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 UN PERCEPTRÓN MULTICAPA 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

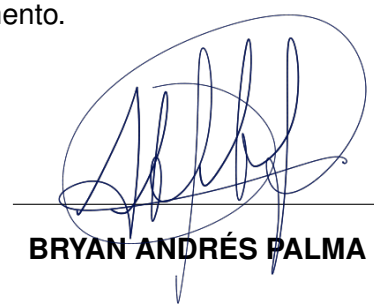
BRYAN ANDRÉS PALMA PONCE
bryan.palma02@epn.edu.ec

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

DMQ, octubre 2022

CERTIFICACIONES

Yo, Bryan Andrés Palma Ponce , 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.



BRYAN ANDRÉS PALMA PONCE

Certifico que el presente trabajo de integración curricular fue desarrollado por Bryan Andrés Palma Ponce, 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.

BRYAN ANDRÉS PALMA PONCE

DRA. GABRIELA LORENA SUNTAXI OÑA

Colaboradores del proyecto integrador:

ANDRÉS MAURICIO LLUMIQUINGA GUAMBA

EDISON JAVIER QUIMBIAMBA GUASGUA

STEVEN JAVIER RIVERA TENELANDA

DEDICATORIA

A Cheli, mi madre, y a mi tía Narciza, quienes estuvieron conmigo en todo momento.

AGRADECIMIENTOS

En primer lugar, quiero agradecer a mi madre, quien se arriesgó, confió en mí, y me permitió estudiar lejos de casa para enfrentarme a esta aventura.

También quiero agradecer a mi tía Narciza, a quien considero mi segunda madre, y que me cuido de esa manera.

A mis grandes amigos de la universidad: David, Edison, Steven, Francisco, Juan Pablo y Pierre; a quienes no tengo palabras para describir lo importante que han sido durante esta etapa de mi vida.

A mis amigas de Manabí: Nicole, Angie, y María José, quienes estuvieron para mí en los momentos más difíciles.

A mi directora de tesis, Dra. Gabriela Suntaxi, gracias por tenerme taaaaaaanta paciencia durante el desarrollo de este proyecto, y también por inspirarme a ponerme nuevos desafíos.

A mi prima Tachy, quien es casi como una hermana, y quien siempre ha creído en mí.

A mi tía Cecilia, quien también ha sido muy importante y siempre me ha brindado su apoyo.

A mi prima Yoyi, quien me acolitó durante mi primer año viviendo en Quito.

A mis amigas Emily y Rebeca, quienes me inspiran a seguir estudiando y llegar más lejos.

A mis roomies Belén y Karellis, quienes aparecieron de manera repentina y se convirtieron en unas de mis mejores amigas.

A esos amigos que fui conociendo desde los primeros días en la universidad y con los que compartí muchos momentos agradables: Joss, Johan, Toa, Sergio, Boris, Caro P., Caro D., Luisa, Sebas, Yajaira, y muchos otros que no me alcanza el espacio para mencionarlos.

A los profesores de Ludolab, quienes además de ser grandes docentes son excelentes personas que incentivarme, y fueron un gran apoyo al final de mi carrera.

Y a todos mis amigos, familiares, profesores y demás personas que de una u otra persona me ayudaron y me apoyaron para llegar a donde me encuentro ahora.

CONTENIDO

RESUMEN

Actualmente, los Ataques de Inyección SQL (SQL Injection Attack o SQLIA por sus siglas en inglés) se encuentran en el tercer lugar de la lista OWASP (Open Web Application Security Project) TOP 10: 2021 [1], principalmente por la fácil explotabilidad que tienen. Sin embargo, la ejecución de estos ataques resulta perjudicial para los sistemas informáticos actuales, en los cuales, debido a su complejidad, no se logran tomar las medidas necesarias para mitigar la amenaza que representan estos ataques. Es este componente se investiga de manera sistemática la literatura relacionada con el uso de algoritmos y técnicas de minería de datos para detectar y prevenir SQLIA. A partir de esta investigación, se realiza la evaluación de un algoritmo de minería de datos, para determinar el grado de efectividad al momento de detectar cadenas de texto que podrían resultar en un posible SQLIA. Posteriormente, se realiza el desarrollo de un prototipo de sistema que permita monitorear logs de registros e ingresos de datos utilizando el algoritmo evaluado para detectar y prevenir los SQLIA. El objetivo de este componente es aumentar la confianza y seguridad de los sistemas de la organización donde sea implementado el sistema desarrollado.

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

ABSTRACT

SQL Injection Attacks (SQLIAs) are currently in third place on the OWASP (Open Web Application Security Project) TOP 10 list: 2021 [1], mainly due to their easy exploitability. However, the execution of these attacks is detrimental to current computer systems. Due to their complexity, it is not possible to take the necessary measures to mitigate these attacks'threats is impossible.

This component systematically investigates the literature related to algorithms and data mining techniques to detect and prevent SQLIAs. Based on this research, a data mining algorithm is evaluated to determine the degree of effectiveness when detecting text strings that could result in a possible SQLIA. Subsequently, a prototype system is developed that allows monitoring log records and data entries using the evaluated algorithm to detect and prevent SQLIAs. This component aims to increase the confidence and security of the organization's systems where the developed system is implemented.

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

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En la actualidad, los sistemas utilizados por las organizaciones resultan cada vez más complejos desarrollar y mantener, debido a que existen muchos más aspectos que deben ser considerados como la seguridad. Debido a esta problemática, existen muchas organizaciones que determinan y clasifican las vulnerabilidades y amenazas más importantes dentro de la ciberseguridad. Un ejemplo de ello es OWASP, que cada cierto tiempo actualiza una lista llamada OWASP Top 10, que clasifica las amenazas más importantes en el internet.

En esta lista históricamente se ha determinado que los ataques de inyección son una amenaza muy importante dentro de la seguridad de la información dentro las organizaciones, debido al impacto que pueden tener, y la facilidad de ejecución. Debido a la complejidad de los sistemas actuales, y a la dificultad de mitigar manualmente todos los puntos donde una aplicación es susceptible a ser objeto de un ataque, es necesario optar por alternativas que permitan controlar esta vulnerabilidad de una manera más automatizada. Por esta razón, es importante realizar una revisión de literatura para determinar cuál es el estado del arte en cuanto a técnicas de detección y predicción de este tipo de ataques.

Otro aspecto relevante dentro de las organizaciones es la gran cantidad de datos que producen, y que muchas veces no son valorados lo suficiente, dado que estos datos pueden brindar información muy valiosa como son patrones de uso de los usuarios. Aquí es donde entra en acción la minería de datos, un proceso que permite generar conocimiento a partir de datos. En este contexto, podría ayudar a detectar posibles ataques dentro los registros de las bases de datos, por medio de diversas técnicas como es la Inteligencia Artificial o el Machine Learning).

Por medio de este proyecto se propone desarrollar una solución basada en minería de datos aplicando un algoritmo de Machine Learning que permita, por medio de una aplicación web, analizar los datos generados por la organización en un caso de estudio, para determinar patrones y posibles ataques en sus diversos sistemas, de manera que se puedan tomar

decisiones oportunas al respecto.

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 SQLIA, 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 la evaluación de un algoritmo de minería de datos para la detección y predicción de ataques de inyección SQL en Big Data, aplicado a un caso de estudio real.

1.2 OBJETIVOS ESPECÍFICOS

- ❒ Realizar un estudio de la literatura para identificar los principales ataques de inyección SQL y analizar los trabajos relacionados en el área de predicción y prevención de ataques.
- ❒ Realizar el análisis y la evaluación de un algoritmo de minería de datos para detección de ataques de inyección SQL
- ❒ Desarrollar un prototipo de software para la detección y predicción de ataques de inyección SQL en Big Data utilizando un algoritmo de minería de datos
- ❒ Evaluar el prototipo desarrollado en un caso de estudio utilizando datos reales

1.3 ALCANCE

Se utilizará la una adaptación de la metodología descrita en [2] para el desarrollo del proyecto. En esta metodología se contemplan las fases de: Planificación, Diseño, Implementación, Evaluación y Comunicación.

- ❒ En la fase de planificación se definirán aspectos generales como justificación, objetivos, alcance, recursos, cronogramas, etc.

- ❒ En la fase de diseño 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 los criterios que serán considerados para su evaluación.

- ❒ En la fase de implementación se tienen consideradas 2 actividades principales:

  - ✧ El proceso de minería de datos utilizando la metodología CRISP-DM [4]. De este proceso se se obtendrá un modelo que se será evaluado bajo los criterios de éxito seleccionados, obteniendo así el grado de efectividad del algoritmo seleccionado al analizar cadenas que representen potenciales SQLIA
  - ✧ El proceso de desarrollo de un prototipo de software para la detección y predicción de ataques de inyección SQL en Big Data utilizando un algoritmo de minería de datos. Para este proceso se utilizará la metodología de desarrollo XP [5]

- ❒ En la fase de evaluación se analizarán los resultados obtenidos del proceso de minería de datos, y de la implementación del prototipo utilizando datos reales.

- ❒ Finalmente, en la fase de comunicación se documentarán los análisis realizados y los resultados obtenidos dentro del desarrollo del proyecto.

2 MARCO TEÓRICO

En esta sección se detallan algunos conceptos fundamentales para el desarrollo del componente. En primer lugar, se describen aspectos esenciales sobre la seguridad en los sistemas de información. Luego se detallan los ataques más comunes dentro de las bases de datos, donde destacan los SQLIA. La parte más importante de este componente reside en el proceso de minería de datos en datos masivos, por lo que finalmente, estos temas son detallados dentro de esta sección.

2.1 SEGURIDAD EN SISTEMAS DE INFORMACIÓN

Se ha demostrado que el uso de datos en la toma de decisiones dentro del ámbito empresarial, aumenta la eficiencia y la rentabilidad en las actividades comerciales, pero esto también puede perjudicar a la empresa, dado que aumentan los riesgos implicados en comprometer la integridad de los sistemas y la infraestructura de las Tecnologías de la Información y Comunicación (TIC) [6], así como también la pérdida o el robo de información sensible. Cuando se habla de seguridad de la información es común que se piense en la tríada CIA (Confidencialidad, Integridad y Disponibilidad) debido a que es la primera línea de acción para proteger los activos una organización [6].

La Tríada CIA

De acuerdo al análisis realizado en [7], la seguridad informática tiene un enfoque funcional, en el cual se hace hincapié en los aspectos técnicos de seguridad y en desarrollo de controles que se centran en la confidencialidad, integridad y disponibilidad de los sistemas de información. Los componentes de la tríada CIA se describen a continuación:

- **Confidencialidad:** La confidencialidad puede relacionarse con la privacidad de la información. La confidencialidad es un aspecto de la seguridad en el cual la información

sensible o confidencial es restringida, de manera que esta solo sea accedida por personas autorizadas. Estos controles se los realiza comúnmente mediante un estricto control de acceso basado en roles que determina que persona puede ver o no cierta información.

- ❒ **Integridad:** La integridad es la encargada de controlar la consistencia, confiabilidad y precisión de los datos. En términos más simples, se enfoca en que los datos no sean modificados por personas no autorizadas. Comúnmente para mitigar esto se utiliza un control de versiones donde se muestra que usuario modificó cierta información.
- ❒ **Disponibilidad:** La disponibilidad garantiza que la información esté siempre disponible para los usuarios autorizados mediante el control adecuado de la infraestructura de todo el hardware y software que está involucrado en proveer información al usuario.

2.2 BIG DATA

El término “Big Data”, tiene su origen debido a que cada día se están creando una gran cantidad de datos, por ejemplo, los datos que se producen actualmente se estiman que tienen un orden de zettabytes y tienen un ritmo de crecimiento constante del 40 por ciento por año[8]. Tomando la premisa anterior, hoy en día los analistas de datos están en la obligación de manejar grandes cantidades de datos, por lo tanto, están en búsqueda constante de nuevos algoritmos y herramientas para manejar estos datos. Según Doung Laney [9], la gestión de Big Data debe considerar las 3V:

- ❒ **Volumen:** cada vez existen más datos, su tamaño sigue aumentando, pero no el porcentaje de datos que las herramientas pueden procesar.
- ❒ **Variedad:** existen diferentes tipos de datos, por ejemplo texto plano, datos de sensores, gráficos, audio, video, etc.
- ❒ **Velocidad:** los datos están llegando continuamente como flujos de datos y es importante obtener información relevante de ellos en tiempo real.

Debido a la cantidad de datos que actualmente dentro de los sistemas de información, un aspecto clave dentro de la gestión de dichos datos es la seguridad que rodea a los motores de base de datos utilizados. Por esta razón, es necesario identificar los principales ataques y amenazas que existen con respecto a las bases de datos.

2.3 ATAQUES Y AMENAZAS A BASES DE DATOS

Según Malik en [10], las bases de datos comprenden unos de los principales objetivos de los atacantes debido a que mediante las bases de datos un atacante puede obtener información valiosa sobre una organización, por lo tanto, se enfrentan a varios tipos de amenazas. A continuación se describen estas amenazas:

- **Exceso de privilegios:** Consiste en el abuso de privilegios donde el atacante obtiene acceso a datos superior al permitido para fines no autorizados. Según Malik en [10], alrededor del 80 por ciento de los ataques a la información o datos de una empresa son realizados por empleados o ex-empleados.

- **Malware:** Consiste en el uso de software con carga maliciosa (Malware) y suplantación de identidad en correos electrónicos (phishing) para ingresar a los sistemas de información de las organizaciones para robar datos confidenciales y como los usuarios no tienen conocimiento de que su equipo está infectado, estos usuarios se vuelven un conductor para que los atacantes accedan a su información[10].

- **Seguimiento de auditoría débil:** Consiste en la falta de control en lo que respecta a las políticas débiles de auditoría de base de datos, los cuales representan varios riesgos para la organización.

- **Exposición de respaldos:** Consiste en la extracción no autorizada de unidades de almacenamiento como discos externos, donde se almacenen respaldos de bases de datos, exponiendo completamente información sensible de la organización.

- **Autenticación débil:** Cuando existen modalidades de autenticación con vulnerabilidades; estas permiten a los atacantes utilizar estrategias como fuerza bruta, ingeniería social, etc., para tomar control de la identidad de los usuarios legítimos de la organización. [10].

- **Mala configuración de bases de datos:** Muchas veces no se toman las medidas necesarias en cuanto a parches y actualizaciones de seguridad en las bases de datos. También es bastante común que existan bases de datos con configuraciones predeterminados para la realización de pruebas. [10]. Los atacantes conocen la forma de explotar estas vulnerabilidades para realizar ataques a la información de una organización. Desafortunadamente, las organizaciones tienen muchas dificultadas para

mantenerse al día con los parches disponibles y con su configuración y mantenimiento, dando como resultado que las organizaciones tarden varios meses en parchear las bases de datos, tiempo en el cual siguen siendo vulnerables.

