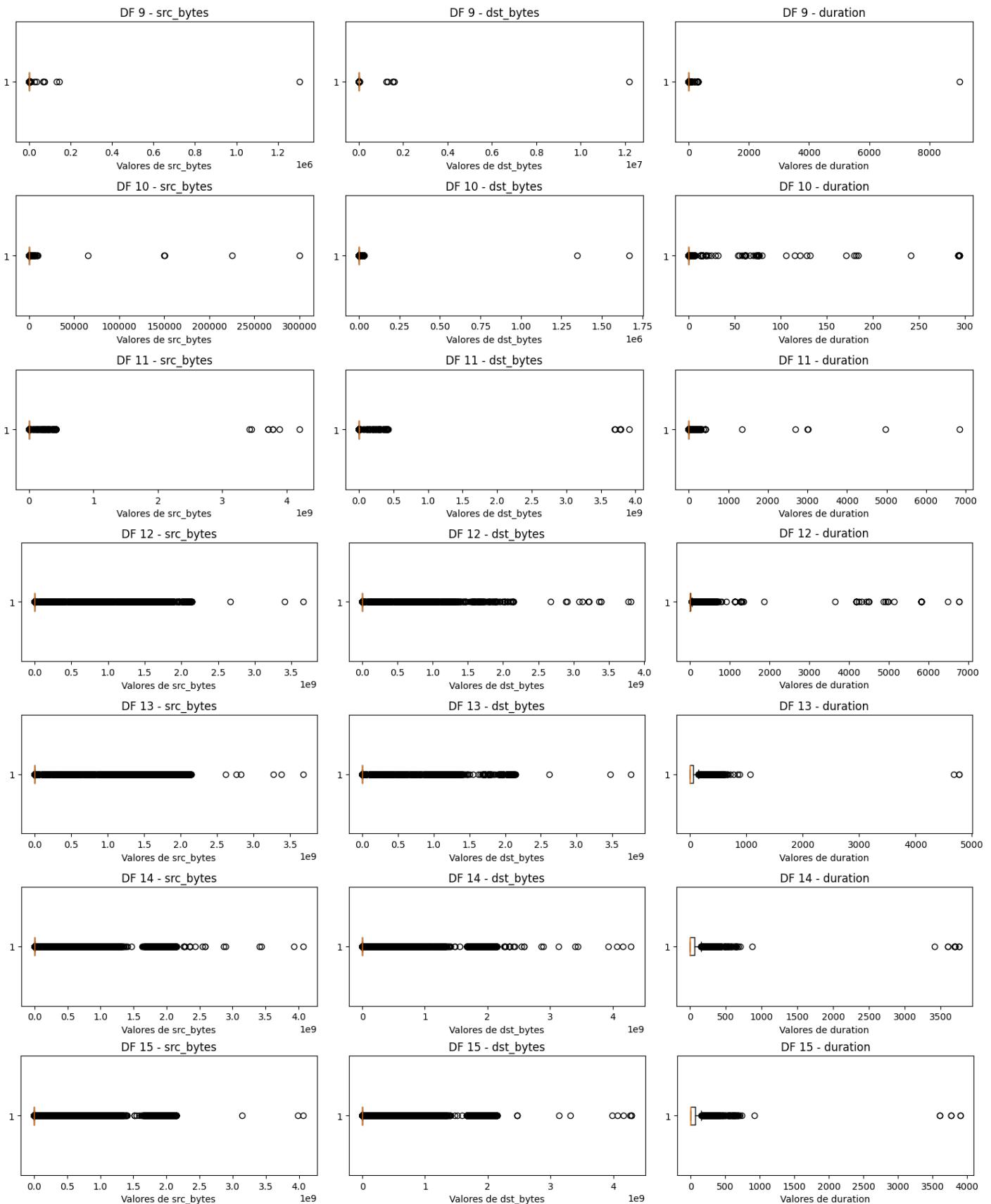
La presencia de numerosos outliers en estos campos confirma la relación entre estos valores extremos y la cantidad de datos maliciosos en los datasets, en comparación con el tráfico normal. Como se explicó anteriormente, los datasets contienen grandes cantidades de datos de ataques y muy pocos de conexiones benignas. Por lo tanto, los outliers reflejan adecuadamente la importancia de estos campos en la detección de posibles ataques.



Sistema de Detección de Intrusos en Ciberseguridad con Inteligencia Artificial  
 Francisco Jose Martin Aguilar



Sistema de Detección de Intrusos en Ciberseguridad con Inteligencia Artificial  
 Francisco Jose Martin Aguilar



Sistema de Detección de Intrusos en Ciberseguridad con Inteligencia Artificial  
 Francisco Jose Martin Aguilar



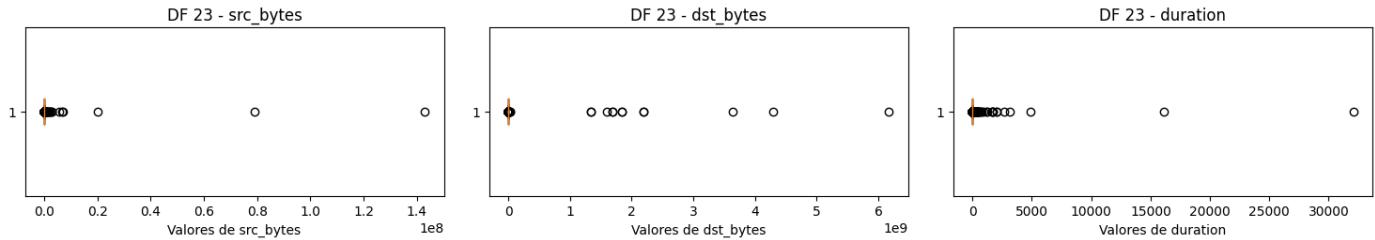


Figura 48: Valores atípicos en src\_byets, dst\_bytes y duration. Fuente: Elaboración propia

Después de verificar los datos y analizar los outliers, se ha realizado un tratamiento temporal para obtener una visión general de cómo se están recogiendo los datos a lo largo del tiempo. Para analizar esta recolección, se ha examinado la cantidad de ataques presentes en todos los datasets y la duración de las conexiones a lo largo del tiempo.

Este enfoque para entender cómo los ataques han sido recolectados a lo largo del tiempo es muy útil para conocer la procedencia de los datos y evaluar su robustez.

En la figura 49, se puede observar cómo los datos han sido recogidos de manera peculiar a lo largo del tiempo. Al inicio de la recolección, todos los datos corresponden a conexiones normales. A partir de ese momento, los datos recolectados en tiempo real consisten mayoritariamente en ataques, salvo algunos datos interpolados donde las conexiones son benignas. Sin embargo, la gran mayoría de los datos recolectados a partir de un punto cercano al 1.5555 son ataques.

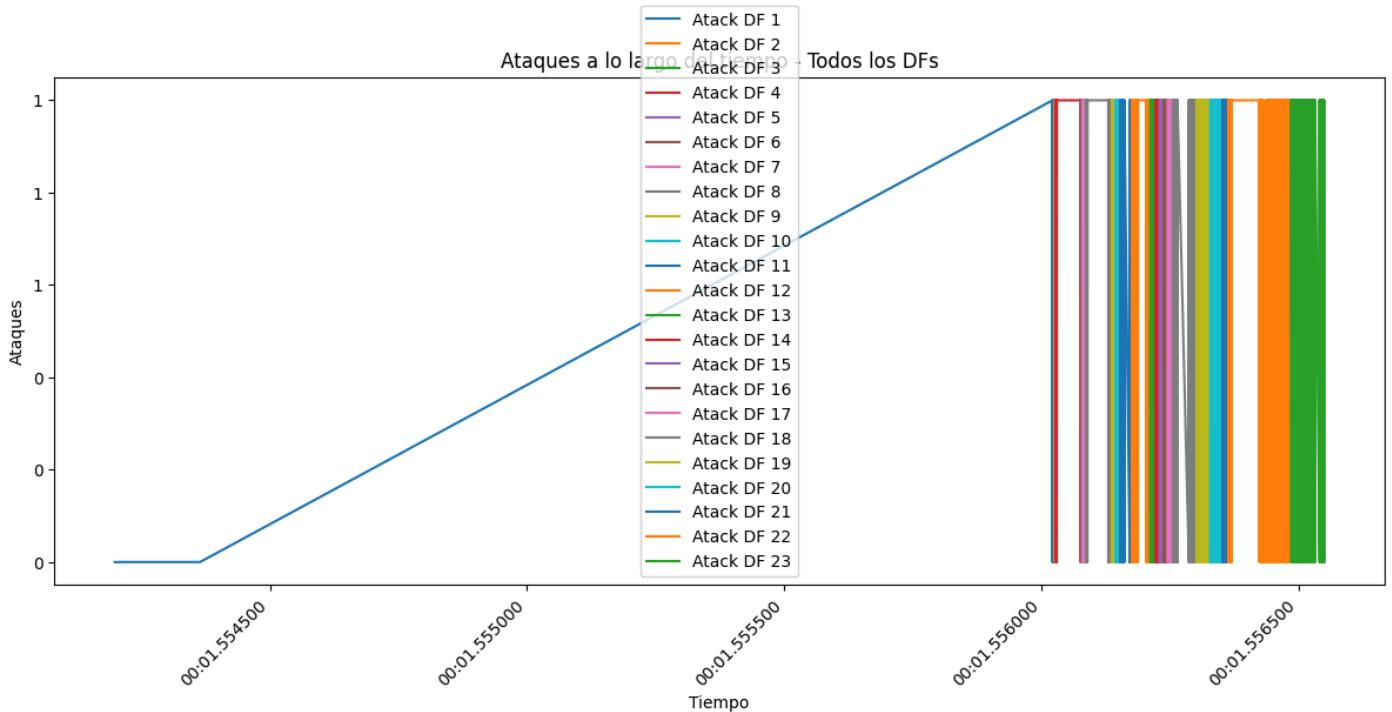


Figura 49: Gráfica de categoría label en tiempo de los datasets. Fuente: Elaboración propia

Además, en la Figura 50 se puede observar cómo varía el campo "duration" de los datos a lo largo del tiempo. Esto es importante junto con la observación de ataques a lo largo del tiempo, ya que puede revelar si existe una relación entre los ataques y la duración de la conexión.

Como se puede ver, la duración de la conexión y el campo "label" tienen algún tipo de relación, puesto que los primeros datos se consideran benignos y, por ello, tienen una duración de conexión apreciablemente mayor que los demás datos.

Observando los períodos en los que aparecen ataques, es decir, a partir del punto 1.5555, se nota una disminución pronunciada en la duración de la conexión. Esto sugiere que los ataques intentan pasar desapercibidos utilizando tiempos de conexión inferiores a los normales. A pesar de ello, también aparecen datos de ataques con duraciones de conexión similares a las de los datos benignos.

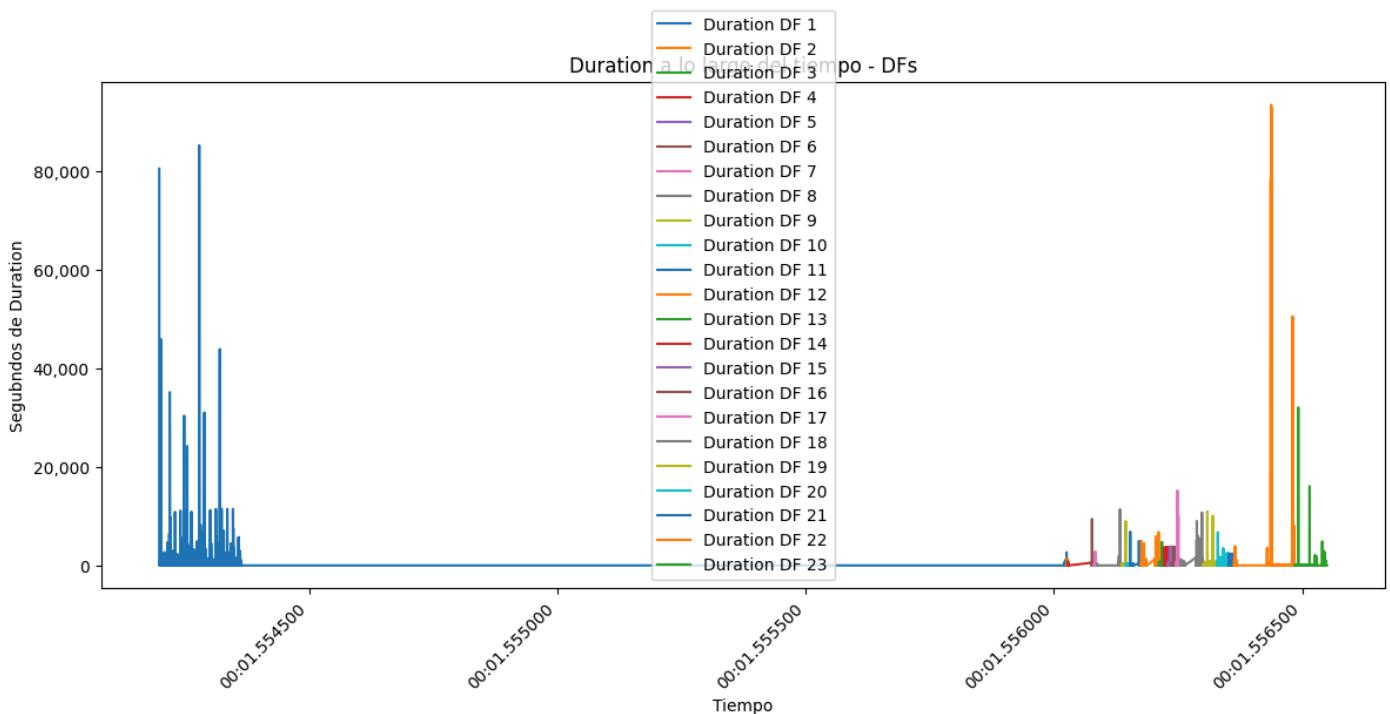


Figura 50: Gráfica de duración en tiempo de los dataset. Fuente: Elaboración propia

Para la sustitución de los datos categóricos por datos numéricos en los datasets, se ha utilizado Label Encoder. Esta elección se debe a que, en términos de rendimiento y eficacia, Label Encoder es más manejable con los datos que tenemos en comparación con One Hot Encoder. Dado que tenemos una gran cantidad de datos y cada campo contiene muchas posibles opciones, el uso de One Hot Encoder sería muy ineficiente, ya que generaría una gran cantidad de datos adicionales de forma masiva, aumentando significativamente el tamaño del dataset. Computacionalmente, utilizar One Hot Encoder no sería práctico en este caso.

Sin embargo, desde el punto de vista de la robustez, One Hot Encoder sería la mejor opción en la mayoría de los casos, incluyendo este proyecto, ya que evita problemas como la atribución de relaciones ordinales implícitas a categorías no ordenadas.

Para la transformación de los datos en este proyecto, no se ha utilizado la metodología de normalización debido a que los resultados de aplicarla son perjudiciales. Es crucial mantener la estructura original de los datos tanto como sea posible para evitar falsos positivos o negativos en el modelo. La pérdida de la distribución original podría confundir al modelo, especialmente dado que los datos contienen numerosos valores atípicos, como hemos observado anteriormente. La presencia de estos valores atípicos tendría un impacto significativo en la normalización del dataset, potencialmente introduciendo errores en los valores normalizados.

Del mismo modo, no se consideró útil aplicar el escalado de los datos, ya que el objetivo es preservar la distribución original de los mismos.

Como se puede observar en las Figuras 51 y 52, después de normalizar los datos, se pierde una cantidad significativa de información. Esto se debe a que prácticamente todos los campos que contienen valores atípicos son afectados, perdiendo su distribución original y resultando en valores entre 0 y 1.

src_ip	src_port	dst_ip	dst_port
192.168.1.31	50948	192.168.1.186	1067
192.168.1.32	54991	192.168.1.250	9

Figura 51: Ejemplo de datos del dataset 1 sin normalizar. Fuente: Elaboración propia

src_ip	src_port	dst_ip	dst_port
0.522	0.807	0.443	1.632e-02
0.565	0.871	0.489	1.376e-04

Figura 52: Ejemplo de datos del dataset 1 normalizado. Fuente: Elaboración propia

Como se puede observar en la Figura 51, en los valores del campo "src\_port" que representa el puerto de origen de la conexión, la diferencia entre ambos puertos en las filas es de 4043. En contraste, en la Figura 52, esta diferencia se disminuye considerablemente a 74. Estas variaciones pueden afectar la capacidad predictiva del modelo, potencialmente llevándolo a sobreajustarse o subajustarse.

Por esta razón, y con el objetivo de mantener la máxima precisión posible, aunque típicamente se recomienda la normalización o el escalado para mejorar el rendimiento del entrenamiento del modelo, se ha decidido no aplicarlo. Esta decisión busca preservar la robustez y la eficacia de los modelos ante la influencia potencialmente distorsionadora de los valores.

Finalmente, para permitir que el modelo realice predicciones basadas únicamente en los datos de entrada sin depender del momento exacto de su generación, hemos decidido eliminar la columna de timestamp. La inclusión de un timestamp en un dataset temporal podría llevar al modelo a generalizar y utilizar el tiempo como un factor predictivo. Esto podría resultar en predicciones incorrectas cuando el modelo se aplique a datos con diferentes marcas de tiempo.

Tras analizar las observaciones previas sobre el preprocesamiento de los datos, se ha concluido en la generación de dos tipos de conjuntos de datos. Unos conjuntos de datos están balanceados para predecir de manera binaria el campo "label" [43], que indica si se trata de un ataque o no. Otros conjuntos están diseñados para predecir de manera multiclasificarse el campo "type", que especifica el tipo de ataque o indica si es normal (benigno).

### 10.1.1.1 DATASETS POR LABEL

Para los datasets destinados a predecir si se trata de un ataque o no, de forma binaria, se han creado cuatro conjuntos de datos distintos. Dos de ellos están equilibrados, lo que significa que contienen la misma cantidad de conexiones benignas y malignas, mientras que los otros dos conservan la distribución original de los datos.

Estos conjuntos de datos proporcionarán información valiosa sobre cómo los modelos de inteligencia artificial aprenden, permitiéndonos observar si equilibrar un conjunto de datos es beneficioso o perjudicial.

Cada subtipo de dataset, tanto equilibrado como con distribución original, consta de dos conjuntos: uno con 1.377.434 registros y otro con 100.000 registros. La generación de ambos tamaños de datasets permite analizar el rendimiento de los modelos frente a diferentes volúmenes de datos, ayudándonos a determinar si una gran cantidad de datos es necesaria para un correcto aprendizaje del modelo.

Para equilibrar los datos de ataques y conexiones normales en el dataset más grande, seleccionamos la cantidad máxima de la clase normal, que es 688.717 registros. Dado que existen 9 clases de ataques, el cálculo es el siguiente:

$$688.717 / 9 \approx 76.524$$

Para cada tipo de ataque, seleccionaremos 76.524 registros. En los casos donde los ataques contienen menos registros que los requeridos, seleccionaremos todos los registros disponibles y compensaremos, los registros faltantes, con los de otros ataques con más datos. La distribución final del dataset más grande equilibrado, para predecir si es un ataque o no, se muestra en la Tabla 39.

Categoría	Registros
Datos normales	688.717
Scanning	93.638
Dos	93.637
Injection	93.637
DDOS	93.637
Password	93.637
XSS	93.637
Backdoor	93.637
Ransomware	32.214
Mitm	1.043
<b>Total</b>	<b>1.377.434</b>

Tabla 39: Detalle del dataset equilibrado 1,3M para label. Fuente: Elaboración propia

Para equilibrar los datos de ataques y conexiones normales en el dataset más pequeño, aplicamos la misma fórmula utilizada para el dataset más grande, ajustada a un tamaño menor. Utilizando un 50% de ataques y un 50% de normales, obtenemos:

$$50.000 / 9 \approx 5.555$$

Para cada tipo de ataque, seleccionamos 5.555 registros. Igual que en el caso anterior, en los casos donde los ataques contienen menos registros que los requeridos, seleccionaremos todos los registros disponibles y compensaremos, los registros faltantes, con los de otros ataques con más datos. La distribución final del dataset más pequeño equilibrado, para predecir si es un ataque o no, se muestra en la Tabla 40.

Categoría	Registros
Datos normales	50.000
Scanning	6.120
Dos	6.120
Injection	6.120
DDOS	6.120
Password	6.120
XSS	6.119
Backdoor	6.119
ransomware	6.119
Mitm	1.043
<b>Total</b>	<b>100.000</b>

Tabla 40: Detalle del dataset equilibrado 100K para label. Fuente: Elaboración propia

Para generar los datasets no equilibrado utilizaré la distribución original de los datos. En el caso del dataset mayor tendrá una cantidad de 1.377.434 registros y para el dataset de menor tamaño con 100.000 registros. La distribución del dataset mayor quedaría reflejada en la tabla 41 y la distribución del dataset menor en la tabla 42.

Categoría	Registros
Datos normales	46.172
Scanning	478.676
Dos	121.739
Injection	30.346
DDOS	407.796
Password	115.212
XSS	141.383
Backdoor	33.881
ransomware	2.159
Mitm	70
<b>Total</b>	<b>1.377.434</b>

Tabla 41: Detalle del dataset no equilibrado 1,3M para label / type. Fuente: Elaboración propia

Categoría	Registros
Datos normales	3.353
Scanning	34.752
Dos	8.838
Injection	2.203
DDOS	29.605
Password	8.364
XSS	10.264
Backdoor	2.459
Ransomware	157
Mitm	5
<b>Total</b>	<b>100.000</b>

Tabla 42: Detalle del dataset no equilibrado 100K para label / type. Fuente: Elaboración propia

Con estas distribuciones, se han generado los datasets equilibrados y no equilibrados. Los datasets equilibrados contienen un 50% de ataques y un 50% de conexiones benignas, disponibles en tamaños mayores y menores. Por otro lado, los datasets no equilibrados se crearon utilizando la distribución original de los datos, también en tamaños mayores y menores.

### 10.1.1.2 DATASETS POR TIPO

Para los datasets destinados a predecir el tipo de ataque o si es categorizado como "normal", se han creado dos conjuntos de datos distintos. Ambos conjuntos están equilibrados en relación a los diferentes valores del campo "type", lo que significa que contienen la misma cantidad de conexiones de cada tipo.

El primer dataset equilibrado consta de 1.377.434 registros, mientras que el segundo contiene 100.000 registros. Como se ha comentado anteriormente, la generación de ambos tamaños de datasets nos permite analizar el rendimiento de los modelos ante diferentes volúmenes de datos.

