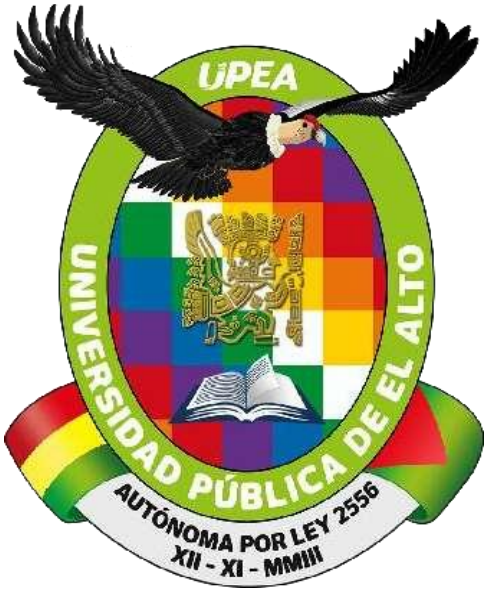
# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS




## TESIS DE GRADO

# "MODELO DE INTELIGENCIA ARTIFICIAL PARA LA DETECCIÓN DE AMENAZAS EN PLUGINS DE WORDPRESS"

**Para Optar al Título de Licenciatura en Ingeniería de Sistemas**
**MENCIÓN: INFORMÁTICA Y COMUNICACIONES**


**Postulante:** Jhonatan Carvajal Corani

**Tutor Metodológico:** M. Sc. Lic. Ing. Fanny Helen Pérez Mamani

**Tutor Especialista:** M. Sc. Lic. Ing. Neil Ramiro Gonzales Burgoa

**Tutor Revisor:** M. Sc. Lic. Ramiro Jhonatan Pardo Foronda


**EL ALTO - BOLIVIA**

**2024**

# DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

**Yo, Jhonatan Carvajal Corani con C.I. 9970850 LP.** mediante la presente **declaro** de manera pública que la propuesta de **TESIS DE GRADO** titulado **"MODELO DE INTELIGENCIA ARTIFICIAL PARA LA DETECCIÓN DE AMENAZAS EN PLUGINS DE WORDPRESS"** es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados.

Autorizo la publicación del resumen de mi propuesta en internet y me comprometo a responder a todos los cuestionamientos que se desprenden de su lectura.

Asimismo, me hago responsable ante la universidad o terceros, de cualquiera irregularidad o daño que pudiera ocasionar, por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que la **TESIS DE GRADO** haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello sometiéndome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El Alto, junio de 2024

**Jhonatan Carvajal Corani**
**C.I. 9970850 LP**
**kira18jhon@gmail.com**

## DEDICATORIA

Este trabajo está dedicado a todas aquellas personas que, de una u otra manera, han sido parte fundamental en este viaje académico.

A mis padres, Eraclio Carvajal y Victoria Corani, por su amor incondicional, su apoyo inquebrantable y por enseñarme el valor del esfuerzo y la perseverancia. Sin ustedes, nada de esto habría sido posible.

A mi hermano, Bladimir, por ser mi fuente constante de inspiración y por su inagotable confianza en mí.

A mis amigos, que con su compañía y ánimo me han ayudado a sobrellevar los momentos de dificultad. Gracias por estar siempre ahí.

A mis docentes y mentores, especialmente al Ing. William Roque Roque, por su guía, paciencia y sabiduría. Sus enseñanzas han sido cruciales para mi desarrollo académico y personal.

Finalmente, a Romina, por su amor, comprensión y por creer en mí incluso cuando yo no lo hacía. Tu apoyo ha sido un pilar fundamental en este logro.

A todos ustedes, gracias de corazón. Esta tesis es tanto mía como suya.

## AGRADECIMIENTOS

La realización de esta tesis ha sido un desafío significativo y una experiencia de aprendizaje inigualable, la cual no habría sido posible sin el apoyo y la colaboración de muchas personas a quienes me gustaría expresar mi más profundo agradecimiento.

En primer lugar, quiero agradecer a mis padres, por su amor, comprensión y apoyo incondicional. Su fe en mí y sus constantes palabras de aliento han sido una fuente constante de motivación.

A mi tutor, Ing. Neil Gonzales, por su valiosa orientación, paciencia y dedicación. Sus comentarios y sugerencias han sido esenciales para la realización de este trabajo. Su experiencia y conocimiento han enriquecido significativamente mi investigación.

A mi hermano, por su constante apoyo y ánimo. Su confianza en mí ha sido fundamental para superar los momentos difíciles.

A mis amigos, por su compañía y apoyo incondicional durante este proceso. Sus palabras de aliento y su disposición para escucharme han sido invaluables.

A mis compañeros de estudios, por su colaboración y por compartir este viaje académico. Su camaradería y apoyo han hecho de este proceso una experiencia más llevadera y enriquecedora.

## INDICE GENERAL

## ÍNDICE DE FIGURAS

## ÍNDICE DE TABLAS

## ÍNDICE DE ECUACIONES

## RESUMEN

En esta tesis se analizan diferentes métodos para la detección de amenazas en plugins de WordPress, es decir, aquellos plugins que se encuentran en nuestro conjunto de datos pero que no tienen un comportamiento normal. Encontrar patrones en el uso de plugins de WordPress sigue siendo un reto interesante en la seguridad informática. Las anomalías en los plugins aparecerán por varias razones, como actividades maliciosas o vulnerabilidades no detectadas. En este sentido, el análisis de los plugins y una combinación de técnicas de Inteligencia Artificial (Machine Learning) puede convertirse en un buen aliado para ir más allá de las firmas y ser capaz así de encontrar patrones previamente desconocidos. En cada predicción realizada por un modelo diferente, la precisión varía, esto dependerá de cómo el algoritmo se comporte con nuestro Data Set y si logra cumplir con la predicción necesaria, lo cual es la detección de amenazas en los plugins de WordPress.

En cada análisis y evaluación de cada método de predicción, los resultados obtenidos indicaron que aquellos métodos menos precisos en la predicción de detección de amenazas fueron los modelos de regresión logística y de clasificación. Por otro lado, el método que ofreció mejores resultados fue el de árboles de decisión, cumpliendo con nuestro objetivo de detectar amenazas en los plugins de WordPress de manera efectiva.

***Palabras Claves:*** Inteligencia Artificial, WordPress, Protección Sitios Web.

## ABSTRACT

This thesis analyzes different methods for detecting threats in WordPress plugins, that is, those plugins that are in our data set but do not have normal behavior. Finding patterns in the use of WordPress plugins continues to be an interesting challenge in computer security. Plugin anomalies will appear for various reasons, such as malicious activities or undetected vulnerabilities. In this sense, the analysis of plugins and a combination of Artificial Intelligence (Machine Learning) techniques can become a good ally to go beyond signatures and thus be able to find previously unknown patterns. In each prediction made by a different model, the accuracy varies, this will depend on how the algorithm behaves with our Data Set and if it manages to meet the necessary prediction, which is the detection of threats in WordPress plugins.

In each analysis and evaluation of each prediction method, the results obtained indicated that the least accurate methods in predicting threat detection were the logistic regression and classification models. On the other hand, the method that offered the best results was decision trees, meeting our objective of detecting threats in WordPress plugins effectively.

***Keywords:*** Artificial Intelligence, WordPress, Website Protection.

# CAPÍTULO I

# CAPITULO I

## MARCO PRELIMINAR

### 1.1. INTRODUCCION

WordPress, como una de las plataformas de gestión de contenido más utilizadas a nivel mundial, ha facilitado la creación de sitios web dinámicos y funcionales para una amplia gama de usuarios. Sin embargo, el crecimiento exponencial en la diversidad y cantidad de plugins disponibles para WordPress ha llevado consigo la proliferación de posibles vulnerabilidades y riesgos de seguridad.

Los plugins, siendo componentes fundamentales para extender la funcionalidad de WordPress, pueden introducir brechas de seguridad significativas si no se gestionan adecuadamente. La constante evolución de amenazas cibernéticas, desde ataques de inyección de código hasta vulnerabilidades de seguridad conocidas, plantea un desafío continuo para mantener la integridad y la seguridad de los sitios web basados en esta plataforma.

En este contexto, esta tesis se centra en explorar y analizar cómo la aplicación de un modelo de Inteligencia Artificial (IA) puede ser aprovechada para mejorar la evaluación, detección y mitigación de vulnerabilidades en los plugins de WordPress. El enfoque estratégico en la integración de la IA en el proceso de seguridad busca ofrecer soluciones innovadoras y efectivas para identificar riesgos y mitigar posibles amenazas de manera proactiva.

El objetivo principal de este estudio es examinar en profundidad cómo la inteligencia artificial puede ser implementada para fortalecer la seguridad en WordPress, centrándose en la identificación temprana de vulnerabilidades, la detección de patrones de comportamiento malicioso y la mitigación proactiva de riesgos en los plugins.

A través de una revisión exhaustiva de la literatura, análisis de casos y experimentación con enfoques de inteligencia artificial, esta tesis tiene como propósito contribuir con conocimientos significativos para mejorar la seguridad y protección de los sitios web en WordPress, ofreciendo métodos adaptativos para la evaluación y gestión de vulnerabilidades en plugins.

## 1.2. ANTECEDENTES

### *1.2.1. Antecedentes internacionales*

(Brito, s.f.) “ESTUDIO DE PATRONES DE INTENTOS DE CIBERATAQUES ASOCIADOS A LAS VULNERABILIDADES DEL COMPLEMENTO REVSLIDER”, se describen los patrones de intentos de ciberataques contra aplicaciones web basadas en WordPress a través del componente RevSlider. La presencia de versiones desactualizadas y vulnerables de RevSlider ha comprometido cientos de miles de sitios desde el año 2014.

### *1.2.2. Antecedentes nacionales*

Cornejo (2020), “DETECCIÓN DE MALWARE EN UNA RED CON MACHINE LEARNING”, tiene como objetivo principal plantear un modelo que detecte de manera automática malware en una red también automatizar la búsqueda de nuevos patrones de ataque, gracias a herramientas el modelo ayudará en la detección de malware, funcionará por medio del tráfico de una red donde los datos serán analizados, como si una persona o un equipo de trabajo analizara. Las herramientas utilizadas fueron principalmente WireShark. El modelo se realizó con la metodología CRISP-DM, La tesis se realizó en la UNIVERSIDAD MAYOR DE SAN ANDRÉS.

Mamani (2020), “MODELO BASADO EN INTELIGENCIA ARTIFICIAL PARA LA DETECCIÓN DE CIBERACOSO”, su objetivo principal es desarrollar un modelo inteligente que detecte el ciberacoso en jóvenes de 15 a 17 años en Bolivia. Se realizó con la ayuda

de la metodología fundamental de ciencia de datos de IBM. Las herramientas que se usaron principalmente para la construcción y entrenamiento de la Red Neuronal Recurrente (RNN) de Deep Learning se utilizó keras juntamente con Jupyter en Anaconda. La tesis se realizó en la UNIVERSIDAD MAYOR DE SAN ANDRÉS.

### *1.2.3. Antecedentes locales*

Mena (2020), “TÉCNICAS DE SEGURIDAD INFORMÁTICA PARA REDUCIR LAS VULNERABILIDADES POR INYECCIÓN SQL EN APLICACIONES WEB”, se tiene como objetivo plantear un conjunto de TÉCNICAS EN SEGURIDAD INFORMÁTICA, mediante los procedimientos de programación PHP, para reducir el nivel de vulnerabilidades por inyección SQL en aplicaciones web desarrollados con PHP y base de datos MySQL. La tesis se realizó en la UNIVERSIDAD PÚBLICA DE EL ALTO.

## 1.3. PLANTEAMIENTO DEL PROBLEMA

La plataforma de sistema de gestión de contenido WordPress ha tenido un crecimiento exponencial a lo largo de los años en el mundo global es ampliamente utilizada para la creación de sitios web y blogs. Sin embargo, al ser muy conocida y bastante utilizada cuenta con una gran integración de plugins de terceros, aunque proveen funcionalidades adicionales también representan un potencial riesgo y amenazas de seguridad. En WordPress la seguridad se ve constantemente desafiada por la presencia de vulnerabilidades en plugins, lo que expone a los sitios web a amenazas como ataques de inyección de código, vulnerabilidades conocidas y otras formas de explotación maliciosa. La detección temprana de estas amenazas se ha convertido en un aspecto primordial para salvaguardar la integridad y la confidencialidad de los datos alojados en esta plataforma.

### *1.3.1. Problema principal*

El problema central que motiva este estudio de investigación radica en las vulnerabilidades inherentes que tiene esta plataforma de WordPress. Así también el hacer uso de plugins desactualizados, representa una amenaza significativa para los usuarios de esta plataforma ya que pueden ser atacados por hackers que podrían intentar obtener el acceso al panel de control de WordPress. Es crucial abordar estas vulnerabilidades y promover prácticas de seguridad sólidas para garantizar la integridad y la protección de los sitios web que utilizan esta plataforma.

### *1.3.2. Problemas secundarios*

Si un sitio de WordPress es comprometido debido a una vulnerabilidad o plugin desactualizado, los datos sensibles de los usuarios, como información personal o de pago, podrían ser comprometidos.

Los usuarios pueden perder la confianza en un sitio web si saben que es vulnerable a ataques cibernéticos. Esto puede resultar en una disminución de tráfico, ventas y reputación de la marca.

Si un sitio de WordPress es pirateado y se utiliza para distribuir malware o spam, la reputación de la empresa propietaria del sitio podría verse seriamente dañada.

Si un sitio de WordPress es atacado con éxito, puede resultar en tiempo de inactividad del sitio mientras se realizan las reparaciones necesarias, lo que podría afectar negativamente a la experiencia del usuario y las operaciones comerciales.

### *1.3.3. Formulación del problema*

¿De qué manera se pueden detectar las amenazas existentes en los sitios web diseñados bajo la plataforma de WordPress?

## 1.4. OBJETIVOS

### *1.4.1. Objetivo general*

Modelar una Inteligencia Artificial para la detección de amenazas en plugins de WordPress.

### *1.4.2. Objetivos específicos*

- Investigar los métodos más adecuados para la detección de amenazas en plugins de WordPress.
- Identificar características específicas de las vulnerabilidades que puedan generar amenazas como datos de entrada para el modelo de inteligencia artificial.
- Desarrollar un modelo predictivo basado en Inteligencia Artificial que pueda prever posibles amenazas en los plugins de WordPress.
- Facilitar la identificación de posibles amenazas en los plugins de WordPress.
- Realizar pruebas de validación utilizando datos para evaluar su precisión en la detección de vulnerabilidades.

## 1.5. HIPÓTESIS

### *1.5.1. Hipótesis de investigación*

La creación de un modelo de Inteligencia Artificial, podrá detectar amenazas en plugins de WordPress, fortaleciendo la seguridad de los sitios en esta plataforma.

### *1.5.2. Hipótesis nula*

La creación de un modelo de Inteligencia Artificial, no podrá detectar amenazas en plugins de WordPress, fortaleciendo la seguridad de los sitios en esta plataforma.

### *1.5.3. Hipótesis alternativa*

El modelo de inteligencia artificial mejorará la detección de amenazas en plugins de WordPress en comparación con los métodos tradicionales de seguridad.

### *1.5.4. Identificación de variables*

**Variable independiente**

Modelo de inteligencia artificial

**Variable dependiente**

Detección de amenazas en plugins de WordPress

## 1.6. JUSTIFICACIÓN

### *1.6.1. Justificación científica*

Abordar un problema actual y relevante en seguridad informática, aprovechando la IA como una herramienta innovadora para mejorar la evaluación y mitigación de vulnerabilidades en la plataforma de WordPress, lo que contribuirá al avance del conocimiento científico en el campo de la seguridad cibernética.

### *1.6.2. Justificación técnica*

Desarrollar y aplicar técnicas innovadoras basadas en Inteligencia Artificial para mejorar la detección y mitigación de vulnerabilidades en plugins de WordPress, lo que contribuirá a fortalecer la seguridad de los sitios web en esta plataforma de gestión de contenido.

### *1.6.3. Justificación económica*

Las vulnerabilidades en los complementos de WordPress pueden provocar costosos incidentes de seguridad, como ataques cibernéticos, robo de datos y pérdida de tiempo y

recursos para reconstruir un sitio. Al identificar y remediar vulnerabilidades de manera proactiva, se puede reducir los costos de mitigación de incidentes.

### *1.6.4. Justificación social*

Las evaluaciones de vulnerabilidad de los complementos de WordPress ayudan a proteger la privacidad de los usuarios del sitio web y evitan la divulgación no autorizada de datos personales y la recopilación de información confidencial. La ciberseguridad es fundamental para prevenir el robo de identidad, el fraude en línea y otros delitos cibernéticos que pueden afectar a personas y empresas.

## 1.7. METODOLOGÍA

### *1.7.1. Método científico*

El método científico proporciona una estructura clara para definir el problema, formular hipótesis, diseñar experimentos, recopilar, analizar datos, y llegar a conclusiones. Esto asegura que el proceso de desarrollo del modelo sea organizado y metódico.

El trabajo de investigación tiene un efecto causal ya que implica investigar si la creación de un modelo de Inteligencia Artificial tiene un efecto en la capacidad de detectar amenazas en plugins de WordPress y fortalecer la seguridad de los sitios en esta plataforma. En este caso, la investigación estaría orientada a determinar si la creación e implementación de este modelo específico tiene un impacto directo en la seguridad de los sitios web basados en WordPress.

#### 1.7.1.1. Técnicas de investigación

Las técnicas de investigación que serán utilizadas en este estudio consisten en realizar estudios de caso detallados de incidentes pasados donde plugins de WordPress

fueron comprometidos, analizando las causas, el proceso de ataque y las medidas de mitigación adoptadas.

El tipo de investigación que se usa es mixto que combina métodos cuantitativos y cualitativos para una comprensión más completa del problema. Los métodos cuantitativos permitirían analizar datos sobre vulnerabilidades y la eficacia del modelo, mientras que los cualitativos, como estudios de casos, proporcionarían un contexto profundo y detallado. Esta combinación ayudaría a validar y mejorar continuamente el modelo de IA, integrando tanto la precisión estadística como las perspectivas prácticas y experienciales.

El estudio de investigación es analítico ya que es un tipo de estudio en el que se trata de establecer relaciones de asociación o de causalidad entre las variables. Es decir, en un estudio analítico se investigan las relaciones entre las diferentes variables de estudio.

El diseño no experimental permite estudiar fenómenos como las amenazas y la efectividad de las medidas de seguridad utilizando datos observados y recopilados en condiciones reales, proporcionando insights prácticos y aplicables para mejorar las prácticas de seguridad sin comprometer la integridad de los sistemas involucrados.

Un enfoque descriptivo es adecuado para estudiar la detección de amenazas en plugins de WordPress porque permite identificar y catalogar de manera detallada las vulnerabilidades y métodos de ataque específicos que afectan a estos plugins. Esto proporciona datos históricos y patrones que son fundamentales para evaluar las medidas de seguridad existentes y guiar mejoras continuas en la protección contra amenazas futuras.

### *1.7.2. Método de ingeniería*

#### 1.7.2.1. Metodología CRISP-DM

CRISP-DM (Cross-Industry Standard Process for Data Mining) es una metodología ampliamente aplicada a proyectos dedicados a minería de datos y aprendizaje automático debido a la estructura clara y adaptable. Surgen algunas limitaciones de la metodología frente a cambios en los proyectos de ciencia de datos que necesitan que se adapten a esos nuevos cambios aplicándolo en ciertas fases de la metodología (IBM, 2021).

***a) Entendimiento del negocio***

Durante esta fase, se definen los objetivos de negocio, se evalúa la situación actual, se traducen los objetivos de negocio en objetivos técnicos de minería de datos y se desarrolla un plan de proyecto detallado. Esta fase sienta las bases para el éxito del proyecto al garantizar que el equipo tenga una comprensión clara de lo que se espera lograr y cómo se integrará el análisis de datos en los objetivos empresariales más amplios.

***b) Entendimiento de los datos***

En esta fase el equipo técnico realiza un análisis exploratorio para obtener una visión general de lo que se puede lograr con los datos. Este análisis se guía por el conocimiento de negocio adquirido en la fase anterior, complementando así el trabajo realizado. Durante esta fase, se recopilan datos relevantes, se describen las características de los datos, se exploran para identificar patrones y valores atípicos, y se verifica la calidad de los datos. Este análisis permite al equipo tener una comprensión más profunda de los datos disponibles y cómo podrían contribuir a alcanzar los objetivos del negocio definidos en la fase anterior.

***c) Preparación de los datos***

La preparación de los datos cubre todas las actividades necesarias para construir el conjunto de datos definitivo que se usará en la fase de modelado.

***d) Modelado***

La fase de modelado elige las técnicas más adecuadas para el tipo de datos y objetivos del proyecto

***e) Evaluación***

El cliente determina la calidad de los resultados obtenidos de los modelos desarrollados en la fase de modelado. Esta evaluación asegura que los resultados cumplan con los objetivos de negocio definidos anteriormente. Basado en esta evaluación, el cliente decide cómo pueden explotarse estos resultados antes de proceder a la fase de despliegue. Esta fase es crucial para validar que los modelos y análisis realizados son adecuados y útiles para los fines del proyecto.

***f) Despliegue***

Se implementan los modelos y resultados obtenidos en un entorno de producción. Esta fase incluye desarrollar un plan detallado de despliegue, establecer procedimientos para monitorear y mantener el modelo, y documentar los resultados y el proceso para futuros proyectos. Además, se realiza una revisión post-implementación para evaluar el éxito del proyecto y las lecciones aprendidas. El objetivo es asegurar que los modelos se integren eficazmente en las operaciones del negocio y proporcionen valor continuo.

## 1.9. HERRAMIENTAS

### *1.9.1. Entorno de Desarrollo*

- **WordPress**

  Es una plataforma de gestión de contenidos (CMS) de código abierto que permite crear y gestionar sitios web y blogs de manera sencilla. Es extremadamente popular debido a su flexibilidad, permitiendo la personalización mediante temas y plugins, y su interfaz intuitiva que facilita el uso para personas sin conocimientos técnicos avanzados. Además, es amigable con el SEO, escalable para diferentes tipos de proyectos y cuenta con una comunidad activa que contribuye a su desarrollo y seguridad constante.

### *1.9.2. IDE (Entorno de Desarrollo Integrado)*

- **Google Colab**

  Es una plataforma gratuita basada en la nube que permite escribir y ejecutar código Python directamente en el navegador, compatible con Jupyter Notebooks. Ofrece acceso gratuito a GPUs y TPUs, lo que es ideal para tareas de aprendizaje automático y procesamiento de grandes datos. Facilita la colaboración en tiempo real, similar a Google Docs, y permite la instalación de bibliotecas directamente en el entorno de ejecución. Además, se integra fácilmente con Google Drive y GitHub, haciendo más sencilla la gestión y compartición de archivos y proyectos.

### *1.9.3. Lenguajes de programación*

- **Python**

  Es un lenguaje de programación de alto nivel y de propósito general que se destaca por su sintaxis clara y legible. Es ampliamente utilizado en una variedad de aplicaciones, desde desarrollo web y scripting hasta análisis de datos, inteligencia

artificial y aprendizaje automático. Gracias a su comunidad activa y extensa biblioteca estándar y de terceros, Python facilita la implementación rápida de soluciones robustas y escalables.

### *1.9.4. Visualización y Análisis de Resultados*

- **Matplotlib**

  Para visualización de datos y resultados de los modelos de IA.

- **Pandas**

  Librería de Python para manipulación y análisis de datos, útil para el análisis de resultados obtenidos durante las pruebas.

- **Numpy**

  Es una librería fundamental en Python utilizada principalmente para el cálculo numérico y el manejo de arrays multidimensionales.

## 1.10. LIMITES Y ALCANCES

### *1.10.1. Limites*

- No se habrá revisión y análisis de métodos y técnicas de detección de amenazas específicamente aplicables a plugins de WordPress.
- No será posible la identificación de patrones y características comunes en amenazas previamente reportadas en plugins de WordPress.
- Desarrollo de un modelo de aprendizaje automático para predecir la presencia de amenazas en plugins de WordPress.
- Evaluación del modelo desarrollado mediante pruebas controladas con plugins de WordPress conocidos y desconocidos.

***Alcances***

- Análisis de tipos de vulnerabilidades explotables.
- Implementación de técnicas como aprendizaje supervisado (clasificación), utilizando conjuntos de datos etiquetados de vulnerabilidades conocidas.
- Desarrollo de un modelo de interfaz de usuario intuitiva para facilitar el uso por parte de administradores de sitios web.
- Medición de métricas como precisión, sensibilidad y especificidad del modelo en la detección de vulnerabilidades.