- ❒ **Datos sensibles mal administrados:** Muchas organizaciones realizan un esfuerzo constante por mantener una descripción detallada de sus bases de datos, pero en ocasiones estas bases de datos pueden ser olvidadas y como poseen información confidencial, se vuelve una vulnerabilidad que los atacantes pueden explotar fácilmente si no se implementan los controles y permisos requeridos [10].

- ❒ **Denegación de servicios:** Consiste en negar completamente el servicio o acceso a aplicaciones o datos de la red a un usuario legítimo de la base de datos.

- ❒ **Experiencia limitada en temas de seguridad:** Los controles internos que se realizan en una organización por lo general no siguen el ritmo del crecimiento de los datos y en algunos casos no están preparadas para hacer frente a una brecha de seguridad de este tipo. Esto se debe a que el personal no tiene la experiencia necesaria para implementar controles de seguridad adecuados o hacer cumplir políticas de seguridad, dando como resultado los ataques a las bases de datos [10].

- ❒ **SQLIA:** Consiste en que un atacante ingresa código de lenguaje de consulta estructurado en el cuadro de entrada de un formulario con el objetivo de obtener acceso o realizar cambios a los datos de un sistema web.

Es importante profundizar sobre los SQLIA, al ser el eje principal de la investigación, y representar un punto importante dentro de los ataques a bases de datos.

2.4 SQLIA

Los SQLIA consisten en la inserción de una o más consulta SQL maliciosas por medio de la entrada de datos dentro de una aplicación. De ejecutarse de manera exitosa, la carga maliciosa contenida en la consulta (exploit), podría acceder a la base de datos y ejecutar acciones que comprometan tanto seguridad de la base de datos como su contenido. Entre las operaciones que podría llevar a cabo un atacante se tienen: operaciones CRUD (Create, Read, Update y Delete) en los datos, ejecución de operaciones administrativas (apagar el motor de la base de datos), o incluso ejecutar comandos directamente en el sistema

operativo [11]. En la Figura 2.1 se observa el esquema general de un ataque de inyección SQL

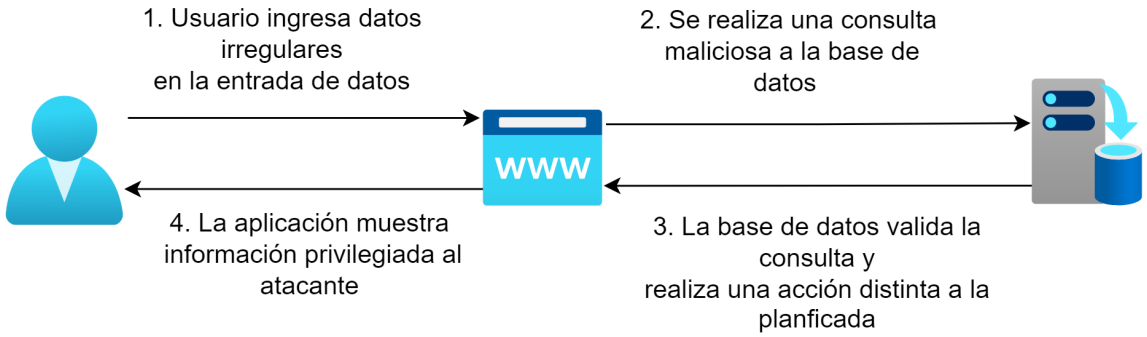


Figura 2.1: Esquema de un ataque de inyección SQL. Fuente: Autoría propia

Tipos de ataques de inyección SQL

De acuerdo con [12], los SQLIA pueden clasificarse de la siguiente manera:

a. Ataque basado en tautologías

Descripción: En este tipo de ataques se ejecutan sentencias condicionales que siempre son evaluadas como verdaderas. Para la ejecución de este ataque se hace uso de la cláusula WHERE, para inyectar una tautología, evadiendo así la restricción inicial que representaba la cláusula. De ejecutarse correctamente podrían realizarse autenticaciones o extracciones de datos no autorizados.

Ejemplo: Dada la siguiente estructura de una consulta SQL:

```
SELECT wage FROM employees WHERE username= '' AND password=''
```

Si en el campo de login se ingresa el texto ***'OR 1=1 --***, en esta cadena la comilla simple completaría la comilla correspondiente al campo login, seguido de una condición de OR, y una tautología. Luego, se insertan dos guiones que darían inicio a un comentario, y por ende el campo de contraseña se convierte en un comentario.

```
SELECT wage FROM employees WHERE username= '' OR 1=1 --' AND password=''
```

Al convertirse en una consulta en donde la condición siempre se cumplirá como verdadera, la consulta retornará todas las cuentas de usuario de la tabla *employees*.

b. Consultas ilegales/incorrectas lógicamente

Descripción: En este tipo de ataques se inyectan sentencias que pueden brindar información importante o sensible sobre la estructura de la base de datos utilizada y sus componentes. Para la ejecución de este tipo de ataques, el atacante intenta ingresar sentencias que produzcan errores lógicos, de sintaxis, de conversión de datos, o cualquier tipo de error que permitan obtener información que comprometa la confidencialidad de la organanización. De ejecutarse correctamente puede obtenerse información sobre tipos de datos, parámetros inyectables, nombres de tablas, motor de la base de datos, etc.

Ejemplo: Dada la estructura de una consulta SQL:

```
SELECT accounts FROM users WHERE login='' AND pass='' AND pin= convert (int,(
    select top 1 name from sysobjects where xtype='u'))
```

En este caso se inyectó la sentencia ***convert (int,(select top 1 name from sysobjects where xtype='u'))***.

Al ejecutar dicha sentencia, se intenta extraer el primer registro de tabla de *users* (xtype='u') de los metadatos de la tabla . En este caso se asume que se está utilizando el motor de base de datos Microsoft SQL Server, que almacena esta información en la tabla *sysobjects*. Al ejecutar la sentencia se obtiene un error por conversión de datos: "*Microsoft OLE DB Provider for SQL Server (0x80040E07) Error converting nvarchar value 'CreditCards'to a column of data type int.*" Este mensaje de error da información valiosa como el motor de la base de datos y el nombre de una tabla de la base de datos. De la misma forma se podría obtener información como tipos de datos o columnas que existen en una tabla, exponiendo información sensible que puede ser utilizada en ataques futuros.

c. Consultas de unión

Descripción: En este tipo de ataques, el atacante se aprovecha de un parámetro vulnerable para obtener un conjunto de datos distinto al que originalmente se pretendía. Con este método, se puede obtener los resultados de una tabla específica, inyectan-

do una sentencia ***UNION*** sobre la consulta original. De ejecutarse correctamente, el atacante puede obtener información de cualquier tabla.

Ejemplo:

```
SELECT accounts FROM users WHERE login='' UNION SELECT cardNo from CreditCards
    where acctNo=10032 -- AND pass='' AND pin=
```

En la consulta anterior se inyectó la sentencia "***‘UNION SELECT cardNo from CreditCards where acctNo=10032 - -***, de manera que, si no existe el login igual a ***‘’***, la base de datos retornará el número de tarjeta de la cuenta 10032.

d. Consultas respaldadas

Descripción: En este tipo de ataques, el atacante inyecta consultas adicionales dentro de la consulta original. Este tipo de ataque se diferencia de otros tipos de ataques, ya que en este no se intenta modificar la intención original de la consulta, sino que se intentan filtrar más consultas que realicen diversas acciones. De ejecutarse exitosamente puede ser muy perjudicial, ya que podría inyectarse prácticamente cualquier tipo de sentencia con operaciones como actualización o borrado de datos.

Ejemplo:

```
SELECT accounts FROM users WHERE login='doe' AND
pass=''; drop table users -- ' AND pin=123
```

En este ejemplo, se inyecta la sentencia ***’; drop table users –***, que convierte la validación del pin en un comentario. De esta manera, se ejecuta en primer lugar la primera sentencia de consulta, que no retorna resultados, y posteriormente ejecuta una sentencia de borrado de los usuarios de la base de datos. La separación de las consultas se la hace con el delimitador ‘;’.

e. Procedimientos almacenados

Descripción: Para este tipo de ataques se intentan ejecutar procedimientos almacenados en la base de datos. En la actualidad los gestores de bases de datos incluyen procedimientos almacenados por defecto que permiten interacciones con el sistema operativo. Si el atacante logra descubrir el gestor de base de datos utilizado, es posible

tomar ventaja de los procedimientos almacenados. Existen la mala concepción de que los procedimientos almacenados inmunizan a las bases de datos contra ataques de inyección. Sin embargo, estos procedimientos pueden dejar a la base de datos igual de vulnerables. Además, debido a los lenguajes de secuencia de comandos (scripting) en los que se escriben los procedimientos, estos pueden tener vulnerabilidades como desbordamiento de búfer (buffer overflow) que permitan al atacante elevar privilegios arbitrariamente.

Ejemplo: Se tiene el siguiente procedimiento almacenado que retorna el estado de autenticación de un usuario mediante un valor de verdadero o falso:

```
CREATE PROCEDURE DBO.isAuthenticated
@userName varchar2, @password varchar2, @codigo int
AS
EXEC("SELECT accounts FROM users
WHERE login='" +@userName+ "' and password='" +@password+
"' and codigo=" +@codigo);
GO
```

Si se inyecta la sentencia ***'; SHUTDOWN; –*** dentro del campo de usuario o contraseña, la consulta ejecutada dentro del procedimiento almacenado queda transformado de la siguiente manera:

```
SELECT description FROM clients WHERE
login='andres' AND pass='445566'; SHUTDOWN; -- AND pin=
```

convirtiéndose en un ataque de consultas respaldadas que apaga la base de datos.

f. Inferencia

Existen 2 tipos de ataques basados en inferencias:

❒ **Inyección a ciegas**

Descripción: Es común que los mensajes de error comprometedores sean ocultados a la vista de los posibles atacantes, dificultando, pero no imposibilitando, encontrar vulnerabilidades en el sistema. Una opción para saltar esta medida de seguridad es enviar consultas con retornos de verdadero y falso.

Ejemplo: Se tienen las siguientes consultas:

```
    SELECT description FROM clients WHERE nombre='andres'
and 1=0 -- ' AND contra='' AND codigo=0
```

```
    SELECT description FROM clients WHERE nombre='andres'
and 1=1 -- ' AND contra='' AND codigo=0
```

En caso de ser una aplicación segura, ambas consultas retornarían un error. En otro caso, en el que el sistema no se encuentre asegurado, la primera consulta retornaría un error. En este punto, el atacante no sabe si el error se debe a una correcta validación, o es por el hecho de tratarse de una consulta incorrecta. Sin embargo, el atacante introduce la segunda consulta, que no retorna ningún error. Al no retornar ningún error, el atacante descubre que el campo es vulnerable a ataques de inyección.

❒ **Temporización de ataques:**

Descripción: Este ataque utiliza la sincronización para permitir al atacante reunir información sobre la base de datos mediante la observación de retrasos en las respuestas de la base de datos. Para ejecutar este ataque se utilizan las cláusulas if-then que hace que la consulta se ejecute con una declaración de retardo dependiendo de la lógica inyectada. Dentro del ataque se realizan preguntas con la sentencia if y la palabra ***WAITFOR*** que retrasa la respuesta del servidor por un tiempo determinado.

Ejemplo: Se tienen las siguientes consultas:

```
    SELECT accounts FROM users WHERE login='legalUser' and
ASCII(SUBSTRING((select top 1 name from sysobjects),1,1))
> X WAITFOR 5 -- ' AND pass='' AND pin=0
```

En esta consulta se inyecta la sentencia ***'legalUser'and ASCII(SUBSTRING((select top 1 name from sysobjects),1,1)) >X WAITFOR 5 —*** Dentro de la sentencia se encuentra la función SUBSTRING, que extrae el primer carácter del nombre

de la tabla. Usando la estrategia de búsqueda binaria, se iteran por diversos valores de X, para determinar el carácter del nombre de la tabla. Si se cumple la condición, el gestor de la base tendrá un retraso de 5 segundos en su respuesta y así dar indicios sobre información delicada al atacante.

g. Codificaciones alternativas

Descripción: En este tipo de ataque, el atacante busca evadir las medidas de seguridad como la detección de caracteres no permitidos. Para saltarse la seguridad, los atacantes utilizan codificaciones alternativas como códigos ASCII, hexadecimal, etc. De esta manera, el sistema no puede detectar, por ejemplo, si se ingresa un operador de comentario. Un ejemplo de esto es la función char(120) para representar el caracter "x". Este tipo de ataques es especialmente difícil, ya que requiere que el atacante pruebe con diversas codificaciones y funciones para llegar a un resultado exitoso.

Ejemplo: En la consulta:

```
SELECT accounts FROM users WHERE login='' AND pass='' AND pin=
```

se inyecta la sentencia ***"legalUser';exec(char(0x73687574646f776e))–"*** obteniéndose la sentencia

```
SELECT accounts FROM users WHERE login='legalUser'; exec(char(0
    x73687574646f776e)) -- AND pass='' AND pin=
```

En este caso la función ***char(0x73687574646f776e)*** retorna el string "SHUTDOWN". Así, al realizarse la consulta, la base de datos ejecutaría el comando ***SHUTDOWN***, apagando el motor de base de datos.

2.5 MINERÍA DE DATOS

Hand define en [13] a la minería de datos como «el descubrimiento de estructuras interesantes, inesperadas o valiosas en extensos conjuntos de datos». La minería de datos no es una disciplina reciente, ya que el uso de estadísticas, ideas, herramienta métodos computacionales y tecnologías para la analítica de datos se ha venido dando a lo largo del siglo XX. Sin

embargo, en la actualidad se presentan nuevos desafíos con respecto al manejo de datos. Los conjuntos de datos son cada vez más grandes y se requieren herramientas, técnicas y procesos más complejos para obtener información relevante dentro de dichos datos. Las técnicas de minería de datos se clasifican en 2 categorías: descriptivas y predictivas. Las técnicas descriptivas nos indican las propiedades de los datos, mientras que las técnicas predictivas permiten realizar inferencias de manera que se pueda predecir información.

2.5.1 Técnicas de minería de datos

De acuerdo con [14], las técnicas de minería de datos pueden clasificarse de la siguiente manera:

a. Clasificación de datos

Es una técnica que se basa en crear modelos a partir de etiquetas y categorías conocidas. Se entrena al modelo para que identifique las características de los datos conocidos. El modelo creado es utilizado con nuevos datos que aún no han sido categorizados. Al ser evaluado por el modelo, se determina a que clase pertenece cada dato. En la figura 2.2, se observan los esquemas de los algoritmos de árboles de decisión y del algoritmo SVM(Support Vector Machine) para la clasificación de datos.

