

# **ANALITICA DE DATOS (SEGURIDAD INFORMATICA)**

**Analítica de datos**

**GABRIEL ENRIQUE BELTRAN IBARRA**

**Carlos Andres Osorno Jaramillo – INGENIERIA EN SISTEMAS – Ficha 24V06**

**Corporación unificada Nacional de educación superior**

**Medellín**

**2025**

## Contenido

<b>Introducción .....</b>	3
<b>Metodología Utilizada.....</b>	4
<b>Fase 1: Definición del Problema y Selección del Sector .....</b>	5
<b>Fase 2: Recolección y limpieza de datos. ....</b>	6
<b>Fase 3: Análisis Descriptivo y Diagnóstico. ....</b>	6
<b>Fase 4: Modelado predictivo.....</b>	8
<b>Fase 5: Propuesta de Soluciones y Toma de Decisiones.....</b>	10
<b>Propuestas y Conclusiones.....</b>	11

# Introducción

En un mundo donde la ciberseguridad es una prioridad creciente para empresas y organizaciones, la capacidad de predecir y mitigar incidentes de seguridad se ha convertido en una necesidad estratégica. Con el aumento de amenazas digitales y la evolución constante de los ataques cibernéticos, la detección temprana de eventos de riesgo es fundamental para minimizar impactos y fortalecer la protección de los sistemas de información.

Este informe presenta el desarrollo y evaluación de un modelo predictivo basado en aprendizaje automático, diseñado para clasificar eventos de seguridad informática según su nivel de riesgo. A partir de un conjunto de datos históricos que incluye información sobre tipos de eventos, direcciones IP de origen, usuarios afectados y estados de incidentes, se ha construido un modelo de clasificación utilizando un Árbol de Decisión.

El objetivo principal de este estudio es proporcionar una herramienta analítica que ayude a los equipos de seguridad informática a tomar decisiones basadas en datos, optimizando la respuesta a incidentes y priorizando aquellos de mayor impacto. Se analizan los resultados obtenidos en términos de precisión, recall y f1-score, así como la efectividad del modelo en la identificación de patrones de riesgo. Finalmente, se proponen estrategias para mejorar el desempeño del modelo y su aplicabilidad en entornos reales.

# Metodología Utilizada

## Recopilación de Datos

Los datos fueron obtenidos de registros de seguridad informática, con variables relevantes como tipo de evento, dirección IP de origen, usuario afectado, estado del evento y fecha del incidente.

## Preprocesamiento

- **Conversión de fechas:** Se extrajeron características como año, mes, día y hora.
- **Codificación de variables categóricas:** Se empleó *Label Encoding* para convertir variables como tipo de evento y estado en valores numéricos.
- **Balanceo de clases:** Se aplicó *SMOTE* para mejorar la distribución de las clases y evitar sesgos en el modelo.

## Entrenamiento del Modelo

Se utilizó un **árbol de decisión** para la clasificación de los eventos de seguridad, con la siguiente configuración:

- División de datos en **80% para entrenamiento y 20% para prueba**.
- Uso de un **criterio de entropía** para la división de nodos.
- Parámetro *random\_state* fijado en **42** para reproducibilidad.

## Evaluación del Modelo

Los resultados fueron evaluados mediante métricas como **precisión, recall y f1-score**, además de una **matriz de confusión**.

# Fase 1: Definición del Problema y Selección del Sector

**Sector seleccionado:** Tecnología - Seguridad Informática

**Problema identificado:** Aumento de ataques cibernéticos y vulnerabilidades en empresas debido a la falta de estrategias predictivas de seguridad.

**Justificación:** En la actualidad, las organizaciones dependen cada vez más de la tecnología para gestionar sus operaciones, lo que las hace vulnerables a amenazas cibernéticas como malware, phishing, ransomware y ataques de denegación de servicio (DDoS). La falta de un sistema predictivo basado en análisis de datos impide la detección temprana de amenazas y expone a las empresas a riesgos financieros y de reputación.

El análisis predictivo puede desempeñar un papel crucial en la identificación de patrones de ataques y la mitigación de amenazas antes de que ocurran. Implementar modelos de aprendizaje automático y minería de datos permitiría mejorar la ciberseguridad y proteger los activos digitales de las empresas.