## 1.11. APORTES

- Al detectar y prever amenazas en plugins de WordPress, se reduce significativamente el riesgo de que los sitios web sean comprometidos por atacantes.
- En lugar de reaccionar a los ataques después de que ocurran, se pueden identificar y mitigar las amenazas antes de que se exploten.
- Los usuarios y desarrolladores de WordPress podrán confiar más en la seguridad de la plataforma, lo que puede llevar a una mayor adopción y uso seguro de plugins.
- Un sistema automatizado basado en IA puede analizar grandes volúmenes de datos y plugins de manera eficiente, lo que no es viable con métodos manuales.

# CAPÍTULO II

# CAPÍTULO II

## MARCO TEÓRICO

### 2.1. Inteligencia

Es preciso hablar de “inteligencias”, y no sólo de “inteligencia”, en el caso de los humanos, en el de los animales, en el de los vegetales, e incluso en el de la tierra según la hipótesis Gaia. Podemos llamar entonces “inteligencia” en sentido amplio a la capacidad de perseguir metas, planificar, prever consecuencias de las acciones y emplear herramientas para alcanzar las metas. La inteligencia sería la capacidad de resolver problemas con instrumentos (Orts, 2019).

### 2.2. Inteligencia Artificial

Los sistemas de IA son sistemas de software (y posiblemente también de hardware), diseñados por humanos que, dada una meta compleja, actúan en la dimensión física o digital percibiendo su entorno mediante la adquisición de datos, interpretando los datos recogidos, estructurados o no estructurados, razonando sobre el conocimiento o procesando la información derivada de estos datos y decidiendo las mejores acciones que hay que realizar para alcanzar la meta. Los sistemas IA pueden utilizar reglas simbólicas o aprender un modelo numérico, y pueden también adaptar su conducta analizando cómo el entorno es afectado por las acciones previas (Orts, 2019).

La inteligencia artificial se refiere a la capacidad de las máquinas para imitar el comportamiento humano y realizar tareas que normalmente requerirían de la inteligencia humana, los avances en la capacidad de procesamiento, el aprendizaje automático y la disponibilidad de grandes conjuntos de datos han impulsado el desarrollo de la IA en diversos sectores (Ramírez, 2024).

Esta capacidad de aprendizaje y toma de decisiones de las máquinas está en constante evolución, permitiéndoles asumir cada vez más tareas que antes eran exclusivas de los humanos. Las aplicaciones de la IA ya están transformando diversos ámbitos de nuestra vida, ofreciendo mejoras significativas en eficiencia y productividad (Rouhiainen, 2018)

### *2.2.1. Inteligencia Artificial como ciencia*

La inteligencia artificial busca comprender y replicar la inteligencia humana, desde procesos cerebrales básicos hasta funciones cognitivas complejas. Su objetivo científico es desarrollar teorías y modelos computacionales que expliquen el comportamiento humano (Antonio Fernandez, 2006).

La IA proporciona herramientas para la neurociencia y la ciencia cognitiva, pero evita debates sobre la equivalencia entre pensar y computar. Se enfoca en resolver problemas prácticos, como la falta de claridad en las especificaciones, la imprecisión y el aprendizaje (Mira, 2008, pág. 7).

Por otra parte, como disciplina científica, la IA incluye varios enfoques y técnicas, tales como Machine Learning (del que son ejemplos el Deep Learning y el reinforcement Learning), el machine reasoning (que incluye planificar, programar, representaciones de conocimiento y razonamiento, búsqueda y optimización), y robótica (que incluye control, percepción, sensores y actuadores (actuators), y la integración de todas las demás técnicas en los sistemas ciber físicos (Orts, 2019).

### *2.2.2. Inteligencia Artificial como ingeniería*

La Ingeniería del Conocimiento, una rama de la inteligencia artificial, enfrenta desafíos debido a su enfoque en el conocimiento abstracto y la falta de una teoría sólida del mismo. Se centra en tareas comunes y científicas, y utiliza sistemas basados en el

conocimiento para modelar interacciones humanas y resolver problemas en diversos campos (Mira, 2008, pág. 7).

Las ingenierías de la materia y la energía se basan en las sólidas teorías de la Física. Sin embargo, la IC no puede basarse en una sólida teoría del conocimiento porque no disponemos de esa teoría. Esta es otra de las razones de la disparidad ostensible entre objetivos y resultados de la IA (Antonio Fernandez, 2006, pág. 9).

### *2.2.3. Orígenes de la inteligencia artificial*

Investigadores de Lancashire como Alan Turing y John McCarty comenzaron a explorar la posibilidad de producir máquinas que pudieran funcionar como un cerebro humano y sistema de aprendizaje en la década de 1950. Los sistemas de lógica simbólica, el aprendizaje automático y las redes neuronales se encuentran entre las muchas técnicas que se han desarrollado para abordar los problemas de la inteligencia artificial

En cuanto a la inteligencia artificial, nace en 1955, en un congreso en Los Ángeles sobre máquinas que aprenden. John McCarthy introduce la expresión “inteligencia artificial” en 1956 y se refiere con ella a la creación de máquinas que pueden tenerse por inteligentes porque interactúan con los seres humanos hasta el punto de que una persona ya no sabe si está hablando con una máquina o con otra persona humana. Es lo que recibe el nombre de “test de Turing”10. La IA puede llegar a constituir un nuevo tipo de inteligencia. (Orts, 2019, pág. 382)

Los orígenes de la inteligencia artificial según López y Meseguer (2017), el nacimiento oficial de la IA se da en el año 1956 en un encuentro científico en el Dartmouth College, aunque como veremos, un año antes ya había tenido lugar una sesión dedicada al tema del aprendizaje automático en un congreso celebrado en Los Ángeles. Después

describimos los primeros programas capaces de demostrar teoremas y resolver una variedad de problemas relativamente sencillos.

En los años setenta, en el Stanford Research Institute, Duda y Hart desarrollaron un sistema basado en redes neuronales capaz de aprender a reconocer instrucciones manuscritas del lenguaje de programación Fortran. En aquellos tiempos, los programadores escribían a mano el programa en tarjetas perforadas que el ordenador leía ópticamente. El sistema cometía solamente un 2% de errores por lo que tenía una tasa de acierto muy alta para la época (Gonzales, 2017).

### *2.2.4. Tipos de inteligencia artificial*

En este ámbito de la inteligencia artificial pueden distinguirse en varias modalidades que plantean problemas éticos diferenciados.

#### 2.2.4.1. Inteligencia Artificial basada en reglas

En la inteligencia artificial (IA), el razonamiento basado en reglas es un enfoque tradicional que se basa en la lógica formal. Utiliza sistemas de inferencia que aplican reglas predefinidas a datos para derivar conclusiones. Estas reglas se expresan típicamente en forma de declaraciones If-Then y siguen principios de lógica deductiva y técnicas de coincidencia de patrones (360, 2024).

A medida que la IA ha avanzado, los motores de reglas han evolucionado para integrar métodos probabilísticos y lógica difusa. Esta integración permite manejar incertidumbres y conocimiento impreciso de manera más efectiva, siendo especialmente útil en campos donde la exactitud absoluta no es posible, como en el diagnóstico médico o la evaluación de riesgos (360, 2024).

Mientras que los sistemas expertos en la era temprana de la IA eran rígidos y estáticos, los sistemas de control actuales son dinámicos y auto-evolutivos, aplicando algoritmos como la optimización de enjambre de partículas (PSO) y algoritmos genéticos, las reglas y sus parámetros se han refinado para aumentar su adaptabilidad (360, 2024).

Además, están surgiendo técnicas de meta-aprendizaje que permiten a los sistemas de control optimizar sus propios procesos de aprendizaje, una capacidad anteriormente limitada al ámbito tradicional del aprendizaje automático (360, 2024).

#### 2.2.4.2. Inteligencia Artificial en aprendizaje automático

El aprendizaje automático, también conocido como Machine Learning, es un subcampo de la inteligencia artificial que permite a las máquinas aprender y mejorar sus capacidades. Los algoritmos y modelos matemáticos permiten a las computadoras hacer predicciones y tomar decisiones analizando grandes cantidades de datos

El aprendizaje automático es el proceso mediante el cual las máquinas pueden aprender de los datos y mejorar su rendimiento a lo largo del tiempo. En lugar de ser programadas de manera manual, las máquinas utilizan algoritmos y modelos matemáticos para aprender de los patrones presentes en los datos y realizar tareas específicas (Informática y Tecnología Digital, 2024).

#### 2.2.4.3. Tipos de aprendizaje automático

Los sistemas informáticos de aprendizaje automático utilizan la experiencia y la evidencia en forma de datos para comprender patrones y comportamientos de forma única.

Esto permite anticipar escenarios e iniciar operaciones que brinden soluciones a tareas específicas.

Se pueden identificar los cuatro tipos principales de aprendizaje automático:

**a) Aprendizaje supervisado**

En este tipo de aprendizaje, la máquina aprende por una instrucción y guía del humano, se entrena a la máquina entregando un conjunto de datos etiquetados. Su objetivo es aprender una función que pueda mapear las entradas a las salidas correctas. Se usa comúnmente en aplicaciones empresariales como chat bots, donde se le enseña al sistema cómo responder a ciertos estímulos basándose en datos históricos. Esto permite automatizar tareas y procesos, mejorando la eficiencia y la precisión (Drew, 2024).

Sirven para resolver problemas como regresión y clasificación, el primero obtiene datos de entrada y predice datos numéricos de salida y en la clasificación obtiene datos de entrada y la categoría a la que pertenece.

**b) Aprendizaje no supervisado**

El aprendizaje no supervisado no trabaja con datos etiquetados, consiste en analizar la información sin etiquetas para encontrar alguna estructura en ella.

Según Drew (2024), en este caso la máquina aprende sin ninguna guía. En el aprendizaje no supervisado, los datos que se utilizan no están etiquetados ni organizados, sino que es la misma computadora la que aprende a agrupar y organizar según patrones, similitudes y normas que logre detectar.

**c) Aprendizaje por refuerzo**

Es un aprendizaje que resuelve problemas mediante prueba y error, basándose en datos entrena distintos escenarios para tomar una serie de decisiones. El agente toma acciones y recibe retroalimentación, por cada acción

recibe una recompensa o castigos. Este enfoque se utiliza en aplicaciones como juegos de video, robótica y control de procesos (MONROY, 2024).

**d) Aprendizaje semi supervisado**

Es una técnica de aprendizaje automático que combina elementos del aprendizaje supervisado y no supervisado. El algoritmo utiliza tanto los datos etiquetados como los no etiquetados para mejorar el rendimiento del modelo. Esto es útil cuando obtener datos etiquetados es costoso o difícil, ya que el modelo puede aprender de una gran cantidad de datos no etiquetados y utilizar la información limitada etiquetada para ajustar y mejorar su desempeño (Zendesk, 2024).

### *2.2.5. Inteligencia Artificial en redes neuronales artificiales*

Las Redes Neuronales Artificiales, ANN (Artificial Neural Networks) están inspiradas en las redes neuronales biológicas del cerebro humano. Están constituidas por elementos que se comportan de forma similar a la neurona biológica en sus funciones más comunes. Estos elementos están organizados de una forma parecida a la que presenta el cerebro humano (Olabe, 2024).

Las ANN al margen de "parecerse" al cerebro presentan una serie de características propias del cerebro. Por ejemplo, las ANN aprenden de la experiencia, generalizan de ejemplos previos a ejemplos nuevos y abstraen las características principales de una serie de datos (Olabe, 2024).

#### 2.2.5.1. Inteligencia Artificial basada en procesamiento del lenguaje natural

El procesamiento de lenguaje natural (NLP) es una tecnología de Machine Learning que brinda a las computadoras la capacidad de interpretar, manipular y comprender el lenguaje humano. Hoy en día, las organizaciones tienen grandes volúmenes de datos de

voz y texto de varios canales de comunicación, como correos electrónicos, mensajes de texto, fuentes de noticias en redes sociales, vídeo, audio y más. Utilizan software de NLP para procesar de forma automática estos datos, analizan la intención o el sentimiento del mensaje y responden en tiempo real a la comunicación humana (Aws, 2024).

### *2.2.6. Machine Learning*

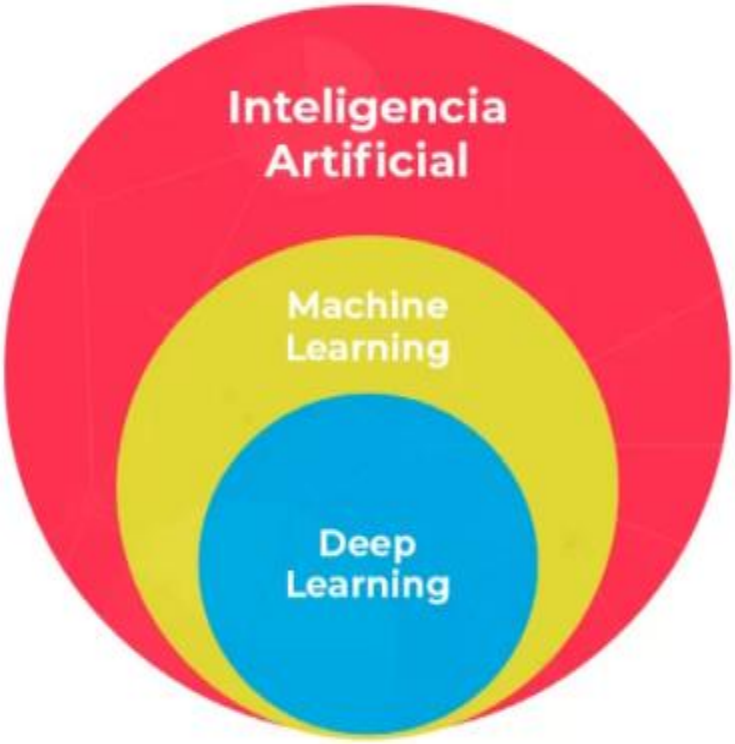
Para explicar la importancia del Machine Learning como factor en el desarrollo global, debemos remontarnos a sus orígenes y, sobre todo, comprender a fondo sus raíces.

En 1943. año en el que el matemático Walter Pitts y el neurofisiólogo Warren McCulloch, dieron a conocer su trabajo enfocado a lo que hoy conocemos como inteligencia artificial, pues en su teoría proponían analizar el cerebro como un organismo computacional y la creación de computadoras que funcionaran igual o mejor que nuestra red neuronal. (RAMÍREZ, 2018).

El aprendizaje automático (Machine Learning) es una subdisciplina de la inteligencia artificial que se centra en desarrollar algoritmos y modelos que permiten a las computadoras aprender y mejorar su rendimiento en tareas específicas a partir de datos, sin necesidad de ser programadas explícitamente para cada tarea (Oracle, 2024).

***Figura 1***

***Campos de la Inteligencia Artificial***




Nota: Campos de la Inteligencia Artificial, 2024, Fuente: Universidad Computense ¿Qué es el Machine Learning? (https://www.masterdatascienceucm.com/que-es-machine-learning/)

Machine Learning es la ciencia que hace que los ordenadores “aprendan” a partir de los datos. En vez de programar, paso a paso, cada solución específica para cada necesidad planteada, tal y como se realiza en el enfoque de la programación convencional, el área de machine learning está dedicada al desarrollo de algoritmos genéricos que pueden extraer patrones de diferentes tipos de datos. De esta manera, un programa de machine learning destinado, por ejemplo, a clasificar números escritos a mano, no va a diferir sustancialmente de un programa destinado a la clasificación de las imágenes de señales de tráfico: ambos se basarán en la existencia de algún tipo de algoritmo de machine learning que clasifique datos etiquetados. En este punto se podría pensar que el proceso completo de machine learning es fácilmente automatizable, cuando realmente no es el caso: un

ingeniero de datos (data scientist) debe llevar a cabo numerosas tareas específicas tales como la identificación de la fuente de datos, su limpieza, la eliminación de información que esté fuertemente correlacionada, la búsqueda de información sesgada, la realización de las normalizaciones necesarias, la identificación de los tipos de soluciones de machine learning cuya aplicación resulte apropiada, la elección del algoritmo más adecuado, el ajuste fino de los hiper-parámetros del método elegido, el análisis de los resultados, la identificación de comportamientos incorrectos, la vuelta a procesos anteriores con el fin de cambiar lo que resulte necesario para mejorar los resultados (Bobadilla, 2020).

La clave de la capacidad de un sistema de Aprendizaje Automático se encuentra en la construcción y adaptación de los árboles de decisiones en base a los datos previamente conocidos por el sistema. Pero también influye la aplicación de fórmulas heurísticas en los nodos que forman el árbol, para el que se elabora un sistema de inferencias (Tarqui, 2020).

El sistema de Machine Learning necesita contar con un volumen de datos de relevancia para poder suministrar respuestas realmente válidas. El mínimo que se recomienda es de 6 entradas de datos reales para cada respuesta nueva diseñada, y esto debe repetirse para cada variable que conforman el sistema de trabajo del sistema (Tarqui, 2020).

Machine Learning resuelve situaciones por sí solo a partir de un análisis de datos y cuantos más datos tengan mejores resultados, además, para realizar el análisis se utilizan algoritmos que diseñan otros datos según las necesidades. A través de los datos de entrada, Machine Learning ejecuta un algoritmo y como resultado, genera más información para el problema (Rojas, 2020).

#### 2.2.6.1. Tipos de Machine Learning

Para abordar una tarea concreta es necesario conocer algunos conceptos de aprendizaje automático, las diferentes opciones que existen, las medidas de calidad más utilizadas, etc.

**a) Aprendizaje supervisado**

Este tipo de aprendizaje automático requiere que insertes objetos ya etiquetados para que pueda aprender a hacer el trabajo, en otras palabras, los sistemas informáticos pueden generar conocimiento a partir de datos que ya han sido etiquetados, también se incluyen los datos cuyos resultados ya se conocen (Madrid, 2024).

En base a esto, la máquina entrena y reconoce patrones que aprenden a clasificar los datos recién agregados, esta es una técnica de aprendizaje automático comúnmente utilizada en herramientas con las que entramos en contacto todos los días, como detectores de spam, detectores de imágenes en captcha y otro tipo de aplicaciones (Madrid, 2024).

Este tipo de aprendizaje se basa en la denominada información de entrenamiento, el sistema se entrena proporcionando una cantidad fija de datos y definiéndolos además con etiquetas. Por ejemplo, puedes proporcionar a una computadora una foto de un perro o un gato con una etiqueta que los identifique (Tarqui, 2020, p. 30).

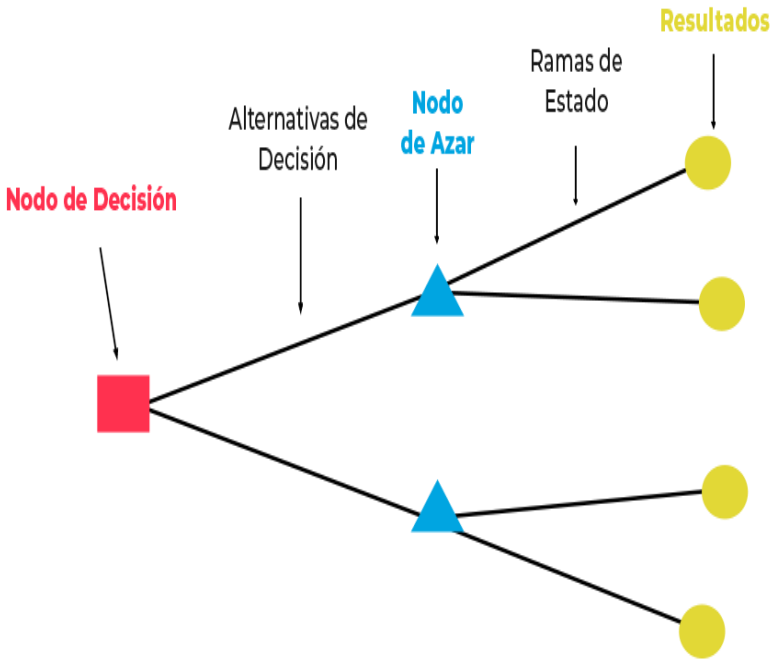
- Regresión. - Según Tarqui (2020), “una forma de desarrollar el aprendizaje automático es predecir valores continuos utilizando diferentes parámetros

que pueden predecir resultados específicos, combinados con la introducción de nuevos datos. Este método se llama regresión (p. 30).

- Árboles de decisión. - Los árboles de decisión permiten construir modelos de análisis de datos predictivos para big data basados en clasificaciones basadas en ciertas características o propiedades, o regresión sobre relaciones entre diferentes variables para obtener el valor de otra variable a predecir. Es un algoritmo estadístico o técnica de aprendizaje automático (Unir, 2024).

***Figura 2***

***Partes de un árbol de decisión***




Nota: Partes de un árbol de decisión, 2024, Fuente: Universidad Computense ¿Qué es el Machine Learning? (https://www.masterdatascienceucm.com/que-es-machine-learning/)

- Clasificación. – Según Tarqui (2020), “una vez que se proporciona una cantidad suficiente de datos, se pueden introducir nuevos datos basados en diferentes patrones registrados durante el entrenamiento sin necesidad de etiquetas. Este sistema se conoce como clasificación” (p. 30).

**b) Aprendizaje no supervisado**

El aprendizaje no supervisado, también conocido como aprendizaje automático no supervisado, utiliza algoritmos de aprendizaje automático para analizar y agrupar conjuntos de datos sin etiquetar. Estos algoritmos descubren patrones y agrupaciones de datos ocultos sin necesidad de intervención humana.

La capacidad de descubrir similitudes y diferencias en la información lo convierte en una solución ideal para el análisis exploratorio de datos, estrategias de venta cruzada, segmentación de clientes y reconocimiento de imágenes (IMB, 2024).

**c) Aprendizaje semi-supervisado**

El aprendizaje semi-supervisado es una técnica de aprendizaje automático que combina el aprendizaje supervisado y no supervisado para aprovechar conjuntos de datos que contienen pocas muestras etiquetadas y muchas muestras sin etiquetar, utiliza algoritmos de aprendizaje no supervisados para extraer características relevantes y representaciones útiles de datos sin etiquetar, y estos conocimientos se utilizan para mejorar la calidad de los modelos de aprendizaje supervisados (Gamco, 2024).

Los modelos de aprendizaje supervisado se entrenan con datos etiquetados y sin etiquetar y utilizan información de los datos sin etiquetar para mejorar la precisión del modelo. El aprendizaje semisupervisado es particularmente útil en

aplicaciones donde la recopilación de datos etiquetados es costosa o difícil, pero hay disponible una gran cantidad de datos sin etiquetar (Gamco, 2024).

Se ha demostrado que el aprendizaje semisupervisado mejora significativamente la precisión de los modelos de aprendizaje automático en aplicaciones de reconocimiento de voz, visión por computadora y procesamiento del lenguaje natural (Gamco, 2024).

**d) Aprendizaje por refuerzo**

Este aprendizaje es aún más diferente a los dos anteriores ya que forma parte de lo que se conoce como aprendizaje profundo. Su principal objetivo es crear un modelo que optimice el rendimiento en función de los resultados obtenidos previamente, para lograrlo, el sistema de aprendizaje se basa en recompensas (Madrid, 2024).

Si la máquina hace algo correctamente, recibe una recompensa (un valor positivo), y si hace algo mal, recibe una "penalización" (un valor negativo). Gracias a este modelo, la máquina puede entrenarse hasta encontrar una solución adecuada, y también puede aprender de forma inteligente una vez que empieza a tomar las decisiones "correctas" (Madrid, 2024).

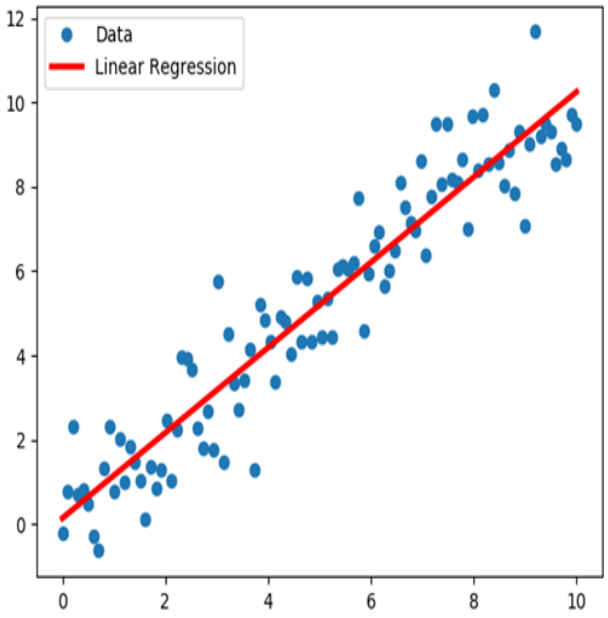
### 2.2.6.2. Algoritmos de machine Learning

**a) Regresión Lineal**

“Los modelos de regresión lineal son relativamente simples y proporcionan una fórmula matemática fácil de interpretar que puede generar predicciones. La regresión lineal se puede aplicar a varias áreas de los estudios empresariales y académicos” (IBM, s.f.)