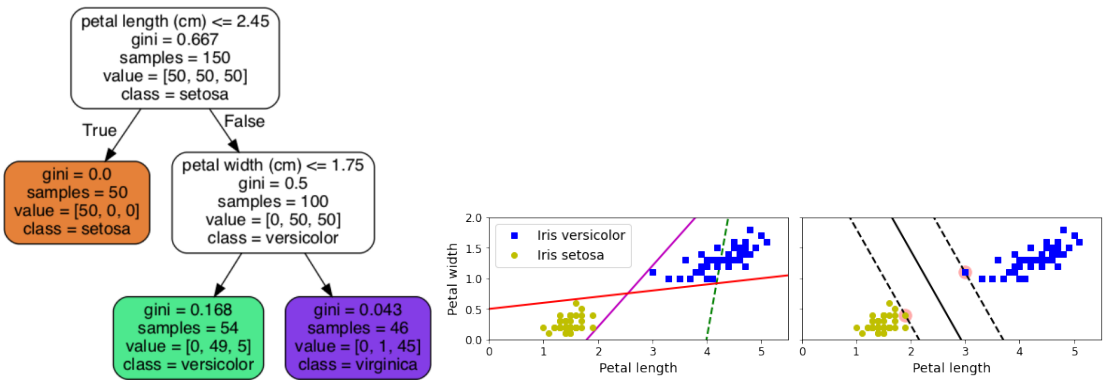


Figura 2.2: Los algoritmos de árboles de decisión (izq.) y el algoritmo SVM (der.) para la clasificación de flores basándose en las características de sus pétalos. Fuente: [15]

b. Predicción de datos

Bajo esta técnica se pueden bosquejar datos faltantes dentro de un conjunto. Esta técnica se diferencia de la clasificación ya que en esta última el objetivo se basa en asignar etiquetas a las diversas clases de datos, de manera que los datos que se presenten sean asignados a cada una de estas clases. La predicción se enfoca en la

probabilidad de que se presenten ciertos patrones, características o valores en datos que no se encuentran en el conjunto de datos inicial.

c. Agrupación de datos

Esta técnica se basa en la separación en grupos a los datos que comparten características similares. Es una técnica autónoma en la que las etiquetas de clase no son predefinidas, sino que estas son determinadas automáticamente por el algoritmo utilizado en base a la similitud que exista entre los datos. En la figura 2.3, se aprecia de manera muy visual la clasificación de datos mediante la utilización del algoritmo k-Means

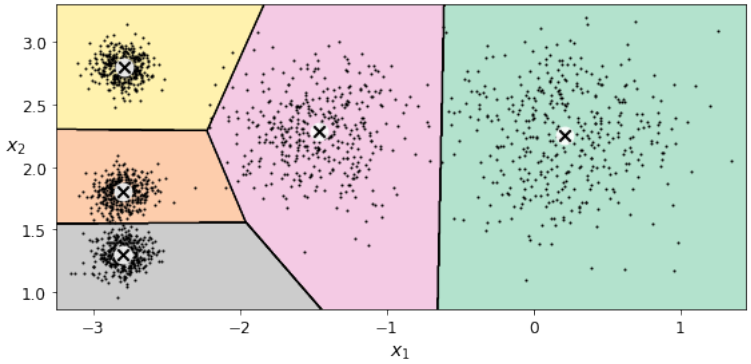


Figura 2.3: El algoritmo k-Means es muy utilizado para realizar clasificación de datos creando grupos o clusters. Fuente: [15]

d. Análisis de anomalías

Esta técnica busca identificar, dentro de un conjunto de datos, cuales de estos datos presenta anomalías con respecto a los otros basándose en su comportamiento y características que posee. La detección de anomalías es un proceso complicado debido al nivel de ruido que puede presentarse dentro de los datos, o la falta de comprensión del negocio, lo que podría significar un obstáculo en la identificación correcta de anomalías en los datos. En la figura 2.4, se aprecia como el algoritmo de Mezcla Gaussiana separa los datos anómalos (puntos rojos del diagrama) del resto de datos.

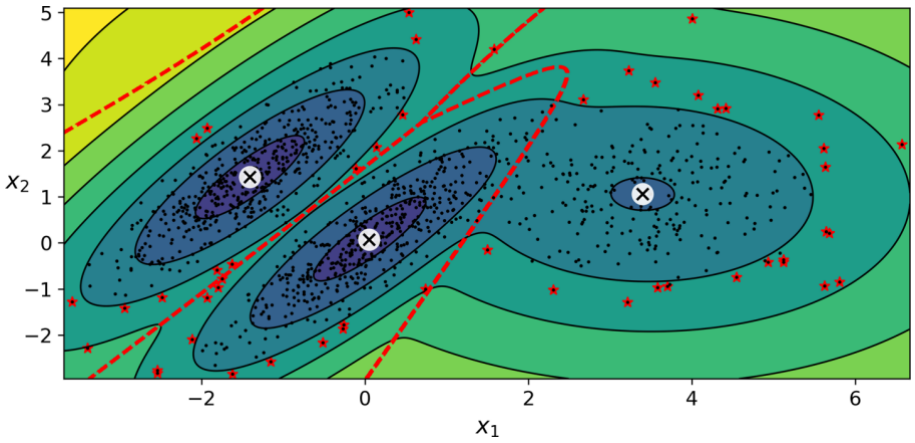


Figura 2.4: El algoritmo de Mezcla Gaussiana es utilizado para la detección de datos anómalos. Fuente: [15]

e. **Reglas de asociación** Esta técnica permite encontrar pertinencia entre diversos datos. Se utiliza principalmente en el aspecto comercial, donde se analizan transacciones de para identificar patrones de compra por parte de los clientes. En la figura 2.5, se observa el esquema de funcionamiento del algoritmo Apriori para creación de reglas de asociación.

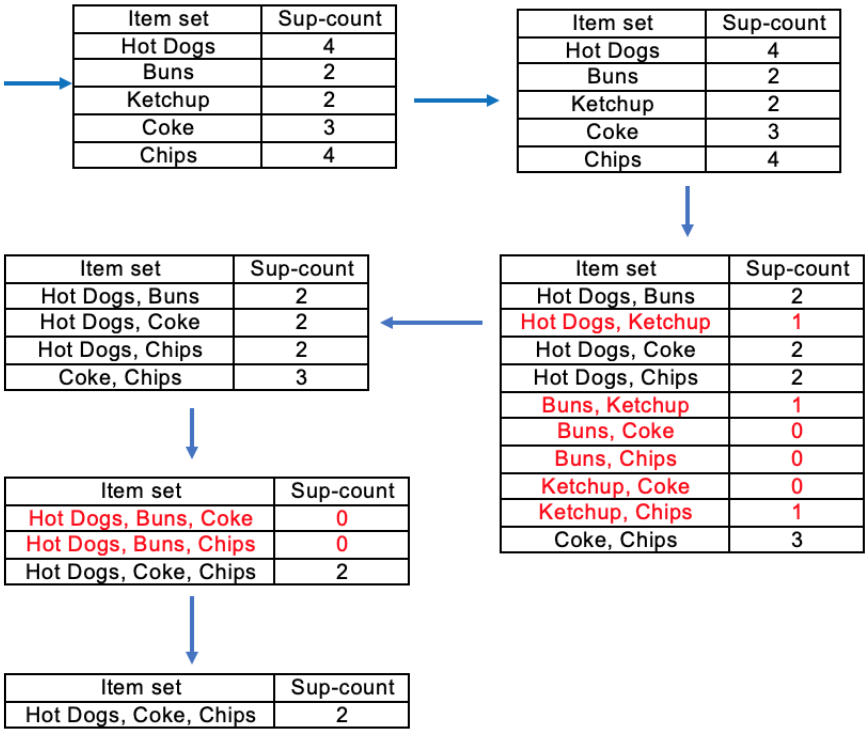


Figura 2.5: El algoritmo Apriori crea reglas de asociación a partir de relaciones entre conjuntos frecuentes de datos. Fuente: [15]

2.5.2 Proceso de minería de datos

De acuerdo con [16], la minería de datos podría verse más como un proceso que como un conjunto de herramientas. En [14], [16], [17] se mencionan varios enfoques para llevar a cabo este proceso, sin embargo en base a [17] se hace un mayor énfasis en el entendimiento del negocio, además de abarcar el proceso de manera mas general. El proceso en cuestión consta de los siguientes pasos:

a. **Dominio de negocio:** Identificar y comprender el dominio de la aplicación, y determinar los objetivos que la empresa desea alcanzar aplicando de la minería de datos.

b. **Obtención de datos:** Seleccionar un conjunto de datos, enfocándose en las variables necesarias para realizar la búsqueda de conocimiento.

c. **Limpieza y preprocesamiento de datos:** Determinar estrategias para mitigar el ruido en los datos seleccionados. Este ruido puede ser provocado por variables innecesarias, anomalías, duplicaciones, datos faltantes, etc.

d. **Reducción de datos y proyección:** Encontrar características que permitan representar los datos de manera adecuada dependiendo del contexto. Con la reducción de dimensionalidad se disminuye la complejidad requerida para obtener información relevante.

e. **Emparejar objetivos con minería de datos:** Emparejar los objetivos del proceso, aplicando algún método particular de minería de datos. Dentro de estos métodos se tiene la clasificación, la asociación, agrupamiento, etc.

f. **Análisis exploratorio:** Seleccionar métodos y algoritmos de minería de datos para encontrar patrones o información relevante dentro de los datos. En este paso definen criterios y parámetros dentro de los datos para que sean analizados.

g. **Interpretación:** En esta fase se procede con la interpretación los patrones minados. Aquí se aplica el enfoque iterativo de la minería de datos, ya de que ser necesario se debe regresar a pasos anteriores si los resultados obtenidos no satisfacen los objetivos que se esperaban alcanzar

h. **Actuar sobre el conocimiento descubierto:**
Tomar Varios caminos a partir del conocimiento descubierto. Es posible utilizar este

nuevo conocimiento en otros sistemas para tomar acciones, o simplemente documentarlo y reportarlo.

En la figura 2.6, se visualiza el proceso genérico dentro del descubrimiento de conocimiento en bases de datos.

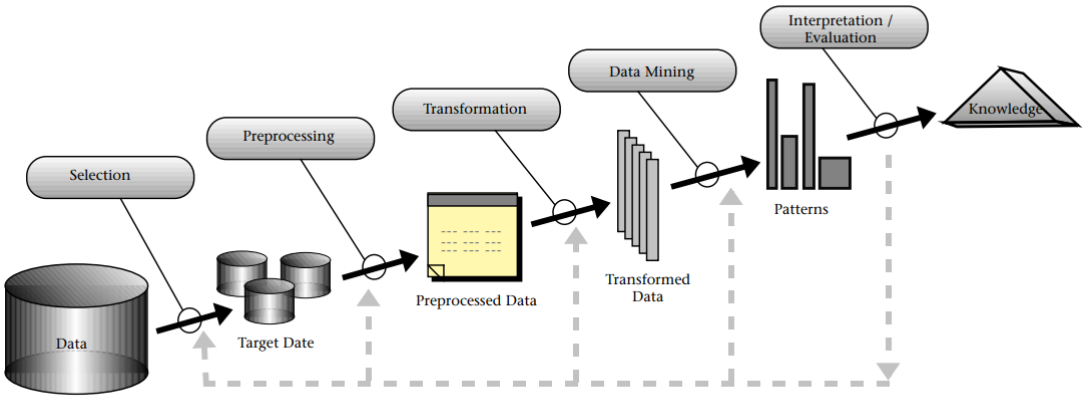


Figura 2.6: Esquema del proceso KDD. Fuente: [17]

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 [18] 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 [19], 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) [20], 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[20].

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

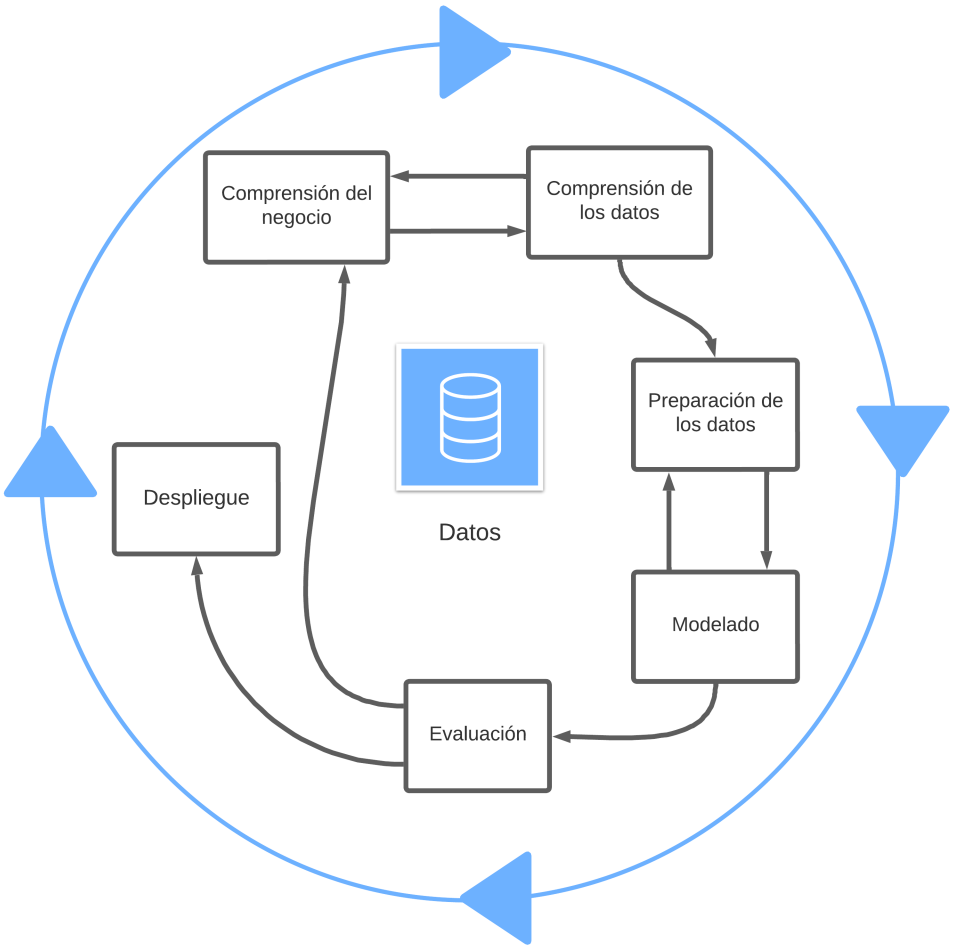


Figura 3.1: Fases de la metodología CRISP-DM. Adaptado de [20].

