ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

EVALUACIÓN DE ALGORITMOS DE MINERÍA DE DATOS PARA DETECCIÓN Y PREDICCIÓN DE ATAQUES DE INYECCIÓN SQL EN BIG DATA

EVALUACIÓN DE UNA RED NEURONAL RECURRENTE PARA LA DETECCIÓN Y PREDICCIÓN DE ATAQUES DE INYECCIÓN SQL EN BIG DATA

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SOFTWARE

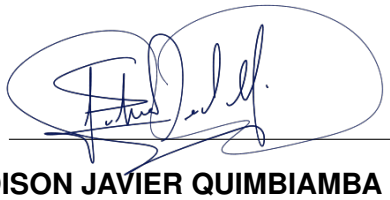
EDISON JAVIER QUIMBIAMBA GUASGUA
edison.quimbiamba@epn.edu.ec

DIRECTORA: PhD. GABRIELA LORENA SUNTAXI OÑA
gabriela.suntaxi@epn.edu.ec

DMQ, octubre 2022

CERTIFICACIONES

Yo, Edison Javier Quimbiamba Guasgua , declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



EDISON JAVIER QUIMBIAMBA GUASGUA

Certifico que el presente trabajo de integración curricular fue desarrollado por Edison Javier Quimbiamba Guasgua, bajo mi supervisión.

PhD. Gabriela Lorena Suntaxi Oña
DIRECTORA DE PROYECTO

Certificamos que revisamos el presente trabajo de integración curricular.

Nombre1 Nombre2 Apellido1 Apellido2
REVISOR 1 DEL TRABAJO
DE INTEGRACIÓN CURRICULAR

Nombre1 Nombre2 Apellido1 Apellido2
REVISOR 1 DEL TRABAJO
DE INTEGRACIÓN CURRICULAR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

EDISON JAVIER QUIMBIAMBA GUASGUA

DRA. GABRIELA LORENA SUNTAXI OÑA

Colaboradores del proyecto integrador:

ANDRÉS MAURICIO LLUMIQUINGA GUAMBA

BRYAN ANDRÉS PALMA PONCE

STEVEN JAVIER RIVERA TENELANDA

DEDICATORIA

Dedico este trabajo a mi futuro, este documento es el reflejo de mi presente y contiene todos mis anhelos, sueños, convicciones y toda mi capacidad como profesional con la que concluyo esta etapa y deseo mantenerlos presente en cada instante de mi vida para nunca olvidar lo que soy y a donde quiero llegar.

Dedico esta tesis a mi futuro, porque esta siempre recordara mis ganas de conquistar el mundo, mis ansias de proporcionar grandes avances científicos y desarrollos tecnológicos e industriales, mis ganas de ayudar a implementar empresas de alta tecnología, mi insaciable sed de investigar y de dominar lo inimaginable, porque esta tesis será mi recordatorio personal de mantenerme en la excelencia respetando a mis padres, a la ley, a la gente y a la naturaleza de dar siempre lo mejor de mí sin esperar nada a cambio.

Dedico esta tesis a mi futuro, porque siendo este el primer paso a seguir no será el último y cada uno de ellos me servirá como base para un sin número de proyectos ligados al desarrollo científico, industrial y de inclusión social. Esta tesis siempre me recordará que con gran esfuerzo un Ecuador de alto desarrollo tecnológico es posible, le dedico este trabajo a mi futuro, para que en mi presente siempre me recuerde que nosotros como indígenas ecuatorianos somos capaces de lograr lo que nos propongamos, siempre manteniéndonos orgullosos del pueblo al que representamos.

AGRADECIMIENTOS

Agradezco a Dios, por todas las lecciones y bendiciones que me ha dado a lo largo de mi vida y carrera profesional. A mi madre, María Tránsito, quien siempre estuvo junto a mí durante mis años de estudio. Sin su apoyo, este grado académico hubiera sido imposible de lograr. A mi padre José Alcides, quien desde muy pequeño me inculco el amor por los libros y la ingeniería. A mis tíos Blanca, Jaime y Marco, por acogerme en mis primeros años de estudio en su hogar, en especial a mi Tía, quien me vio crecer como persona y profesional. A Daniela Cumbal y Leonel Yepez, quienes siempre escucharon mis sueños y con quienes compartí algunos de los momentos más felices de mi vida. Un especial agradecimiento a Junior, mi hermano y Wilson, mi mascota, por todo su apoyo durante esta carrera.

A mis amigos y compañeros de Titulación, Andrés Palma y Steven Rivera, quienes me acompañaron a lo largo de esta carrera, entre polémicas y risas, siempre me brindaron su apoyo incondicional y me enseñaron el valor de la confianza. A mis entrañables amigos de la Facultad de Ingeniería Mecánica, Washington de la Cruz, Roberto Calva y Gabriela Armas, por su apoyo continuo y enseñarme que con gran esfuerzo todos podemos alcanzar nuestras metas. También a mis amigos de la Carrera de Ciencias de la Computación, Andrés García y Erick Gallardo, en especial Andrés García, quien siempre me apoyo en el ámbito personal y profesional. A Jairo Muenala, más que un amigo, un hermano, por estar siempre a mi lado, brindándome su apoyo moral, espiritual y económico de manera incondicional y es quien me inculco la pasión por el software. También quiero agradecer a mi amigo que ya partió, Santiago Sinchiguano quien creyó en mis capacidades innatas y siempre tendré presente. A Sid Mohamed, quien me brindo su valiosa amistad y su formación profesional. A mis compañeros de LudoLab, con quienes trabajamos en proyectos de vinculación con la sociedad. En especial al profesor Boris Astudillo quien siempre nos dio ánimos para continuar con nuestros proyectos. A Celinda Camacho y Luis Aguiar, por siempre tenderme una mano y brindarme su sabiduría. A Javier Páez, quien contribuyó a mi desarrollo personal y profesional. A la Dra. Gabriela Suntaxi, por darnos su apoyo y confianza para culminar exitosamente este trabajo, por lo cual estaré eternamente agradecido. Cada uno de ellos creyeron en mis capacidades y tengan por seguro que nunca los defraudaré.

CONTENIDO

RESUMEN

En la actualidad los Ataques de Inyección SQL(SQL Injection Attack or SQLIAS por sus siglas en inglés), se encuentra en la tercera posición de la lista OWASP (Open Application Security Project) [1], este tipo de ataque es muy fácil de explotar Y comprometer la seguridad de un sistema. Sin embargo, existen herramientas y tecnologías que permiten mitigar la amenaza que representan estos ataques. En este trabajo se realizó una Revisión Sistemática de la Literatura relacionada al uso de algoritmos de minería de datos para detectar y prevenir ataques de inyección SQL. A partir de los resultados de esta investigación, se ha seleccionado las Redes Neuronales Artificiales, como objeto de evaluación para determinar el grado de efectividad al detectar secuencias de texto que podrían resultar en un Ataque de Inyección SQL. Finalmente, se implementa un aplicativo web que permite procesar y monitorear la información contenida en logs de registros de ingresos de datos utilizando el modelo antes creado para la detección y prevención de los ataques de inyección SQL. Es componente tiene como objetivo elevar la confianza y seguridad de los sistemas de la organización donde sea implementado.

PALABRAS CLAVE: Minería de Datos, CRISP-DM, RNN, SQLIA, Seguridad de la Información

ABSTRACT

Currently, SQL Injection Attack (SQL Injection Attack or SQLIAS) is in the third position on the OWASP (Open Application Security Project) list [1].This attack is very easy to exploit and compromises the security of a system. However, some tools and technologies mitigate the threat posed by these attacks. In this component, a Systematic Review of the Literature related to the use of data mining algorithms to detect and prevent SQL Injection Attacks was carried out. From the results of this research, Artificial Neural Networks has been selected as the object of evaluation to determine the degree of effectiveness in detecting text sequences that could result in a SQL Injection Attack. Finally, a web application is implemented to process and monitor the information contained in logs of data entry records using the model previously created for deteting and preventing of SQL Injection Attacks. This component aims to increase the confidence and security of the organization's systems where it is implemented.

KEYWORDS: Data Mining, CRISP-DM, RNN, SQLIA, Information Security

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Hoy en día los sistemas informáticos han aumentado su complejidad debido a la transformación digital que han tenido las organizaciones. Por lo que ninguna organización está exenta de ser víctima de un ataque cibernético para romper su seguridad. Frente a esta problemática, las organizaciones se ven en la obligación de implementar estrategias y medidas para reducir la probabilidad de sufrir un ataque que violente la seguridad de sus sistemas informáticos. Uno de los principales ataques que sufren los sistemas informáticos es la inyección, este ataque ocupa el tercer puesto en la lista de OWASP Top 10:2021. Uno de los tipos de ataques de inyección más comunes es la inyección SQL, este tipo de ataque puede poner en riesgo la seguridad de la información. Existen muchas herramientas que ayudan a mitigar esta amenaza. Por lo cual es importante realizar una revisión sistemática de la literatura, para determinar el estado del arte para la detección y predicción de ataques de inyección SQL.

Uno de los aspectos fundamentales de las organizaciones, es el volumen de información que se produce en un tiempo reducido. Por lo cual es importante analizar esta información para identificar patrones que indiquen cuál es el comportamiento de los usuarios. Para identificar estos patrones se aplican técnicas de minería de datos que automaticen el proceso de identificación de comportamientos que comprometan la seguridad de los sistemas informáticos.

En este proyecto se propone, construir un modelo de Machine Learning basado en una Red Neuronal Recurrente a partir de la metodología CRISP-DM, para posteriormente ser implementado en un aplicativo web, de manera que se facilite la interacción de los usuarios Dada la complejidad que existe al analizar las técnicas y algoritmos existentes, este trabajo forma parte de un proyecto integrador dividido en 4 componentes enfocados en diversos aspectos, particularmente, en la evaluación de los algoritmos más utilizados en la minería de datos para la detección de SQLIAs, dado que en cada componente se evalúa un algoritmo

distinto. Este trabajo cubre uno de los cuatro componentes. Por lo tanto, secciones de este documento han sido trabajadas de forma grupal; las partes comunes del proyecto serán identificadas explícitamente al inicio de cada sección.

1.1 OBJETIVO GENERAL

Realizar el análisis y evaluación de algoritmo de minería de datos para detección de ataques de inyección SQL en BIG DATA.

1.2 OBJETIVOS ESPECÍFICOS

- ❒ Realizar un estudio de la literatura para identificar y analizar trabajos relacionados con la minería de datos para la predicción y prevención de ataques de inyección SQL.
- ❒ Desarrollar un prototipo de software que permita detectar y predecir ataques de inyección SQL en BIG DATA.
- ❒ Evaluar el prototipo desarrollado en un caso de estudio utilizando datos reales.

1.3 ALCANCE

En el presente trabajo se realizará con base en una variación de la metodología descrita en [2], la cual contiene 5 fases:

1. Fase de planificación: se definirá los requerimientos y aspectos generales del proyecto tales como recursos, justificación , cronograma, etc.
2. Fase de diseño: dentro de esta fase se realizará la revisión sistemática de la literatura utilizando una adaptación de la metodología descrita en [3]. A partir de esta revisión se seleccionará un algoritmo de minería de datos para ser evaluado, así como las métricas que serán consideradas para su evaluación.
3. Fase de implementación: en esta fase se aplicará la metodología de minería de datos CRISP-DM para obtener un modelo que permita definir la efectividad de este algoritmo en la detección de ataques SQL en Big Data [4].Además, se desarrollará un prototipo

que permitirá monitorear e identificar ataques de inyección SQL en los sistemas de una organización. Para este proceso se utilizará la metodología XP[5].

4. Fase de evaluación: se analizarán los resultados obtenidos del proceso de minería de datos en un caso de estudio real.

5. Fase de comunicación: finalmente en esta fase presentará un informe de los resultados obtenidos en la evaluación de este sistema software.

2 MARCO TEÓRICO

2.1 SEGURIDAD DE LA INFORMACIÓN

La seguridad de la información es el conjunto de procedimientos reactivos y preventivos establecidos por una entidad u organización, así como de los sistemas y la información que pretenden proteger. Tiene como objetivo preservar la seguridad de la información en cada una de sus aristas. En la seguridad de la información también se integran los pasos y protocolos que debe seguir el personal de una organización para prevenir posibles ataques y exposición de activos de acceso restringido.

2.1.1 Servicios de seguridad y mecanismos de seguridad

Los servicios de seguridad son aquellos que se brindan para preservar la seguridad tanto de los activos como de los procedimientos que se manejan dentro de la empresas, mediante la aplicación de normas y políticas bien establecidas [6].

Existen conceptos muy importante dentro de la seguridad de la información que son considerados como pilares fundamentales al momentos de tratar estos temas. Estos aspectos son:[6].

- **Confidencialidad:** Se trata de la protección de activos de información de manera que estos no sean expuestos a personas no autorizadas por la organización. La confidencialidad también hace referencia a los mecanismos utilizados para lograr esta protección como puede ser la implementación de controles de acceso rigurosos [6].
- **Integridad de datos:** Se trata de la protección de activos de información de manera que estos no sean modificados o eliminados por personas no autorizadas por la organización. La integridad también hace referencia a los mecanismos utilizados para

lograr esta protección como puede ser la implementación de validación de certificados de autenticidad [6].

- ❒ **Disponibilidad:** Se trata de la protección de activos de información de manera que estos estén disponibles por el personal autorizado por la organización la mayor parte del tiempo. La disponibilidad también hace referencia a los mecanismos utilizados para lograr esta protección como puede ser la implementación de discos redundantes [6].

2.2 ATAQUES A BASES DE DATOS

Las amenazas mas comunes que afectan a las bases de datos y que deben ser mitigadas con la fortificación de la seguridad en los servidores de bases de datos y agregando técnicas y procedimientos de seguridad informática, que se muestran a continuación.

- ❒ **Gestión de permisos inadecuada:** los servidores de bases de datos mantienen la configuración de seguridad predeterminada, siendo blanco fácil de atacantes que conocen los permiso predeterminados. Usuarios que utilizan los privilegios para acceder a una parte no autorizada de un motor de base de datos, para la divulgación de información confidencial [7].

- ❒ **Ataques de inyección de base de datos:** Es una forma muy común de ataque de inyección. El mecanismo de este tipo de ataques consiste en la inserción de comando en lenguaje de consulta estructurada dentro de campos de ingreso en las aplicaciones. Estos comandos son ejecutados en la base de datos, generando diversos efectos que pueden vulnerar cualquier aspecto de la seguridad de la información. [7].

- ❒ **Vulnerabilidades a base de datos explotables:** Las organizaciones no aplican parches a su software principal de DataBase Management System (DBMS), incluso si el proveedor lanza un parche para eliminarla, puede pasar mucho tiempo antes que las organizaciones actualicen sus sistemas[7].

- ❒ **Existencia de servidores de base de datos ocultos:** Los usuarios no cumplen con las políticas de seguridad de una organización, e instalan las base de datos a sus discreción para la solución de necesidades particulares, creando nuevos servidores asociados a la red interna de la empresa. Estos servidores se mantienen ocultos, y

pueden utilizarse para varios fines como puede ser la divulgación de la información privada de la empresa [7].

- **Copia de seguridad accesible:** los servidores de bases de datos puede estar protegidos por varias capas de seguridad, los usuarios sin autorización pueden acceder a las copias de seguridad de la base de datos. Una de las mayores problemáticas es que los usuarios autorizados pueden realizar copias de seguridad para extraer información confidencial y sacarla de la organización sin autorización [7].

2.2.1 Ataques de inyección SQL

Estos ataques son quizás uno de los mas devastadoras para la información, ya que puede llevar a la exposición de información sensible que se encuentra en una base de datos, incluyendo nombres de usuarios, contraseñas, direcciones, números de teléfono y detalles de tarjeta de crédito. Esta vulnerabilidad se produce cuando un atacante tiene la capacidad de modificar las consultas de un lenguaje de consultas SQL. Este tipo de ataque no solo afecta a los aplicativos web si no a todos los aplicativos que acepten entradas de una fuente no fiable y luego use esas entradas para formar sentencias SQL [8]. Inyectar código SQL en aplicaciones no requiere de un gran esfuerzo, simplemente requiere del entendimiento de la semántica SQL. Uno de los mayores problemas es que una gran cantidad de aplicaciones toman las entradas de usuario y los introducen en una consulta predefinida. Posteriormente la consulta se entregara a la base de datos para su ejecución como se muestra en la Figura 2.1. Si los desarrolladores no han utilizados buenas practicas de desarrollo para proteger las entradas inesperadas de datos por parte de los usuarios, se puede producir un comportamiento inesperado en el software y como resultado se tiene la revelación de información privada y confidencial [8].

2.2.2 Tipos de ataques de inyección SQL

Existen tres clasificaciones importantes de ataques de inyección SQL: Inyección por unión, inyección por error y ataques ciegos SQL [8]. Otra forma de clasificar este tipo de ataque es mediante el medio utilizado para perpetuar el ataque, como pueden ser mediante modificación de enlaces, modificación de solicitudes al servidor, o simplemente en entradas de texto simple dentro de las aplicaciones [8].

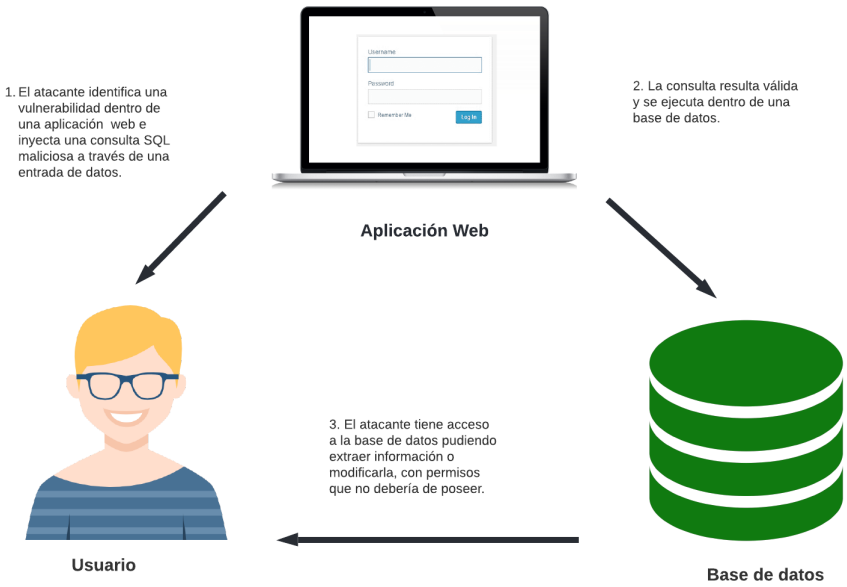


Figura 2.1: Ataque de inyección SQL. Fuente: Propia

A continuación, se explican los tipos de ataques de inyección SQL en la actualidad.

2.2.2.1 Ataque de inyección SQL por unión

Es un tipo de ataque bastante frecuente, basado en el uso de la sentencia “UNION"para la extracción de información proveniente de una base de datos. Este operador se lo utiliza de la siguiente manera:

```
SELECT a,b FROM table where field = '' UNION SELECT c,d FROM tabla1 --'
```

Esta consulta es una combinación de dos sentencias, la sentencia original donde se obtienen los campos a y b de la tabla *table*, y una sentencia UNION, de manera que también se retornan los campos c y d de la tabla *tabla1*. Para la realización de estos ataques, el atacante debe tener conocimiento de la estructura de la base de datos de manera que no existan problemas de compatibilidad al retornar los datos [8].

2.2.2.2 Ataque de inyección SQL por error

En este tipo de ataques, la mecánica consiste en la generación de errores al momento de realizar consultas en la base de datos para obtener información importante sobre la estructura de la base de datos, así como de las tablas y los campos que existen. Varios motores de bases de datos poseen ciertos mensajes de error particulares que facilitan el reconocimiento de la estructura de estos, y avanzar a la realización de ataques más enfocados [8] .

2.2.2.3 Ataque de inyección SQL ciegos

Estos ataques tienen dos subclasificaciones: ataques basados en operaciones booleanas, y ataques basados en el tiempo. En el primer tipo de ataque, se envían “preguntas’a la base de batos, estas preguntas solo pueden retornar un valor booleano (cierto o falso). Con la ejecución de este tipo de preguntas, los atacantes pueden obtener e igual manera información sensible como la existencia a o no de ciertos elementos en la base de datos [8].