*Figura 3*

*Gráfico de regresión*




*Nota: Grafico de regresión, 2019, Fuente: Uso de TensorFlow para realizar una regresión lineal simple por Saiteja Suvarna Hacia la ciencia de datos (https://towardsdatascience.com/linear-regression-using-tensorflow-fd732e1b690d)*

*Ecuación 1 : Ecuación de regresión lineal*

$$y = \beta\theta + \beta 1x + \varepsilon \qquad (1)$$

Donde:

- Y es la variable dependiente.
- β0 es el intercepto.
- β1 el coeficiente de la variable independiente x
- x es la variable independiente.
- ϵ es el término de error.

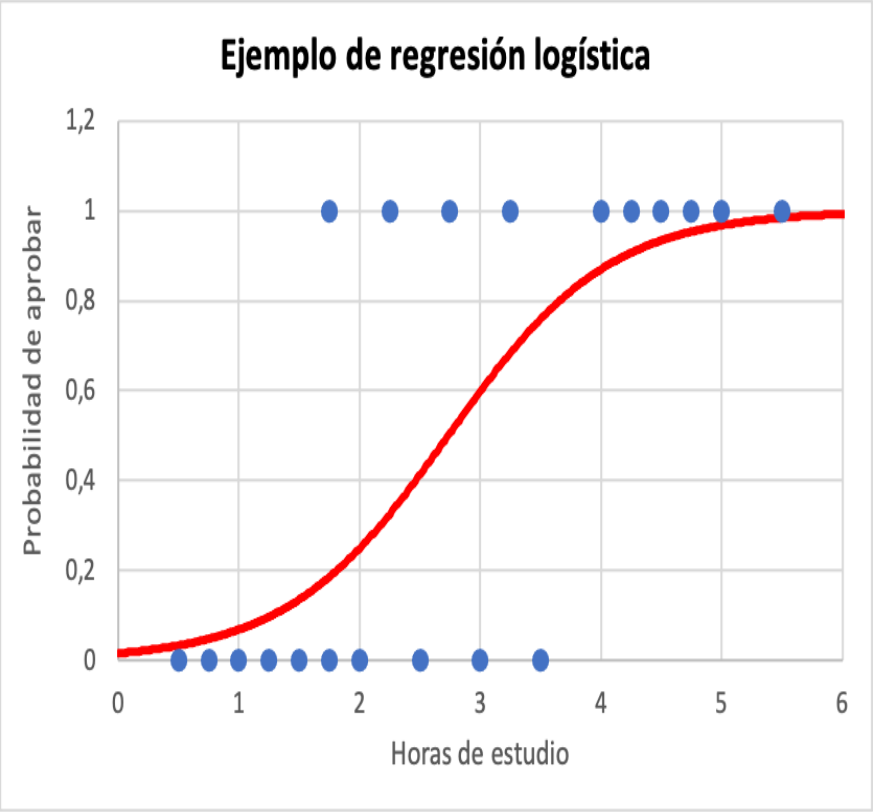
Este modelo lineal simple es fundamental en estadística y aprendizaje automático para entender la relación entre una variable independiente y una variable dependiente, y para hacer predicciones basadas en esta relación.

**b) Regresión Logística**

Es un algoritmo de aprendizaje supervisado utilizado para problemas de clasificación. Aunque se llama regresión, se usa para predecir la probabilidad de que una instancia pertenezca a una clase particular. Es especialmente útil cuando las clases son linealmente separables

*Figura 4*

*Gráfico de regresión logística*



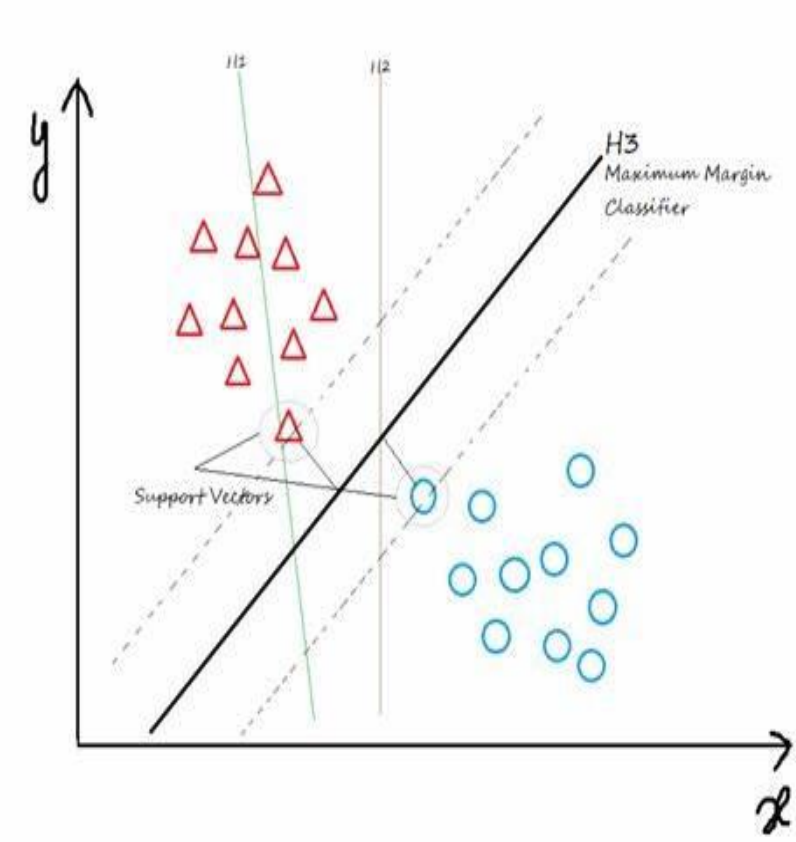

*Nota: Grafico de regresión logística, 2023, Fuente: Regresión logística (https://www.probabilidadyestadistica.net/regresion-logistica/#google_vignette)*

**c) Máquinas de Vectores de Soporte (SVM)**

También es un algoritmo de aprendizaje supervisado utilizado para clasificación y regresión. Su objetivo es encontrar el hiperplano óptimo que mejor divide un conjunto de datos en clases. Puede manejar datos no lineales mediante el uso de funciones de kernel.

***Figura 5***

*Gráfico de svm*




*Nota: Grafico de SVM, 2021 Fuente: Algoritmo SVM | Algoritmo de máquina vectorial de soporte para científicos de datos (https://www.analyticsvidhya.com/blog/2021/07/svm-support-vector-machine-algorithm/)*

**d) Árboles de decisión y bosques aleatorios**

Los árboles de decisión dividen el conjunto de datos en subconjuntos más pequeños basados en características. Los bosques aleatorios son conjuntos de múltiples árboles de decisión que se combinan para mejorar la precisión y evitar el sobreajuste.

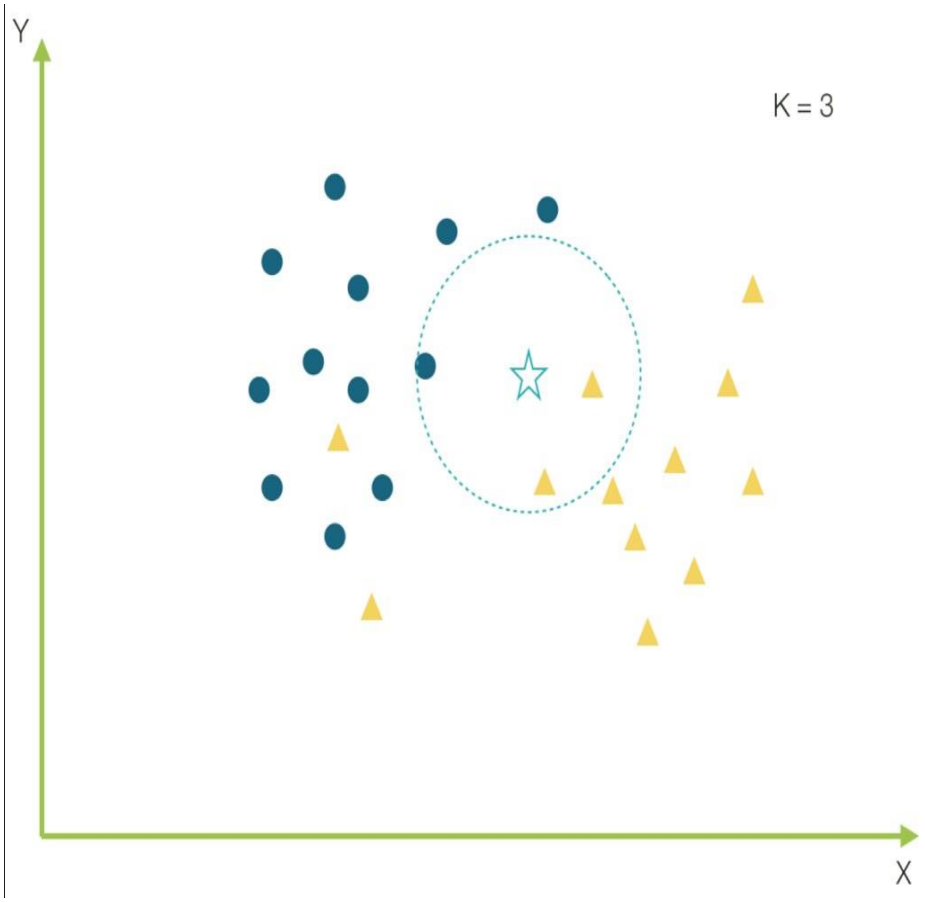
**e) Vecinos más Cercanos (k-Nearest Neighbors, k-NN)**

Es un algoritmo simple de aprendizaje supervisado utilizado tanto para clasificación como para regresión. Funciona prediciendo el valor de un punto de datos basado en los

puntos de datos vecinos más cercanos en el espacio de características. Es sensible a la elección del número de vecinos (k).

***Figura 6***

*Grafico kNN*



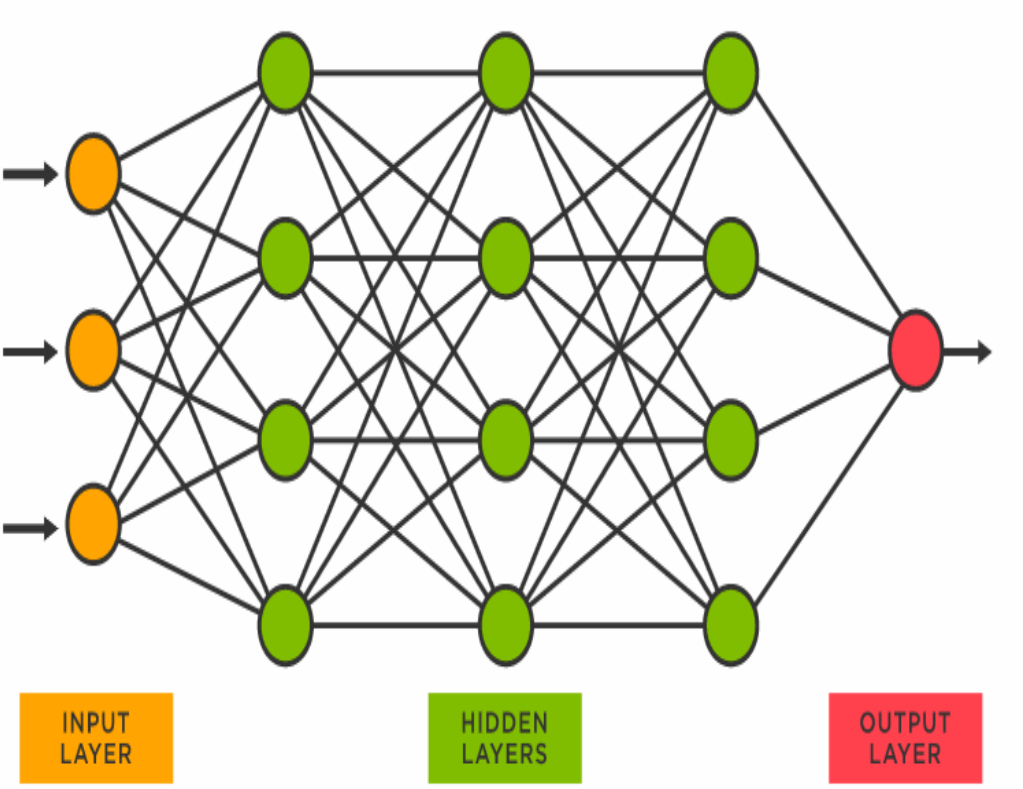

*Nota: Grafico de kNN, 2019 Fuente: K Vecinos más Cercanos - Teoría - Aprende IA(https://aprendeia.com/algoritmo-k-vecinos-mas-cercanos-teoria-machine-learning/)*

**f) Redes Neuronales Artificiales**

Inspiradas en la estructura del cerebro humano, las redes neuronales artificiales son modelos complejos de aprendizaje profundo. Consisten en capas de neuronas artificiales conectadas entre sí, con capacidad para aprender representaciones complejas de datos. Se utilizan en una amplia gama de aplicaciones, desde reconocimiento de imágenes hasta procesamiento del lenguaje natural.

***Figura 7***

*Grafico de redes neuronales*




*Nota: Grafico de redes neuronales, 2021 Fuente: Introducción a las redes neuronales [ 7 recursos de aprendizaje] (https://geekflare.com/es/neural-networks/)*

## 2.3. Detección de Vulnerabilidades

La detección de vulnerabilidades es un proceso crucial en ciberseguridad que implica identificar y evaluar las debilidades en sistemas informáticos, redes, aplicaciones y otros activos digitales que podrían ser explotadas por ciber atacantes. Estas vulnerabilidades pueden incluir fallos de seguridad en el software, configuraciones incorrectas, falta de actualizaciones de seguridad, o cualquier otro factor que pueda comprometer la integridad, confidencialidad o disponibilidad de los datos.

Hay varias técnicas y herramientas utilizadas en la detección de vulnerabilidades, entre las que se incluyen:

a) **Escaneo de vulnerabilidades.**

Utilización de herramientas automáticas para examinar sistemas en busca de debilidades conocidas. Estas herramientas pueden analizar puertos, servicios, y configuraciones en busca de vulnerabilidades comunes.

b) **Análisis estático de código.**

Revisión del código fuente de una aplicación en busca de posibles vulnerabilidades de seguridad, como problemas de inyección de SQL, XSS (cross-site scripting), o vulnerabilidades de control de acceso.

c) **Análisis dinámico de seguridad.**

Ejecución de pruebas de penetración en sistemas en tiempo de ejecución para identificar posibles puntos débiles. Esto puede incluir pruebas de inyección, fuzzing, y otras técnicas de ataque.

d) **Auditorías de seguridad.**

Revisión exhaustiva de la configuración de sistemas y redes para identificar posibles vulnerabilidades y debilidades en la seguridad.

e) **Gestión de parches.**

Mantenimiento de sistemas actualizados con los últimos parches de seguridad para mitigar vulnerabilidades conocidas.

f) **Honey pots y trampas de seguridad.**

Implementación de sistemas y servicios falsos para atraer y detectar actividades maliciosas, proporcionando información sobre posibles amenazas.

La detección de vulnerabilidades es un proceso continuo y en constante evolución debido al aumento de las amenazas cibernéticas y al desarrollo de nuevas tecnologías. Es fundamental para garantizar la seguridad de la infraestructura digital y proteger los datos sensibles contra posibles ataques.

## 2.4. WordPress

WordPress es un sistema de gestión de contenido (CMS, por sus siglas en inglés) de código abierto que se utiliza ampliamente para la creación y administración de sitios web y blogs. Es una plataforma altamente flexible y versátil que permite a los usuarios crear desde simples blogs personales hasta complejas tiendas en línea, portafolios profesionales y sitios web corporativos.

a) **Fácil de usar.**

   WordPress cuenta con una interfaz intuitiva y amigable que permite a usuarios de todos los niveles de habilidad crear y administrar contenido sin necesidad de conocimientos técnicos avanzados.

b) **Personalizable.**

   WordPress ofrece una amplia variedad de temas (themes) y plugins que permiten personalizar el diseño y la funcionalidad de un sitio web de acuerdo a las necesidades específicas del usuario.

c) **Amplia comunidad de usuarios y desarrolladores.**

   WordPress cuenta con una gran comunidad de usuarios y desarrolladores que contribuyen con temas, plugins, soporte técnico y recursos educativos, lo que facilita la resolución de problemas y la obtención de ayuda cuando sea necesario.

**d) Optimizado para SEO.**

WordPress incluye herramientas integradas para optimización de motores de búsqueda (SEO) que ayudan a mejorar la visibilidad y el posicionamiento de un sitio web en los resultados de búsqueda.

**e) Escalabilidad.**

WordPress es altamente escalable, lo que significa que puede crecer junto con las necesidades de un sitio web a medida que aumenta el tráfico y la cantidad de contenido.

**f) Seguridad.**

Si se mantienen actualizados regularmente, WordPress y sus plugins pueden ser seguros y confiables. Además, existen medidas de seguridad adicionales que se pueden implementar para proteger un sitio web contra posibles amenazas.

WordPress es una poderosa plataforma que permite a los usuarios crear y gestionar sitios web de manera eficiente y efectiva, incluso sin experiencia previa en desarrollo web. Su flexibilidad, facilidad de uso y amplia comunidad de soporte la convierten en una opción popular para una variedad de proyectos en línea.

### *2.4.1. Plugins*

Los plugins en WordPress son herramientas de software que agregan funcionalidades adicionales a tu sitio web. Estas herramientas pueden ser instaladas y activadas de manera sencilla, lo que permite a los usuarios extender las capacidades de su sitio sin necesidad de conocimientos avanzados de programación. Aquí tienes información útil sobre los plugins en WordPress:

**a) Funcionalidades adicionales.**

Los plugins pueden proporcionar una amplia variedad de funcionalidades adicionales, como formularios de contacto, galerías de imágenes, integración con redes sociales, optimización para motores de búsqueda (SEO), seguridad, tiendas en línea y mucho más. Los plugins permiten ampliar las capacidades básicas de WordPress sin necesidad de modificar el código principal.

**b) Instalación y activación.**

La mayoría de los plugins se pueden instalar directamente desde el repositorio de plugins de WordPress o mediante la carga de archivos ZIP desde tu ordenador. Una vez instalados, generalmente se activan con un solo clic desde el panel de administración de WordPress. Una vez instalados, se activan desde el panel de administración y, en muchos casos, requieren configuración adicional para adaptarse a las necesidades específicas del sitio web.

**c) Personalización y configuración.**

Después de activar un plugin, es posible que necesites configurar su funcionamiento de acuerdo a tus necesidades. Muchos plugins proporcionan opciones de configuración que puedes ajustar según tus preferencias.

**d) Actualizaciones y mantenimiento.**

Es importante mantener tus plugins actualizados para garantizar la seguridad y el rendimiento de tu sitio web. WordPress te notificará cuando haya actualizaciones disponibles para los plugins instalados, y puedes realizar las actualizaciones directamente desde el panel de administración.

**e) Compatibilidad y rendimiento.**

Al elegir plugins para tu sitio web, es importante verificar que sean compatibles con la versión de WordPress que estás utilizando y que no afecten negativamente al rendimiento de tu sitio. Además, es recomendable limitar el número de plugins instalados para evitar posibles conflictos y problemas de rendimiento.

**f) Seguridad.**

Aunque los plugins pueden agregar funcionalidades útiles a tu sitio web, también pueden introducir vulnerabilidades de seguridad si no se actualizan o mantienen adecuadamente. Es importante utilizar plugins de fuentes confiables y mantenerlos actualizados regularmente para proteger tu sitio contra posibles amenazas.

Los plugins en WordPress son una excelente manera de agregar funcionalidades adicionales a un sitio web de forma rápida y sencilla.

#### 2.4.1.1. Tipos de Plugins

Existen plugins gratuitos y otros de pago que pueden tener versiones básicas gratuitas con funcionalidades limitadas y versiones premium con características avanzadas. Hay plugins diseñados para propósitos específicos como SEO (optimización para motores de búsqueda), seguridad, rendimiento, análisis, integración con redes sociales, entre otros. Algunos usuarios desarrollan plugins personalizados para cumplir con requisitos específicos que no están cubiertos por los plugins disponibles públicamente.

**a) Plugins de Seguridad y Limpieza**

Para proteger tu web y mantenerla segura, es esencial contar con plugins que ayuden a gestionar el spam y realizar copias de seguridad. Uno de los más recomendados es Akismet, que es ligero y eficiente en la detección de spam, funcionando de manera distinta a otros plugins similares. Además, es crucial tener un plugin que permita recuperar

datos perdidos mediante copias de seguridad. Duplicator es una excelente opción para esta tarea.

**a) Plugins de Analítica y SEO para WordPress**

Mantener tu web optimizada para motores de búsqueda es fundamental. Yoast SEO es uno de los mejores plugins para SEO, ideal para principiantes. Te guía en la estructuración de contenido para hacerlo más atractivo para los usuarios y motores de búsqueda. Además, ofrece una versión premium con características avanzadas. Google Analytics es otra herramienta indispensable; permite monitorizar el tráfico de tu sitio web, proporcionando datos sobre visitantes y las palabras clave que los atraen.

**b) Plugins de Optimización de Velocidad de Carga**

La velocidad de carga es crucial para la retención de usuarios. Lazy Load es una herramienta gratuita y fácil de instalar que mejora la velocidad de tu sitio al cargar imágenes solo cuando son visibles en la pantalla. Esto es especialmente útil para sitios con muchas imágenes.

**c) Plugins de Botones de Acción, Formularios y Edición**

Para impulsar la interacción con los usuarios, necesitas plugins de formularios y botones de acción. Jetpack es una excelente opción que, aunque ofrece poca personalización, incluye todo lo necesario para que los clientes dejen sus datos correctamente. También optimiza automáticamente las imágenes.

**d) Plugins de Redes Sociales**

Las redes sociales son esenciales para difundir tu contenido. Social Metrics Tracker incluye una barra en tu WordPress que muestra las interacciones en redes sociales. SumoMe es otro plugin popular que permite añadir botones sociales configurables, soportando 18 redes sociales. Es importante configurar adecuadamente estos botones para no incomodar a los lectores.

**e) Plugins de Comercio Electrónico**

Para el comercio digital y las tiendas virtuales, existen plugins específicos que facilitan las ventas online. Estos plugins optimizan el proceso de compra y mejoran la experiencia del usuario.

**f) Plugins de Contenido**

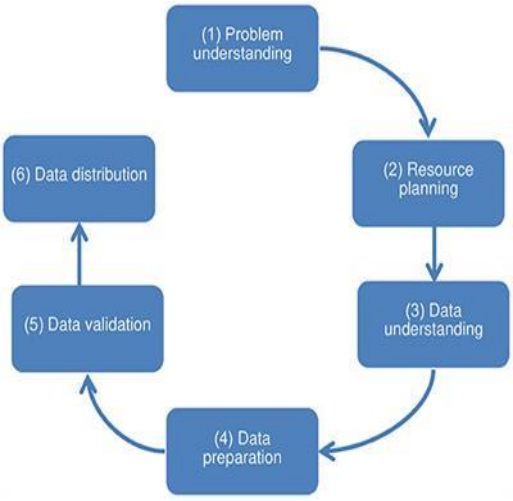
Además de buenos plugins, es crucial tener contenido de calidad. WP Popular Posts es un plugin que destaca tus artículos más populares en una barra lateral, haciendo tu contenido más atractivo para los visitantes.

## 2.5. Metodología CRISP-DM

CRISP-DM (Cross Industry Standard Process for Data Mining) es una metodología estándar utilizada en el proceso de minería de datos para guiar proyectos desde la comprensión inicial del problema hasta la implementación de la solución. Aquí tienes una descripción de las seis fases principales de la metodología CRISP-DM.

*Figura 8*

*Fases de CRISP DM*




*Nota: Fases de crisp dm, 2021 Fuente: Fronteras, Desafíos operativos en el uso de datos secundarios estructurados para la investigación en salud (https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2021.642163/full)*

- **Comprensión del Negocio (Business Understanding)**

  En esta fase, se establece una comprensión clara de los objetivos y requisitos del negocio. Se identifican los problemas a resolver y se definen los criterios de éxito del proyecto de minería de datos.