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 [20]. 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 [20]. 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 [20]. 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 [20].

- ❒ **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 [20].

- ❒ **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 [20].

- ❒ **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 [20].

- ❒ **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 [20].

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 [20].

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

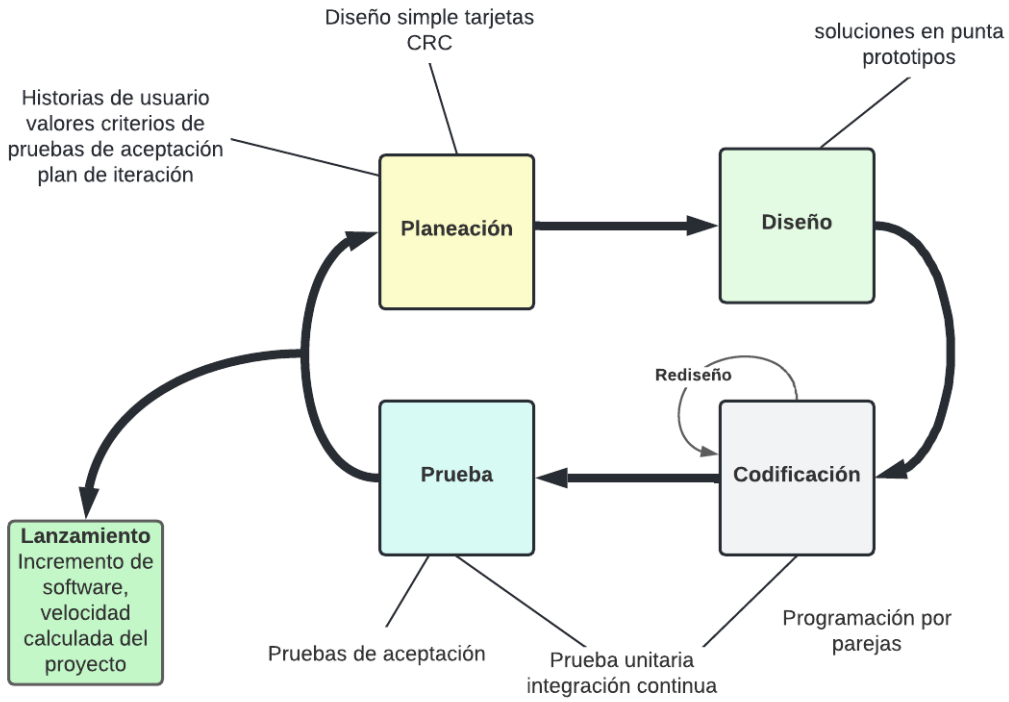


Figura 3.2: Fases de la programación extrema [21].

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 [21].

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 [21].

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 [21].

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 [21]. 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 [21].

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 [21].

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 [21].

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 [21].

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 [21].

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 [22], 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 en base a 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 [23]. 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.

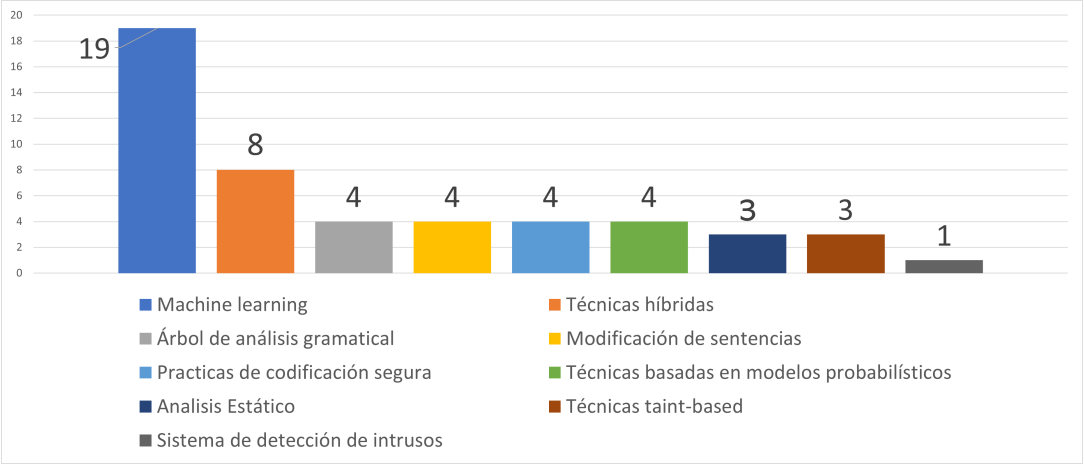


Figura 4.1: Distribución de clasificación de las técnicas de detección de SQLIA en los artículos analizados. Fuente: Los Autores

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.

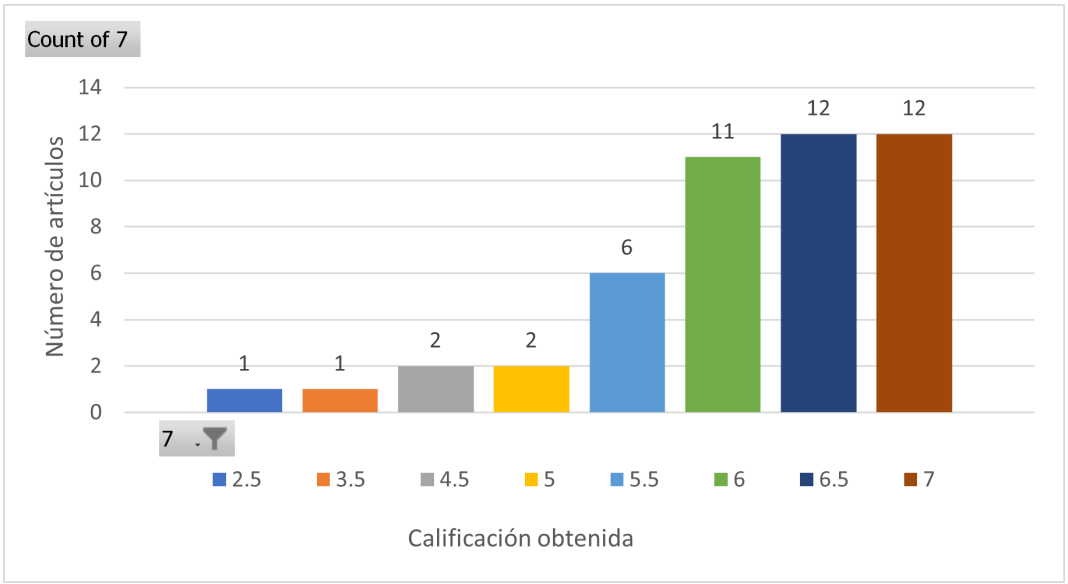


Figura 4.2: Distribución de calificaciones de la evaluación de calidad de los artículos analizados. Fuente: Los Autores

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 [24].
- **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 [24].
- **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 [24].

- **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 [24].
- **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 [12].
- **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 [12].
- **Técnicas híbridas:** algunas técnicas combinan un análisis estático durante el desarrollo con la combinación de una supervisión dinámica en tiempo de ejecución [24], como tal es el caso de AMNESIA [25], 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 [12].
- **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 [12].
- **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 [26], 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 describen de manera detallada cada uno de los pasos seguidos para el desarrollo del presente componente. En primer lugar, se describe el proceso de minería de datos CRSIP-DM en cada una de sus fases. Luego de culminar la última fase de CRISP-DM, se da inicio a la descripción del proceso de desarrollo del sistema de monitoreo, donde es aplicado el modelo generado por la metodología de minería de datos. Para este proceso de desarrollo se aplicó la metodología XP con cada una de sus etapas.

El proceso de CRISP-DM y de la metodología XP fue realizado de manera conjunta entre todos los miembros del equipo de desarrollo, debido a que existen elementos entre los distintos componentes que son comunes para mantener consistencia al momento de realizar las evaluaciones pertinentes. Los elementos que se mantienen en común en el desarrollo del proyecto son: planificaciones, cronogramas, historias de usuario, interfaces de usuario, y de manera general, todo el proceso metodológico se lo abarca de una manera muy similar en todos los componentes del proyecto desarrollado. En el documento desarrollado de cada uno de los componentes se describen de la misma manera los aspectos mencionados anteriormente.

5.1 PROCESO DE MINERÍA DE DATOS APLICANDO LA METODOLOGÍA CRISP-DM

A continuación se muestra la ejecución de la metodología CRISP-DM en cada una de sus fases:

5.1.1 Comprensión del negocio

En esta fase se determinaron las expectativas que tenía la CSIRT (Computer Security Incident Response Team) con respecto a la implementación de la minería de datos.

a. Determinación de los objetivos comerciales

A medida que aumenta la demanda de acceso a la información, los sistemas deben estar preparados no solo para soportar la carga que existe por parte de los usuarios, si no también estar preparados para evitar la exposición a posibles ataques. Por tal razón, el principal objetivo de la CSIRT es contar con herramientas que permitan detectar eficientemente posibles ataques a los distintos sistemas de información, y así tomar medidas que ayuden a mitigar estos ataques.

b. Evaluación de la situación

En cuanto a la evaluación de la CSIRT, es necesario tomar en cuenta los recursos disponibles para realizar la investigación. Para esto se consideran los siguientes recursos:

- **Personal:** Se cuenta con la asesoría de personal experimentado en el manejo de sistemas de información. Este personal permitió solventar las dudas correspondientes al acceso y manejo de registros en las bases de datos.
- **Datos:** Los datos provistos para la realización de la minería de datos provinieron de extractos fijos de diversos registros obtenidos de los sistemas de información utilizados por la organización
- **Riesgos:** El principal riesgo fue el cronograma y el tiempo requerido para realizar la investigación. Otro riesgo a tomar en cuenta fue la el acceso a los datos, debido a que estos son de uso interno de la organización y su divulgación podría comprometer la confidencialidad de la información.

Debido a que la investigación se realizó en un entorno académico, no se consideraron aspectos como costos y patrocinios en el proyecto.

c. Determinación de los objetivos de minería de datos

El objetivo principal dentro de la minería de datos en este proyecto, es producir un modelo que logre tomar registros de sentencias SQL y clasificar cuales de estos registros

son posibles SQLIA. Esta clasificación debe realizarse de la manera más eficiente posible debido a la gran cantidad de registros a analizarse. El modelo producido debe reducir la cantidad de falsos negativos en la mayor medida posible.

d. Producción de un plan de proyecto

El proceso de minería se lo realizó durante 4 semanas siguiendo el cronograma establecido en la Tabla 5.1.

Tabla 5.1: Cronograma para la ejecución del proyecto bajo la metodología CRISP-DM

Fase	Tiempo	Recursos	Riesgos
Comprensión del negocio	Semana 1	Miembros del CSIRT	Falta de disponibilidad de los expertos, Falta de entendimiento del negocio.
Comprensión de los datos	Semana 1 y 2	Miembros del CSIRT y equipos de desarrollo	Anomalías en los datos, confidencialidad de los datos.
Preparación de los datos	Semana 2 y 3	Equipo de desarrollo	Falta de comprensión en los datos, anomalías en los datos.
Modelado	Semana 3 y 4	Equipo de desarrollo	Dificultad en la implementación de los modelos requeridos.
Evaluación	Semana 4	Miembros del CSIRT y equipo de desarrollo	Resultados insatisfactorios, modelos que no se adapten a las necesidades del negocio.
Despliegue	Semana 4	Equipo de desarrollo	Dificultad para lograr los resultados esperados en un entorno de producción.

Una vez realizado el análisis de la organización, sus objetivos, necesidades, recursos y riesgos, es posible proceder con la siguiente fase de la metodología

5.1.2 Comprensión de los datos

En la siguiente fase se estudian más de cerca los datos provistos para obtener una idea más detallada del proceso que se debe llevar a cabo al realizar la minería de datos.

a. Recopilación de datos iniciales

Para la recopilación de datos iniciales se identifican los datos disponibles para el aná-

lisis. En este caso, se tiene el extracto de un log de sentencias SQL, este log fue obtenido de las bases de datos manejadas por la organización. Dicho log, se extrajo mediante el uso de scripts propios, razón por la cual siempre se mantiene la misma estructura dentro del archivo. La extensión del archivo es de tipo **JSON** (JavaScript Object Notation), un formato de texto para intercambio de datos, con un tamaño es de aproximadamente **2.5GB**

b. Descripción de los datos

La estructura del log cuenta con una estructura similar a la mostrada en el Esquema 5.1:

Esquema 5.1: Esquema del log

```
...
 "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": "...",
   },
```

```
28    "error": {
29        "message":"Provided Grok expressions do not match field value:[
30        timestamp=... ,process_id=... ,session_number=... ,user=... ,db
              =... ,app=....,client=... ,LOG: duration: ... ms bind ...:
              SELECT...
31        timestamp=... ,process_id=...,session_number=...,user=...,db=...,
              app=...,client=...,DETAIL ...
32       ...
33       ...
34       ...]",
35    }
36  },
37  "input": {
38     "type": "...",
39  },
40 ...
```

En el esquema anterior se aprecia en la línea 19 la llave `"message"`, cuyo valor representan los diversos logs que son de interés para el estudio. En cada línea como en la 20 y 21 se muestran diversos datos de la consulta SQL registrada, y al final se encuentra la sentencia SQL. Los registros del log se dividen en consultas y parámetros. En la línea 20 por ejemplo, el registro termina con la sentencia `SELECT` representando una consulta en SQL, y de la misma manera en la línea 21 se observa que el registro termina en `DETAIL`, donde se especifican los valores o parámetros utilizados en dicha consulta. A lo largo del log se mantiene la estructura mostrada anteriormente, y se repite con numerosas consultas de SQL.

c. Exploración de datos

A partir de la estructura obtenida de los logs, los datos o valores que parecen más prometedores de los registros es precisamente las consultas de SQL y los parámetros de dichas consultas.

d. Verificación de calidad de datos

En su gran mayoría, las consultas parecen mantener un formato consistente y fácil de procesar en siguientes etapas:

Dado que se pudo tener un acceso correcto a los datos para la evaluación, y se analizaron los campos de mayor valor, se puede proceder a realizar el procesamiento de dichos datos.