El ataque basado en el tiempo se lo realiza mediante operaciones que detienen el funcionamiento de la base de datos en función de ciertos parámetros. Por ejemplo, para descubrir ciertos aspectos de la estructura de una tabla como el nombre. Si el nombre de dicha tabla es users, el sistema espera 15 segundos, si el nombre es usuario, espera 20 segundos, y así con varias combinaciones de manera que el atacante pueda conjeturar información privilegiada de la base de datos [8].

2.2.2.4 Ataque de inyección SQL basado en entradas de un usuario

En prácticamente todos los sistemas actuales se permite al usuario ingresar texto de cualquier tipo en sus entradas. Si estas entradas de datos no son sanitizadas, pueden ser un punto muy vulnerable para que el atacante ingrese sentencias SQL maliciosas y estas se ejecuten en la base de datos [8].

2.2.2.5 Ataque de inyección basado en las cookies

Debido a que muchas aplicaciones suelen utilizar cookies para realizar ciertas operaciones con las bases de datos. Es muy comun que los atacantes realicen alteraciones a dichas cookies para inyectar código SQL malicioso y que este se ejecuten en las bases de datos [8] .

2.2.2.6 Ataques de inyección SQL basada en la cabeceras HTTP

De manera similar a los ataques con cookies, es muy común que los atacantes realicen modificaciones a las cabeceras de una solicitud al servidor, insertando sentencias SQL maliciosas. Este tipo de ataques es de especial importancia, ya que hace notar que las validaciones de datos deben realizarse tanto en el cliente como en el servidor del sistema [8].

2.3 BIG DATA

El término BIG DATA se aplica a datos que ocupan mucho volumen en cuanto memoria, de manera que se requieren técnicas elaboradas para su procesamiento [9].

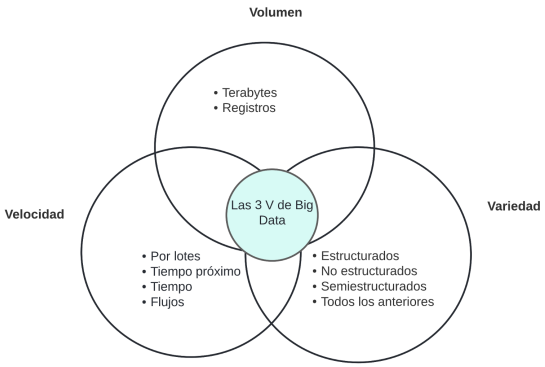


Figura 2.2: Las 3 V del BIG DATA. Fuente: Propia.

2.3.0.1 Características de BIG DATA

Comúnmente se conocen tres características principales para BIG DATA: Volumen, Velocidad y Variedad, que se muestran en la Figura. 2.2.

- ❒ **Volumen:** Se refiere al crecimiento acelerado de los datos en las organizaciones (Exabytes, terabytes y petabytes), estos datos son generados por personas y máquinas. Las cantidades que hoy en día nos parecen enormes en pocos años serán normales [10].
- ❒ **Velocidad:** Hace referencia a que a medida que los datos crecen, de la misma manera debe aumentar la velocidad con la que estos se procesan, ya que existe mucha información que puede resultar valiosa y su transformación en conocimientos debe realizarse casi en tiempo real, de manera que se puedan tomar decisiones a partir de estos datos [10].
- ❒ **Variedad:** La variedad significa que los datos masivos pueden presentar una gran cantidad de formatos, pueden ser estructurados o no estructurados y pueden provenir de cualquier fuente. Cuando los datos son variados, existe una mayor probabilidad de encontrar valor en estos [10].

2.4 MINERÍA DE DATOS

La minería de datos es un proceso lógico que por lo general se la utiliza para tratar una gran cantidad de datos con la finalidad de encontrar datos útiles, e identificar patrones que a simple vista son desconocidos. La minería de datos y reglas son tecnologías importantes que tiene como finalidad la de tratar un gran cantidad de datos acumulados en el campo de la seguridad informática. En la actualidad existen muchos mecanismos de supervisión que se encargan de recoger datos de seguridad de la información y posteriormente son analizados utilizando tecnologías de minería de datos. Estas técnicas facilitan la manipulación de la información y la toma de decisiones eficaces. La minería de datos utiliza tareas y pasos fundamentas y son los siguientes:

- ❒ **Exploración**: es el primer paso para la exploración de datos, para completar esta paso los datos deben ser limpiados y se transforma a otra forma de visualización,

en donde se identifica las variables más importantes y la naturaleza de los datos en función del problema [11].

- ❒ **Identificación de patrones:** en este paso se identifican y seleccionan los patrones que facilitarán la predicción [11].
- ❒ **Despliegue:** Los patrones se despliegan para obtener el resultado deseado [11].

2.4.1 Técnica de minería de datos

En lo que respecta a la manipulación de datos y su posterior transformación en información a partir de las bases de datos, se utilizan diferentes algoritmos y técnicas como la clasificación, la agrupación, la regresión, la inteligencia artificial, las redes neuronales, las reglas de asociación, los árboles de decisión, el algoritmo genético, etc [11]. La clasificación que se toma como punto de partida en lo que respecta a las técnicas utilizadas en el proceso de la minería de datos, son las técnicas que se relacionan con la predicción, en donde sus variables pueden tener una clasificación en independientes y dependientes, técnicas auxiliares y técnicas descriptivas.

2.4.1.1 Técnicas predictivas

Para la utilización de estas técnicas se debe especificar primeramente el modelo que se va a utilizar para los datos, basándose en un conocimiento teórico que ha sido adquirido previamente. Esta técnica empieza en las fases de identificación objetiva, en la cual se aplican una serie de reglas que ayudan a la identificación del mejor modelo posible, que se ajuste a los datos que se van a analizar. Luego se continúa calculando los parámetros del modelo que se ha seleccionado en la fase de la identificación. Posteriormente, se contrasta la validez del modelo y se predice y se valida los datos [11].

2.4.1.2 Técnicas descriptivas

En este tipo de técnicas no se les asigna ningún rol predeterminado a las variables. Tampoco se infiere la existencia de ciertas variables dependientes o independientes. Además, tampoco se infiere la existencia de algún tipo de modelo previo que se utilizara para los

datos. Estos modelos se crean de forma automática tomando como referencia el reconocimiento de patrones. En este grupo mencionado anteriormente se pueden incluir varias técnicas como la clusterización y segmentación, que son técnicas que utilizan la clasificación para su funcionamiento, las técnicas basadas en la exploración y análisis de los datos, las técnicas basadas en la asociación y dependencia de variables, etc.

Las técnicas mencionadas anteriormente, es decir, las técnicas predictivas y descriptivas, se centran en descubrir el conocimiento que pueden proporcionar los datos. Adicionalmente, a estas técnicas mencionadas anteriormente, existen técnicas complementarias, las cuales son herramientas de carácter más superficial y limitado. Estas herramientas son métodos basados en técnicas descriptivas y estadísticas que están por lo general enfocadas a la verificación de los datos [11].

A continuación se detalla la clasificación de las técnicas de minería de datos según [11].

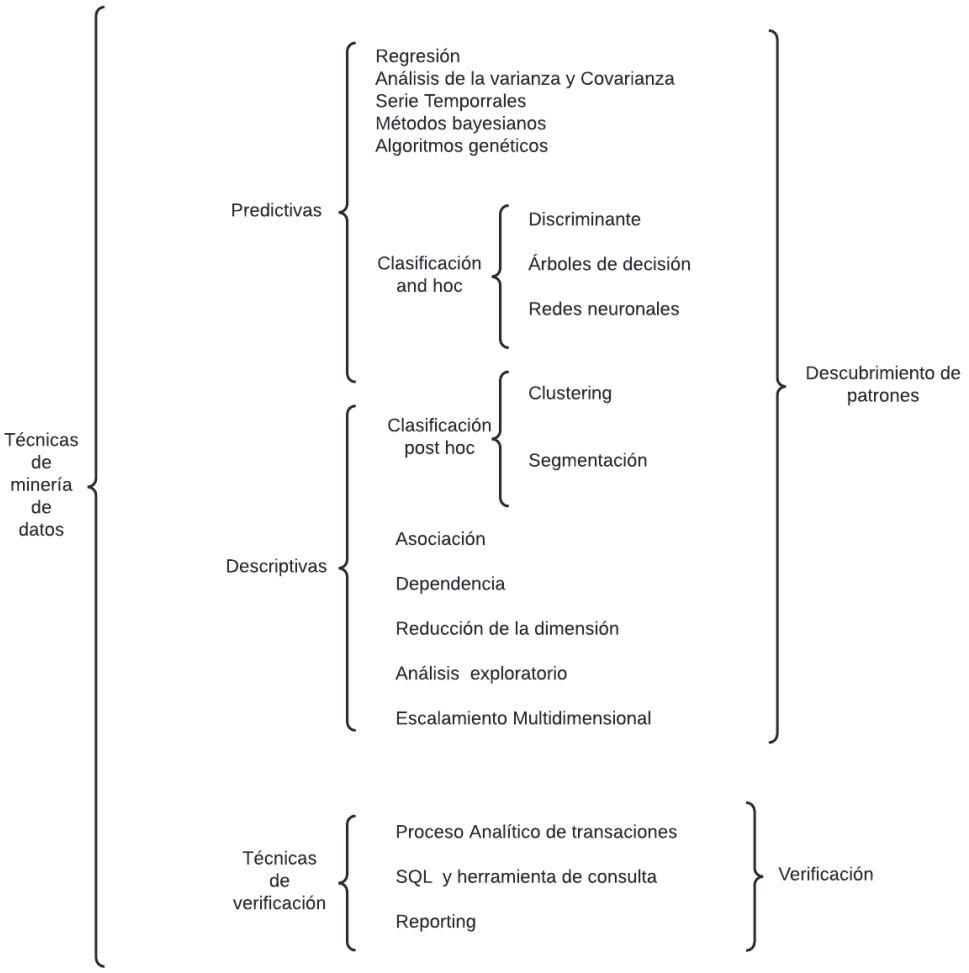


Figura 2.3: Técnicas de minería de datos [11]

Como se observa en la Figura 2.3 las técnicas de clasificación pueden pertenecer a un mismo grupo de clasificación y también pertenecer a las técnicas predictivas.

2.4.2 Herramientas de minería de datos

Según [12], las herramientas que se utilizan en la minería de datos asociada al proceso que involucra la extracción de información se pueden clasificar en dos grandes grupos:

- ❒ **Técnicas de verificación:** En este tipo de técnicas el sistema únicamente se centra en comprobar la hipótesis que ha sido suministrada por el usuario.
- ❒ **Método de descubrimiento:** En lo que respecta al método de descubrimiento, se enfoca en buscar patrones que ayuden a mejorar la toma de decisiones automáticamente. En este grupo se pueden incluir todas las técnicas de predicción.

En la Tabla 2.1, se analiza y se compara cada una de las herramientas que más se utilizan en el mercado actual: WEKA, KNIME, Rapidminer, SAS y Orange. Es importante mencionar que los usuarios por lo general utilizan más de una de estas herramientas o también combinando entre diferentes herramientas.

Herramienta	**Características**	**Lenguaje de programación utilizado**	**Sistema operativo**	**Tipo de Licencia**
WEKA	Posee diversos métodos de clasificación	Java	Windows, macOs, Linux	Software libre GPL
Orange	Posee una visualización de datos atractiva sin que se requiera amplios conocimientos previos para su tulizacion	Núcleo del software: C++, ampliación y lenguaje de entrada: Python	Windows,macOs, Linux	Software libre (GLP)
KNIME	Es un software de data mining de código abierto que ha facilitado el acceso a los análisis predictivos	Java	Windows, macOS, Linux	Software libre (GPL)
SAS	Su costo es elevado, pero eficiente para el lenguaje SAS que utilizan las grandes empresas	Lenguaje SAS	Windows, macOS, Linux	Freeware limitado a instituciones públicas, su costo se establece tras una previa solicitud, posee diversos modelos para su elección

Tabla 2.1: Herramientas de minería de datos [13] .

3 METODOLOGÍA

En esta sección, se describe de manera general las metodologías utilizadas para la realización del proyecto en cada una de las fases del proyecto. Se contemplaron las metodologías para: la revisión sistemática de la literatura, el proceso de minería de datos, y el proceso de desarrollo de un sistema software donde se implementará el resultado del proceso de minería de datos.

Dado que este trabajo forma parte de un proyecto integrador, las metodologías utilizadas para el desarrollo de cada componente son comunes entre los cuatro componentes del proyecto y fueron trabajadas de manera grupal.

3.1 METODOLOGÍA DE REVISIÓN SISTEMÁTICA DE LA LITERATURA

La metodología por la que se optó para la revisión sistemática de la literatura es la propuesta por el autor Kitchenham [14] dado que propone una metodología orientada al desarrollo de software. La estructura de esta metodología se divide en 3 secciones generales: planificación, realización y presentación de informes. A continuación, se describen cada una de estas secciones con sus respectivas fases.

3.1.1 Planificación de la revisión

3.1.1.1 Identificación de la necesidad de una revisión

En esta fase es necesario asegurarse de que la revisión sistemática es necesaria, por lo tanto, los interesados en realizar la revisión, necesitan determinar el estado del arte con

respecto al tema a investigarse en función de los parámetros de evaluación adecuados. Para comprobar esto, se realiza una lista de comprobación que contiene las siguientes preguntas propuestas en **khan2001undertaking**, que ayudarán a identificar la necesidad de la revisión:

- ❒ ¿Cuáles son los objetivos de la revisión?
- ❒ ¿Cuáles fueron las fuentes utilizadas para identificar los estudios primarios? ¿Existieron restricciones?
- ❒ ¿Qué criterios de inclusión y exclusión se definieron y de que manera se aplicaron?
- ❒ ¿Cuáles fueron los criterios utilizados para evaluar la calidad de los estudios primarios y cómo se aplicaron?
- ❒ ¿De qué manera se extrajeron los datos de los estudios primarios?
- ❒ ¿Cómo se sintetizaron los datos? ¿Cómo se investigaron las diferencias entre los estudios? ¿Cómo se combinaron los datos? ¿Era razonable combinar los estudios?
- ❒ ¿Las conclusiones se desprenden de las pruebas?

3.1.1.2 Elaboración de un protocolo de revisión

El protocolo de revisión se encarga de especificar cuáles son los métodos a utilizarse al momento de desarrollar una revisión sistemática. Aplicando un protocolo, se disminuyen las probabilidades de generar un sesgo en la investigación. Los siguientes componentes son contemplados al momento de realizar un protocolo de revisión:

- ❒ Preguntas de investigación con las que se pretende responder con la revisión
- ❒ Estrategias que se utilizarán para buscar estudios primarios, incluyendo términos de búsqueda, recursos en los cuales se realizará la búsqueda, incluyendo bases de datos, revistas científicas, conferencias.
- ❒ Criterios y métodos para la selección de estudios. Estos criterios de selección de estudios especifican lo que se excluye o incluye dentro de la revisión.
- ❒ Listas de comprobación y procedimientos de evaluación de la calidad de los estudios.

- ❒ Definición de la manera en la que se obtendrá la información necesaria de cada estudio primario.
- ❒ Resumen de la extracción de datos, definición de la manera en la que se realizará el resumen y sus estrategias.

3.1.2 Realización de la revisión

Cuando los investigadores han acordado el protocolo a seguir, se puede empezar la revisión, esto implica las siguientes fases:

3.1.2.1 Identificación de la investigación

El objetivo principal de una revisión sistemática es encontrar la mayor cantidad de estudios primarios con la mayor relevancia, siempre y cuando estos aporten a contestar las preguntas de investigación. Esta búsqueda debe realizarse de manera objetiva y libre de sesgos. Por lo tanto, se realizan algunos pasos adicionales comparados con las revisiones tradicionales:

a. Generar una estrategia de búsqueda

Una estrategia de búsqueda suele ser iterativa y se benefician de búsquedas preliminares, en las cuales se identifican las revisiones sistemáticas existentes donde se evalúan el volumen de estudios potencialmente relevantes. Un enfoque recomendado es desglosar la pregunta de investigación en fases individuales, por ejemplo, población, intervención, resultados, diseños de estudio. Luego se elabora una lista de sinónimos y abreviaturas. A continuación, construir cadenas de búsqueda sofisticadas utilizando combinaciones booleanas AND y OR.

b. Sesgo de publicación

El sesgo de publicación hace referencia a la tendencia que existe de seleccionar artículo que presenten un resultado particular, ya que es el investigador quien asigna un valor de que tan bueno o malo es un artículo. Por lo tanto, el investigador debe informarse sobre la problemática y explorar la literatura, conferencias o contactar con expertos e investigadores que el área de interés que puedan guiarlo en la investigación y no recaer en el sesgo de publicación.

c. Gestión de la bibliografía y recuperación de documentos

La gestión de la bibliografía permite gestionar un gran número de referencias que se pueden obtener de una investigación bibliografía exhaustiva. Por lo tanto, es importante tener un sistema para esto, por ejemplo, se pueden utilizar paquetes bibliográficos como Reference Manager o Endnote o simplemente un Excel con toda esta información.

d. Documentación de la búsqueda

Todo el proceso del desarrollo de una revisión sistemática debe mantener transparencia y ser reproducible, por lo tanto, la revisión debe: estar documentada con un detalle suficiente para su reproducción, esto incluye anotar los cambios que se realicen durante la búsqueda, así como la justificación pertinente.

3.1.2.2 Selección de estudios primarios

Cuando se han conseguido los artículos principales de la búsqueda, se procede a evaluar la relevancia real de estos.

a. Criterios para la selección de estudios

Los criterios de selección de los estudios tienen por objetivo el de identificar los estudios primarios que aportan pruebas directas a la pregunta de investigación. Los criterios de inclusión y exclusión deben tomar como referencia la pregunta de investigación. Estos criterios deben probarse para garantizar que su interpretación es fiable y que los estudios están clasificados correctamente.

b. Proceso de selección de estudios

La selección de estudios es un proceso de varias etapas donde se empieza con los criterios de selección que los debe interpretar el investigador, a menos que los estudios puedan excluirse porque las copias obtenidas no están completas. Por lo tanto, una vez obtenidos los artículos a ser analizados, se realiza el proceso de aplicación de los criterios de inclusión y exclusión, y mantener una lista de estudios excluidos en los cuales se debe identificar el motivo de la exclusión.

3.1.2.3 Evaluación de la calidad del estudio

Además de los criterios de inclusión y exclusión, es necesario considerar la evaluación de la calidad de los estudios primarios:

- ❒ Proporcionar criterios de inclusión/exclusión aún más detallados.
- ❒ Investigar si las diferencias de calidad explican las diferencias en los resultados de los estudios.
- ❒ Como medio para ponderar la importancia de los estudios individuales cuando se sintetizan los resultados.
- ❒ Orientar la interpretación de los resultados y determinar la fuerza de las inferencias.
- ❒ Orientar las recomendaciones para futuras investigaciones.

3.1.2.4 Extracción y seguimiento de datos

En esta fase se diseñan los formularios de extracción de datos, los cuales cumplen la función de registrar con precisión la información que los investigadores han obtenido de los estudios primarios. Con el objetivo de reducir el sesgo, los formularios de extracción de datos deben definirse y probarse cuando se defina el protocolo del estudio.

a. Diseño de formularios de extracción de datos

Estos formularios de datos deben tener un diseño que permita la recolección de información que se necesite para abordar las preguntas de la revisión y los criterios de calidad del estudio. En la mayoría de los casos, la extracción de datos se definirá como un conjunto de valores numéricos que deben extraerse para cada estudio. Una recomendación importante es utilizar formularios electrónicos, ya que facilitan el análisis posterior.

b. Contenido de los formularios para la recolección de datos

Los formularios ayudan a complementar las preguntas de investigación. Dentro de estos formularios se debe proporcionar información relevante que incluya lo siguiente:

- ❒ Nombre de la revisión

- ❒ Fecha de extracción de datos
- ❒ Título, autores, revista, detalles de publicación
- ❒ Espacio para notas adicionales

c. Procedimientos para la extracción de datos

Para el procedimiento de extracción de datos de los estudios primarios es importante realizarlo de manera independiente por dos o más investigadores. Luego, estos datos extraídos deben compararse y solucionar los desacuerdos que podrían presentarse mediante un consenso entre los investigadores. Es recomendable utilizar un formulario aparte para marcar y corregir los errores o desacuerdos presentados.

d. Múltiples publicaciones de los mismos datos

Es importante tener en cuenta evitar la inclusión de múltiples publicaciones que contengan los mismos datos en la revisión sistemática, debido a que estos informes duplicados podrían sesgar gravemente cualquier resultado obtenido de la investigación. En caso de que existieren publicaciones duplicadas, es recomendable utilizar la publicación más reciente para la revisión sistemática.

e. Datos no publicados, datos que faltan y datos que requieren manipulación

Si existiera el caso de que se dispone de información de estudios que se están desarrollando o están en curso, debe incluirse siempre y cuando sea posible información de calidad sobre el estudio. Los informes no siempre presentan todos los datos relevantes, también pueden estar mal redactados y ser ambiguos, por lo tanto, es necesario contactar a los autores para conseguir la información necesaria. En ciertas ocasiones los estudios primarios no proporcionan todos los datos, pero en algunas situaciones se pueden recrear estos datos necesarios a partir de la manipulación de los datos publicados. Por lo tanto, si se diera el caso de manipulación de datos, es importante someterlos a un análisis de sensibilidad para su posterior uso.