Como en todos los conjuntos de datos solo hay una cantidad de 1.043 para el tipo "mitm", se ha decidido hacer un dataset donde los datos no sean tan precisos en cuanto a la distribución, porque si no serían datasets muy pequeñas. Para esto, la distribución que se ha llevado a cabo para el dataset equilibrado por tipo mayor se puede observar en la tabla 43 y para el dataset equilibrado por tipo de tamaño menor se describe en la tabla 44.

Categoría	Registros
Datos normales	168.023
Scanning	168.022
Dos	168.022
Injection	168.022
DDOS	168.022
Password	168.022
XSS	168.022
Backdoor	168.022
Ransomware	32.214
Mitm	1.043
<b>Total</b>	<b>1.377.434</b>

Tabla 43: Detalle del dataset equilibrado 1,3M para type. Fuente: Elaboración propia

Categoría	Registros
Datos normales	10.996
Scanning	10.996
Dos	10.995
Injection	10.995
DDOS	10.995
Password	10.995
XSS	10.995
Backdoor	10.995
Ransomware	10.995
Mitm	1.043
<b>Total</b>	<b>100.000</b>

Tabla 44: Detalle del dataset equilibrado 100K para type. Fuente: Elaboración propia

En este caso, no se han creado conjuntos de datos no equilibrados adicionales, ya que el dataset necesario fue previamente creado en la creación del dataset no equilibrado para label. Dado que la distribución de los datos es consistente en ambos conjuntos, utilizaremos el mismo para intentar predecir el tipo de ataque, manteniendo la distribución no equilibrada de los datos originales.

### 10.1.2 ANÁLISIS DE LOS DATASETS PREPROCESADOS

Una vez creados los seis datasets anteriores, pasamos a aplicar un encoder a los mismos dando como resultado la transformación de todos los valores categóricos en numéricos. En la Figura 53 se puede observar un ejemplo del estado del dataset equilibrado para label una vez aplicado LabelEncoder.

src_ip	src_port	dst_ip	dst_port	proto	service	duration	src_bytes	dst_bytes	conn_state
12	50948	58	1067	1	0	0.000e+00	0	0	6
13	54991	64	9	1	0	9.616e-01	0	0	1
11	42908	70	1641	1	0	0.000e+00	0	0	6
11	42909	70	2004	1	0	0.000e+00	0	0	6
13	54770	71	16080	1	0	0.000e+00	0	0	6
...	...	...	...	...	...	...	...	...	...
12	60816	13	443	1	0	2.319e+01	32	31	10
12	56692	32	443	1	0	2.361e+01	32	31	10
12	41054	33	443	1	0	2.287e+01	32	31	10
12	49718	9	443	1	0	4.980e-04	0	31	10
12	46512	77	443	1	0	3.786e+01	399	399	10

Figura 53: Dataset equilibrado 100K para label codificado. Fuente: Elaboración propia

Como podemos observar, los valores de "src\_ip" y "dst\_ip", que en el dataset original eran de tipo "object" y categóricos, ahora se han convertido en valores numéricos. Este cambio permite que el modelo de inteligencia artificial pueda entrenar de manera más eficiente y precisa.

Posteriormente a la codificación, buscamos relaciones entre las características utilizando una matriz de correlación. Esta matriz nos informa sobre la relación que pueden tener los datos entre sí. Para nuestro proyecto, nos interesa enfocar este análisis en la relación que pueden tener las categorías "label" y "type" con respecto al resto de las características. De esta manera, podemos identificar o encontrar una relación entre "label" o "type" y las demás características, proporcionando valiosa información para el modelo.

Para analizar los resultados de las matrices de correlación de los datasets, hemos tomado una única matriz de correlación como ejemplo. Dado que contamos con seis datasets, cada uno con diferentes tamaños y distribuciones, cada uno muestra relaciones distintas. En la figura 54 se presenta una de estas matrices de correlación para ilustrar el procedimiento y entender mejor las relaciones entre las variables.

En esta matriz de correlación, que corresponde al dataset equilibrado por "label" de menor tamaño, se puede observar que para la clasificación por "label", los campos "src\_ip", "conn\_state", "dst\_port" y "proto" son las variables que muestran una relación con el campo "label".

En contraste, al observar la correlación del campo "type" con los demás, no se ha encontrado ninguna relación significativa. Esto indica que el dataset no está diseñado para identificar relaciones con el campo "type", sino más bien para relacionarse con el campo "label", que es la finalidad de este dataset ya que la distribución de este dataset es de 50% de datos benignos y 50% de datos de ataques.

Sistema de Detección de Intrusiones en Ciberseguridad con Inteligencia Artificial  
Francisco Jose Martin Aguilar

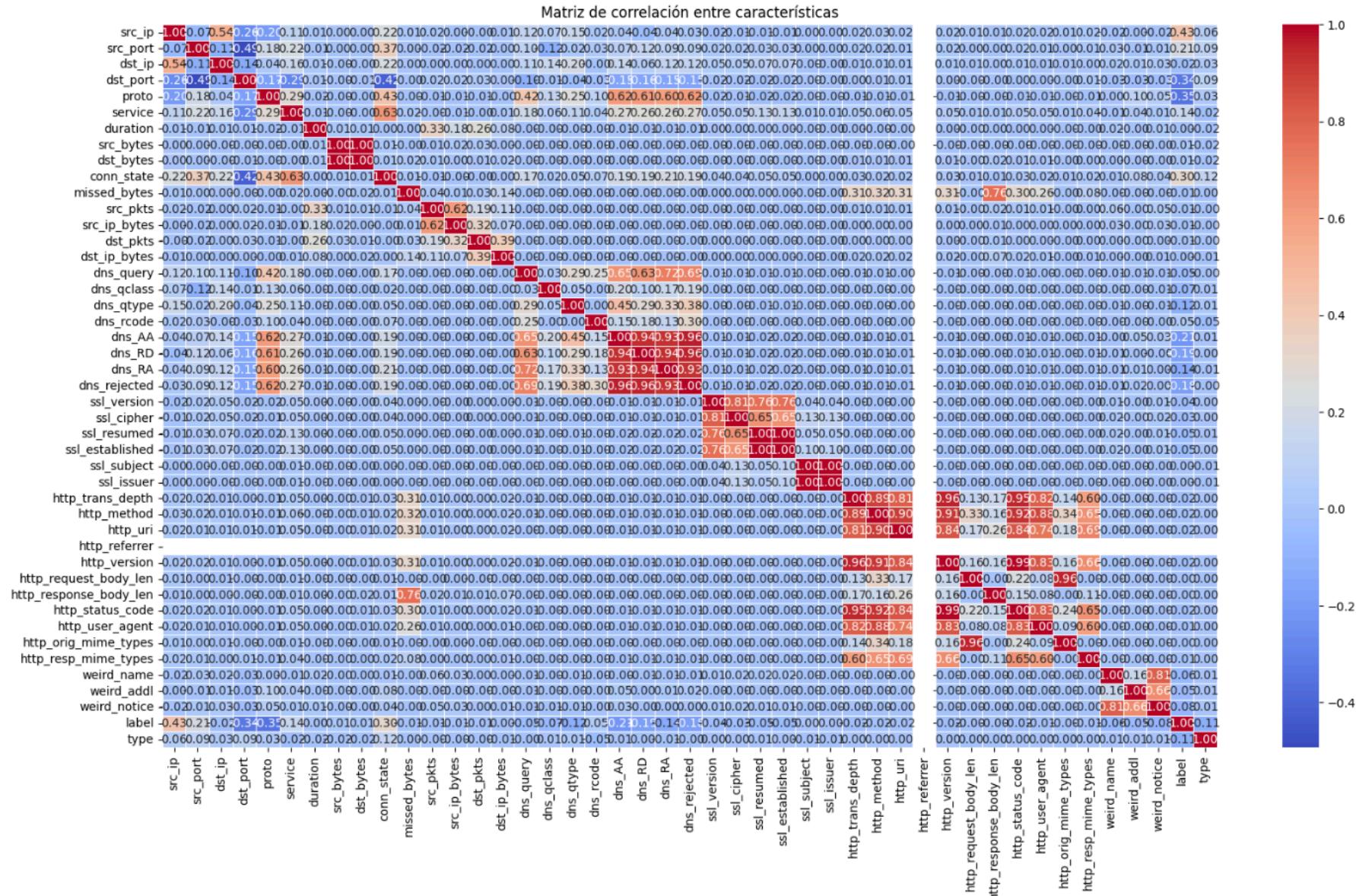


Figura 54: Matriz de correlación del dataset equilibrado 100K para label codificado. Fuente: E. propia

## 10.2 EJECUCIÓN DE MODELOS

Una vez que tenemos los datasets codificados, debemos ejecutar los modelos de inteligencia artificial sobre ellos. Previamente, hemos comprobado el correcto funcionamiento de la validación cruzada probando el modelo Naive Bayes con y sin validación cruzada.

En la Figura 55, se pueden observar los resultados de aplicar el algoritmo de Naive Bayes al dataset equilibrado de 100.000 registros, prediciendo por etiqueta con validación cruzada de 5 folds. En la Figura 56, se muestran los resultados sin usar la validación cruzada. Como se puede ver, es más efectivo utilizar K-Fold, ya que obtenemos resultados más precisos sobre cómo el modelo aprende de los datos. Por esta razón, se utiliza K-Fold. A pesar de la mayor necesidad computacional y el tiempo de ejecución que implica usar K-Fold, este método nos asegura que todos los datos sean utilizados tanto para entrenamiento como para validación. Así, el modelo será entrenado y validado con la totalidad del dataset, garantizando una evaluación más completa y robusta. Además, la métrica de informe de clasificación nos proporciona información de la cantidad de datos evaluada . Como podemos ver en los valores de “support”, con KFold se evalúa todos los datos.

```
Accuracy: 0.5582
F1 Score: 0.6878
Precision: 0.5318
Recall: 0.9734

Matriz de Confusión Promedio:
[[ 7151 42849]
 [ 1329 48671]]
Classification Report Promedio:
precision    recall    f1-score   support
0            0.85     0.14      0.24      50000
1            0.53     0.98      0.69      50000

accuracy                           0.56      100000
macro avg                            0.69      0.47      100000
weighted avg                          0.69      0.56      0.47      100000
```

Figura 55: Métricas de rendimiento usando K-Fold. Fuente: Elaboración propia

```
Accuracy: 0.5549
F1 Score: 0.6843
Precision: 0.5284
Recall: 0.9707

Matriz de Confusión:
[[1451 8611]
 [ 291 9647]]
Classification Report:
precision    recall    f1-score   support
0            0.83     0.14      0.25      10062
1            0.53     0.97      0.68      9938

accuracy                           0.55      20000
macro avg                            0.68      0.47      20000
weighted avg                          0.68      0.55      0.46      20000
```

Figura 56: Métricas de rendimiento sin usar K-Fold. Fuente: Elaboración propia

### 10.2.1 ALGORITMOS BINARIOS

Para la ejecución de algoritmos destinados a predecir de manera binaria si una etiqueta corresponde a un ataque o no, se ha empleado los siguientes modelos:

- Naive Bayes (clasificación binaria)
- Decision Tree
- Random Forest
- XGBoost
- LightGBM
- CatBoost
- Redes Neuronales (ANN)

Cada uno de los modelos ha sido entrenado utilizando varios conjuntos de datos creados previamente para predecir etiquetas de ataques o normales. Esto incluye conjuntos de datos equilibrados y desequilibrados de diferentes tamaños. Esta variedad nos ha permitido analizar cómo el tamaño y la distribución de los datos afectan el rendimiento de los diferentes modelos.

Al seleccionar conjuntos de datos con una proporción equilibrada del 50% de ataques y 50% de benignos, nuestro objetivo fue asegurar que los modelos estuvieran entrenados de manera equitativa para predecir correctamente ambos tipos de instancias. Además, para cada modelo se proporcionaron conjuntos de datos de mayor y menor tamaño para evaluar cuánto influyen la cantidad de datos en su rendimiento y capacidad de entrenamiento.

Además, se han evaluado las siguientes métricas de rendimiento:

- Matriz de confusión
- Accuracy
- Precision
- Recall
- F1-Score
- Classification Report
- Gráfica de la curva ROC
- Gráfica de la curva PR

Los resultados obtenidos mediante estas métricas de rendimiento se han analizado específicamente para cada caso, considerando tanto el conjunto de datos como el modelo utilizado.

### 10.2.2 ALGORITMOS MULTICLASIFICACIÓN

Para los algoritmos de clasificación multiclas, hemos utilizado las mismas métricas de rendimiento y modelos de inteligencia artificial que las usadas en el entrenamiento de algoritmos de clasificación binaria. Sin embargo, para transformar el código de predicciones binarias a predicciones multiclas, es necesario ajustar cómo se manejan las etiquetas y las probabilidades de predicción. Para ello hemos seleccionado que la variable a predecir sea el campo de "type". Aunque el modelo Naive Bayes que utilizamos (GaussianNB) soporta naturalmente la clasificación multiclas, necesitamos modificar el cálculo de métricas y la generación de curvas ROC y PR para cada clase, y luego promediarlas para obtener un resultado general.

Es importante destacar que al usar K-Fold Cross-Validation, tanto las métricas como las curvas ROC se calculan sumando y promediando los resultados de cada fold. Esto proporciona una evaluación más robusta del modelo en múltiples conjuntos de datos de entrenamiento y validación.

En el caso específico de CatBoost, cuando se enfrentan a conjuntos de datos con distribución desigual y gran tamaño, puede resultar en la generación de un gran número de variables. Esto ha llevado a tiempos de ejecución que superan las 5 horas en entornos como Google Colab, que tiene una restricción de tiempo máxima de ejecución. Cuando se supera este límite, la ejecución se interrumpe y se produce el error mencionado.

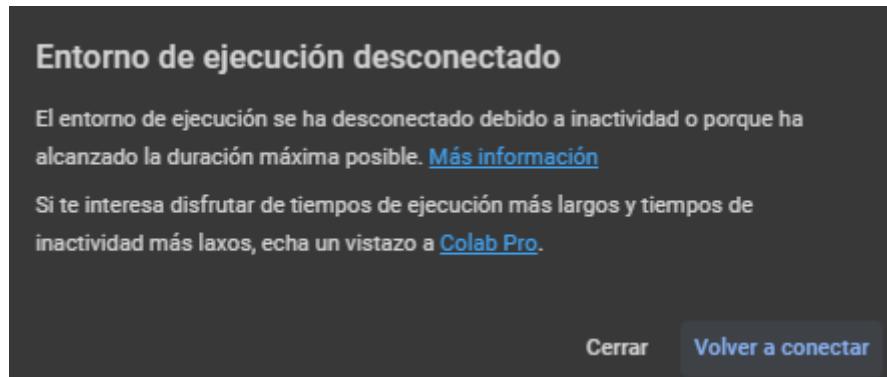


Figura 57: Error de ejecución en entorno Google Colab. Fuente: Elaboración propia

Este error nos indica que hemos superado la duración máxima posible por lo que el dataset de gran tamaño no equilibrado para el algoritmo CatBoost no se ha podido realizar.

## 10.3 RESULTADOS

A continuación, se presentan los resultados obtenidos tras la aplicación de diversos modelos de inteligencia artificial y el análisis de las métricas de rendimiento correspondientes, evaluados en diferentes configuraciones de datos y algoritmos.

### 10.3.1 ANÁLISIS DE RESULTADOS

En este apartado, se analizan en detalle los resultados obtenidos de los modelos, así como las métricas de rendimiento utilizadas, como la precisión, recall, F1-score, matriz de confusión, entre otras. Este análisis abarca tanto los modelos de clasificación binaria como los de clasificación multiclas.

En la Tabla 45 se muestran las métricas de rendimiento (precisión, recall, F1-score, accuracy y tiempo de ejecución) de los modelos de clasificación binaria, evaluados en datasets equilibrados de gran y pequeño tamaño. Los resultados revelan que modelos como Random Forest, XGBoost, LightGBM y CatBoost alcanzan un rendimiento cercano al perfecto (1.0) en todas las métricas, tanto en datasets pequeños como grandes. Por ejemplo, Random Forest y XGBoost logran una precisión, recall, F1-score y accuracy de 1.0 en ambos tamaños de dataset, con tiempos de ejecución de aproximadamente 18 minutos para Random Forest en el dataset grande, y 4 minutos para XGBoost en el mismo dataset.

Estos modelos son conocidos por su capacidad para capturar relaciones complejas en los datos y ajustarse bien a los patrones subyacentes. Decision Tree, Random Forest, XGBoost, LightGBM y CatBoost utilizan estrategias de ensemble learning o boosting que les permiten mejorar la precisión y la generalización del modelo.

En contraste, Naive Bayes muestra un rendimiento más bajo / moderado con precisión y recall variados ya que obtiene un F1-score de 0.7273 y 0.5227 para los conjuntos de datos grandes y pequeños respectivamente. Sin embargo, se destaca por sus tiempos de ejecución más cortos, con aproximadamente 50 segundos para el dataset grande y 5 segundos para el pequeño. Naive Bayes asume independencia condicional entre las características, lo que puede ser una limitación en conjuntos de datos donde las características están correlacionadas. Esto explica la variabilidad en sus métricas de rendimiento, especialmente en recall y precisión. Sin embargo, es rápido y eficiente en términos de tiempo de ejecución.

Clasificación Binaria						
Datasets Equilibrados						
Codificación por Etiqueta						
Dataset	Modelo	Accuracy	F1-Score	Recall	Precision	Tiempo
Pequeño	Naive Bayes	0.5582	0.6878	0.9734	0.5318	0:00:05.3784
Pequeño	Decision Tree	0.9997	0.9997	0.9997	0.9998	0:00:05.3865
Pequeño	R. Forest	0.9999	0.9999	0.9999	0.9999	0:00:56.5806
Pequeño	XGBoost	1.0	1.0	1.0	1.0	0:00:18.7286
Pequeño	LightGBM	1.0	1.0	0.9999	1.0	0:00:13.9119
Pequeño	CatBoost	0.9999	0.9999	0.9999	0.9999	0:04:21.0638
Pequeño	R. N. ANN	0.6110	0.5227	0.2863	0.9503	0:07:46.5251
Grande	Naive Bayes	0.6299	0.7273	0.9870	0.5758	0:00:49.4530
Grande	Decision Tree	1.0	1.0	1.0	1.0	0:01:12.1572
Grande	R. Forest	1.0	1.0	1.0	1.0	0:18:29.3455
Grande	XGBoost	1.0	1.0	1.0	1.0	0:04:10.5756
Grande	LightGBM	1.0	1.0	1.0	1.0	0:03:01.0909
Grande	CatBoost	1.0	1.0	1.0	1.0	0:22:55.4585
Grande	R. N. ANN	0.9377	0.9377	0.9458	0.9312	0:37:00.6049

Tabla 45: Métricas de rendimiento para label con datos equilibrados. Fuente: Elaboración propia

En la Tabla 46 se presentan las métricas de rendimiento (precisión, recall, F1-score, accuracy y tiempo de ejecución) de modelos de clasificación binaria evaluados en datasets no equilibrados, considerando tanto datasets pequeños como grandes. Los resultados destacan diferencias significativas entre los modelos evaluados.

Al igual que en las ejecuciones de modelos usando dataset equilibrado, los modelos como Decision Tree, Random Forest, XGBoost, LightGBM y CatBoost muestran un rendimiento casi perfecto en todos los indicadores de rendimiento para ambos tamaños de dataset no equilibrados. Por ejemplo, todos estos modelos logran una precisión, recall, F1-score y accuracy de 1.0, indicando una capacidad excepcional para clasificar correctamente tanto las instancias positivas como negativas del dataset. Además, los tiempos de ejecución de estos modelos, aunque varían, permanecen razonables incluso en grandes volúmenes de datos, con tiempos que van desde unos pocos segundos hasta aproximadamente 20 minutos para CatBoost en el dataset grande.

Naive Bayes, por otro lado, muestra un rendimiento más bajo en comparación con los modelos mencionados anteriormente. Mientras que es notable por su tiempo de ejecución rápido (alrededor de 50 segundos para el dataset grande y 2 segundos para el pequeño), sus métricas de rendimiento como precisión, recall y F1-score varían significativamente. Por ejemplo, alcanza un F1-score de 0.9507 para el dataset grande y 0.0094 para el pequeño.

Clasificación Binaria						
Datasets No Equilibrados						
Codificación por Etiqueta						
Dataset	Modelo	Accuracy	F1-Score	Recall	Precision	Tiempo
Pequeño	Naive Bayes	0.0381	0.0094	0.0048	0.9941	0:00:01.7811
Pequeño	Decision Tree	1.0	1.0	1.0	1.0	0:00:05.2926
Pequeño	R. Forest	1.0	1.0	1.0	1.0	0:00:42.9518
Pequeño	XGBoost	1.0	1.0	1.0	1.0	0:00:37.2016
Pequeño	LightGBM	1.0	1.0	1.0	1.0	0:00:20.6465
Pequeño	CatBoost	0.9999	1.0	1.0	0.9999	0:03:44.7519
Pequeño	R. N. ANN	0.9801	0.9769	0.9955	0.9843	0:05:26.0711
Grande	Naive Bayes	0.5384	0.5843	0.5405	0.9753	0:00:21.8018
Grande	Decision Tree	1.0	1.0	1.0	1.0	0:01:09.4222
Grande	R. Forest	1.0	1.0	1.0	1.0	0:16:13.1143
Grande	XGBoost	1.0	1.0	1.0	1.0	0:01:35.4340
Grande	LightGBM	1.0	1.0	1.0	1.0	0:01:36.6861
Grande	CatBoost	1.0	1.0	1.0	1.0	0:20:18.4383
Grande	R. N. ANN	0.9668	0.9507	1.0	0.9668	0:49:52.8273

Tabla 46: Métricas de rendimiento para label con datos no equilibrados. Fuente: Elaboración propia

En la Tabla 47 se presentan las métricas de rendimiento (accuracy, F1-score, recall, precision y tiempo de ejecución) de modelos de clasificación múltiple evaluados en datasets equilibrados, tanto para datasets pequeños como grandes. Los resultados revelan diferencias significativas en el rendimiento de los modelos según el algoritmo utilizado y el tamaño del dataset.

Decision Tree y Random Forest muestran un rendimiento excelente en todos los indicadores de rendimiento para ambos tamaños de dataset. Estos modelos alcanzan un accuracy, F1-score, recall y precision cercanos a 1.0, lo que indica una capacidad casi perfecta para clasificar correctamente las múltiples clases del dataset. Además, los tiempos de ejecución son relativamente cortos, con menos de un minuto para el Decision Tree y aproximadamente 12 minutos para Random Forest en el dataset grande.

XGBoost y LightGBM también muestran un rendimiento muy alto en todas las métricas evaluadas. Aunque ligeramente inferiores a Decision Tree y Random Forest, estos modelos logran un accuracy y F1-score por encima del 0.99 en ambos tamaños de dataset, con tiempos de ejecución que varían entre aproximadamente 6 minutos para XGBoost y 17 minutos para LightGBM en el dataset grande.

