PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE ESMERALDAS

ESCUELA DE SISTEMAS Y COMPUTACIÓN

TESIS DE GRADO

ANÁLISIS DE LAS VULNERABILIDADES EN SISTEMAS GESTORES DE BASES DE DATOS.

PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO DE SISTEMAS Y COMPUTACIÓN

AUTOR:

PAÚL ANDRÉS GUZMÁN QUESADA

ASESOR:

MGT. VÍCTOR XAVIER QUIÑÓNEZ KU

Esmeraldas – Mayo, 2019

TRIBUNAL DE GRADUACIÓN

Trabajo de tesis aprobado luego de haber dado cumplimiento a los requisitos exigidos por el reglamento de Grado de la PUCESE previo a la obtención del título de INGENIERO DE SISTEMAS Y COMPUTACIÓN.

……..

Mgt. Víctor Xavier Quiñonez Ku

Tutor de Tesis

……..……..

Mgt. Susana Gabriela Patiño Rosado

Lector 1

…….

Mgt. José Luis Carvajal Carvajal

Lector 2

……..…….…….

Mgt. Víctor Xavier Quiñonez Ku

Director de la Escuela de Sistemas y Computación

……..

Mgt. Maritza Demera Mejía

Secretaria general PUCESE

Esmeraldas, Ecuador, Mayo 2019

AUTORÍA

Yo, PAÚL ANDRÉS GUZMÁN QUESADA portador de la cédula de identidad No. 0802978825 declaro que los resultados obtenidos en la investigación que presento como informe final, previo a la obtención del título de "Ingeniero de Sistemas y Computación" son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola, exclusiva responsabilidad legal y académica.

……..…….....

Paúl Andrés Guzmán Quesada

C.I. 080297882-5

CERTIFICACIÓN

Mgt. Víctor Xavier Quiñonez Ku docente investigador de la PUCESE, certifica que:

La investigación realizada por PAÚL ANDRÉS GUZMÁN QUESADA bajo el título

"ANÁLISIS DE LAS VULNERABILIDADES EN SISTEMAS GESTORES DE BASES DE DATOS" reúne los requisitos de calidad, originalidad y presentación, además de haber sido incorporadas al documento final, las sugerencias dadas; en consecuencia, está en condiciones de ser sometida a la valoración del Tribunal encargada de juzgarla.

Y para que conste a los efectos oportunos, firma la presente en Esmeraldas, mayo del 2019.

Mgt. Víctor Xavier Quiñonez Ku.

ASESOR

DEDICATORIA

Este trabajo de investigación va dedicado a mis padres que gracias a la paciencia y apoyo recibido por ellos fue que he logrado realizar todo lo que he hecho hasta ahora. Por ser mi motivación para seguir adelante a pesar de las dificultades que se han presentado y a todas las personas que me apoyan y quieren lo mejor para mí.

AGRADECIMIENTO

Agradezco principalmente a mis padres por ser quienes han estado junto a mí en todo el transcurso de mi vida estudiantil, por siempre estar ahí cuando más los he necesitado, por el apoyo y consejos que me han dado cada vez que he tenido dudas o dificultades, y por todos los valores que me han inculcado para poder llegar a ser la clase de persona que soy.

A mi hermana por siempre estar pendiente de mí y cuidándome los pasos que doy desde pequeño, por su apoyo y los buenos momentos que hemos tenido como hermanos, a mi hermano pequeño que, a pesar de todo, verlo con ganas de superarse me da ánimos para seguir adelante y darle un buen ejemplo. A la señora que nos ha cuidado a los 3 desde pequeño y ser como mi segunda madre ya que siempre ha velado por nuestra salud y bienestar.

A aquellos maestros que han tenido el don de la enseñanza y que desde el jardín hasta la universidad han compartido sus conocimientos conmigo y han contribuido con mi formación académica.

Agradezco a los amigos que he conseguido a lo largo de todos estos años, por lo buenos momentos que hemos pasado y a todas las personas que me de una u otra manera me han ayudado, aconsejado o apoyado en algún momento de mi vida y de esa manera contribuido para crecer como persona.

RESUMEN

La presente investigación se realizó con el fin de comparar la respuesta de cada una de las bases de datos a la implementación de mecanismos de seguridad y su forma de implementarlos en las distintas bases de datos seleccionadas previamente, tales fueron: MySQL, PostgreSQL y SQL Server.

Para la ejecución de la investigación se recreó un ambiente en donde se instalaron los diferentes gestores de base de datos, además de una revisión bibliográfica de cada una de las bases de datos y artículos relacionados a seguridad en las mismas para determinar los puntos clave donde fue necesario centrar nuestra atención e implementar los mecanismos evitando en lo posible el surgimiento de vulnerabilidades que comprometan la integridad de la información almacenada, de esta manera se creó una lista de chequeos donde se anotaron los temas claves encontrados, clasificándolos y agrupándolos en cuatro grupos, los cuales fueron: Gestión de usuarios y roles, Gestión de privilegios, Métodos de encriptación y Creación de disparadores para auditoria.

Posteriormente se ejecutaron comandos DDL (Lenguaje de Definición de Datos) de manera individual a cada una de las bases de datos comparando el antes y después de la implementación del mecanismo, obteniendo resultados similares y en cierta forma diferentes dependiendo del comando utilizado. De los cuales resaltaron 2 en donde el uno se caracterizó por tener la posibilidad de configurar los mecanismos y aplicarlos de manera global a toda la base de datos, mientras su contraparte se diferenció de los demás por mantener sus configuraciones de manera independiente, teniendo que crear en algunos casos los mismos mecanismos para las diferentes bases de datos ya que no comparten los mecanismos de seguridad entre ellas.

Dado los resultados se logró identificar los puntos fuertes y débiles en cada base de datos partiendo de los resultados obtenidos a raíz de la implementación de los mecanismos de seguridad. Reafirmando lo dicho en los artículos bibliográficos sobre los puntos clave en donde se debe poner énfasis y fortalecer su seguridad para otorgar un alto grado de seguridad a la base de datos y de esta manera mantener segura la información almacenada en ellas.

Palabras clave: MySQL, PostgreSQL, SQL Server, base de datos, mecanismos, seguridad.

ABSTRACT

This research was conducted in order to compare the response of each of the databases to the implementation of security mechanisms and how to implement them in different databases previously selected, such as MySQL, PostgreSQL and SQL Server.

For the execution of the investigation an environment was recreated where the different database managers were installed, in addition to a bibliographic review of each of the databases and articles related to security in them to determine the key points where it was necessary to focus our attention and implement the mechanisms avoiding as much as possible the emergence of vulnerabilities that compromise the integrity of the stored information. In this way a list of checks was created where the key issues found were noted, classifying them and grouping them in four groups, which were: Management of users and roles, Management of privileges, Methods of encryption and Creation of triggers for audit.

Subsequently, DDL commands were executed individually to each of the databases comparing the before and after the implementation of the mechanism, obtaining similar results and in a certain way different depending on the command used. Of which 2 were highlighted where one was characterized by having the ability to configure the mechanisms and apply them globally to the entire database, while its counterpart differed from the others by maintaining their configurations independently, having to create in some cases the same mechanisms for different databases as they do not share the security mechanisms between them.

Given the results, it was possible to identify the strong and weak points in each database based on the results obtained as a result of the implementation of the security mechanisms. Reaffirming what has been said in the bibliographic articles on the key points where emphasis should be placed and their security strengthened in order to provide a high degree of security to the database and thus maintain the security of the information stored in them.

Keywords: MySQL, PostgreSQL, SQL Server, database, mechanisms, security.

ÍNDICE

INDICE FIGURAS

INTRODUCCIÓN

PRESENTACIÓN DE LA INVESTIGACIÓN

La presente investigación explica la implementación de mecanismos de seguridad que permite proteger en cierta medida las bases de datos evitando de esta forma las posibles vulnerabilidades, comparando el resultado en cada una de las diferentes bases de datos con la finalidad de determinar cuál de ellas nos proporciona el mayor nivel de seguridad necesario para su utilización y de esta manera lograr mantener la información segura.

Este proyecto de investigación consta de 6 capítulos, los cuales serán descritos a continuación:

El capítulo I consiste en el desarrollo del marco teórico donde se explicada cada uno de los conceptos y términos necesarios para un mejor entendimiento de la investigación.

El capítulo II explica la metodología que se utilizó para el desarrollo de la presente investigación, describiendo el tipo de investigación, los métodos, técnicas de procesamiento de información e instrumento utilizado.

El capítulo III manifiesta el proceso de construcción del ambiente de prueba junto con los comandos DDL utilizados y los resultados obtenidos a partir de la implementación de cada uno de los mecanismos de seguridad poniendo énfasis en las notables diferencias que se encontraron en las distintas bases de datos.

El capítulo IV contrasta los resultados obtenidos en la investigación con los encontrados en las investigaciones pasadas, de esta forma apoyando o refutando los resultados, y en cierta forma actualizando la información con la obtenida en el proyecto.