3.1.2.5 Síntesis de datos

La síntesis de datos consiste en cotejar y resumir los resultados obtenidos de los estudios primarios. La síntesis que se realiza sobre los resultados puede ser descriptiva (no cuantitativa). En algunos casos también es posible complementar el análisis cualitativo con una síntesis cuantitativa. El uso de técnicas que utilizan la estadística para su desarrollo se las

denomina meta-analisis.

a. Síntesis descriptiva

La información extraída de los estudios, como la población, el contexto, el tamaño de la muestra, los resultados y la calidad del estudio, debe ser tabulada de una forma coherente, siempre teniendo presente la pregunta de la revisión. Estas tablas deben tener una estructura tal que permita evidenciar la relación que existe entre los estudios analizados.

b. Sensibilidad del análisis

La realización de un análisis de sensibilidad es importante cuando se realiza un meta-analisis. El meta-analisis se utiliza para proporcionar una estimación global del efecto de tratamiento y su variabilidad en el estudio. En estos casos lo ideal es la repetición de varios subconjuntos de estudios primarios con el objetivo de determinar si estos resultados son robustos o no. Los tipos de subconjuntos pueden ser:

- ❒ Solo estudios primarios de alta calidad
- ❒ Estudios primarios de tipos particulares
- ❒ Estudios primarios para los que la extracción de datos no presento dificultades

3.1.3 Presentación de informes

Esta fase es una de las más importantes dado que permite comunicar de forma eficaz los resultados de la revisión realizada. Generalmente, estas revisiones son presentadas de dos maneras:

- ❒ Mediante informes técnicos dentro de documentos académicos como tesis
- ❒ En revistas o conferencias especializadas

3.2 METODOLOGÍA DE MINERÍA DE DATOS CRISP-DM

CRISP-DM (Cross Industry Process for Data Mining) [15], es una metodología de minería de datos que incluye descripciones de las etapas que deben seguirse dentro un proyecto, así como las tareas requeridas en cada una de estas etapas. CRISP-DM también se puede

considerar como un modelo de proceso de minería de datos que ayuda a los expertos en la materia a resolver un problema.

CRISP-DM, está estructurado en seis etapas, algunas de las cuales son bidireccionales, es decir, cada una puede retroceder para hacer revisiones y correcciones, esto implica que los segmentos de las etapas no necesariamente están dispuestos en el orden que se muestra en la Figura 3.1.

El modelo de CRISP-DM es flexible y se puede configurar fácilmente de modo que se adapta a las actividades de una organización, generando soluciones que brinden el mayor valor posible para solventar sus necesidades. Permitiendo crear un modelo de minería de datos que se adapte a sus necesidades concretas[15].

En la Figura 3.1 se muestran las fases que constan en la metodología y que son detalladas a continuación:

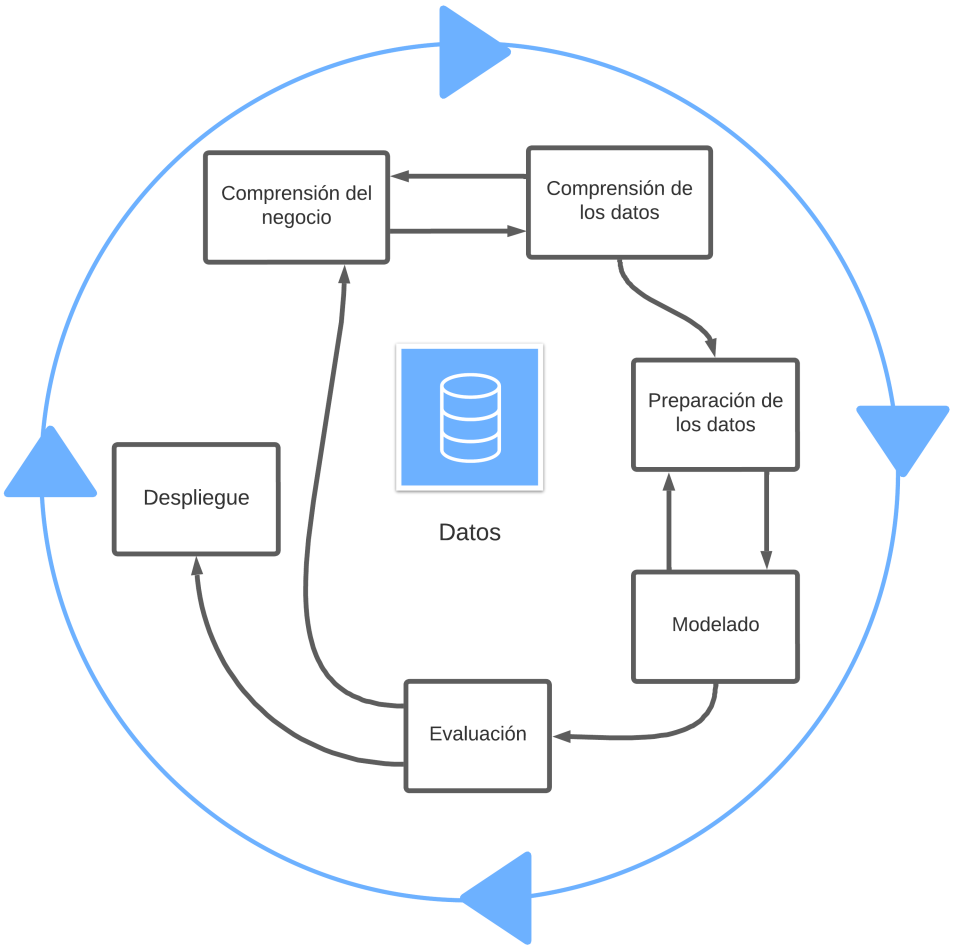


Figura 3.1: Fases de la metodología CRISP-DM. Adaptado de [15].

3.2.1 Comprensión del negocio

En la primera fase, que es quizás la más importante, se analiza a la empresa desde una perspectiva comercial, permitiendo traducir sus necesidades y objetivos de negocio hacia requerimientos más técnicos [15]. Es decir, que si no se llega a la comprensión de los objetivos del negocio, ningún algoritmo por complejo que sea podrá lograr resultados confiables. Para la extracción de datos de manera más efectiva, es indispensable tener una buena compresión del problema que desea resolver, lo que le permitirá recopilar los datos necesarios y poder interpretar los resultados correctamente. Al final de esta etapa se obtiene un plan de proyecto donde se describen los objetivos de la empresa como un proyecto de minería de datos. Las tareas que se llevan a cabo durante esta etapa son:

- **Determinar los objetivos de negocio**

  Esta es la primera tarea desarrollada y su objetivo es identificar el problema que necesita ser resuelto, por lo cual se debe hace la siguiente pregunta ¿Por qué usar Data Mining?, y la realidad es que en la época actual, existen muchos problemas que los datos pueden brindar información valiosa, y a partir de la minería de datos, obtener conocimiento para tomar decisiones oportunas. Un ejemplo de esto podría ser la ubicación estratégica de productos dentro de un supermercado según los hábitos de compra de un usuario. Para ello es posible utilizar datos obtenidos de facturar, analizar dichos datos, y obtener patrones que determinen que productos son comprados generalmente juntos y ubicarlos de mejor manera en las estanterías [15]. Como objetivo de la empresa debe determinar los criterios para decidir que la minería de datos se implementó correctamente o no. En el caso anterior podría ser el aumento en las ventas de un producto en particular.

- **Evaluación de la situación**

  Es importante matizar el estado de la situación antes de proceder a realizar el proceso de minería de datos, teniendo en cuenta aspectos como: ¿Qué conocimiento tiene disponible sobre la materia?, ¿Se requiere conteo de datos?, ¿Resulta rentable realizar minería de datos?. En esta fase se identifican requerimientos de problemas, tanto de negocio como de minería de datos. El propósito de esta tarea es analizar la mayor cantidad de aspectos posibles que se deben tomar en cuenta antes de proceder a realizar la minería de datos [15]. Estos aspectos incluyen, pero no se limitan a,

personal, datos, riesgos, etc.

- **Realizar el plan del proyecto** Al final de la primera etapa de CRISP-DM, es necesario desarrollar un plan de proyecto donde se detallen como se procederá con el proyecto, así como las técnicas que se utilizarán en cada paso.

3.2.2 Compresión de los datos

Esta etapa inicia con la obtención de los datos que serán utilizados y sigue con tareas relacionadas con la comprensión de dichos datos, tales como identificación de anomalías, análisis de calidad, identificación de atributos, generación de hipótesis, etc.

La comprensión de datos se encuentra fuertemente relacionada con la comprensión del negocio, ya que es indispensable comprender los datos disponibles para continuar con la ejecución del plan elaborado [15].

- **Recolectar los datos iniciales**

  En esta tarea, los datos más importantes son recopilados en su totalidad para su procesamiento futuro. Al final de esta fase, se prepara un documento donde se detallan aspectos, los aspectos más relevantes sobre los datos, incluyendo las técnicas utilizadas para su recolección y los problemas presentados [15].

- **Descripción de los datos** Después de haber obtenido los datos primarios, se deben describir. Este proceso incluye contabilizar el volumen de datos (recuento de datos y atributos). Asimismo, se debe brindar una explicación sobre el significado de cada atributo [15].

- **Exploración de los datos** Esta tarea abarca la descripción estadística de los atributos de los datos. En esta descripción se obtienen tablas, gráficas, distribuciones de datos, etc. Una vez hecha la descripción de los datos, se procede a explorarlos, el propósito de esto es encontrar una estructura general para los datos. Implica aplicar pruebas estadísticas básicas para revelar las propiedades de los datos recién adquiridos, generar tablas de frecuencia y construir gráficos de distribución. Como resultado de esta tarea se obtiene un documento donde se describe un de análisis de los datos [15].

- **Verificar la calidad de los datos** En esta tarea, se realizan pruebas en los datos para determinar la consistencia de los valores de campo individuales, el número y la

distribución de ceros, y para encontrar valores fuera de rango que puedan convertirse en ruido para el proceso. La idea de cuando se llega a este punto es poder garantizar la integridad y exactitud de los datos [15].

3.2.3 Preparación de los datos

En esta etapa se contemplan las actividades relacionadas con la construcción de un conjunto de datos que pueda ser analizado por herramientas especializadas para minería de datos. Esta fase abarca aspectos como la sección, procesamiento, análisis gramatical, limpieza, construcción de nuevos datos, integración, y el formato de los datos obtenidos. Esta tarea se realiza de manera iterativa, ya que es muy probable que se deba revisar varias veces los datos antes de obtener un conjunto adecuado para proceder con la fase de modelado [15].

3.2.4 Modelado

En esta etapa se genera un modelo, el cual tenga la capacidad de brindar información útil para alcanzar los objetivos propuestos. En esta fase se deberá:

- ❒ Determinar una técnica de modelado apropiada para los datos obtenidos y los objetivos planteados
- ❒ Definir métricas para la evaluación de desempeño del modelo generado.
- ❒ Crear un modelo utilizando las técnicas previamente definidas sobre los datos.
- ❒ Adecuar el modelo generado a partir de los resultados obtenidos de sus métricas y su efecto en los objetivos del negocio.

3.2.5 Evaluación

En esta etapa se centra realizar una evaluación del modelo, analizando que tan cerca se encuentra de alcanzar los objetivos de negocio antes establecidos.

En esta fase contempla:

- ❒ Realizar una evaluación de modelo o modelos generados
- ❒ Realizar una retrospectiva del proceso de minería de datos que se ha realizado durante todo el tiempo.
- ❒ Establecer los siguientes pasos a seguir. Como el proceso de CRISP-DM es iterativo, esto puede implicar regresar a fases anteriores si el modelo no se ajusta a la realidad del negocio, o continuar hacia el despliegue si se cumplen las expectativas del modelo.

3.2.6 Despliegue o implementación

Esta fase se centra en implementar los resultados obtenidos en un ambiente real, de manera que pueda ser de utilidad en la toma de decisiones en la organización. En esta fase se definen varias tareas relativas al mantenimiento e implementación del modelo. Estas taras son:

- ❒ Diseñar un plan de despliegue de modelos
- ❒ Realizar la monitorización y mantenimiento
- ❒ Producir el informe final
- ❒ Revisar el proyecto en su totalidad

3.3 METODOLOGÍA DE DESARROLLO DE SOFTWARE XP

La Programación Extrema (Extreme Programing o XP) es una metodología ágil muy utilizada dentro del desarrollo de software. Esta metodología define cuatro actividades o fases principales: planeación, diseño, codificación y pruebas; así como las tareas y prácticas sugeridas para ejecutar en cada una de estas. Estas fases se muestran en la Figura 3.2.

3.3.1 Planeación

La planeación también es denominada juego de planeación, inicia con las actividades para la elicitación de requisitos, facilitando a que el equipo de desarrollo comprendan la problemática del negocio y se puede dar solución a la misma, mediante las características y

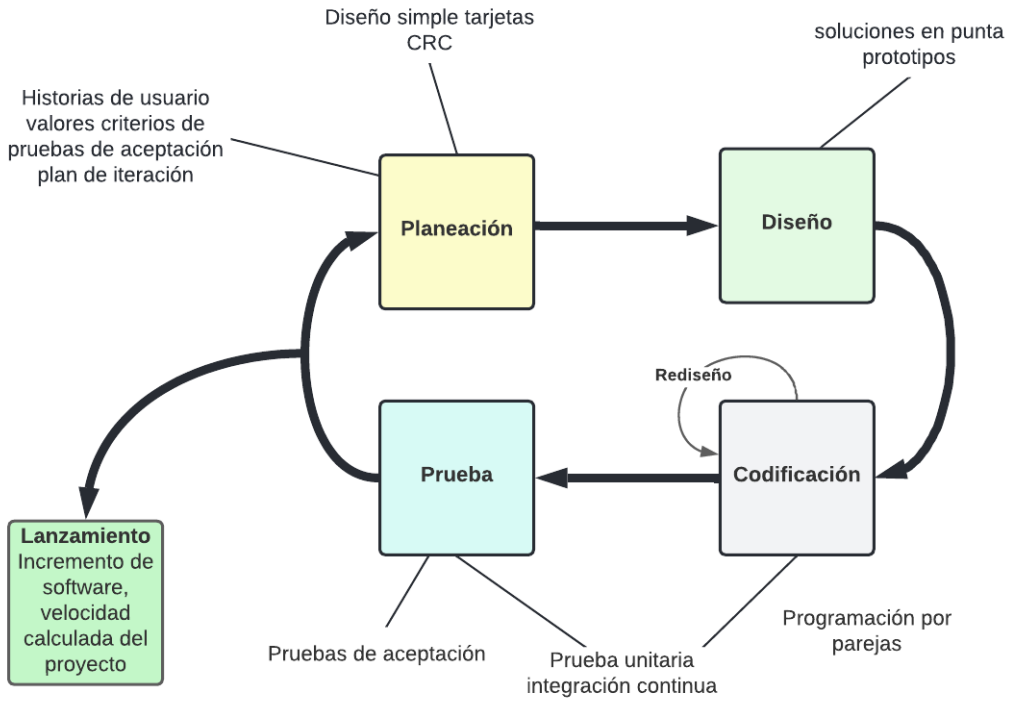


Figura 3.2: Fases de la programación extrema [16].

funcionalidades principales del sistema software. En esta fase se elaboran las historias de usuario que describen el valor de los requerimientos obtenidos, así como las funcionalidades del software a desarrollarse [16].

Las historias de usuario deben contener una prioridad la cual se asignará respecto al valor que asigne el cliente a la característica o función. Cada una de las historias de usuario son evaluadas y se les asigna un costo, el cual es medido en semanas de desarrollo, es importante mencionar que las historias de usuario puede ser modificadas o escribir más historias de usuario. Una vez escritas las historias de usuario se las debe agrupar y seleccionar el orden desarrollo, posteriormente se establece una fecha de entrega y también se debe incluir otros aspectos o detalles del proyecto. Una vez que se llega a un acuerdo, se establece la fecha de entrega, tomando en cuenta algunos factores fundamentales como el hecho de que todas las historias se implementan de forma inmediata, es decir, en pocas semanas y las historias con más valor serán implementadas primero [16].

Una vez se ha realizado la primera entrega, el equipo XP debe calcular cuantas historias pudieron ser desarrolladas en esta entrega o incremento. Esta velocidad ayudará a realizar una mejor planificación y gestión de actividades y fechas de entrega durante el resto de desarrollo. Es importante mencionar que a medida que el proyecto avanza, el cliente puede

añadir, modificar el valor de una historia existente o eliminar historias si es necesario. Si esto ocurre, el equipo de desarrollo debe estimar un nuevo tiempo de entrega para las historias faltantes o modificadas [16].

3.3.2 Diseño

El diseño XP se basa en en el principio de mantener un diseño sencillo. Mantener el diseño sencillo aporta más valor que un diseño complejo. Además, se debe considerar que el diseño guía la implementación de una historia de usuario y se debe no se debe considerar funcionalidades adicionales asumidas por el desarrollador [16]. En la metodología XP se promueve el uso de tarjetas CRC (Clase-Responsabilidad-Colaborador), como una herramienta para mantener el enfoque de orientación a objetos. Las tarjetas CRC deben contener información sobre la clase, responsabilidad y colaborador que permiten identificar y organizar las posibles clases y métodos que puedan ser implementadas en el software [16].

Es recomendable la utilización de un prototipo que permita entender de mejor manera el diseño. Este prototipo es evaluado teniendo como objetivo disminuir el riesgo en al momento de implementar el sistema real, a la vez que valida las estimaciones en la historia donde se complica entender el diseño. Es decir, tiene un rediseño, lo que involucra cambiar un sistema software de tal forma que no se modifique el comportamiento externo del código, pero se optimice la estructura interna. Realizando el rediseño se reducen las probabilidades de que se introduzcan defectos en el código [16].

3.3.3 Codificación

Cuando las historias de usuario han sido desarrolladas y se ha realizado un diseño previo, lo primero que se debe realizar antes de codificar, es diseñar pruebas unitarias, y a partir de estas, generar el código necesario para su funcionamiento. Cada prueba unitaria tiene como objetivo guiar al desarrollador a enfocarse en realizar solo lo necesario para pasar la prueba unitaria y no añadir ninguna funcionalidad extra y manteniendo el principio MS. Las pruebas unitarias brindan retroalimentación inmediata a los desarrolladores, agilizando el tiempo de codificación [16].

Durante la codificación se puede utilizar la programación es parejas, que es recomendada por esta metodología, ya que se tiene la premisa de que si dos personas desarrollan las

historias en conjunto, se agiliza este proceso debido a que se obtiene retroalimentación instantánea, se solucionan rápidamente los inconvenientes encontrados, además de que se aportan ideas para optimizar el código desarrollado. Finalmente, el código desarrollado se integra con el trabajo de equipo de desarrollo, utilizando la estrategia de “integración continua”para evitar los inconvenientes relacionados con la compatibilidad y descubriendo errores en etapas iniciales de desarrollo [16].

3.3.4 Pruebas

Dentro del enfoque XP, la realización de pruebas es un aspecto clave, ya que permiten verificar y validar el software de manera eficiente. Las pruebas deben tratar de automatizarse en la mayor medida de lo posible, de modo que puedan características funcionales generales del sistema [16].

4 REVISIÓN SISTEMÁTICA DE LA LITERATURA DE TÉCNICAS DE DETECCIÓN Y PREDICCIÓN DE SQLIA

Para la realización de este componente, un aspecto fundamental residió en la determinación del estado del arte en cuanto a las técnicas para la detección y predicción de SQLIA. Para ello, se realizó una revisión de literatura con respecto a este tema. A partir de los resultados obtenidos de esta revisión se procedió con el desarrollo del componente.