Las Redes Neuronales Artificiales también muestran un buen rendimiento con un accuracy cercano a 0.98 y F1-score alrededor de 0.98 para ambos tamaños de dataset. Sin embargo, los tiempos de ejecución son significativamente mayores en comparación con los modelos basados en árboles y boosting, con aproximadamente 18 minutos para el dataset pequeño y más de 3 horas para el dataset grande.

Por otro lado, Naive Bayes muestra un rendimiento más bajo en comparación con los modelos mencionados anteriormente, especialmente en términos de F1-score y precision, con valores alrededor del 0.18 para ambos tamaños de dataset. Aunque ofrece tiempos de ejecución relativamente cortos (alrededor de 7 segundos para el dataset pequeño y 43 segundos para el dataset grande), su capacidad para manejar relaciones complejas entre características correlacionadas es limitada, lo que afecta su rendimiento en problemas de clasificación múltiple.

Una observación interesante es el rendimiento inferior de LightGBM en el dataset grande en comparación con otros modelos. Aunque todavía alcanza un accuracy y F1-score de aproximadamente 0.90, muestra una ligera disminución en la precisión y recall en comparación con los datasets pequeños y otros modelos evaluados. Esto sugiere que LightGBM puede ser sensible al tamaño del dataset y a la distribución de clases en problemas de clasificación múltiple.

En conclusión, modelos como Decision Tree, Random Forest, XGBoost y Redes Neuronales Artificiales destacan por su capacidad para manejar eficazmente datasets equilibrados de múltiples clases, ofreciendo un rendimiento robusto en términos de precisión y generalización. Estos modelos utilizan estrategias avanzadas como árboles de decisión, boosting y redes neuronales que les permiten capturar relaciones complejas en los datos y adaptarse bien a diferentes tamaños de dataset.

Por otro lado, Naive Bayes, aunque rápido en términos de tiempo de ejecución, muestra limitaciones en su capacidad para abordar problemas de clasificación múltiple con datos equilibrados, reflejado en su rendimiento más modesto en las métricas evaluadas.

Clasificación Múltiple						
Datasets Equilibrados						
Codificación por Etiqueta						
Dataset	Modelo	Accuracy	F1-Score	Recall	Precision	Tiempo
Pequeño	Naive Bayes	0.2602	0.1814	0.2602	0.4214	0:00:07.619245
Pequeño	Decision Tree	0.9950	0.9950	0.9949	0.9950	0:00:06.917932
Pequeño	R. Forest	0.9961	0.9961	0.9961	0.9961	0:01:22.230544
Pequeño	XGBoost	0.9964	0.9964	0.9964	0.9965	0:06:53.677666
Pequeño	LightGBM	0.9967	0.9967	0.9967	0.9967	0:01:23.159641
Pequeño	CatBoost	0.9960	0.9960	0.9960	0.9960	0:33:05.725057
Pequeño	R. N. ANN	0.9840	0.9840	0.9840	0.9844	0:18:10.860757
Grande	Naive Bayes	0.2227	0.1121	0.2227	0.3719	0:00:42.674935
Grande	Decision Tree	0.9942	0.9942	0.9942	0.9942	0:00:55.299085
Grande	R. Forest	0.9951	0.9951	0.9951	0.9951	0:11:52.933014
Grande	XGBoost	0.9941	0.9941	0.9941	0.9941	0:14:00.898903
Grande	LightGBM	0.9039	0.9022	0.9039	0.9124	0:17:45.014000
Grande	CatBoost	-	-	-	-	<b>5h+</b>
Grande	R. N. ANN	0.9762	0.9758	0.9762	0.9760	3:24:28.369172

Tabla 47: Métricas de rendimiento para type con datos equilibrados. Fuente: Elaboración propia

Igual que en las anteriores tablas, en la Tabla 48 se presentan las métricas de rendimiento (accuracy, F1-score, recall, precision y tiempo de ejecución) de modelos de clasificación múltiple evaluados en datasets no equilibrados, tanto para datasets pequeños como grandes. Estos resultados proporcionan una previsión sobre cómo los diferentes modelos manejan el desbalance de clases y la complejidad asociada.

Los algoritmos Decision Tree y Random Forest muestran un rendimiento excelente en todos los indicadores de rendimiento para ambos tamaños de dataset no equilibrados. Estos modelos alcanzan un accuracy, F1-score, recall y precision cercanos a 1.0, lo que indica una capacidad muy alta para clasificar correctamente múltiples clases desbalanceadas. Además, los tiempos de ejecución son relativamente cortos, con menos de 3 minutos para Decision Tree y aproximadamente 15 minutos para Random Forest en el dataset grande.

XGBoost también muestra un rendimiento destacado con valores cercanos a 0.996 para accuracy, F1-score, recall y precision en ambos tamaños de dataset. Aunque ligeramente inferior a Decision Tree y Random Forest en algunos casos, sigue siendo una opción robusta para clasificación en datos no equilibrados, con tiempos de ejecución que oscilan alrededor de los 16 minutos en el dataset grande.

LightGBM exhibe un rendimiento variable dependiendo del tamaño del dataset y la distribución de clases. Mientras que obtiene valores altos en métricas como accuracy y F1-score (alrededor de 0.88), muestra una ligera disminución en precisión en comparación con otros modelos evaluados. Los tiempos de ejecución para LightGBM son significativos, con aproximadamente 15 minutos en el dataset grande.

Las Redes Neuronales Artificiales muestran un rendimiento más bajo en comparación con los modelos basados en árboles y boosting. Aunque logran un accuracy y F1-score de aproximadamente 0.70 en el dataset grande, su precisión y recall son inferiores en comparación con Decision Tree, Random Forest y XGBoost. Además, los tiempos de ejecución son considerablemente mayores, superando los 47 minutos en el dataset grande.

Naive Bayes, como en las métricas anteriores, muestra un rendimiento más bajo en todos los indicadores de rendimiento en datasets no equilibrados. Con valores de accuracy y F1-score alrededor de 0.39 para el dataset grande, muestra una limitación significativa en la capacidad para manejar relaciones complejas y desbalances en los datos, a pesar de su rápido tiempo de ejecución.

Clasificación Múltiple						
Datasets No Equilibrados						
Codificación por Etiqueta						
Dataset	Modelo	Accuracy	F1-Score	Recall	Precision	Tiempo
Pequeño	Naive Bayes	0.4387	0.3662	0.4387	0.6859	0:00:05.600342
Pequeño	Decision Tree	0.9946	0.9946	0.9946	0.9946	0:00:08.313975
Pequeño	R. Forest	0.9957	0.9957	0.9957	0.9957	0:01:02.163253
Pequeño	XGBoost	0.9960	0.9960	0.9960	0.9960	0:03:41.797997
Pequeño	LightGBM	0.6129	0.6052	0.6129	0.6813	0:01:31.932975
Pequeño	CatBoost	0.9958	0.9957	0.9958	0.9957	0:36:40.668191
Pequeño	R. N. ANN	0.5178	0.5689	0.5178	0.5689	0:04:59.943427
Grande	Naive Bayes	0.3928	0.2524	0.3928	0.5259	0:00:46.108898
Grande	Decision Tree	0.9966	0.9966	0.9966	0.9966	0:02:04.907613
Grande	R. Forest	0.9972	0.9972	0.9972	0.9972	0:14:54.834946
Grande	XGBoost	0.9966	0.9966	0.9966	0.9966	0:16:21.867416
Grande	LightGBM	0.8806	0.8810	0.8806	0.8859	0:15:07.119466
Grande	CatBoost	-	-	-	-	5h+
Grande	R. N. ANN	0.6968	0.6099	0.6968	0.5949	0:47:30.443626

Tabla 48: Métricas de rendimiento para type con datos no equilibrados. Fuente: Elaboración propia

Además, CatBoost no se pudo completar la ejecución en el dataset grande, tanto para el dataset equilibrado como para el no equilibrado, dentro del tiempo límite que expone Google Colab, lo cual sugiere que puede enfrentar desafíos significativos al manejar datasets de gran tamaño. Esto destaca la sensibilidad de CatBoost a la distribución de clases y la complejidad computacional asociada.

### 10.3.2 COMPARACIÓN ENTRE DATASETS

Basado en los datos proporcionados, se observa una variación significativa en las métricas de rendimiento de los modelos de clasificación en función de si el dataset está equilibrado o no, así como del tamaño del dataset (pequeño o grande).

#### Dataset Equilibrado vs. No Equilibrado:

En datasets equilibrados, modelos como Decision Tree, Random Forest, XGBoost, LightGBM y CatBoost muestran una capacidad casi perfecta para clasificar correctamente, alcanzando valores muy altos cercanos a 1.0 en métricas como precisión y recall. Por el contrario, en datasets no equilibrados, aunque estos modelos mantienen un rendimiento alto (con valores cercanos a 1.0), se observan ligeros descensos en las métricas debido al desbalance de clases. Este efecto es más notable en modelos como LightGBM y Redes Neuronales Artificiales, donde la precisión puede verse más afectada por el desequilibrio.

El F1-score, que es la medida armónica de precisión y recall, también puede disminuir en casos extremos de desbalance, especialmente notable en modelos como LightGBM en datasets pequeños para clasificación múltiple desequilibrada. En casos extremos de desbalance, el F1-score puede disminuir si el modelo no logra capturar correctamente las instancias de las clases minoritarias.

Aunque el tiempo de ejecución suele ser más afectado por el tamaño del dataset que por el desequilibrio per se, algunos modelos sensibles al desbalance pueden experimentar una ligera variación en el tiempo de ejecución debido a la necesidad de tomar decisiones más complejas para manejar las clases desbalanceadas.

#### Tamaño del Dataset (Pequeño vs. Grande):

En general, modelos como Decision Tree, Random Forest, XGBoost y CatBoost mantienen un alto rendimiento tanto en datasets pequeños como grandes, con valores cercanos a 1.0 en las métricas de precisión y recall. Sin embargo, modelos como LightGBM pueden mostrar una ligera disminución en su rendimiento en datasets grandes en comparación con los pequeños.

La capacidad para mantener altos niveles de recall y precisión es consistente en ambos tamaños de dataset para la mayoría de los modelos, aunque pueden existir fluctuaciones menores.

El tiempo de ejecución tiende a ser más prolongado en datasets grandes debido al mayor volumen de datos procesados. Modelos como Random Forest y Redes Neuronales Artificiales muestran tiempos de ejecución más largos en datasets grandes en comparación con los pequeños. Es notable que modelos como CatBoost pueden ser particularmente sensibles al tamaño del dataset, con tiempos de ejecución significativamente más largos en datasets grandes, tanto equilibrados como no equilibrados.

Además, con datasets pequeños, existe un riesgo mayor de sobreajuste (overfitting), donde el modelo aprende demasiado de los datos de entrenamiento específicos y no generaliza bien con nuevos datos. Por otro lado, con datasets grandes, se reduce este riesgo ya que hay suficientes datos para representar adecuadamente la distribución del problema.

### 10.3.3 RENDIMIENTO DE LOS MODELOS

Como se puede observar en la Figura 58, de manera gráfica, se presentan las métricas de rendimiento (accuracy, precision, recall y F1-score) para cada modelo entrenado en un contexto binario utilizando diferentes datasets. La figura destaca el desempeño de los modelos Decision Tree, Random Forest, XGBoost, LightGBM y CatBoost, como se discutió anteriormente.

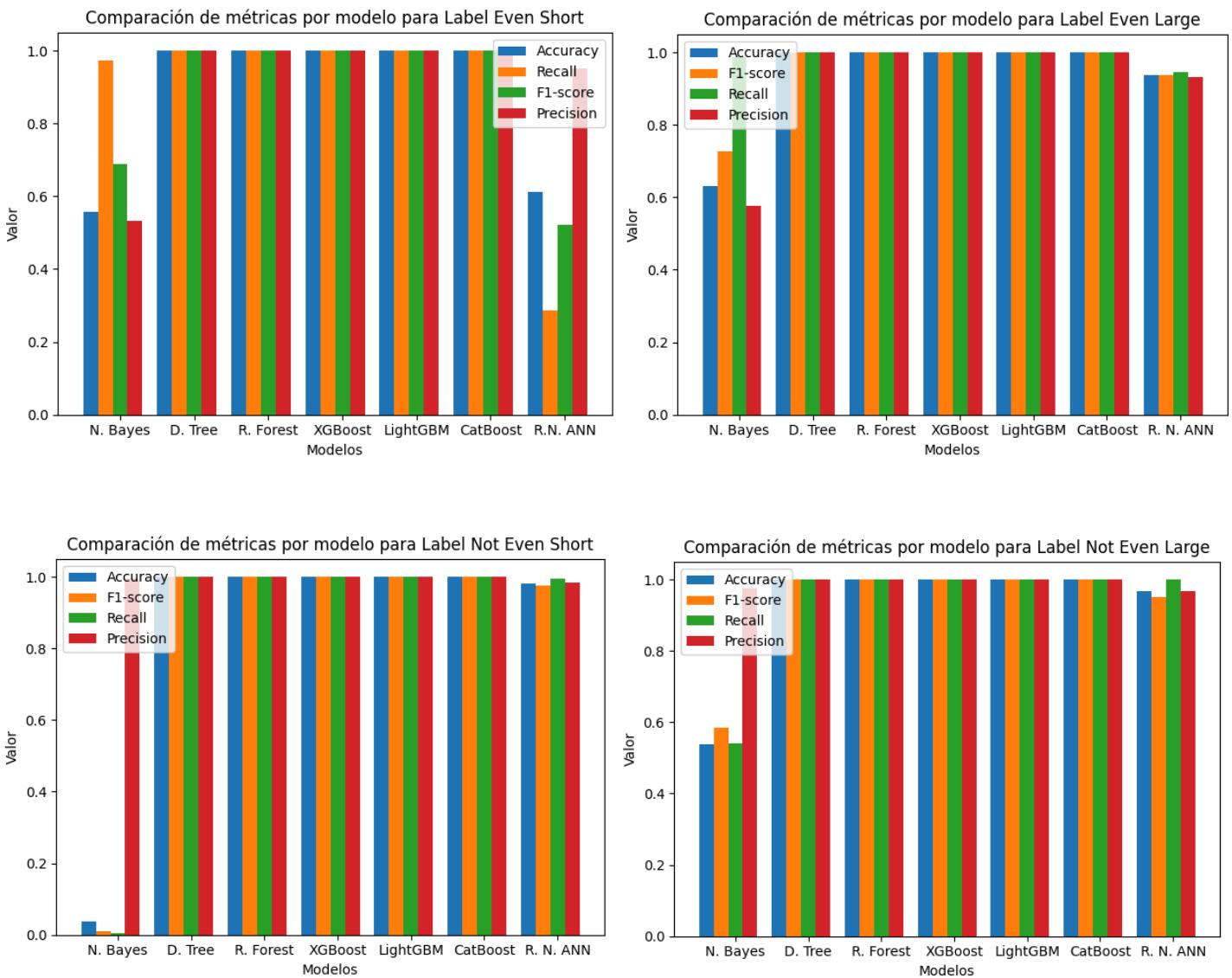


Figura 58: Métricas de rendimiento para label. Fuente: Elaboración propia

En la Figura 59 se presentan de forma gráfica las métricas de rendimiento (accuracy, precision, recall y F1-score) para cada modelo entrenado en un contexto de multiclasicación utilizando diversos datasets. La figura destaca el desempeño de modelos como Decision Tree, Random Forest y XGBoost, discutidos previamente. Además, se observa la falta de ejecución de CatBoost en datasets de gran tamaño, así como el rendimiento inferior de Naive Bayes en general, LightGBM y red neuronal ANN en datasets no equilibrados.

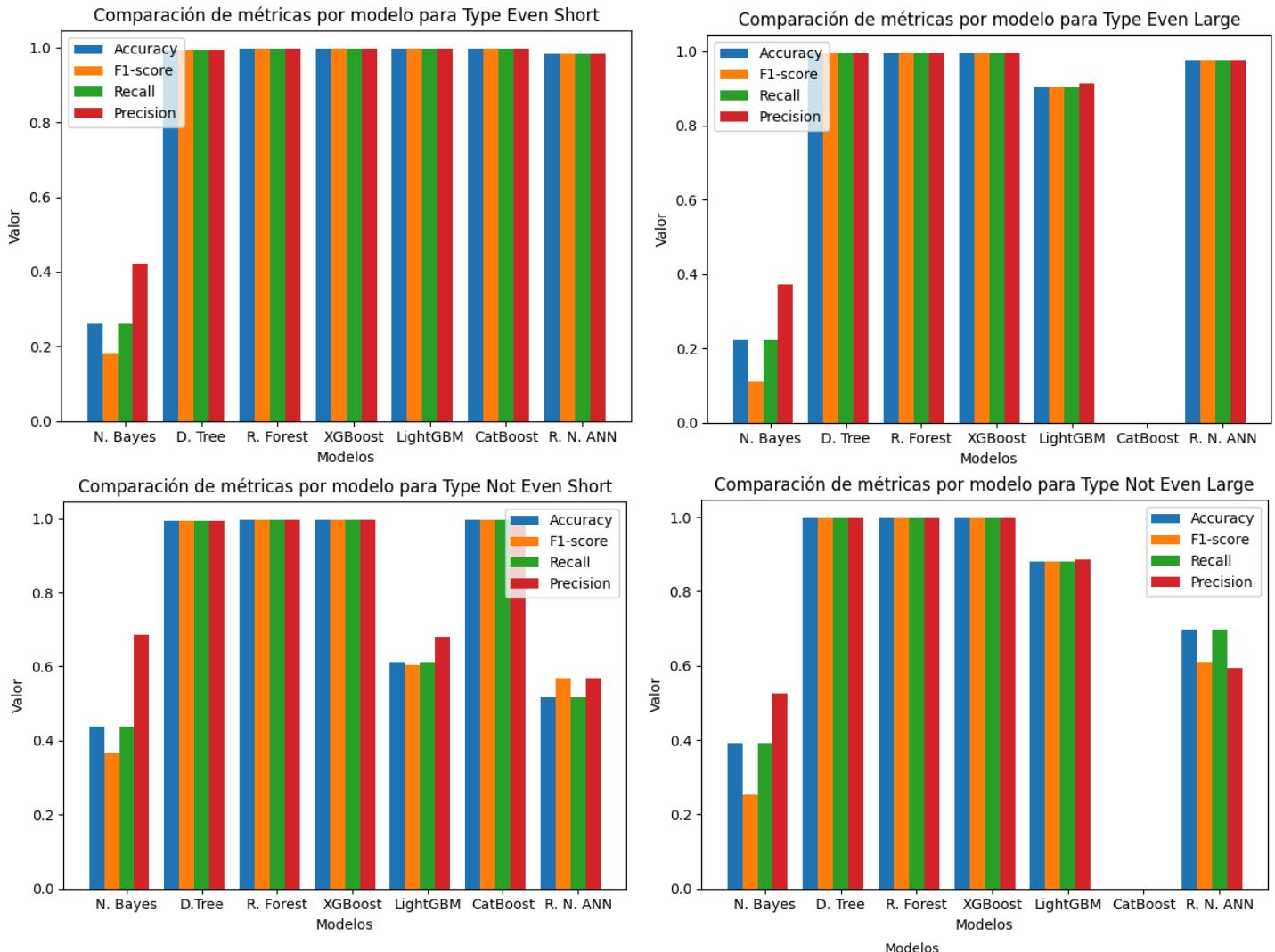


Figura 59: Métricas de rendimiento para type. Fuente: Elaboración propia

### 10.3.4 ANÁLISIS DE ERRORES

En este apartado vamos a analizar las métricas de confusión de distintos modelos. Observaremos una matriz de confusión de un modelo que generaliza correctamente, otra que lo hace moderadamente, y finalmente una que no generaliza correctamente.

En la Figura 60 se muestra la matriz de confusión del modelo XGBoost entrenado con el dataset de gran tamaño equilibrado.

El modelo demuestra una capacidad casi perfecta para predecir tanto los ataques como los datos benignos, alcanzando una precisión cercana al 100%. Se observan solo 2 falsos positivos de 688.717 predicciones totales, lo cual es un dato notablemente bajo. Respecto a los falsos negativos, el modelo clasifica incorrectamente solo 5 de 688.717 casos, lo cual también es muy bueno. Además, ha clasificado correctamente 688.715 verdaderos positivos y 688.712 verdaderos negativos.

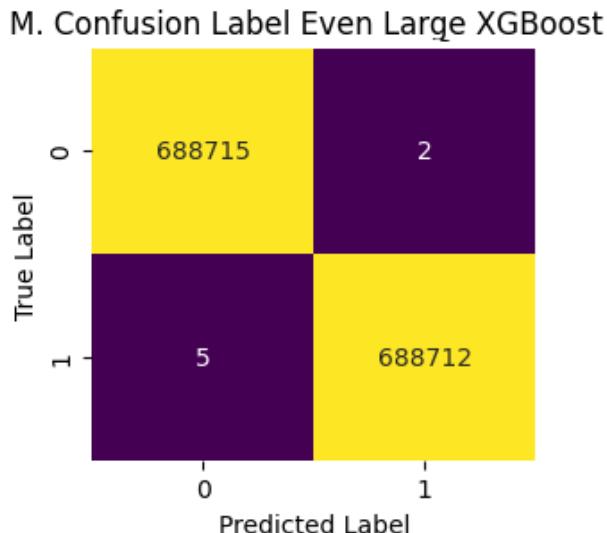


Figura 60: Matriz de confusión para label con dataset grande equilibrado. Fuente: Elaboración propia

Por lo contrario, en la Figura 61 se presenta la matriz de confusión del modelo Naive Bayes entrenado con el dataset de menor tamaño no equilibrado. El modelo muestra una capacidad baja para predecir tanto los verdaderos positivos como los verdaderos negativos, con un total de 3 falsos positivos de 100.000 predicciones totales, lo cual indica una tasa aceptable. Por otro lado, en cuanto a los falsos negativos, el modelo clasifica incorrectamente 96.186 de 96.647 casos negativos, lo cual sugiere una mayor dificultad en la clasificación de esta clase. Además, ha clasificado correctamente 3.350 verdaderos positivos y 461 verdaderos negativos de 100.000 casos.