El capítulo V y VI se centran en las conclusiones y recomendaciones respectivamente entorno a los objetivos planteados en esta investigación, partiendo de los resultados obtenidos se pudo concluir las bases de datos que acogen los mecanismos de seguridad de manera más eficiente y la manera más adecuada de proteger la información en sus sistemas.

PLANTEAMIENTO DEL PROBLEMA

Los sistemas de gestión de bases de datos tienen como objetivo poder controlar lo que se realiza en ellas ya sea modificar, almacenar o extraer información de la misma, estos sistemas también constan de métodos para controlar el acceso que los usuarios tienen a los datos, mantener integridad de la información y poder recuperar datos en caso de una falla en el sistema.

Proporciona varios métodos para la visualización de la información almacenada, presentar los datos en varios formatos, un generador de informes y la graficación de información mediante gráficos o tablas.

Un sistema de gestor de bases de datos ideal sería aquel que cuente con una seguridad alta ya que se maneja información valiosa dentro de ellas, pero a la vez de poder ser entendible por el manejador para evitar no perder la ubicación de los datos.

Sin embargo, el avance de la tecnología hace que surjan nuevas formas de poder vulnerar la seguridad de los gestores de bases de datos, exponiendo la información valiosa a personas no deseadas o con malas intenciones. Por lo cual es indispensable saber sobre las diferentes formas de fortalecer la seguridad de las bases de datos mediante: protocolos de seguridad, auditorias, evaluación de los sistemas, monitoreo de los logs, identificación de los puntos más sensibles y propensos a ser vulnerados, tener un buen sistema de control de acceso.

Según [1] “La gran mayoría de los datos sensibles del mundo están almacenados en sistemas gestores de bases de datos comerciales tales como Oracle, Microsoft SQL Server entre otros”. Por lo que realizar una investigación sobre las bases de datos más utilizadas puede resultar útil, ya que al ser utilizadas por grandes cantidades de personas no proporciona un grado más de confianza para las empresas o entidades que adquieren los servicios de estos diferentes gestores.

Una de las maneras de evaluar su sistema de gestor de bases de datos es mediante el testeo de todas las posibles formas de acceder a ella, ya sea que lo realice el mismo administrador o un profesional de seguridad informática que realice “Hacking Ético”, como detalla [1]: “Hacking ético es en sí una auditoría efectuada por profesionales de seguridad de la información, quienes reciben el nombre de “pentester”. A la actividad que realizan se le conoce como hacking ético o pruebas de penetración”.

Como puede ser el caso de los Bancos que manejan grandes cantidades de dineros e información muy confidencial de sus clientes tienen como prioridad mantener su seguridad lo más alta posible para que no haya filtración de información ni de manipulación del dinero almacenado. Esto no quiere decir que los gestores que no sean de paga sean menos confiables, ya que, si se le presta la debida atención a la configuración y protocolos de seguridad, pueden llegar a serlos.

JUSTIFICACIÓN

Actualmente con el apogeo de la tecnología y su utilización en diferentes campos de trabajo, la ciencia dedicada a buscar y encontrar formas para poder penetrar y burlar la seguridad en los sistemas de igual manera han avanzado, ya sea para su posterior fortalecimiento en cuanto a seguridad se trata o para actos maliciosos con el fin de dañar la integridad de dichos sistemas.

Las posibles vulnerabilidades en su gestor de base de datos pueden traer consigo serias consecuencias, como una pérdida de dinero o de credibilidad hacia la seguridad de dicha organización. Por ende, es de vital importancia el mantenerla en un lugar donde no puedan ser manipuladas por personas ajenas a dichas entidades.

Debido a estos inconvenientes, surge la idea de la realización de esta investigación, con el fin de dar a conocer a los usuarios que administran una Base de Datos o personas que estén interesadas y requieran de información en lo que se refiere a seguridad y vulnerabilidades de los Sistemas Gestores de Base de Datos. Esto servirá como una forma de dar a conocer las posibles vulnerabilidades y ataques que se podrán realizar a los sistemas y a la vez, una medida de como contrarrestar estos ataques.

Aunque se sabe que la seguridad absoluta no existe, es importante tener en cuenta una serie de parámetros a cumplir al momento de seleccionar un SGBD a utilizar, entre los más importantes, debe de ser totalmente seguro para que la credibilidad e integridad de la información no se vea comprometida, tener un buen sistema de respaldo en caso de que la información se pueda perder, la interfaz para la administración de datos debe ser de fácil comprensión para el usuario quien lo controle.

OBJETIVO GENERAL:

Analizar las vulnerabilidades en los sistemas gestores de bases de datos (MySQL, SQL Server, PostgreSQL) mediante la realización de una serie de pruebas con el fin de medir la seguridad de las mismas.

OBJETIVOS ESPECÍFICOS:

- Identificar las principales características de los sistemas gestores de bases de datos a través de una revisión bibliográfica.
- Construir un entorno virtual con cada uno de los sistemas gestores de bases de datos para la configuración de mecanismos de seguridad.
- Ejecutar las pruebas de ataques a las bases de datos configuradas con los mecanismos de seguridad.
- Determinar las fortalezas y debilidades de los sistemas gestores de bases de datos, a partir de los resultados obtenidos de cada una de las pruebas realizadas.

CAPÍTULO I: MARCO TEÓRICO

ANTECEDENTES

Con el transcurso del tiempo se han desarrollado nuevas técnicas de almacenamiento de información en reemplazo al innecesario uso del papel y al proceso lento en la recolección de datos de manera manual. Es así como surgen las bases de datos, brindando un ahorro de espacio, la disminución del uso del papel y la agilidad con la que se puede consultar la información almacenada. Las bases de datos son aplicaciones informáticas creadas con el fin de almacenar información en ellas y gracias a su gestor facilita la manipulación de dichos datos [2].

En una investigación realizada, [1] afirma que una gran parte de los datos sensibles en el mundo son almacenados en sistemas gestores de base de datos (SGBD) comerciales tales como Microsoft SQL Server, Oracle entre otros, e intentar vulnerar su seguridad es uno de los objetivos de los criminales. Esto explicaría por qué han aumentado los ataques externos, tales como la inyección SQL a estos gestores de una manera agresiva en el 2009 en un 345%. Por lo que es de suma importancia mantener la seguridad de los gestores en un nivel alto para que así su integridad y fiabilidad no sea afectada.

Según [3] definió al SGBD como: "Una agrupación coordinada de programas, procedimientos, lenguajes, etc. que suministra los medios necesarios para describir, recuperar y manipular los datos almacenados en la base de datos, manteniendo su integridad, confidencialidad y seguridad." Este trabajo se relaciona con la investigación planteada, debido a que trata algunos de los temas importantes para el desarrollo de la investigación, tales como: tipos de seguridad informática, amenazas a la seguridad, medidas de prevención, todo esto orientado hacia la vulnerabilidad de los gestores de las bases de datos.

En la investigación realizada por [4], detalla que es muy habitual ver a los especialistas que desarrollan sistemas informáticos la manera en que centra su atención en lograr tener una seguridad de alto nivel en el sistema, dejando a un lado la seguridad de donde sus aplicaciones extraen la información para su funcionamiento.

Por lo descrito en esta investigación es recomendable mantener a la par la seguridad tanto en su aplicación como en la fuente de información por lo que una vulnerabilidad en su centro de información puede causar ya sea resultados erróneos o fallos en el sistema por falta de datos.

Según lo dicho por [5] en su tesis denominada "Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL", los SGBD no constan de una seguridad total ya que están en constante amenaza por diversos factores, por lo que es preciso analizar y mantenerse al tanto de cada módulo que el SGBD cuenta, tales como procesador LMD (Lenguaje de Manejo de Datos) , los ficheros, registros de peticiones y procesador de consultas. Aunque resulte que la seguridad absoluta no existe, si es recomendable la utilización de los SGBD con la mayor cantidad de métodos de seguridad o que vayan teniendo actualizaciones que rectifiquen cada vulnerabilidad nueva que va apareciendo.

Los SGBD, al igual que todos los sistemas informáticos, contienen vulnerabilidades que pueden ser aprovechadas por terceras personas no autorizadas para acceder a la información, vulnerabilidades que pueden ser debido a problemas de seguridad del software o a una mala configuración por parte del administrador de sistemas [6].

Este estudio determinó que una de las principales causas de fuga de información se debe a que el sistema no cuenta con una seguridad suficientemente robusta, ya sea por problemas en el software o la falta de configuración de administrador.

De acuerdo a lo dicho por [7], detalla que la supervisión de los usuarios privilegiados, es requisito para la gobernabilidad de datos y cumplimiento de regulaciones como SOX (Ley Sarbanes-Oxley) y regulaciones de privacidad. También, ayuda a detectar intrusiones, ya que muchos de los ataques más comunes se hacen con privilegios de usuario de alto nivel.

La supervisión dinámica también es un elemento esencial de la evaluación de vulnerabilidad, le permite ir más allá de evaluaciones estáticas o forenses. Un ejemplo clásico se aprecia cuando múltiples usuarios comparten credenciales con privilegios o un número excesivo de inicios de sesión de base de datos.