La revisión de literatura fue trabajada de manera conjunta entre todos los miembros del proyecto integrador. Esto fue debido a que como resultado de esta revisión, se obtuvieron los algoritmos que fueron evaluados individualmente en cada componente.

4.1 METODOLOGÍA

Para la realización de este estudio, se utilizó la metodología propuesta por Kitchenham en [17], cuyos pasos se detalla la sección 3.1.2. En este artículo, se describe en la sección una serie de pasos a seguir para llevar a cabo una revisión sistemática. Estos pasos se describen a continuación.

a. Preguntas de investigación

Las preguntas de investigación se realizaron con el objetivo de determinar las técnicas que existen actualmente para la detección y predicción de ataques de inyección SQL.

De esta manera se obtuvieron las siguientes preguntas de investigación:

RQ1 ¿Cuáles son las técnicas que se están utilizando para la detección y predicción de SQLIA?

RQ2 ¿Cuáles son las técnicas más utilizadas para detección y predicción de SQLIA?

RQ3 ¿Es posible clasificar las técnicas para la detección y predicción de SQLIA?

Mediante las preguntas RQ1 y RQ2 se realizó un análisis de publicaciones que proponen nuevas técnicas para la detección y predicción de SQLIA.

Para contestar la pregunta RQ3 se examinó los resultados obtenidos en las preguntas anteriores para proponer una clasificación de las técnicas identificadas.

b. Proceso de búsqueda

Para realizar la búsqueda se utilizó la siguiente cadena de búsqueda basándose en las preguntas de investigación planteadas:

(detection OR prediction) AND (SQLIA OR (SQL AND injection)) AND NOT (survey OR review)

Esta cadena fue adaptada de manera que cumpliera con las especificaciones de búsqueda de cada librería o base de datos utilizada. Sin embargo, el esquema general de los términos y conectores utilizados se mantuvo fijo en cada búsqueda.

c. Fuentes y bases de datos para la búsqueda

Para la búsqueda, se utilizaron las bases de datos más conocidas dentro del área de Ciencias de la Computación. Las librerías digitales utilizadas fueron:

- ACM Digital Library
- IEEE Xplore
- ScienceDirect
- SpringerLink

Posteriormente, se realizó una búsqueda más exhaustiva en el motor de búsqueda bibliográfica Google Scholar. Esta búsqueda se la realizó con el fin de obtener artículos que no hayan sido publicados en las librerías digitales consideradas inicialmente.

d. Criterios de inclusión y exclusión

Para la inclusión de un artículo se tomaron en cuenta los siguientes criterios de inclusión:

- Artículos que se hayan publicado entre el 1 de enero de 2012 y el 13 de abril de 2022

- ❒ Artículos cuyos títulos cumplieran la cadena de búsqueda considerada para la búsqueda
- ❒ Artículos publicados en conferencias o revistas especializadas

Una vez determinados los artículos que cumplieron con los criterios de inclusión, se utilizaron los siguientes criterios de exclusión para la selección de artículos:

- ❒ Artículos que traten sobre la detección de ataques de inyección SQL en tecnologías o entornos específicos.
- ❒ Artículos que traten sobre la detección de otros ataques, además de los ataques de inyección SQL
- ❒ Artículos que no propongan de técnicas específicas para la detección de ataques de inyección SQL. Por ejemplo, revisiones sistemáticas de la literatura o artículos científicos que presenten comparativas entre técnicas existentes.
- ❒ Artículos que no tengan DOI (Digital Object Identifier)
- ❒ Artículos que tengan menos 5 páginas de contenido sin tomar en cuenta la sección de referencias bibliográficas
- ❒ Artículos escritos en un idioma distinto al inglés

La aplicación de los criterios de exclusión se la realizó de manera manual realizando un escaneo sobre los artículos obtenidos

e. Selección de artículos La selección de artículos se la realizó en seis fases de búsqueda en las cuales se fueron aplicando los distintos criterios de inclusión y exclusión hasta llegar a los artículos que fueron seleccionados para formar parte de la revisión sistemática.

En la primera fase de búsqueda, se seleccionaron todos los resultados que incluyeran la cadena de búsqueda en cualquier lugar del artículo, ya sea en el título, resumen, contenido, metadatos, etc. En esta primera búsqueda se obtuvieron un total de 4048 resultados.

En la segunda fase de búsqueda se filtraron solo los resultados comprendidos entre el 1 de enero de 2012 y el 13 de abril de 2022. En esta búsqueda, los resultados disminuyeron a 2957 resultados.

En la tercera fase de búsqueda se seleccionaron los resultados en donde la cadena de búsqueda se encontrase solo en el título, así, se redujo el resultado de la búsqueda a 233 resultados.

En la cuarta fase de búsqueda se descartaron los resultados que no fuesen propiamente artículos científicos, por ejemplo, aquellos artículos que hayan sido publicados en revistas indexadas o conferencias especializadas. En esta fase se obtuvieron 198 artículos.

En la quinta fase de búsqueda, se procedió a filtrar los artículos duplicados. En este filtrado se logró obtener un total de 156 artículos.

En la sexta fase de búsqueda, fueron tomados los 156 artículos obtenidos hasta la quinta fase de búsqueda y se realizó un análisis manual de cada artículo, aplicando los criterios de exclusión definidos anteriormente. De esta búsqueda se obtuvieron finalmente 51 artículos que fueron seleccionados para la revisión.

En la Figura 4.1, se muestra de manera resumida el proceso de filtrado de los artículos mediante las diversas búsquedas en las que se fueron aplicando los criterios de inclusión y exclusión especificados anteriormente.

Tabla 4.1: Resumen del proceso de búsqueda y selección de artículos. Fuente: El autor.

Fase de búsqueda	Artículos por fuente	Total
Primera Fase	Science Direct: 213 SpringerLink: 973 IEEE Xplore: 274 ACM Digital Library: 1478 Google Scholar:1110	4048
Segunda Fase	Science Direct: 403 SpringerLink: 745 IEEE Xplore: 212 ACM Digital Library: 831 Google Scholar	2957
Tercera Fase	Science Direct: 213 SpringerLink: 973 IEEE Xplore: 274 ACM Digital Library: 1478 Google Scholar:766	233

Fase de búsqueda	Artículos por fuente	Total
Cuarta Fase	Science Direct: 8 SpringerLink: 12 IEEE Xplore: 53 ACM Digital Library: 28 Google Scholar: 132	198
Quinta Fase	Los artículos ya no se clasificaron según su fuente	156
Sexta Fase	Los artículos ya no se clasificaron según su fuente	51

f. Evaluación de calidad

Para la evaluación de calidad se utilizaron los criterios propuestos en [18]. Así, se definieron las preguntas descritas a continuación:

QA1 ¿El artículo describe los objetivos de investigación de manera clara?

QA2 ¿El artículo describe una revisión de literatura, antecedentes y contexto de investigación?

QA3 ¿El artículo muestra trabajos relacionados de trabajos anteriores para mostrar la principal contribución de la investigación?

QA4 ¿El artículo describe la arquitectura propuesta o la metodología usada?

QA5 ¿El artículo tiene resultados de la investigación?

QA6 ¿El artículo muestra conclusiones que son relevantes al propósito/problema de investigación?

QA7 ¿El artículo recomienda trabajo o mejoras a realizar para el futuro?

Los puntajes para cada pregunta varían entre 0,0.5 y 1 de acuerdo con lo indicado en el Anexo B

g. Extracción de datos y resultados

Los datos extraídos de cada estudio fueron:

- ❒ Información bibliográfica (título, año de publicación, conferencia o revista, autores)
- ❒ Número de citas en el Google Scholar

- ❒ Nombre o descripción de la técnica utilizada para la detección de SQLIA
- ❒ Clasificación a la que pertenece la técnica utilizada para la detección de SQLIA
- ❒ Algoritmo utilizado para la detección de SQLIA (si aplica)
- ❒ Tamaño y fuente del conjunto de datos utilizado para la aplicación de la técnica (si aplica)
- ❒ Tipos de SQLIA que abarca la técnica descrita

4.2 RESULTADOS

a. Resultados de búsqueda

Los 51 artículos seleccionados se describen en la el Anexo A. Por cada artículo se muestran: las citas obtenidas en Google Scholar, el año de publicación, el nombre de la técnica utilizada para la detección o predicción de SQLIA, la clasificación a la que pertenece la técnica utilizada, el o los algoritmos utilizados (si aplica), el tamaño y fuente del conjunto de datos utilizado para la evaluación de la técnica propuesta (si aplica), los tipos de SQLIA que abarca la técnica descrita. Los artículos fueron ordenados por el número de citas que obtuvieron en Google Scholar. Para los artículos que no cumplan alguno de los criterios se colocarán las iniciales “N/A”, indicando que no aplica el criterio en dicho artículo.

Basándose en la Figura 4.1 se observa que la técnica de Machine Learning es utilizada en 19 artículos para la detección y predicción de SQLIA, y las técnicas menos usadas son: Sistema de detección de intrusión, técnicas híbridas y otras técnicas. Con base en la lectura de los artículos se determinó una clasificación para las técnicas de detección y predicción de SQLIA. Esta clasificación se puede apreciar en la columna “Clasificación”, del Anexo A y se explica a mayor profundidad en el apartado **c.** de la Sección 4.3.

b. Resultados de la evaluación de calidad

En el Anexo C, se muestran los puntajes de la evaluación de calidad de los artículos analizados

Para esta revisión se determinó que todos los artículos con una calificación mayor a 5 puntos de 7 posibles, son considerados como artículos de alta calidad.

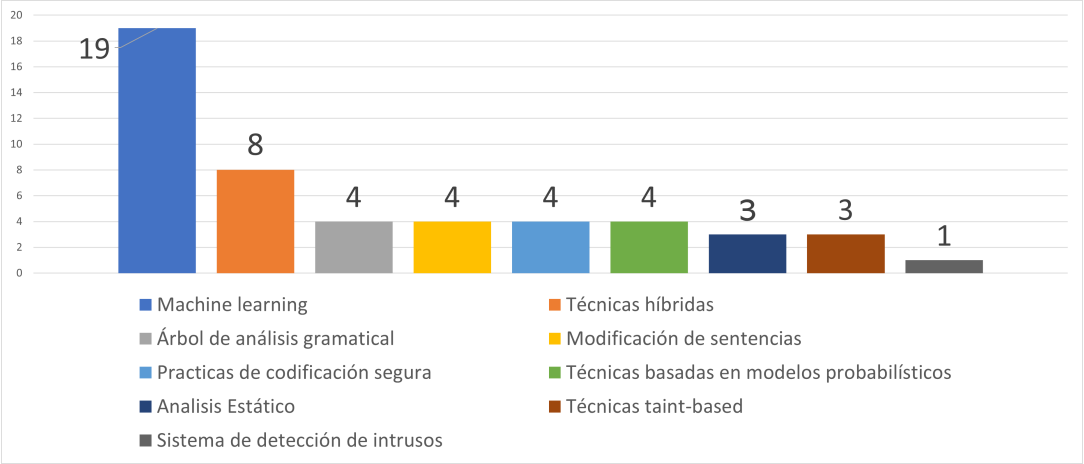


Figura 4.1: Distribución de clasificación de las técnicas de detección de SQLIA en los artículos analizados. Fuente: Los Autores

Como se puede observar en el Anexo C, el promedio de la evaluación de calidad es aproximadamente de 5.75, por lo que se puede determinar que de manera general, los artículos poseen una buena calidad.

En el Anexo C, se observa que a pesar de mantener una calidad relativamente alta, existen 4 artículos que muestran una baja calificación. Por contra parte, 12 artículos alcanzaron una calificación perfecta, lo que resulta en un buen indicador en cuanto a la calidad de los estudios realizados para proponer nuevas técnicas para la detección y predicción de SQLIA.

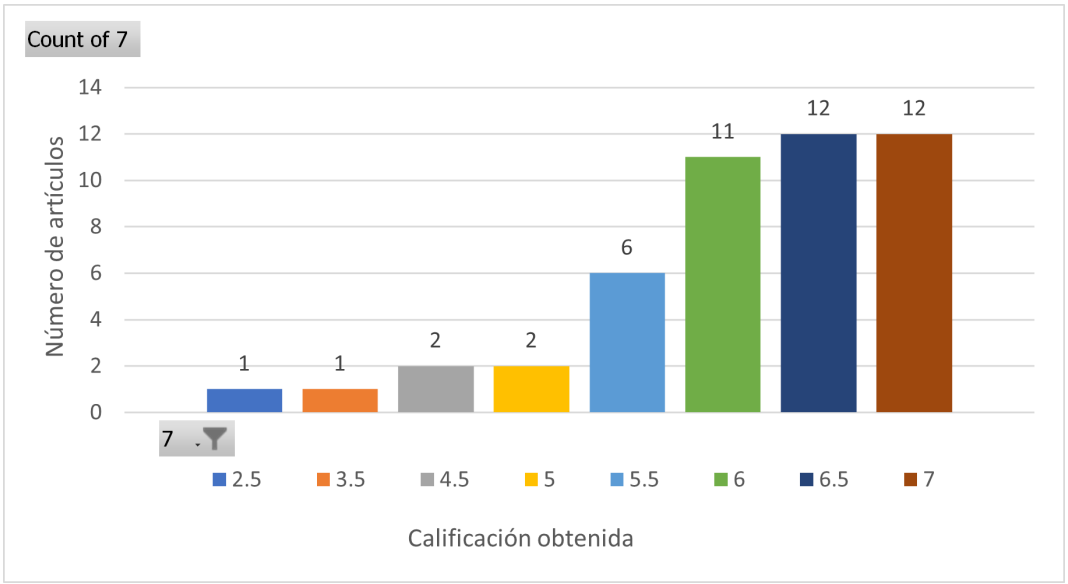


Figura 4.2: Distribución de calificaciones de la evaluación de calidad de los artículos analizados. Fuente: Los Autores

4.3 DISCUSIÓN DE LAS PREGUNTAS DE INVESTIGACIÓN

En esta sección se discuten las respuestas a las preguntas de investigación descritas en el apartado **a.** de la Sección 4.1

a. ***¿Cuáles son las técnicas más utilizadas para detección y predicción de SQLIA?***

De acuerdo con la investigación realizada, las técnicas con mayor impacto resultan ser aquellas que hacen usos de algoritmos de machine learning. Aproximadamente el 38 % de los artículos analizados utilizan algoritmos de machine learning.

b. ***¿Cuáles son las técnicas que se están utilizando para la detección y predicción de SQLIA?***

Para responder esta pregunta, se puede observar el Anexo A, donde se resumen las técnicas que han sido propuestas para la detección y predicción de SQLIA en los últimos 10 años.

c. ***¿Es posible clasificar las técnicas para la detección y predicción de SQLIA?***

Como se observa en el Anexo A se realizó una clasificación de las técnicas para la detección y predicción de SQLIA. En este estudio se determinó la siguiente clasificación:

- **Análisis estático:** los enfoques estáticos detectan o contrarrestan la posibilidad de un ataque de inyección SQL en la fase de compilación. Este enfoque se centra en escanear la aplicación y aprovechar el análisis del flujo de información para detectar los códigos que podrían tener vulnerabilidades [19].
- **Modificación de sentencias:**esta técnica se centra en reconstruir las consultas en tiempo de ejecución utilizando una clave criptográfica que es inaccesible para los atacantes. Esta técnica permite a los desarrolladores crear consultas SQL utilizando palabras clave aleatorias en lugar de normales, donde un proxy entre la aplicación web y la base de datos intercepta las sentencias SQL y desaleatoriza las palabras clave [19].
- **Árbol de análisis gramatical:** esta técnica comprueba en tiempo de ejecución si las consultas entrantes se ajustan a un modelo de consulta esperado. El modelo se decide en tiempo de ejecución donde examina las estructuras de la consulta

antes y después de las peticiones del cliente, es decir, se encarga de asegurar las sentencias SQL vulnerables comparándolo con un árbol de análisis sintáctico de una sentencia con el de la original y únicamente permitirá que se ejecute una sentencia con una comparación coincidente [19].

- **Técnicas Taint-based:** esta técnica aplica varias políticas de seguridad marcando los datos no fiables y rastreando sus flujos a través de los programas mediante un análisis sensible y minucioso al contexto para rechazar las consultas SQL si estas tienen una entrada no fiable [19].
- **Técnicas basadas en modelos probabilísticos:** las técnicas basadas en modelos probabilísticos se los realiza en tiempo de ejecución, donde se asume que el valor de una sentencia SQL está relacionada con la presencia o ausencia de vulnerabilidades en su estructura y de esta manera permitir la detección de un ataque de inyección SQL [20].
- **Sistemas de detección de intrusos:** los sistemas de detección de intrusos se basan en una técnica de aprendizaje automático que se entrena utilizando un conjunto de consultas típicas en aplicaciones web. La técnica empieza construyendo modelos de las consultas típicas y luego las supervisa las consultas que ingresan a la aplicación en tiempo de ejecución para identificar las consultas que no coinciden con el modelo construido [20].
- **Técnicas híbridas:** algunas técnicas combinan un análisis estático durante el desarrollo con la combinación de una supervisión dinámica en tiempo de ejecución [19], como tal es el caso de AMNESIA [21], que asocia un modelo de consulta con la ubicación de cada consulta en la aplicación y luego monitoriza la aplicación para detectar si alguna consulta se desvía del modelo esperado [20].
- **Prácticas de Codificación Segura:** las principales vulnerabilidades de inyección SQL se deben a la insuficiente validación de las entradas. Por lo tanto, la solución directa para eliminar estas vulnerabilidades es aplicar prácticas de codificación segura. Algunos ejemplos de las mejores prácticas son: comprobar el tipo de entrada en la consulta, codificación de las entradas, coincidencias positivas de patrones e identificar todas las fuentes de la entrada [20].
- **Técnicas basadas en Machine Learning:** las técnicas basadas en Machine Learning consisten en utilizar diferentes clasificadores, para detectar en una sentencia SQL los posibles ataques mediante la clasificación de los datos [22], es

decir, separar las sentencias SQL en dos grupos que contienen una etiqueta que identifique si son o no ataques. Dependiendo del clasificador que se utilice, los resultados para su detección pueden verse afectados. Unos ejemplos de estos clasificadores pueden ser Naive Bayes, Redes Neuronales Artificiales, Perceptrón Multicapa, etc.

4.4 CONCLUSIONES DE LA REVISIÓN SISTEMÁTICA DE LA LITERATURA

A partir de esta revisión, se pudo realizar una clasificación de los diferentes tipos de técnicas de detección de SQLIA, donde se encontró una cierta prevalencia en las técnicas que utilizan Machine Learning, particularmente los algoritmos más utilizados en esta área fueron las Redes Neuronales Artificiales (Artificial Neural Networks o ANN por sus siglas en inglés), Las Máquinas de Vectores de Soporte (Support Vector Machine o SVM por sus siglas en inglés), el Perceptrón Multicapa, y el algoritmo Naive Bayes. La gran mayoría de los artículos que fueron evaluados obtuvieron una buena calificación en la evaluación de calidad, lo que indica que se ha realizado una investigación exhaustiva, con un marco metodológico bien definido y con resultados confiables. Es importante notar el avance que ha existido en cuanto a investigaciones en el campo de las técnicas para la detección y predicción de SQLIA, ya que como se puede apreciar, existe mucha información en cuanto a la investigación dentro de esta área. En un futuro es posible profundizar en otros aspectos como los SQLIA en entornos específicos u otros ataques similares como podrían ser los ataques Cross Site Scripting (XSS). Como parte de esta investigación se pudo determinar los algoritmos más utilizados para la detección y predicción de SQLIA, y a partir de estos, realizar un análisis más a profundidad, evaluándolos con cantidades masivas de datos, utilizando datos reales en un caso de estudio, como es lo que se realizará en el desarrollo de este componente

5 DESARROLLO E IMPLEMENTACIÓN

En esta sección se describe el proceso de minería guiada por la metodología CRISP-DM para la obtención de un modelo que permita definir la efectividad de una red neuronal recurrente para la detección de ataques de inyección SQL en BIG DATA. Además, en esta sección se desarrollará un prototipo que permitirá monitorear e identificar ataques de inyección SQL en los sistemas de una organización, utilizando la metodología XP.