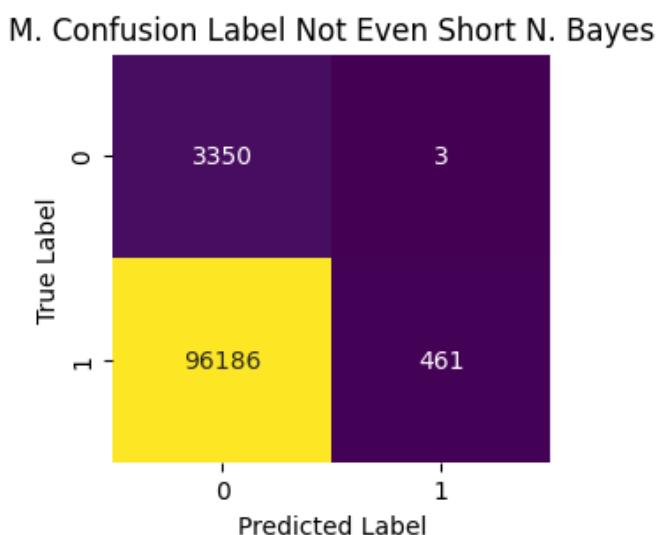


Figura 61: Matriz de confusión para label con dataset pequeño no equilibrado. Fuente: E. propia

Por último, quería comentar el caso del modelo de Redes Neuronales ANN cuando el dataset es de menor tamaño y está equilibrado, como se muestra en la Figura 62. La matriz de confusión revela que el modelo ha logrado clasificar correctamente 46.839 instancias de la clase positiva (verdaderos positivos) de un total de 50.000, lo que indica una predicción casi perfecta para esta clase. Por lo que se han clasificado incorrectamente tan solo 3.161 casos como positivos que en realidad pertenecían a la clase negativa (falsos positivos).

Sin embargo, el modelo ha tenido más dificultades para clasificar la clase negativa. Se observa que de 50.000 instancias que realmente pertenecían a la clase negativa, el modelo ha predicho incorrectamente 35.740 como positivos (falsos negativos). Esto sugiere una tasa relativamente alta de falsos negativos en comparación con los verdaderos negativos correctamente clasificados, que fueron 14.260.

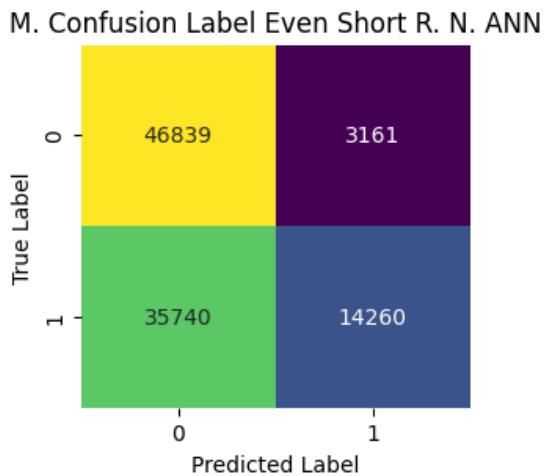


Figura 62: Matriz de confusión para label con dataset pequeño equilibrado. Fuente: E. propia

Se han analizado las matrices de confusión anteriores utilizando varios casos de ejemplo. Se observó que el modelo Naive Bayes mantuvo un rendimiento consistentemente bajo en todos los datasets, como se mostró en la figura 56. Además, se discutió el desempeño de la Red Neuronal en un caso específico, donde el modelo mostró un rendimiento notablemente inferior.

No se han analizado las demás matrices debido a que, como se observó en el modelo XGBoost en la figura 55, proporcionaron resultados satisfactorios.

En la Figura 63 se muestra la matriz de confusión del modelo Random Forest entrenado con el dataset de gran tamaño equilibrado. El modelo demuestra una capacidad robusta para clasificar múltiples clases, como se puede observar en los valores de la matriz. Destaca por su capacidad de clasificar correctamente una gran cantidad de verdaderos positivos y negativos en la mayoría de las clases, reflejando su habilidad para generalizar bien en este escenario complejo y equilibrado. Se observan algunos errores, especialmente en las clases menos representadas, pero en general, el modelo muestra un desempeño sólido en la clasificación multiclase de este dataset particular.

En este tipo de matrices no se emplean los términos "positivos" y "negativos" como en la clasificación binaria, ya que estamos tratando con una clasificación multiclase donde se distinguen valores del 0 al 9. Por lo tanto, nos referiremos a clasificaciones "verdaderas" de cada clase, por ejemplo, verdaderos 0, y a clasificaciones "falsas" de cada clase, como falsos 0, y así sucesivamente para el resto de clases. De forma genérica, se observan muchos más casos predichos de forma verdadero 0-9 que los casos predichos como falsos 0-9.

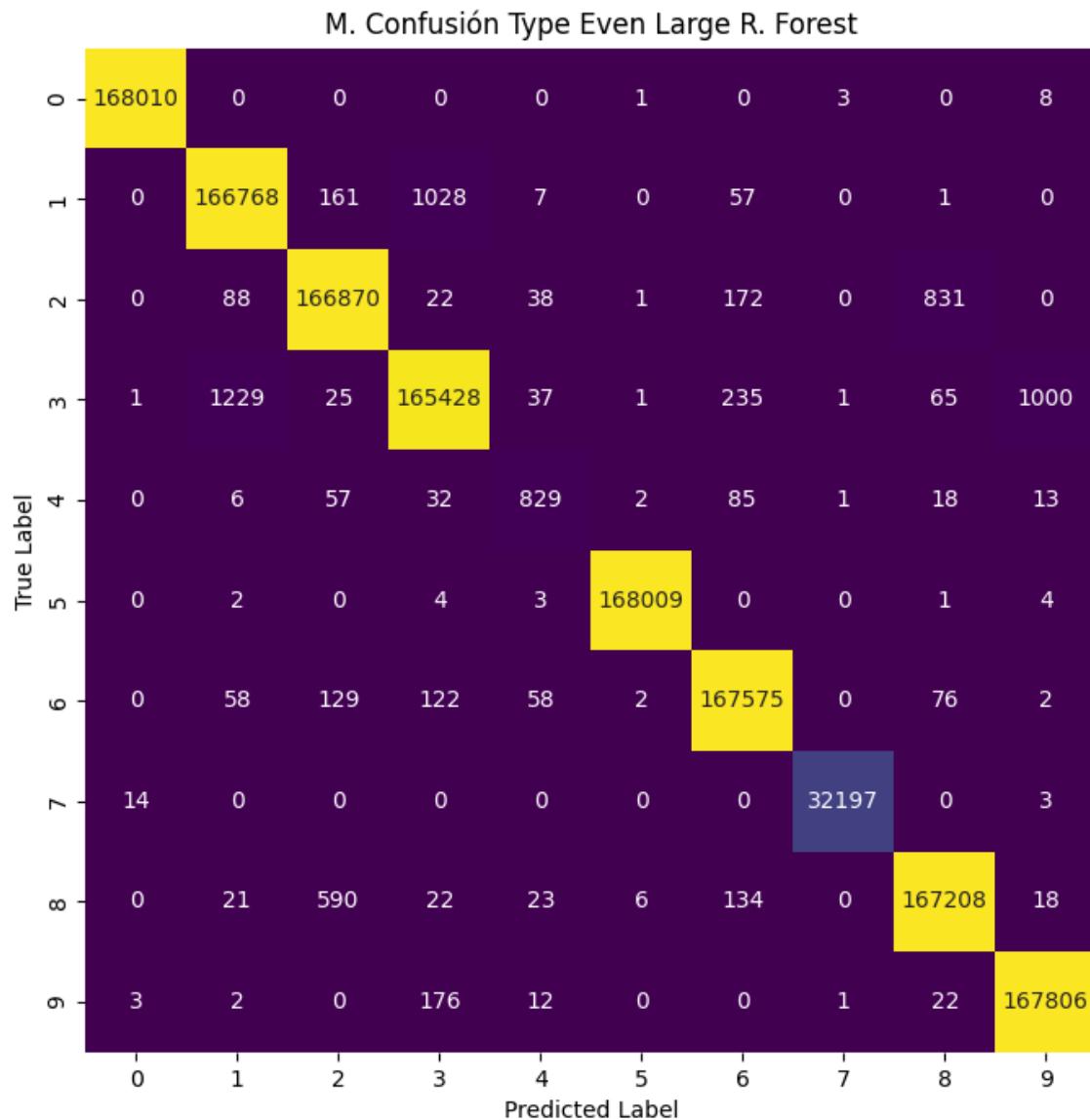


Figura 63: Matriz de confusión para type con dataset grande equilibrado. Fuente: Elaboración propia

En la Figura 64 se muestra la matriz de confusión del modelo Naive Bayes entrenado con un dataset grande y equilibrado para la clasificación multiclase. El modelo revela un rendimiento bajo, como ha ido mostrando en el resto de ejecuciones que se han realizado, en la clasificación de las diferentes clases representadas en el dataset. Se observa que el modelo no ha tenido éxito en ninguna clase salvo la clase 8 que ha logrado clasificar casi en su totalidad.

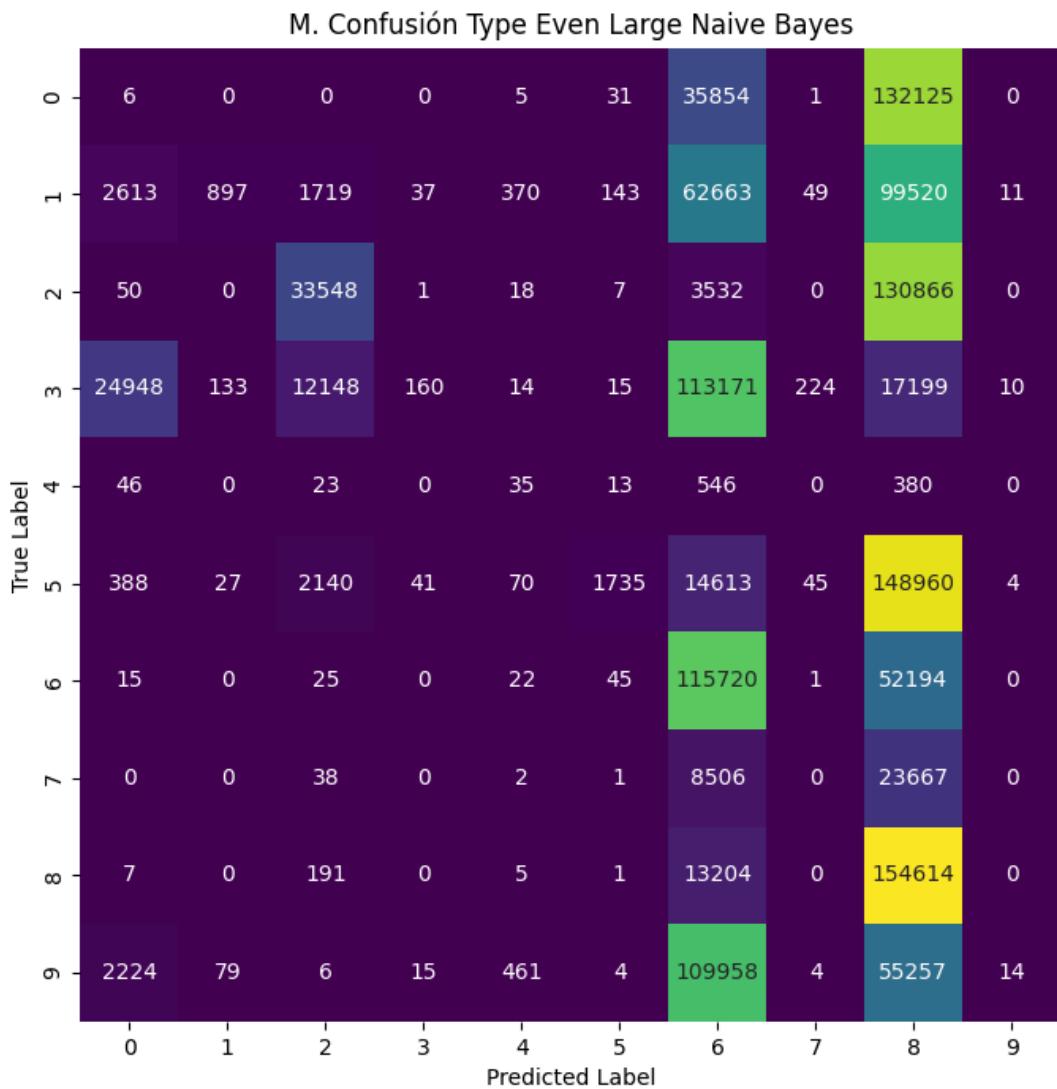


Figura 64: Matriz de confusión para type con dataset grande equilibrado. Fuente: Elaboración propia

En la Figura 65 se muestra la matriz de confusión del modelo LightGBM entrenado con un dataset pequeño y no equilibrado para la clasificación multiclase. El modelo revela un rendimiento variado en la clasificación de las diferentes clases representadas en el dataset. Se observa que el modelo ha tenido éxito en clasificar correctamente 4 casos que son las clases 1,2,8 y 9 como verdaderos y negativos en varias clases.

Sin embargo, también se pueden identificar errores significativos en la clasificación, especialmente en clases menos frecuentes donde se observan valores bajos de verdaderos positivos. Esto indica que el modelo puede tener dificultades para generalizar en clases minoritarias debido al desequilibrio en los datos. A pesar de estos desafíos, LightGBM muestra una capacidad considerable para manejar la clasificación multiclase en este escenario específico, destacándose en la precisión en clases más representadas mientras que muestra oportunidades para mejorar en clases menos frecuentes.

También es importante mencionar que, en el contexto de la clasificación múltiple, las Redes Neuronales Artificiales (ANN) muestran desafíos similares a LightGBM cuando se enfrentan a datasets desequilibrados.

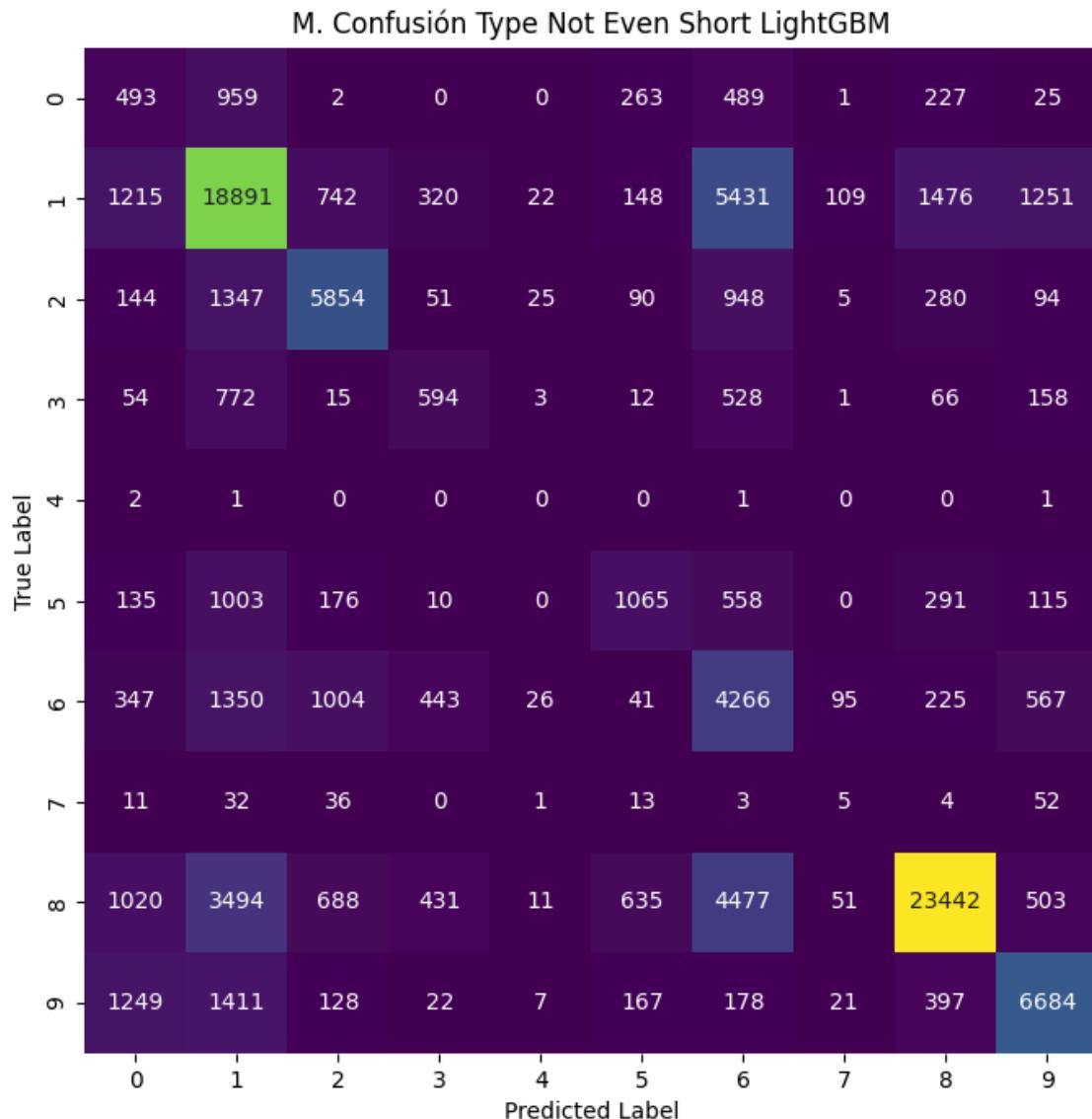


Figura 65: Matriz de confusión para type con dataset pequeño no equilibrado. Fuente: E. propia

### 10.3.5 MODELO A ESCOGER

En la elección del modelo más apropiado para este proyecto, se destaca la eficacia y versatilidad de los árboles de decisión, específicamente el algoritmo Decision Tree. Este modelo se ha seleccionado por varias razones fundamentales que lo hacen ideal para nuestras necesidades:

*Interpretabilidad y Transparencia:* Los árboles de decisión son modelos altamente interpretables, lo que significa que podemos entender fácilmente cómo toman decisiones clasificadorias. Esto es crucial para nuestro problema, ya que nos permite explicar y justificar las predicciones de manera clara.

*Consistencia en Rendimiento:*

- *Binaria (Label) con Dataset Pequeño y Grande:*
  - Para dataset pequeño, el Decision Tree obtuvo una precisión (accuracy) de 0.9997 y un F1-score de 0.9997 con un tiempo de entrenamiento muy corto (0:00:05.386587).
  - Para dataset grande, el rendimiento fue perfecto con una precisión (accuracy) y F1-score de 1.0, y un tiempo de entrenamiento razonable (0:01:12.157236).
- *Multiclas (Type) con Dataset Pequeño y Grande:*
  - Para dataset pequeño, el Decision Tree alcanzó una precisión (accuracy) de 0.9950 y un F1-score de 0.9950 en un tiempo muy corto (0:00:06.917932).
  - Para dataset grande, mantuvo un alto rendimiento con una precisión (accuracy) de 0.9942 y un F1-score de 0.9942 con un tiempo de entrenamiento adecuado (0:00:55.299085).

*Manejo Natural de Características Categóricas:* Dado que nuestro dataset contiene características categóricas y numéricas, los árboles de decisión son capaces de manejar naturalmente estas variables sin requerir transformaciones adicionales complejas.

*Flexibilidad y Adaptabilidad:* Los árboles de decisión pueden ser utilizados tanto para problemas de clasificación binaria como multiclas, lo cual es ventajoso si en el futuro necesitamos explorar modelos predictivos en diferentes dominios.

*Comparación con Otros Modelos:*

- Naive Bayes mostró un rendimiento inferior en ambos contextos, con una precisión (accuracy) y F1-score significativamente más bajos.
- Random Forest y XGBoost también tuvieron un rendimiento excelente, pero a costa de tiempos de entrenamiento mucho más largos.
- LightGBM y CatBoost presentaron buenos resultados, pero con variabilidad en el rendimiento dependiendo del equilibrio del dataset.
- Redes Neuronales (ANN) tuvieron un rendimiento aceptable, pero con tiempos de entrenamiento muy largos y una mayor complejidad.

En resumen, el modelo Decision Tree se posiciona como la mejor opción para abordar este proyecto dada su capacidad para proporcionar resultados interpretables y precisos en un contexto de datasets variados y mixtos, manteniendo un excelente balance entre rendimiento y eficiencia computacional.

## 10.4 DISEÑO DEL NIDS-AI E IMPLEMENTACIÓN EN LOCAL

Para nuestro entorno de pruebas local, el diseño e implementación de un Sistema de Detección de Intrusiones basado en Inteligencia Artificial (NIDS-AI), implica la configuración cuidadosa de varios componentes fundamentales para asegurar su funcionamiento efectivo y eficiente dentro de un entorno local.

Este proceso abarca desde la infraestructura necesaria hasta la implementación de herramientas específicas para la captura, análisis y visualización de datos de seguridad.

### 10.4.1 INFRAESTRUCTURA

En la etapa inicial, definimos y configuramos la infraestructura requerida para soportar el NIDS-AI. Esto incluye la utilización de contenedores Docker para facilitar la implementación y gestión de servicios críticos como Zeek, Kibana, Elasticsearch y Filebeat. Docker proporciona un entorno seguro y aislado para ejecutar estos servicios, permitiendo una fácil escalabilidad y mantenimiento.

Para la recolecta de datos en tiempo real, se ha utilizado la herramienta Zeek, que nos permite mediante un archivo .zeek asegurar la calidad de los datos que se obtienen en tiempo real. Con Elastic Stack nos permitirá visualizar el tráfico que va recibiendo la aplicación Zeek, para poder monitorizar el tráfico que se mueve por la red.

Además se utilizará la librería de tkinter en Python para tener un formato más visual de los resultados que nuestro NIDS-AI va proporcionando a medida que se va analizando el tráfico en tiempo real. En la figura 66 podemos observar más en detalle el flujo de trabajo que tendrá nuestro NIDS.