- **Comprensión de los Datos (Data Understanding)**

  En esta fase, se recopila y se familiariza con los datos disponibles. Se lleva a cabo una exploración inicial de los datos para comprender su calidad, estructura, distribución y posibles problemas.

- **Preparación de los Datos (Data Preparation)**

  En esta fase, se preparan los datos para el análisis mediante la limpieza, integración, transformación y selección de características. El objetivo es asegurar que los datos estén en el formato adecuado y sean de alta calidad para el modelado.

- **Modelado (Modeling)**

  En esta fase, se seleccionan y se aplican técnicas de modelado de datos para construir modelos que aborden los objetivos del negocio. Esto puede incluir técnicas de aprendizaje supervisado, no supervisado o por refuerzo, dependiendo del problema y los datos disponibles.

- **Evaluación (Evaluation)**

  En esta fase, se evalúan y se comparan los modelos desarrollados para determinar su calidad y su idoneidad para su implementación en el mundo real. Se ajustan y refinan los modelos según sea necesario para mejorar su rendimiento.

- **Despliegue (Deployment)**

En esta fase, se implementan los modelos en el entorno de producción y se integran en los procesos comerciales existentes. Se desarrollan planes de monitoreo y mantenimiento para garantizar que los modelos sigan siendo eficaces y se adapten a los cambios en el entorno.

CRISP-DM es un marco flexible y iterativo, lo que significa que las fases no son necesariamente lineales y pueden volver a visitarse según sea necesario a lo largo del proyecto. Esta metodología proporciona una estructura sólida para el desarrollo de proyectos de minería de datos y ayuda a garantizar que los resultados sean relevantes y accionables para el negocio.

# CAPÍTULO III

# CAPÍTULO III

## DISEÑO METODOLOGICO

### *3.1.1. Tipo de investigación*

La investigación mixta, que combina métodos cuantitativos y cualitativos, se aplica el análisis cuantitativo para evaluar y validar el modelo de inteligencia artificial. Este tipo de análisis permite recopilar y analizar grandes volúmenes de datos sobre amenazas en plugins de WordPress, lo cual es crucial para identificar patrones y frecuencias de amenazas. Además, el análisis cuantitativo proporciona las métricas necesarias para medir la precisión, sensibilidad y especificidad del modelo, así como su rendimiento en términos de velocidad de detección y capacidad para manejar grandes conjuntos de datos. Estas métricas son esenciales para garantizar que el modelo sea eficiente y fiable en la detección de amenazas.

Por otro lado, el análisis cualitativo aporta una comprensión profunda y contextualizada de las amenazas en plugins de WordPress. El estudio de casos específicos de ataques a plugins proporciona detalles sobre cómo se llevan a cabo las amenazas y cómo el modelo de inteligencia artificial puede ser optimizado para detectarlas.

Finalmente, la integración de resultados de ambos tipos de análisis permite una triangulación de datos que enriquece la comprensión del problema y la eficacia del modelo. Al combinar hallazgos cuantitativos y cualitativos, se puede obtener una visión más completa, lo que facilita el desarrollo y refinamiento del modelo de inteligencia artificial.

### *3.1.2. Diseño de la investigación*

El diseño no experimental sería adecuado ya que permite la observación y análisis de datos en su contexto natural sin manipular variables. En el caso de la detección de amenazas en plugins de WordPress, es importante analizar datos históricos y actuales

sobre amenazas y vulnerabilidades tal como ocurren. Esto proporciona una comprensión realista de los patrones y comportamientos de las amenazas, lo cual es fundamental para el desarrollo de un modelo eficaz de inteligencia artificial.

Además, el estudio se centra en un fenómeno existente las amenazas a los plugins de WordPress. No es necesario crear un experimento para generar datos sobre estas amenazas, ya que existen registros y bases de datos extensas que documentan incidentes de seguridad y vulnerabilidades en plugins. El diseño no experimental es ideal para explorar y analizar estos datos sin intervención directa, lo que facilita la identificación de patrones y tendencias relevantes.

### *3.1.3. Variables de la investigación*

Para investigar se identifican variables clave las independientes incluyen características de plugins, tipos de amenazas, algoritmos de IA utilizados y configuraciones del entorno. Las variables dependientes abarcan la eficacia del modelo en términos de detección y rendimiento, parámetros de configuración del entorno y características de los usuarios.

#### 3.1.3.1. Variable Independiente

Esta variable representa el enfoque metodológico y técnico utilizado para desarrollar un sistema capaz de detectar amenazas en plugins de WordPress. Puede incluir diferentes tipos de algoritmos de IA como los árboles de decisión, redes neuronales, máquinas de vectores soporte (SVM), entre otros. Además, abarca la configuración específica de estos modelos, como los hiperparámetros y técnicas de entrenamiento utilizadas para optimizar la detección de amenazas.

#### 3.1.3.2. Variable Dependiente

Esta variable refleja la capacidad del modelo de inteligencia artificial para identificar y clasificar amenazas potenciales o activas en los plugins utilizados en el sistema de gestión de contenidos WordPress. Las métricas asociadas incluyen la precisión en la identificación de amenazas (tasa de verdaderos positivos), así como la capacidad para minimizar falsos positivos y negativos, asegurando una detección efectiva y confiable de vulnerabilidades y comportamientos maliciosos.

### *3.1.4. Ambiente de la investigación*

El ambiente de investigación se sitúa principalmente en un entorno que busca resolver un problema práctico específico relacionado con la ciberseguridad en WordPress mediante el desarrollo y la evaluación de modelos avanzados de inteligencia artificial. El enfoque aplicado se centra en la implementación directa de conocimientos teóricos en soluciones prácticas, adaptadas a las necesidades reales de detección y mitigación de amenazas. Por otro lado, el ambiente experimental implica la realización de pruebas sistemáticas y controladas para validar la eficacia de diferentes modelos de IA, ajustar parámetros y medir métricas de rendimiento clave como la precisión y la eficiencia en la detección de amenazas. Este enfoque integral no solo busca mejorar la seguridad de los plugins de WordPress, sino también avanzar en la capacidad general de la inteligencia artificial para abordar desafíos en entornos dinámicos y vulnerables como el de los sistemas de gestión de contenidos.

### *3.1.5. Descripción de metodología del desarrollo*

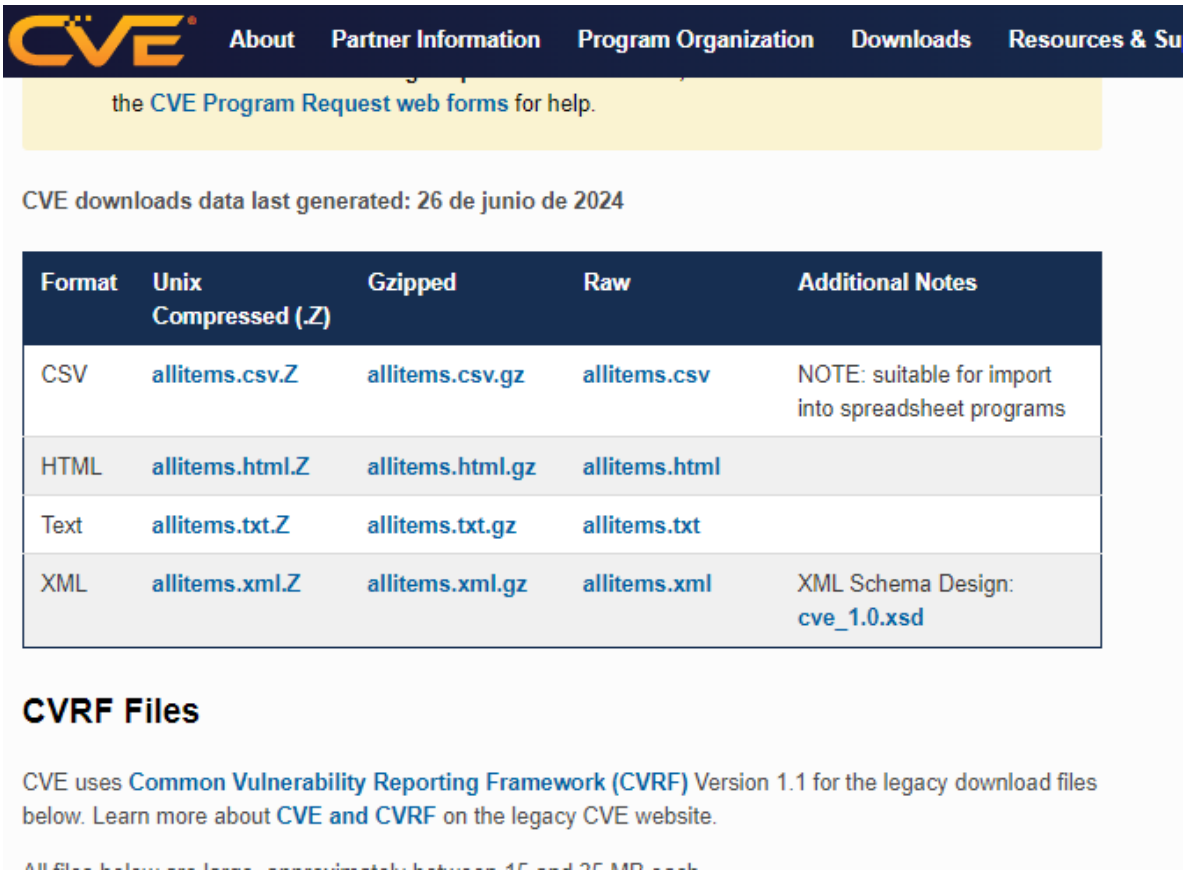
Para la construcción del sistema inteligente se determinó que metodología usar, es decir, que se estableció una guía del desarrollo del mismo, como se implementara la base

de conocimiento y el entrenamiento de la misma, principalmente y como complemento se debe elegir los lenguajes utilizados.

La base del conocimiento se construyó a partir de la información suministrada una base de datos de las vulnerabilidades existentes en la actualidad que afecta a páginas web bajo la plataforma de WordPress, como también de puertos de enlace y de la bibliografía especializada en el tema, que permitió establecer procedimientos que vinculan las variables y encadenar sus conclusiones.

### 3.1.5.1. Recolección de información

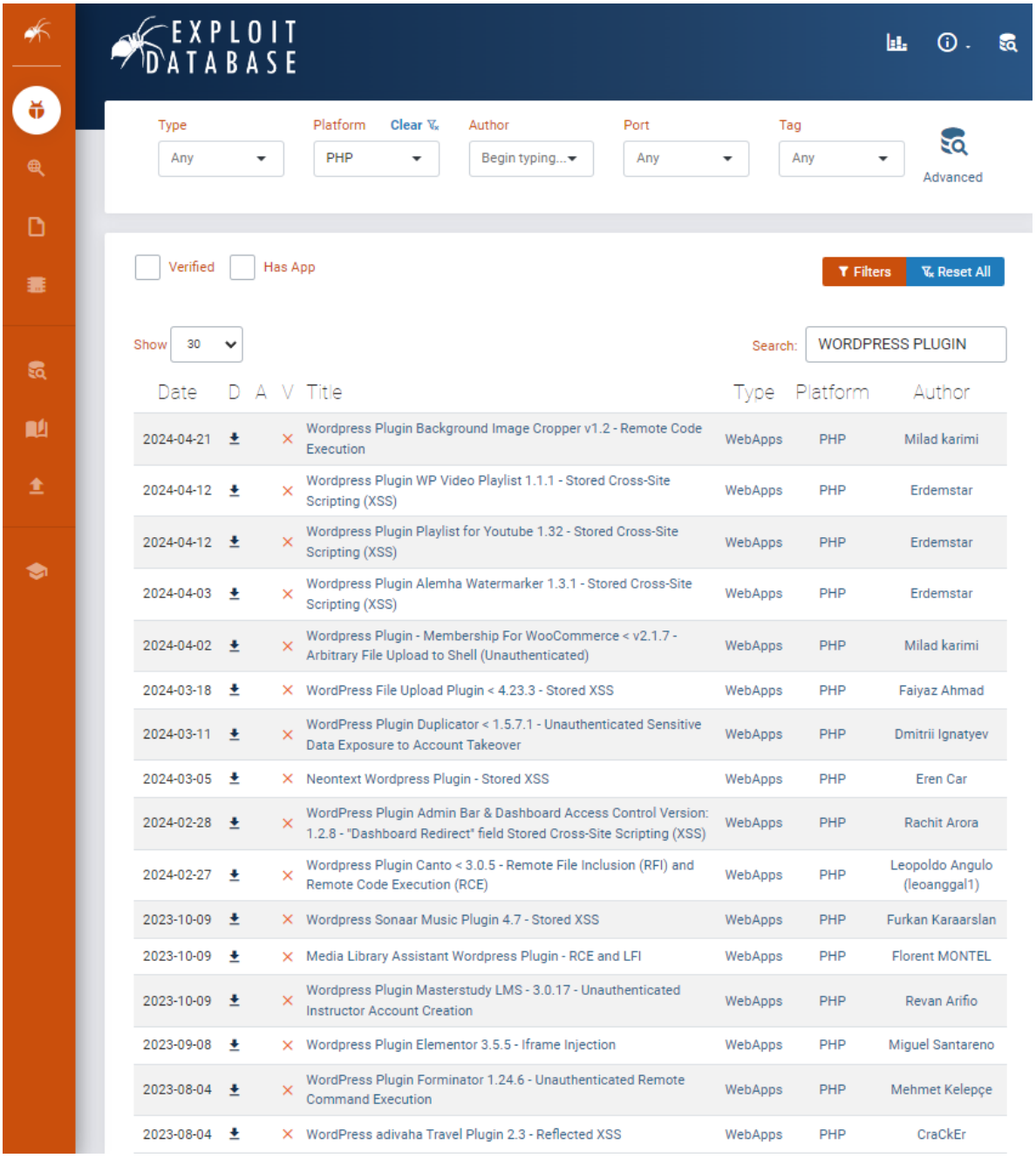
*Figura 1: Pagina donde se obtienen el reporte de vulnerabilidades encontradas*




Nota: Pagina donde se obtienen el reporte de vulnerabilidades encontradas 06,2024
Fuente: https://www.cve.org/Downloads

***Figura 2:***

***Pagina donde se puede observar los plugins vulnerables***




| Date | D | A | V | Title | Type | Platform | Author |
|---|---|---|---|---|---|---|---|
| 2024-04-21 | | | | Wordpress Plugin Background Image Cropper v1.2 - Remote Code Execution | WebApps | PHP | Milad karimi |
| 2024-04-12 | | | | Wordpress Plugin WP Video Playlist 1.1.1 - Stored Cross-Site Scripting (XSS) | WebApps | PHP | Erdemstar |
| 2024-04-12 | | | | Wordpress Plugin Playlist for Youtube 1.32 - Stored Cross-Site Scripting (XSS) | WebApps | PHP | Erdemstar |
| 2024-04-03 | | | | Wordpress Plugin Alemha Watermarker 1.3.1 - Stored Cross-Site Scripting (XSS) | WebApps | PHP | Erdemstar |
| 2024-04-02 | | | | Wordpress Plugin - Membership For WooCommerce < v2.1.7 - Arbitrary File Upload to Shell (Unauthenticated) | WebApps | PHP | Milad karimi |
| 2024-03-18 | | | | WordPress File Upload Plugin < 4.23.3 - Stored XSS | WebApps | PHP | Faiyaz Ahmad |
| 2024-03-11 | | | | WordPress Plugin Duplicator < 1.5.7.1 - Unauthenticated Sensitive Data Exposure to Account Takeover | WebApps | PHP | Dmitrii Ignatyev |
| 2024-03-05 | | | | Neontext Wordpress Plugin - Stored XSS | WebApps | PHP | Eren Car |
| 2024-02-28 | | | | WordPress Plugin Admin Bar & Dashboard Access Control Version: 1.2.8 - "Dashboard Redirect" field Stored Cross-Site Scripting (XSS) | WebApps | PHP | Rachit Arora |
| 2024-02-27 | | | | Wordpress Plugin Canto < 3.0.5 - Remote File Inclusion (RFI) and Remote Code Execution (RCE) | WebApps | PHP | Leopoldo Angulo (leoanggal1) |
| 2023-10-09 | | | | Wordpress Sonaar Music Plugin 4.7 - Stored XSS | WebApps | PHP | Furkan Karaarslan |
| 2023-10-09 | | | | Media Library Assistant Wordpress Plugin - RCE and LFI | WebApps | PHP | Florent MONTEL |
| 2023-10-09 | | | | Wordpress Plugin Masterstudy LMS - 3.0.17 - Unauthenticated Instructor Account Creation | WebApps | PHP | Revan Arifio |
| 2023-09-08 | | | | Wordpress Plugin Elementor 3.5.5 - Iframe Injection | WebApps | PHP | Miguel Santareno |
| 2023-08-04 | | | | WordPress Plugin Forminator 1.24.6 - Unauthenticated Remote Command Execution | WebApps | PHP | Mehmet Kelepçe |
| 2023-08-04 | | | | WordPress adivaha Travel Plugin 2.3 - Reflected XSS | WebApps | PHP | CraCkEr |

Nota: Pagina donde se puede observar los plugins vulnerables 06,2024
Fuente: https://www.exploit-db.com/?platform=php

**a) Identificación de activos**

En la identificación de activos, se tomaron como muestras páginas web diseñadas en WordPress utilizadas para la inspección y evaluación, además de servicios públicos en línea, para identificar los riesgos que existen en las páginas web. A continuación, se muestra en detalle siguiente cuadro.

Poner el cuadro con las webs y servicios que han sido evaluados si se tomó en cuenta.

**b) Identificación de tecnologías**

Las tecnologías y herramientas que se utilizaron para desarrollar este sistema inteligente fueron las siguientes:

WordPress, es una plataforma de gestión de contenidos, representa un software de código abierto altamente versátil que permite a los usuarios crear y desarrollar sitios web personalizados de manera sencilla, adaptable y profesional, con una muy buena experiencia de usuario.

Frameworks es un conjunto de herramientas, guías y estructuras predefinidas que se utilizan para desarrollar y organizar software de manera eficiente.

**c) Recolección de información de las páginas web**

Explicar de qué forma se recolecto la información de las páginas web y servicios que fueron parte de la evaluación para detectar las amenazas y vulnerabilidades.

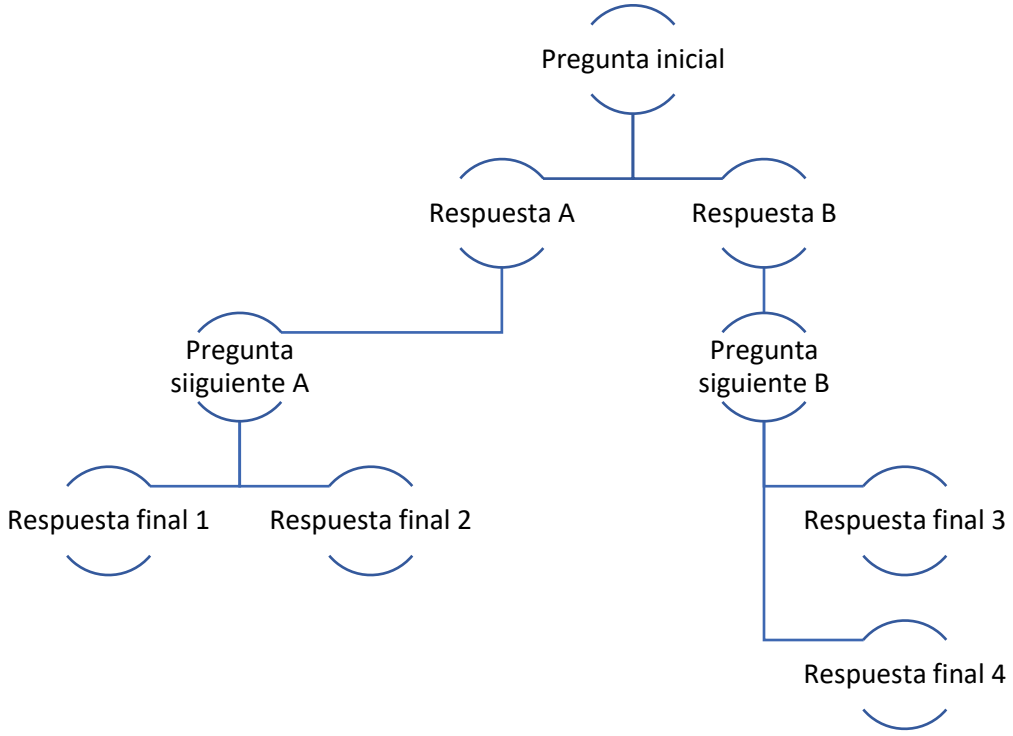
### 3.1.5.3. Análisis de la arquitectura y diseño

#### a) Esquema de arquitectura

Aquí se tiene una estructura básica de un árbol de decisión en forma de diagrama:

**Figura 9**

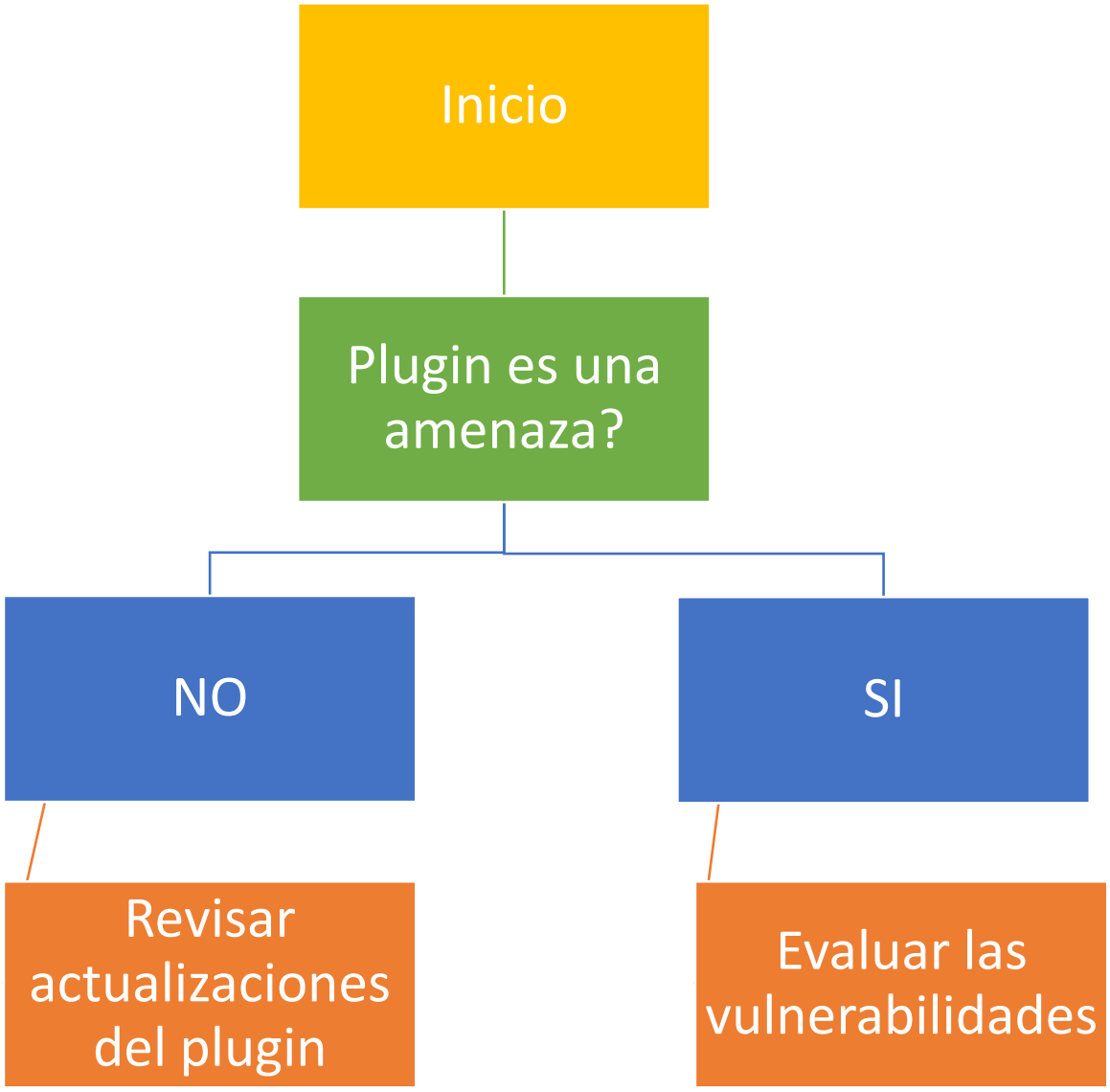
*Estructura del árbol de decisión*




Aquí se muestra un esquema básico de cómo se relaciona el algoritmo de árbol de decisión con el tema de WordPress y la detección de amenazas en plugins.