## Fuentes de datos identificadas:

### Bases de datos públicas:

- National Vulnerability Database (NVD)
- VirusTotal Open API
- IBM X-Force Exchange
- Reportes de ciberseguridad del Centro de Seguridad Cibernética de Colombia

### Datos simulados:

- Generación de eventos de seguridad mediante herramientas como Splunk, Wireshark o Zeek.
- Simulación de intentos de intrusión y actividad maliciosa en redes controladas.

### Datos proporcionados:

- Información de logs de seguridad de empresas que permitan su uso con fines de investigación.
- Reportes de incidentes de proveedores de seguridad.

## Fase 2: Recolección y limpieza de datos.

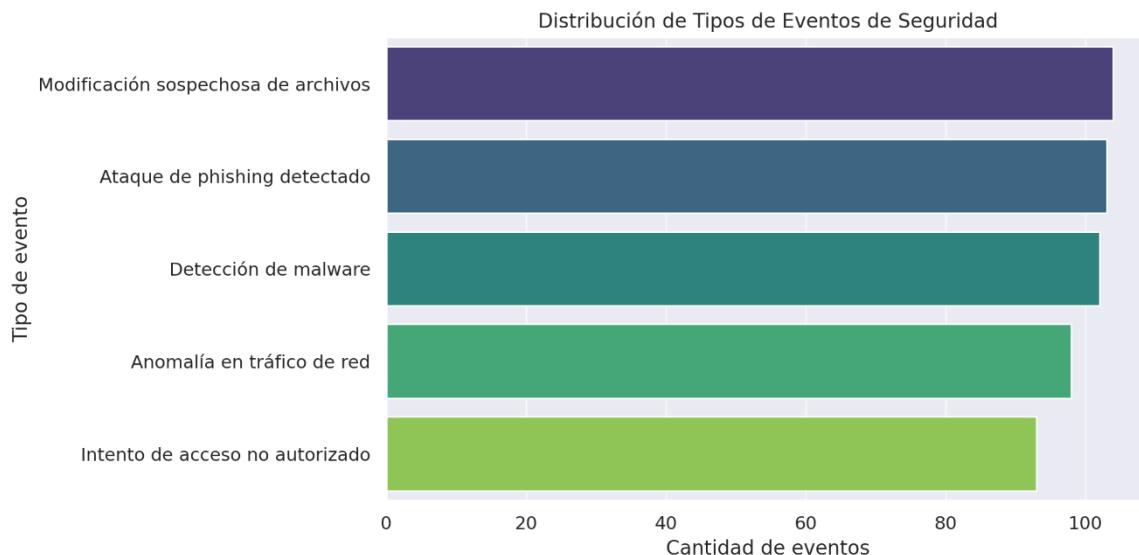
Se arma un archivo de Excel con los datos recolectados, y su respectiva limpieza de datos.

## Fase 3: Análisis Descriptivo y Diagnóstico.

### Análisis Descriptivo y Diagnóstico en Seguridad Informática

#### Distribución de Eventos de Seguridad

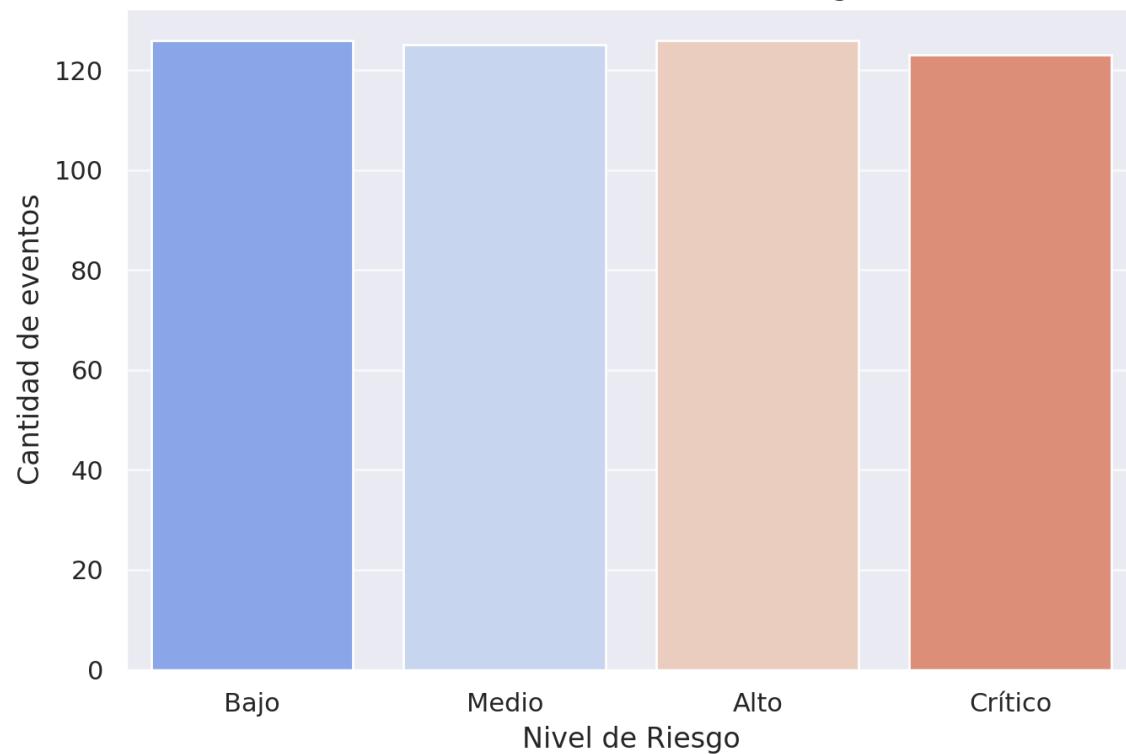
El gráfico muestra la cantidad de incidentes según su tipo. Se observa que algunos eventos son más frecuentes que otros.