Como lo explica [8], los privilegios de base de datos se puede abusar de muchas maneras. El usuario puede abusar de privilegio para fines no autorizados. Este tipo de amenaza es más peligroso porque los usuarios autorizados están haciendo mal uso de los datos.

Es necesario desarrollar una política de seguridad acorde que permita al personal encargado la correcta toma de decisiones en la configuración y administración de este servidor; de manera que al presentarse un incidente de seguridad se pueda responder de manera adecuada tanto para evitar la fuga de información como para facilitar la recolección de evidencia de los hechos sucedidos [9].

1. MARCO TEÓRICO

1.1. Base de Datos

Con el uso del papel para la recolección de datos y el lento proceso de búsqueda, surge como solución la creación de las bases de datos. Ofreciendo un ahorro significativo en espacio, la velocidad con la que se realiza la búsqueda de información y el gasto innecesario de papel.

Las evoluciones de las bases de datos van desde el simple almacenamiento de ficheros, pasando por las bases de datos relacional hasta llegar a las bases de datos distribuidos en diferentes sitios conectados entre ellos. Figura 1

Años	
60´s y 70´s	Sistemas de ficheros y sistemas centralizados: un ordenador potente y terminales deficientes que accedían a los ficheros
80´s	Aparecen las bases de datos relacionales
80's y 90´s	Base de datos distribuidos, redes. Tecnología Cliente/Servidor. Un sistema de base de datos distribuidos se compone de un conjunto de sitios, conectados entre sí mediante algún tipo de comunicaciones.

Figura 1: Evolución Base de datos [5]

Una base de datos es un conjunto de datos almacenados sin redundancia en un soporte informático y accesible simultáneamente por distintos usuarios y aplicaciones. Lo datos deben de estar estructurados y almacenados de forma totalmente independiente de las aplicaciones que la utilizan [10].

Es una colección o depósito de datos integrados, con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real: los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas [11].

Por lo tanto, se puede definir a la base de datos como el conjunto de archivos relacionados entre sí que se encuentran almacenados sistemáticamente, los cual podrán ser visualizados y modificados posteriormente por personas que tengan los permisos correspondientes.

A la persona encargada de administrar la base de datos (BDD) se le brindará diversas operaciones que podrá aplicar en ella, tales como:

- Creación de tablas.
- Agregar datos nuevos en las diferentes tablas creadas.
- Realizar consultas de información referente a las tablas.
- Actualizar la información existente.
- Borrar datos existentes.
- Eliminar información, tablas o campos ya existentes en la BDD.

Como sugiere [3] para una mejor compresión de bases de datos, es requerido conocer 3 conceptos básicos que son:

- Campo. - Cada campo contiene un dato acerca de la entidad a tratar. Este debe ser único y ser de un tipo que englobe y no ocasione error al momento del ingreso de registros.
- Registro. - La asociación de información llenada en los diferentes campos acerca de una entidad u objeto.
- Tabla. - Está conformada por filas y columnas donde las filas son los distintos registros almacenados y las columnas son los campos.

1.1.1. Tipos de Base de Datos

De acuerdo con [12] existen diferentes tipos de modelado de base de datos, entre los cuales se encuentran lo más comunes que son:

1.1.1.1. Modelo Jerárquico

El modelo Jerárquico se enfoca en la estructuración de los campos en nodos convirtiéndola en una estructura similar a un grafo. Figura 2.

Cada uno de los nodos se encuentran interconectados entre ellos transformándolos en un árbol invertido. Los nodos superiores o nodos padres pueden tener uno o varios nodos que se desprende de él, denominados como nodos hijos.

Aquel nodo que no posee un nodo padre es llamado “Raíz” el cual será el tope del árbol, en cuanto a encontrar. los que no tiene hijos se los conoce como “Hojas”, para la búsqueda de un campo se requerirá empezar por el nodo raíz, desplazándose por los hijos hasta llegar con el campo

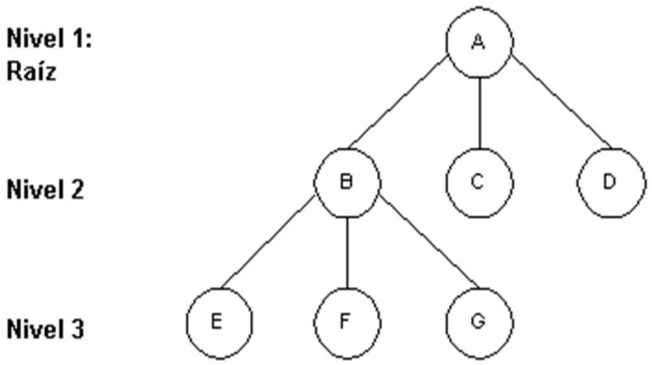

Figura 2: Estructura de un Árbol Jerárquico [12].

1.1.1.1.1. Características del Modelo Jerárquico

- Los nodos están enlazados mediante la relación de uno a muchos.
- Cada nodo tiene uno o más campos.
- Cada nodo padre puede tener uno o varios nodos hijos.
- Todo nodo hijo solo tendrá un único nodo padre excepto la raíz.

1.1.1.2. Modelo Red

Este modelado contiene cierta similitud con el modelo jerárquico ya que se manejan por medio de nodos, con la diferencia de que un nodo hijo puede tener más de un nodo padre, permitiendo así la relación de muchos a muchos, como lo muestra la Figura 3.

Consta de punteros los cuales se encargan de la conexión entre los diferentes tipos de nodos creando así vías de acceso alternativas a otros nodos.

Son representados como una versión mejorada del modelo jerárquico.

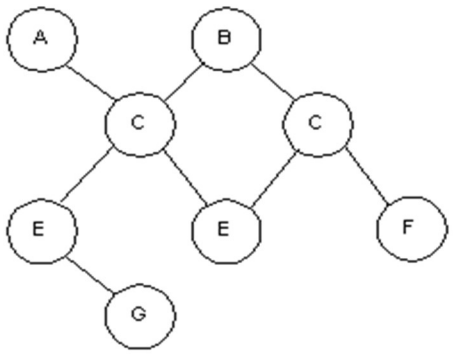

Figura 3: Estructura de datos en Red [12].

1.1.1.2.1. Características del Modelo de Red

- El nodo padre será denominado como propietario de sus nodos subordinados, mientras que los nodos subordinados serán denominados como miembros.
- Un nodo padre puede tener uno o varios nodos hijos, al igual que los nodos hijos pueden tener uno o más nodos padres.
- Sólo se permite que un registro miembro aparezca una vez en las ocurrencias de conjuntos del mismo tipo.
- Se pueden delimitar niveles múltiples de jerarquías donde un nodo puede ser miembro en un conjunto y al mismo tiempo propietario en otro conjunto diferente.

1.1.1.3. Modelo Relacional

Según [13] define al modelo relacional como un avance con respecto a los anteriores modelos, ofreciendo una mayor flexibilidad, ya que los datos son almacenados en diferentes tablas, este tipo de modelado tiene como característica principal el de poder relacionar diferentes tablas mediante campos.

Las tablas constan de filas y columnas, en donde las filas contienen los registros y las columnas los campos. Las tablas relacionadas constan de un campo en común, mediante el cual una tabla puede enlazarse con los datos almacenados en otra.

Este tipo de modelado es el mayormente utilizado en la actualidad gracias a su sencillez para la realización de operaciones tales como la inserción, modificación y eliminación de datos.

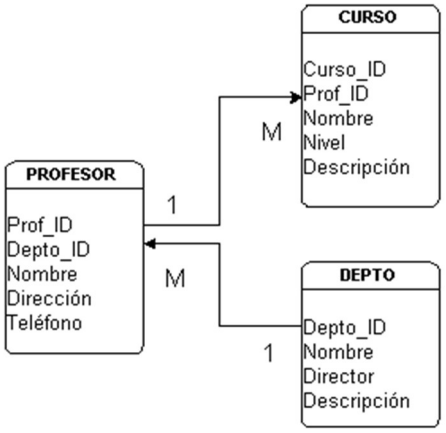

Figura 4: Estructura base de datos relacional [12].

1.1.1.3.1. Claves Primarias

Cada una de las tablas poseen diferentes campos, de los cuales habrá uno que es único e irrepetible, a este campo se le denominará como clave primaria, por medio de esta clave se podrá identificar a un registro de la tabla.

Además, una clave primaria puede ser simple (formada por un solo campo) o compuesta (formada por más de uno) [14].

1.1.1.3.2. Claves Foráneas

Como bien afirma [2] una clave foránea es aquella que identifica a una columna o varias de una tabla que se refiere a una columna o varios de otra tabla a la cual se referencia.

La columna de la tabla a la cual se está refiriendo debe de tener la clave foránea de la otra tabla como la clave principal en la tabla que se referencia.

1.1.1.3.3. Características del Modelo Relacional.

- Estructura los datos en forma de relaciones que se modelan mediante tablas de dos dimensiones. La estructura denominada "relación", permite representar tanto los objetos como las relaciones entre ellos.
- Está constituida por una extensión para cada una de las relaciones de su esquema.
- Está basado en un modelo matemático con reglas y algoritmos algebraicos
- establecidos, lo que permite el desarrollo de lenguajes de acceso y manipulación potentes.
- Permite la incorporación de aspectos semánticos del universo del discurso mediante el establecimiento de reglas de integridad. Estas reglas permiten trasladar al esquema conceptual restricciones o comportamientos de los datos presentes en el universo del discurso que no se podrían modelar exclusivamente con tablas.