***Figura 10***

*Esquema del modelo de detección de amenazas*



### c) Análisis de flujos de datos

**Figura 11**

*Diagrama de flujo de datos*

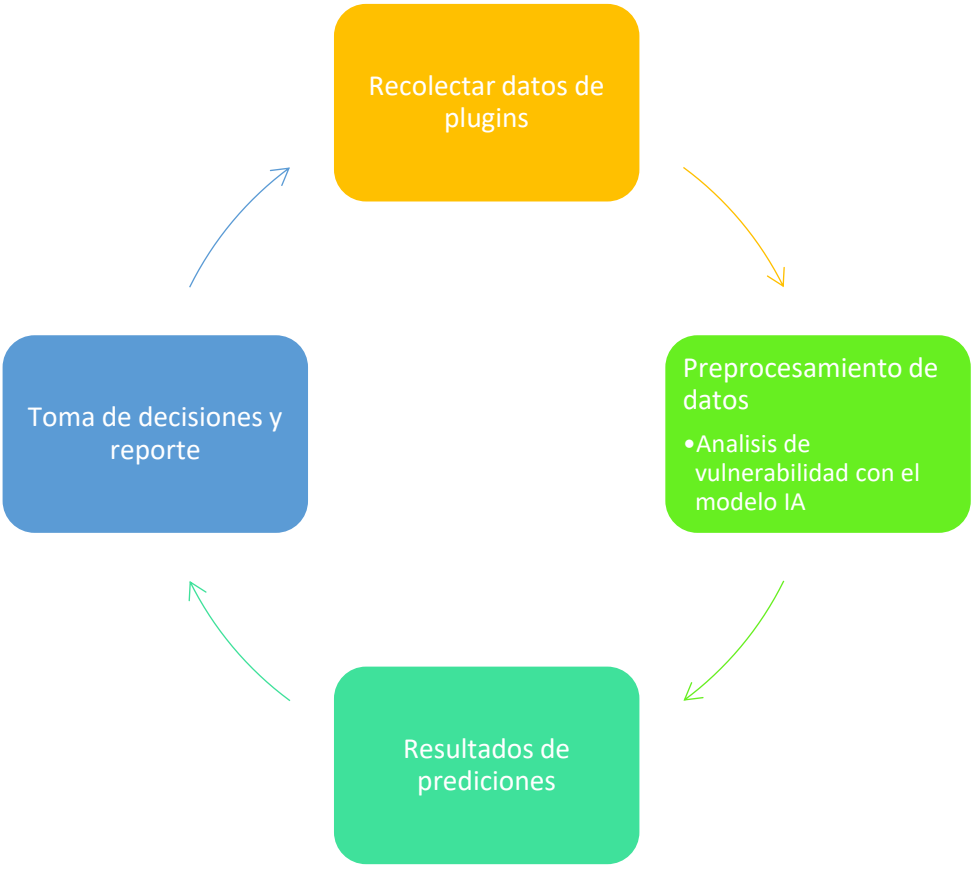



**Figura 12**

*Entrada y salida de datos*

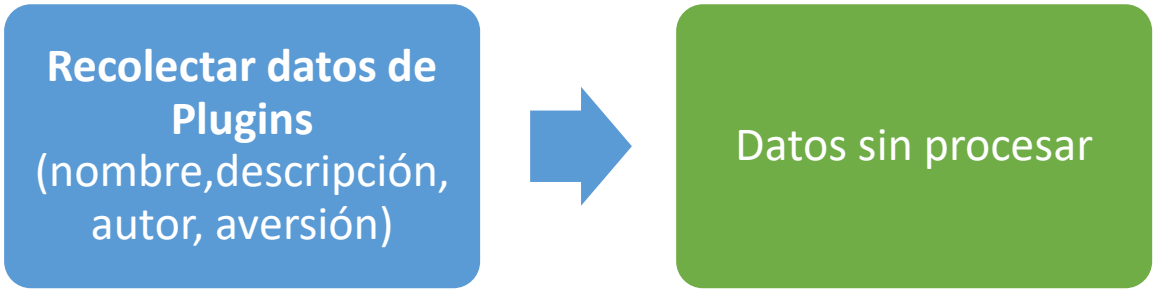


**Figura 13**

*Análisis de amenazas con el modelo*

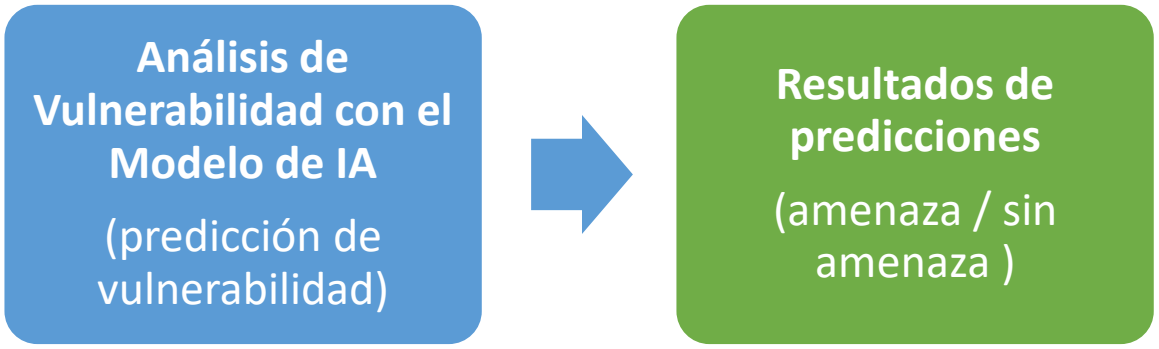



**Figura 14**

*Toma de decisiones y reporte (entrada y salida)*

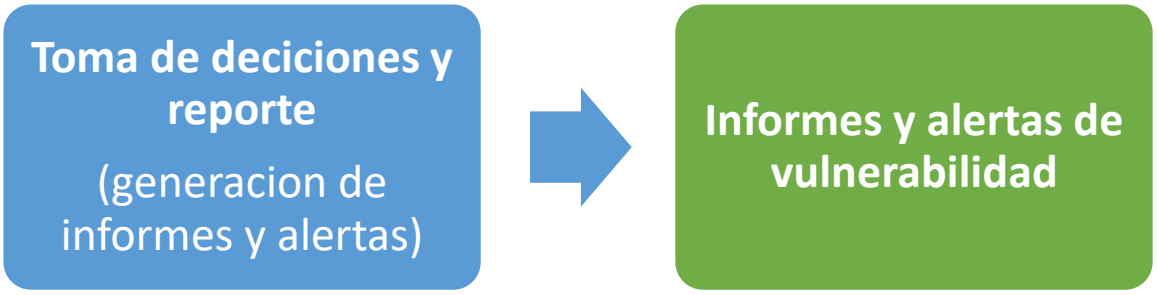
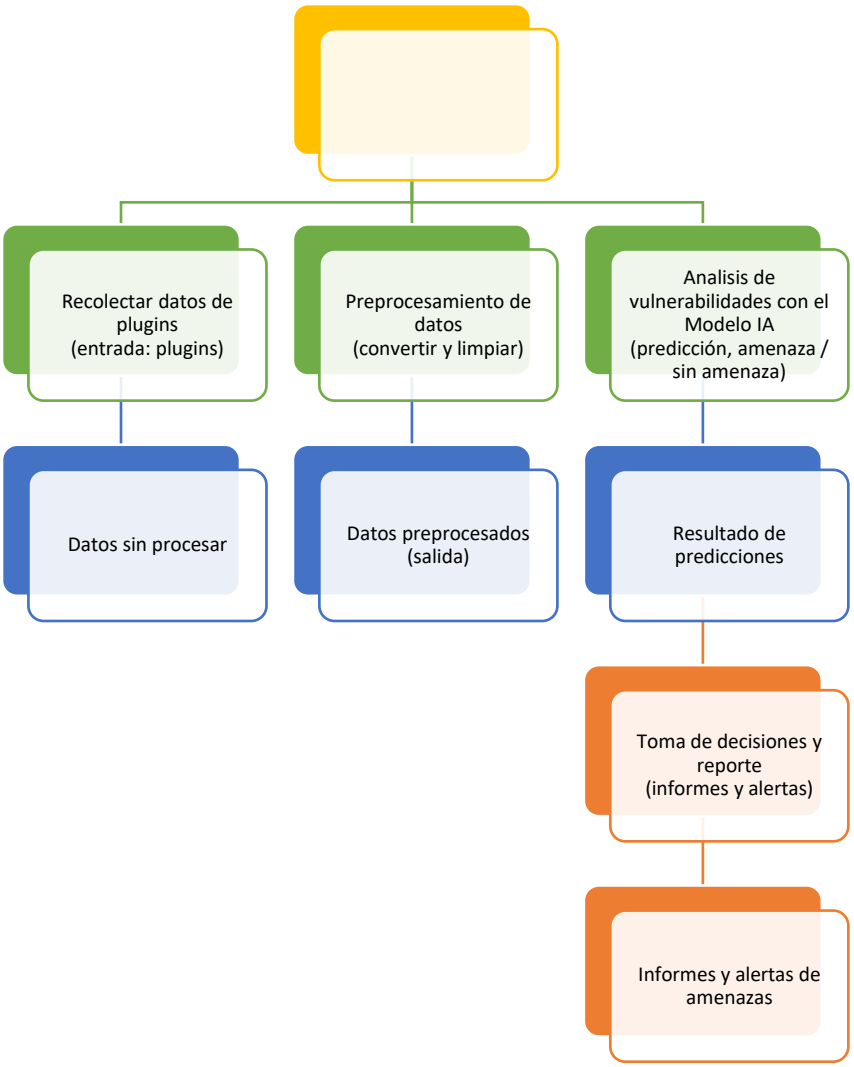


**Figura 15**

*Diagrama completo de funcionamiento del modelo*




### 3.1.5.4. Entrenamiento de la IA

#### a) Entrenamiento

Se recopilan datos de plugins etiquetados como seguros o inseguros, incluyendo su código fuente, comportamiento y metadatos. Se preprocesan los datos limpiándolos y normalizándolos y se dividen en 80% para entrenamiento y 20% para prueba. Utilizando el algoritmo y optimizando hiperparámetros como la profundidad del árbol, se entrena el modelo para identificar características que indican amenazas.

**b) Validación y ajuste**

Se realiza una validación cruzada de 10 pliegues para evaluar la estabilidad y la capacidad de generalización del modelo, utilizando métricas como precisión, recall, F1-score y AUC-ROC. Tras ajustar hiperparámetros para mejorar el rendimiento, el modelo se evaluó en datos de prueba, mostrando una precisión de 0.91, un recall de 0.86, un F1-score de 0.88 y un AUC-ROC de 0.92, demostrando un buen equilibrio y efectividad en la detección de amenazas.

### 3.1.5.5. Análisis de código y configuración

**a) Revisión de código fuente**

La auditoría del código para identificar vulnerabilidades en plugins de WordPress se llevó a cabo mediante un análisis estático. Se revisaron líneas de código buscando patrones comunes de vulnerabilidades como inyecciones SQL, XSS (Cross-Site Scripting), uso inseguro de eval() y otras funciones peligrosas. Se emplearon herramientas automatizadas de análisis estático (como SonarQube o CodeQL) para detectar posibles vulnerabilidades y se complementó con una revisión manual por expertos en seguridad para asegurar una cobertura exhaustiva.

**b) Análisis de configuraciones**

Se realizaron una revisión detallada de las configuraciones de WordPress, se buscaron configuraciones inseguras que pudieran generar vulnerabilidades, tales como permisos de archivos y directorios incorrectos y el uso de contraseñas débiles.

**c) Pruebas de penetración**

Se llevaron a cabo pruebas de penetración para identificar vulnerabilidades mediante el uso de herramientas como

Escaneo de Vulnerabilidades: Uso de WPScan para identificar vulnerabilidades conocidas en plugins y configuraciones de WordPress.

Pruebas de Inyección: Intentos de inyección SQL y XSS utilizando Burp Suite para detectar posibles puntos de entrada de ataques.

### 3.1.5.6. Análisis y reporte

**a) Documentación de vulnerabilidades**

Documentar las vulnerabilidades halladas, su grado afectación y posibles recomendaciones a tomar en cuenta. Se puede explicar en un cuadro

**b) Clasificación de vulnerabilidades**

Ordenar las vulnerabilidades según su impacto y profundidad de daño.

**c) Reporte de resultados** obtenidos

Elaboras informe de resultados obtenidos, recomendaciones y conclusiones

### 3.1.5.7. Implementación de soluciones

Que soluciones se plantean para cada una de las vulnerabilidades halladas. Se puede elaborar un cuadro.

**Tabla 1:**

*Ejemplo de vulnerabilidad y soluciones hallados*

| ***Vulnerabilidad*** | ***Problema*** | ***Solución*** |
|---|---|---|
| *Inyección SQL* | ***Entrada de usuario no filtrada permite ejecutar comandos SQL arbitrarios.*** | ***Implementar consultas SQL parametrizadas y validar/escapar todas las entradas de usuario.*** |
| *Cross-Site Scripting (XSS)* | ***Permite la inyección de scripts maliciosos en páginas web vistas por otros usuarios.*** | *Validar y escapar las entradas y salidas de datos, usar Content Security Policy (CSP).* |
| *Cross-Site Request Forgery (CSRF)* | *Información sensible se expone a usuarios no autorizados.* | *Configurar permisos adecuados, cifrar datos sensibles, y evitar exponer información innecesaria en los logs.* |

#### 3.1.5.8. Control y evaluación

a) *Monitoreo Continuo. -El monitoreo continuo es esencial para detectar cualquier intento de explotación de vulnerabilidades y para asegurarse de que las medidas de seguridad implementadas estén funcionando correctamente.*

b) *Estrategias de Implementación. – Estrategias de Implementación. - Implementar IDS para monitorear el tráfico de red y detectar actividades sospechosas. Configurar registros detallados de actividades y accesos a los plugins, revisando estos logs regularmente*

c) *Evaluación Periódica. - La evaluación periódica permite revisar y actualizar las medidas de seguridad para adaptarse a nuevas amenazas y vulnerabilidades que puedan surgir.*

d) *Actualización de Plugins. - Asegurarse de que todos los plugins estén siempre actualizados a las últimas versiones que contienen parches de seguridad.*

## 3.2. HERRAMIENTAS

### *3.2.1. HERRAMIENTAS A USAR*

Para desarrollar un modelo de inteligencia artificial para la detección de amenazas en plugins de WordPress, se puede considerar utilizar una combinación de herramientas y tecnologías que permitan recopilar datos, procesarlos, entrenar modelos y luego implementarlos de manera efectiva. Aquí te propongo algunas herramientas y tecnologías clave para cada etapa del proyecto:

#### 3.2.1.1. Python

Aplicando los principios de Python al desarrollo del modelo se aprovecha la versatilidad del lenguaje para estructurar y gestionar eficientemente la detección de

vulnerabilidades. Mediante la manipulación de datos con estructuras como listas y diccionarios, se facilita el almacenamiento y procesamiento de información crítica sobre plugins y amenazas. La programación funcional permite modularizar el código en funciones reutilizables, esenciales para tareas como la extracción de características y la evaluación de severidad de amenazas. Además, la orientación a objetos facilita la creación de clases y objetos que encapsulan métodos y atributos relacionados con la seguridad, mientras que el manejo robusto de excepciones asegura la estabilidad del modelo en un entorno dinámico como WordPress. La organización en módulos y paquetes no solo estructura el proyecto, sino que también permite la integración fluida de diferentes componentes del modelo, desde la recolección inicial de datos hasta la implementación y evaluación de algoritmos de IA.

#### 3.2.1.2. Pandas

Pandas es esencial para el desarrollo del modelo facilita la carga, limpieza y preparación de datos, incluyendo características de plugins y registros de actividad. Además, permite realizar análisis exploratorio de datos para identificar patrones relevantes y relaciones clave entre variables. Integra datos con algoritmos de aprendizaje automático, facilitando la evaluación del modelo y la comparación de predicciones con resultados reales, asegurando así una implementación efectiva del modelo de seguridad cibernética.

#### 3.2.1.3. Scikit-learn

Scikit-learn desempeña un papel crucial, esta biblioteca facilita la selección y entrenamiento de modelos de aprendizaje automático, como árboles de decisión y SVM, para identificar patrones de comportamiento malicioso en plugins. Además, ofrece herramientas para el preprocesamiento de datos, asegurando que la información esté lista para el análisis. Scikit-learn también proporciona métodos de

validación y métricas de evaluación que ayudan a medir la precisión y eficacia del modelo, junto con técnicas para ajustar hiperparámetros y mejorar su rendimiento. En conjunto, Scikit-learn se integra eficientemente con otras bibliotecas como Pandas, permitiendo una implementación efectiva y optimizada del modelo de seguridad cibernética en el entorno de WordPress.

#### 3.2.1.4. Google Colab

Google Colab es una plataforma accesible proporciona un entorno de desarrollo en la nube donde es posible ejecutar código Python, aprovechando bibliotecas como Pandas y scikit-learn sin necesidad de configuración local. Colab ofrece recursos computacionales como GPU y TPU, acelerando el entrenamiento de modelos complejos de aprendizaje automático. Además, facilita la colaboración en tiempo real y el intercambio de notebooks, lo que es beneficioso para equipos que trabajan en la implementación y ajuste de modelos de seguridad cibernética. Integrado con Google Drive, permite el almacenamiento y compartición de datos y modelos entrenados de manera eficiente. En conjunto, Colab se presenta como una herramienta integral para investigadores y desarrolladores que buscan optimizar la detección de amenazas en plugins de WordPress mediante la implementación y evaluación de modelos avanzados de IA en un entorno colaborativo y accesible desde cualquier lugar.

Estas herramientas proporcionan una base sólida para abordar cada etapa del desarrollo de un modelo de inteligencia artificial para la detección de amenazas en plugins de WordPress.

# CAPÍTULO IV

# CAPITULO IV

## 4.1. PRUEBAS Y RESULTADOS

### *4.1.1. Introducción*

Este capítulo detalla las pruebas realizadas en el prototipo del modelo predictivo. Se presentan los resultados obtenidos con diferentes algoritmos aplicados al modelo y se evalúa el desempeño en diferentes escenarios. Además, estos resultados se analizan para probar las hipótesis formuladas en este estudio y para obtener una descripción clara de la precisión del modelo.

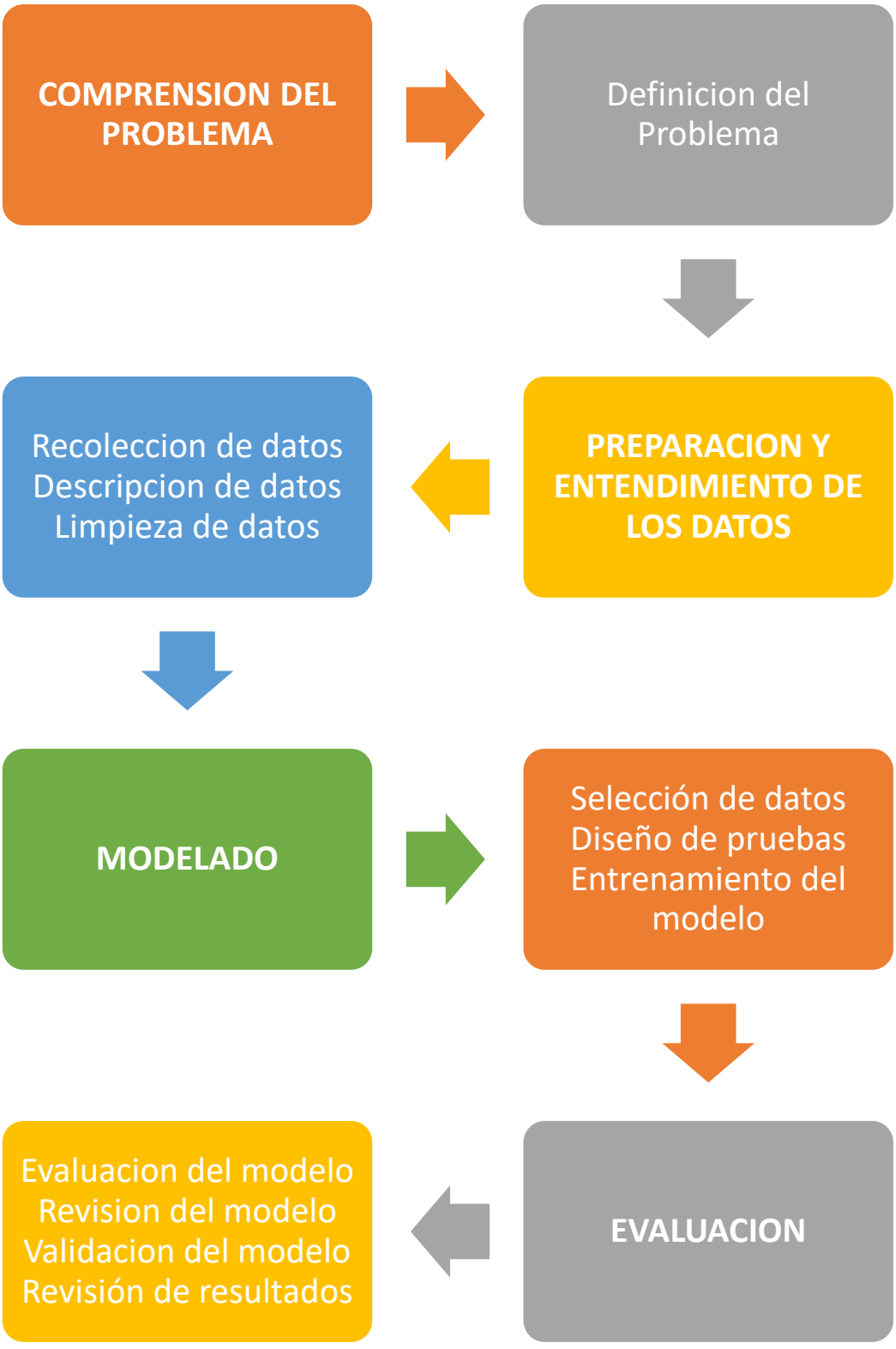
### *4.1.2. Presentación del modelo*

**Figura 11**

*Metodología CRISP DM*

**Figura 12**

*Fases del modelo*



### *4.1.3. Comprensión de los datos*

Los registros dentro de las páginas Exploit Databases y CVE, proporcionan un registro especifico de plugins que llegaron a ser vulnerables. Estos registros incluyen la información sobre el tipo de vulnerabilidad. La especificación de estos datos es esencial para identificar patrones en los tipos de ataques que se realizan y generan amenazas a lo largo del tiempo. Esta información es fundamental al momento de entrenar un modelo de Inteligencia Artificial que pueda predecir amenazas futuras. Los datos proporcionados por las páginas Exploit Databases y CVE son oficiales y verificables, lo que garantiza su precisión y legitimidad.