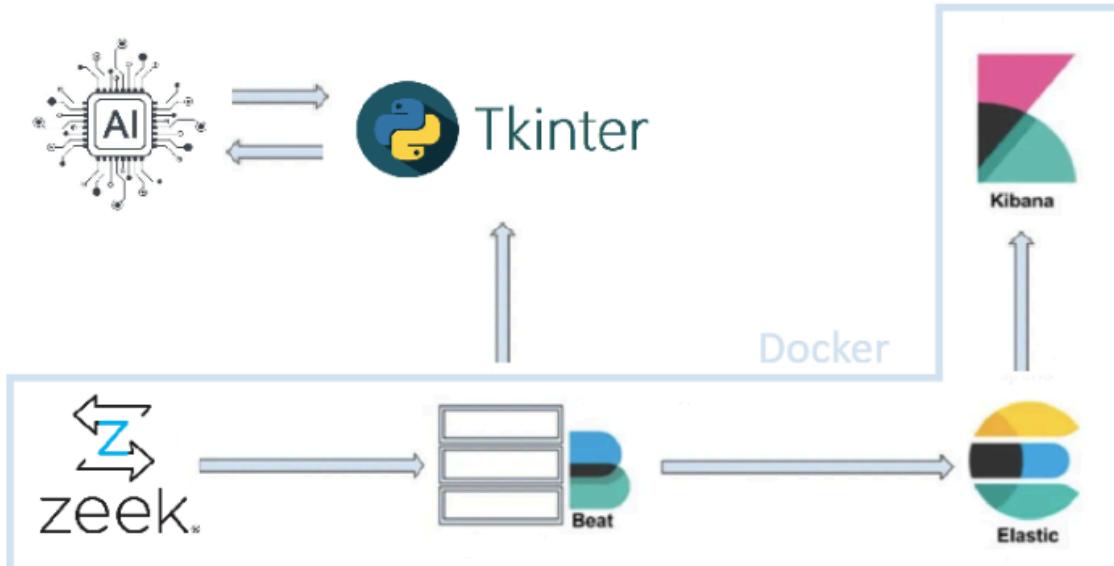


Figura 66: Flujo de trabajo del NIDS-AI. Fuente: Elaboración propia

## 10.5 CAPTURA DE TRÁFICO REAL CON ZEEK

Para capturar tráfico en tiempo real en nuestro entorno de pruebas, utilizaremos Zeek (anteriormente conocido como Bro), un potente marco de análisis de tráfico de red. Ejecutaremos Zeek dentro de un contenedor Docker para garantizar un entorno aislado y replicable. Para asegurar que Zeek recopile toda la información posible, configuraremos el contenedor Docker para replicar una dirección de host, evitando el uso de NAT. Esto es crucial porque ejecutar Zeek bajo un contenedor con NAT limitaría su capacidad para recolectar datos completos del tráfico de red. De esta manera, Zeek operará como si estuviera ejecutándose directamente en el host, asegurando la captura completa y precisa del tráfico de red en tiempo real.

Para este servicio, creé un Dockerfile utilizando Alpine Linux [44] como base para mantener el tamaño pequeño y eficiente del contenedor. El Dockerfile consta de tres etapas: builder, geoip y la etapa final.

En la etapa de builder, declaramos la versión de zeek a utilizar que en nuestro caso es la versión 4.1.1. Seguidamente, instalamos todas las dependencias necesarias para el entorno:

---

```
1 FROM alpine:3.14 as builder
2 RUN apk add --no-cache zlib openssl libstdc++ libpcap libgcc
3 RUN apk add --no-cache -t .build-deps \
    bsd-compat-headers \ libmaxminddb-dev \ linux-headers \ openssl-dev \
    libpcap-dev \ python3-dev \ zlib-dev \ binutils \ fts-dev \ cmake \ clang
    \ bison \ bash \ swig \ perl \ make \ flex \ git \ g++ \ fts
```

---

Después de instalar las dependencias necesarias, pasamos a la clonación del repositorio de Zeek, más concretamente de la versión prevista 4.1.1, en el directorio /tmp del contenedor. Cuando acabamos de clonar el repositorio, y con todas las dependencias instaladas, compilamos el servicio con el comando clang y utilizando varias opciones de configuración para reducir el tamaño y deshabilitar componentes no necesarios:

---

```
4 RUN echo "=> Cloning zeek..." \
&& cd /tmp \
&& git clone --recursive --branch v$ZEEK_VERSION https://github.com/zeek/zeek.git
5 RUN echo "=> Compiling zeek..." \
&& cd /tmp/zeek \
&& CC=clang ./configure --prefix=/usr/local/zeek \
--build-type=MinSizeRel \
--disable-broker-tests \
--disable-zeekctl \
--disable-auxtools \
--disable-python \
&& make -j 2 \
&& make install
```

---

Para continuar con la compilación de Zeek, se instalan y configuran varios plugins y paquetes de Zeek:

El plugin `af_packet` permite a Zeek capturar paquetes directamente desde interfaces de red utilizando el mecanismo “AF\_PACKET” del kernel de Linux. Esto puede ser más eficiente que otros métodos de captura como `libpcap`. Se utiliza principalmente para entornos de alta velocidad donde se requiere un rendimiento de captura más eficiente.

El paquete `file-extraction` permite a Zeek extraer archivos de los flujos de red. Es útil para análisis forense y detección de malware, donde es necesario capturar archivos transferidos a través de la red. Este paquete, junto con la configuración que se le da al servicio Zeek permite extraer automáticamente archivos de sesiones HTTP, FTP, SMTP, entre otros.

El paquete `Community-Id` genera identificadores únicos y consistentes para flujos de red. Estos identificadores son utilizados para correlacionar flujos de los eventos de red, ya que a la hora de unificar el flujo de conexiones con los datos de HTTP, DNS, ... se necesita de un identificador. Por defecto existe un identificador llamado `uid`, pero es un identificador bastante pequeño en longitud, como el flujo de red es constante es posible llenar todos los `uids` muy rápidamente, por ello se utiliza `community-id` que genera un identificador para manejar grandes cantidades. Por ejemplo:

Community-id: 1:HgdQJEEQt5k0ARnEdxRtT0n7v4=

uid: CdkjTp2fHJQ1bGeW25

Por último el paquete `json-streaming-logs` configura Zeek para exportar logs en formato JSON, lo cual es útil para integrar Zeek con sistemas de análisis y monitoreo que consumen JSON. Proporciona una forma eficiente de emitir logs en un formato estructurado y ampliamente compatible, lo que facilita la integración con el NIDS-AI.

Una vez configurado correctamente Zeek, pasamos a descargar datos de Geolocalización de IP para obtener información precisa del geoposicionamiento de cada una de las IP's analizadas. Esto nos va a permitir que Zeek determine la ubicación geográfica de una IP, lo cual es útil para detectar accesos sospechosos desde regiones inesperadas. Además ayuda a identificar patrones de ataque que podrían estar asociados a ciertas regiones geográficas y proporciona un contexto adicional sobre el origen del tráfico malicioso, facilitando una respuesta más informada.

Para lograr esto hemos ejecutado un `alpine 3.12`, un sistema ligero, para descargar los datos de GeolP [45] y descargamos los 3 siguientes archivos:

- **GeoLite2-City.tar.gz**: Contiene la base de datos para mapear IPs a ciudades específicas.
- **GeoLite2-Country.tar.gz**: Contiene la base de datos para mapear IPs a países.
- **GeoLite2-ASN.tar.gz**: Contiene la base de datos para mapear IPs a números de sistema autónomo (ASN), que son identificadores únicos para los ISPs.

Estos archivos descargados se descomprimen y se extraen para obtener los archivos con extensión `.mmdb`. Por último se mueven estos archivos al directorio `/usr/share/GeolP` donde estarán disponibles para que Zeek pueda acceder. Las sentencias utilizadas en el Dockerfile son:

```
1  FROM alpine:3.12 as geoip
2  ENV MAXMIND_CITY
https://geolite.maxmind.com/download/geoip/database/GeoLite2-City.tar.gz
3  ENV MAXMIND_CNTRY
https://geolite.maxmind.com/download/geoip/database/GeoLite2-Country.tar.gz
4  ENV MAXMIND ASN
http://geolite.maxmind.com/download/geoip/database/GeoLite2-ASN.tar.gz
5  ENV GITHUB_CITY
https://github.com/blacktop/docker-zeek/raw/master/maxmind/GeoLite2-City.tar.gz
6  ENV GITHUB_CNTRY
https://github.com/blacktop/docker-zeek/raw/master/maxmind/GeoLite2-Country.tar.gz
7  RUN cd /tmp
&& mkdir -p /usr/share/GeoIP \
&& wget ${GITHUB_CITY} \
&& tar xzvf GeoLite2-City.tar.gz \
&& mv GeoLite2-City*/GeoLite2-City.mmdb /usr/share/GeoIP/
```

---

En la etapa final de nuestro Dockerfile declaramos un entorno, el cual será el entorno final, utilizando alpine:3.14, para la ejecución de nuestro servicio Zeek.

En este entorno, inicialmente, se instalan las dependencias necesarias. A continuación se copia la aplicación de Zeek configurada desde el builder y se añade nuestro archivo local.zeek para que zeek sepa que tiene que capturar y cómo debe estructurar los logs finales. Para el correcto funcionamiento se utilizan dos ficheros del repositorio de blacktop para que el script de zeek le indique a la aplicación como tratar los datos.

Seguidamente copiamos las bases de datos de geolocalización de IPs en el entorno final y declaramos que nuestro entorno de trabajo es la carpeta pcap. Esta carpeta será donde Zeek guarde la información de los eventos en tiempo real de la red en los archivos logs correspondientes.

Finalmente se declaran los paths en la variable global PATH para ejecutar Zeek y se define el comando que se ejecutará cuando se levante el contenedor y la opción -h .

---

```
1  FROM alpine:3.14
2  RUN apk --no-cache add ca-certificates zlib openssl libstdc++ libpcap libgcc fts
libmaxminddb
3  COPY --from=builder /usr/local/zeek /usr/local/zeek
4  COPY local.zeek /usr/local/zeek/share/zeek/site/local.zeek
5  ADD
https://raw.githubusercontent.com/blacktop/docker-zeek/master/scripts/conn-ad
d-geodata.zeek \
/usr/local/zeek/share/zeek/site/geodata/conn-add-geodata.zeek
```

```
6  ADD https://raw.githubusercontent.com/blacktop/docker-zeek/master/scripts/log-pass
words.zeek \
/usr/local/zeek/share/zeek/site/passwords/log-passwords.zeek
7  COPY --from=geoip /usr/share/GeoIP /usr/share/GeoIP
8  WORKDIR /pcap
9  ENV ZEEKPATH
./data/config:/usr/local/zeek/share/zeek:/usr/local/zeek/share/zeek/policy:/usr/local/zeek/share/zeek/site
10 ENV PATH $PATH:/usr/local/zeek/bin
11 ENTRYPOINT ["zeek"]
12 CMD ["-h"]
```

---

### 10.5.1 SCRIPT LOCAL.ZEEK

El archivo local.zeek [46] creado es un archivo de configuración para Zeek que personaliza el comportamiento de Zeek para analizar, detectar y registrar diferentes actividades de red. En el Anexo se puede ver el código completo, el cual a continuación, desglosó cada sección y carga de scripts para entender mejor su funcionamiento:

Primero, hemos asegurado nuestra configuración estableciendo un valor de semilla para hashes de digestión, una medida esencial para la integridad de los IDs de archivos que creamos. Luego, se ha implementado un riguroso sistema de registro que nos permite rastrear qué scripts están activos en cada ejecución para una auditoría exhaustiva.

Para optimizar el rendimiento y precisión, hemos aplicado ajustes predeterminados, asegurando que Zeek esté afinado para capturar pérdidas y registrar estadísticas cruciales de memoria, paquetes y retrasos. Para detectar software vulnerable y cambios en versiones de software, se han cargado módulos específicos. Además, se han integrado firmas para detectar shells de Windows en texto plano.

Para monitorear protocolos específicos como FTP, SMTP, SSH y HTTP, se han habilitado detecciones específicas y se ha implementado la detección de aplicaciones web. En el ámbito de DNS, se ha implementado la detección de nombres externos para identificar posibles riesgos de seguridad.

Monitoreamos actividades sospechosas en sesiones FTP y realizamos seguimiento de activos de red y certificados SSL conocidos. También, se han enriquecido las detecciones geográficas en conexiones SSH, detectado intentos de fuerza bruta y monitoreado nombres de host interesantes.

Para proteger contra inyecciones SQL en tráfico HTTP, se han integrado módulos específicos y se ha habilitado el hash MD5 y SHA1 para el manejo de archivos. Además, se han extendido las alertas por correo electrónico para incluir nombres de host y se ha detectado el ataque Heartbleed en SSL.

Para un registro detallado, se ha habilitado el registro de VLAN y direcciones de capa de enlace. También, se utiliza CommunityID para asignar identificadores únicos a flujos de red y se enriquecen las detecciones GeolP personalizadas en conexiones. Se registran contraseñas en texto plano utilizadas en conexiones HTTP/FTP.

Finalmente, para facilitar la integración con nuestros sistemas de monitoreo, se han redirigido todos los registros a un formato JSON.

Esta configuración no solo mejora nuestra capacidad para detectar tráfico red en tiempo real, sino que también establece una base sólida para el monitoreo continuo y la seguridad de nuestra infraestructura de red guardando en archivos logs los eventos de red detectados.

## 10.5.2 RESPUESTAS LOG

Los registros de actividad generados por Zeek son esenciales para comprender el comportamiento del tráfico de red y detectar posibles anomalías o intrusiones. Estos logs son procesados y utilizados como entrada para el modelo de IA y también se envían a Elasticsearch a través de Filebeat para su indexación y almacenamiento.

La estructura de logs que genera Zeek con el archivo local.zeek configurado es la siguiente:

### Logs Generales

- **conn.log**: Registra todas las conexiones establecidas, incluyendo detalles como direcciones IP, puertos, protocolos, duración de la conexión, entre otros.
- **x509.log**: Contiene registros relacionados con certificados SSL/TLS, aunque aquí solo se registran los certificados de hospedaje.
- **ssl.log**: Registra eventos relacionados con el protocolo SSL/TLS y la negociación de certificados.
- **dns.log**: Registra consultas DNS realizadas desde tu red y respuestas recibidas.
- **http.log**: Contiene registros de todas las solicitudes y respuestas HTTP realizadas a través de la red.
- **smtp.log**: Registra detalles de transacciones SMTP, incluyendo envíos de correo electrónico.
- **files.log**: Contiene registros de todos los archivos detectados y manejados por Zeek.
- **notice.log**: Registra eventos de interés, como detecciones de vulnerabilidades, cambios de versión de software, y otros eventos significativos.

## Logs Específicos a Protocolos

- **ftp.log:** Registra actividades relacionadas con el protocolo FTP, como la autenticación y transferencia de archivos.
- **ssh.log:** Contiene registros de actividades SSH, como conexiones, intentos de inicio de sesión y comandos ejecutados.
- **smtp.log:** Registra detalles de transacciones SMTP, incluyendo envíos de correo electrónico.

## Logs Adicionales

- **capture\_loss.log:** Registra la pérdida estimada de captura de paquetes.
- **loaded\_scripts.log:** Registra qué scripts fueron cargados durante cada ejecución de Zeek.
- **stats.log:** Contiene estadísticas de memoria, paquetes y lag.
- **detect.log:** Registra detecciones de escaneos de red.
- **software.log:** Registra descubrimientos de versiones vulnerables de software.
- **windows\_shells.log:** Registra detecciones de shells de Windows en texto plano.
- **detect-webapps.log:** Registra detecciones de aplicaciones web.
- **detect-external-names.log:** Registra detecciones de nombres externos en DNS.
- **detect-sqli.log:** Registra detecciones de ataques de inyección SQL.
- **hashes.log:** Registra hash MD5 y SHA1 de todos los archivos.
- **detect-MHR.log:** Registra detecciones de sumas SHA1 en el Registro de Hash de Malware de Team Cymru.
- **extend-email.log:** Extiende las alertas de correo electrónico para incluir nombres de host.

## Otros Logs

- **conn\_mac.log:** Registra las direcciones de nivel de enlace para cada punto final de conexión.
- **conn\_vlan.log:** Registra VLANs para cada conexión.
- **community\_id.log:** Registra identificadores de comunidad para correlacionar flujos de red.

Todos los archivos log que escribe el servicio Zeek, los está guardando en el directorio /pcap. A pesar de la generación de todos los logs anteriores, en la parte de validación de la práctica del proyecto solo son útiles los archivos conn.log, http.log, dns.log, ssl.log y weird.log. Esto se debe a que los datos de nuestra validación son completados con esos 4 archivos.

## 10.6 TKINTER

En este proyecto, he utilizado Tkinter [47], más concretamente una versión customizada, para desarrollar el entorno visual que tendrá el NIDS-AI. Desde el principio, la elección de Tkinter fue clave debido a su facilidad de uso y su integración perfecta con Python, lo cual me ha permitido concentrarme en la funcionalidad central de la aplicación sin preocuparme demasiado por los detalles de implementación de la interfaz de usuario.

Uno de los aspectos más fascinantes de trabajar con Tkinter ha sido su flexibilidad para diseñar y personalizar widgets como botones, cuadros de texto, menús desplegables y más. Esto me ha permitido crear una experiencia de usuario intuitiva y amigable, donde los usuarios pueden interactuar de manera fluida con la aplicación y realizar tareas fácilmente.

Previamente al desarrollo visual del NIDS usando Tkinter, se ha realizado un diseño base en la herramienta Figma. Este diseño base se puede visualizar en la figura 67.

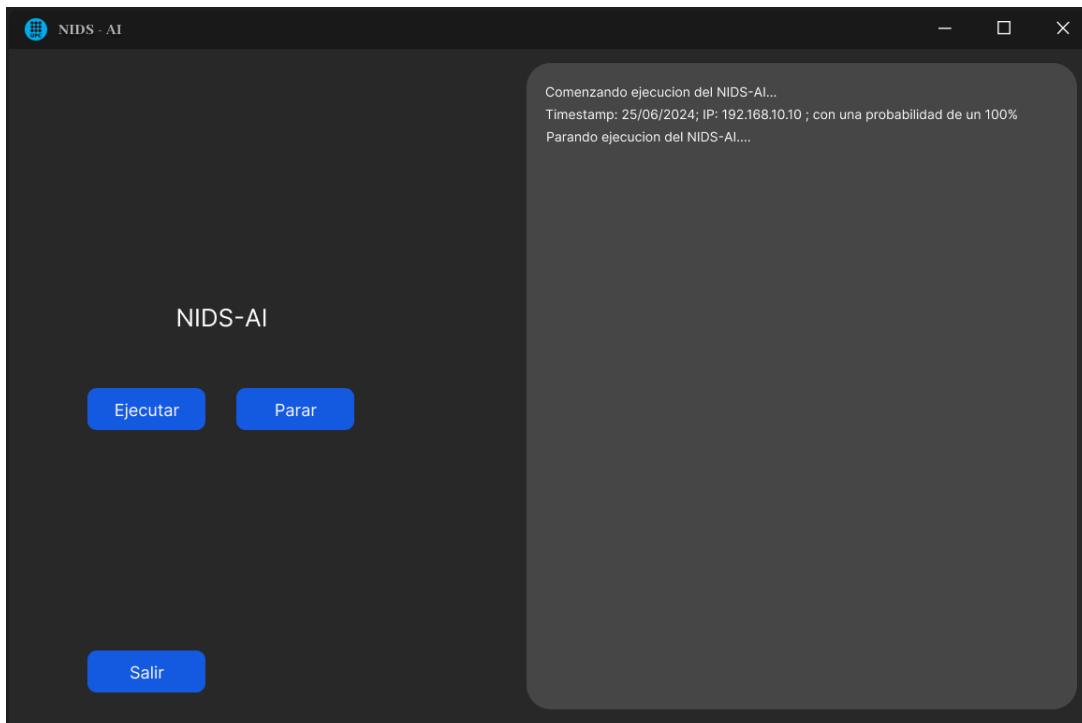


Figura 67: Diseño en Figma de la app principal del NIDS-AI. Fuente: Elaboración propia

En cuanto a la arquitectura interna del NIDS-AI, he seguido prácticas recomendadas para mantener un código limpio y modular. Para ello he creado diferentes clases, una para cada servicio. Una clase para la parte visual, otra para leer en tiempo real los logs de Zeek y otra para cargar el modelo de inteligencia artificial y hacer predicciones. Cada una de las clases se importan en el programa principal de ejecución.

Inicialmente, el programa principal carga la clase de Módulo. Esta clase está diseñada con una función de inicialización, una función de preprocesamiento y una función de predicción. La función de inicialización carga el modelo guardado en un archivo, luego carga los encoders utilizados en el entrenamiento del modelo, y finalmente guarda un diccionario con todas las claves que deben contener los datos para realizar predicciones.

La función de preprocessamiento recibe los datos en tiempo real de los logs de Zeek, los procesa aplicando un Label Encoder y selecciona solo los datos cuyas claves aparezcan en el diccionario inicializado previamente. Por último, la función de predicción recibe los datos preprocessados, los envía al modelo cargado y devuelve la predicción realizada por el modelo.

Seguidamente, el programa principal carga la parte visual del NIDS-AI, que se compone de un Tkinter personalizado. Esta parte visual consta principalmente de dos botones, uno para ejecutar y otro para parar la ejecución, y de un área de texto donde se imprimen los ataques detectados.

Finalmente, el programa principal inicia la clase de lectura. Esta última clase tiene la funcionalidad de leer en tiempo real el archivo conn.log que genera y rellena Zeek a medida que se recolectan datos. Mediante el identificador de cada dato que aparece en conn.log, se buscan más datos en los archivos http.log, dns.log, ssl.log, y weird.log para completar todas las claves que el modelo de inteligencia artificial usa. Todo esto se mantiene en un bucle infinito hasta el cierre de la aplicación.

La parte visual del proyecto nos permitirá identificar las detecciones realizadas por nuestro sistema NIDS-AI en tiempo real, mostrando información sobre las conexiones sospechosas. En la Figura 68 se muestra la parte visual de la aplicación del proyecto.

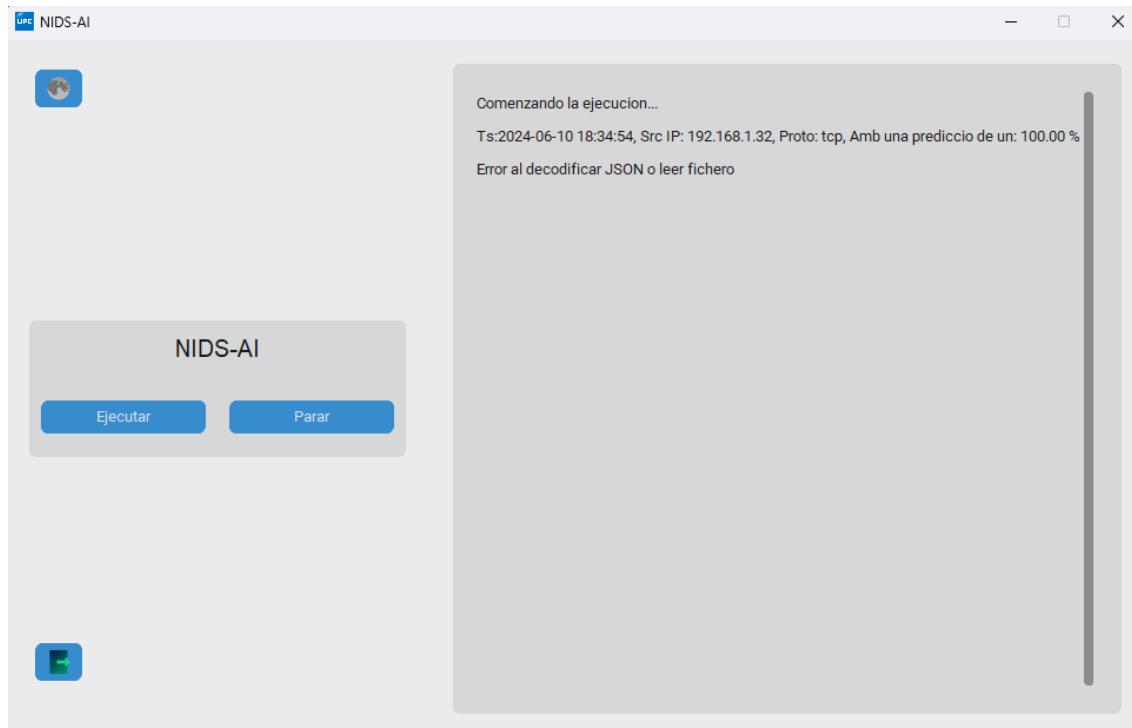


Figura 68: Diseño final de la app principal del NIDS-AI. Fuente: Elaboración propia

En la aplicación final para dar mejores resultados se ha implementado un botón en cada conexión destacada como sospechosa para visualizar los datos que contiene esta conexión en profundidad, ya que en la ventana principal no aparecen muchos datos.