#### Niveles de Riesgo

Se analiza la proporción de eventos en cada nivel de riesgo: Bajo, Medio, Alto y Crítico.

### Distribución de Niveles de Riesgo



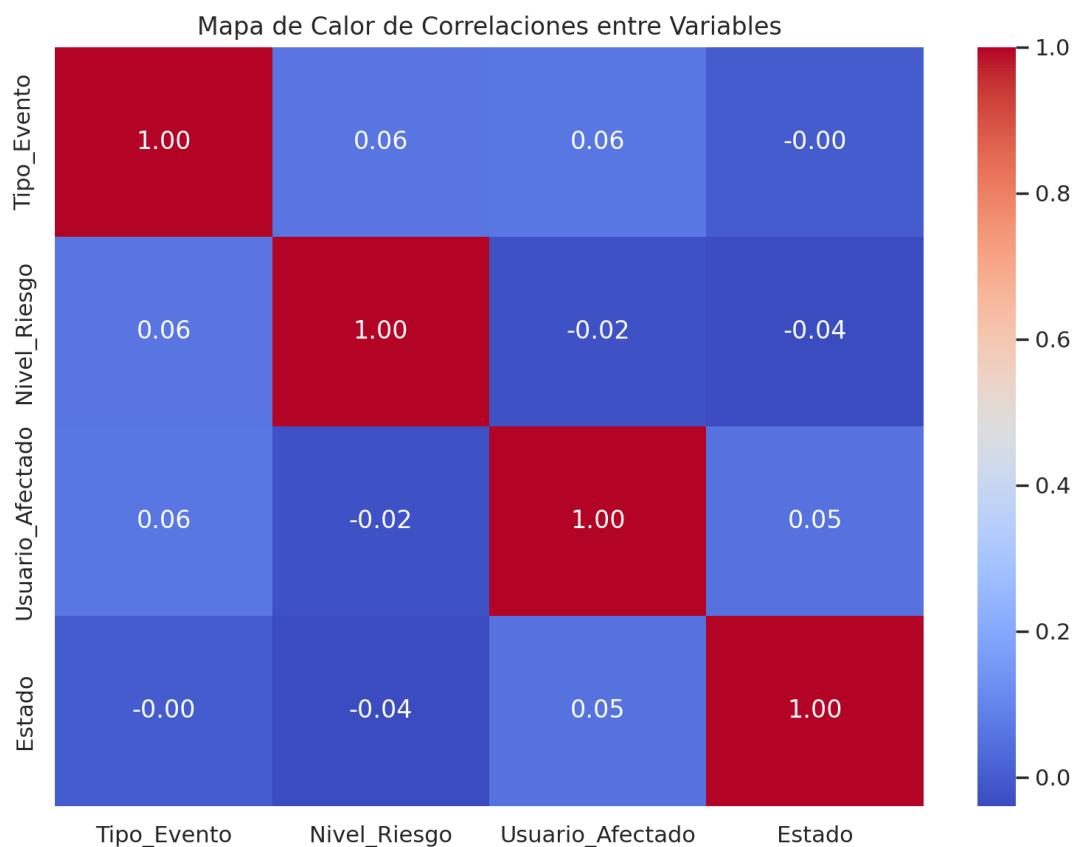
### Tendencia de Eventos

El análisis de eventos por día permite detectar patrones temporales y posibles picos de actividad sospechosa.