1.1.1.4. Modelo Orientado a Objetos

Según [15] es un tipo de modelado relativamente nuevo que ha generado un gran interés, en el campo de las bases de datos.

Es una adecuación del paradigma de la programación Orientada a Objetos hacia las bases de datos. Se apoya en el concepto del encapsulamiento de información, atributos, características, etc.

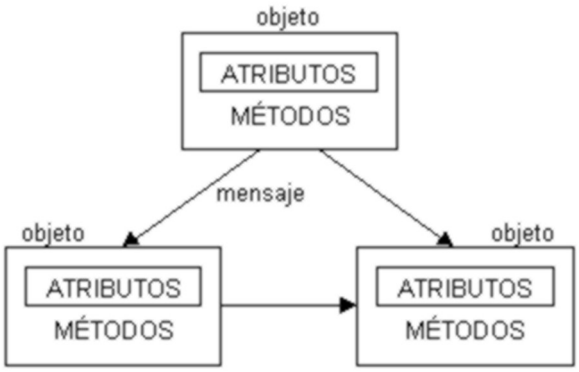

Figura 5: Estructura Base de Datos Orientada a Objetos [12].

1.1.1.4.1. Características del Modelo Orientado a Objetos

- Incluyen algún tipo de lenguaje para realizar consultas, lo cual permite que los objetos sean encontrados utilizando un enfoque de programación declarativa.
- Las aplicaciones multimedia se agilizan debido a que los métodos de clase asociados con los datos son responsables de una correcta interpretación
- Jerarquía de clases a partir de la que los objetos heredan comportamientos.
- Propiedad de una operación que permite aplicarse a objetos de distinta tipología.

1.2. Sistemas Gestores de Base de Datos

A grandes rasgos [3] define como Sistema Gestores de Base de Datos como a la capa intermedia que sirve como conexión entre la base de datos, las aplicaciones y las personas que lo manejan.

Este sistema permite la manipulación de los datos a través de una interfaz gráfica para visualizar la información de una mejor manera. De igual manera genera una gran cantidad de herramientas al encargado de administrar la base de datos tales como: herramientas para el desarrollo de aplicaciones, generador de informes, lenguaje específico de acceso a datos, entre otras.

1.2.1. Características de los SGBD

La mayoría de los sistemas gestores de base de datos consta de una gama de características que comparten entre ellos que son:

- Mantener de manera independiente la base de datos y los softwares que usan su información.

- Permitir que las personas puedan almacenar datos, modificarlos y visualizarlos.
- Contener un catálogo accesible por los usuarios en el que se almacenen las descripciones de los datos de forma centralizada.
- Garantizar que las operaciones que se realizan sobre la base de datos se desarrollen completamente, caso contrario no se realiza ninguna modificación.
- Debe permitir que los usuarios pueden acceder a la base de datos al mismo tiempo, actualizando los datos en tiempo real.
- Debe contar con un método de recuperación de la base de datos en caso de que ocurra algún fallo en el sistema o se dañe, recuperando totalmente los datos.
- Mantener el control de acceso solamente a las personas autorizadas, permitiendo definir entre diferentes tipos de niveles de acceso.
- Garantizar la integridad de la base de datos.
- El acceso a la base de datos debe contar con una disponibilidad continua.
- Proporcionar herramientas para la administración de la base de datos. Entre ellas deben estar las siguientes funcionalidades: importar y exportar datos, generación de reportes, monitoreo de su funcionamiento, reorganización de índices y la de optimizar el espacio libre para la reutilización.
- Debe contar con un programa gestor de comunicaciones, ya que varios usuarios ingresan a la base de datos mediante terminales remotas. Aunque no forme parte del gestor, debe permitir integrarse fácilmente a él.
- Debe aprovechar los recursos de la máquina donde se encuentra alojada, aumentando su capacidad de proceso.
- Poseer un lenguaje para la manipulación de datos, que permita la inserción, modificación, consulta y eliminación de información de la base de datos.
- Permitir el almacenamiento de grandes cantidades de información sin afectar el rendimiento de la máquina [3].

1.2.2. Arquitectura de los SGBD

De acuerdo con [16] en su investigación, afirma que en 1975, el comité ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Committee) propuso una arquitectura de tres niveles para los SGBD cuyo objetivo principal era el de separar los programas de aplicación de la base de datos física. En esta arquitectura el esquema de una base de datos se define en tres niveles de abstracción distintos:

- Nivel interno o físico: el más cercano al almacenamiento físico, es decir, tal y como están almacenados en el ordenador. Describe la estructura física de la base de datos mediante un esquema interno. Este esquema se especifica con un modelo físico y describe los detalles de cómo se almacenan físicamente los datos: los archivos que contienen la información, su organización, los métodos de acceso a los registros, los tipos de registros, la longitud, los campos que los componen, etcétera.
- Nivel externo o de visión: es el más cercano a los usuarios, es decir, es donde se describen varios esquemas externos o vistas de usuarios. Cada esquema describe la parte de la base de datos que interesa a un grupo de usuarios en este nivel se representa la visión individual de un usuario o de un grupo de usuarios.
- Nivel conceptual: describe la estructura de toda la base de datos para un grupo de usuarios mediante un esquema conceptual. Este esquema describe las entidades, atributos, relaciones, operaciones de los usuarios y restricciones, ocultando los detalles de las estructuras físicas de almacenamiento. Representa la información contenida en la base de datos.

1.2.3. Componentes

Los SGBD son programas que está compuesto con una serie de servicios que ayudan con la lectura y modificación de la información de forma eficiente que se encuentra almacenada en las bases de datos. Están compuestos principalmente por:

1.2.3.1. Lenguajes de los SGBD

Cada uno de los SGBD proporcionan servicios e interfaces diferentes, dependiendo del tipo de usuario: administradores, programadores de aplicaciones y usuarios. Dichos lenguajes permiten al administrador de la base de datos modificar la estructura, las relaciones que

existen en ella, las reglas de integridad, el control de acceso y las vistas externas de los usuarios. Los lenguajes del SGBD se clasifican en:

- **Lenguaje de definición de datos (LDD o DDL):** se utiliza para especificar el esquema de la base de datos, las vistas de los usuarios y las estructuras de almacenamiento. Es el que define el esquema conceptual y el esquema interno. Lo utilizan los diseñadores y los administradores de la base de datos.
- **Lenguaje de manipulación de datos (LMD o DML):** se utilizan para leer y actualizar los datos de la base de datos. Es el utilizado por los usuarios para realizar consultas, inserciones, eliminaciones y modificaciones. Los hay procedurales, en los que el usuario será normalmente un programador y especifica las operaciones de acceso a los datos llamando a los procedimientos necesarios. Estos lenguajes acceden a un registro y lo procesan. Las sentencias de un LMD procedural están embebidas en un lenguaje de alto nivel llamado anfitrión. Las bases de datos jerárquicas y en red utilizan estos LMD procedurales [16].

1.3. Seguridad

Debido a la cantidad de información que son almacenadas en las bases de datos, convierte a estas en uno de los principales objetivos por personas con intereses maliciosos, haciendo que centren gran parte del tiempo en encontrar vulnerabilidades que puedan ser aprovechadas para violar su seguridad, es por ello que se debe de tener en cuenta muchos aspectos al momento de aumentar la seguridad y evitar riesgos en la integridad [6].

Según lo detalla [16], un sistema gestor de base de datos proporciona mecanismos para garantizar la seguridad de los datos:

- Debe garantizar la protección de los datos contra accesos no autorizados, tanto intencionados como accidentales. Debe controlar que sólo los usuarios autorizados accedan a la base de datos.
- Los SGBD ofrecen mecanismos para implantar restricciones de integridad en la base de datos. Estas restricciones van a proteger la base de datos contra daños accidentales. Los valores de los datos que se almacenan deben satisfacer ciertos tipos de restricciones de consistencia y reglas de integridad, que especificará el administrador

de la base de datos. El SGBD puede determinar si se produce una violación de la restricción.

- Proporciona herramientas y mecanismos para la planificación y realización de copias de seguridad y restauración.
- Debe ser capaz de recuperar la base de datos llevándola a un estado consistente en caso de ocurrir algún suceso que la dañe.
- Debe asegurar el acceso concurrente y ofrecer mecanismos para conservar la consistencia de los datos en el caso de que varios usuarios actualicen la base de datos de forma concurrente.