5.1.3 Preparación de los datos

Una vez comprendidos los datos que se utilizarán y el proceso a llevar a cabo para la minería de datos, se procedió a preparar los datos de una manera en la que se pueda aplicar un modelo de manera efectiva. Este proceso abarcó la limpieza, formato e integración de datos en un formato estándar.

a. **Selección de datos**

   Los campos seleccionados para la minería de datos son las consultas de SQL y sus respectivos parámetros. Cabe recalcar que se debió realizar un proceso más exhaustivo para unir las consultas con sus parámetros.

b. **Limpieza de datos**

   Para la limpieza de datos se tomaron solo los registros del log, para posteriormente realizar un parseo de cada registro. Si el registro correspondía a una consulta, esta fue separada del registro, de la misma manera si el registro correspondía a un parámetro.

c. **Construcción de nuevos datos**

   De manera paralela con la limpieza de datos, los parámetros fueron insertados en sus respectivas consultas, de manera que dichas consultas se completen y puedan ser analizadas en fases posteriores.

d. **Integración de datos**

   Debido a que los datos provienen de una sola fuente en un solo formato, no es necesario realizar una integración de los datos.

e. **Formato de datos**

   Para el formato de los datos, las consultas completas fueron almacenadas de manera secuencial en un archivo con extensión **CSV** para su posterior análisis.

Al finalizar la fase de preparación de datos se obtuvo un archivo que contiene aproximadamente 3 millones de sentencias de SQL.

5.1.4 Modelado

A partir de la revisión de literatura realizada, se obtuvo que uno de los algoritmos más utilizados en el campo de la detección de SQLIA es el Perceptrón Multicapa (Multilayer Përceptron o MLP por sus siglas en inglés). Razón por la cual se eligió este algoritmo para evaluar su rendimiento analizando las consultas obtenidas de la fase anterior. Es importante este análisis dada la gran cantidad de registros obtenidos

a. Selección de técnicas de modelado

Perceptrón Multicapa

De acuerdo con [27], los MLPs son suplementos de las redes neuronales de avance. Constan de 3 tipos de capas:

- ❒ **Capa de entrada:** En esta capa se recibe la señal a ser procesada.
- ❒ **Capa de salida:** En esta capa se obtiene la predicción o clasificación requerida.
- ❒ **Capa Oculta:** Pueden ser una o varias capas ocultas que realizan los cálculos computacionales necesarios para obtener predicciones, aproximaciones o clasificaciones.

En la Figura 5.1 se puede observar el esquema de un MLP.

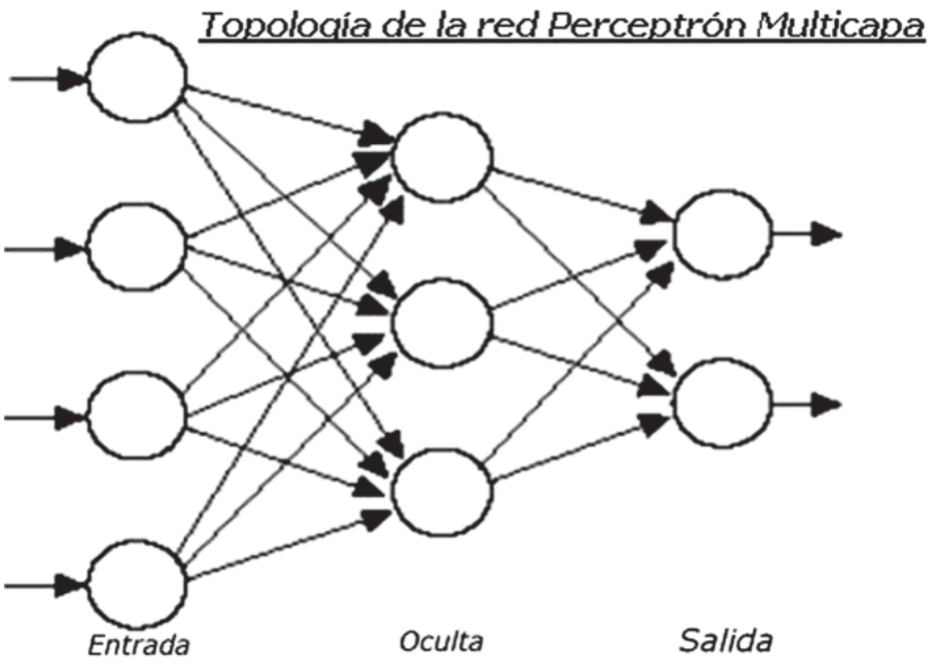


Figura 5.1: Esquema de un MLP. Fuente: [28]

De acuerdo con [15], las capas ocultas del MLP constan de pequeñas unidades lógicas de umbral o "*neuronas*". Cada una de las capas a excepción de la capa de salida se encuentran totalmente conectadas con la capa siguiente. Este algoritmo realiza una modificación a los algoritmos de perceptrones normales. Dado que en cada neurona se reemplaza la función de activación step (donde no funciona el descenso gradiente ya que este no se puede mover sobre una superficie plana), por alguna de las siguientes funciones:

- ❒ **Tangente hiperbólica:** $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- ❒ **Función logística o sigmoide:** $\sigma(x) = \frac{1}{1+e^{-x}}$
- ❒ **Función rectificador :** $ReLU(x) = max(0, x)$

El algoritmo MLP utiliza la técnica de backpropagation descrita en [29]. En esta técnica, se realizan diferentes pasadas por toda la red, cada pasada se conoce como *epoch*. Cada epoch funciona de la siguiente manera:

a) Se realiza la computación de cada una de las neuronas.

b) El resultado pasa a la siguiente capa como entrada, donde se procede a realizar los cálculos respectivos en dicha capa.

c) Una vez pasada por todas las capas, el algoritmo calcula el error de la salida, es decir, cuanto difiere el resultado obtenido del esperado.

d) Se calcula cuanto contribuyó cada conexión de salida al error aplicando la regla de la cadena del cálculo infinitesimal.

e) Se recorre en reversa cada capa aplicando el criterio mencionado anteriormente para calcular la contribución al error.

f) Finalmente, el algoritmo aplica el paso del Descenso Gradiente para ajustar los pesos de las conexiones en toda la red, utilizando el error que los gradientes calcularon.

b. Generación de un diseño de comprobación Una vez elegida la técnica de modelado, el siguiente paso es definir los datos con los cuales se comprobará dicho algoritmo. Para este caso se eligió un dataset obtenido de internet. Este dataset contiene sentencias aproximadamente 10 mil sentencias divididas entre legales e ilegales(con SQLIA) de manera uniforme. Además, se definieron las métricas necesarias para medir el rendimiento del algoritmo. Para los algoritmos de clasificación existen diversas métricas

para medir el desempeño de estos algoritmos. En [30]-[33], existe cierta concordancia en cuanto las métricas que son utilizadas para evaluar los modelos creados. Por lo cual se determinó que las siguientes métricas son las más útiles y más utilizadas para realizar la evaluación:

- **Matriz de confusión:** Es una matriz de 2x2 donde se muestras 4 valores muy importantes para el análisis de resultados en una clasificación binaria. Generalmente se distribuyen de la siguiente manera:
  - Los Verdaderos Positivos (True Positives o TP por sus siglas en inglés), que se encuentran en la esquina superior izquierda, y representa la cantidad de datos que fueron clasificados de manera positiva correctamente.
  - Los Verdaderos Negativos (True Negatives o TN por sus siglas en inglés), que se encuentran en la esquina inferior derecha, y representa la cantidad de datos que fueron clasificados de manera negativa correctamente.
  - Los Falsos Negativos (False Negatives o FN por sus siglas en inglés), que se encuentran en la esquina superior derecha, y representa la cantidad de datos que fueron clasificados de manera negativa incorrectamente.
  - Los Falsos Positivos (False Positives o FP por sus siglas en inglés), que se encuentran en la esquina inferior izquierda, y representa la cantidad de datos que fueron clasificados de manera positiva incorrectamente.

  En la Figura 5.2, se muestra el ejemplo de una matriz de confusión donde se clasifica si un dato representa un paciente con neumonía (clasificación positiva) o sin afectación (clasificación negativa).
- **Exactitud (AC):** Esta medida representa la proporción de predicciones correctas con respecto al total de predicciones realizadas. Está dada por la ecuación

$$AC = \frac{TP + TN}{TP + TN + PF + FN} \tag{5.1}$$

- **Sensibilidad o Recall (SN):** Representa la proporción de clasificaciones positivas realizadas correctamente con respecto a todas los datos que son realmente positivos. Está dada por la fórmula:

$$SN = \frac{TP}{TP + FN} \tag{5.2}$$

- **Especificidad (SP):** Representa la proporción de clasificaciones negativas rea-

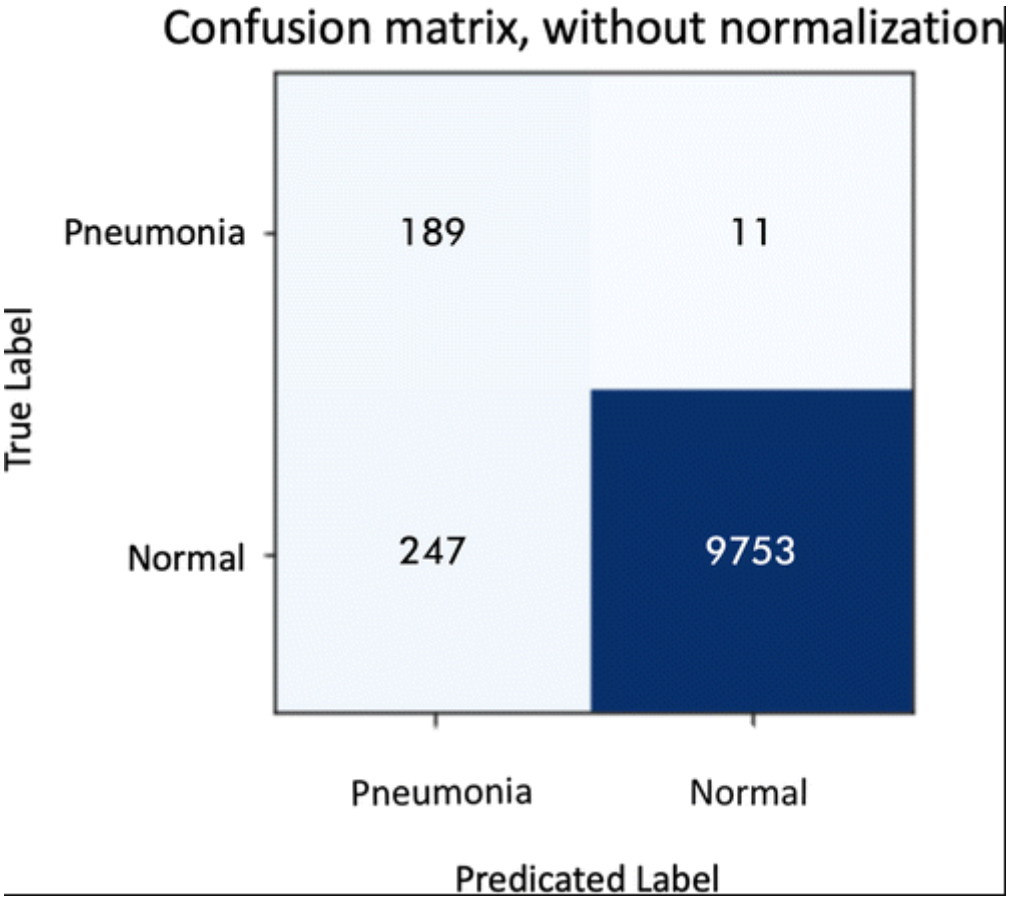


Figura 5.2: Ejemplo de una matriz de confusión. Fuente: [33]

lizadas correctamente con respecto a todas los datos que son realmente negativos. Está dada por la fórmula:

$$SP = \frac{TN}{TN + FP} \tag{5.3}$$

- **Precisión (P):** Representa la proporción de clasificaciones positivas correcta, con respecto a todas las clasificaciones tomadas como positivas. Está dada por la fórmula:

$$P = \frac{TP}{TP + FP} \tag{5.4}$$

- **F1-Score:** Se define como la media armónica entre la precisión y la sensibilidad. Esta métrica resulta de mucho interés, ya que a medida que su valor se aproxima a 1 (100 %), significa que existe armonía entre las métricas que relaciona, y el nivel de FP o FN es bastante reducido. Está métrica está dada por la fórmula:

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

c. Generación del modelo

Para realizar la generación del modelo, el primer paso es realizar el procesamiento de los datos de entrenamiento para que el algoritmo pueda realizar un procesamiento adecuado. En este caso se realizó la vectorización de las sentencias aplicando la técnica CountVectorizer. Esta técnica consiste en transformar una colección de texto en una matriz de conteo de aparición de cada token dentro del texto [34]. Una vez se encuentran tratados los datos, estos son divididos en dos conjuntos, uno de entrenamiento y otro de prueba. Los datos de entrenamiento son analizados por el algoritmo. Luego del entrenamiento, se procede con la predicción utilizando los datos de prueba. Luego de este proceso se generó un modelo que posteriormente es evaluado bajo las métricas definidas anteriormente.

d. Evaluación del modelo

Luego de generar el modelo y realizar las predicciones correspondientes se obtuvieron los siguientes resultados de las métricas definidas:

❒ **Matriz de confusión:**

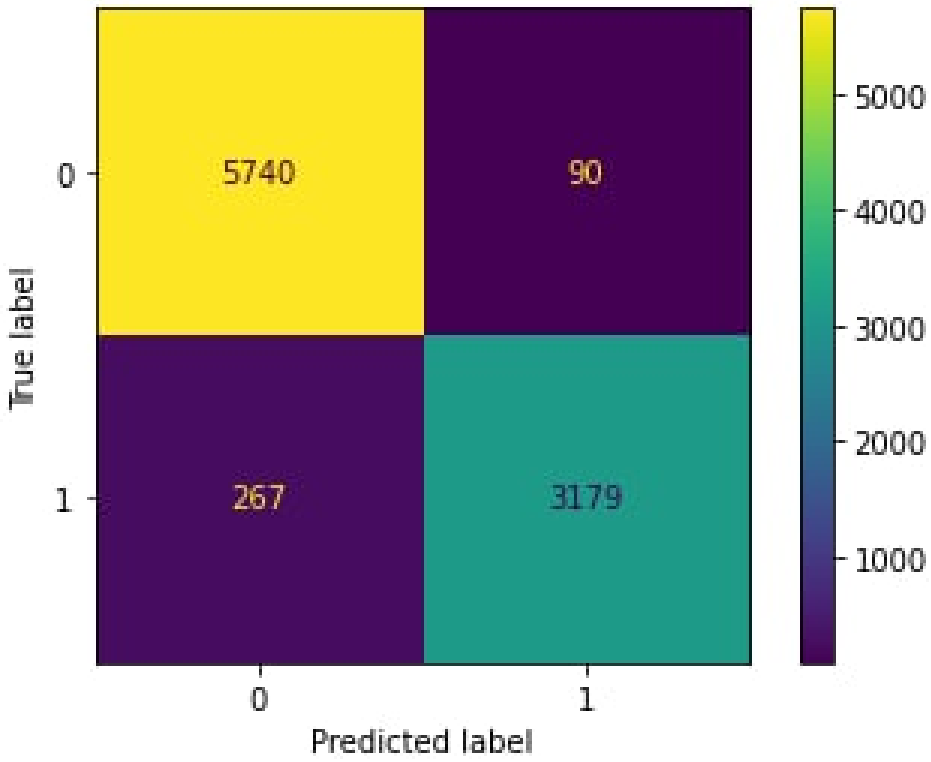


Figura 5.3: Matriz de confusión del algoritmo MLP . Fuente: El Autor

De la Figura 5.3, se obtuvieron los siguientes resultados

- ✧ $Verdaderos\ positivos\ o\ TP = 5740$
- ✧ $Falsos\ Positivos\ o\ FP = 90$
- ✧ $Falsos\ Negativos\ o\ FN = 267$
- ✧ $Verdaderos\ Negativos\ o\ TN = 3179$