5.1 METODOLOGÍA DE MINERÍA DE DATOS CRISP-DM

A continuación se muestra la ejecución de la metodología CRISP-DM en el tratamiento de la información proporcionados por la organización en cada una de sus fases:

5.1.1 Comprensión del negocio

Para la comprensión del negocio se identificaron las expectativas de la organización (CSIRT-EPN) con respecto a la implementación de algoritmos que serán utilizados para la minería de dados con la finalidad de detectar ataques de inyección SQL.

a. Determinación de los objetivos comerciales

El principal objetivo de la CSIRT-EPN es contar con una herramienta que permita detectar posibles ataques inyección SQL realizados a los distintos sistemas informáticos pertenecientes a la organización, de manera que se facilite la toma de decisiones y ayuden a mitigar los ataques de inyección SQL.

b. Evaluación de la situación

Para evaluar la situación de la CSIRT perteneciente a la EPN es necesario e importante tomar en cuenta los siguientes factores que pueden afectar la minería de datos:

- ❒ **Personal:** La organización cuenta con personal capacitado y experimentado en la compresión y manejo de los sistemas de información pertenecientes a la organización (CSIRT-EPN), quienes ayudaron solventar dudas correspondientes al acceso, manejo de información y a la manipulación de la base de datos.
- ❒ **Datos:** Los datos provistos por la organización para la minería de datos provienen de una muestra de diversos registros generados por los sistemas de información utilizados por la organización.
- ❒ **Riesgos:** El principal riesgo presente en el proyecto está relacionado con el manejo de información confidencial, por lo que se debe asegurar la no divulgación de la información. Durante el proceso de minería de datos se debe tomar en cuenta los riegos relacionados con la planificación en un tiempo corto para el desarrollo del proyecto.

c. Determinación de los objetivos de minería de datos

El proyecto debe tener como resultado final un modelo que permita clasificar los datos provenientes de diversas fuentes, tomando registros de sentencias SQL. La información puede clasificarse en dos tipos: Datos con posibles anomalías y datos sin anomalías. Además, este modelo debe reducir la cantidad de falsos negativos posibles.

d. Producción de un plan de proyecto

En la realización del plan de proyecto se elaboró un cronograma con cada una de las fases de la metodología, con un tiempo estimado de 4 semanas. En este cronograma se especifican los tiempos, recursos, y riesgos asociados en cada fase. El plan del proyecto se puede visualizar en la Tabla 5.1

Tabla 5.1: Plan de proyecto de minería de datos aplicando las fases de la metodología CRISP-DM

Fase	Tiempo	Recursos	Riesgos
Comprensión del negocio	1 semana	Miembros de la CSIRT	Disponibilidad de expertos en el negocio
Comprensión de los datos	2 semanas	Miembros de la CSIRT	Anomalías en la estructura de los datos entregados por la
Preparación de los datos	3 semanas	Equipo de desarrollo	Entendimiento de los datos proporcionados por la CSIRT

Fase	Tiempo	Recursos	Riesgos
Modelado	2 semanas	Equipo de desarrollo	Dificultad en la manipulación de tecnología para la creación de modelos
Evaluación	1 semana	Miembros de la CSIRT y equipo de desarrollo	Problemas en los entrenamientos de los modelos.
Despliegue	1 semana	Equipo de desarrollo	Recursos de computación limitados

5.1.2 Comprensión de los datos

Para la compresión de los datos se realizó un estudio de los datos provistos para identificar los atributos más importantes en los datos proporcionados por la organización (CSIRT-EPN), de esta manera se tiene una idea clara de la minería de datos.

a. **Recopilación de datos iniciales**

Para la identificación y recolección de datos disponibles se lo realizó a partir de un extracto de un log de sentencias SQL, facilitados por la organización. Estos datos fueron entregados en un archivo de extensión JSON y con un tamaño de 2.5 GB.

b. **Descripción de los datos**

La estructura de logs proporcionados por la organización se muestra a continuación:

```
...
"log": {
"file": {
"path": "...",
},
"offset": "...",
"flags": ["..."]
},
"fileset": {
```

```
"name": "...",

},
"message":
"timestamp=... ,process_id=... ,session_number=... ,user=... ,db
=... ,app=....,client=... ,LOG: duration: ... ms bind ...:
SELECT...
timestamp=... ,process_id=...,session_number=...,user=...,db=...,
app=...,client=...,DETAIL ...
...
...
...",
"fileset":{
"name": "...",
},
"error": {
"message":"Provided Grok expressions do not match field value:[
timestamp=... ,process_id=... ,session_number=... ,user=... ,db
=... ,app=....,client=... ,LOG: duration: ... ms bind ...:
SELECT...
timestamp=... ,process_id=...,session_number=...,user=...,db=...,
app=...,client=...,DETAIL ...
...
...
...]",
}
},
"input": {
"type": "...",
},
...
```


En la estructura JSON se puede observar la clave "message", y su valor representan los diversos logs para este estudio, también se muestra los datos referentes a la consulta SQL registrada, esta consulta está dividida, en un apartado de consulta y parámetro. El parámetro es de suma importancia, ya que este apartado puede contener información de posibles ataques de SQL inyección.

d. **Verificación de calidad de datos**

   Luego de analizar la estructura de un log, se pudo determinar un formato consistente, por lo que se puede realizar una limpieza de la información para la obtención de los atributos más importantes para predicción e identificación de ataques de inyección SQL.

5.1.3 Preparación de los datos

Una vez comprendida la estructura de la muestra de logs proporcionada por la organización, se realiza el proceso de minería de datos, para poder crear un modelo de manera efectiva, para este proceso se debe realizar la limpieza de datos, formato e integración que permitan estandarizar la información.

a. **Selección de datos**

   La selección de los campos relevantes para la minería de datos son las consultas de SQL con sus parámetros en donde se podrá identificar anomalías relacionadas con los ataques de inyección SQL. Además, es importante mencionar que se realizó la unión de las consultas SQL con sus respectivos parámetros.

b. **Limpieza de datos**

   Para la limpieza de los datos se tomaron los registros de la muestra proporcionada por la organización, este logpasó por un proceso de parseo de cada uno de los registros. Dentro de este Log, los parámetros se encuentran separados de la consulta para su procesamiento.

c. **Construcción de nuevos datos** Para poder obtener un formato estándar de consulta SQL, los parámetros fueron insertados en sus consultas respectivas, para que posteriormente puedan ser analizadas en las siguientes fases.

d. **Integración de datos**

   Los datos proporcionados para esta investigación provienen de una sola fuente, por lo cual no es necesario la integración con otras fuentes.

e. **Formato de datos**

   Las consultas completadas e integradas fueron almacenadas de manera ordenada en un archivo CSV para su posterior análisis.

Como resultado de esta fase se obtuvo un archivo que contiene 3 millones de registros de sentencias SQL almacenados en un archivo de extensión CSV.

5.1.4 Modelado

A partir de la revisión de la literatura realizada en el Capítulo 4, se obtuvieron algunos de los algoritmos y tecnologías más utilizadas para la detección de ataques de inyección SQL, entre las cuales se destacan las Redes Neuronales Artificiales, el cual será seleccionado para la evaluación de rendimiento y efectividad en este proceso.

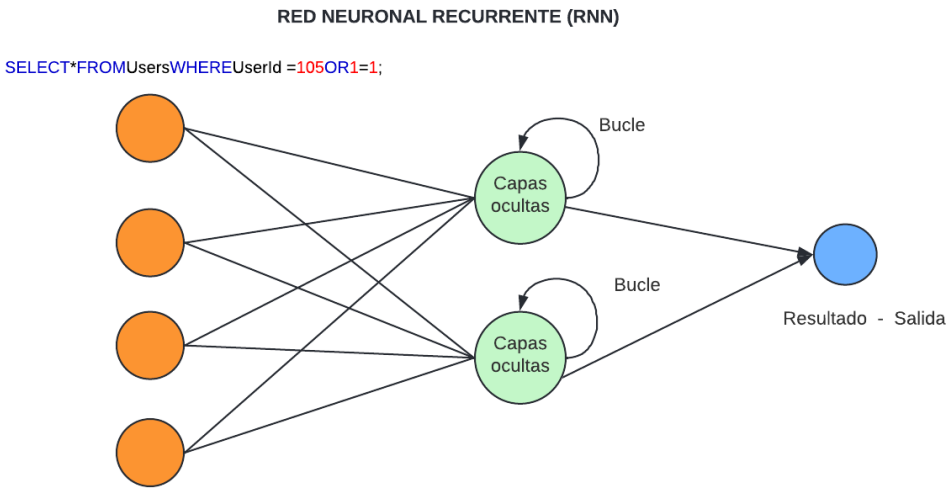


Figura 5.1: Red neuronal recurrente - RNN. Fuente: El Autor

❒ **Selección de técnica de modelado**

Uno de los principales problemas y desafíos en la detección y prevención de ataques de inyección SQL es la variedad de consultas SQL maliciosas, por lo cual es importante estudiar y diseñar una solución basada en aprendizaje profundo mediante la detección automática. Además, una de las principales características es que los enfoques basados en redes neuronales profundas no requieren de la extracción manual de características. Entonces, es importante notar que la entrada de la Red Neuronal recurrente es una secuencia de palabras, las cuales son tomadas y almacenadas. Para la detección de una cadena que puede resultar en un ataque de inyección SQL, se realiza la predicción de la siguiente palabra, y se arma una frase con sentido dependiendo de los caracteres anteriores de una frase, como se muestra en la Figura 5.1. Al final de la red neuronal se obtiene un resultado, el cual es un número entre 0 y 1, y

si este es mayor a un porcentaje establecido, la cadena de texto es identificado como posible ataque de inyección SQL.

❒ Generación de un diseño de comprobación

Como paso final antes de generar el modelo, se debe tomar en cuenta la forma de comprobar los resultados de los modelos. Para lo cual se va a establecer métricas de evaluación para dicho modelo. Para calcular las métricas de evaluación se debe obtener la matriz de confusión que permitirá observar el rendimiento del modelo supervisado de Machine Learning utilizando los datos de prueba. Además, las métricas permitirán identificar donde el modelo está confundiendo dos clases de sentencia SQL. Para generar un diseño de comprobación se debe establecer, cuáles son los resultados arrojados por la matriz de confusión, estos se muestran en la Figura 5.2.

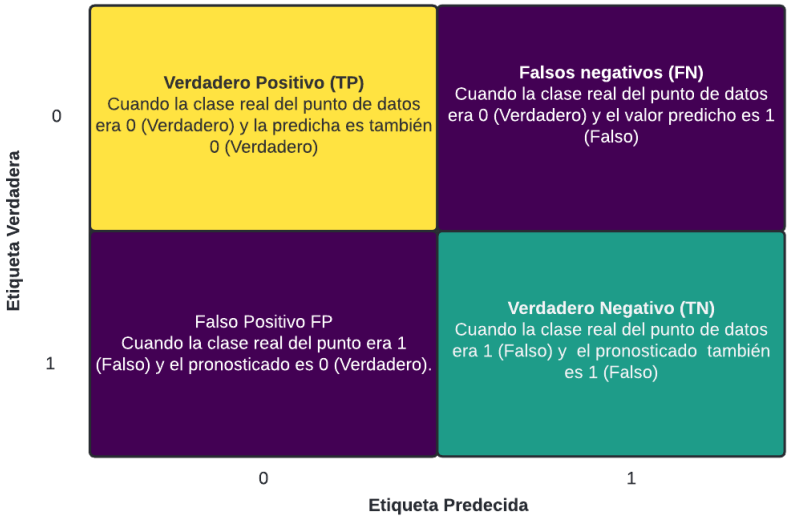


Figura 5.2: Matriz de confusión. Fuente: El Autor

❒ Generación de modelos

Para la entrenar los modelos se tomó un conjuntos de datos obtenidos de diversas fuentes los cuales fueron etiquetados de la siguiente forma :

Tabla 5.2: Etiquetado de cadenas SQL

Secuencia de texto	Descripción	Etiqueta	Cantidad
Sentencias Sql Normales	Sentencias no anómalas que usadas para la consulta de información	1	11382
Sentencias Sql anómalas	Sentencias que contiene código SQL	0	19537

❒ **Evaluación del modelo**

Como se muestra en la Tabla 5.2, el porcentaje de ataques de inyección SQL presentes en el conjunto de datos utilizado para el entrenamiento del modelo es de 61.19 % , y el 38.81 % son consultas SQL sin anomalías utilizadas para la obtención de información de la base de datos de la organización. Además, para el entrenamiento se tomó un 70 % del total del conjunto de datos y el restante 30 % para el entrenamiento.

5.1.5 Evaluación

Una vez finalizado el entrenamiento del modelo, se obtiene la matriz de confusión a partir del 30 % de total de conjunto de datos. A partir Figura 5.3, se puede obtienen los siguientes resultados:

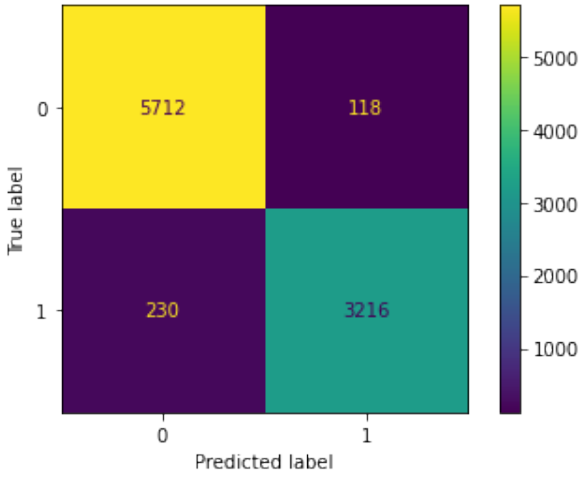


Figura 5.3: Matriz de confusión del Modelo-RNN. Fuente: El Autor

A partir de la Tabla 5.3, se determinan las siguientes métricas:

Tabla 5.3: Resultados de la matriz de confusión - Modelo RNN

Verdaderos Positivos	TP= 5712	Falsos Negativos	FN= 118
Falso Positivos	FP = 230	Verdadero Negativo	TN=3216

❒ **Exactitud (A):** El porcentaje total de elementos clasificados correctamente.

$$A = \frac{TP - TN}{TP + FP + FN + TN} \tag{5.1}$$

A continuación se realiza el cálculo de la Exactitud utilizando la Ecuación 5.1 y la matriz de confusión de la Figura 5.3:

$$A = \frac{5712 + 3216}{5712 + 230 + 118 + 3216}(100\,\%) = 96.24\,\%$$

❒ **Sensibilidad o Tasa de verdaderos positivos (TPR):** Es el número de logs identificados correctamente como logs positivos del total de positivos verdaderos.

$$TPR = \frac{TP}{TP + TN} \tag{5.2}$$

A continuación se realiza el cálculo de Sensibilidad utilizando la Ecuación 5.2 y la matriz de confusión de la Figura 5.3:

$$TPR = \frac{5712}{5712 + 3216}(100\,\%) = 63.97\,\%$$

❒ **Precisión (P)**: Es el número de logs identificados correctamente como positivos de un total de logs identificados como positivos.

$$P = \frac{TP}{TP + FP} \tag{5.3}$$

A continuación se realiza el cálculo de la Precisión utilizando la Ecuación 5.3 y la matriz de confusión de la Figura 5.3:

$$P = \frac{5712}{5712 + 230}(100\,\%) = 96.12\,\%$$

❒ **Especificidad o Tasa negativa verdadera (TNR):** Es el número de ítems correctamente identificados como negativos fuera del total de negativos.

$$TNR = \frac{TN}{TN + FP} \tag{5.4}$$

A continuación se realiza el cálculo de la Especificidad utilizando la Ecuación 5.4 y la matriz de confusión de la Figura 5.3:

$$TNR = \frac{3216}{3216 + 230}(100\,\%) = 93.32\,\%$$

❒ **F1-Score:** Esta métrica es la media armónica entre la Precisión y la Sensibilidad. Esta

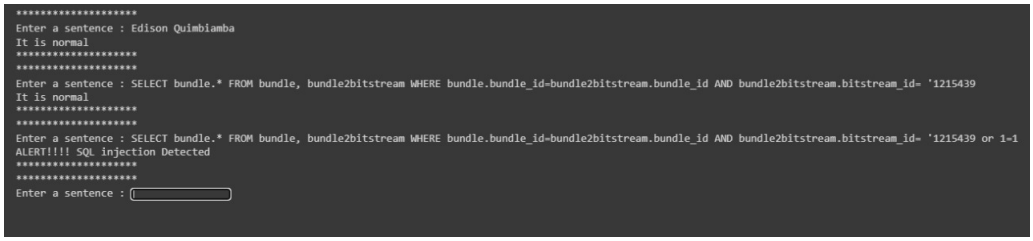


Figura 5.4: Evaluación simple. Fuente:El Autor

métrica indica la relación que existe entre dichas métricas y que tan equilibradas se encuentran.

$$F1 - Score = \frac{2 * TPR * P}{TPR + P} \tag{5.5}$$

A continuación se realiza el cálculo del F1-Score utilizando la Ecuación 5.5 y la matriz de confusión de la Figura 5.3:

$$F1 - Score = \frac{2 * 0.6397 * 0.9612}{0.6397 + 0.9612}(100\,\%) = 76.81\,\%$$

Una vez entrenado el modelo se realiza comprobaciones tanto con sentencias SQL normales, y ataques de SQL inyección, como se muestra en la Figura 5.4.

Para finalizar la fase de evaluación se tomó una muestra de un 1 millón de registros proporcionados por la organización para ser procesados mediante el modelo obtenido a partir RNN, y se obtuvieron los siguientes resultados que se muestran en la Tabla 5.4.

Tabla 5.4: Resultados de análisis de las muestras

Cadena de texto	Cantidad	Porcentaje
Sentencias SQL Normales	974600	97.46 %
Sentencias SQL anómalas	25400	2.54 %

5.1.6 Despliegue

En esta última fase se la realiza luego de la evaluación y validación del modelo. En este proyecto se propone desarrollar un aplicativo web que muestre los resultados obtenidos al analizar una muestra de logproporcionada por la organización. Para el desarrollo de un prototipo web se ha seleccionado la metodología XP.

5.2 DESARROLLO DE SOFTWARE CON LA METODOLOGÍA XP

Para el desarrollo del sistema web se ha realizado una identificación de las principales funcionalidades y características que debe tener este sistema. Comprender estos aspectos al momento de implementar el aplicativo web es de vital importancia, ya que de esto dependerá la facilidad con la que los usuarios manipularán el sistema.

5.2.1 Planeación

5.2.1.1 Historias de usuario

Para desarrollo del aplicativo web se ha realizado una identificación de las principales funcionalidades y características que debe tener este sistema. Comprender estos aspectos al momento de implementar el aplicativo web es de vital importancia, ya que de esto dependerá la facilidad con la que los usuarios manipularán el sistema. Una vez identificados, los requerimientos se han estructurado en forma de historias de usuario y posteriormente han sido clasificadas, por prioridad, para lo cual se ha considerado 1 como máxima prioridad y 4 como mínima prioridad, como se muestra en la Tabla 5.5.

5.2.1.2 Plan de entrega del proyecto

En este apartado se define el plan de entrega del proyecto, tomando como referencia la prioridad de las historias de usuario de la Tabla 5.5. Se ha asignado una semana de desarrollo para cada historia de usuario. El plan de entrega del proyecto se muestra en la Tabla 5.6.

Finalmente, se realiza la asignación de parejas (dos miembros del equipo de desarrollo), para la codificación y la asignación de tareas para cada iteración, como se muestra en la Tabla 5.7.

Tabla 5.5: Historias de usuario

Código	Título	Descripción	Prioridad
HU - 01	Control de acceso	Como administrador deseo poder mantener un control de acceso dentro del sistema para evitar la divulgación de información sensible de la organización	1
HU-02	Carga de datos	Como usuario deseo poder realizar la carga de un log de cualquier tamaño, para que pueda ser procesado y limpiado, de manera que pueda ser utilizado para detectar sentencias de ataque de inyección SQL	2
HU - 03	Análisis	Como usuario deseo evaluar diferentes modelos de Machine Learning para detectar ataques de inyección SQL	3
HU - 04	Visualización de logs anómalos	Como administrador del sistema, deseo visualizar las sentencias SQL detectadas como posibles ataques de inyección SQL para aumentar la seguridad de los sistemas de donde se obtuvo la información	4

Tabla 5.6: Plan de entrega de proyecto

Código historias de usuario	Iteración	Prioridad	Duración en semanas
HU - 01	1	1	1
HU - 02	1	2	1
HU - 03	2	3	1
HU - 04	2	4	1

Tabla 5.7: Asignación de parejas y responsabilidades.