En la Figura 69 se puede observar el diseño realizado en esta segunda ventana para mostrar más información importante del posible ataque.

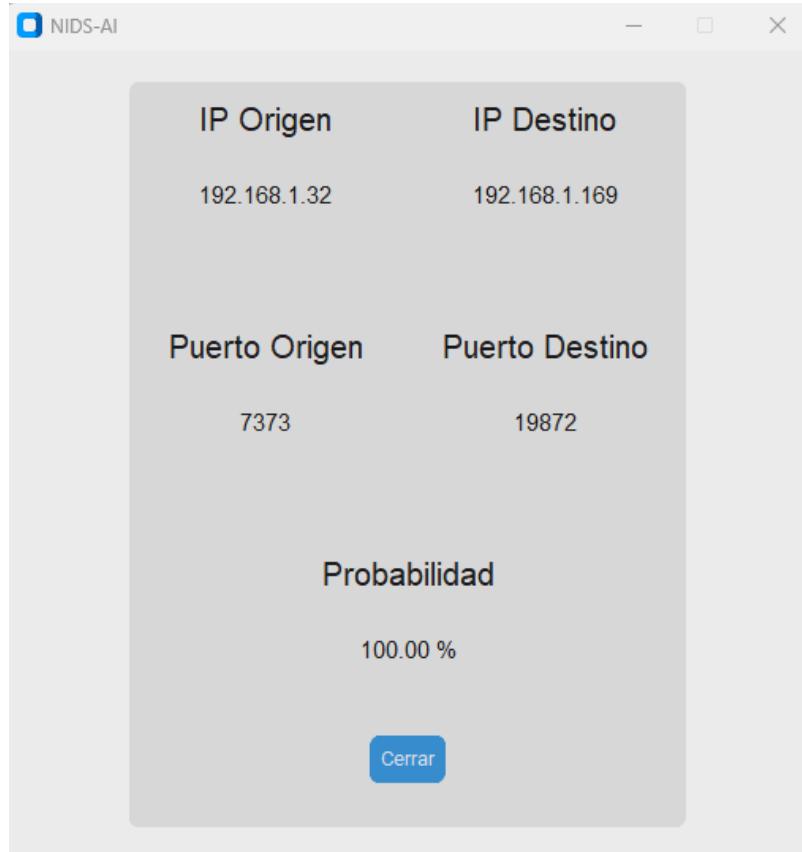


Figura 69: Diseño final de la app secundaria del NIDS-AI. Fuente: Elaboración propia

## 10.7 ELASTIC STACK

Inicialmente se configura Filebeat para que lea y envíe los datos, que aparecen en los archivos log generados por Zeek, a Elasticsearch.

Durante la configuración de Elasticsearch, se definen los índices y mappings [48] necesarios para almacenar y organizar los datos de manera eficiente, garantizando una recuperación rápida y precisa de la información relevante. Elasticsearch se utiliza para almacenar e indexar los registros de eventos, proporcionando una base de datos robusta y escalable para la consulta rápida de datos históricos y en tiempo real.

La configuración de Kibana implica la creación de paneles y visualizaciones personalizadas que reflejan las métricas y estadísticas clave generadas por Zeek. Estos dashboards proporcionan a los usuarios una vista comprensiva del estado de la seguridad de la red, permitiendo una respuesta rápida ante posibles amenazas. Durante la creación del panel que aparece en la figura 69, se ha debido indexar correctamente, con Filebeat, los datos que posteriormente se almacenan en Elasticsearch para su visualización en Kibana.

En conjunto, la implementación del NIDS-AI y su infraestructura asociada no solo fortalece la seguridad de la red mediante la detección proactiva de intrusiones, sino que también facilita la monitorización y gestión efectiva de eventos de seguridad a través de interfaces intuitivas y herramientas avanzadas de análisis de datos. Este enfoque integrado asegura que las organizaciones puedan responder de manera rápida y eficiente a las amenazas emergentes, protegiendo así sus activos críticos y datos sensibles.



Figura 70: Diseño del dashboard de Kibana. Fuente: Elaboración propia

## 10.8 EVALUACIÓN DEL NIDS-AI

Claramente visible en la figura 63, el NIDS-AI diseñado demuestra su capacidad para detectar amenazas en tiempo real. Para validar exhaustivamente su funcionamiento, luego de ejecutar la aplicación, se introdujo manualmente un valor específico previamente identificado y entrenado como un tipo de ataque en los datos de entrada simulados. Cuando el NIDS-AI procesa este dato, identifica con precisión el ataque y presenta una notificación clara en pantalla, demostrando su efectividad y fiabilidad en la detección oportuna de amenazas.

Además de la detección de amenazas, el NIDS puede mostrar la probabilidad de que un dato en particular sea sospechoso, demostrando la fiabilidad en su elección. Esto asegura la confiabilidad para el usuario, mostrando mediante un porcentaje la probabilidad que el NIDS asigna a una conexión como sospechosa.

La aplicación presenta un formato básico e intuitivo, facilitando su uso incluso para usuarios con pocos conocimientos, quienes pueden entender fácilmente su flujo.

El sistema también realiza un seguimiento activo de la integridad de los datos procesados por el NIDS-AI. Esto implica monitorear continuamente cualquier anomalía o dificultad en la lectura de datos, garantizando así un funcionamiento óptimo del sistema y evitando errores críticos en la captura y análisis de datos. Este enfoque fortalece la robustez y confiabilidad del sistema, asegurando una respuesta eficaz ante posibles fallos o interrupciones en la operación del NIDS-AI, mejorando significativamente la seguridad y el rendimiento general de la aplicación.

Finalmente, se llevaron a cabo pruebas unitarias manuales para validar la robustez de la aplicación, asegurando su correcto funcionamiento frente a posibles fallos que los usuarios podrían encontrar al usarla. Durante estas pruebas, se verificó el comportamiento adecuado de todos los botones y funcionalidades principales, garantizando así una experiencia de usuario fluida y confiable.

## 11. LEYES Y REGULARIZACIONES

Cualquier proyecto se encuentra bajo una serie de leyes y de regulaciones que deben cumplir. Principalmente se deben cumplir dos leyes, que cualquier proyecto de datos debe tener presente: la ley de protección de datos y la ley de propiedad intelectual.

### 11.1 NORMATIVA DEL TRABAJO FINAL DE CARRERA

Este trabajo está sujeto a una normativa establecida por la Universidad Politécnica de Catalunya, en la que se determinan las condiciones con las que el trabajo debe ser desarrollado. Este documento recoge información relativa a la carga de trabajo, inscripción y matriculación, las diferentes modalidades que se pueden realizar, las metas y métodos de evaluación, formalización del tribunal, contenidos de la memoria, y el marco jurídico impuesto por la Universidad Politécnica de Cataluña. [49]

### 11.2 LEY DE PROTECCIÓN DE DATOS

La Ley de Protección de Datos, como el Reglamento General de Protección de Datos (RGPD) [50] en la Unión Europea, establece un marco legal para la recolección, procesamiento y almacenamiento de datos personales. Esta ley tiene como objetivo principal proteger la privacidad y los derechos de las personas en relación con sus datos personales, asegurando que las organizaciones manejen esta información de manera responsable y segura. Para ello, impone una serie de obligaciones y responsabilidades tanto para los controladores de datos (quienes determinan los fines y medios del procesamiento de datos personales) como para los procesadores de datos (quienes procesan datos personales en nombre de los controladores)

Entre sus disposiciones más importantes se encuentra la necesidad de obtener el consentimiento explícito de los individuos antes de recopilar y utilizar sus datos personales. Esto significa que las organizaciones deben informar claramente a los interesados sobre qué datos se van a recoger, con qué propósito se utilizarán y cómo se almacenarán. Además, los individuos tienen el derecho de acceder a sus datos personales, rectificarlos si están incorrectos, borrarlos cuando ya no sean necesarios para los fines para los cuales fueron recopilados y oponerse a su procesamiento en determinadas circunstancias

La ley también establece principios fundamentales para el manejo de datos personales, tales como la legalidad, equidad y transparencia en el procesamiento de datos; la limitación de la finalidad, que indica que los datos deben ser recopilados con fines específicos, explícitos y legítimos; y la minimización de datos, que estipula que solo deben recopilarse los datos personales necesarios para cumplir con los fines establecidos. Asimismo, la exactitud es crucial, ya que los datos deben ser precisos y, cuando sea necesario, actualizados; la limitación del almacenamiento, que señala que los datos personales deben conservarse en una forma que permita la identificación de los interesados solo durante el tiempo necesario para los fines del procesamiento; y la integridad y confidencialidad, que requieren que los datos sean procesados de manera segura para protegerlos contra el procesamiento no autorizado o ilegal y contra la pérdida, destrucción o daño accidental

Para asegurar el cumplimiento de estas normativas, la ley requiere que las organizaciones implementen medidas técnicas y organizativas apropiadas.

Esto incluye la realización de evaluaciones de impacto de protección de datos cuando un tipo de procesamiento pueda representar un alto riesgo para los derechos y libertades de las personas, y la designación de un Delegado de Protección de Datos (DPO) en determinadas circunstancias. El DPO actúa como punto de contacto para los interesados y las autoridades de protección de datos, supervisa la estrategia de protección de datos y su implementación para garantizar el cumplimiento de los requisitos de la ley

En el contexto de la creación de un Sistema de Detección de Intrusiones en Redes (NIDS) con inteligencia artificial utilizando herramientas como Zeek, Kibana, Filebeat y Elasticsearch, no es fundamental garantizar que cumpla con Ley de Protección de Datos puesto que no recopila ningún dato del usuario. El manejo de los datos recopilados los administra directamente el usuario, por lo que no existe el cumplimiento de esta ley.

### **11.3 AVISO LEGAL DE LA APLICACIÓN**

Al acceder y utilizar esta aplicación, se acepta cumplir con los términos y condiciones descritos en este apartado. Si no se está de acuerdo con estos términos, no se tendrá el derecho al usufructo de la aplicación.

Todo el contenido incluido en la aplicación, como el código fuente, los algoritmos, la documentación, los gráficos, y otros materiales, está protegido por derechos de autor y otras leyes de propiedad intelectual. Estos derechos son propiedad exclusiva del creador. No se permite la reproducción, distribución, modificación, exhibición o ejecución pública de ningún contenido de la aplicación sin el permiso expreso y por escrito del propietario.

La aplicación puede incluir software de terceros que está licenciado bajo diferentes términos y condiciones. Es responsabilidad del usuario asegurarse de que cumple con las licencias aplicables al utilizar dicho software. Esto incluye, pero no se limita, las licencias de Zeek, Kibana, Filebeat y Elasticsearch. Los usuarios deben revisar y respetar los términos de estas licencias al utilizar la aplicación.

El uso de la aplicación es bajo su propio riesgo. La aplicación se proporciona "tal cual" y "según disponibilidad", sin garantías de ningún tipo, ya sean expresas o implícitas. No se garantiza que la aplicación sea segura, libre de errores, ininterrumpida o que cumpla con las expectativas. En ningún caso, yo como creador, seré responsable por cualquier daño, directo o indirecto, que surja del uso o la imposibilidad de uso de la aplicación.

Me reservo el derecho de modificar o discontinuar la aplicación en cualquier momento y sin previo aviso. Esto incluye la actualización, eliminación de funcionalidades o cambios en el acceso a la aplicación. Es responsabilidad del usuario revisar periódicamente este aviso legal para estar al tanto de cualquier cambio.

La privacidad de los usuarios es una prioridad. La aplicación implementa medidas de seguridad para proteger los datos recopilados contra accesos no autorizados, pérdidas o alteraciones. Los usuarios son responsables de mantener la confidencialidad de sus credenciales de acceso y cualquier otra información sensible.

Este aviso legal se rige y se interpreta de acuerdo con las leyes del país donde se desarrolla y opera la aplicación. Cualquier disputa que surja en relación con el uso de la aplicación estará sujeta a la jurisdicción exclusiva de los tribunales de dicho país.

## 11.4 TÉRMINOS Y CONDICIONES DE GOOGLE COLAB

Los términos y condiciones de Google Colab [51] regulan el uso de esta plataforma. Incluyen:

- Para utilizar Google Colab, es necesario tener una cuenta de Google.
- Prohibiciones sobre el uso malintencionado o ilegal.
- Los usuarios mantienen la propiedad de sus notebooks, pero otorgan a Google ciertos derechos de uso.
- Google no se hace responsable de pérdidas derivadas del uso de Colab.
- Google puede suspender o terminar el acceso a Google Colab en caso de violación de los términos y condiciones.
- No permite el uso Google Colab para transmitir, distribuir o almacenar datos que puedan violar las leyes y regulaciones de exportación y reexportación aplicables, incluidas las regulaciones de control de exportaciones de EE. UU
- Google se reserva el derecho de modificar o discontinuar Google Colab en cualquier momento, con o sin previo aviso.

## 11.5 TÉRMINOS Y CONDICIONES DE GIT

GitHub, como plataforma utilizada para alojar y colaborar en el desarrollo de nuestro sistema de detección de intrusiones en redes (NIDS), establece ciertas condiciones y limitaciones para su uso [52]. A continuación se detallan algunos de los términos más relevantes:

- Los usuarios deben utilizar GitHub de acuerdo con todas las leyes y regulaciones aplicables.
- Está prohibido usar GitHub para actividades ilícitas, abusivas o dañinas.
- Cumplir con la declaración de privacidad de GitHub.
- Los usuarios conservan la propiedad de los contenidos y proyectos que suben a GitHub. Al subir contenido, los usuarios otorgan a GitHub una licencia mundial, no exclusiva, sin royalties para alojar, almacenar, reproducir, modificar y mostrar el contenido.
- GitHub se compromete a proteger la privacidad y los datos personales de los usuarios. Los usuarios son responsables de la seguridad de sus propios proyectos y datos subidos a la plataforma.
- Los usuarios no deben subir contenido que infrinja los derechos de autor, marcas registradas u otros derechos de propiedad intelectual de terceros. GitHub tiene el derecho de eliminar contenido que considere infractor de los derechos de autor.

## 11.6 TÉRMINOS Y CONDICIONES DE LOS DATASETS

El uso de los datasets de UNSW-NB15 de UNSW Canberra at ADFA, específicamente TON para propósitos de investigación académica es gratuito y está autorizado de manera perpetua. Esto significa que los investigadores y estudiantes pueden utilizar estos datos sin costo alguno para realizar estudios, análisis y otros trabajos académicos, siempre y cuando se mantenga el propósito académico y no comercial del uso de los datos. Este permiso facilita la libre investigación y el avance del conocimiento en diversas áreas, asegurando que los datos sean accesibles para la comunidad académica global.

Sin embargo, el uso de los datasets para fines comerciales está condicionado a la obtención de un permiso explícito del autor, Dr. Nour Moustafa, quien ha afirmado sus derechos de autor sobre estos datos. Para utilizar los datasets en proyectos, productos o servicios comerciales, es necesario contactar al Dr. Moustafa y obtener su aprobación. Esto asegura que el uso comercial de los datos se realice de manera controlada y que los derechos del autor sean respetados.

Al utilizar los datasets, se deben seguir las siguientes condiciones:

- Los datasets pueden ser utilizados de forma gratuita para fines académicos y de investigación. Cualquier uso comercial debe ser autorizado por el Dr. Nour Moustafa.
- Dr. Nour Moustafa retiene todos los derechos de autor sobre los datasets. Se debe reconocer la autoría del Dr. Moustafa en cualquier publicación, presentación o producto que utilice estos datos.
- Para utilizar los datasets con fines comerciales, los interesados deben contactar al Dr. Nour Moustafa y solicitar permiso. La solicitud debe incluir una descripción del propósito comercial y cómo se utilizarán los datos.

Estas condiciones garantizan que el uso de los datasets sea respetuoso y beneficioso tanto para la comunidad académica como para el autor. Cumplir con estos términos y condiciones es esencial para evitar la desaprobación del Dr. Nour Moustafa respecto al uso de los datos.

## 11.7 LICENCIA DE LIBRERÍAS Y PROGRAMARIO

Cada una de las herramientas y librerías de software que se usa en el proyecto tiene su propia licencia de uso, que debe ser respetada tanto por los desarrolladores como por los usuarios.

Python se distribuye bajo la licencia Python Software Foundation License (PSF License), que es una licencia permisiva similar a la licencia BSD. Esta licencia permite el uso, la modificación y la distribución del código fuente de Python con muy pocas restricciones.

Luego tenemos Scikit-learn, una librería esencial para el aprendizaje automático. Esta se rige bajo la licencia BSD de 3 cláusulas, que también es bastante permisiva. Nos permite utilizar, modificar y distribuir el software, siempre y cuando mantengamos las atribuciones y no utilicemos el nombre de los autores para promocionar productos derivados sin permiso.

Ahora, pasando a herramientas del kit Elastic, como Kibana, Elasticsearch y Filebeat, nos encontramos con la licencia Elastic License 2.0 (ELv2). Esta licencia nos brinda la libertad de utilizar el software de forma gratuita en la mayoría de los casos de uso, incluidos los comerciales. Sin embargo, debemos cumplir ciertas condiciones, como no ofrecer el software como un servicio competidor.

Finalmente, tenemos a Docker que se distribuye bajo la licencia Apache 2.0, una licencia muy permisiva que nos permite usar, modificar y distribuir el software sin preocupaciones. Además, proporciona una patente explícita y una garantía de que el software está libre de reivindicaciones de patentes por parte de los contribuyentes.

## 11.8 PROPIEDAD INTELECTUAL

En el ámbito del machine learning y los sistemas de detección de intrusiones en redes (NIDS, por sus siglas en inglés), la propiedad intelectual juega un papel crucial. Es fundamental entender cómo afecta la propiedad intelectual al desarrollo y uso de tecnologías de machine learning.

Una persona que posee un algoritmo o sistema de machine learning no implica que tenga la propiedad intelectual sobre el contenido o los datos utilizados. La propiedad intelectual sobre estos algoritmos y sistemas debe ser claramente establecida y protegida a través de derechos de autor, patentes y licencias adecuadas.

Para certificar legalmente la propiedad de un algoritmo o sistema de machine learning, es necesario obtener una licencia sobre los derechos de la persona o empresa que ha creado dicho contenido. Esto implica formalizar la transferencia de derechos a través de acuerdos legales que especifiquen el uso permitido, las restricciones y las condiciones de explotación comercial del algoritmo o sistema.

Además, la propiedad intelectual en el ámbito del machine learning incluye la protección de los datos utilizados para entrenar los algoritmos. Los conjuntos de datos, especialmente aquellos que contienen información personal o sensible, deben gestionarse con cuidado para evitar violaciones de derechos y cumplir con las normativas de privacidad.

## 12. SOSTENIBILIDAD

### 12.1 ÁMBITO AMBIENTAL

La huella ecológica de este proyecto depende mucho de la tecnología detrás de los algoritmos de machine learning. En el caso de que se mejore el algoritmo por uno más efectivo y sostenible, mejoraría sustancialmente la huella ecológica del proyecto. De lo contrario, el proyecto probablemente se está volviendo cada vez más contaminante, ya que sería necesario ordenadores más potentes para poder mantener el sistema.

### 12.2 ÁMBITO SOCIAL

Este proyecto no es perjudicial para ningún segmento de la población. Sin embargo, es cierto que, en el caso de la tecnología usada, puesto que es innovadora, hay algunos grupos sociales que no tienen acceso y/o no tienen una formación suficiente para poder usarla. Este factor puede tener un incidente directo en aquellos que se ven afectados por la brecha digital.

Este proyecto tiene la necesidad de llevarse a cabo para el uso del mismo a un nivel personal o profesional.

A nivel personal, este proyecto me ha aportado, principalmente, una gran formación en esta nueva tecnología innovadora, con una previsión de crecimiento a futuro, que es el machine learning. Además, aprender a realizar proyectos a nivel profesional teniendo en cuenta diferentes procesos que se llevan a cabo y la comunicación que se debe tener.

### 12.3 ÁMBITO ECONÓMICO

Un factor que no depende de nosotros, pero que puede tener un impacto negativo en la economía del proyecto, es el cambio de política de los diferentes servicios externos utilizados por el sistema.

En el momento que se decida establecer un sistema de pago para su uso, causaría un nuevo concepto de gasto no contemplado en el presupuesto.

Como ya se ha mencionado, este proyecto quiere evitar la dependencia de terceros para garantizar la propiedad del software. Para los servicios de estas organizaciones, generalmente se paga por un alto costo forzado. Las bases de esta propuesta es garantizar la propiedad con un pequeño costo inicial por el usufructo de la aplicación por parte del usuario de forma permanente.

En esta documentación se ha analizado la gestión económica del proyecto, teniendo en cuenta recursos humanos, materiales necesarios y otros tipos de gastos para la realización del mismo. Como se puede observar el coste final del proyecto se asemeja a la predicción previa a la realización del mismo. El presupuesto final se ha reducido gracias a una disminución del cómputo de algunas horas de trabajo y principalmente del uso del servidor, que no ha sido necesario durante el desarrollo del proyecto.

## 12.4 AUTOEVALUACIÓN

Tras completar este proyecto, me he dado cuenta de que mi comprensión de los conceptos de sostenibilidad en proyectos de desarrollo son insuficientes. Aunque son conceptos familiares en un nivel general, he descubierto que carezco del conocimiento necesario para aplicarlos efectivamente en mi vida cotidiana y en proyectos específicos, como este.

Para evaluar el progreso del proyecto, incluí preguntas sobre sostenibilidad y proyectos relacionados. Sin embargo, al enfrentar estas preguntas, me di cuenta de que carezco del conocimiento y la comprensión necesarios para abordar adecuadamente estos temas.