❒ **Exactitud (AC):**

$$AC = \frac{5740 + 3179}{5740 + 3179 + 90 + 267} = 0.9615 = 96.16\,\% \tag{5.6}$$

❒ **Sensibilidad o Recall (SN):**

$$SN = \frac{3179}{3179 + 267} = 0.9225 = 92.25\,\% \tag{5.7}$$

❒ **Especificidad (SP):**

$$SP = \frac{5740}{5740 + 90} = 0.9845 = 98.45\,\% \tag{5.8}$$

❒ **Precisión (P):**

$$P = \frac{3179}{3179 + 90} = 0.9724 = 97.24\,\% \tag{5.9}$$

❒ **F1-Score:**

$$F1 - Score = \frac{2 * 0.9225 * 0.9724}{0.9225 + 0.9724} = 0.9468 = 94.68\,\% \tag{5.10}$$

De este resultado se puede observar que el modelo tiene un desempeño bastante alto, puesto que en todas las métricas sobrepasa el umbral del 90 %, sin embargo es necesario realizar las pruebas necesarias con el registro de sentencias SQL para determinar el desempeño del algoritmo en un ambiente real.

5.1.5 Evaluación

De la fase anterior, se obtuvo un modelo que fue evaluado con los registros del log brindado por la organización. A partir de los resultados obtenidos, se determinó el desempeño del modelo con respecto a aspectos comerciales.

❒ **Evaluación de los resultados**

El primer paso dentro de esta fase es la evaluación de los datos provistos por la organización bajo el modelo generado. Para esta evaluación se tomo una muestra de sentencias reales del registro dividas entre sentencias válidas y sentencias con SQLIA. Así se obtuvieron las resultados mostrados en la Figura 5.4.

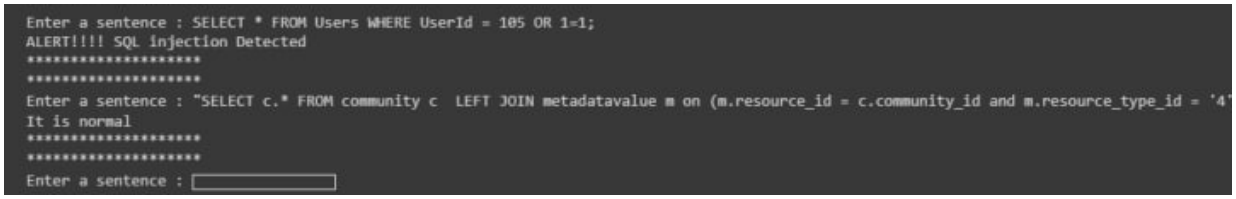


Figura 5.4: Resultados obtenidos de la evaluación del modelo utilizando una muestra de los registros reales provistos por la organización. Fuente: El autor

❒ **Proceso de revisión**

Como se observó en la Figura 5.4, los resultados de la evaluación con datos reales, resultaron favorables y reflejan de manera acertada la realidad de los datos. Se debe tener en cuenta que la sensibilidad roza el umbral del 90 %, que si bien aún es un valor aceptable, indica que existe una cantidad considerable de falsos negativos, cantidad que debería ser mínima para este caso de estudio.

❒ **Determinación de los pasos siguientes**

Como parte del proyecto se requiere hacer la implementación del modelo seleccionado en un sistema real, independientemente de los resultados. Esto se debe a que en el sistema se debe poder seleccionar entre una serie de modelos con la finalidad de comparar el desempeño de estos en un entorno real de producción.

5.1.6 Despliegue

En esta fase se describe a manera de resumen las salidas, resultados y descubrimientos obtenidos del proceso de minería de datos. Así también se realizaron los planes de implementación, mantenimiento, y una revisión final de este proceso.

❒ **Planificación de despliegue**

Del proceso de minería de datos es importante detallar los resultados obtenidos para su implementación dentro de la organización. Los artefactos obtenidos de este proceso son:

- ✧ Un modelo de machine learning aplicando el algoritmo MLP. Este modelo puede ser implementado en cualquier sistema para realizar evaluación de consultas de SQL y determinar la validez o no de estas.
- ✧ Un algoritmo de preprocesamiento de datos. Este algoritmo es muy importante ya que fue creado específicamente para la transformación y limpieza de datos provenientes de los logs brindados por la organización.

Es importante notar que los logs de la organización deben tener el mismo formato al que fue utilizado originalmente, caso contrario no podrán ser analizados por el modelo. Si se realizan cambios en este formato es necesario realizar modificaciones también en el algoritmo de preprocesamiento.

❒ **Planificación del control y del mantenimiento**

El control del modelo resulta intuitivo al tratarse de un algoritmo de clasificación binaria, por lo que se puede determinar con facilidad si los resultados obtenidos son confiables y reflejan de manera correcta la realidad de los datos. Como se indicó en el literal anterior, es importante que los logs ingresados para su análisis con el modelo cumplan con el mismo formato con el que se realizó el procesamiento inicial. Si bien el modelo no tiene una fecha de expiración, es necesario controlar la consistencia de los datos para poder asegurar que su funcionamiento sea el esperado. También es importante notar que el modelo utilizado puede ser mejorado en futuras iteraciones para aumentar el desempeño y obtener un mejor resultado en cuanto a sus métricas

❒ **Revisión final del proyecto**

En esta fase se realiza una retrospectiva sobre la realización del proceso de minería de datos, los retos que se presentaron, las lecciones aprendidas y los trabajos futuros. Para realizar este proceso fue muy importante analizar y comprender los datos provistos por la organización, caso contrario no sería posible realizar el preprocesamiento y limpieza de estos. No todos los algoritmos resultan efectivos para realizar minería de datos por lo que el análisis del algoritmo también representó un reto de realizar el proceso de CRISP-DM.

El modelo obtenido requirió ser evaluado por los usuarios mediante una interfaz web. En esta interfaz se realizó la evaluación del modelo utilizando un conjunto de datos mucho más grande para analizar el rendimiento en un entorno real. El desarrollo y evaluación de dicho sistema se detallan en los siguientes capítulos.

5.2 DESARROLLO DEL PROTOTIPO APLICANDO LA METODOLOGÍA XP

Para la evaluación del modelo obtenido de la minería de datos, se desarrolló un sistema web para poder evaluar datos de logs reales de manera intuitiva para el usuario, de manera que se pueda visualizar de mejor manera los resultados obtenidos. Este sistema se lo realizó aplicando la metodología XP, esto debido a su fuerte énfasis en la realización de pruebas, además de tratarse una metodología ágil, lo que favorece la flexibilidad y crecimiento del sistema en caso requerir nuevas funcionalidades. El desarrollo de este sistema permite obtener un mejor panorama del desempeño del modelo obtenido en la minería de datos, favoreciendo a la toma de decisiones y al análisis de los diferentes modelos que fueron implementados en este sistema.

5.2.1 Planeación

La primera fase del desarrollo aplicando la metodología XP, es la planeación. En esta fase se definen aspectos importantes como los requerimientos funcionales y no funcionales con los que debe constar el sistema. También se definen en esta fase, los roles necesarios para el desarrollo, así como un plan de entregas para las historias definidas

5.2.1.1 Definición de historias de usuario

Las historias de usuario definidas para este componente se obtuvieron mediante entrevistas a los stakeholders y un estudio sobre el dominio del negocio. Estas historias se describen en la Tabla 5.2.

5.2.1.2 Plan de entregas

El desarrollo de las historias de usuario se lo realizó en dos iteraciones de dos semanas de acuerdo al cronograma mostrado en la Tabla 5.2

Tabla 5.2: Historias de usuario para el desarrollo del sistema

Código	Título	Descripción	Prioridad	Esfuerzo
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	2	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	4	2
HU-03	Análisis	Como usuario deseo evaluar diferentes modelos de Machine Learning para detectar ataques de inyección SQL	3	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	1	1

Tabla 5.3: Cronograma de entregas de las historias de usuario

Código	Iteración
HU-01	1
HU-02	2
HU-03	2
HU-04	1

5.2.2 Diseño

En esta fase se definieron aspectos relacionados con el diseño arquitectónico y de interfaces de usuario del sistema desarrollado.

5.2.2.1 Diseño arquitectónico

Para el desarrollo de aplicación se optó por utilizar el patrón arquitectónico de software Modelo Vista Controlador (MVC). Este patrón permite realizar una correcta diferenciación entre el modelo (lógica del negocio, datos de la aplicación y reglas de la aplicación), vista (interfaces gráficas e interacción del usuario por medio de un cliente), y el controlador (mediador de la interacción del usuario por medio de la vista, y el modelo, realizando las operaciones necesarias para su comunicación). El uso de este patrón arquitectónico permi-

te que exista una mayor modularidad, escalabilidad y flexibilidad al momento de realizar su implementación [35].

Aplicando dicho patrón, se realizó un esquema de la arquitectura planteada que se observa en la Figura 5.5.

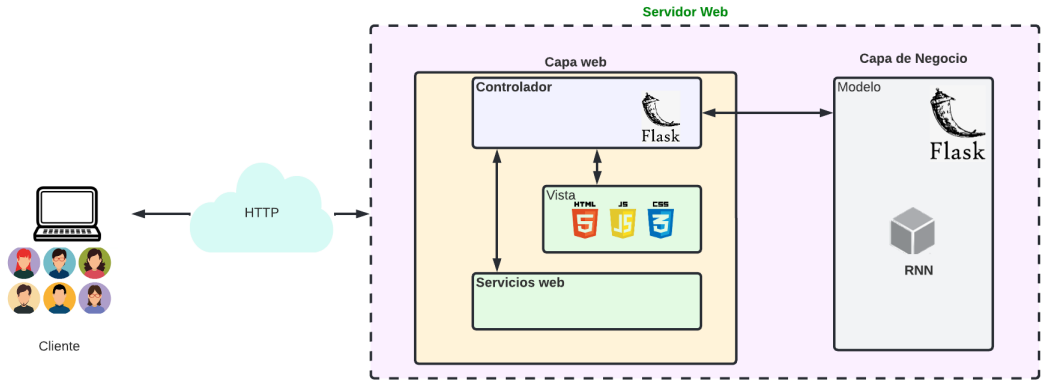


Figura 5.5: Diseño arquitectónico propuesto para el sistema realizado. Fuente: Los autores

5.2.2.2 Diagrama de actividades

Para obtener una mejor comprensión del flujo de actividades dentro del sistema, se realizó un diagrama de actividades que modele el comportamiento del sistema durante su uso. Este esquema se aprecia mejor en la Figura 5.6.

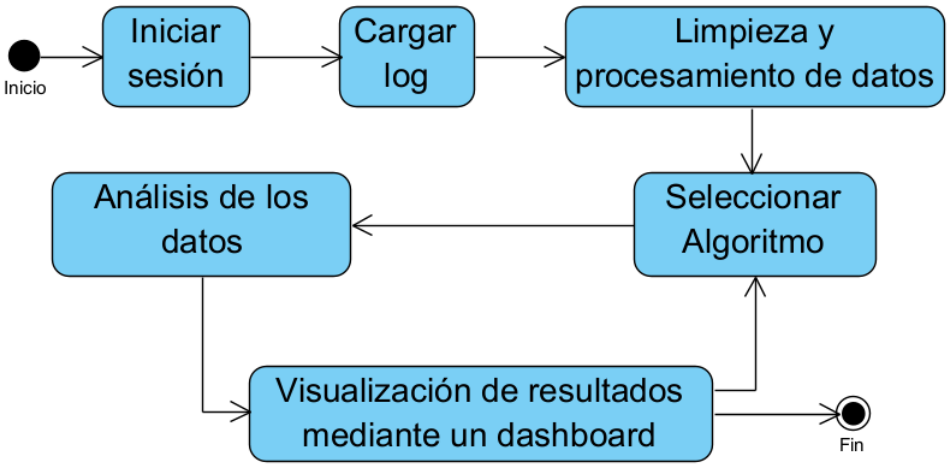


Figura 5.6: Diagrama de actividades del sistema realizado. Fuente: Los autores

5.2.2.3 Diseño de las interfaces

Una vez comprendido el flujo de actividades dentro del sistema, se realizaron bosquejos(mockups) de las interfaces gráficas de usuario se serían implementadas posteriormente. A continuación se muestran los mockups realizados para la realización del sistema:

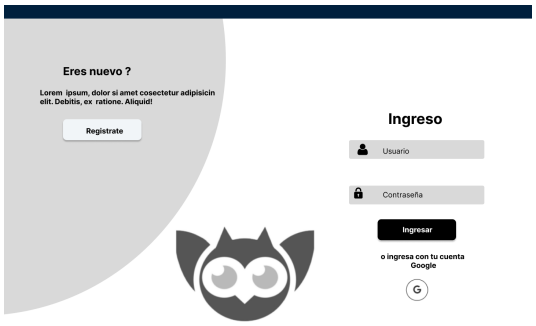


Figura 5.7: Mockup de la pantalla de inicio de sesión. Fuente: Los Autores

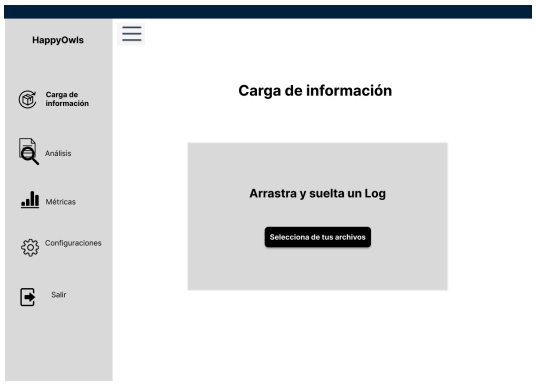


Figura 5.8: Mockup de la pantalla de selección y carga del archivo de logs. Fuente: Los Autores