**Figura 13**

*Fragmento de plugins vulnerables*

| id | file | description | ate_publishe | author | type | verif |
|---|---|---|---|---|---|---|
| 28382 | exploits/php/webapps/28382.txt | WordPress Plugin WP-DB Backup 1.6/1.7 - 'edit.php' Direct | 2006-08-14 | marc & shb | webapps | 1 |
| 28295 | exploits/php/webapps/28295.txt | Joomla! Plugin JD-WordPress 2.0-1.0 RC2 - 'wp-comments | 2006-07-28 | Drago84 | webapps | 1 |
| 28296 | exploits/php/webapps/28296.txt | Joomla! Plugin JD-WordPress 2.0-1.0 RC2 - 'wp-feed.php' F | 2006-07-28 | Drago84 | webapps | 1 |
| 28297 | exploits/php/webapps/28297.txt | Joomla! Plugin JD-WordPress 2.0-1.0 RC2 - 'wp-trackback.j | 2006-07-28 | Drago84 | webapps | 1 |
| 30036 | exploits/php/webapps/30036.html | WordPress Plugin Akismet 2.1.3 - Cross-Site Scripting | 2007-05-14 | David Kierznows | webapps | 1 |
| 30403 | exploits/php/webapps/30403.txt | WordPress Plugin WP-FeedStats 2.1 - HTML Injection | 2007-07-26 | David Kierznows | webapps | 1 |
| 30637 | exploits/php/webapps/30637.js | WordPress Plugin Google FeedBurner FeedSmith 2.2 - Cro | 2007-10-04 | David Kierznows | webapps | 1 |
| 31092 | exploits/php/webapps/31092.txt | WordPress Plugin WP-Footnotes 2.2 - Multiple Remote Vul | 2008-02-02 | NBBN | webapps | 1 |
| 31230 | exploits/php/webapps/31230.txt | WordPress Plugin wp-people 2.0 - 'wp-people-popup.php' S | 2008-02-18 | S@BUN | webapps | 1 |
| 31836 | exploits/php/webapps/31836.txt | WordPress Plugin Upload File - 'wp-uploadfile.php' SQL Inje | 2008-05-24 | eserg.ru | webapps | 1 |
| 31030 | exploits/php/webapps/31030.pl | WordPress Plugin SpamBam - Key Calculation Security Byp | 2007-01-15 | Romero | webapps | 1 |
| 31029 | exploits/php/webapps/31029.pl | WordPress Plugin Peter's Math Anti-Spam 0.1.6 - Audio CA | 2008-01-15 | Romero | webapps | 1 |
| 9110 | exploits/php/webapps/9110.txt | WordPress Core / MU / Plugins - '/admin.php' Privileges Un | 2009-07-10 | Core Security | webapps | 1 |
| 17702 | exploits/php/webapps/17702.rb | WordPress Plugin Block-Spam-By-Math-Reloaded - Bypass | 2011-08-20 | Tiago Ferreira & | webapps | 0 |
| 33371 | exploits/php/webapps/33371.txt | WordPress Plugin WP-Cumulus 1.x - 'tagcloud.swf' Cross-S | 2009-11-09 | MustLive | webapps | 1 |
| 10228 | exploits/php/webapps/10228.txt | WordPress Plugin WP-Cumulus 1.20 - Full Path Disclosure | 2009-11-25 | MustLive | webapps | 1 |
| 12098 | exploits/php/webapps/12098.txt | WordPress Plugin NextGEN Gallery 1.5.1 - Cross-Site Scrip | 2010-04-06 | Alejandro Rodrig | webapps | 1 |
| 14441 | exploits/php/webapps/14441.txt | WordPress Plugin myLDlinker - SQL Injection | 2010-07-22 | H-SK33PY | webapps | 1 |
| 34946 | exploits/php/webapps/34946.txt | WordPress Plugin cformsII 11.5/13.1 - 'lib_ajax.php' Multiple | 2010-11-01 | Wagner Elias | webapps | 1 |
| 35067 | exploits/php/webapps/35067.txt | WordPress Plugin Safe Search - 'v1' Cross-Site Scripting | 2010-12-08 | John Leitch | webapps | 1 |
| 35066 | exploits/php/webapps/35066.txt | WordPress Plugin Processing Embed 0.5 - 'pluginurl' Cross | 2010-12-08 | John Leitch | webapps | 1 |
| 17814 | exploits/php/webapps/17814.txt | WordPress Plugin Event Registration 5.44 - SQL Injection | 2011-09-09 | serk | webapps | 0 |
| 34976 | exploits/php/webapps/34976.txt | WordPress Plugin Vodpod Video Gallery 3.1.5 - 'vodpod_ga | 2010-11-08 | John Leitch | webapps | 1 |
| 35261 | exploits/php/webapps/35261.txt | WordPress Plugin RSS Feed Reader 0.1 - 'rss_url' Cross-Si | 2011-01-23 | AutoSec Tools | webapps | 1 |
| 16235 | exploits/php/webapps/16235.txt | WordPress Plugin Forum Server 1.6.5 - SQL Injection | 2011-02-24 | High-Tech Bridg | webapps | 0 |
| 17119 | exploits/php/webapps/17119.txt | WordPress Plugin Custom Pages 0.5.0.1 - Local File Inclusi | 2011-04-05 | AutoSec Tools | webapps | 0 |
| 17861 | exploits/php/webapps/17861.txt | WordPress Plugin AllWebMenus 1.1.3 - Remote File Inclusi | 2011-09-19 | Ben Schmidt | webapps | 0 |
| 17602 | exploits/php/webapps/17602.txt | WordPress Plugin TimThumb 1.32 - Remote Code Executio | 2011-08-03 | MaXe | webapps | 1 |
| 17872 | exploits/php/webapps/17872.txt | Multiple WordPress Plugins - 'timthumb.php' File Upload | 2011-09-19 | Ben Schmidt | webapps | 0 |
| 36408 | exploits/php/webapps/36408.txt | WordPress Plugin Pretty Link 1.5.2 - 'pretty-bar.php' Cross- | 2011-12-06 | Amir | webapps | 1 |
| 36324 | exploits/php/webapps/36324.txt | WordPress Plugin Advanced Text Widget 2.0 - 'page' Cross | 2011-11-21 | Amir | webapps | 1 |
| 17888 | exploits/php/webapps/17888.txt | WordPress Plugin AdRotate 3.6.5 - SQL Injection | 2011-09-24 | Miroslav Stampa | webapps | 1 |
| 18114 | exploits/php/webapps/18114.txt | WordPress Plugin AdRotate 3.6.6 - SQL Injection | 2011-11-14 | Miroslav Stampa | webapps | 1 |
| 18126 | exploits/php/webapps/18126.txt | WordPress Plugin jetpack - 'sharedaddy.php' ID SQL Injecti | 2011-11-19 | longrifle0x | webapps | 0 |
| 18039 | exploits/php/webapps/18039.txt | WordPress Plugin wptouch - SQL Injection | 2011-10-27 | longrifle0x | webapps | 0 |
| 36325 | exploits/php/webapps/36325.txt | WordPress Plugin Adminimize 1.7.21 - 'page' Cross-Site Sc | 2011-11-21 | Amir | webapps | 1 |
| 36317 | exploits/php/webapps/36317.txt | WordPress Plugin Flexible Custom Post Type - 'id' Cross-Si | 2011-11-17 | Amir | webapps | 1 |

Nota: Fragmento de plugins vulnerables, 2024, Fuente: (CVE, 2024)

### *4.1.4. Preparación de datos*

Para poder realizar una mejor predicción con el modelo propuesto se hizo una selección de los datos más relevantes al momento de realizar el entrenamiento.

**Figura 14**

*Datos modificados para el entrenamiento del modelo*

| version | plugin | description | author | verified |
|---|---|---|---|---|
| 1 | drag and drop fileFile Upload | Arbitrary File Upload | Adrien Thierry | 1 |
| 1 | Front File Manager | Arbitrary File Upload | Adrien Thierry | 1 |
| 1 | Accept Signups | Cross Site Scripting | clshack | 1 |
| 1 | Accept Signups | Cross Site Scripting | clshack | 1 |
| 1 | Featured Content | Cross Site Scripting | AutoSec Tools | 1 |
| 1 | iFrame Admin Pages | Cross Site Scripting | Heine Pedersen | 1 |
| 1 | Leaflet Maps Marker | Cross Site Scripting | Heine Pedersen | 1 |
| 1 | Leaflet Maps Marker | Cross Site Scripting | Heine Pedersen | 1 |
| 1 | Placester | Cross Site Scripting | John Leitch | 1 |
| 1 | RSS Feed Reader | Cross Site Scripting | AutoSec Tools | 1 |
| 1 | Trashbin | Cross Site Scripting | MustLive | 1 |
| 1 | WP Cumulus x | Cross Site Scripting | MustLive | 1 |
| 1 | WP FaceThumb | Cross Site Scripting | d3v1l | 1 |
| 1 | ThinkIT | Multiple Vulnerabilities | Yashar shahinzadeh | 0 |
| 1 | Allow PHP in Posts and Pages RC | SQL Injection | Miroslav Stampar | 1 |
| 2 | Asset Manager | Arbitrary File Upload | Sammy FORGIT | 1 |
| 2 | Download Manager | Arbitrary File Upload | SaO | 1 |
| 2 | Daily Maui Photo Widget | Cross Site Scripting | High-Tech Bridge SA | 1 |
| 2 | oQey Gallery | Cross Site Scripting | AutoSec Tools | 1 |
| 2 | Simple Fields | Remote File Inclusion | Graeme Robinson | 0 |
| 2 | Adserve | SQL Injection | enter_the_dragon | 1 |
| 2 | Pyrmont x | SQL Injection | Gamoscu | 0 |
| 3 | oQey Headers | SQL Injection | Miroslav Stampar | 1 |
| 3 | WP Cal | SQL Injection | Houssamix | 1 |
| 3 | visitors app | Stored Cross Site Scripting | Mesut Cetin | 0 |
| 4 | Age Verification | Open Redirect | Gianluca Brindisi | 1 |
| 4 | Age Verification | Open Redirect | Gianluca Brindisi | 1 |
| 4 | Livesig | Remote File Inclusion | Ben Schmidt | 0 |
| 4 | Gallery Objects | SQL Injection | Claudio Viviani | 0 |
| 5 | Processing Embed | Cross Site Scripting | John Leitch | 1 |
| 6 | Tagregator | Cross Site Scripting | ManhNho | 0 |

Importamos las librerías necesarias para el desarrollo del modelo.

**Figura 15**

*Insertar librerías*

```
# Importar librerias
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder
import joblib
```

Se hace una lectura de los datos obtenidos de los plugins

**Figura 16**

*Lectura de datos*

```
# Cargando datos csv
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/plugins_vuln3.csv')
```

Se hace la transformación de variables categóricas a números para el entrenamiento

**Figura 17**

*Transformación de datos categóricos a numéricos*

```
# Transformacion de los atributos categóricos a numéricos
labelencoder = LabelEncoder()
df["plugin"] = labelencoder.fit_transform(df["plugin"])
df["description"] = labelencoder.fit_transform(df["description"])
df["author"] = labelencoder.fit_transform(df["author"])
df["version"] = labelencoder.fit_transform(df["version"])
```

Se realiza la selección de variables para el eje x e y

**Figura 18**

*Selección de características y etiquetas*

```
# Seleccionar características y etiquetas
X = df.drop('verified', axis=1)
y = df['verified']
```

### *4.1.5. Modelo*

La selección del modelo con los algoritmos de Inteligencia Artificial con un entrenamiento inicial se ajusta hiperparámetros para optimizar el rendimiento.

Se divide el conjunto de datos en datos de entrenamiento y datos de prueba.

**Figura 19**

*División de los datos de entrenamiento*

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

Se entrena el modelo de algoritmo árbol de decisiones

**Figura 20**

*Entrenamiento del modelo*

```
# Entrenar el modelo de árbol de decisiones
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
```

Finalmente se hace la predicción con el conjunto de datos de prueba previamente reservados

**Figura 21**

*Predicción del conjunto de prueba*

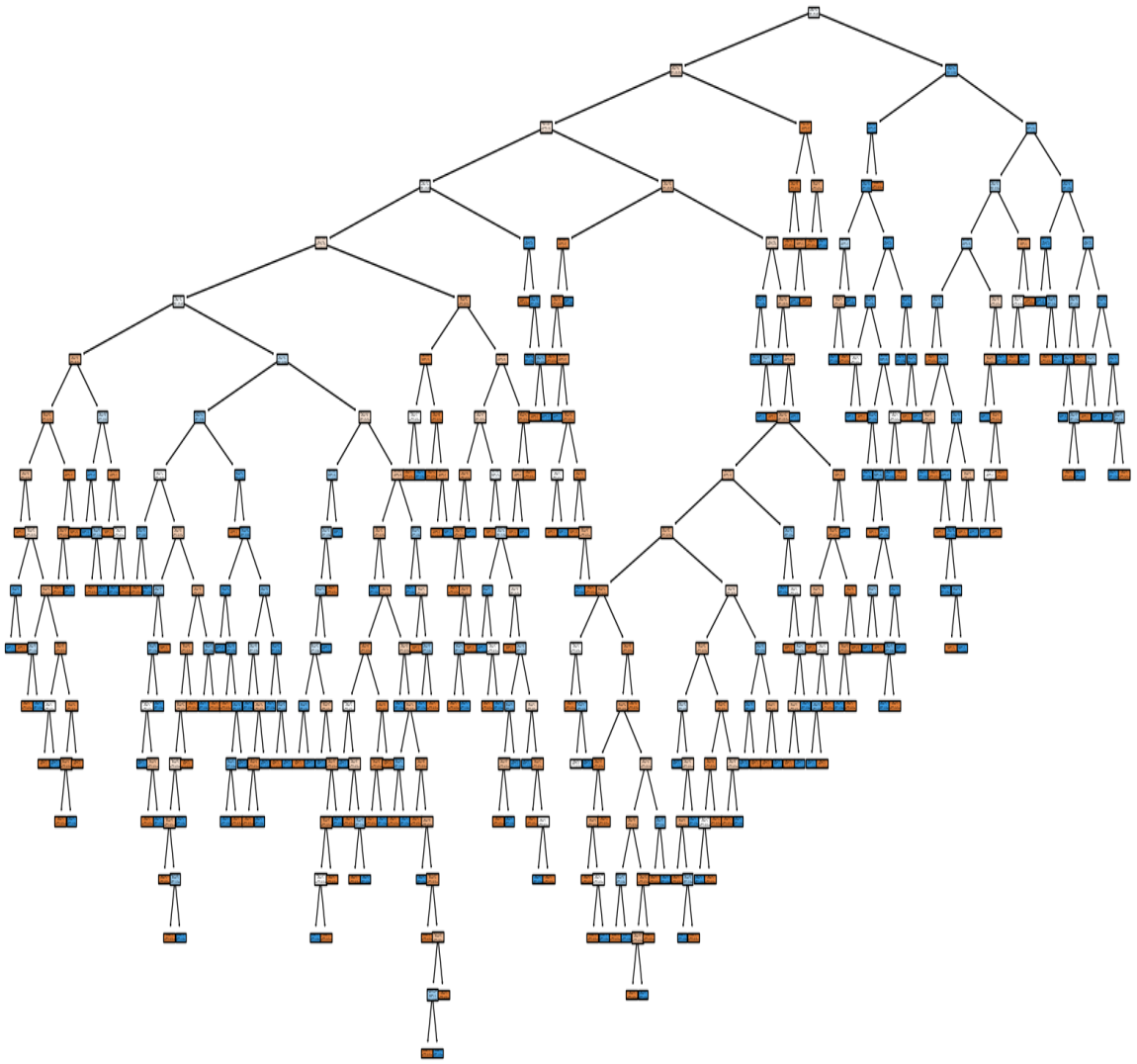
```
# Predecir en el conjunto de prueba
y_pred = model.predict(X_test)
```

### *4.1.6. Algoritmos Machine Learning*

Un árbol de decisiones está compuesto por nodos internos (decisiones), ramas (resultados de las decisiones) y nodos hoja (resultados finales o predicciones). Cada nodo interno representa una pregunta o prueba sobre una característica del plugin. Las ramas representan las posibles respuestas o resultados de esa prueba. Las hojas representan las decisiones finales amenaza o no amenaza.

*Figura 22*

*Grafica del Árbol de decisión del modelo*



### *4.1.7. Evaluación*

En el contexto del Machine Learning, el accuracy (exactitud) se refiere a la medida de qué tan preciso es un modelo en predecir la clase correcta de un conjunto de datos. Es una métrica fundamental para evaluar el rendimiento de clasificación de un modelo.

*Ecuación 2: Fórmula para calcular el accuracy*

$$Acuraci = \frac{Número\ de\ predicciones\ correctas}{Número\ total\ de\ predicciones} \quad (2)$$

**Figura 23**

*Evaluación del modelo*

```
# Evaluar el modelo
accuracy = accuracy_score(y_test, y_pred)
print(f'Precision: {accuracy}')
print('Reporte de clasificacion:')
print(classification_report(y_test, y_pred))
```

*Ecuación 3: índice Gini*

$$Gini(D) = 1 - \sum_{i=1}^{C} p_i^2 \quad (3)$$

Donde

- **pi** es la proporción de elementos en la clase **i** para el conjunto de datos D.
- **i** es Cada una de las clases
- **c** es el número de clases

*Ecuación 4: Entropía*

$$Entropia(D) = -\sum_{i=1}^{C} p_i^2 log_2(p_i) \quad (4)$$

Donde:

- pi es la probabilidad de que el evento i ocurra.
- $p_i^2$ pondera la probabilidad antes de aplicar el logaritmo, lo que puede dar mayor peso a eventos más probables.

*Ecuación 5: Ganancia de Información*

$$Ganancia(D, A) = Entropia(D) - \sum_{Valores(A)} \frac{|D_v|}{|D|} Entropia(D_V) \quad (5)$$

***Donde:***

- Entropía (D) es la entropía del conjunto de datos D.
- $\sum_{Valores(A)} \frac{|D_v|}{|D|} Entropia(D_V)$ es la suma de las entropías ponderadas de los subconjuntos de D creados al dividir D por el atributo A.
- V representa cada valor posible del atributo A.
- $|D_v|$ es el tamaño del subconjunto de D donde el atributo A tiene el valor v.
- |D| es el tamaño total del conjunto de datos D.
- $Entropia(D_V)$ es la entropía del subconjunto .

*Figura 24*

*Reporte del modelo*

Reporte de clasificacion:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.72 | 0.68 | 0.70 | 126 |
| 1 | 0.68 | 0.71 | 0.70 | 119 |
| accuracy | | | 0.70 | 245 |
| macro avg | 0.70 | 0.70 | 0.70 | 245 |
| weighted avg | 0.70 | 0.70 | 0.70 | 245 |

### *4.1.8. Desarrollo del modelo*

Esta sección de desarrollo del modelo, se detalla cómo implementar el algoritmo seleccionado (Árbol de decisiones) para predecir para predecir si un plugin genera una amenaza. Se explicará los pasos específicos de cada algoritmo.

A continuación, se muestra la secuencia de pasos que sigue cada algoritmo.

**Tabla 2**

*Estructura de pasos Árbol de decisiones*

| |
|---|
| Iniciar |
| Recopilación de datos |
| Explorar y preparar datos (limpieza y normalización) |
| Creación y división del conjunto de datos de entrenamiento |
| Ajuste del modelo de árbol de decisiones a los datos de entrenamiento |
| Evaluar el rendimiento del modelo |
| Realizar predicciones |
| Finalizar |

### *4.1.9. Demostración del prototipo*

**Figura 25**

*Extracción de plugins en sitios web*

```
import time
import requests
from bs4 import BeautifulSoup
import sys

lista_2 =[]
lista_plugins =[]

def main():
    url = input("Ingrese una url objetivo >>>> ")
    agent = {'User-Agent':'Firefox'}
    peticion = requests.get(url=url,headers=agent)
    soup = BeautifulSoup(peticion.text,'html.parser')

    for enlace in soup.find_all('link'):
        if '/wp-content/plugins' in enlace.get('href'):
            href = enlace.get('href')
            href = href.split('/')
            posicion = href.index('plugins')
            plugin = href[posicion+1]
            lista_2.append(plugin)


    archivo = open("/content/drive/MyDrive/Colab Notebooks/plugins.txt","w")
    archivo.write("version,plugin,description,author,verified\n")
    for plugin in lista_2:
        if plugin in lista_plugins:
            pass
        else:
            lista_plugins.append(plugin)
            archivo.write(","+str(plugin)+",,,\n")

    archivo.close()
```

**Figura 26**

*Listado de plugins*

```
    for plugin in lista_plugins:
        print("(+) Se encontro el plugin --->>>  {}".format(plugin))
        time.sleep(0.5)

    # Leer el archivo de texto
    txt_file_path = '/content/drive/MyDrive/Colab Notebooks/plugins.txt'
    csv_file_path = '/content/drive/MyDrive/Colab Notebooks/plugins.csv'

    # Supongamos que los datos en el archivo de texto están separados por comas
    df = pd.read_csv(txt_file_path, delimiter=',')

    # Guardar los datos en un archivo CSV
    df.to_csv(csv_file_path, index=False)

    print(f"Archivo CSV guardado en {csv_file_path}")



if __name__ == '__main__':
    try:
        main()
    except KeyboardInterrupt:
        sys.exit()
```

**Figura 27**

*Implementación del modelo mediante el método de Árbol de decisión*

```
# Importar librerias
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder
import joblib

# Cargando datos csv
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/plugins_vuln3.csv')
df.head(10)

# Transformacion de los atributos categóricos a numéricos
labelencoder = LabelEncoder()
df["plugin"] = labelencoder.fit_transform(df["plugin"])
df["description"] = labelencoder.fit_transform(df["description"])
df["author"] = labelencoder.fit_transform(df["author"])
df["version"] = labelencoder.fit_transform(df["version"])

# Seleccionar características y etiquetas
X = df.drop('verified', axis=1)
y = df['verified']

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

# Entrenar el modelo de árbol de decisiones
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Predecir en el conjunto de prueba
y_pred = model.predict(X_test)

# Evaluar el modelo
accuracy = accuracy_score(y_test, y_pred)
print(f'Precision: {accuracy}')
print('Reporte de clasificacion:')
print(classification_report(y_test, y_pred))

# Guardar el modelo entrenado
joblib.dump(model, '/content/drive/MyDrive/Colab Notebooks/decision_tree_model.joblib')
```

**Figura 28**

*Reporte de clasificación y precisión del modelo*

```
Precision: 0.7020408163265306
Reporte de clasificacion:
              precision    recall  f1-score   support

           0       0.72      0.68      0.70       126
           1       0.68      0.72      0.70       119

    accuracy                           0.70       245
   macro avg       0.70      0.70      0.70       245
weighted avg       0.70      0.70      0.70       245
```

**Figura 29**

*Implementación del modelo con lista de plugins encontrados en sitios web*

```
# Ruta al archivo del modelo entrenado
modelo_ruta = '/content/drive/MyDrive/Colab Notebooks/decision_tree_model.joblib'

# Cargar el modelo
modelo = joblib.load(modelo_ruta)

# Ruta al archivo CSV
csv_file_path = '/content/drive/MyDrive/Colab Notebooks/plugins.csv'
#df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/plugins_vuln3.csv')

#------------------------------------------------------------
data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/plugins.csv')

# Extraer los datos de la columna 'plugin'
plugin_column = data['plugin']

# Guardar los datos en un array
plugin_array = plugin_column.tolist()
```

**Figura 30**

*Continuación de implementación del modelo con lista de plugins encontrados en sitios web*

```
#print(plugin_array)

#-----------------------------------------------------------------

# Transformamos los valores de los atributos categóricos a numéricos
labelencoder = LabelEncoder() # Create an instance of LabelEncoder
df["plugin"] = labelencoder.fit_transform(df["plugin"])
df["description"] = labelencoder.fit_transform(df["description"])
df["author"] = labelencoder.fit_transform(df["author"])
df["version"] = labelencoder.fit_transform(df["version"])
# Remove the target variable from the features
X_nuevos = df.drop('verified', axis=1)

# Realiza predicciones en los nuevos datos
y_pred_nuevos = modelo.predict(X_nuevos) # Use X_nuevos for prediction

#y_pred_nuevos


# Imprimir las predicciones
for plugin, prediction in zip(plugin_array, y_pred_nuevos):
    if prediction == 1:
        print(f"El plugin {plugin} --->>Es una amenaza")
        time.sleep(0.5)
    else:
        print(f"El plugin {plugin} --->>No es una amenaza")
        time.sleep(0.5)
```

**Figura 31**

*Resultados de la demostración con árbol de decisión*

```
El plugin xpro-elementor-addons --->>No es una amenaza
El plugin elementor --->>No es una amenaza
El plugin opal-widgets-for-elementor --->>No es una amenaza
El plugin contact-form-7 --->>No es una amenaza
El plugin skt-templates --->>No es una amenaza
El plugin the-post-grid --->>No es una amenaza
El plugin wp-social --->>No es una amenaza
El plugin header-footer-elementor --->>No es una amenaza
El plugin xpro-theme-builder --->>No es una amenaza
El plugin metform --->>No es una amenaza
El plugin wp-ultimate-review --->>No es una amenaza
El plugin elementskit-lite --->>No es una amenaza
El plugin skyboot-custom-icons-for-elementor --->>No es una amenaza
El plugin royal-elementor-addons --->>No es una amenaza
```

## 4.2. PRUEBA DE LA HIPOTESIS

El análisis estadístico se realiza para verificar o refutar las suposiciones hechas durante el desarrollo del modelo. Esta sección describe las hipótesis específicas que se probarán y describe los métodos estadísticos utilizados para evaluarlas. Estos análisis nos permiten comprender mejor las relaciones entre las variables predictivas y las tasas de amenazas y validar el modelo que desarrollamos.

### *4.2.1. Pruebas estadísticas*

Las pruebas estadísticas son herramientas fundamentales en la investigación para tomar decisiones y probar hipótesis basadas en datos. El complemento de WordPress ThreatDetection utiliza pruebas estadísticas para evaluar si los modelos de inteligencia artificial (IA) mejoran significativamente la detección de amenazas en comparación con los métodos tradicionales.