Por lo tanto, es esencial realizar un ejercicio de reflexión sobre cómo se desarrollan los procesos en este proyecto y otros, y cómo podemos utilizar diversas métricas para analizar estos factores y mejorar. Este proceso de reflexión nos permitirá identificar áreas de mejora y desarrollar estrategias para integrar los principios de sostenibilidad de manera más efectiva en nuestras actividades diarias y proyectos futuros.

## 13. CONCLUSIONES

La meta final marcada al principio del proyecto ha sido alcanzada satisfactoriamente, habiendo realizado la totalidad de las tareas definidas en la sección 4 con la funcionalidad de la prueba visual añadida más adelante. Durante el Trabajo de Fin de Grado se ha desarrollado un sistema completo, a partir de una especificación de requisitos (sección 9) y de un posterior diseño (explicado en la sección 10), una aplicación frontend y backend con Python, utilizando diversas librerías y herramientas como Docker y Elastic Stack (<https://github.com/Franmartin09/TFG-FrontBackVisual>). Todas estas partes del sistema, que se comunican entre ellas, han sido verificadas (sección 10), y una vez validado su comportamiento se ha dado por cerrado el desarrollo. En el mismo repositorio, en el directorio de preproceso está el Notebook que se ha usado para realizar el preprocesamiento de los datos.

### 13.1 ANÁLISIS DE ALTERNATIVAS Y MEJORAS

Durante la ejecución del proyecto se han tenido que tomar decisiones y/o cambios en la forma de realizar algunas tareas de diversos puntos. El primero de los casos ha sido la decisión de crear un backend para nuestra aplicación y no utilizar únicamente el notebook de google colab como parte práctica. Era un tema que había estado revisando con mi supervisor y director de proyecto porque hasta la hora de ponernos no teníamos claro si el tiempo y la necesidad de tenerlo presente sería realmente crucial. Al final, una vez empezado el desarrollo hemos llegado a la conclusión de que sería necesario, principalmente, para la validación correcta del mejor modelo extraído del notebook y la validación del funcionamiento de la recolecta y preprocesado de los datos. Nuestro interés era recrear un análisis exhaustivo de diferentes modelos de inteligencia artificial y entender el procesamiento que hay para este NIDS.

Una vez visto que se tenía que tener un backend propio para visualizar correctamente el funcionamiento, se debía decidir en qué lenguaje se desarrollaba. Tenemos dos alternativas: desarrollarlo con lenguajes de programación web y hospedar el modelo en el propio servidor utilizando Javascript; o bien, dado que se trata de un proyecto innovador que utiliza tecnologías con las que aún no había trabajado, como son tecnologías de machine learning, utilizar una tecnología que hubiera trabajado previamente y que contenga una documentación completa.

Por decisión propia y con la ambición de conocer nuevas tecnologías, decidí la segunda opción. Hice una investigación de los lenguajes más utilizados por backends para machine learning y vi que los dos más utilizados eran Javascript y Python. En mi caso, solo había trabajado con este último lenguaje por scripts y por algún entrenamiento de modelos de inteligencia artificial, pero nada exhaustivo, por tanto, era una tecnología ya utilizada por mí y con una gran documentación en base a inteligencia artificial, así que encajaba con mi interés.

La vinculación y uso de GitHub ha hecho tomar varias decisiones y cambios en la forma de realizar los procedimientos. El primer caso trata de cómo gestionar la obtención de recursos de la cuenta de GitHub del usuario.

Este paso sucede a la hora de crear un nuevo repositorio, ya que existen dos formas de colgar un código fuente. La primera es arrastrando o pulsando para abrir el explorador de archivos y adjuntando archivos, o bien vinculando el perfil con la cuenta de Github y obtener los archivos de los repositorios mediante SSH. Para poder tener la experiencia completa, se debe utilizar SSH para hacer clonación y subidas de forma segura y con la misma estructura del proyecto.

El proceso es el siguiente:

1. Creamos una key SSH dentro de GitHub en el apartado de desarrolladores. Con ello, nos proporcionan una clave secreta que funciona como identificador único del usuario.
2. Mediante comandos, declarando la clave ssh en el ordenador host, el usuario puede clonar el repositorio.
3. Una vez obtenida la clonación del repositorio el usuario puede proceder con la ejecución del mismo siguiendo las instrucciones que aparecen en el archivo ReadMe del repositorio.

El desarrollo del proyecto queda finalizado en este punto, ya que no hay previsión de darle una continuidad ni acabar colgando en un servidor en un entorno de producción. Sin embargo, en caso de continuidad debería mejorar el aspecto visual y adaptarse a dispositivos móviles y tabletas, así como a diferentes sistemas operativos. Actualmente, se ha diseñado solo para ordenadores, ya que es una aplicación de escritorio de ordenador que une servicios lanzados en Docker.

Por otra parte, sería interesante la posibilidad de mejorar las pipelines que hacen que se conecte los diferentes servicios de Docker con la aplicación visual y utilizar únicamente un entorno, es decir unificar todo a contenedores o unificar todo en una aplicación. Un último punto a considerar por una posible continuidad es la mejora de los modelos, entrenándolos en un entorno computacional mejor y con diferentes datos. Actualmente sólo se permite el entrenamiento bajo unos pocos protocolos, lo que mejoraría drásticamente el rendimiento y la eficacia del modelo.

## 13.2 CAMBIOS Y DESVIACIONES

Tomando como referencia la planificación establecida durante la fase inicial de la gestión del proyecto (sección 4), examinaremos los cambios que se han producido en ella. Durante el primer período de trabajo, que incluyó la formación en tecnologías de machine learning y análisis de datos, se cumplió con el cronograma previsto. Asimismo, todas las entregas relacionadas las tareas explícitas en la sección se realizaron dentro de los plazos establecidos.

Durante la redacción de estos documentos, se tenía previsto llevar a cabo la especificación de requisitos del proyecto, el diseño de la aplicación, la preparación del entorno y el inicio del desarrollo de los modelos de IA. Esta fase también se completó a tiempo. Sin embargo, durante el proceso de desarrollo posterior del proyecto, se realizaron ajustes en los modelos de IA para satisfacer necesidades específicas que no se habían identificado previamente. Estos cambios y mejoras no afectaron directamente la planificación, ya que fueron necesidades muy claras y concretas que ya se habían contemplado en el tiempo dedicado a las tareas subsiguientes.

Después de finalizar los modelos de IA, se procedió con la construcción de la aplicación. El desarrollo visual se realizó sin retrasos. A partir de este punto, se inició la gestión de la recuperación de información de los archivos de registro de Zee, lo cual planteó decisiones que antes estaban en duda. Finalmente, se decidió integrar un backend propio para la aplicación, permitiendo una gestión eficiente de la información y asegurando la privacidad de los datos, cumpliendo con los reglamentos pertinentes.

Se asignó una semana adicional para la preparación del entorno y la formación en el framework seleccionado para este backend.

Este ajuste afectó la planificación, pero dado que ya había generado algunas dudas sobre su implementación, se consideró parte de este tiempo dedicado en diversas tareas, especialmente en las vinculadas a la gestión de usuarios.

En este punto, justo después de completar la estructura del backend de la aplicación y finalizar la gestión del entorno, se redujeron considerablemente las horas dedicadas al proyecto para priorizar otras tareas urgentes en la empresa, incluidos proyectos que debían entregarse a los clientes.

Este contratiempo provocó un retraso de dos semanas en el proyecto. No se pudo compensar las horas dedicadas en ese momento debido a la implicación directa en un nuevo proyecto de cliente que requería comenzar el desarrollo de inmediato. Por lo tanto, se mantuvo la dedicación prevista a media jornada y se continuó con la funcionalidad del desarrollo de NIDS. Después se ajustó el orden de las demás tareas, para prever la correcta finalización del trabajo.

Finalizado el proceso de desarrollo, se llevaron a cabo todas las pruebas, tanto unitarias como de integración. Sin embargo, esta última tarea se realizó fuera del plazo inicialmente establecido, correspondiente a la semana siguiente a la finalización prevista del desarrollo del proyecto (sin incluir la documentación).

### 13.3 IMPACTO ECONÓMICO

A pesar de los inconvenientes y problemas surgidos durante el desarrollo del proyecto, el número de horas que se han dedicado al proyecto son las mismas que se ven reflejadas en la gestión inicial. Hay que decir que ciertas tareas se han conseguido desarrollar con menos tiempo de lo previsto, consiguiendo así compensar los imprevistos y la incorporación de las nuevas tareas no previstas.

Por el contrario, ha habido una modificación en la planificación inicial. Como se ha comentado anteriormente, motivada por la necesidad de dar respuesta, con carácter urgente, a otros proyectos y tareas dentro de la empresa, que provocó detener durante dos semanas el desarrollo del proyecto.

Por este motivo, aunque tenemos prevista como fecha de finalización del desarrollo el día 21 de junio a partir del diagrama de Gantt (sección 5.1), se ha finalizado realmente el día 23, es decir, dos días más tarde. Esta diferencia temporal se debe al parón previo que hubo.

El equipamiento software ha sido ligeramente modificado, y este cambio afecta a la planificación económica inicial. Se ha eliminado un servicio software que tenía un coste económico. Éste es el servidor de pruebas, que tenía un coste total de 234,24€ durante todo el periodo del proyecto.

Este coste se ha conseguido eliminar, ya que al final no ha sido necesaria el alquiler del servicio de un servidor porque todo el código se ha realizado en un entorno local de desarrollo, y esta aplicación no tendrá una vida útil una vez finalizado el proyecto.

Teniendo en cuenta que el presupuesto software inicial ascendía a un total de 372,00 €, se ha logrado reducir a 137,76 €.

Continuamos con los gastos generales. Este apartado incluye los materiales complementarios necesarios como coste de los servicios de electricidad, Internet, etc.

Teniendo en cuenta la situación actual del incremento del precio de la electricidad y de la gasolina, consideraremos que el gasto final tanto del desplazamiento como de la electricidad ha aumentado en 5€ cada una de ellas, causando un incremento de los costes generales de 20€ .

	<b>Coste inicial</b>	<b>Coste final</b>
Coste de las tareas	16.718,57€	16.718,57€
Coste de gastos generales	779,63€	545,39€
Contingencias	2.624,73€	2.624,73€
Imprevistos	610,75€	610,75€
<b>TOTAL</b>	<b>20.733,68€</b>	<b>20.499,44€</b>

Tabla 49: Tabla de comparación de costes iniciales y finales. Fuente: Elaboración propia.

Podemos concluir que el proyecto ha experimentado un descenso de costos de 234,24€.

## 14. INTEGRACIÓN DE CONOCIMIENTOS

En este proyecto se están haciendo uso de muchos conocimientos, muchos aprendidos durante el grado, sobre todo en la especialidad, y otros más concretos como pueden ser las tecnologías que se están aprendiendo durante el propio desarrollo del proyecto. A continuación, paso a detallar los conocimientos aplicados adquiridos en las asignaturas del grado.

Este proyecto tiene como base la creación de una sistema NIDS-AI, desarrollada a través de la programación y la programación basada en objetos para el backend. Por lo tanto, se ha aplicado conocimiento de las asignaturas de Programación 1 y Estructura de la Información, donde se aprenden las bases de la programación y orientación de objetos, siguiendo con aspectos de la propia programación.

Hay otras asignaturas fuera de la especialidad que también han ayudado a desarrollar este proyecto. Un ejemplo de esto es “Interacción y Diseño de Interfaces”, donde aprendí muchos aspectos del diseño de interfaces y cómo el usuario interactúa. Esto ha ayudado a realizar un mejor diseño de la aplicación y evitar acciones no deseadas por parte del usuario.

La asignatura de Empresa ha sido también muy útil a la hora de realizar la gestión del proyecto inicial para poder calcular todos los costes que implican el proyecto. En esta asignatura se explican aspectos como los costes de personal, retenciones de IRPF y de Seguridad Social que deben tenerse en cuenta a la hora de calcular los gastos en recursos humanos y de cómo gestionar los diversos recursos necesarios para el buen funcionamiento de una actividad. También se trataban aspectos utilizados en los cálculos como las amortizaciones.

“Proyectos de Tecnologías de la Información” y “Administración de Sistemas Operativos” fueron las primeras asignaturas en las que realizamos un proyecto real, donde se tenían que especificar requisitos y organizar las diversas tareas del proyecto. Estas asignaturas me han ofrecido las bases de especificación de requisitos, gestión de proyectos y arquitectura del software a aplicar en cualquier proyecto. Han sido muy clave a la hora de saber cómo debía gestionar el proyecto de la mejor forma posible, y garantizar la finalización del mismo de manera satisfactoria. Conceptos clave como Agile y Scrum, han ayudado a que este proyecto salga adelante.

Con las asignaturas de “Redes de Computadores” e “Internet” aprendí en profundidad sobre los protocolos y las transmisiones de datos. Estos conocimientos han sido esenciales para entender cómo los datos se mueven a través de las redes, lo cual ha sido fundamental para desarrollar nuestro sistema de detección de intrusos basado en inteligencia artificial, ya que nos permite detectar y analizar las anomalías en el tráfico de red de manera eficiente. Además nos permite entender cómo funciona la transmisión de datos bajo los protocolos que se han llevado a cabo. Estas materias han sido vitales para comprender las complejidades de las redes de computadoras y optimizar su rendimiento, algo crucial para la implementación efectiva de un sistema de detección de intrusos.

Seguidamente está una asignatura similar en la que también se profundiza en redes y en el futuro tecnológico. “Future Internet” ha sido otra asignatura clave. Aquí, se aborda la ciberseguridad y el futuro de la tecnología. Los conceptos aprendidos sobre hacia dónde va el mundo tecnológico y la seguridad actual y futura, nos ha permitido entender el funcionamiento que debe tener un NIDS utilizando Inteligencia Artificial.

En Matemática Discreta se trabajan datos y matrices. La base matemática proporcionada ha sido indispensable para la modelización y resolución de problemas complejos en el desarrollo del sistema de detección de intrusos con inteligencia artificial, permitiendo un análisis riguroso y preciso de los datos de la red y de los modelos de inteligencia artificial.

A continuación, me gustaría compartir cómo las asignaturas de mi especialidad me han ayudado a realizar este trabajo de la mejor manera posible.

La asignatura de Minería de Datos se centra en el Machine Learning. Las técnicas aprendidas en esta materia han sido aplicadas para analizar grandes volúmenes de datos y mejorar la precisión del sistema de detección de intrusos con IA. La capacidad de aprender y adaptarse a nuevas amenazas es una característica clave de este proyecto.

En Sistemes Operatius Distribuïts i en Xarxa, a pesar que no es específica pero es muy esencial, tratamos la ciberseguridad y los entornos de sistemas operativos en red. Los conocimientos adquiridos sobre la seguridad en sistemas distribuidos han sido esenciales para asegurar que nuestro sistema de detección de intrusos funcione de manera segura y eficiente en diversos entornos de red.

Dejando a un lado el orden ascendente comentado anteriormente, hay una última asignatura que aunque no forma parte de la especialidad y que, en muchos casos, incluso se podría considerar que no tiene mucha relación con un proyecto como éste, es la asignatura de "Técnicas de Escritura para Ingeniería", impartida en Inglés. Cabe señalar que se trata de una asignatura que me está ayudando en la redacción de este documento, ya que aunque se imparte en inglés, se explican muchos conceptos de cómo redactar un documento técnico correctamente, que se pueden aplicar a cualquier idioma.

## 14.1 COMPETENCIAS TÉCNICAS

Durante el transcurso de este proyecto, he aplicado una serie de competencias técnicas adquiridas durante mi especialidad en Ingeniería Informática. A continuación, detallaré cómo estas competencias se han integrado en el desarrollo de nuestro sistema NIDS-AI, destacando su relevancia y la manera en que han facilitado la culminación exitosa de este trabajo.

*CE1.1: Definir y gestionar los requisitos de un sistema software.*

La especificación y gestión de requisitos ha sido una parte fundamental de este proyecto. Desde la identificación de requisitos iniciales hasta la documentación detallada en formato UML, cada requisito ha sido cuidadosamente considerado para asegurar su correcta implementación y verificación.

*CE1.2: Identificar, evaluar y gestionar los riesgos potenciales asociados a la construcción de software.*

Durante la fase inicial del proyecto, se identificaron y documentaron los posibles riesgos, evaluando su probabilidad y severidad, y desarrollando estrategias de mitigación. Estos riesgos se han gestionado a lo largo del desarrollo, adaptando la planificación cuando ha sido necesario para asegurar la finalización exitosa del proyecto.

*CE1.3: Especificar, diseñar, implementar y evaluar NIDS.*

En este proyecto se ha seguido un enfoque riguroso para la especificación diseño implementación y evaluación de un Sistema de Detección de Intrusos en la Red (NIDS). Se realizó un análisis exhaustivo de los requisitos del sistema identificando las características clave que el NIDS debía cumplir como la detección de amenazas en tiempo real, la capacidad de manejar grandes volúmenes de tráfico de red y la flexibilidad para adaptarse a diferentes entornos de red. El diseño del NIDS se estructuró en varios módulos principales: captura de tráfico, análisis de paquetes, detección de anomalías, almacenamiento y reportes. La implementación se llevó a cabo utilizando lenguajes y herramientas adecuados para cada módulo. Se aseguraron la modularidad y la escalabilidad del sistema, se diseñaron y ejecutaron pruebas exhaustivas para evaluar la efectividad del NIDS asegurando que el sistema cumpla con los requisitos especificados y esté preparado para enfrentar las amenazas de seguridad en redes modernas

*CE2.1: Desarrollar, mantener y evaluar sistemas y servicios software complejos y/o críticos.*

En este proyecto, hemos desarrollado un sistema de software complejo que incluye varios servicios interconectados, cada uno construido con tecnologías innovadoras. Estos servicios abarcan desde la configuración de Zeek y su integración con Docker, hasta la implementación de un sistema de detección de intrusos basado en inteligencia artificial, pasando por un entorno conectado para la visualización de datos. La validación del sistema se ha realizado mediante diversas herramientas de testing, incluyendo pruebas unitarias automatizadas y pruebas de integración, asegurando así su robustez y fiabilidad.

*CE2.2: Dar solución a problemas de integración en función de las estrategias, de los estándares y de las tecnologías disponibles.*

Para este proyecto, hemos tenido que solucionar múltiples problemas de integración, desde la comunicación entre diferentes servicios hasta la implementación de pipelines de datos. Por ejemplo, se ha desarrollado un programa orientado a objetos en Python que interactúa con Zeek y con el modelo de inteligencia artificial, integrando todo mediante un flujo de datos eficiente y seguro. La elección de estrategias y estándares adecuados ha sido crucial para superar los desafíos técnicos.

*CE2.3: Desarrollar, mantener y evaluar servicios y aplicaciones distribuidas con soporte de red.*

Nuestro sistema NIDS-AI se basa en tecnologías de red y aplicaciones distribuidas. La integración de servicios como Zeek, Kibana, Elasticsearch, y Filebeat en un entorno Dockerizado ha sido esencial para la implementación efectiva del sistema. Estos servicios distribuidos permiten el monitoreo y análisis en tiempo real del tráfico de red, garantizando un alto nivel de seguridad y rendimiento.

*CE2.4: Controlar la calidad y diseñar pruebas en la producción de software.*

Hemos realizado extensas pruebas unitarias y de integración para garantizar la calidad del software. Además, se han llevado a cabo pruebas con usuarios reales para obtener feedback y asegurar que el sistema cumpla con todos los requisitos funcionales y no funcionales.

## 14.2 CONCLUSIONES PERSONALES

Este proyecto ha sido muy gratificante a nivel personal y profesional. A nivel personal, me ha ofrecido la oportunidad de trabajar de primera mano con tecnologías desconocidas, motivándome a no parar de aprender y saciando mi curiosidad. Ver que he podido asumir este reto ha sido muy satisfactorio, comprobando que he podido superarlo.

Siguiendo mi motivación de aprender nuevas tecnologías, he tenido que hacer frente a un gran reto al trabajar con tecnologías que no conocía a nivel técnico, lo que ha requerido un esfuerzo significativo de investigación y formación. Esta decisión también implicaba el riesgo de quedarme estancado si no lograba dominar la nueva tecnología.