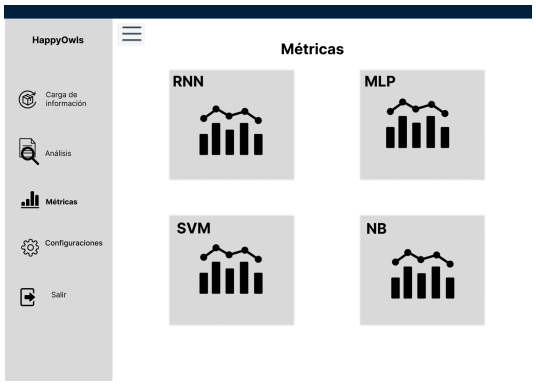


Figura 5.9: Mockup de la pantalla de selección del algoritmo para evaluar el log. Fuente: Los Autores

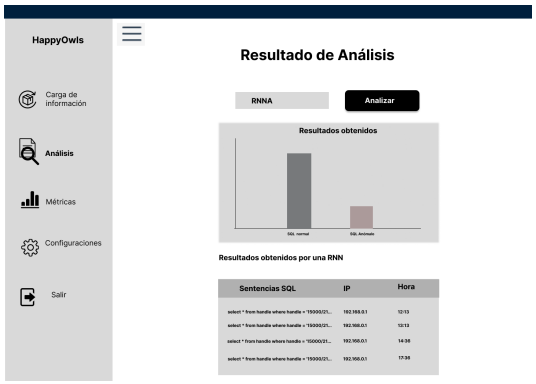


Figura 5.10: Mockup del dashboard de resultados del análisis del log. Fuente: Los Autores

5.2.3 Codificación

La codificación del proyecto se la realizó en dos iteraciones. Las cuales son detalladas con sus respectivos resultados a continuación:

5.2.3.1 Iteración 1:

En la primera iteración se desarrollaron las historias HU-01 y HU-04, dado que se determinó, en colaboración con los stakeholders que estas son las historias que aportan el mayor valor a la organización. En estas historias se define un control de acceso a la aplicación, esto es primordial debido a que los resultados provistos podrían comprometer la confidencialidad de la información. También es implementado el modelo que permite realizar la detección de los ataques, lo que representa la funcionalidad principal propuesta para el sistema desarrollado.

En la Figura 5.11, se muestra la implementación del control de acceso dentro de la aplicación.

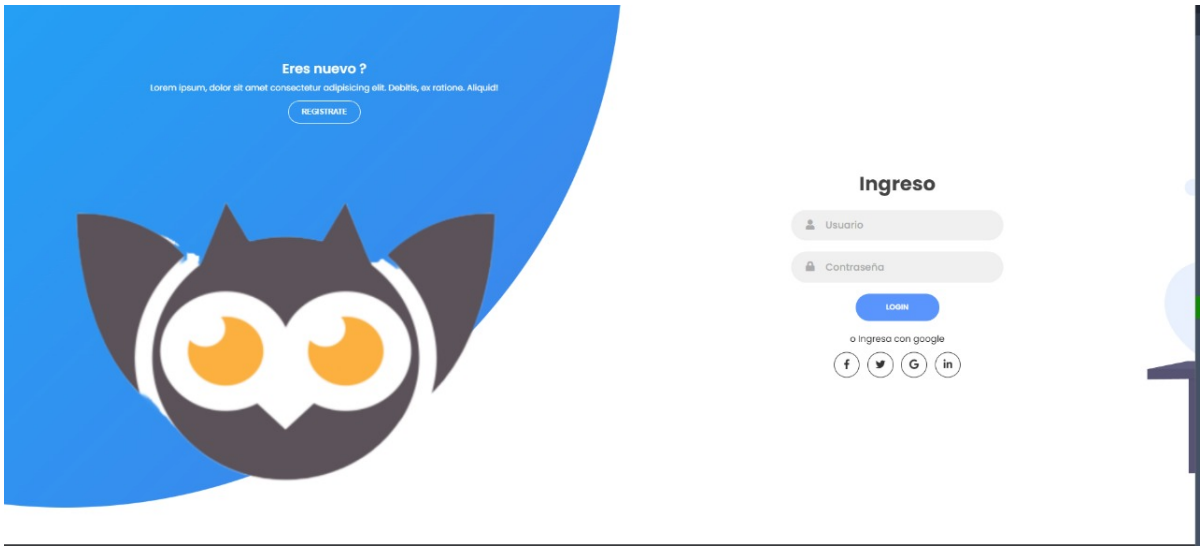


Figura 5.11: Interfaz de login implementada en el sistema. Fuente: El autor

En la Figura 5.12, se muestra la implementación de la funcionalidad principal del sistema, la cual es el uso del modelo generado por el proceso de minería de datos para realizar la detección de posibles SQLIA en los logs ingresados.

5.2.3.2 Iteración 2:

En esta iteración se realizó la integración de los modelos desarrollados en los otros componentes del proyecto, de manera que se tenga la opción de elegir entra cada uno de estos y realizar una comparativa y análisis de los mismos. También se añadió la funcionalidad que permite al usuario ingresar un log crudo, que el sistema procesa y limpia para poder ser

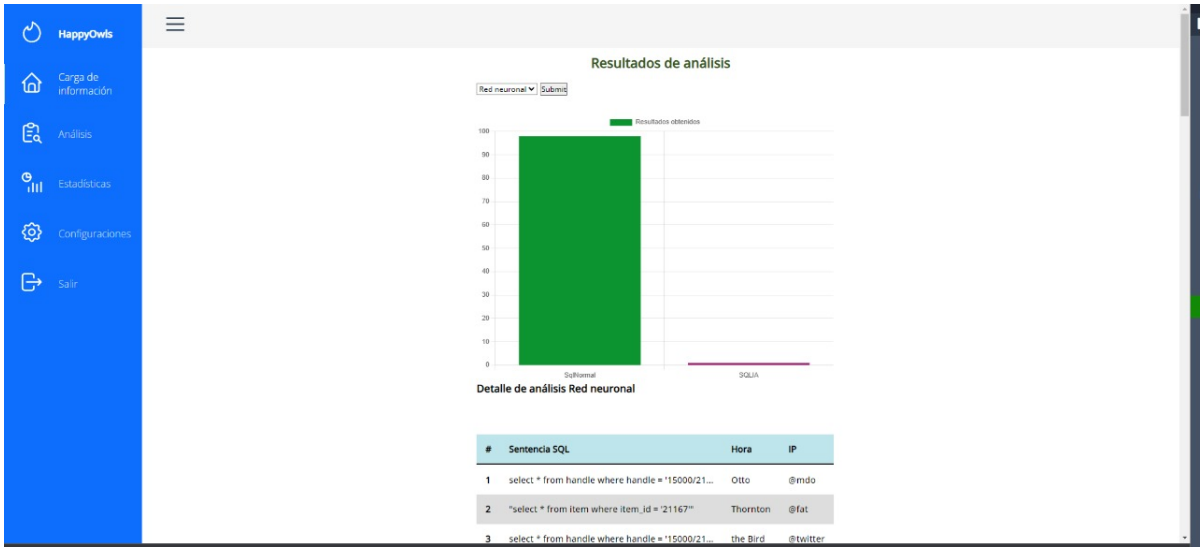


Figura 5.12: Interfaz de análisis de logs implementada en el sistema. Fuente: Los Autores

analizado por el modelo. Esta funcionalidad mejora la eficiencia del sistema, evitando que el usuario deba procesar manualmente el log.

En el Esquema 5.2, se muestra la integración entre los distintos algoritmos evaluados en los distintos componentes.

Esquema 5.2: Esquema de integración de los algoritmos analizados

```
df_sample = df.sample(n=100)
    df_sample = df_sample.apply(clean_data, axis=1)
    df_sample['Predict'] = ""
    if request.method == 'POST':
        opcion = request.form.get('selection')
        df_rnn = df_sample
        if opcion == 'rnn':
            for i in range(len(df_rnn) - 1):
                df_rnn['Predict'].iloc[i] = predict(df_rnn['QUERY'].values[i],
                    myModelRnn)
        elif opcion == 'svm':
            for i in range(len(df_rnn) - 1):
                df_rnn['Predict'].iloc[i] = predict(df_rnn['QUERY'].values[i],
                    myModelSvm)
        elif opcion == 'nb':
            for i in range(len(df_rnn) - 1):
                df_rnn['Predict'].iloc[i] = predict(df_rnn['QUERY'].values[i],
                    myModelGnb)
```

```
    elif opcion == 'mlp':
        df_rnn = df_sample
        for i in range(len(df_rnn) - 1):
            df_rnn['Predict'].iloc[i] = predict(df_rnn['QUERY'].values[i],
                myModelMlp)
    result = df_rnn['Predict'].value_counts().to_numpy()
    axis[0] = result[0]
    axis[1] = result[1]
```

En la Figura 5.13, se muestra la interfaz donde se permite realizar la carga de un log crudo para su posterior procesamiento y análisis

Figura 5.13: Interfaz carga de log en el sistema. Fuente: Los Autores

5.2.4 Pruebas

Luego de realizar la implementación del sistema utilizando el modelo generado en la metodología CRISP-DM, se procedió a realizar las pruebas respectivas del sistema utilizando datos reales provistos por la organización. Para la prueba se cargo un log comprimido en formato ZIP, que es descomprimido, procesado y posteriormente analizado por el modelo. El sistema se configuró para procesar un millón de sentencias SQL.

Una vez analizado el log en el sistema, se obtuvieron los siguientes resultados mostrados en la Figura 5.14:

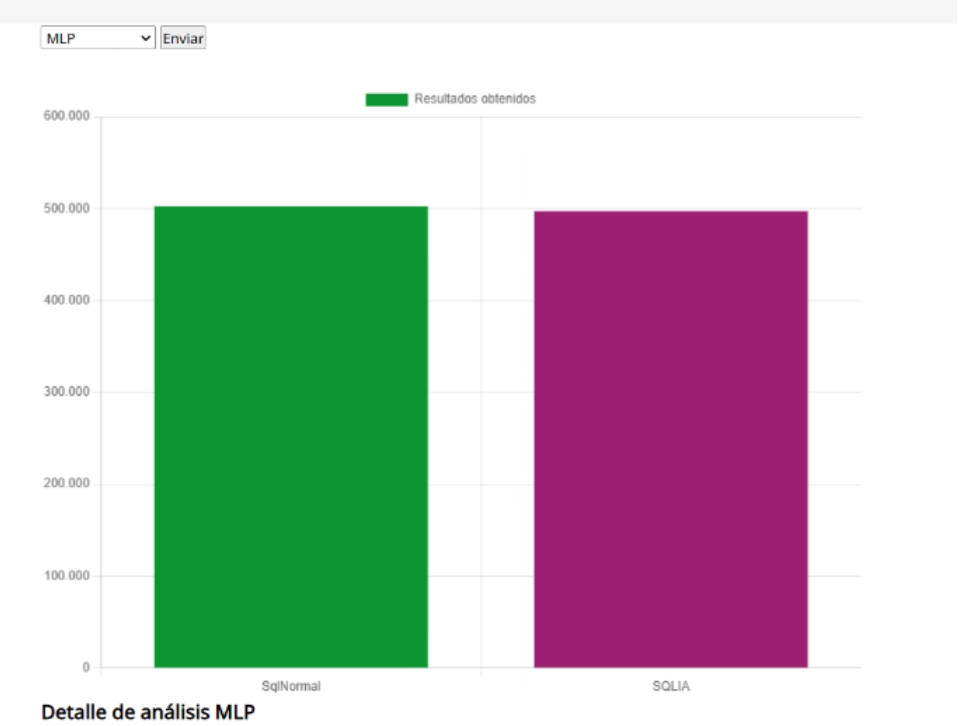


Figura 5.14: Resultado del análisis de un millón de sentencias dentro del log cargado en el sistema. Fuente: El autor

Como se puede apreciar, el sistema aún muestra una gran tendencia a detectar sentencias SQL válidas como posibles SQLIA, por lo que no resulta en un modelo muy efectivo para detectar estos ataques en este caso de estudio. Sin embargo, el modelo fue correctamente implementado en el sistema y la funcionalidad se encuentra integrada de manera adecuada.

6 ANÁLISIS DE RESULTADOS, CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

Una vez realizado el desarrollo del componente en cada una de sus fases, se realizó un análisis de los resultados obtenidos para así, extraer conclusiones, recomendaciones y remarcar las mejoras o extensiones que pueda realizarse a futuro en el trabajo realizado.

6.1 ANÁLISIS DE RESULTADOS

Por medio de los resultados obtenidos de la evaluación del modelo en el sistema web, se pudo determinar que el modelo no tiene un correcto desempeño, ya que el nivel de falsos positivos es bastante alto. Es necesario entonces realizar un análisis más profundo para determinar las posibles razones por las cuales el desempeño del algoritmo desciende tan drásticamente al evaluarse en un ambiente real, más aún, cuando dicho algoritmo resulta uno de los más utilizados de acuerdo con la revisión de literatura realizada. Este análisis queda fuera del alcance del presente componente de integración curricular.

6.2 CONCLUSIONES

- ❒ Se realizó un estudio de la literatura con respecto a las técnicas relacionadas con la detección y predicción de SQLIA, por medio de este estudio se determinaron las técnicas más utilizadas para realizar la predicción, donde prevalecen las técnicas basadas en Machine Learning. También se determinaron en este estudio los algoritmos más utilizados en el ámbito de Machine Learning para este fin. Para este componente se seleccionó el algoritmo MLP para realizar su evaluación y desempeño en la detección y predicción de SQLIA.

- ❒ Con la aplicación de la metodología CRISP-DM, se logró realizar el proceso completo de minería de datos hasta obtener un modelo de Machine Learning con el algoritmo MLP. El entrenamiento de este modelo mostró resultados favorables en cuanto a las métricas definidas para calificar su desempeño. Sin embargo, este modelo fue evaluado con datos reales de un caso de estudio, donde los resultados se mostraron menos favorables, razón por la cual convendría realizar un análisis de los datos evaluados para determinar la razón del bajo desempeño.

- ❒ Para complementar la evaluación del algoritmo seleccionado, se realizó un prototipo de sistema web aplicando la metodología XP. Mediante este sistema se logró que los stakeholders puedan realizar la evaluación de los modelos obtenidos de una manera mas intuitiva.