Miembros	Número de parejas	Historias de usuario
Edison Quimbiamba, Andrés Palma	1	HU-01, HU-03
Steven Rivera, Andrés Llumiquinga	2	HU-02, HU-04

5.2.2 Diseño

5.2.2.1 Arquitectura del sistema software

El prototipo web será creado utilizando la arquitectura Cliente - Servidor, que permite que el cliente pueda interactuar con el servidor por medio de solicitudes HTTP, como se observa en la Figura 5.5. Además, para el prototipo está basado en el patrón de arquitectura Modelo Vista Controlador (MVC), que nos ayudará para separar los datos de lógica. También este patrón de arquitectura MVC, propone tres componentes que son, la vista, el controlador,

el modelo. Este patrón facilita la interacción del usuario con los componentes y también permite representación de la información de manera sencilla, además reduce la complejidad del desarrollo de software y su posterior mantenimiento.

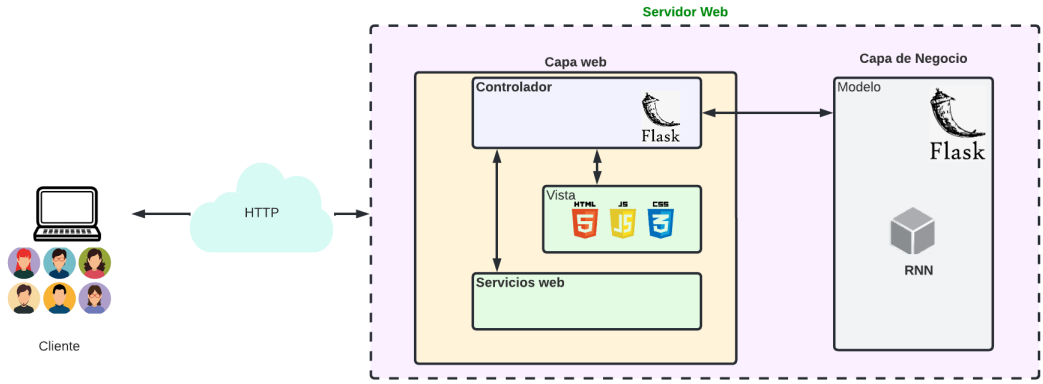


Figura 5.5: Arquitectura MVC del prototipo web. Fuente: Los Autores.

5.2.3 Diagrama de actividades

Para explicar el flujo de actividades para la manipulación del sistema, se realizó el modelado de comportamiento del sistema mediante un diagrama de actividades, como se visualiza en la Figura 5.6.

5.2.3.1 Diseño interfaces

En esta fase tiene por objetivo asegurar que la interacción del usuario con el sistema se puede efectuar de forma intuitiva, para lo cual se han desarrollado las siguientes interfaces

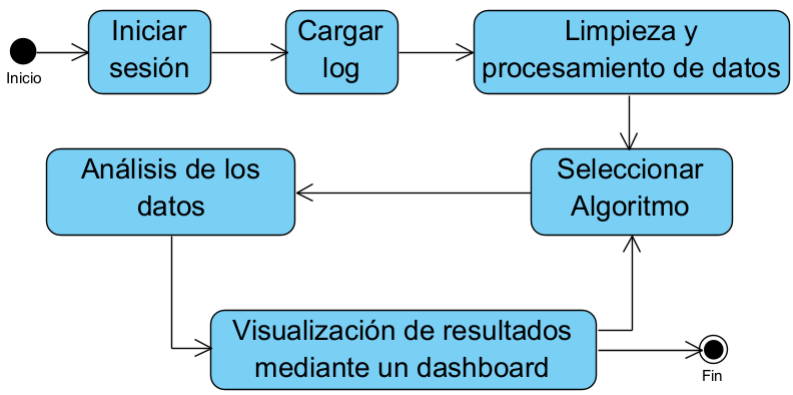
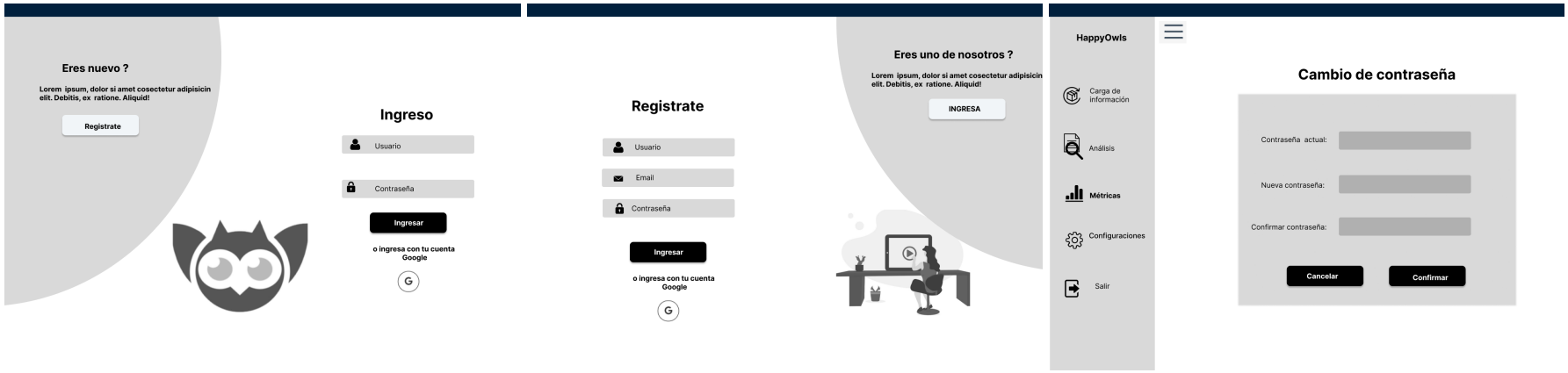


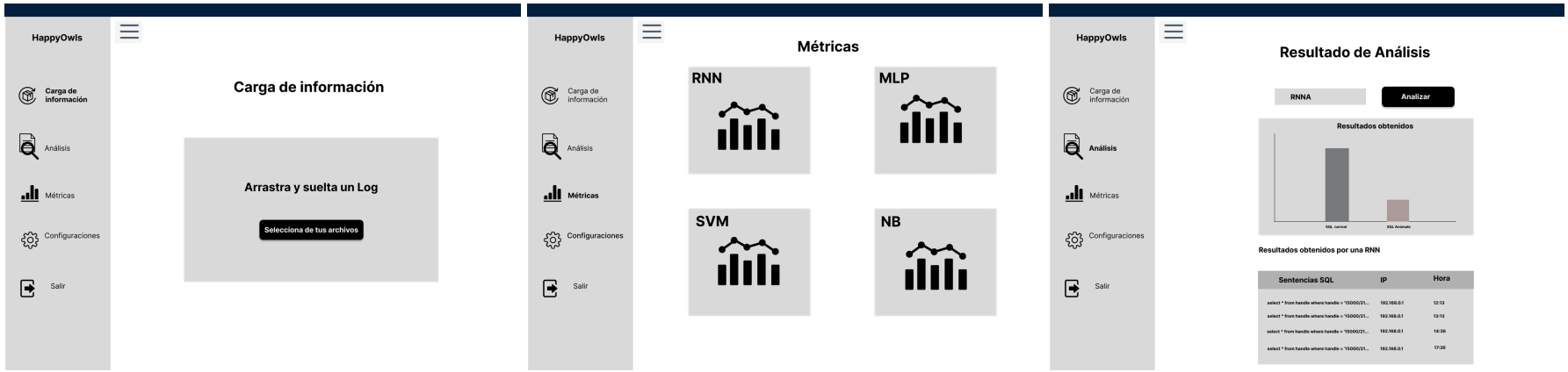
Figura 5.6: Diagrama de actividades. Fuente: Los Autores.

de baja fidelidad, consideradas de esta manera debido a que no representan una visión detallada de la interfaz, sino más bien la distribución de los elementos en esta. Para el desarrollo de la interfaz, estas se realizaron con base en los requerimientos establecidos en la Tabla 5.5. En la Figura 5.7, se muestran las interfaces relacionadas con el control de acceso del sistema web y en la Figura 5.8, se contempla las vistas relacionadas con los módulos de análisis de datos. Una vez definidas los prototipos de baja fidelidad (visión inicial del aplicativo web), con el fin de aclarar los requerimientos para la construcción del aplicativo web, en el siguiente capítulo se describe la fase de codificación.



(a) Login **(b)** Registro **(c)** Cambio de contraseña

Figura 5.7: Interfaces para control de acceso



(a) Carga de información **(b)** Métricas de algoritmos **(c)** Análisis de los datos

Figura 5.8: Módulos del prototipo web. Fuente: Los Autores

5.2.4 Codificación

Una vez definido los requerimientos y la arquitectura del aplicativo web, se procede a la construcción del mismo. Según la planificación definida en las fases anteriores, se considerarán 2 iteraciones: la primera iteración, la cual está relacionada con la configuración del control de acceso y la segunda iteración en la cual se construirán los módulos del análisis de datos.

5.2.4.1 Primera iteración

Esta iteración tiene como objetivo generar un incremento funcional para que los usuarios puedan registrarse e iniciar sección, de forma que se pueda controlar el acceso de los usuarios a la información.

En la Tabla 5.8 se presenta una lista de tareas relacionadas con las historias seleccionadas para esta iteración.

Tabla 5.8: Lista de tareas para la primera iteración

Id / **Pareja**	HU - 01 / Pareja 1	HU - 02 / Pareja 2
Historia de usuario	Como administrador deseo poder mantener un control de acceso dentro del sistema para evitar la divulgación de información sensible de la organización.	Como usuario deseo poder realizar la carga de un logde cualquier tamaño, para que pueda ser procesado y limpiado de manera que pueda ser utilizado para detectar sentencias de ataque de inyección SQL.
Tarea 1	Crear ambiente de desarrollo y configuraciones necesarias para Flask con sus respectivas dependencias.	Diseñar los prototipos para las vistas del módulo de carga de datos.
Tarea 2	Diseño de prototipos para la vista de login del sistema y su implementación en HTML.	Desarrollar las funciones para carga y limpieza de información.
Tarea 3	Desarrollar el controlador para el login.	Desarrollar funciones para limpieza de datos.

En la Figura 5.9 se observa la implementación del login para el control de acceso del sistema. De igual forma, en la Figura 5.10, se muestra la implementación del módulo de carga de información, en el cual el usuario sube un logcomprimido y posteriormente es limpiado automáticamente por el sistema. Esta funcionalidad tiene como objetivo ahorrar el tiempo del usuario al cargar la información, también es importante mencionar que al subir un archivo comprimido su peso es menor.

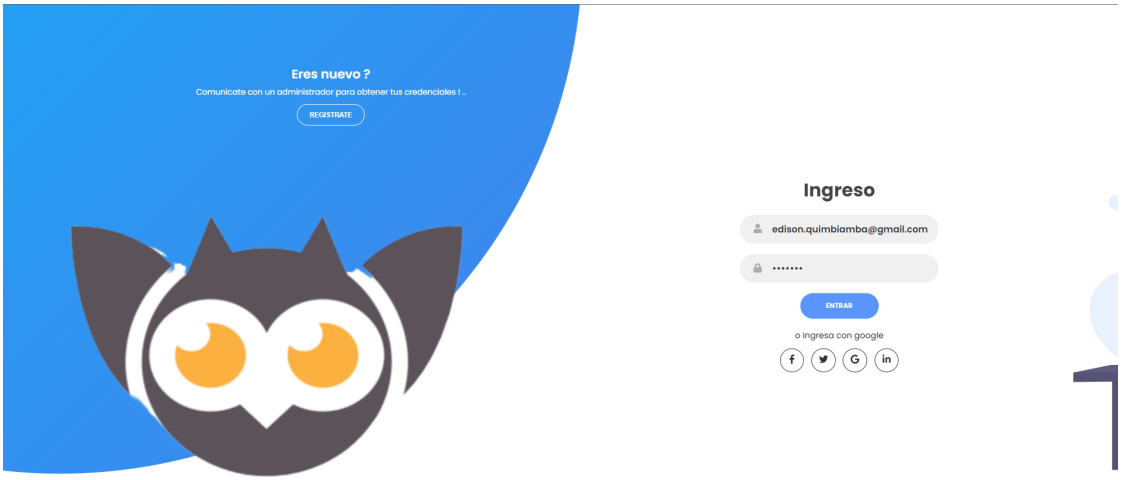


Figura 5.9: Implementación del control de acceso del sistema. Fuentes: Los Autores

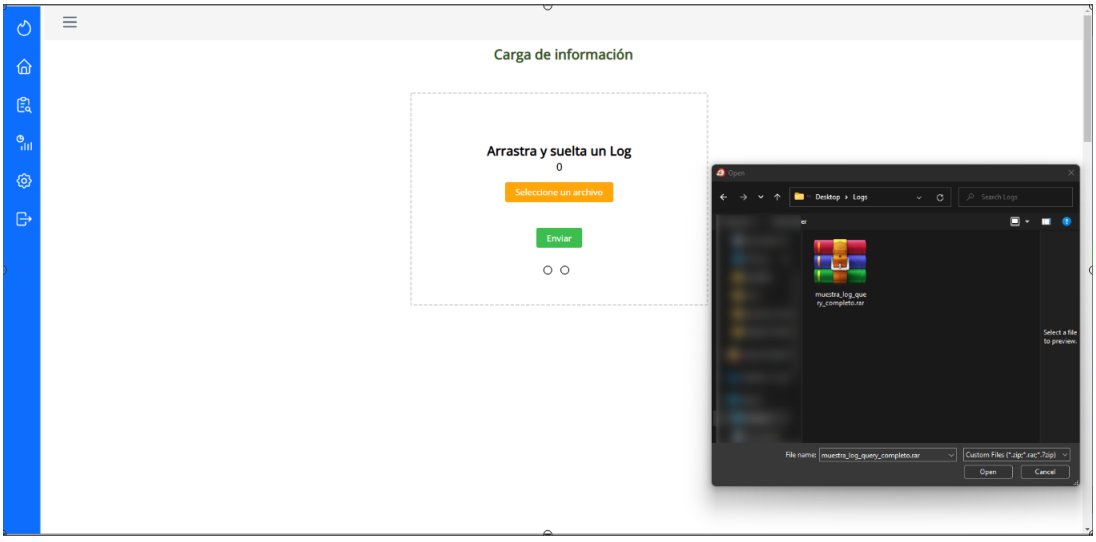


Figura 5.10: Implementación de carga de información. Fuentes: Los Autores

5.2.4.2 Segunda iteración

Esta iteración tiene como objetivo genera un incremento funcional que permita seleccionar los algoritmos de minería de datos y visualización de los resultados obtenidos del análisis de Logs. También en esta interacción se implementará la visualización de las métricas de evaluación de cada uno los algoritmos.

En la Tabla 5.9, se presenta una lista de tareas relacionadas con las historias seleccionadas para esta iteración.

En la Figura , observa el módulo de análisis de información, el cual muestra una gráfica

Tabla 5.9: Lista de tareas para la primera iteración

Id / Pareja	HU - 03 / Pareja 1	HU - 04 / Pareja 2
Historia de usuario	Como usuario deseo evaluar diferentes modelos de Machine Learning para detectar ataques de inyección SQL	Como administrador del sistema deseo visualizar las sentencias SQL detectadas como posibles ataques de inyección SQL para aumentar la seguridad de los sistemas de donde se obtuvo la información.
Tarea 1	Realizar la lectura de los modelos de Machine Learning desde el aplicativo web.	Desarrollar las funciones para obtener las cadenas de texto maliciosas de la limpieza de información
Tarea 2	Crear la vista de módulo de análisis de información, utilizando HTML y CSS	Crear las vistas para el módulo de métricas y tabla de sentencias
Tarea 3	Desarrollar funcionalidad para la selección de algoritmos	Crear gráficas para mostrar métricas de los algoritmos

comparativa de la cantidad de posibles ataques de inyección SQL. Además, se muestra la secuencia de texto anómala encontrada.

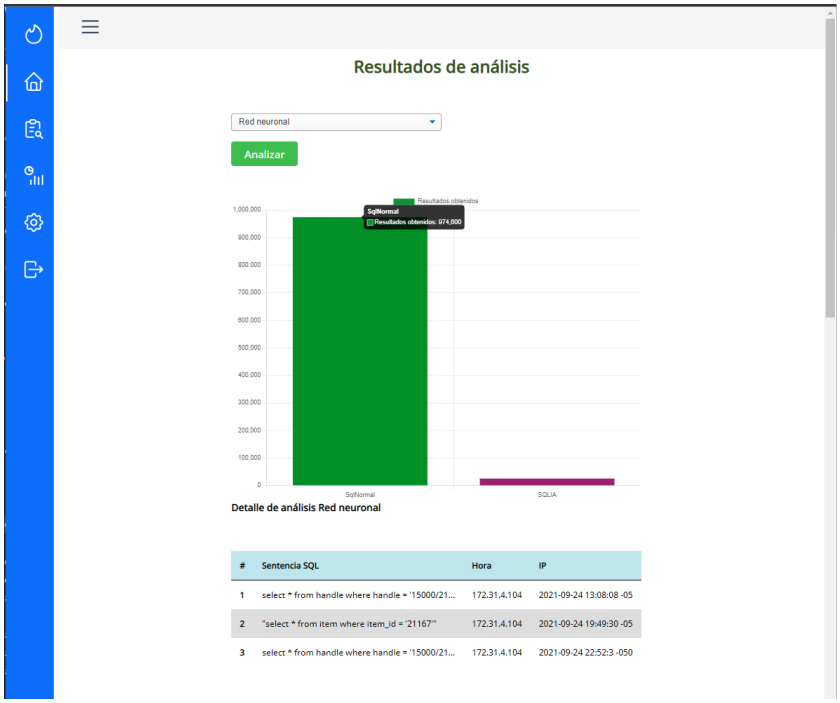


Figura 5.11: Módulo de análisis de información. Fuente: Los Autores

5.3 PRUEBAS

Una vez finalizada la fase de codificación del aplicativo web que permite usar los modelos de Machine Lerarning obtenidos con base en la metodología CRISP-DM, se inicia con la

fase de pruebas utilizando datos reales proporcionados por la CSIRT-EPN. Para lo cual se realizó lo siguiente:

- ❒ Control de acceso: en este paso se verifica que solo las cuentas registradas por el administrado puedan ingresar al sistema.

- ❒ Carga de información: para la comprobación de que un log ha sido cargado correctamente, se verificó que el sistema acepte las extensiones .rar y .zip, debido a que este es el formato del log que facilito la organización (CSIRT).

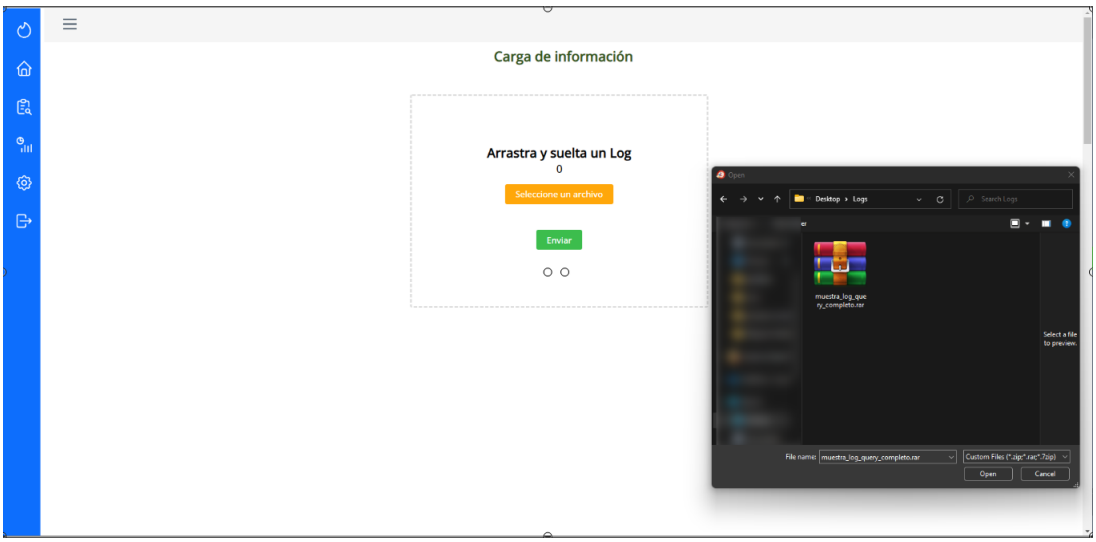


Figura 5.12: Verificación para carga de log. Fuente: El Autor.