1.3.1. Vulnerabilidades

A continuación, [17] detalla las vulnerabilidades más comunes que pueden encontrar a la hora de trabajar con bases de datos

- **Nombre de usuario/contraseña en blanco o bien hacer uso de uno débil. -** Hoy en día no es raro encontrar pares de datos usuario/contraseña del tipo admin/12345 o similar. Esta es la primera línea de defensa de entrada a la información y se debe optar por el uso de algo más complejo que sea complicado de conseguir por parte de cualquier atacante. A la hora de generar una contraseña para un usuario es recomendable usar tanto letras como números, así como de caracteres especiales tipo ¡, ¿, %... y con una longitud superior a 8 caracteres. De esta manera se tiene la seguridad de que la contraseña sea lo suficientemente fuerte para que no pueda ser adivinada por ningún proceso automático.
- **Preferencia de privilegios de usuario por privilegios de grupo. -** En ocasiones muchos usuarios reciben más privilegios sobre la base de datos de los que realmente necesitan, lo que a la larga se puede convertir en un importante problema. Es recomendable modificar los privilegios otorgados a los usuarios que estarán en contacto con la información con el fin de que no puedan realizar modificaciones más allá de las autorizadas. Si por ejemplo un usuario sólo realizará consultas a la base de datos, pero no podrá modificar ningún registro ni insertar nada nuevo, no tiene sentido ofrecer esos privilegios, ya que lo que se está haciendo es abrir una puerta para un eventual ataque.

- **Características de bases de datos innecesariamente habilitadas**. - Cada instalación de base de datos viene con una serie de paquetes o módulos adicionales de distintas formas y tamaños que en muy pocas ocasiones todos ellos son utilizadas por las compañías, lo que las convierten en una posible puerta de entrada para sufrir algún tipo de ataque si en esos paquetes se descubre cualquier problema de seguridad. Para reducir riesgos, es recomendable que los usuarios detecten esos paquetes que no se utilizan y se desactiven del servidor donde estén instalados. Esto no sólo reduce los riesgos de ataques, sino que también simplifica la gestión de parches ya que únicamente será de máxima urgencia actualizar aquellos que hagan referencia a un módulo que se este utilizando.
- **Desbordamiento de búfer. -** Se trata de otro de los medios favoritos utilizados por los piratas y que se dan por el exceso de información que se puede llegar a enviar por medio del ingreso de información mediante el uso de formularios, es decir, se recibe mucha más información de lo que la aplicación espera. Por poner un ejemplo, si se espera la entrada de una cuenta bancaria que puede ocupar unos 25 caracteres y se permite la entrada de muchos más caracteres desde ese campo, se podría dar este problema.
- **Bases de datos sin actualizar. -** Como ocurre con cualquier tipo de aplicación que se tiene instalada en un computador, es necesario ir actualizando la versión de la base de datos con las últimas versiones lanzadas al mercado, ya que en ellas se solucionan aquellos problemas de seguridad detectados. Para ello es muy importante estar informados de todas las noticias relacionadas con la base de datos que se está utilizando.
- **Datos sensibles sin cifrar**. – La encriptación de datos importante en la base de datos ayuda a dificultar la tarea en caso de que una persona ajena acceda a ella sin permiso y desee visualizar la información. Debido a que no se encuentra legible y le imposibilita la lectura de la información. Un ejemplo de ello seria, las contraseñas de acceso a un sitio por parte de los usuarios podrían ser cifradas utilizando el algoritmo MD5. De esta forma una contraseña del tipo “YUghd73j%” en base de datos se almacenaría con el siguiente valor “93e5b4451e0216d8e089c4”. Como se puede aprecia, se trata de un valor que poco o nada tiene que ver con el original.

1.3.2. Tipos de ataques

1.3.2.1. Ataques Externos

1.3.2.1.1. Inyección SQL

Consiste en la inserción de código SQL por medio de los datos de entrada desde la parte del cliente hacia la aplicación. Es decir, por medio de la inserción de este código el atacante puede modificar las consultar originales que debe realizar la aplicación y ejecutar otras totalmente distintas con la intención de acceder a la herramienta, obtener información de alguna de las tablas o borrar los datos almacenados, entre otras muchas cosas. Como consecuencias de estos ataques y dependiendo de los privilegios que tenga el usuario de la base de datos bajo el que se ejecutan las consultas, se podría acceder no sólo a las tablas relacionadas con la aplicación, sino también a otras tablas pertenecientes a otras bases de datos alojadas en ese mismo servidor [18].

1.3.2.1.2. Fuerza Bruta

Un ataque de fuerza bruta refiere a la forma de recuperar una contraseña probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso. Una de las desventajas de este ataque es que puede tener un costo elevado ya que utilizan un método de prueba y error el cual puede llevar días semanas incluso meses en poder descifrar una clave para tomar el control y son muy costosos en tiempo computacional [19].

1.3.2.1.3. Ataque de Diccionario

Este ataque se caracteriza por el uso de palabras escritas en el diccionario como fuente para descifrar la contraseña. Este tipo de ataque es más optimo que el de fuerza bruta, ya que un gran porcentaje de personas utilizan palabras de su lengua, siendo una forma más fácil de recordar la clave.

Los ataques de diccionario tienen pocas probabilidades de éxito con sistemas que emplean contraseñas fuertes con letras en mayúsculas y minúsculas mezcladas con números (alfanuméricos) y con cualquier otro tipo de símbolos. Sin embargo, para la mayoría de los usuarios recordar contraseñas tan complejas resulta complicado. Existen variantes que comprueban también algunas de las típicas sustituciones (determinadas letras por números, intercambio de dos letras, abreviaciones), así como distintas combinaciones de mayúsculas y minúsculas [19].

1.3.2.2. Ataques Internos

Estos tipos de ataques se realizan cuando una persona que trabaja dentro de la empresa realiza modificaciones a la base de datos.

1.3.2.2.1. Usuario administrador que modifica datos

Un usuario con privilegios de administrador tiene la capacidad de modificar absolutamente toda la información almacenada en la base de datos.

1.3.2.2.2. Usuario de la empresa que modifica datos

Un usuario trabajador de la empresa tendrá acceso a la información almacenada, dependiendo de los privilegios que tenga y haya sido dados por el administrador, estará en la capacidad de modificar los registros almacenada en ella.

1.3.3. Principios Básicos de Seguridad

Según [17] junto con [1] detallan una serie de recomendaciones para a tener en cuenta para fortalecer la seguridad de la base de datos.

1.3.3.1. Identifique su sensibilidad

Es necesario que el administrador o encargado de la base de datos realice simulaciones de ataques para así detectar las posibles zonas menos protegidas o lugares por donde podrían atacar y así vulnerabilidad los protocolos de seguridad. Siendo estas realizadas por el mismo administrador o un especialista en Hacking Ético.

1.3.3.2. Evaluación de la vulnerabilidad y la configuración

Evaluar la configuración de la base de datos es otra forma de prevenir ataques y no dejar huecos en su seguridad. Esto comprende también la verificación de cómo se instaló la base de datos y el sistema operativo en donde está alojada.

Así como también mantener actualizada la versión de la base de datos ya que a medida que surgen vulnerabilidades, la empresa se encarga de corregirlas en las nuevas versiones que son lanzadas al mercado.

Para ello el administrador también deberá de poner algunas reglas como:

- No permitir consultas desde aplicaciones a la base de datos.
- Limitar el acceso a la base de datos a personas que no lo requieran.
- Limitar el acceso a procedimientos para ciertas personas.

1.3.3.3. Endurecimiento

Al concluir la revisión del sistema de seguridad surge posibles mejoras. Este es el comienzo para el endurecimiento de la base de datos. Es necesario la aplicación de un estricto protocolo de lo que es necesario mejorar y en que no es necesario la implementación de más seguridad.

1.3.3.4. Auditoria

La aplicación de una auditoría permite el monitoreo de los accesos a la base de datos, esto permite determinar:

- Quien ingresa a la base de datos.
- Cuando se ingresa a la base de datos.
- Desde dónde y cómo se accede a la base de datos.
- Las consultas que se ejecuten.
- Alguna modificación que se realice en la base de datos.

Mediante la implementación de la auditoría en la base de datos se podrá:

- Monitorizar y registrar el uso de los datos por los usuarios.
- Generar alertas en tiempo real.

1.3.3.5. Monitoreo

Con el monitoreo a tiempo real de la base de datos va a permitir la detección de intrusos o de alguna modificación que se esté haciendo y tener un registro de ella. Gracias al monitoreo se puede detectar la presencia de ataques con Inyección SQL, cambio en las tablas o modificación en los privilegios de las cuentas de usuarios.

1.3.3.6. Autenticación y control de acceso

No todas las personas pueden realizar inserciones, modificaciones, consultas o eliminar información de la base de datos por lo que se debe de administrar los privilegios para limitar

las operaciones que se realicen, ratificando de manera periódica los privilegios, esto como adición al método de auditoría.

1.4. Bases Legales

Las bases legales de investigación se encuentran representadas en: La Constitución de la República del Ecuador, Ley Orgánica de Transparencia y Acceso a la Información Pública, Ley Orgánica de Servicio Público del Ecuador y el Código Orgánico Integral Penal de la República del Ecuador.

La Ley Orgánica de Transparencia y Acceso a la Información Pública del Ecuador, en su Artículo 17, define como reservada a toda información que tenga relación con la defensa nacional, equipamiento militar, informes de inteligencia y fondos para la seguridad nacional.