- ❒ Se realizó una evaluación del modelo generado por la metodología CRISP-DM por medio del sistema web desarrollado. En esta evaluación se utilizó una muestra grande de datos (más de un millón de datos), obteniendo resultados desfavorables en cuanto al desempeño del sistema, por lo cual se determinó que el algoritmo seleccionado no resulta óptimo para este caso de estudio.

6.3 RECOMENDACIONES

- ❒ Uno de los aspectos más importantes dentro del desarrollo de este componente fue la utilización de metodologías especificas dentro de cada fase, ya que estas facilitaron mucho el desarrollo del componente. Dicho esto, es importante que al momento de realizar un proyecto se seleccione las metodologías adecuadas para su desarrollo.

- ❒ Es indispensable realizar un análisis exhaustivo de los datos provistos para realizar el modelado, entrenamiento y pruebas de manera efectiva.

- ❒ Las metodologías de minería de datos (CRISP-DM), y de desarrollo de software (XP), son de naturaleza iterativa. Esto se debe tener en cuenta debido a que generalmente se debe dar cabida a los cambios que existan durante el desarrollo. De existir un cambio, siempre se debe tener medidas para actuar oportunamente y seguir adelante con el proyecto.

- ❒ Debido al volumen de datos utilizados para el desarrollo del componente, el poder

computacional representó una limitante al momento de realizar los análisis respectivo. Por esta razón es conveniente contar con equipos con altas prestaciones cuando se requiera trabajar con grandes volúmenes de datos. Además de las prestaciones del equipo, es importante conocer si las herramientas utilizadas se encuentran optimizadas para manejar esta cantidad de datos eficientemente.

6.4 TRABAJO FUTURO

Es importante recalcar las mejoras o extensiones que pueden realizarse para trabajos futuros a partir del trabajo desarrollado, y que quedan fuera del alcance delimitado inicialmente.

- ❒ En [1], se menciona que si bien los SQLIA son muy comunes, existen otros tipos de ataques de inyección como los ataques Cross Site Scripting (XSS), que pueden llegar a ser igual de perjudiciales, por esta razón resulta importante realizar un estudio de literatura analizando las técnicas que existen para la detección de otros tipos de ataques de inyección aparte de los SQLIA.
- ❒ Un aspecto relevante dentro del procesamiento de datos masivos es la velocidad; en este contexto, existen tecnologías propias de Big Data como son Apache Hadoop, Apache Spark, o el uso de multiprocesamiento, que no fueron consideradas en este componente debido a las limitaciones en cuanto al poder computacional. Por esta razón, se sugiere el uso de dichas tecnologías y herramientas para obtener un resultado más eficiente en cuanto a velocidad y tiempo de procesamiento, y comparar dicho resultado con los obtenidos en el presente trabajo.
- ❒ Debido al bajo desempeño del algoritmo MLP dentro del componente desarrollado, es conveniente realizar un análisis de la implementación del algoritmo para poder determinar las causas de este desempeño, y así poder optimizarlo para futuras implementaciones.

7 REFERENCIAS BIBLIOGRÁFICAS

[1] OWASP, *OWASP top 10:2021*, en, https://owasp.org/Top10/, Accessed: 2022-1-3, 2021.

[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] *Vulnerabilidades y violaciones digitales en productos de software: análisis de regresión logística*. DOI: 10.1109/ICCWS48432.2020.9292397.

[7] S. Samonas y D. Coss, «The CIA strikes back: Redefining confidentiality, integrity and availability in security.,» *Journal of Information System Security*, vol. 10, n.º 3, 2014.

[8] M. Chen, S. Mao e Y. Liu, *Big data: A survey*, 2014.

[9] W. Fan y A. Bifet, *Mining big data: current status, and forecast to the future*, 2013.

[10] M. Malik y T. Patel, «Database security-attacks and control methods,» *International Journal of Information*, vol. 6, n.º 1/2, págs. 175-183, 2016.

[11] *SQL Injection*, en, https://owasp.org/www-community/attacks/SQL_Injection.

[12] 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.

[13] D. J. Hand, «Principles of Data Mining,» *Drug Safety*, vol. 30, págs. 621-622, 7 2007, ISSN: 1179-1942. DOI: 10.2165/00002018-200730070-00010. dirección: https://doi.org/10.2165/00002018-200730070-00010.

[14] S. Agarwal, «Data mining: Data mining concepts and techniques,» en *2013 international conference on machine intelligence and research advancement*, IEEE, 2013, págs. 203-207.

[15] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. .ºReilly Media, Inc.", 2019.

[16] M. K. Obenshain, «Application of data mining techniques to healthcare data,» *Infection Control & Hospital Epidemiology*, vol. 25, n.º 8, págs. 690-695, 2004.

[17] U. Fayyad, G. Piatetsky-Shapiro y P. Smyth, «From data mining to knowledge discovery in databases,» *AI magazine*, vol. 17, n.º 3, págs. 37-37, 1996.

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

[19] K. S. Khan, G. Ter Riet, J. Glanville, A. J. Sowden, J. Kleijnen y col., *Undertaking systematic reviews of research on effectiveness: CRD's guidance for carrying out or commissioning reviews*, 4 (2n. NHS Centre for Reviews y Dissemination, 2001.

[20] 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*, Springer-Verlag London, UK, vol. 1, 2000, págs. 29-39.

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

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] S. Abirami y P. Chitra, «Chapter Fourteen - Energy-efficient edge based real-time healthcare support system,» en *The Digital Twin Paradigm for Smarter Systems and Environments: The Industry Use Cases*, ép. Advances in Computers 1, P. Raj y P. Evangeline, eds., vol. 117, Elsevier, 2020, págs. 339-368. DOI: https://doi.org/10.1016/bs.adcom.2019.09.007. dirección: https://www.sciencedirect.com/science/article/pii/S0065245819300506.

[28] G. A. T. Bayona e I. A. L. Salcedo, «Evaluación de las redes neuronales artificiales Perceptron Multicapa y Fuzzy-Artmap en la clasificación de imágenes satelitales,» *Ingenierı´a*, vol. 17, n.º 1, págs. 61-72, 2012.

[29] D. E. Rumelhart, G. E. Hinton y R. J. Williams, «Learning internal representations by error propagation,» California Univ San Diego La Jolla Inst for Cognitive Science, inf. téc., 1985.

[30] M. Hossin y M. N. Sulaiman, «A review on evaluation metrics for data classification evaluations,» *International journal of data mining & knowledge management process*, vol. 5, n.º 2, pág. 1, 2015.

[31] M. Grandini, E. Bagli y G. Visani, «Metrics for multi-class classification: an overview,» *arXiv preprint arXiv:2008.05756*, 2020.

[32] Z. Vujović, «Classification model evaluation metrics,» *International Journal of Advanced Computer Science and Applications*, vol. 12, n.º 6, págs. 599-606, 2021.

[33] B. J. Erickson y F. Kitamura, «Magician's corner: 9. Performance metrics for machine learning models,» *Radiology: Artificial Intelligence*, vol. 3, n.º 3, 2021.

[34] *skalearn.feature_extraction.text.CountVectorizer*. dirección: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html.

[35] J. Bucanek, «Model-view-controller pattern,» *Learn Objective-C for Java Developers*, págs. 353-402, 2009.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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.

[70] 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.

[71] 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.

[72] 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.

[73] 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.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

[78] 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.

[79] 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.

[80] 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.

[81] 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.

[82] 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.

[83] 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.

[84] 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.

[85] 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	[36]	166	2012	Técnica basada en Removing Sql query attribute values	Machine learning	—	— SQL 5.0	—
2	[37]	75	2013	Técnica basada en idMas-SQL	Machine learning	CNN	Base de datos SQL 5.0	Todos
3	[38]	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	[39]	52	2012	Técnica basada en Detection of SQL injection attacks	Machine learning	CNN	PostgreSql v 9.2.3	Todos
5	[40]	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	[41]	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	[42]	38	2020	Técnica basada en Machine learning predictive analytics to SQL injection attac	Machine Learning	SVM	N/A	Todos
8	[43]	28	2020	Técnica basada en Artificial neural network	Machine Learning	CNN	Generador de URL	Todos
9	[44]	27	2019	Técnica basada en Adaptive Deep Forest	Machine Learning	AdaBoost	Exploit-Db y wooyun-Db	Todos
10	[45]	27	2019	Técnica basada en Neural networks	Machine Learning	CNN	Generador de URL	Todos
11	[46]	26	2015	Técnica basada en SQLiDDs	Técnicas taint-based	K-meas	N/A	Todos
12	[47]	24	2017	Técnica basada en análisis estático y dinámico	Técnicas Hibridas	N/A	N/A	Todos
13	[48]	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	[49]	23	2012	Técnica basada en Progam tracing and machine learning	Técnicas híbridas	N/A	N/A	Todos
15	[50]	23	2019	Técnica basada en Diava	Modificación de sentencias	N/A	Almacenamiento en la nube, análisis de trafico de red	Todos
16	[51]	22	2019	Técnica basada en Removing the parameter values of SQL query	Modificación de sentencias	N/A	Aplicaciones web vulnerables	Todos
17	[52]	22	2019	Técnica basada en Machine learning for SQL injection detection	Modificación de sentencias	N/A	Aplicaciones web vulnerables	Todos
18	[53]	17	2020	Técnica basada en Ontology to cloud web security	Practicas de codificación segura	N/A	Información guardada en la nube	Todos
19	[54]	16	2014	Técnica basada en análisis estático y dinámico	Técnicas híbridas	N/A	N/A	Todos
20	[55]	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	[56]	14	2012	Técnicas basadas en filtrado de atributos	Prácticas de códificación	N/A	N/A	Todos
22	[57]	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	[58]	13	2017	Técnica basada en analisis de comportamiento	Otras técnicas	N/A	N/A	Solo los 6 tipos de SQLIA más básicos
24	[59]	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	[60]	13	2014	Tecnica basada en análisis estático y dinamico	Técnicas híbridas	N/A	N/A	Todos
26	[61]	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	[62]	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	[63]	12	2014	Técnica basada en la funcion de distribución Zeta	Técnicas basadas en modelos probabilísticos	N/A	Datos de ejemplo	Todoss
29	[64]	12	2016	Técnica basada en creación de reglas	Técnicas taint-based	N/A	N/A	Todos
30	[65]	12	2012	Técnica asado en creacioón de banco de consultas	Árbol de análisis gramatical	N/A	N/A	Todos
31	[66]	11	2013	Tecnica basada en análisis estático y dinamico	Técnicas híbridas	N/A	N/A	Todos
32	[67]	11	2021	Técnica basada en deep learning	Machine learning	CNN y MLP	Datos de ejemplo obtenidos de internet	Todos
33	[68]	11	2015	Técnica basada en machine learning	Machine learning	K-means	No especifica	Todos
34	[69]	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	[70]	9	2017	Técnica basada en Dynamic Analyzer and Testing Model	Taint-based Technique	N/A	datos reales de una configuración de prueba	Todos
36	[71]	9	2016	Técnica basada en Expectation Criterion	Probabilístico	N/A	datos de ejemplo	Todos
37	[72]	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	[73]	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	[74]	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	[75]	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	[76]	6	2019	Técnica basada en ResNet	Machine Learning	ResNet	uso de una herramienta (no especifica) y datos de internet	Todos
42	[77]	6	2015	Técnica basada en Dynamic SQLIA Detection (DSD)	Parse Tree	Dynamic SQLIA Detection (DSD)	datos reales de una configuración de prueba	Todos
43	[78]	5	2019	Técnica basada en rechazo	Análisis Estático	N/A	datos reales de una configuración de prueba	Todos
44	[79]	4	2015	Técnica basada en anomalias de rechazo	Analisis Estático	Linear Discriminant Analysis(LDA)	datos generados por un servicio HTTP	todos
45	[80]	4	2015	Técnica basada en una Red Neuronal	Machine Learning	Red Neuronal	datos de ejemplo	Todos
46	[81]	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	[82]	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	[83]	1	2021	Técnica basada en CNN-BiLSTM	Machine Learning	CNN-BiLSTM	datos obtenidos de sitios de internet	Todos
49	[84]	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	[85]	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
[36]	1	1	1	1	1	1	1	7
[37]	1	1	1	1	1	1	0	6
[38]	1	1	1	1	1	0.5	0	5.5
[39]	1	1	1	1	1	1	0	6
[40]	1	1	1	1	1	1	1	7
[41]	1	1	1	1	1	1	0	6
[42]	1	0.5	0.5	1	1	0.5	1	5.5
[43]	1	1	1	1	1	1	1	7
[44]	1	1	1	1	1	1	0	6
[45]	1	1	1	0	1	1	0.5	5.5
[46]	1	1	1	1	1	1	1	7
[47]	1	1	1	1	1	1	1	7
[48]	1	1	1	1	1	1	0.5	6.5
[49]	1	1	1	1	1	1	0.5	6.5
[50]	1	1	1	1	1	1	0	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	1	1	7
[54]	1	1	0.5	1	1	1	1	6.5
[55]	1	1	1	1	1	1	0	6
[56]	1	1	0.5	1	1	1	1	6.5
[57]	1	1	1	1	1	1	1	7
[58]	0.5	1	1	1	1	1	1	6.5
[59]	1	0.5	1	1	1	1	1	6.5
[60]	1	1	1	1	1	1	1	7
[61]	1	1	0	1	1	0.5	0	4.5
[62]	1	1	1	1	1	1	1	7
[63]	1	1	0	1	1	1	1	6
[64]	1	1	0	1	1	1	0	5
[65]	1	1	1	1	1	1	1	7
[66]	1	1	1	1	1	0.5	1	6.5

#	QA1	QA2	QA3	QA4	QA5	QA6	QA7	Total
[67]	1	1	0.5	1	1	1	1	6.5
[68]	1	1	0.5	1	1	0.5	1	6
[69]	1	1	0	1	1	0.5	1	5.5
[70]	1	1	0.5	1	1	1	0.5	6
[71]	1	0,5	1	0,5	1	1	1	6
[72]	1	0	0	0.5	0	0.5	0.5	2.5
[73]	1	0.5	1	1	1	1	1	6.5
[74]	1	1	1	0.5	1	1	0	5.5
[75]	1	1	1	1	1	1	0.5	6.5
[76]	1	1	1	1	1	1	0	6
[77]	1	0.5	0.5	1	1	1	0.5	5.5
[78]	0.5	0.5	0	0.5	1	1	1	4.5
[80]	1	1	1	1	1	1	1	7
[81]	1	0.5	0.5	1	1	1	0	5
[82]	1	1	1	1	1	1	1	7
[83]	1	1	1	1	1	1	1	7
[84]	1	0.5	0	1	1	0	0	3.5
[85]	1	1	1	1	1	1	0.5	6.5