La prueba t-Student se utiliza para evaluar la importancia de los coeficientes de regresión. Para valorar la significación estadística de los resultados se establece un nivel de significación (α) generalmente aceptado, α=0,05. La distribución t-Student es una distribución de probabilidad que se utiliza principalmente en el análisis estadístico para estimar la media de una población distribuida normalmente cuando el tamaño de la muestra es pequeño y se desconoce la varianza de la población.

A continuación, se muestra una definición y descripción detallada de los procedimientos de prueba estadística realizados:

**Figura 32**

*Tabla t student*

| Grados de libertad | 0.25 | 0.1 | 0.05 | 0.025 | 0.01 | 0.005 |
|---|---|---|---|---|---|---|
| 1 | 1.0000 | 3.0777 | 6.3137 | 12.7062 | 31.8210 | 63.6559 |
| 2 | 0.8165 | 1.8856 | 2.9200 | 4.3027 | 6.9645 | 9.9250 |
| 3 | 0.7649 | 1.6377 | 2.3534 | 3.1824 | 4.5407 | 5.8408 |
| 4 | 0.7407 | 1.5332 | 2.1318 | 2.7765 | 3.7469 | 4.6041 |
| 5 | 0.7267 | 1.4759 | 2.0150 | 2.5706 | 3.3649 | 4.0321 |
| 6 | 0.7176 | 1.4398 | 1.9432 | 2.4469 | 3.1427 | 3.7074 |
| 7 | 0.7111 | 1.4149 | 1.8946 | 2.3646 | 2.9979 | 3.4995 |
| 8 | 0.7064 | 1.3968 | 1.8595 | 2.3060 | 2.8965 | 3.3554 |
| 9 | 0.7027 | 1.3830 | 1.8331 | 2.2622 | 2.8214 | 3.2498 |
| 10 | 0.6998 | 1.3722 | 1.8125 | 2.2281 | 2.7638 | 3.1693 |
| 11 | 0.6974 | 1.3634 | 1.7959 | 2.2010 | 2.7181 | 3.1058 |
| 12 | 0.6955 | 1.3562 | 1.7823 | 2.1788 | 2.6810 | 3.0545 |
| 13 | 0.6938 | 1.3502 | 1.7709 | 2.1604 | 2.6503 | 3.0123 |
| 14 | 0.6924 | 1.3450 | 1.7613 | 2.1448 | 2.6245 | 2.9768 |
| 15 | 0.6912 | 1.3406 | 1.7531 | 2.1315 | 2.6025 | 2.9467 |
| 16 | 0.6901 | 1.3368 | 1.7459 | 2.1199 | 2.5835 | 2.9208 |
| 17 | 0.6892 | 1.3334 | 1.7396 | 2.1098 | 2.5669 | 2.8982 |
| 18 | 0.6884 | 1.3304 | 1.7341 | 2.1009 | 2.5524 | 2.8784 |
| 19 | 0.6876 | 1.3277 | 1.7291 | 2.0930 | 2.5395 | 2.8609 |
| 20 | 0.6870 | 1.3253 | 1.7247 | 2.0860 | 2.5280 | 2.8453 |
| 21 | 0.6864 | 1.3232 | 1.7207 | 2.0796 | 2.5176 | 2.8314 |
| 22 | 0.6858 | 1.3212 | 1.7171 | 2.0739 | 2.5083 | 2.8188 |
| 23 | 0.6853 | 1.3195 | 1.7139 | 2.0687 | 2.4999 | 2.8073 |
| 24 | 0.6848 | 1.3178 | 1.7109 | 2.0639 | 2.4922 | 2.7970 |
| 25 | 0.6844 | 1.3163 | 1.7081 | 2.0595 | 2.4851 | 2.7874 |
| 26 | 0.6840 | 1.3150 | 1.7056 | 2.0555 | 2.4786 | 2.7787 |
| 27 | 0.6837 | 1.3137 | 1.7033 | 2.0518 | 2.4727 | 2.7707 |
| 28 | 0.6834 | 1.3125 | 1.7011 | 2.0484 | 2.4671 | 2.7633 |
| 29 | 0.6830 | 1.3114 | 1.6991 | 2.0452 | 2.4620 | 2.7564 |
| 30 | 0.6828 | 1.3104 | 1.6973 | 2.0423 | 2.4573 | 2.7500 |
| 31 | 0.6825 | 1.3095 | 1.6955 | 2.0395 | 2.4528 | 2.7440 |
| 32 | 0.6822 | 1.3086 | 1.6939 | 2.0369 | 2.4487 | 2.7385 |
| 33 | 0.6820 | 1.3077 | 1.6924 | 2.0345 | 2.4448 | 2.7333 |
| 34 | 0.6818 | 1.3070 | 1.6909 | 2.0322 | 2.4411 | 2.7284 |
| 35 | 0.6816 | 1.3062 | 1.6896 | 2.0301 | 2.4377 | 2.7238 |
| 36 | 0.6814 | 1.3055 | 1.6883 | 2.0281 | 2.4345 | 2.7195 |
| 37 | 0.6812 | 1.3049 | 1.6871 | 2.0262 | 2.4314 | 2.7154 |
| 38 | 0.6810 | 1.3042 | 1.6860 | 2.0244 | 2.4286 | 2.7116 |
| 39 | 0.6808 | 1.3036 | 1.6849 | 2.0227 | 2.4258 | 2.7079 |
| 40 | 0.6807 | 1.3031 | 1.6839 | 2.0211 | 2.4233 | 2.7045 |
| 41 | 0.6805 | 1.3025 | 1.6829 | 2.0195 | 2.4208 | 2.7012 |
| 42 | 0.6804 | 1.3020 | 1.6820 | 2.0181 | 2.4185 | 2.6981 |
| 43 | 0.6802 | 1.3016 | 1.6811 | 2.0167 | 2.4163 | 2.6951 |
| 44 | 0.6801 | 1.3011 | 1.6802 | 2.0154 | 2.4141 | 2.6923 |
| 45 | 0.6800 | 1.3007 | 1.6794 | 2.0141 | 2.4121 | 2.6896 |
| 46 | 0.6799 | 1.3002 | 1.6787 | 2.0129 | 2.4102 | 2.6870 |
| 47 | 0.6797 | 1.2998 | 1.6779 | 2.0117 | 2.4083 | 2.6846 |
| 48 | 0.6796 | 1.2994 | 1.6772 | 2.0106 | 2.4066 | 2.6822 |
| 49 | 0.6795 | 1.2991 | 1.6766 | 2.0096 | 2.4049 | 2.6800 |

*Nota: Tabla t student, 2015, Fuente: Por el Departamento de Matemática – Universidad de Buenos Aires(https://cms.dm.uba.ar/academico/materias/1ercuat2015/probabilidades_y_estadistica_C/tabla_tstudent.pdf)*

**Figura 33**

*Continuación de la tabla t student*

| | | | | | | |
|---|---|---|---|---|---|---|
| 50 | 0.6794 | 1.2987 | 1.6759 | 2.0086 | 2.4033 | 2.6778 |
| 51 | 0.6793 | 1.2984 | 1.6753 | 2.0076 | 2.4017 | 2.6757 |
| 52 | 0.6792 | 1.2980 | 1.6747 | 2.0066 | 2.4002 | 2.6737 |
| 53 | 0.6791 | 1.2977 | 1.6741 | 2.0057 | 2.3988 | 2.6718 |
| 54 | 0.6791 | 1.2974 | 1.6736 | 2.0049 | 2.3974 | 2.6700 |
| 55 | 0.6790 | 1.2971 | 1.6730 | 2.0040 | 2.3961 | 2.6682 |
| 56 | 0.6789 | 1.2969 | 1.6725 | 2.0032 | 2.3948 | 2.6665 |
| 57 | 0.6788 | 1.2966 | 1.6720 | 2.0025 | 2.3936 | 2.6649 |
| 58 | 0.6787 | 1.2963 | 1.6716 | 2.0017 | 2.3924 | 2.6633 |
| 59 | 0.6787 | 1.2961 | 1.6711 | 2.0010 | 2.3912 | 2.6618 |
| 60 | 0.6786 | 1.2958 | 1.6706 | 2.0003 | 2.3901 | 2.6603 |
| 61 | 0.6785 | 1.2956 | 1.6702 | 1.9996 | 2.3890 | 2.6589 |
| 62 | 0.6785 | 1.2954 | 1.6698 | 1.9990 | 2.3880 | 2.6575 |
| 63 | 0.6784 | 1.2951 | 1.6694 | 1.9983 | 2.3870 | 2.6561 |
| 64 | 0.6783 | 1.2949 | 1.6690 | 1.9977 | 2.3860 | 2.6549 |
| 65 | 0.6783 | 1.2947 | 1.6686 | 1.9971 | 2.3851 | 2.6536 |
| 66 | 0.6782 | 1.2945 | 1.6683 | 1.9966 | 2.3842 | 2.6524 |
| 67 | 0.6782 | 1.2943 | 1.6679 | 1.9960 | 2.3833 | 2.6512 |
| 68 | 0.6781 | 1.2941 | 1.6676 | 1.9955 | 2.3824 | 2.6501 |
| 69 | 0.6781 | 1.2939 | 1.6672 | 1.9949 | 2.3816 | 2.6490 |
| 70 | 0.6780 | 1.2938 | 1.6669 | 1.9944 | 2.3808 | 2.6479 |
| 71 | 0.6780 | 1.2936 | 1.6666 | 1.9939 | 2.3800 | 2.6469 |
| 72 | 0.6779 | 1.2934 | 1.6663 | 1.9935 | 2.3793 | 2.6458 |
| 73 | 0.6779 | 1.2933 | 1.6660 | 1.9930 | 2.3785 | 2.6449 |
| 74 | 0.6778 | 1.2931 | 1.6657 | 1.9925 | 2.3778 | 2.6439 |
| 75 | 0.6778 | 1.2929 | 1.6654 | 1.9921 | 2.3771 | 2.6430 |
| 76 | 0.6777 | 1.2928 | 1.6652 | 1.9917 | 2.3764 | 2.6421 |
| 77 | 0.6777 | 1.2926 | 1.6649 | 1.9913 | 2.3758 | 2.6412 |
| 78 | 0.6776 | 1.2925 | 1.6646 | 1.9908 | 2.3751 | 2.6403 |
| 79 | 0.6776 | 1.2924 | 1.6644 | 1.9905 | 2.3745 | 2.6395 |
| 80 | 0.6776 | 1.2922 | 1.6641 | 1.9901 | 2.3739 | 2.6387 |
| 81 | 0.6775 | 1.2921 | 1.6639 | 1.9897 | 2.3733 | 2.6379 |
| 82 | 0.6775 | 1.2920 | 1.6636 | 1.9893 | 2.3727 | 2.6371 |
| 83 | 0.6775 | 1.2918 | 1.6634 | 1.9890 | 2.3721 | 2.6364 |
| 84 | 0.6774 | 1.2917 | 1.6632 | 1.9886 | 2.3716 | 2.6356 |
| 85 | 0.6774 | 1.2916 | 1.6630 | 1.9883 | 2.3710 | 2.6349 |
| 86 | 0.6774 | 1.2915 | 1.6628 | 1.9879 | 2.3705 | 2.6342 |
| 87 | 0.6773 | 1.2914 | 1.6626 | 1.9876 | 2.3700 | 2.6335 |
| 88 | 0.6773 | 1.2912 | 1.6624 | 1.9873 | 2.3695 | 2.6329 |
| 89 | 0.6773 | 1.2911 | 1.6622 | 1.9870 | 2.3690 | 2.6322 |
| 90 | 0.6772 | 1.2910 | 1.6620 | 1.9867 | 2.3685 | 2.6316 |
| 91 | 0.6772 | 1.2909 | 1.6618 | 1.9864 | 2.3680 | 2.6309 |
| 92 | 0.6772 | 1.2908 | 1.6616 | 1.9861 | 2.3676 | 2.6303 |
| 93 | 0.6771 | 1.2907 | 1.6614 | 1.9858 | 2.3671 | 2.6297 |
| 94 | 0.6771 | 1.2906 | 1.6612 | 1.9855 | 2.3667 | 2.6291 |
| 95 | 0.6771 | 1.2905 | 1.6611 | 1.9852 | 2.3662 | 2.6286 |
| 96 | 0.6771 | 1.2904 | 1.6609 | 1.9850 | 2.3658 | 2.6280 |
| 97 | 0.6770 | 1.2903 | 1.6607 | 1.9847 | 2.3654 | 2.6275 |
| 98 | 0.6770 | 1.2903 | 1.6606 | 1.9845 | 2.3650 | 2.6269 |
| 99 | 0.6770 | 1.2902 | 1.6604 | 1.9842 | 2.3646 | 2.6264 |
| 100 | 0.6770 | 1.2901 | 1.6602 | 1.9840 | 2.3642 | 2.6259 |
| ∞ | 0.6745 | 1.2816 | 1.6449 | 1.9600 | 2.3263 | 2.5758 |

*Nota: Tabla t student, 2015, Fuente: Por el Departamento de Matemática – Universidad de Buenos Aires(https://cms.dm.uba.ar/academico/materias/1ercuat2015/probabilidades_y_estadistica_C/tabla_tstudent.pdf)*

Según la tabla se determina n=14, grado de libertad será 13, entonces se determina con la tabla t-student que $t_{crit} = \pm 2.160$ para $d_f = 13\ y\ \alpha = 0.05$

Consultando una tabla t-Student o una calculadora de valores críticos t, obtenemos.

$$t_{crit} = \pm 2.160$$

*Ecuación 6: Ecuación t student*

$$t = \frac{\bar{d}}{S_d/\sqrt{n}} \quad (6)$$

**Donde:**

- $\bar{d}$ = Media de las diferencias entre las parejas de observaciones
- n = Numero de pares de datos
- $S_d$ = Desviación estándar de la diferencia

**Tabla 3**

*Recopilación de datos*

| Plugin | Método Tradicional | Modelo IA |
|---|---|---|
| xpro-elementor-addons | 5 | 7 |
| elementor | 6 | 9 |
| opal-widgets-for-elementor | 4 | 6 |
| contact-form-7 | 3 | 5 |
| skt-templates | 5 | 8 |
| the-post-grid | 2 | 4 |
| the-post-grid | 2 | 4 |
| wp-social | 4 | 6 |
| header-footer-elementor | 3 | 5 |
| xpro-theme-builder | 4 | 7 |
| metform | 6 | 9 |
| wp-ultimate-review | 5 | 8 |
| elementskit-lite | 4 | 6 |
| skyboot-custom-icons-for-elementor | 3 | 5 |
| royal-elementor-addons | 5 | 7 |

**Tabla 4**

*Cálculo de diferencias di*

| Plugin | Método Tradicional | Modelo IA | Diferencia $d_i$ |
|---|---|---|---|
| xpro-elementor-addons | 5 | 7 | 2 |
| elementor | 6 | 9 | 3 |
| opal-widgets-for-elementor | 4 | 6 | 2 |
| contact-form-7 | 3 | 5 | 2 |
| skt-templates | 5 | 8 | 3 |
| the-post-grid | 2 | 4 | 2 |
| wp-social | 4 | 6 | 2 |
| header-footer-elementor | 3 | 5 | 2 |
| xpro-theme-builder | 4 | 7 | 3 |
| metform | 6 | 9 | 3 |
| wp-ultimate-review | 5 | 8 | 3 |
| elementskit-lite | 4 | 6 | 2 |
| skyboot-custom-icons-for-elementor | 3 | 5 | 2 |
| royal-elementor-addons | 5 | 7 | 2 |

*Ecuación 7: Cálculo de diferencias* $\boldsymbol{d_i}$

$$d_i = Modelo\ IA - Método\ Tradicional \quad (7)$$

Calculamos la media $\bar{d}$ y la desviación estándar $s_d$ de las diferencias:

$$\bar{d} = \frac{\sum \mathrm{di}}{n}$$

$$\frac{2+3+2+2+3+2+2+2+3+3+3+2+2+2}{14}$$

$$\frac{34}{14} = 2.4286$$

$$s_d = \sqrt{\frac{\sum(\mathrm{di} - \mathrm{d}^-)^2}{n-1}}$$

Primero calculamos cada termino $(\mathrm{d}_i - \bar{d})^2$

**Tabla 5**

*Resultados*

$$(2-2.4286)^{\mathbf{2}} = 0.1837$$
$$(3-2.4286)^{\mathbf{2}} = 0.3265$$
$$(2-2.4286)^{\mathbf{2}} = 0.1837$$
$$(2-2.4286)^{\mathbf{2}} = 0.1837$$
$$(3-2.4286)^{\mathbf{2}} = 0.3265$$
$$(2-2.4286)^{\mathbf{2}} = 0.1837$$
$$(2-2.4286)^{\mathbf{2}} = 0.1837$$
$$(2-2.4286)^{\mathbf{2}} = 0.1837$$
$$(3-2.4286)^{\mathbf{2}} = 0.3265$$
$$(3-2.4286)^{\mathbf{2}} = 0.3265$$
$$(3-2.4286)^{\mathbf{2}} = 0.3265$$
$$(2-2.4286)^{\mathbf{2}} = 0.1837$$
$$(2-2.4286)^{\mathbf{2}} = 0.1837$$
$$(2-2.4286)^{\mathbf{2}} = 0.1837$$

***Sumando todos los términos***

$$\sum(d_i - \bar{d})^2 = 0.1837 + 0.3265 + 0.1837 + 0.1837 + 0.3265 + 0.1837 + 0.1837 + 0.1837 + 0.3265 + 0.3265 + 0.3265 + 0.1837 + 0.1837 + 0.1837 = 3.123$$

*Ecuación 8: Calculamos $s_d$*

$$s_d = \sqrt{\frac{3.123}{14-1}} \quad (8)$$

$$s_d = \sqrt{\frac{3.123}{13}}$$

$$s_d = \sqrt{0.2402} \approx 0.49$$

*Ecuación 9: Cálculo del estadístico t*

$$t = \frac{\bar{d}}{s_d/\sqrt{n}} \quad (9)$$

$$t = \frac{2.4286}{0.49/\sqrt{14}}$$

$$t = \frac{2.4286}{0.131} \approx 18.54$$

**Determinación del valor crítico y p-valor**

Para un nivel de significancia $\alpha = 0.5\ y\ d_f = n - 1 = 13$ grados de libertad, el valor crítico t para una prueba de dos colas es aproximadamente 2.160.

*Ecuación 10: Determinación t critico*

$$t_{crit} = 2.160\ (para \alpha = 0.05, doscolas) \quad (10)$$

. Dado que nuestro valor de t calculado (18.54) es mucho mayor que el valor crítico (2.160), rechazamos la hipótesis nula $H_0$

El valor de t es significativamente alto, lo que sugiere que hay una diferencia significativa entre el método tradicional y el método con el modelo de IA en la detección de amenazas en plugins de WordPress. Esto apoya la hipótesis de que el modelo de IA mejora la detección de amenazas.

La probabilidad de que el modelo de IA funcione mejor que el método tradicional es alta, dada la significancia estadística del test t-student.

# CAPÍTULO V

# CAPÍTULO V

## 5.1. CONCLUSIONES Y RECOMENDACIONES

### *5.1.1. CONCLUSIONES*

Se ha evaluado y seleccionado un conjunto de métodos efectivos para la detección de vulnerabilidades en plugins de WordPress, proporcionando una base sólida para mejorar la seguridad en este entorno crítico de desarrollo web.

La identificación detallada y sistemática de características específicas de las vulnerabilidades ha enriquecido el modelo de inteligencia artificial, mejorando su capacidad para discernir y prever potenciales amenazas en plugins de WordPress de manera más precisa y eficiente.

El desarrollo del modelo predictivo basado en inteligencia artificial ha resultado en una herramienta avanzada que no solo predice amenazas potenciales en tiempo real, sino que también ofrece una defensa proactiva contra vulnerabilidades emergentes en el ecosistema de plugins de WordPress.

La implementación del modelo predictivo ha facilitado de manera significativa la identificación y evaluación de amenazas en plugins de WordPress, proporcionando a los administradores y desarrolladores una metodología estructurada y eficaz para gestionar la seguridad de manera proactiva.

La capacidad del modelo para analizar automáticamente grandes volúmenes de datos ha mejorado sustancialmente la capacidad de detección temprana y la respuesta rápida frente a amenazas en plugins de WordPress, fortaleciendo la postura de seguridad del sistema frente a posibles ataques.

Las pruebas de validación han confirmado la alta precisión del modelo en la detección de vulnerabilidades en plugins de WordPress, validando su eficacia como herramienta fundamental para mejorar la seguridad cibernética en este entorno crítico de desarrollo web.

### *5.1.2. RECOMENDACIONES*

Para integrar modelos de IA en producción, recomendamos utilizar herramientas como Flask y FastAPI para implementar API RESTful junto con servidores web como Nginx y Apache. Es importante automatizar el proceso de análisis y predicción de vulnerabilidades en nuevos complementos cargados en el sistema. También es importante realizar pruebas exhaustivas para garantizar que el modelo funcione correctamente en una variedad de condiciones y configurar mecanismos de registro y monitoreo para detectar problemas en tiempo real. La educación y capacitación del usuario es esencial para un uso eficaz del sistema.

Además, se debe brindar capacitación a los administradores del sitio de WordPress para instruirlos en la interpretación de los resultados y la adopción de acciones correctivas basadas en las predicciones del modelo. Es importante brindar soporte continuo y recursos de capacitación, y recopilar comentarios para mejorar la experiencia del usuario.

La mejora continua del modelo requiere continuar recopilando datos sobre la detección de amenazas y su eficacia. Esto incluye comentarios de los usuarios, nuevos datos de vulnerabilidades y actualizaciones de complementos. Se recomienda que se vuelva a entrenar periódicamente el modelo utilizando nuevos datos para mantener y mejorar la precisión.

Automatizar el proceso de validación y reentrenamiento del modelo es importante para el mantenimiento. En resumen, implementar modelos de IA para mejorar la seguridad

de un sitio de WordPress requiere una integración cuidadosa en un entorno de producción, una capacitación adecuada del usuario y un enfoque continuo en mejorar los modelos. Trabajar con expertos en ciberseguridad e implementar estrategias de mantenimiento garantizará que sus modelos sigan siendo efectivos y relevantes frente a nuevas amenazas.

# BIBLIOGRAFIA

**Bibliografía**

360, I. A. (16 de 05 de 2024). *Inteligencia Artificial 360*. Obtenido de https://inteligenciaartificial360.com/glosario/razonamiento-basado-en-reglas/

Antonio Fernandez, S. M. (2006). *50 Años de la Inteligencia Artificial.* España: Unidad de Gestion Sociocultural.

Aws. (29 de 04 de 2024). *Aws Amazon*. Obtenido de https://aws.amazon.com/es/what-is/nlp/

Bobadilla, J. (2020). *Machine Learning y Deep Learning: Usando Python, Scikit y Keras.* Bogotá - México, D. F.: Ra-Ma.

CVE. (15 de 03 de 2024). *Exploit Database*. Obtenido de https://www.exploit-db.com/

Digital, I. y. (2024). *Aprendizaje Automático: Conceptos Básicos y Avanzados*. Obtenido de https://informatecdigital.com/inteligencia-artificial/aprendizaje-automatico-conceptos-basicos-y-avanzados/#31_El_Aprendizaje_Supervisado

Drew. (06 de 05 de 2024). *Drew*. Obtenido de https://blog.wearedrew.co/concepts/inteligencia-artificial/que-es-y-en-que-consiste-el-aprendizaje-automatico

Gamco. (04 de 26 de 2024). Obtenido de https://gamco.es/glosario/aprendizaje-semisupervisado/#:~:text=El%20aprendizaje%20semisupervisado%20es%20una,y%20muchos%20ejemplos%20no%20etiquetados.

Gonzales, R. L. (2017). *¿Qué sabemos de? Inteligencia Artificial.* Madrid: Catarata.

IBM. (s.f.). *¿Qué es la regresión lineal?* Obtenido de https://www.ibm.com/mx-es/topics/linear-regression

IBM. (2021). *Conceptos básicos de ayuda de CRISP-DM.* Obtenido de https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview

IBM. (05 de 06 de 2024). *IMB*. Obtenido de https://www.ibm.com/es-es/topics/unsupervised-learning#:~:text=El%20aprendizaje%20no%20supervisado%2C%20tambi%C3%A9n,necesidad%20de%20ninguna%20intervenci%C3%B3n%20humana.

Madrid, U. C. (29 de 05 de 2024). *¿Qué es el Machine Learning?* Obtenido de https://www.masterdatascienceucm.com/que-es-machine-learning/

MONROY, S. (23 de 03 de 2024). *Apd*. Obtenido de https://www.apd.es/que-es-reinforcement-learning/#:~:text=Reinforcement%20learning%20(o%20aprendizaje%20de,tomar%20una%20serie%20de%20decisiones.