- ❒ Descompresión de archivos: en este paso se verifica que log comprimido ha sido descomprimido y el resultado sea un archivo con extensión .CSV. En la Figura 5.13, se puede visualizar la entrada y la salida de esta funcionalidad.

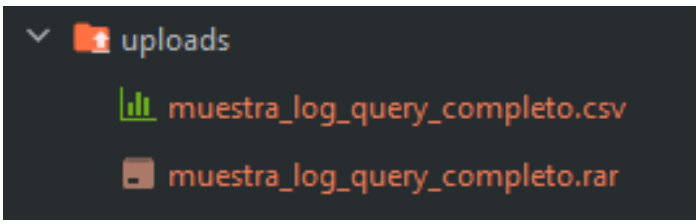


Figura 5.13: Descompresión de archivos. Fuente: El Autor.

- ❒ Limpieza y procesamiento de información: en este apartado el sistema debe realizar la limpieza (extracción de los atributos más importantes y unión de sentencias importantes para el análisis). Una vez limpia la información, los modelos deben ser capaces

```
                                                "SELECT 1"
0                                               "SELECT 1"
1                                               "SELECT 1"
2   "select * from handle where handle = '15000/21...
3   "select * from handle where handle = '15000/21...
4        "select * from item where item_id = '21167'"
```

Figura 5.14: Resultado de limpieza de log. Fuente: El autor

de identificar las cadenas de texto correspondientes a sentencias SQL legítimas y sentencias SQL anómalas.

❒ Presentación de gráficas de resultados: finalmente el sistema muestra los resultados en una gráfica de barras fácil de interpretar para el usuario, así como también se identifican las cadenas de texto anómalas encontradas, como se muestra en la Figura 5.15.

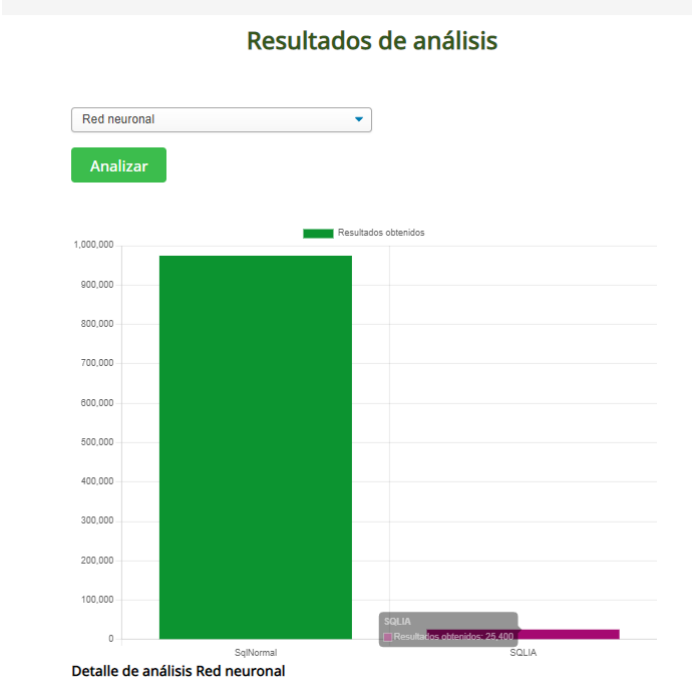


Figura 5.15: Gráfica de resultados. Fuente: El autor.

Como se puede apreciar, el sistema cuenta con los requerimientos establecidos y utilizando el modelo basado en una Red Neuronal Recurrente se pudo identificar que el 2.54 % de las cadenas de texto analizadas pueden resultar en un ataque de inyección SQL. Al finalizar el sistema se puede observar que los modelos de Machine Learning se encuentra correctamente integrados al aplicativo web.

6 ANÁLISIS DE RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En esta sección se presentarán los resultados obtenidos de la revisión sistemática de la literatura, evaluación del modelo de clasificación utilizando datos reales y de entrenamiento y desarrollo del aplicativo web para el procesamiento de información. Así como las conclusiones y recomendaciones.

6.1 ANÁLISIS DE RESULTADOS

A partir de los resultados obtenidos en la revisión sistemática de la literatura, se pudo determinar la clasificación de los diferentes tipos de técnicas de detección de ataques de inyección SQL. También se determinó que, en promedio, el puntaje de evaluación de calidad de los artículos revisados, es de 5.75 sobre 7 puntos. De acuerdo con la investigación realizada, se pudo determinar que el 38 % (19) de los artículos analizados utilizan algoritmos de Machine Learning. Esto se debe a que esta tecnología está diseñada para ejecutar más de una tarea específica, por ejemplo, aprender de un gran conjunto de datos, este aprendizaje puede ser para la detención de ataques de inyección SQL dentro de un gran volumen de información.

Dentro de las técnicas de Machine Learning se encuentran las Redes Neuronales Artificiales. Para esta investigación se desarrolló un modelo basado en una Red Neuronal Recurrente, ya que es un tipo de Red Neuronal Artificial especializada en procesar secuencias de datos, permitiendo realizar predicciones a partir de los datos que se ingresaron. De esta forma, estas Redes tienen la capacidad de recordar el texto procesado y permite la asociación de conceptos con las nuevas frases que va analizando.

El modelo desarrollado en esta investigación logra alcanzar puntajes muy altos (mayor a

90 %) en todas las métricas, a excepción de la sensibilidad, donde el puntaje es menor al 70 %. Sin embargo, esta métrica no afecta mucho al desempeño general del algoritmo, por lo que se puede determinar que el modelo es óptimo para la detección de ataques de inyección SQL.

Una vez obtenido el modelo se realizó la evaluación utilizando una muestra de un millón de registros de logs, en donde 2.54 % del total fueron identificados como posibles ataques de inyección SQL, por lo cual se puede determinar que el modelo es apto para ser desplegado en un ambiente real. No obstante, se puede mejor la velocidad de análisis de un gran volumen de información utilizando técnicas de multiprocesamiento de información.

6.2 CONCLUSIONES

- ❒ Se realizó una Revisión Sistemática de la Literatura, a partir de la cual se determinó los algoritmos y técnicas para de detección de ataques de inyección SQL, a través de este estudio, se identificaron los métodos más utilizados para detección de ataques de inyección SQL, siendo dominantes las técnicas de Machine Learning para este propósito. Para este componente se eligió las Redes Neuronales Recurrentes (RNN), para la evaluación de su desempeño a la hora de detectar y predecir ataques de inyección SQL.

- ❒ Con base en la metodología CRISP-DM se obtuvo un modelo basado en una Red Neuronal Recurrente (RNN) que permite detectar ataques de inyección SQL. El entrenamiento de este modelo arrojó resultados favorables sobre las métricas establecidas para evaluar su desempeño. Una vez obtenido, el modelo fue puesto a prueba utilizando datos reales provistos por la organización, en donde los resultados fueron también favorables, razón por la cual se puede determinar que las RNN pueden ser utilizadas en un entorno real para la predicción y detección de ataques de inyección SQL.

- ❒ Utilizando la metodología XP, se desarrolló un aplicativo web, con la finalidad de brindar al usuario un sistema intuitivo y fácil de usar. Sin embargo, el aplicativo web desarrollado se encuentra limitado, por variables de cómputo, especialmente al analizar grandes volúmenes de información. No obstante, el algoritmo seleccionado puede ser utilizado en BIG DATA utilizando técnicas de multiprocesamiento de información, y así obtener un comportamiento deseable en un entorno real, considerando factores como la velocidad de análisis, variedad de datos y volumen de datos.

6.3 RECOMENDACIONES

- ❒ Es importante comprender los datos provistos por la organización e identificar los campo que serán extraídos en la limpieza para poder realizar un modelo, entrenamiento y evaluaciones de manera efectiva.
- ❒ Se recomienda el uso de metodologías que guíen cada etapa de desarrollo de proyecto, tanto la revisión sistemática de la literatura, minería de dato y desarrollo del aplicativo. Esto ayudará a reducir la complejidad del proyecto.
- ❒ Para iniciar con la revisión sistemática de la literatura es necesario identificar la problemática que se desea abordar y contrastarla con preguntas que serán respondidas a lo largo la investigación. Además, la recopilación de información debe sé exhaustiva tanto con estudios publicados y no publicados, por lo que se recomienda establecer criterios de inclusión y exclusión para evitar incurrir en el sesgo de la selección de estudios relevantes.

6.4 TRABAJO FUTURO

Debido a factores como el tiempo para el desarrollo del este proyecto y al alcance limitado, es importante mencionar mejoras que pueden ser implementadas para trabajos futuros a partir de esta investigación.

- ❒ El ataque de inyección SQL, no es el único ataque inyección, sino también existen ataques de inyección como los ataques de Cross Site Scriting (XSS), en donde el Machine Learning es una herramienta muy potente para garantizar la seguridad de un sistema.
- ❒ Uno de los factores fundamentales en el BIG DATA es la velocidad de procesamiento. En esta investigación se utilizaron herramientas propias de Machine Learning. Sin embargo, existen tecnologías propias de BIG DATA como, Apache Spark, Apache Hadoop o el uso de multiprocesamiento, que debido a las limitaciones computacionales del proyecto, no fueron considerados en esta investigación. Por lo cual aún queda pendiente el uso estas tecnologías que pueden brindar otros resultados que podría ser interesantes de indagar.

7 REFERENCIAS BIBLIOGRÁFICAS

[1] *OWASP Top ten*, en, https://owasp.org/www-project-top-ten/, Accessed: 2022-9-8.

[2] M. Figueroa y A. Gustavo, «La metodologı´a de elaboración de proyectos como una herramienta para el desarrollo cultural,» 2005.

[3] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey y S. Linkman, «Systematic literature reviews in software engineering–a systematic literature review,» *Information and software technology*, vol. 51, n.º 1, págs. 7-15, 2009.

[4] P. Chapman, J. Clinton, R. Kerber y col., «CRISP-DM 1.0: Step-by-step data mining guide,» 2000.

[5] R. S. Pressman, *Software engineering: a practitioner's approach*. Palgrave macmillan, 2005.

[6] M. Soriano, «Seguridad en redes y seguridad de la información,» *Obtenido de http://improvet. cvut. cz/project/download/C2ES/Seguridad_de_Red_e_Informacion. pdf*, 2014.

[7] *Las amenazas de bases de datos más peligrosas y cómo prevenirlas*, en, https://geekflare.com/es/database-threats-and-prevention-tools/, Accessed: 2022-7-17, ene. de 2022.

[8] P. Fernández Gayol, «Estudio de los principales tipos de ataques por inyección de código a aplicaciones web y sistema para determinar si un código fuente es vulnerable a SQL Injection,» 2022.

[9] J. J. Camargo-Vega, J. F. Camargo-Ortega y L. Joyanes-Aguilar, «Conociendo big data,» *Revista Facultad de Ingenierı´a*, vol. 24, n.º 38, págs. 63-77, 2015.

[10] L. J. Aguilar, *Big Data, Análisis de grandes volúmenes de datos en organizaciones*. Alfaomega Grupo Editor, 2016.

[11] C. Pérez López y D. Santin González, *Minerı´a de datos. Técnicas y herramientas: técnicas y herramientas*. Editorial Paraninfo, 2007.

[12] Y. R. Suárez y A. D. Amador, «Herramientas de minerı´a de datos,» *Revista Cubana de Ciencias Informáticas*, vol. 3, n.º 3-4, págs. 73-80, 2009.

[13] *Software de data mining: realiza análisis de datos más efectivos*, es, https://www.ionos.es/digitalguide/online-marketing/analisis-web/software-de-data-mining-las-mejores-herramientas/, Accessed: 2022-8-8.

[14] B. Kitchenham, «Procedures for performing systematic reviews,» *Keele, UK, Keele University*, vol. 33, n.º 2004, págs. 1-26, 2004.

[15] R. Wirth y J. Hipp, «CRISP-DM: Towards a standard process model for data mining,» en *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, Manchester, vol. 1, 2000, págs. 29-39.

[16] R. S. Pressman y J. M. Troya, «Ingenierı´a del software,» 1988.

[17] B. Kitchenham y S. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering*, 2007.

[18] R. K. Jamra, B. Anggorojati, D. I. Sensuse, R. R. Suryono y col., «Systematic Review of Issues and Solutions for Security in E-commerce,» en *2020 International Conference on Electrical Engineering and Informatics (ICELTICs)*, IEEE, 2020, págs. 1-5.

[19] Y.-C. Chung, M.-C. Wu, Y.-C. Chen y W.-K. Chang, «A Hot Query Bank approach to improve detection performance against SQL injection attacks,» *computers & security*, vol. 31, n.º 2, págs. 233-248, 2012.

[20] W. G. Halfond, J. Viegas, A. Orso y col., «A classification of SQL-injection attacks and countermeasures,» en *Proceedings of the IEEE international symposium on secure software engineering*, IEEE, vol. 1, 2006, págs. 13-15.

[21] W. G. J. Halfond y A. Orso, «AMNESIA: Analysis and Monitoring for NEutralizing SQL-Injection Attacks,» en *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering*, ép. ASE '05, Long Beach, CA, USA: Association for Computing Machinery, 2005, págs. 174-183, ISBN: 1581139934. DOI: 10.1145/1101908.1101935. dirección: https://doi.org/10.1145/1101908.1101935.

[22] M. Hasan, Z. Balbahaith y M. Tarique, «Detection of SQL injection attacks: a machine learning approach,» en *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, IEEE, 2019, págs. 1-6.

[23] I. Lee, S. Jeong, S. Yeo y J. Moon, «A novel method for SQL injection attack detection based on removing SQL query attribute values,» *Mathematical and Computer Modelling*, vol. 55, n.º 1, págs. 58-68, 2012, Advanced Theory and Practice for Cryptography and Future Security, ISSN: 0895-7177. DOI: https://doi.org/10.1016/j.mcm.2011.01.050. dirección: https://www.sciencedirect.com/science/article/pii/S0895717711000689.

[24] C. I. Pinzón, J. F. De Paz, Á. Herrero, E. Corchado, J. Bajo y J. M. Corchado, «idMAS-SQL: Intrusion Detection Based on MAS to Detect and Block SQL injection through data mining,» *Information Sciences*, vol. 231, págs. 15-31, 2013, Data Mining for Information Security, ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2011.06.020. dirección: https://www.sciencedirect.com/science/article/pii/S0020025511003148.

[25] M.-Y. Kim y D. H. Lee, «Data-mining based SQL injection attack detection using internal query trees,» *Expert Systems with Applications*, vol. 41, n.º 11, págs. 5416-5430, 2014, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2014.02.041. dirección: https://www.sciencedirect.com/science/article/pii/S0957417414001171.

[26] H. Shahriar y M. Zulkernine, «Information-Theoretic Detection of SQL Injection Attacks,» en *2012 IEEE 14th International Symposium on High-Assurance Systems Engineering*, 2012, págs. 40-47. DOI: 10.1109/HASE.2012.31.

[27] S. Som, S. Sinha y R. Kataria, «Study on sql injection attacks: Mode detection and prevention,» *International Journal of Engineering Applied Sciences and Technology*, vol. 1, n.º 8, págs. 23-29, 2016.

[28] I. Balasundaram y E. Ramaraj, «An Efficient Technique for Detection and Prevention of SQL Injection Attack using ASCII Based String Matching,» *Procedia Engineering*, vol. 30, págs. 183-190, 2012, International Conference on Communication Technology and System Design 2011, ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2012.01.850. dirección: https://www.sciencedirect.com/science/article/pii/S1877705812008600.

[29] T. Latchoumi, M. S. Reddy y K. Balamurugan, «Applied machine learning predictive analytics to SQL injection attack detection and prevention,» *European Journal of Molecular & Clinical Medicine*, vol. 7, n.º 02, pág. 2020, 2020.

[30] P. Tang, W. Qiu, Z. Huang, H. Lian y G. Liu, «Detection of SQL injection based on artificial neural network,» *Knowledge-Based Systems*, vol. 190, pág. 105 528, 2020,

ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2020.105528. dirección: https://www.sciencedirect.com/science/article/pii/S0950705120300332.

[31] Q. Li, W. Li, J. Wang y M. Cheng, «A SQL Injection Detection Method Based on Adaptive Deep Forest,» *IEEE Access*, vol. 7, págs. 145 385-145 394, 2019. DOI: 10.1109/ACCESS.2019.2944951.

[32] N. M. Sheykhkanloo, «Employing Neural Networks for the Detection of SQL Injection Attack,» en *Proceedings of the 7th International Conference on Security of Information and Networks*, ép. SIN '14, Glasgow, Scotland, UK: Association for Computing Machinery, 2014, págs. 318-323, ISBN: 9781450330336. DOI: 10.1145/2659651.2659675. dirección: https://doi.org/10.1145/2659651.2659675.

[33] D. Kar, S. Panigrahi y S. Sundararajan, «SQLiDDS: SQL Injection Detection Using Query Transformation and Document Similarity,» en *Distributed Computing and Internet Technology*, R. Natarajan, G. Barua y M. R. Patra, eds., Cham: Springer International Publishing, 2015, págs. 377-390, ISBN: 978-3-319-14977-6.

[34] A. Ghafarian, «A hybrid method for detection and prevention of SQL injection attacks,» en *2017 Computing Conference*, 2017, págs. 833-838. DOI: 10.1109/SAI.2017.8252192.

[35] X. Xie, C. Ren, Y. Fu, J. Xu y J. Guo, «SQL Injection Detection for Web Applications Based on Elastic-Pooling CNN,» *IEEE Access*, vol. 7, págs. 151 475-151 481, 2019. DOI: 10.1109/ACCESS.2019.2947527.

[36] Y. Wang y Z. Li, «SQL injection detection via program tracing and machine learning,» en *International Conference on Internet and Distributed Computing Systems*, Springer, 2012, págs. 264-274.

[37] H. Gu, J. Zhang, T. Liu y col., «DIAVA: A Traffic-Based Framework for Detection of SQL Injection Attacks and Vulnerability Analysis of Leaked Data,» *IEEE Transactions on Reliability*, vol. 69, n.º 1, págs. 188-202, 2020. DOI: 10.1109/TR.2019.2925415.

[38] R. A. Katole, S. S. Sherekar y V. M. Thakare, «Detection of SQL injection attacks by removing the parameter values of SQL query,» en *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, 2018, págs. 736-741. DOI: 10.1109/ICISC.2018.8398896.

[39] K. Ross, M. Moh, T.-S. Moh y J. Yao, «Multi-Source Data Analysis and Evaluation of Machine Learning Techniques for SQL Injection Detection,» en *Proceedings of the ACMSE 2018 Conference*, ép. ACMSE '18, Richmond, Kentucky: Association for Computing Machinery, 2018, ISBN: 9781450356961. DOI: 10.1145/3190645.3190670. dirección: https://doi.org/10.1145/3190645.3190670.

[40] K. N. Durai, R. Subha y A. Haldorai, «A Novel Method to Detect and Prevent SQLIA Using Ontology to Cloud Web Security,» *Wireless Personal Communications*, vol. 117, n.º 4, págs. 2995-3014, 2021.

[41] J. O. Atoum y A. J. Qaralleh, «A hybrid technique for SQL injection attacks detection and prevention,» *International Journal of Database Management Systems*, vol. 6, n.º 1, pág. 21, 2014.

[42] N. M. Sheykhkanloo, «A learning-based neural network model for the detection and classification of SQL injection attacks,» *International Journal of Cyber Warfare and Terrorism (IJCWT)*, vol. 7, n.º 2, págs. 16-41, 2017.

[43] S. Bangre, A. Jaiswal y col., «SQL Injection Detection and Prevention Using Input Filter Technique,» *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 1, n.º 2, págs. 145-150, 2012.

[44] M. Hasan, Z. Balbahaith y M. Tarique, «Detection of SQL Injection Attacks: A Machine Learning Approach,» en *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, 2019, págs. 1-6. DOI: 10.1109/ICECTA48151.2019.8959617.

[45] Z. Xiao, Z. Zhou, W. Yang y C. Deng, «An approach for SQL injection detection based on behavior and response analysis,» en *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, 2017, págs. 1437-1442. DOI: 10.1109/ICCSN.2017.8230346.

[46] D. Kar, K. Agarwal, A. K. Sahoo y S. Panigrahi, «Detection of SQL injection attacks using Hidden Markov Model,» en *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, 2016, págs. 1-6. DOI: 10.1109/ICETECH.2016.7569180.