Como se especifica en el Art. 18. el tiempo de vigencia en el cual la información se considera reservada y bajo qué parámetros puede ser desclasificada. A la información reservada de acuerdo a la ley, se le fija un plazo máximo de reserva de quince años desde su clasificación, pudiendo desclasificarse antes cuando la situación lo amerite o ampliarse el plazo si se justifican las causas que la originaron, lo que es de competencia del Consejo de Seguridad Nacional, organismo que también tiene facultades para clasificar y desclasificar la información. [20]

En consecuencia, la única información que se puede clasificar y desclasificar es la información reservada y solamente está facultado para ello el Consejo de Seguridad Nacional.

La información confidencial, por referirse a derechos personalísimos relacionados con la intimidad personal y se contempla en el Art. 6 de la Ley Orgánica de Transparencia y Acceso a la Información Pública. [20]

Donde respecto a la información confidencial, que se refiere a los derechos personalísimos, no existe organismo alguno que clasifique o desclasifique ésta, porque pertenece a la vida privada e intimidad de las personas, en las cuales ninguna autoridad tiene facultades para interferir, pero si se contiene dentro de determinada base de datos se podría divulgar ilícitamente.

La divulgación de información reservada será considerada como espionaje y sancionada de acuerdo en lo estipulado en el Art. 354. del Código Orgánico Integral Penal.

Es interesante lo que se detalla en el Art. 229 emitido en el Código Orgánico Integral Penal, con relación a la divulgación ilegal de base de dato. La persona que, en provecho propio releve información registrada en fichero, archivos, base de datos o medios semejantes será sancionada con pena privativa de libertad de uno a tres años, en el caso de tratarse de un funcionario público, empleadas o empleados bancarios internos o de instituciones de la economía popular y solidaria que realicen intermediación financiera o contratistas, será sancionada con pena privativa de libertad de tres a cinco años. [21]

La o el servidor público que tenga acceso a una base de datos tiene obligación de mantenerlas en reserva, lo que se contiene en el inciso final del Art. 22 de la Ley Orgánica de Servicio Público, que señala, dentro de los deberes de las y los servidores públicos "Custodiar y cuidar la documentación e información que, por razón de su empleo, cargo o comisión tenga bajo su responsabilidad el impedir o evitar su uso indebido, sustracción, ocultamiento o inutilización". [22]

CAPÍTULO II: METODOLOGÍA

2.1. Descripción de la investigación

Se recreo un ambiente virtual donde se instalaron sistemas gestores de bases de datos. Para esta investigación se usarán gestores de: MySQL, SQL Server, PostgreSQL, con el fin de realizar una serie de ataques para medir el nivel de seguridad existente.

2.2. Tipo de investigación

En función del objetivo

- Investigación teórica. - Ya que se realizó distintas pruebas de vulnerabilidad, por lo que fue necesario la recopilación de información sobre los distintos tipos de prueba juntos con su aplicación en los SGBD compatibles.
- Investigación aplicada. - Una vez recolectada la información necesaria, se ejecutaron las pruebas necesarias para burlar la seguridad en los SGBD, y de esa manera poder realizar un análisis sobre la eficacia de sus mecanismos de seguridad.

En base al nivel de profundización

- Explicativa. – Una vez obtenidos los resultados, mediante un análisis profundo de cada uno de ellos, se consiguió dar una explicación al ¿Por qué? de cada uno de los resultados.
- Descriptiva. – Después de obtener la información necesaria de cada uno de los gestores de base de datos, se realizó el respectivo análisis individual del material. También para realizar las pruebas en los ambientes propuesto se necesitó de información para el montaje del entorno de prueba y evitar el surgimiento de errores al momento de su instalación.

En base a los datos que se utilizan

- Cualitativa. – Con la información recopilada se extrajo cada una de las características principales de los SGBD, las cuales lo definen como un programa seguro de utilizar.

Según el grado de manipulación de variables

- Experimental. - Ya que se concluyó a partir de los resultados obtenidos de las pruebas de vulnerabilidad que se realizaron en los diferentes gestores de base de datos.

2.3. Métodos y técnicas

Los métodos y técnicas que se utilizó en la siguiente investigación fueron:

- Método Deductivo: Se realizó una investigación deductiva partiendo de los conceptos generales de seguridad y vulnerabilidades de los sistemas gestores de base de datos, que se analizaron posteriormente cada uno de estos de manera individual y se aplicaron los ataques.
- Método Inductivo: Ya que, a partir de los resultados obtenidos en cada una de las pruebas a los gestores de base de datos, se pudo llegar a una conclusión sobre el nivel de seguridad de cada uno de ellos.
- Técnica Observación: Se utilizó esta técnica ya que mediante la observación del resultado se pretendió entender a simple vista si el SGBD fue capaz de soportar los ataques o si fue posible burlar su seguridad y acceder a la información que es almacenada dentro de él.
- Técnica Experimental: Debido a que se recreó un entorno en donde se instalaron cada uno de los SGBD en los cuales se ejecutaron los comandos necesarios para implementar cada uno de los mecanismos de seguridad.

2.4. Población y muestra (técnicas de muestreo)

Dado que las pruebas serán realizadas por el investigador y los resultados serán obtenidos por experiencia propia, es prescindible la ayuda de una población y por ende la realización de una técnica de muestreo.

2.5. Descripción del instrumento

Lista de Chequeo: Una vez revisada la información bibliográfica se procedió a la creación de una lista de chequeo (ANEXO 1) donde se detallaron los temas claves para implementar la seguridad, agrupándolo en 4 grupos, además de ello a medida que se realizaban las pruebas se fue llenando la lista partiendo de los resultados obtenidos, una vez lleno todos los datos se pudo concluir y clasificar a los diferentes gestores.

2.6. Técnicas de procesamiento y análisis

Para el procesamiento y análisis, una vez obtenidos los resultados de las pruebas se procedió a la creación de un cuadro comparativo donde se enlistaron los SGBD y los distintos tipos de ataques realizados, resaltando el resultado del ataque para así poder determinar qué tan efectivo fue al momento de evitar la fuga de información.

2.7. Normas éticas

Es de suma importancia mantener las normas éticas necesarias una vez finalizada la investigación ya que lo que se pretendió fue medir la seguridad de los SGBD y diferentes medios para burlar su seguridad, mediante los resultados obtenidos se pudo saber qué tipo de pruebas son eficientes al momento de ingresar y extraer información de manera ilícita de una base de datos, ya sea de paga (Microsoft SQL Server) o gratis (PostgreSQL y MySQL), por lo que la información obtenida al finalizar esta investigación puede ser utilizada tanto como para fines de fortalecer la seguridad o para fines ilícitos.

CAPÍTULO III: RESULTADOS

3.1 CONSTRUCCIÓN ENTORNO VIRTUAL

La descarga de las bases de datos de MySQL, PostgreSQL y SQL Server, se ingresó a las páginas oficiales de cada una de ellas las cuales cuentan con toda la información necesario acerca de las versiones y el instalador de cada una ya sea para 32 y 64 bits, posteriormente dimos clic en la pestaña de descargas y seleccionamos la versión con la cual se planteó trabajar, la instalación se lo realizó en una máquina marca Dell Inspiron 13 500 Series, la cual trabaja con un sistema operativo de Windows 10 (x64) desarrollado por Microsoft y con un procesador Intel Core i7.

En la instalación de MySQL se descargó la versión 8.0.12. liberada en Junio del 2018, en el caso de PostgreSQL se instaló en su versión 10.4 lanzado en Mayo del 2018 y para SQL Server en su versión 14.0.1. habilitada en Octubre del 2017. Se configuraron los puertos para cada una de ellas siendo: 3306 para MySQL, 5432 en PostgreSQL y 1433 en SQL Server, además se utilizaron herramientas GUI (Graphical User Interface) para la interacción con las bases de datos que fueron SqlYog en su versión 11.11. para MySQL, PgAdmin 4 en el caso de PostgreSQL y SQL Server Managment Studio 17.8.1 para SQL Server.

Una vez finalizada la instalación de los distintos sistemas gestores de base de datos, se continuó con la ejecución de los comandos para la configuración de los mecanismos de seguridad planteados con anterioridad.

Para lo cual se crearon 3 tablas en cada uno de los SGBD con las que se realizaron las pruebas, las cuales fueron: usuarios, productos y audit_productos.

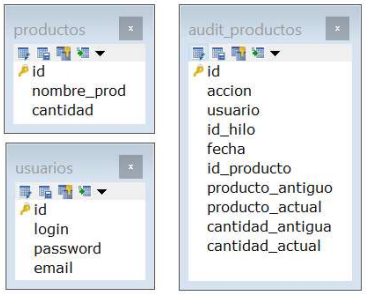

Figura 6: Diagrama de tablas.