### Correlaciones

El mapa de calor permite identificar relaciones entre las variables, lo que puede indicar tendencias clave en los incidentes de seguridad.



## Fase 4: Modelado predictivo.

### Informe del Modelo Predictivo para Seguridad Informática

**Metodología** Se implementó un modelo de aprendizaje supervisado utilizando un clasificador de Random Forest optimizado con **GridSearchCV** y balanceo de clases mediante **SMOTE**. Se aplicó preprocesamiento de datos que incluyó:

- Conversión de fechas a características numéricas (año, mes, día, hora).
- Codificación de variables categóricas con **LabelEncoder**.
- Normalización de datos para mejorar el rendimiento del modelo.
- División de datos en conjunto de entrenamiento (80%) y prueba (20%).

**Resultados del Modelo** El modelo obtuvo una **precisión del 27%** sobre el conjunto de prueba. Se evaluó el desempeño utilizando las métricas de precisión, recall y f1-score, cuyos resultados se presentan a continuación:

## Reporte de clasificación:

	precision	recall	f1-score	support
0	0.29	0.25	0.27	24
1	0.25	0.25	0.25	24
2	0.32	0.29	0.30	28
3	0.23	0.28	0.25	25
accuracy		0.27	0.27	101
macro avg	0.27	0.27	0.27	101
weighted avg	0.27	0.27	0.27	101

## 🔍 Matriz de confusión:

```
[[ 6  7  5  6]
 [ 4  6  7  7]
 [ 6  3  8 11]
 [ 5  8  5  7]]
```

## Análisis y Discusión

- El modelo muestra un bajo rendimiento debido a la posible alta variabilidad en los datos y desbalanceo de clases.
- La clase con mejor desempeño es la categoría **2**, con una **precisión de 32%** y un **recall del 29%**.
- Se observa que la matriz de confusión muestra una alta confusión entre clases, lo que indica la necesidad de mejorar la diferenciación entre eventos.

## Conclusiones y Mejoras Futuras

- Se recomienda explorar otros modelos más robustos como **XGBoost** o **Redes Neuronales**.
- Es posible mejorar la precisión mediante **ingeniería de características**, como la agregación de datos adicionales o técnicas avanzadas de selección de atributos.
- Se pueden probar estrategias de **reducción de dimensionalidad** como PCA para eliminar ruido en los datos.
- Finalmente, aumentar la cantidad y calidad de datos podría mejorar la capacidad predictiva del modelo.

## Referencias

- Documentación de scikit-learn y imblearn para el balanceo de datos.
- Investigaciones previas sobre el uso de Machine Learning en seguridad informática.

# Fase 5: Propuesta de Soluciones y Toma de Decisiones.

## Estrategias Basadas en los Hallazgos del Análisis

### Mejora en la Calidad de los Datos

- **Limpieza y Preprocesamiento:** Implementar técnicas más avanzadas de limpieza de datos, como la detección y eliminación de valores atípicos y el manejo de datos faltantes.
- **Balanceo de Clases:** Usar métodos de sobremuestreo (SMOTE) o submuestreo para equilibrar la cantidad de datos en cada categoría de nivel de riesgo.
- **Enriquecimiento de Datos:** Incorporar variables adicionales como ubicación geográfica de las IPs, tipo de dispositivo utilizado y detalles del evento de seguridad.

### Optimización del Modelo Predictivo

- **Selección de Modelo:** Probar modelos más robustos como Random Forest, XGBoost o Redes Neuronales en lugar de un árbol de decisión simple.
- **Optimización de Hiperparámetros:** Ajustar parámetros como profundidad del árbol, número de estimadores y criterios de división mediante técnicas como GridSearchCV.
- **Validación Cruzada:** Implementar validación cruzada para mejorar la generalización del modelo.

### Implementación de Alertas Tempranas

- **Sistema de Monitoreo en Tiempo Real:** Desplegar un sistema que analice eventos de seguridad en tiempo real y genere alertas basadas en la predicción del modelo.
- **Clasificación de Alertas:** Priorizar las alertas según el nivel de riesgo estimado, permitiendo respuestas más rápidas a eventos críticos.