[47] L. Yan, X. Li, R. Feng, Z. Feng y J. Hu, «Detection Method of the Second-Order SQL Injection in Web Applications,» en *Proceedings of the Third International Workshop on Structured Object-Oriented Formal Language and Method - Volume 8332*, Berlin, Heidelberg: Springer-Verlag, 2013, págs. 154-165, ISBN: 9783319049144. DOI: 10.

1007/978-3-319-04915-1_11. dirección: https://doi.org/10.1007/978-3-319-04915-1_11.

[48] Z. C. S. S. Hlaing y M. Khaing, «A Detection and Prevention Technique on SQL Injection Attacks,» en *2020 IEEE Conference on Computer Applications(ICCA)*, 2020, págs. 1-6. DOI: 10.1109/ICCA49400.2020.9022833.

[49] P. Li, L. Liu, J. Xu y col., «Application of Hidden Markov Model in SQL Injection Detection,» en *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, 2017, págs. 578-583. DOI: 10.1109/COMPSAC.2017.64.

[50] T. Oosawa y T. Matsuda, «SQL injection attack detection method using the approximation function of zeta distribution,» en *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, págs. 819-824. DOI: 10.1109/SMC.2014.6974012.

[51] K. Wang e Y. Hou, «Detection method of SQL injection attack in cloud computing environment,» en *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2016, págs. 487-493. DOI: 10.1109/IMCEC.2016.7867260.

[52] Y.-C. Chung, M.-C. Wu, Y.-C. Chen y W.-K. Chang, «A Hot Query Bank approach to improve detection performance against SQL injection attacks,» *Computers & Security*, vol. 31, n.º 2, págs. 233-248, 2012, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2011.11.007. dirección: https://www.sciencedirect.com/science/article/pii/S016740481100143X.

[53] P. Kumar, «The multi-tier architecture for developing secure website with detection and prevention of sql-injection attacks,» *International Journal of Computer Applications*, vol. 62, n.º 9, 2013.

[54] D. Chen, Q. Yan, C. Wu y J. Zhao, «SQL Injection Attack Detection and Prevention Techniques Using Deep Learning,» *Journal of Physics: Conference Series*, vol. 1757, n.º 1, pág. 012 055, ene. de 2021. DOI: 10.1088/1742-6596/1757/1/012055. dirección: https://doi.org/10.1088/1742-6596/1757/1/012055.

[55] G. Singh, D. Kant, U. Gangwar y A. P. Singh, «Sql injection detection and correction using machine learning techniques,» en *Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) Volume 1*, Springer, 2015, págs. 435-442.

[56] C.-c. Shi, T. Zhang, Y. Yu y W. Lin, «A new approach for SQL-injection detection,» en *Instrumentation, Measurement, Circuits and Systems*, Springer, 2012, págs. 245-254.

[57] R. M. Nadeem, R. M. Saleem, R. Bashir y S. Habib, «Detection and prevention of SQL injection attack by dynamic analyzer and testing model,» *International Journal of Advanced Computer Science and Applications*, vol. 8, n.º 8, págs. 209-214, 2017.

[58] L. Xiao, S. Matsumoto, T. Ishikawa y K. Sakurai, «SQL Injection Attack Detection Method Using Expectation Criterion,» en *2016 Fourth International Symposium on Computing and Networking (CANDAR)*, 2016, págs. 649-654. DOI: 10.1109/CANDAR.2016.0116.

[59] R. M. Nadeem, R. M. Saleem, R. Bashir y S. Habib, «Detection and Prevention of SQL Injection Attack by Dynamic Analyzer and Testing Model,» *International Journal of Advanced Computer Science and Applications*, vol. 8, n.º 8, 2017. DOI: 10.14569/IJACSA.2017.080827. dirección: http://dx.doi.org/10.14569/IJACSA.2017.080827.

[60] M. S. Aliero e I. Ghani, «A component based SQL injection vulnerability detection tool,» en *2015 9th Malaysian Software Engineering Conference (MySEC)*, 2015, págs. 224-229. DOI: 10.1109/MySEC.2015.7475225.

[61] H. Zhang, B. Zhao, H. Yuan, J. Zhao, X. Yan y F. Li, «SQL Injection Detection Based on Deep Belief Network,» en *Proceedings of the 3rd International Conference on Computer Science and Application Engineering*, ép. CSAE 2019, Sanya, China: Association for Computing Machinery, 2019, ISBN: 9781450362948. DOI: 10.1145/3331453.3361280. dirección: https://doi.org/10.1145/3331453.3361280.

[62] G. Bafghi, «A Simple and Fast Technique for Detection and Prevention of SQL Injection Attacks (SQLIAs),» *International Journal of Security and Its Applications*, vol. 7, n.º 5, págs. 53-66, 2013.

[63] Sangeeta, S. Nagasundari y P. B. Honnavali, «SQL Injection Attack Detection using ResNet,» en *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2019, págs. 1-7. DOI: 10.1109/ICCCNT45670.2019.8944874.

[64] T.-Y. Wu, J.-S. Pan, C.-M. Chen y C.-W. Lin, «Towards SQL injection attacks detection mechanism using parse tree,» en *Genetic and Evolutionary Computing*, Springer, 2015, págs. 371-380.

[65] L. Saoudi, K. Adi e Y. Boudraa, «A rejection-based approach for detecting SQL injection vulnerabilities in web applications,» en *International Symposium on Foundations and Practice of Security*, Springer, 2019, págs. 379-386.

[66] R. Kozik, M. Choraś y W. Hołubowicz, «Hardening Web Applications against SQL Injection Attacks Using Anomaly Detection Approach,» en *Image Processing & Communications Challenges 6*, Springer, 2015, págs. 285-292.

[67] N. M. Sheykhkanloo, «A Pattern Recognition Neural Network Model for Detection and Classification of SQL Injection Attacks,» *International Journal of Computer and Information Engineering*, vol. 9, n.º 6, págs. 1436-1446, 2015, ISSN: eISSN: 1307-6892. dirección: https://publications.waset.org/vol/102.

[68] O. Hubskyi, T. Babenko, L. Myrutenko y O. Oksiiuk, «Detection of sql injection attack using neural networks,» en *International scientific-practical conference*, Springer, 2020, págs. 277-286.

[69] D. E. Nofal y A. A. Amer, «SQL Injection Attacks Detection and Prevention Based on Neuro-Fuzzy Technique,» en *International Conference on Advanced Intelligent Systems and Informatics*, Springer, 2019, págs. 722-738.

[70] N. Gandhi, J. Patel, R. Sisodiya, N. Doshi y S. Mishra, «A CNN-BiLSTM based Approach for Detection of SQL Injection Attacks,» en *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, 2021, págs. 378-383. DOI: 10.1109/ICCIKE51210.2021.9410675.

[71] R. A. Dalimunthe y S. Sahren, «Intrusion detection system and modsecurity for handling sql injection attacks,» en *International Conference on Social, Sciences and Information Technology*, vol. 1, 2020, págs. 187-194.

[72] A. O. Agbakwuru y D. O. Njoku, «SQL Injection Attack on Web Base Application: Vulnerability Assessments and Detection Technique,» *International Research Journal of Engineering and Technology*, vol. 8, n.º 3, págs. 243-252, 2021.

8 ANEXOS

En esta sección se presentan las tablas obtenidas de la revisión sistemática de la literatura detallada en el Capítulo 4

A ARTÍCULOS SELECCIONADOS PARA LA REVISIÓN

Tabla 8.1: Artículos seleccionados para la revisión

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
1	[23]	166	2012	Técnica basada en Removing Sql query attribute values	Machine learning	—	— SQL 5.0	—
2	[24]	75	2013	Técnica basada en idMas-SQL	Machine learning	CNN	Base de datos SQL 5.0	Todos
3	[25]	66	2014	Técnica basada en Data-mining based Sql injection attacj deteccion in internal query tress	Machine learning	CNN	PostgreSql v 9.2.3	Todos
4	[26]	52	2012	Técnica basada en Detection of SQL injection attacks	Machine learning	CNN	PostgreSql v 9.2.3	Todos
5	[27]	43	2016	Técnica basada en análisis estático	Prácticas de codificación segura	N/A	N/A	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
6	[28]	42	2016	Técnica basada en basada ASCII based string matching	Técnicas híbridas	String Matching	Generador de claves basado en texto, graficos SQL utilizando FMS	Todos
7	[29]	38	2020	Técnica basada en Machine learning predictive analytics to SQL injection attac	Machine Learning	SVM	N/A	Todos
8	[30]	28	2020	Técnica basada en Artificial neural network	Machine Learning	CNN	Generador de URL	Todos
9	[31]	27	2019	Técnica basada en Adaptive Deep Forest	Machine Learning	AdaBoost	Exploit-Db y wooyun-Db	Todos
10	[32]	27	2019	Técnica basada en Neural networks	Machine Learning	CNN	Generador de URL	Todos
11	[33]	26	2015	Técnica basada en SQLiDDs	Técnicas taint-based	K-meas	N/A	Todos
12	[34]	24	2017	Técnica basada en análisis estático y dinámico	Técnicas Hibridas	N/A	N/A	Todos
13	[35]	24	2019	Técnica basada en Elastic pooling - CNN	Machine learning	CNN	Registros de web reales en entorno de producción	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
14	[36]	23	2012	Técnica basada en Progam tracing and machine learning	Técnicas híbridas	N/A	N/A	Todos
15	[37]	23	2019	Técnica basada en Diava	Modificación de sentencias	N/A	Almacenamiento en la nube, análisis de trafico de red	Todos
16	[38]	22	2019	Técnica basada en Removing the parameter values of SQL query	Modificación de sentencias	N/A	Aplicaciones web vulnerables	Todos
17	[39]	22	2019	Técnica basada en Machine learning for SQL injection detection	Modificación de sentencias	N/A	Aplicaciones web vulnerables	Todos
18	[40]	17	2020	Técnica basada en Ontology to cloud web security	Practicas de codificación segura	N/A	Información guardada en la nube	Todos
19	[41]	16	2014	Técnica basada en análisis estático y dinámico	Técnicas híbridas	N/A	N/A	Todos
20	[42]	15	2017	Técnica basada en redes neuronales	Machine learning	CNN	Generador y clasificador de URLs	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
21	[43]	14	2012	Técnicas basadas en filtrado de atributos	Prácticas de códificación	N/A	N/A	Todos
22	[44]	14	2019	Técnica basada en clasificadores de machine learning	Machine learning	23 clasificadores	Ejemplos de sentencias SQL de W3School (benignos) y sentencias del OWASP SecLists Project	Todos
23	[45]	13	2017	Técnica basada en analisis de comportamiento	Otras técnicas	N/A	N/A	Solo los 6 tipos de SQLIA más básicos
24	[46]	13	2016	Técnica basada en el modelo oculto de Markov	Técnicas basadas en modelos probabilísticos	HMM	Datos reales de una configuración de prueba	Todos
25	[47]	13	2014	Tecnica basada en análisis estático y dinamico	Técnicas híbridas	N/A	N/A	Todos
26	[48]	12	2020	Técnica basada en creacion de lexicos y tokenizacion de cadenas	Árbol de análisis gramatical	N/A	N/A	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
27	[49]	12	2017	Técnica basada en el modelo oculto de Markov	Técnicas basadas en modelos probabilísticos	HMM	Datos reales de una configuración de prueba	Todos
28	[50]	12	2014	Técnica basada en la funcion de distribución Zeta	Técnicas basadas en modelos probabilísticos	N/A	Datos de ejemplo	Todoss
29	[51]	12	2016	Técnica basada en creación de reglas	Técnicas taint-based	N/A	N/A	Todos
30	[52]	12	2012	Técnica asado en creacioón de banco de consultas	Árbol de análisis gramatical	N/A	N/A	Todos
31	[53]	11	2013	Tecnica basada en análisis estático y dinamico	Técnicas híbridas	N/A	N/A	Todos
32	[54]	11	2021	Técnica basada en deep learning	Machine learning	CNN y MLP	Datos de ejemplo obtenidos de internet	Todos
33	[55]	11	2015	Técnica basada en machine learning	Machine learning	K-means	No especifica	Todos
34	[56]	10	2015	Técnica basada en creación de librerias de conocimiento	Árbol de análisis gramatical	N/A	N/A	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
35	[57]	9	2017	Técnica basada en Dynamic Analyzer and Testing Model	Taint-based Technique	N/A	datos reales de una configuración de prueba	Todos
36	[58]	9	2016	Técnica basada en Expectation Criterion	Probabilístico	N/A	datos de ejemplo	Todos
37	[59]	9	2013	Técnica basada en la detección del lado del cliente utilizando cuatro métricas de entropía condicional	Practicas de Codificación Segura	N/A	datos reales de una configuración de prueba	Todos
38	[60]	8	2015	Técnica basada en herramientas de detección de vulnerabilidades basada en Rastreo de la web, análisis de los ataques y elaboración de informes, análisis de los ataques y elaboración de informes	Análisis Estático	N/A	datos reales de una configuración de prueba	Todos
39	[61]	7	2019	Técnica basada en Deep Belief Network	Machine Learning	Deep Belief Network (DBN)	datos de ejemplo	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
40	[62]	6	2013	Técnica basada en modelos de consulta válidos obtenidos de una aplicación web	Análisis Estático y Dinámico	N/A	datos reales de una configuración de prueba	Todos
41	[63]	6	2019	Técnica basada en ResNet	Machine Learning	ResNet	uso de una herramienta (no especifica) y datos de internet	Todos
42	[64]	6	2015	Técnica basada en Dynamic SQLIA Detection (DSD)	Parse Tree	Dynamic SQLIA Detection (DSD)	datos reales de una configuración de prueba	Todos
43	[65]	5	2019	Técnica basada en rechazo	Análisis Estático	N/A	datos reales de una configuración de prueba	Todos
44	[66]	4	2015	Técnica basada en anomalias de rechazo	Analisis Estático	Linear Discriminant Analysis(LDA)	datos generados por un servicio HTTP	todos
45	[67]	4	2015	Técnica basada en una Red Neuronal	Machine Learning	Red Neuronal	datos de ejemplo	Todos
46	[68]	4	2020	Técnica basada en Artificial Neural Networks	Machine Learning	Artificial Neural Networks	datos obtenidos de sitios de internet	Todos

ID	Ref.	Citas	Año	Técnica	Clasificación	Algoritmo(s)	Dataset	Tipos de SQLIA
47	[69]	3	2019	Técnica basada en Neuro-Fuzzy	Machine Learning	Adaptive Neuro-Fuzzy Inference System (ANFIS) / Fuzzy C-Means (FCM) / ScaledConjugate Gradient (SCG)"	datos reales de una configuración de prueba	Todos
48	[70]	1	2021	Técnica basada en CNN-BiLSTM	Machine Learning	CNN-BiLSTM	datos obtenidos de sitios de internet	Todos
49	[71]	1	2020	Técnica basada en un Sistema de Detección de Intrusos y Cortafuegos(ModSecurity)	Sistema de detección de intrusos	N/A	Datos de ejemplo	Todos
50	[72]	1	2021	Técnica basada en fuzzy rule-based classification system (FRBCS)	Machine Learning	Algoritmo Genético Simple	datos reales de una configuración de prueba	Todos

B CRITERIOS DE CALIFICACIÓN PARA LAS PREGUNTAS DE EVALUACIÓN DE CALIDAD

Tabla 8.2: Criterios de calificación para las preguntas de evaluación de calidad

Pregunta	Puntajes posibles		
QA1	0: Si no se indica en el abstract o en la introducción de manera la técnica a desarrollar en el artículo. en el artículo.	0.5: Si se indica en el abstract o en la introducción de manera implícita la técnica a desarrollar en el artículo.	1: Si se indica en el abstract o en la introducción de manera explícita la técnica a desarrollar
QA2	0: Si no se describe ningún tema que dé contexto de la investigación en el artículo.	0.5: si se describe un solo tema que dé contexto de la investigación en el artículo.	1: Si se describen al menos dos temas diferentes que den contexto de la investigación en el artículo.
QA3	0: Si no existe ningún indicio o mención a trabajos relacionados dentro del artículo.	0.5: Si existe un indicio o breve referencia a trabajos relacionados en el artículo pero no se los describe de manera detallada.	1: Si existe una sección de trabajos relacionados en el artículo o se describen de manera detallada algunos trabajos relacionados.
QA4	0: Si no existe una descripción de la arquitectura o metodología propuesta en el artículo.	0.5: Si existe una descripción inconsistente, incompleta o ambigua de la arquitectura o metodología propuesta en el artículo.	1: Si existe una descripción clara, completa y detallada de la arquitectura o metodología propuesta en el artículo
QA5	0: Si no se muestran ni la evaluación ni los resultados de la metodología o técnica propuesta en el artículo.	0.5: Si se solo se evalúa la metodología o técnica propuesta en el artículo sin mostrar los resultados o solo se muestran los resultados de la metodología o técnica propuesta sin mostrar la evaluación.	1: Si se evalúa de manera detallada la metodología o técnica propuesta en el artículo y se muestran los resultados de dicha evaluación.

Pregunta	Puntajes posibles		
QA6	0: Si la conclusión de la investigación difiere completamente de los objetivos propuestos en el artículo o si no existen conclusiones en el artículo.	0.5: Si la conclusión de la investigación difiere un poco de los objetivos propuestos en el artículo.	1: Si la conclusión de la investigación muestra concordancia con los objetivos propuestos en el artículo.
QA7	0: Si no se indican trabajos futuros en el artículo.	0.5: Si los trabajos futuros no se detallan claramente o no se relacionan directamente con la investigación principal del artículo.	1: si los trabajos futuros se detallan claramente y se relacionan directamente con la investigación principal del artículo.

C PUNTAJE DE LA EVALUACIÓN DE CALIDAD DE LOS ARTÍCULOS ANALIZADOS

Tabla 8.3: Puntaje de la evaluación de calidad de los artículos analizados.

#	QA1	QA2	QA3	QA4	QA5	QA6	QA7	Total
[23]	1	1	1	1	1	1	1	7
[24]	1	1	1	1	1	1	0	6
[25]	1	1	1	1	1	0.5	0	5.5
[26]	1	1	1	1	1	1	0	6
[27]	1	1	1	1	1	1	1	7
[28]	1	1	1	1	1	1	0	6
[29]	1	0.5	0.5	1	1	0.5	1	5.5
[30]	1	1	1	1	1	1	1	7
[31]	1	1	1	1	1	1	0	6
[32]	1	1	1	0	1	1	0.5	5.5
[33]	1	1	1	1	1	1	1	7
[34]	1	1	1	1	1	1	1	7
[35]	1	1	1	1	1	1	0.5	6.5
[36]	1	1	1	1	1	1	0.5	6.5
[37]	1	1	1	1	1	1	0	6
[38]	1	1	0	1	1	1	0	5
[39]	1	1	1	1	1	1	1	7
[40]	1	1	1	1	1	1	1	7
[41]	1	1	0.5	1	1	1	1	6.5
[42]	1	1	1	1	1	1	0	6
[43]	1	1	0.5	1	1	1	1	6.5
[44]	1	1	1	1	1	1	1	7
[45]	0.5	1	1	1	1	1	1	6.5
[46]	1	0.5	1	1	1	1	1	6.5
[47]	1	1	1	1	1	1	1	7
[48]	1	1	0	1	1	0.5	0	4.5
[49]	1	1	1	1	1	1	1	7
[50]	1	1	0	1	1	1	1	6
[51]	1	1	0	1	1	1	0	5
[52]	1	1	1	1	1	1	1	7
[53]	1	1	1	1	1	0.5	1	6.5

#	QA1	QA2	QA3	QA4	QA5	QA6	QA7	Total
[54]	1	1	0.5	1	1	1	1	6.5
[55]	1	1	0.5	1	1	0.5	1	6
[56]	1	1	0	1	1	0.5	1	5.5
[57]	1	1	0.5	1	1	1	0.5	6
[58]	1	0,5	1	0,5	1	1	1	6
[59]	1	0	0	0.5	0	0.5	0.5	2.5
[60]	1	0.5	1	1	1	1	1	6.5
[61]	1	1	1	0.5	1	1	0	5.5
[62]	1	1	1	1	1	1	0.5	6.5
[63]	1	1	1	1	1	1	0	6
[64]	1	0.5	0.5	1	1	1	0.5	5.5
[65]	0.5	0.5	0	0.5	1	1	1	4.5
[67]	1	1	1	1	1	1	1	7
[68]	1	0.5	0.5	1	1	1	0	5
[69]	1	1	1	1	1	1	1	7
[70]	1	1	1	1	1	1	1	7
[71]	1	0.5	0	1	1	0	0	3.5
[72]	1	1	1	1	1	1	0.5	6.5