3.2. FORTALEZAS Y DEBILIDADES

3.2.1. Gestión de usuarios y roles

La implementación de gestión de usuarios y roles resulta muy útil cuando existen una gran cantidad de personas que por diferentes motivos necesitan acceder a una base de datos para realizar diferentes acciones sobre ella, por lo que para mantener un control de acceso a ella se crean credenciales para cada uno de ellos. En lo que respecta a la gestión de usuarios, los tres SGBD se desenvuelven de manera muy similar siendo la mayor diferencia en que SQL Server debe crear un inicio de sesión (Login) para asignarle a un usuario y este debe ser creado en la base de datos a la cual desea acceder ya que los usuarios de una base de datos no constan en las otras. Aunque al igual que en las demás se le da mucha seguridad al momento de crear usuarios de manera que su contraseña se mantiene oculta (PostgreSQL *, SQL Server ●) o en otros casos encriptándola (MySQL Sha1), añadiendo cada nuevo usuario sin ningún tipo de permisos para la modificación, inserción o eliminación de archivos en la base de datos, permitiendo acceder solo a la información que lo involucre. (Ver figura 7)

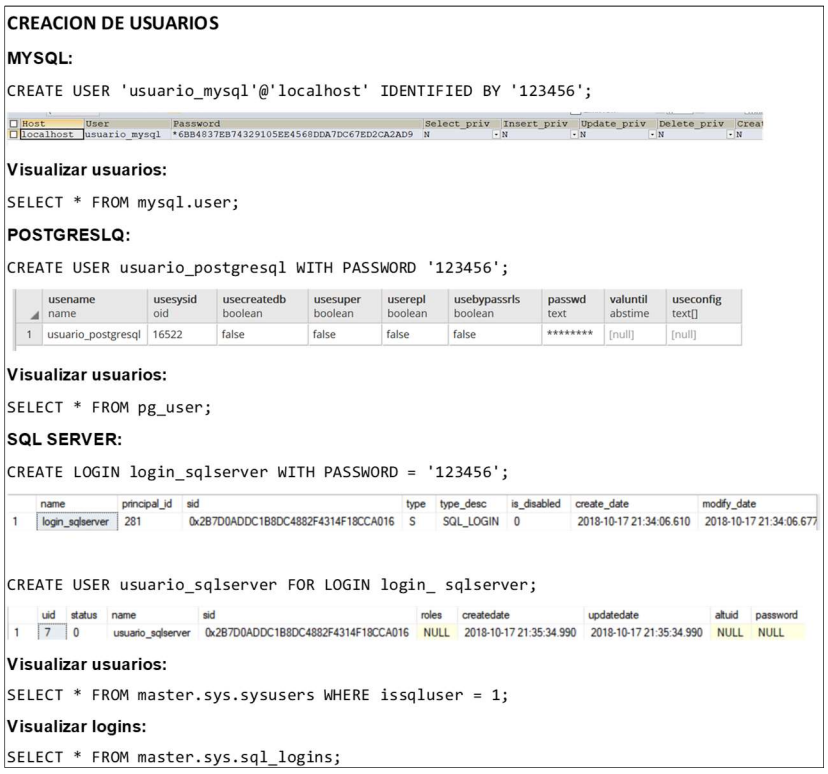

Figura 7: Creación de usuarios.

De igual manera al momento de eliminar usuarios, en algunos casos (MySQL, SQL Server) es posible hacerlo, aunque estos cuenten aun con algún tipo de permisos, en contrario a las políticas de PostgreSQL que no es posible eliminar usuarios sin antes revocar los permisos que tengan sobre todo tipo de objeto en la base de datos.

Una de las características que resultó más interesante fue que en PostgreSQL es posible crear grupos de usuarios, característica que no comparte con MySQL y SQL Server, cosa que resulta muy útil si lo que se desea es otorgar permisos a un gran número de usuarios sin la necesidad de hacer de forma individual en repetidas ocasiones. (Ver figura 8)

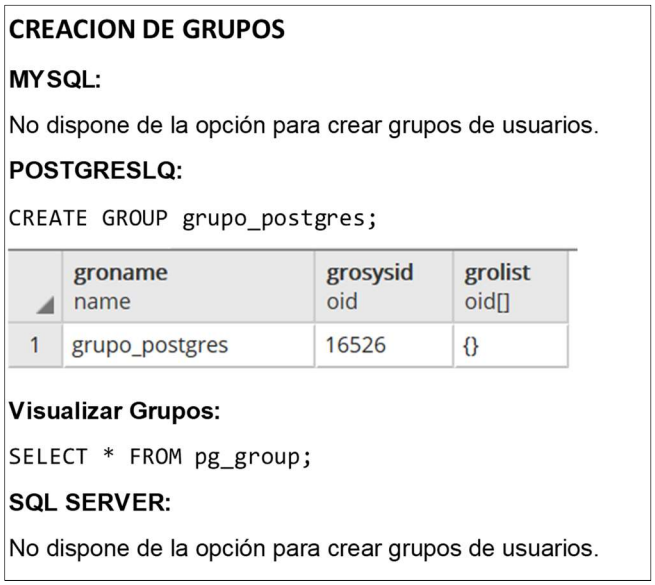

Figura 8: Creación de roles.

En lo que respecta a la gestión de roles, estos al momento de su creación son agregados sin ningún tipo de permisos y en el caso de SQL Server, es necesario especificar en cual base de datos desea crear el rol ya que no comparte roles entre las bases de datos. Un inconveniente que surgió al momento de asignar un rol a un usuario o grupo fue que, en el caso de MySQL, el usuario no heredo los permisos que tenía el rol, solo funcionando los permisos que tenía el usuario asignado de manera individual. De igual manera cuando se eliminan los roles es necesario revocar los permisos que le fueron otorgados en el caso de PostgreSQL.

3.2.2. Gestión de privilegios

Los privilegios que se otorgan a los diferentes usuarios, roles o grupos, dan la facultad de realizar diferentes acciones sobre distintos tipos de objetos en la base de datos por lo cual es necesario tener la debida precaución al momento de conceder permisos debido a que estos

influyen de gran manera al momento de realzar o degradar la seguridad, una gran diferencia que existe entre PostgreSQL y SQL Server con MySQL, es el que MySQL permite el uso de permisos globales y a nivel de base de datos, los mismos que otorgan permisos para realizar todo tipo de acción ya sea en todas las bases de datos de un servidor como en una respectivamente. Vistos desde diferentes puntos de visto estos permisos pueden resultar beneficioso o perjudicial. (Ver figura 9)

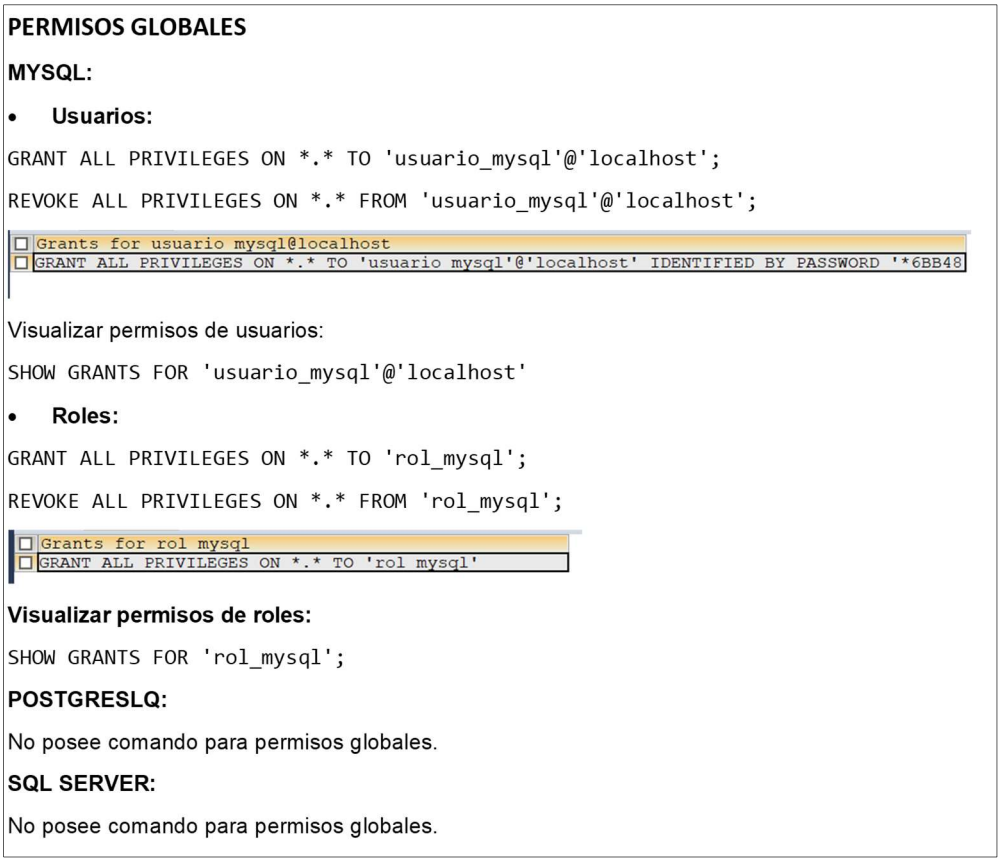

Figura 9: Permisos globales.