Desarrollar el proyecto desde el inicio hasta el testing, realizando todas las fases iniciales de gestión de proyecto, especificación de requisitos, diseño del sistema y desarrollo, me ha permitido aplicar todos los conocimientos adquiridos durante el grado y especialmente durante la especialidad. Además, ver cómo todos los conocimientos aprendidos se pueden plasmar en un proyecto concreto, encajando de manera correcta, me ha demostrado que una idea sencilla y sin organización puede convertirse en un gran proyecto bien planificado y con ambición

## 15. BIBLIOGRAFÍA

- [1] NIDS, en Wikipedia. ¿Qué es un NIDS? Enlace: <https://es.wikipedia.org/wiki/NIDS>.
- [2] Aprendizaje automático (ML), en Microsoft, ¿Qué es el aprendizaje automático? Enlace: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-machine-learning-platform>
- [3] Inteligencia artificial (IA), en Wikipedia. ¿Qué es la inteligencia artificial? Enlace: [https://es.wikipedia.org/wiki/Inteligencia\\_artificial](https://es.wikipedia.org/wiki/Inteligencia_artificial)
- [4] PYMES, en Wikipedia, ¿Qué son las PYMES? Enlace: [https://es.wikipedia.org/wiki/Peque%C3%A1\\_a\\_y\\_media\\_empresa](https://es.wikipedia.org/wiki/Peque%C3%A1_a_y_media_empresa)
- [5] Scrum, en Blog, Metodología Scrum, Roles, Procesos y Artefactos. Enlace: <https://blog.innevo.com/metodologia-scrum>
- [6] Sprint, en Atlassian, ¿Qué son los sprints en la gestión de proyectos? Enlace: <https://www.atlassian.com/es/agile/scrum/sprints>
- [7] Control de versiones Git, en Atlassian, Gitflow workflow Enlace: <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>
- [8] Linter, en Wikipedia, Lint. Enlace: <https://es.wikipedia.org/wiki/Lint>
- [9] Ciberseguridad, en IBM, ¿Qué es la ciberseguridad? Enlace: <https://www.ibm.com/es-es/topics/cybersecurity>
- [10] OWASP, en OWASP, Listado de 10 riesgos en aplicaciones. Enlace: <https://owasp.org/www-project-top-ten/>
- [11] Firewall, en Wikipedia, ¿Qué es un Firewall? Enlace: [https://es.wikipedia.org/wiki/Cortafuegos\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Cortafuegos_(inform%C3%A1tica))
- [12] IPS, en Wikipedia, Sistema de prevención de intrusos. Enlace: [https://es.wikipedia.org/wiki/Sistema\\_de\\_prevenci%C3%B3n\\_de\\_intrusos](https://es.wikipedia.org/wiki/Sistema_de_prevenci%C3%B3n_de_intrusos)
- [13] Glassdoor, en Glassdoor, Búsqueda de sueldos. Enlace: <https://www.glassdoor.es/Sueldos/index.htm>
- [14] Real Decreto 1777/2004, BOE, Amortización de materiales. Enlace: <https://www.boe.es/buscar/act.php?id=BOE-A-2004-14600>
- [15] IDS, en Clavei, ¿Qué es un IDS? Enlace: <https://www.clavei.es/blog/que-es-un-ids-o-intrusion-detection-system/>
- [16] HIDS vs NIDS, en GeekForGeek, Comparación entre HIDS y NIDS. Enlace: <https://www.geeksforgeeks.org/difference-between-hids-and-nids/>
- [17] Anomalías y firmas, en AbcExperts. Detección de anomalías y firmas. Enlace: <https://abcxperts.com/que-es-un-sistema-de-deteccion-de-intrusiones-ids/>

[18] Aplicación Inteligencia Artificial, en Wikipedia, Inteligencia artificial. Enlace: [https://en.wikipedia.org/wiki/Artificial\\_intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence)

[19] Algoritmos de machine learning, en Medium, Tipos de Aprendizaje Automático. Enlace: <https://medium.com/soldai/tipos-de-aprendizaje-autom%C3%A1tico-6413e3c615e2LG>

[20] Clustering, en Bookdown, K-means Clustering . Enlace: [https://bookdown.org/tpinto\\_home/Unsupervised-learning/k-means-clustering.html](https://bookdown.org/tpinto_home/Unsupervised-learning/k-means-clustering.html)

[21] Reducción de la dimensionalidad, en RPbus, Ejemplo de Reducción de Dimensionalidad con la Base de Datos Iris. Enlace: <https://rpubs.com/dsolis/iris-pca-lda>

[22] Detección de anomalías, en GeeksForGeeks. Detección de anomalias en series de datos temporales Enlace: <https://www.geeksforgeeks.org/anomaly-detection-in-time-series-data/>

[23] Naive Bayes, en IBM, Naive Bayes. Enlace: <https://www.ibm.com/es-es/topics/naive-bayes#:~:text=El%20clasificador%20Naive%20Bayes%20es,para%20realizar%20tareas%20de%20clasificaci%C3%B3n>

[24] Decision Tree, en IBM, ¿Qué es árbol de decisión? Enlace: <https://www.ibm.com/es-es/topics/decision-trees>

[25] Random Forest, en IBM, ¿Qué son árboles aleatorios? Enlace: <https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly.both%20classification%20and%20regression%20problems>

[26] GBM, XGBoost, LightGBM y CatBoost, en Ciencia de datos, Series Temporales. Enlace: <https://cienciadedatos.net/documentos/py39-forecasting-series-temporales-con-skforecast-xgboost-lightgbm-catboost>

[27] Redes neuronales artificiales (ANN), en IBM, Neural Network. Enlace: <https://www.ibm.com/es-es/topics/neural-network>

[28] Redes neuronales recurrentes (RNN), en IBM, Recurrent Neural Network. Enlace: <https://www.ibm.com/es-es/topics/recurrent-neural-networks>

[29] Bagging / Boosting, en Machine learning para todos, Diferencia entre Bagging y Boosting. Enlace: <https://machinelearningparatodos.com/cual-es-la-diferencia-entre-los-metodos-de-bagging-y-los-de-boosting/>

[30] Métricas de Rendimiento, en Data Source, Métricas de Evaluación de Modelos. Enlace: <https://datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico>

[31] UNSW-NB15, en UNSW Sydney, Datasets UNSW-NB15. Enlace: <https://research.unsw.edu.au/projects/unsw-nb15-dataset>

[32] Preprocesamiento de datos, en Klippa, ¿Qué es la preparación de los datos? Enlace: <https://www.klippa.com/es/blog/informativo/que-es-preparacion-datos/>

[33] Transformación de datos, en Medium, Transformación de datos. Enlace: <https://nicolasurrego.medium.com/transformando-datos-en-oro-c%C3%B3mo-la-estandarizaci%C3%B3n-y-normalizaci%C3%B3n-mejoran-tus-resultados-fbe0840d2b94>

[34] Encoders, en Interactive Chaos, Codificación de datos categóricos. Enlace: <https://interactivechaos.com/es/manual/tutorial-de-machine-learning/codificacion-de-caracteristicas>

[35] Desbalance de clases, en Neptune.ai, Overfitting / Underfitting. Enlace: <https://medium.com/analytics-vidhya/undersampling-and-oversampling-an-old-and-a-new-approach-4f984a0e8392>

[36] SMOTE y ADASYN, en RPbus, Desbalance de clases avanzado. Enlace: <https://rpubs.com/argamrp/imbalance>

[37] Selección de características, en Aprendeia, Metodos de seleccion de características. Enlace: <https://aprendeia.com/metodos-de-seleccion-de-caracteristicas-machine-learning/>

[38] Hold-out Validation, en , Validación de pruebas. Enlace: <https://www.comet.com/site/blog/understanding-hold-out-methods-for-training-machine-learning-models/>

[39] Cross Validation, en Sharp Sight Labs , Validación cruzada. Enlace: <https://www.sharp-sight-labs.com/blog/cross-validation-explained/>

[40] División estratificada, en Question Pro, Muestreo estratificado. Enlace: <https://www.questionpro.com/blog/es/muestreo-estratificado/>

[41] Zeek, en Zeek, ¿Qué es Zeek y cuales son sus funcionalidades?. Enlace: <https://zeek.org/>

[42] Elastic Stack, en Elastic, ¿Qué es Elastic Stack y qué componentes tiene?. Enlace: <https://www.elastic.co/es/elastic-stack>

[43] Label, en Developers Google , Etiquetas. Enlace: <https://developers.google.com/machine-learning/crash-course/framing/ml-terminology?hl=es-419>

[44] Alpine Linux, en Wikipedia, Alpine Linux . Enlace: [https://es.wikipedia.org/wiki/Alpine\\_Linux](https://es.wikipedia.org/wiki/Alpine_Linux)

[45] GeolIP, en Maxmind, GeoLite2 Free Geolocation Data. Enlace: <https://dev.maxmind.com/geoip/geolite2-free-geolocation-data>

[46] Local.zeek, en Medium, Zeek Scripts. Enlace: <https://medium.com/@haircutfish/tryhackme-zeek-task-7-zeek-scripts-scripts-and-signatures-task-8-zeek-scripts-frameworks-1bbab9f9be74>

[47] Tkinter, en Documentación Python, Documentación librería Tkinter. Enlace: <https://docs.python.org/es/3/library/tkinter.html>

[48] Índices y Mapping, en KeepCoding, ¿Para qué sirven los Indexes y Maps en ElasticSearch? Enlace: <https://keepcoding.io/blog/indexes-maps-elasticsearch/>

[49] Normativa TFE UPC, en Drive, “Normativa acadèmica dels estudis de Grau i Màster de l’Escola Politècnica Superior d’Enginyeria de Vilanova i la Geltrú”. Enlace: <https://drive.google.com/file/d/13T3SvlkXqm3HugovlIDKerb4hS5iXb5/view>

[50] RGPD, en BOE, Reglamento general de protección de datos. Enlace: <https://www.boe.es/DOUE/2016/119/L00001-00088.pdf>

[51] Terminos y condiciones Google Colab, en Cloud Google, Términos de Servicio de Google Cloud Platform . Enlace: <https://cloud.google.com/terms?hl=es-419>

[52] Terminos y condiciones Git, en Github Docs, Términos de Servicio de GitHub . Enlace: <https://docs.github.com/es/site-policy/github-terms/github-terms-of-service>

## 16. ÍNDICE DE FIGURAS

Figura 1: Metodología Agile Scrum. Fuente: Innevo.	12
Figura 2: Sistema de control de versiones distribuido. Fuente: GitFlow.	14
Figura 3: Diagrama de Gantt TFG - NIDS-A. Fuente: Elaboración Propia.	22
Figura 4: Esquema de componentes IDS. Fuente: Clavei.	31
Figura 5: Fases de un NIDS. Fuente: Elaboración propia	35
Figura 6: Infraestructura NIDS. Fuente: TeamOK	36
Figura 7: Modelo de aprendizaje no supervisado. Fuente: Medium	37
Figura 8: Clustering (Agrupamiento). Fuente: BookDown	38
Figura 9: Reducción de dimensionalidad. Fuente: RPubs	38
Figura 10: Detección de anomalías en datos. Fuente: GeeksForGeeks	39
Figura 11: Modelo de aprendizaje supervisado. Fuente: Medium	40
Figura 12: Modelo Naive Bayes. Fuente: Medium	41
Figura 13: Ejemplo de Árbol de Decisión, Supervivientes del Titanic. Fuente: Wikipedia	43
Figura 14: Random Forest. Fuente: Medium	44
Figura 15: Modelo Gradient Boosting Machine. Fuente: Medium	45
Figura 16: Crecimiento Nivelado. Fuente: Medium	46
Figura 17: Crecimiento Foliar. Fuente: Medium	46
Figura 18: Crecimiento de árbol en CatBoost. Fuente: HandsOnCloud	47
Figura 19: Red Neuronal Artificial. Fuente: OpenWebinars	49
Figura 20: Red Neuronal Recurrente. Fuente: OpenWebinars	51
Figura 21: Modelo de aprendizaje reforzado. Fuente: Medium	51
Figura 22: Ejemplo votación por mayoría. Fuente: NSOSSR	53
Figura 23: Bagging. Fuente: Machine learning para todos	54
Figura 24: Boosting. Fuente: Machine learning para todos	55
Figura 25: Matriz de Confusión. Fuente: Aprendeia	57
Figura 26: Curva de Precision-Recall. Fuente: Elaboración propia.	60
Figura 27: Curva ROC. Fuente: Elaboración propia.	61
Figura 28: Informe de Clasificación (Classification Report). Fuente: Elaboración propia.	63
Figura 29: Escalado Estándar vs Escalado Robusto. Fuente: Medium.	69
Figura 30: Submuestreo (UnderSampling). Fuente: Neptune.ai	71
Figura 31: Sobremuestreo (OverSampling). Fuente: Neptune.ai	72
Figura 32: Gráfico de equilibrio de clases SMOTE y ADASYN. Fuente: RPubs	72
Figura 33: Validación Hold-Out. Fuente: Comet.	75
Figura 34: Validación Hold-Out. Fuente: Comet. [38]	75
Figura 35: Validación Cruzada K-Fold donde K=5. Fuente: Sharp Sight. [39]	76
Figura 36: Validación Estratificada. Fuente: QuestionPro. [40]	76
Figura 37: Esquema Elastic Stack. Fuente: Medium.	79
Figura 38: Diagrama de casos de uso del NIDS-AI. Fuente: Elaboración propia	80
Figura 39: Datos vacíos e información de los Datasets. Fuente: Elaboración propia	92
Figura 40: Comprobación de duplicados en los Datasets. Fuente: Elaboración propia	93
Figura 41: Comprobación de duplicados entre Datasets contiguos. Fuente: Elaboración propia	94
Figura 42: Inconsistencias en tipo de la columna src_bytes. Fuente: Elaboración propia	95
Figura 43: Distribución en cada dataset del campo label. Fuente: Elaboración propia	98
Figura 44: Distribución del conjunto de datasets del campo label. Fuente: Elaboración propia	98
Figura 45: Distribución de los tipos de categoría en cada dataset. Fuente: Elaboración propia	101
Figura 46: Distribución de los tipos de categoría en los datasets. Fuente: Elaboración propia	102
Figura 47: Distribución de los protocolos en los datasets. Fuente: Elaboración propia	103
Figura 48: Valores atípicos en src_byets, dst_bytes y duration. Fuente: Elaboración propia	109
Figura 49: Gráfica de categoría label en tiempo de los datasets. Fuente: Elaboración propia	109
Figura 50: Gráfica de duración en tiempo de los dataset. Fuente: Elaboración propia	110

Sistema de Detección de Intrusiones en Ciberseguridad con Inteligencia Artificial  
Francisco Jose Martin Aguilar

Figura 51: Ejemplo de datos del dataset 1 sin normalizar. Fuente: Elaboración propia	111
Figura 52: Ejemplo de datos del dataset 1 normalizado. Fuente: Elaboración propia	111
Figura 53: Dataset equilibrado 100K para label codificado. Fuente: Elaboración propia	116
Figura 54: Matriz de correlación del dataset equilibrado 100K para label codificado. Fuente: E. propia	117
Figura 55: Métricas de rendimiento usando K-Fold. Fuente: Elaboración propia	118
Figura 56: Métricas de rendimiento sin usar K-Fold. Fuente: Elaboración propia	118
Figura 57: Error de ejecución en entorno Google Colab. Fuente: Elaboración propia	120
Figura 58: Métricas de rendimiento para label. Fuente: Elaboración propia	127
Figura 59: Métricas de rendimiento para type. Fuente: Elaboración propia	128
Figura 60: Matriz de confusión para label con dataset grande equilibrado. Fuente: Elaboración propia	129
Figura 61: Matriz de confusión para label con dataset pequeño no equilibrado. Fuente: E. propia	129
Figura 62: Matriz de confusión para label con dataset pequeño equilibrado. Fuente: E. propia	130
Figura 63: Matriz de confusión para type con dataset grande equilibrado. Fuente: Elaboración propia	131
Figura 64: Matriz de confusión para type con dataset grande equilibrado. Fuente: Elaboración propia	132
Figura 65: Matriz de confusión para type con dataset pequeño no equilibrado. Fuente: E. propia	133
Figura 66: Flujo de trabajo del NIDS-AI. Fuente: Elaboración propia	135
Figura 67: Diseño en Figma de la app principal del NIDS-AI. Fuente: Elaboración propia	142
Figura 68: Diseño final de la app principal del NIDS-AI. Fuente: Elaboración propia	143
Figura 69: Diseño final de la app secundaria del NIDS-AI. Fuente: Elaboración propia	144
Figura 70: Diseño del dashboard de Kibana. Fuente: Elaboración propia	145

## 17. ÍNDICE DE TABLAS

Tabla 1: Resumen de tareas. Fuente: Elaboración propia.	21
Tabla 2: Recursos humanos del proyecte con sus costes. Fuente: Elaboración propia.	25
Tabla 3: Recursos humanos del proyecte con sus costes. Fuente: Elaboración propia.	26
Tabla 4: Tabla de costes del equipamiento de hardware necesario. Fuente: Elaboración propia.	27
Tabla 5: Tabla de costes del equipamiento de software necesario. Fuente: Elaboración propia.	28
Tabla 6: Tabla de costes de los costes generales del proyecto. Fuente: Elaboración propia.	29
Tabla 7: Tabla de costes de imprevistos. Fuente: Elaboración propia.	29
Tabla 8: Tabla de costes totales. Fuente: Elaboración propia.	30
Tabla 9: Características HIDS vs NIDS. Fuente: Elaboración propia	32
Tabla 10: Características de los tipos de aprendizaje. Fuente: Elaboración propia	52
Tabla 11: Características de los conjuntos UNSW-NB15. Fuente: Elaboración propia.	65
Tabla 12: UC01 - Ejecutar Aplicación. Fuente: Elaboración propia	82
Tabla 13: UC02 - Configurar NIDS con IA. Fuente: Elaboración propia	82
Tabla 14: UC03 - Configurar Zeek. Fuente: Elaboración propia	83
Tabla 15: UC04 - Recopilar Datos de Red. Fuente: Elaboración propia	83
Tabla 16: UC05 - Configurar Elasticsearch. Fuente: Elaboración propia	84
Tabla 17: UC06 - Configurar Filebeat. Fuente: Elaboración propia	84
Tabla 18: UC07 - Configurar Kibana. Fuente: Elaboración propia	84
Tabla 19: UC08 - Almacenar e Indexar los Datos. Fuente: Elaboración propia	85
Tabla 20: UC09 - Analizar los Datos con IA. Fuente: Elaboración propia	85
Tabla 21: UC10 - Detectar Posibles Intrusiones. Fuente: Elaboración propia	86
Tabla 22: UC11 - Generar Alertas de Seguridad. Fuente: Elaboración propia	86
Tabla 23: UC12 - Responder a Posibles Ataques. Fuente: Elaboración propia	86
Tabla 24: UC13 - Visualizar Datos. Fuente: Elaboración propia	87
Tabla 25: UC14 - Detener la Aplicación. Fuente: Elaboración propia	87
Tabla 26: RNF01 - Fàcil de utilizar. Fuente: Elaboración propia	88
Tabla 27: RNF02 - Estilo. Fuente: Elaboración propia	88
Tabla 28: RNF03 - Comprensión y cortesía. Fuente: Elaboración propia	88
Tabla 29: RNF04 - Precisión. Fuente: Elaboración propia	88
Tabla 30: RNF05 - Compatibilidad. Fuente: Elaboración propia	89
Tabla 31: RNF06 - Integridad. Fuente: Elaboración propia	89
Tabla 32: RNF10 - Privacidad. Fuente: Elaboración propia	89
Tabla 33: RNF10 - Legalidad. Fuente: Elaboración propia	89
Tabla 34: Descripción de los datasets. Fuente: Elaboración propia	91
Tabla 35: Detalle del conjunto de datasets por label. Fuente: Elaboración propia	98
Tabla 36: Detalles de los tipos de categoría en los datasets. Fuente: Elaboración propia	102
Tabla 37: Detalle de los protocolos en los datasets. Fuente: Elaboración propia	103
Tabla 38: Detalle de los servicios de las conexiones en los datasets. Fuente: Elaboración propia	104
Tabla 39: Detalle del dataset equilibrado 1,3M para label. Fuente: Elaboración propia	112
Tabla 40: Detalle del dataset equilibrado 100K para label. Fuente: Elaboración propia	113
Tabla 41: Detalle del dataset no equilibrado 1,3M para label / type. Fuente: Elaboración propia	113
Tabla 42: Detalle del dataset no equilibrado 100K para label / type. Fuente: Elaboración propia	114
Tabla 43: Detalle del dataset equilibrado 1,3M para type. Fuente: Elaboración propia	115
Tabla 44: Detalle del dataset equilibrado 100K para type. Fuente: Elaboración propia	115
Tabla 45: Métricas de rendimiento para label con datos equilibrados. Fuente: Elaboración propia	121
Tabla 46: Métricas de rendimiento para label con datos no equilibrados. Fuente: Elaboración propia	122
Tabla 47: Métricas de rendimiento para type con datos equilibrados. Fuente: Elaboración propia	124
Tabla 48: Métricas de rendimiento para type con datos no equilibrados. Fuente: Elaboración propia	125

## ANEXO 1 - LOCAL.ZEEK

---

```
# Installation-wide salt value that is used in some digest hashes, e.g., for
# the creation of file IDs. Please change this to a hard to guess value.
redef digest_salt = "franma";

# This script logs which scripts were loaded during each run.
@load misc/loaded-scripts

# Apply the default tuning scripts for common tuning settings.
@load tuning/defaults

# Estimate and log capture loss.
@load misc/capture-loss

# Enable logging of memory, packet and lag statistics.
@load misc/stats

# Load the scan detection script. It's disabled by default because
# it often causes performance issues.
@load misc/scan

# Detect traceroute being run on the network. This could possibly cause
# performance trouble when there are a lot of traceroutes on your network.
# Enable cautiously.
#@load misc/detect-traceroute

# Generate notices when vulnerable versions of software are discovered.
# The default is to only monitor software found in the address space defined
# as "local". Refer to the software framework's documentation for more
# information.
@load frameworks/software/vulnerable

# Detect software changing (e.g. attacker installing hacked SSHD).
@load frameworks/software/version-changes

# This adds signatures to detect cleartext forward and reverse windows shells.
@load-sigs frameworks/signatures/detect-windows-shells

# Load all of the scripts that detect software in various protocols.
@load protocols/ftp/software
@load protocols/smtp/software
@load protocols/ssh/software
@load protocols/http/software

# The detect-webapps script could possibly cause performance trouble when
# running on live traffic. Enable it cautiously.
@load protocols/http/detect-webapps

# This script detects DNS results pointing toward your Site::local_nets
# where the name is not part of your local DNS zone and is being hosted
# externally. Requires that the Site::local_zones variable is defined.
@load protocols/dns/detect-external-names

# Script to detect various activity in FTP sessions.
@load protocols/ftp/detect

# Scripts that do asset tracking.
@load protocols/conn/known-hosts
```

```
@load protocols/conn/known-services
@load protocols/ssl/known-certs

# This script enables SSL/TLS certificate validation.
@load protocols/ssl/validate-certs

# This script prevents the logging of SSL CA certificates in x509.log
@load protocols/ssl/log-hostcerts-only

# If you have GeoIP support built in, do some geographic detections and
# logging for SSH traffic.
@load protocols/ssh/geo-data
# Detect hosts doing SSH bruteforce attacks.
@load protocols/ssh/detect-bruteforcing
# Detect logins using "interesting" hostnames.
@load protocols/ssh/interesting-hostnames

# Detect SQL injection attacks.
@load protocols/http/detect-sqli

##### Network File Handling #####
# Enable MD5 and SHA1 hashing for all files.
@load frameworks/files/hash-all-files

# Detect SHA1 sums in Team Cymru's Malware Hash Registry.
@load frameworks/files/detect-MHR

# Extend email alerting to include hostnames
@load policy/frameworks/notice/extend-email/hostnames

# Uncomment the following line to enable detection of the heartbleed attack. Enabling
# this might impact performance a bit.
@load policy/protocols/ssl/heartbleed

# Uncomment the following line to enable logging of connection VLANs. Enabling
# this adds two VLAN fields to the conn.log file.
@load policy/protocols/conn/vlan-logging

# Uncomment the following line to enable logging of link-layer addresses. Enabling
# this adds the link-layer address for each connection endpoint to the conn.log file.
@load policy/protocols/conn/mac-logging

# Comment this to unload Corelight/CommunityID
@load Corelight/CommunityID

# Custom conn geoip enrichment
@load geodata/conn-add-geodata.zeek
# Log all plain-text http/ftp passwords
@load passwords/log-passwords.zeek

@load file-extraction

# JSON Plugin
# @load json-streaming-logs
# redef JSONStreaming::disable_default_logs=T;

# Redirigir todos los logs al mismo archivo final.log
redef LogAscii::use_json=T;
```

## **ANEXO 2 - SOLICITUD DE CÓDIGO ADICIONAL**

Para obtener fragmentos de código adicionales que no se incluyen en este documento, por favor solicítelos bajo demanda. Esto permite mantener el enfoque en los aspectos más relevantes del proyecto y facilita la revisión y comprensión del contenido principal.

Si necesita código específico o ejemplos adicionales para complementar la información aquí presentada, puede ponerse en contacto a través de los siguientes medios:

- Correo Electrónico: [franmartinaguilar@gmail.com](mailto:franmartinaguilar@gmail.com)

Por favor, incluya en su solicitud los detalles específicos del código que necesita.