Olabe, X. B. (25 de 03 de 2024). *Redes Neuronales y sus Aplicaciones*. Obtenido de https://ocw.ehu.eus/pluginfile.php/40137/mod_resource/content/1/redes_neuro/contenidos/pdf/libro-del-curso.pdf

Oracle. (19 de 05 de 2024). *Oracle*. Obtenido de https://www.oracle.com/cl/artificial-intelligence/machine-learning/what-is-machine-learning/#:~:text=el%20machine%20learning%3F-,Definici%C3%B3n%20de%20aprendizaje%20autom%C3%A1tico,de%20los%20datos%20que%20consumen.

Orts, A. C. (2019). *Ética de la Inteligencia Artificial.* Valencia: Ministerio de Ciencia, Innovación y Universidades.

*Probabilidad y Estadistica.* (s.f.). Obtenido de Tipos de estudio: https://www.probabilidadyestadistica.net/tipos-de-estudio/#estudio-observacional

RAMÍREZ, D. H. (15 de 05 de 2018). *Universidad Libre*. Obtenido de https://repository.unilibre.edu.co/handle/10901/17289

Ramírez, E. R. (30 de 05 de 2024). *cielolaboral*. Obtenido de https://www.cielolaboral.com/wp-content/uploads/2023/06/ruiz_noticias_cielo_n6_2023.pdf

Rodriguez. (2021). *Plugins de WordPress, ¿para qué sirven y cuáles son sus tipos?* Obtenido de https://citeia.com/innovaciones-en-tecnologia/wordpress/plugins-de-wordpress-para-que-se-usan-y-funciones#De_seguridad_y_limpieza

Rojas, E. M. (2020). Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo. *Revista Ibérica de Sistemas e Tecnologias de Informação*, 586-599.

Rouhiainen. (2018). *Inteligencia artificial.* Obtenido de https://archive.org/details/inteligencia-artificial-by-lasse-rouhiainen/page/n27/mode/2up

Schools, T. D. (2024). *Qué es la Inteligencia artificial (IA).* Obtenido de https://thedataschools.com/que-es/inteligencia-artificial/

Tarqui, F. R. (2020). *Detección de malware en una red con machine learning (Tesis de Grado, Universidad Mayor de San Andres).* Repositorio institucional. Obtenido de https://repositorio.umsa.bo/bitstream/handle/123456789/27851/T-3639.pdf

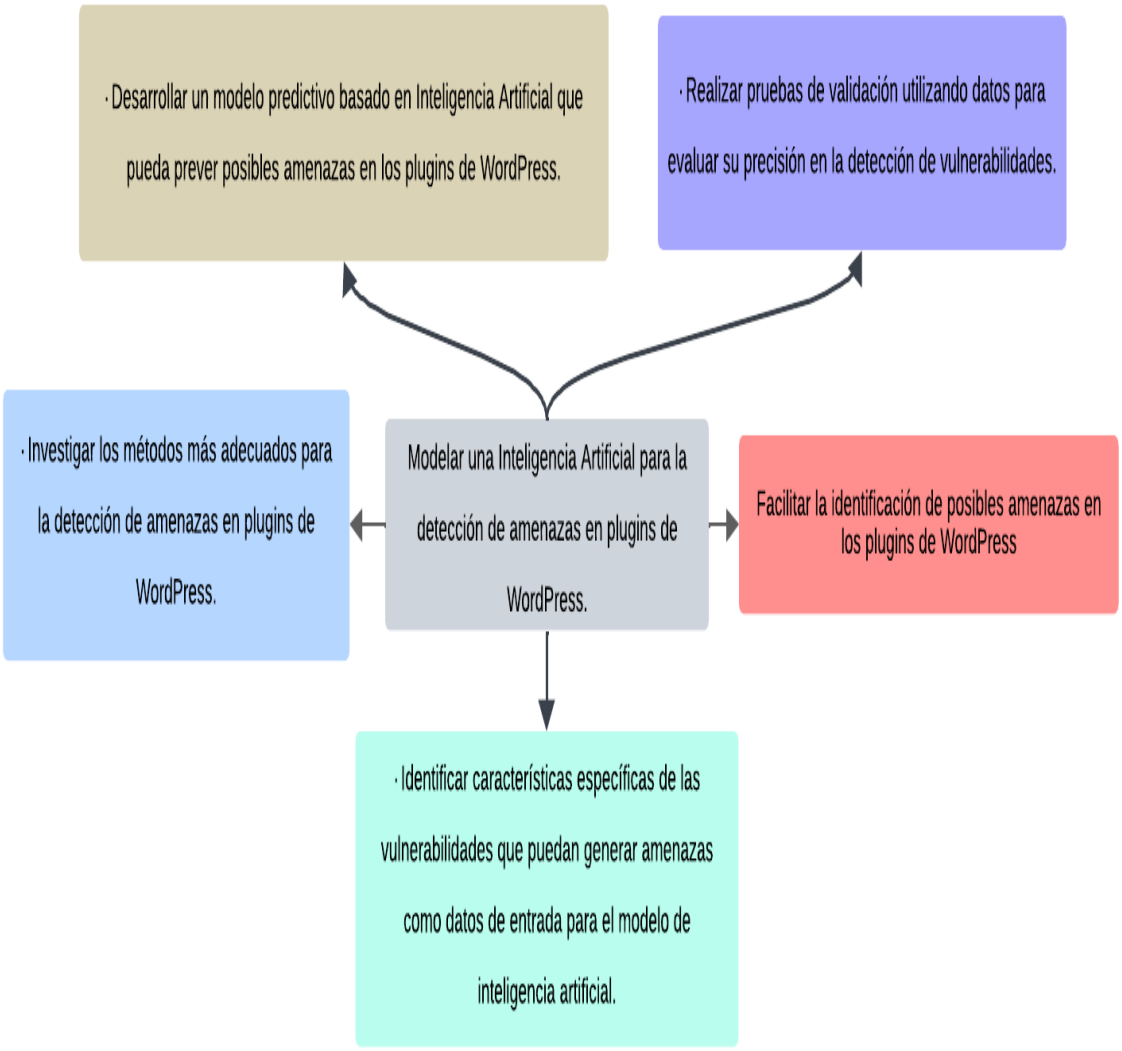
TecnoDigital. (2024). Obtenido de Aprendizaje Automático: Conceptos Básicos y Avanzados: https://informatecdigital.com/inteligencia-artificial/aprendizaje-automatico-conceptos-basicos-y-avanzados/#31_El_Aprendizaje_Supervisado

Unir. (15 de 05 de 2024). *Universidad de la Internet*. Obtenido de https://www.unir.net/ingenieria/revista/arboles-de-decision/

Zendesk. (26 de 04 de 2024). *Zendesk*. Obtenido de https://www.zendesk.com.mx/blog/tipos-de-aprendizaje-inteligencia-artificial/

# ANEXOS

**ANEXO A ARBOL DE OBJETIVOS**



## ANEXO B. LECTURA DE DATOS CON PYTHON SIN MODIFICACIONES

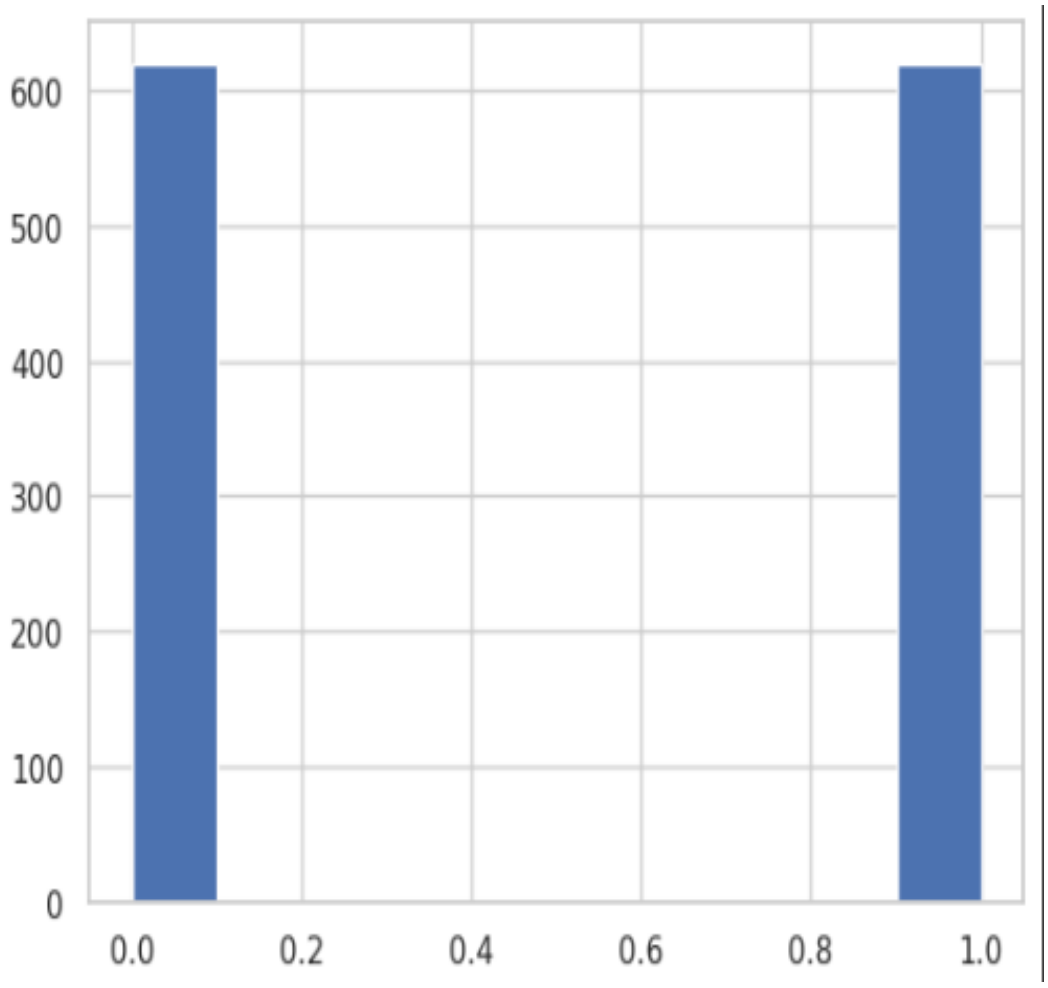
```
# Lectura del conjunto de datos mediante funciones de Python
with open("/content/drive/MyDrive/Colab Notebooks/plugins_vuln2.csv") as train_set:
    df = train_set.readlines()
df
```

```
['id,version,plugin,description,date_published,author,verified\n',
 '38209,,Gallery ,Arbitrary  Download,2013-01-10,Beni_Vanda,1\n',
 '38515,,wp FileManager ,Arbitrary  Download,2013-05-15,ByEge,1\n',
 '39291,,KenBurner Slider ,Arbitrary  Download,2014-08-24,MF0x,1\n',
 '50752,2.7.5,MasterStudy LMS,Arbitrary  Download,2022-02-18,numan türle,0\n',
 '47154,1.4.6,Hybrid Composer,Arbitrary  Download,2019-07-24,yasin,0\n',
 '50420,1.3.26,Duplicator,Arbitrary  Read,2021-10-18,nam3lum,0\n',
 '49288,1.3.26,Duplicator,Arbitrary  Read,2020-12-18,SunCSR Team,1\n',
 '49317,,W Total Cache ,Arbitrary  Read,2020-12-22,SunCSR Team,1\n',
 '38782,,SEO Watcher ,Arbitrary Code Execution,2013-10-03,wantexz,1\n',
 '38783,,Woopra Analytics ,Arbitrary Code Execution,2013-10-07,wantexz,1\n',
 '47443,3.7.1,ARforms,Arbitrary File  Deletion,2019-09-30,Ahmad Almorabea,0\n',
 '50746,1.1.1,Error Log Viewer,Arbitrary File  Deletion,2022-02-16,Ceylan BOZOĞULLARINDAN,0\n',
 '50830,1.2.13,iQ Block Country,Arbitrary File  Deletion,2022-03-21,Ceylan BOZOĞULLARINDAN,0\n',
 '44520,1.0,Woo Import Export,Arbitrary File  Deletion,2018-04-24,Lenon Leite,0\n',
 '50503,1.0.3,Backup and Restore,Arbitrary File  Deletion,2021-11-08,Murat DEMİRCİ,0\n',
 '39584,1.1.0,Image Export,Arbitrary File  Disclosure,2016-03-21,AMAR^SHG,0\n',
 '23970,,Google Document Embedder ,Arbitrary File  Disclosure,2013-01-08,Metasploit,1\n',
 '23970,,Google Document Embedder ,Arbitrary File  Disclosure,2013-01-08,Metasploit,1\n',
 '40295,1.3,CYSTEME Finder,Arbitrary File  Disclosure,2016-08-24,T0w3ntum,1\n',
 '50624,2.2.2,The True Ranker,Arbitrary File  Read,2022-01-05,Liad Levy,0\n',
 '50564,6.45,DZS Zoomsounds,Arbitrary File  Read ,2021-12-03,Uriel Yochpaz,0\n',
 '46252,1.0.11,Ad Manager WD,Arbitrary File Download,2019-01-28,41!kh4224rDz,0\n',
 '18276,,Mailing List ,Arbitrary File Download,2011-12-26,6Scan,1\n',
 '36559,1.0,aspose doc exporter,Arbitrary File Download,2015-03-30,ACC3SS,0\n',
 '35378,,DB Backup ,Arbitrary File Download,2014-11-26,Ashiyane Digital Security Team,1\n',
 '25440,,wp FileManager ,Arbitrary File Download,2013-05-14,ByEge,1\n',
 '50706,4.1.4.1,Learnpress,Arbitrary File Download,2022-02-02,Ceylan BOZOĞULLARINDAN,0\n',
 '35493,1.2,Ajax Store Locator,Arbitrary File Download,2014-12-08,Claudio Viviani,0\n',
 '36554,4.1.4,Slider REvolution,Arbitrary File Download,2015-03-30,Claudio Viviani,0\n',
 '37494,0.7,SBubble Cloud Video With Adverts & Analytics,Arbitrary File Download,2015-07-05,CrashBandicot,1\n',
 '39589,1.0.0,HB Audio Gallery Lite,Arbitrary File Download,2016-03-22,CrashBandicot,1\n',
 '36801,1.0.5,MiwoFTP,Arbitrary File Download,2015-04-21,dadou dz,0\n',
 '39593,3.1.5,Memphis Document Library,Arbitrary File Download,2016-03-22,Felipe Molina,1\n',
 '37559,1.0.5,CP Image Store with Slideshow,Arbitrary File Download,2015-07-10,i0akiN SEC-LABORATORY,1\n',
 '41568,1.0,Apptha Slider Gallery,Arbitrary File Download,2017-03-09,Ihsan Sencan,0\n',
 '41566,3.0,Mac Photo Gallery,Arbitrary File Download,2017-03-09,Ihsan Sencan,0\n',
 '36466,2.4.0,Marketplace,Arbitrary File Download,2015-03-22,Kacper Szurek,0\n',
 "37254,1.1.1,History Collection,Arbitrary File Download,2015-06-10,Kuroi'SH,0\n",
 '37754,1.0,Candidate Application Form,Arbitrary File Download,2015-08-10,Larry W. Cashdollar,0\n',
```

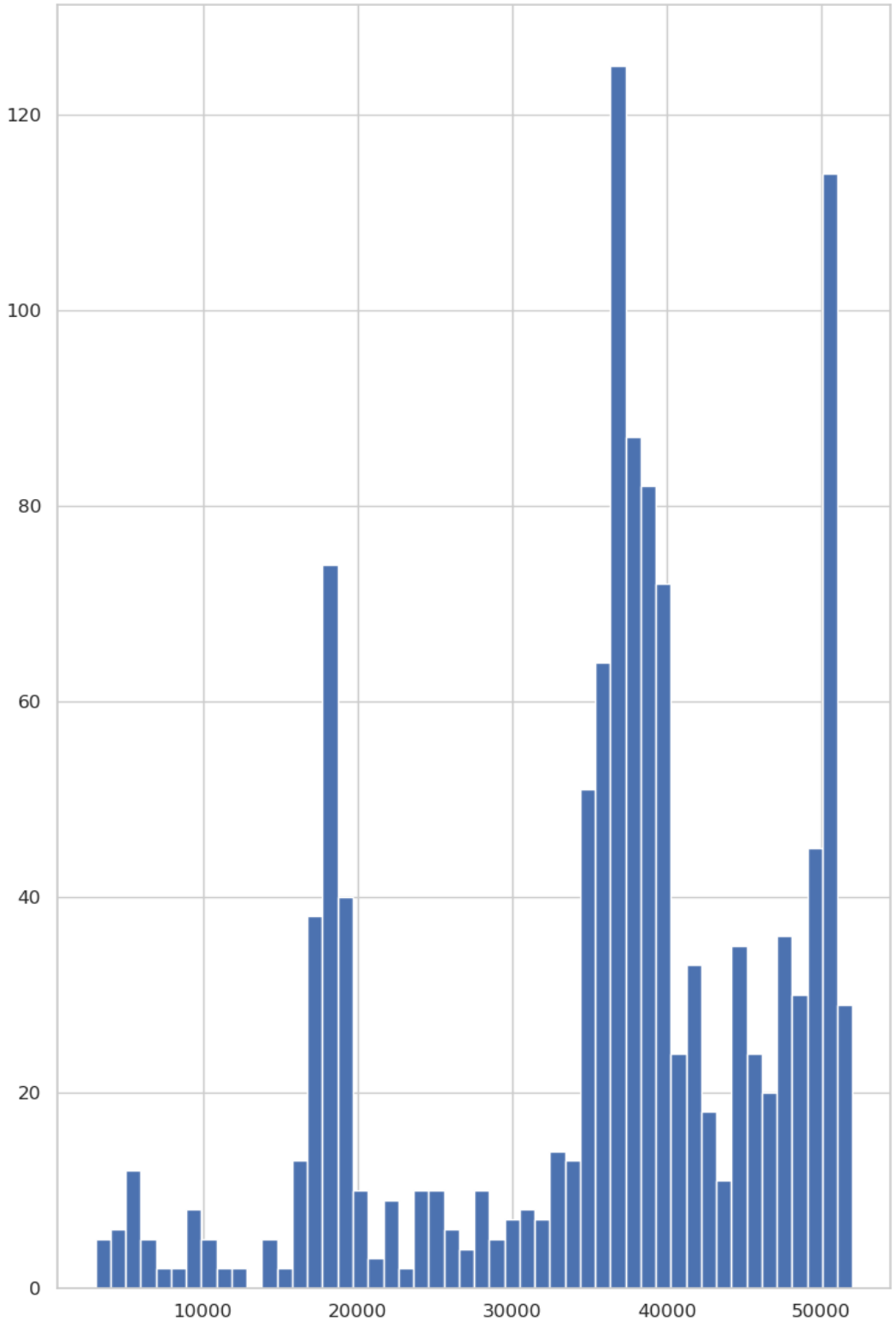
## ANEXO C. LECTURA DE DATOS EN TABLAS CON PYTHON SIN MODIFICACIONES

| | id | version | plugin | description | date_published | author | verified |
|---|---|---|---|---|---|---|---|
| 0 | 33851 | 1.07 | Multiple WordPress Plugins (TimThumbWordThumb ) | Remote Code Execution | 2014-06-24 | @u0x | 1 |
| 1 | 50299 | 5.4.3 | WooCommerce Booster Plugin | Authentication Bypass | 2021-09-17 | 0xB455 | 0 |
| 2 | 50324 | 3.1.7 | Advanced Order Export For WooCommerce | Reflected Cross Site Scripting (XSS) | 2021-09-23 | 0xB9 | 0 |
| 3 | 50344 | 1.7.14 | Contact Form | Reflected Cross Site Scripting (XSS) | 2021-09-28 | 0xB9 | 0 |
| 4 | 50703 | 1.0.2 | Contact Form Check Tester | Broken Access Control | 2022-02-02 | 0xB9 | 0 |
| 5 | 49984 | 1.2.2.6 | Database Backups | CSRF | 2021-06-11 | 0xB9 | 0 |
| 6 | 50849 | 1.6.2 | Easy Cookie Policy | Broken Access Control to Stored XSS | 2022-03-30 | 0xB9 | 0 |
| 7 | 50325 | 1.9.5 | Fitness Calculators | Cross Site Request Forgery (CSRF) | 2021-09-23 | 0xB9 | 0 |
| 8 | 50346 | 1.10.4 | Popup | Reflected Cross Site Scripting (XSS) | 2021-09-28 | 0xB9 | 0 |
| 9 | 50705 | 2.1.1 | Post Grid | Cross Site Scripting (XSS) | 2022-02-02 | 0xB9 | 0 |

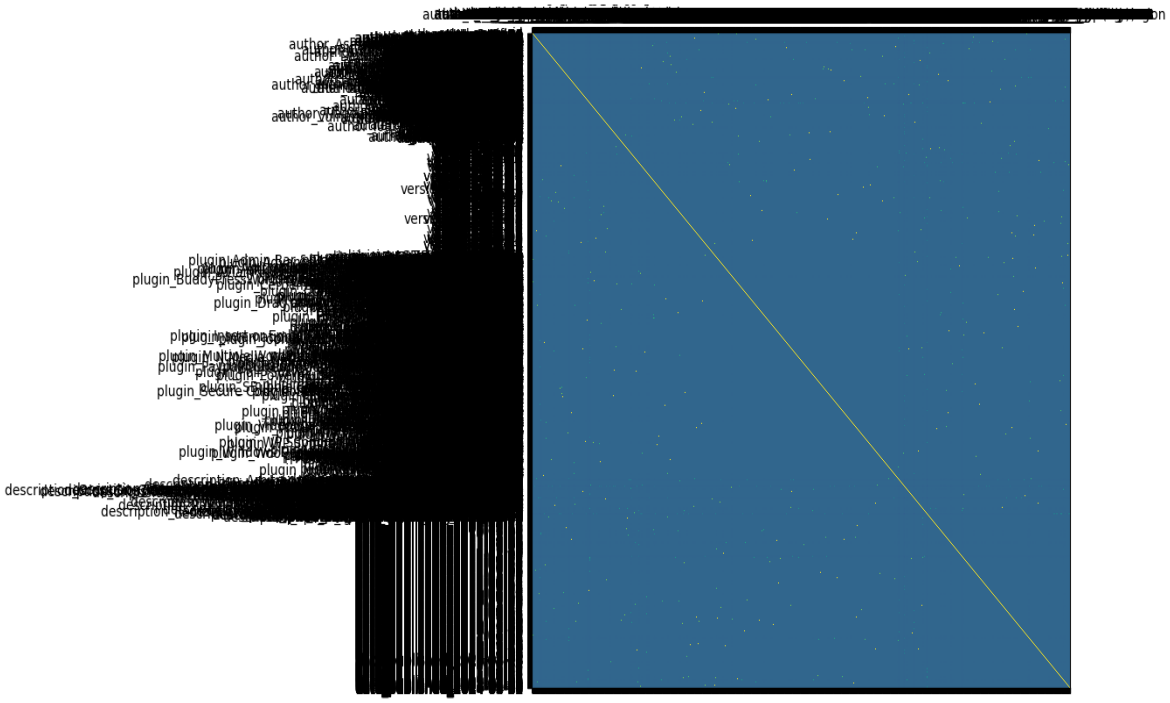
**ANEXO D. INFORMACION DE PLUGINS VULNERABLES VERIFICADOS Y SIN VERIFICAR**



**ANEXO E. GRAFICO EN BARRAS DE LAS VARIABLES CATEGORICAS**



## ANEXO F. PRUEBA DE CORRELACION DE VARIABLES CON EL MODELO DE REGRESION LOGISTICA



## ANEXO G. TRANSFORMACION DEL DATAFRAME EN PANDAS

| | id | version_0.0.1 | version_0.1 | version_0.1.0 | version_0.1.13 | version_0.1.5 | version_0.1.6 | version_0.10 |
|---|---|---|---|---|---|---|---|---|
| 307 | 0.437579 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 481 | -0.158253 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 969 | 0.719639 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 593 | -0.936507 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 346 | -2.405647 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 334 | 0.196689 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 285 | 0.795033 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1019 | 0.925272 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 97 | -0.088403 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 522 | 0.181314 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

10 rows × 3013 columns

## ANEXO H. COMPARACION DE DATOS ENTRENADOS

```
# Comparamos resultados entre escalado y sin escalar
evaluate_result(y_pred, y_val, y_prep_pred, y_val, f1_score)
```

```
f1_score WITHOUT preparation: 0.806527316314085
f1_score WITH preparation: 0.806527316314085
```