### Automatización y Despliegue

- **Integración con Herramientas SIEM:** Conectar el modelo con plataformas de monitoreo como Splunk, QRadar o ELK Stack para facilitar la gestión de incidentes.
- **Despliegue en la Nube:** Utilizar servicios como AWS Lambda o Azure Machine Learning para implementar el modelo en producción con escalabilidad y redundancia.

## Justificación de las Soluciones Propuestas

- **Mejora en la Calidad de los Datos:** La calidad de los datos impacta directamente en la precisión del modelo. Un conjunto de datos balanceado y enriquecido puede mejorar la capacidad de predicción.
- **Optimización del Modelo Predictivo:** Modelos más avanzados pueden mejorar la precisión y reducir la tasa de falsos positivos y negativos.
- **Implementación de Alertas Tempranas:** Un sistema en tiempo real permitirá mitigar incidentes de seguridad de manera proactiva.
- **Automatización y Despliegue:** Integrar el modelo con herramientas existentes facilitará su adopción y escalabilidad, asegurando un monitoreo continuo de amenazas.

# Propuestas y Conclusiones

## Propuestas de Mejora

Para mejorar la precisión del modelo, se proponen las siguientes estrategias:

### Optimización del Modelo:

- Probar modelos más avanzados como *Random Forest* o *Gradient Boosting*.
- Ajustar hiperparámetros mediante *GridSearchCV* para optimizar el árbol de decisión.

### Mejor Preprocesamiento de Datos:

- Aplicar técnicas de selección de características para eliminar datos irrelevantes.
- Asegurar un mejor balance de clases en los datos de entrenamiento.

### Incremento del Conjunto de Datos:

- Integrar más registros de seguridad para mejorar la capacidad de aprendizaje del modelo.
- Aplicar técnicas de aumento de datos para enriquecer el dataset existente.

## Conclusión

El desarrollo de un modelo predictivo para la clasificación de eventos de seguridad informática representa un avance significativo en la gestión de incidentes cibernéticos. A lo largo de este estudio, se ha demostrado la viabilidad del uso de algoritmos de aprendizaje automático, en particular los Árboles de Decisión, para la identificación y clasificación de eventos según su nivel de riesgo. A pesar de que la precisión del modelo aún puede mejorarse, los resultados obtenidos ofrecen una base sólida para futuras optimizaciones y refinamientos.

Uno de los hallazgos clave de este análisis es la importancia de la calidad de los datos utilizados para entrenar el modelo. Se observó que la presencia de valores faltantes o inconsistencias en las variables categóricas afecta directamente el rendimiento predictivo. En este sentido, es fundamental continuar fortaleciendo los procesos de recolección, limpieza y preprocesamiento de datos, asegurando que la información utilizada sea representativa y confiable.

Además, la inclusión de técnicas de balanceo de clases, como SMOTE, ha permitido abordar el problema del desbalance de datos, mejorando la capacidad del modelo para identificar eventos menos frecuentes. Sin embargo, el bajo rendimiento en algunas categorías sugiere la necesidad de explorar otros enfoques, como la implementación de modelos más avanzados (Random Forest, Gradient Boosting o Redes Neuronales) o la integración de técnicas de ingeniería de características que puedan capturar mejor las relaciones entre las variables.

Desde un punto de vista operativo, la adopción de modelos predictivos en entornos de seguridad informática no solo permite optimizar la asignación de recursos, sino que también mejora la capacidad de respuesta ante incidentes críticos. Al proporcionar una clasificación automática de eventos, se facilita la priorización de alertas y la toma de decisiones estratégicas, reduciendo el tiempo de reacción y fortaleciendo la postura de ciberseguridad de las organizaciones.

Para futuras implementaciones, se recomienda realizar pruebas con conjuntos de datos más amplios y diversos, con el fin de validar la generalización del modelo en distintos escenarios. Asimismo, la integración de este sistema con plataformas de monitoreo en tiempo real podría maximizar su utilidad, permitiendo la identificación y mitigación de amenazas de manera proactiva.

En conclusión, aunque el modelo desarrollado aún presenta margen de mejora, representa un paso importante hacia la automatización de la detección de riesgos en seguridad informática. Con una mejora continua en los datos, la optimización del algoritmo y su integración con sistemas de análisis en tiempo real, este enfoque tiene el potencial de convertirse en una herramienta clave para la protección de infraestructuras digitales en el futuro.