En cuanto a los permisos a nivel de tablas y columnas, las 3 base de datos funciona de manera similar con excepción del permiso “ALL” en SQL Server, devolviendo un error (ALL es obsoleto y solo se mantiene con fines de compatibilidad. NO implica los permisos ALL definidos en la entidad.) aunque en la página oficial de SQL Server todavía se encuentra vigente este comando, debido a este inconveniente es necesario conceder los permisos que sean necesarios de manera unitaria. (Ver figura 10)

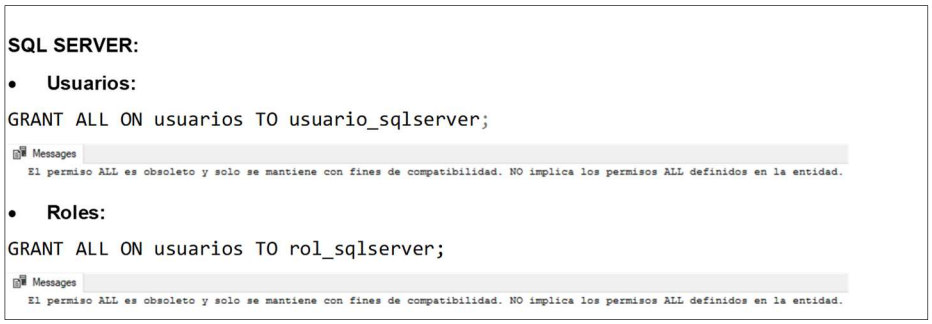

Figura 10: Permisos a nivel de tablas SQL Server.

3.2.3. Métodos de encriptación

Una forma de asegurar que la información almacenada en la base de datos no sea revelada es usando métodos de encriptación para proteger la información sensible que dado el caso de fuga de información, esta no pueda ser legible para el usuario que la sustrae, dado que existe una amplia gama de algoritmos de encriptado de datos, se han utilizado únicamente 3 (MD5, SHA, AES) aunque como lo descrito anteriormente, en las páginas oficiales de cada base de datos se encuentran todos los métodos de encriptación posibles. El método MD5 es un algoritmo estándar y al ejecutarlo en los distintos SGBD lo reconocen sin ningún tipo de inconveniente, caso que no sucede de igual manera con SHA y AES, donde para utilizar SHA en PostgreSQL es necesario habilitar la extensión “pgcripto” que nos dará acceso a este método de encriptado y AES en SQL Server se debe seguir una serie de pasos para la creación de claves y certificados. (Ver figura 11)

METODOS DE ENCRIPTACION

MYSQL:

- **MD5**

```
UPDATE usuarios SET PASSWORD=MD5('123456')
WHERE id = 1;
```

id	password
1	e10adc3949ba59abbe56e057f20f883e

- **SHA**

```
UPDATE usuarios SET PASSWORD=SHA('123456')
WHERE id = 1;
```

id	password
1	7c4a8d09ca3762af61e59520943dc26494f8941b

- **AES**

```
UPDATE usuarios SET PASSWORD=AES_ENCRYPT('123456','key')
WHERE id = 1;
```

id	password
1	»–§ü”¯–ÆEö•¦UÌâ▯

Visualizar registros:

```
SELECT id, password from usuarios where id=1;
```

POSTGRESLQ:

- **MD5**

```
UPDATE usuarios SET password=MD5('123')
where id = 1;
```

	id integer	password character varying (200)
1	1	202cb962ac59075b964b07152d234b70

- **SHA**

```
UPDATE usuarios SET password=ENCODE(DIGEST('123456', 'sha1'), 'hex')
where id = 1;
```

	id integer	password character varying (200)
1	1	7c4a8d09ca3762af61e59520943dc26494f8941b

- **AES**

```
update usuarios set password=ENCRYPT('123456', 'key', 'aes')
where id = 1;
```

	id integer	password character varying (200)
1	1	\273\226\247\374\224\257-\306E\366\225\246U\314\342\031

Visualizar registros:

```
select id, password from usuarios where id=1;
```

SQL SERVER:

- **MD5**

```
UPDATE usuarios SET PASSWORD=HashBytes('MD5', '12345')
WHERE id = 1;
```

	id	PASSWORD
1	1	.|E▯êŠpIL4¡h˙øN{

- **SHA**

```
UPDATE usuarios SET PASSWORD=HashBytes('SHA', '12345')
WHERE id = 1;
```

	id	PASSWORD
1	1	Œ²#}▯yÊˆÛddêÆ ©cEQ9d

Visualizar registros:

```
select id, password from usuarios where id=1;
```

Figura 11: Métodos de encriptación.

3.2.4. Creación de trigger

El trigger o disparador es utilizado a menudo para realizar acciones sobre las tablas de una base de datos de manera automáticas, se activan cuando se ejecutan unas sentencias especificas (INSERT, UPDATE, DELETE), pudiéndole dar varias utilidades siendo una de estas para fortalecer la seguridad mediante trigger que registren las acciones que se realizan en una tabla. (Ver figura 12, 13 y 14)

MySQL:

```
CREATE TRIGGER audit_productos_update
BEFORE UPDATE ON inyeccion.`productos`
FOR EACH ROW
  INSERT INTO inyeccion.`audit_productos`(accion, usuario, id_hilo, fecha, id_producto, producto_antiguo,
  producto_actual, cantidad_antigua, cantidad_actual)
      VALUES ('UPDATE', CURRENT_USER(), CONNECTION_ID(), NOW(), NEW.id, OLD.nombre_prod, NEW.nombre_prod, OLD.cantidad, NEW.cantidad);

CREATE TRIGGER audit_productos_insert
BEFORE INSERT ON inyeccion.`productos`
FOR EACH ROW
  INSERT INTO inyeccion.`audit_productos`(accion, usuario, id_hilo, fecha, id_producto, producto_antiguo,
  producto_actual, cantidad_antigua, cantidad_actual)
      VALUES ('INSERT', CURRENT_USER(), CONNECTION_ID(), NOW(), NEW.id, NULL , NEW.nombre_prod, NULL, NEW.cantidad);

CREATE TRIGGER audit_productos_delete
BEFORE DELETE ON inyeccion.`productos`
FOR EACH ROW
  INSERT INTO inyeccion.`audit_productos`(accion, usuario, id_hilo, fecha, id_producto, producto_antiguo,
  producto_actual, cantidad_antigua, cantidad_actual)
      VALUES ('DELETE', CURRENT_USER(), CONNECTION_ID(), NOW(), OLD.id, OLD.nombre_prod , NULL, OLD.cantidad, NULL);
```

Figura 12: Trigger MySQL.

PostgreSQL:

```
CREATE OR REPLACE FUNCTION auditoria_productos() RETURNS trigger AS
$BODY$
BEGIN
  IF (TG_OP = 'DELETE') THEN
    INSERT INTO audit_productos (accion, usuario, id_hilo , fecha, id_producto , producto_antiguo, producto_actual, cantidad_antigua, cantidad_actual)
           VALUES ('Delete', USER, 1, NOW(), OLD.id, OLD.nombre_prod, '', OLD.cantidad, '');
    RETURN OLD;
  ELSIF (TG_OP = 'UPDATE') THEN
    INSERT INTO audit_productos (accion, usuario, id_hilo , fecha, id_producto , producto_antiguo, producto_actual, cantidad_antigua, cantidad_actual)
           VALUES ('Update', USER, 1, NOW(), NEW.id, OLD.nombre_prod, NEW.nombre_prod, OLD.cantidad, NEW.cantidad);
    RETURN NEW;
  ELSIF (TG_OP = 'INSERT') THEN
    INSERT INTO audit_productos (accion, usuario, id_hilo , fecha, id_producto , producto_antiguo, producto_actual, cantidad_antigua, cantidad_actual)
           VALUES ('Insert', USER, 1, NOW(), NEW.id, '', NEW.nombre_prod, '', NEW.cantidad);
    RETURN NEW;
  END IF;
  RETURN NULL;
END;
$BODY$
LANGUAGE 'plpgsql';

CREATE TRIGGER tbl_atributos_tg_audit AFTER INSERT OR UPDATE OR DELETE
ON productos FOR EACH ROW EXECUTE PROCEDURE auditoria_productos();
```

Figura 13: Trigger PostgreSQL

SQL Server:

```
CREATE TRIGGER dbo.AuditProductosInsert ON productos FOR INSERT
AS
BEGIN INSERT dbo.audit_productos SELECT 'INSERT', SUSER_NAME (), GETDATE(), id, null, nombre_prod,
null, cantidad FROM INSERTED
END

CREATE TRIGGER dbo.AuditProductosDelete ON productos FOR DELETE
AS
BEGIN INSERT dbo.audit_productos SELECT 'DELETE', SUSER_NAME (), GETDATE(), id,  nombre_prod, null,
cantidad, null FROM DELETED
END

CREATE TRIGGER dbo.AuditProductosUpdate ON productos FOR Update
AS
DECLARE
@oldName varchar(20),
@oldValue int

SELECT @oldName = nombre_prod, @oldValue = cantidad FROM DELETED

BEGIN INSERT dbo.audit_productos SELECT 'UPDATE', SUSER_NAME (), GETDATE(), id, @oldName, nombre_prod,
@oldValue, cantidad FROM INSERTED
END
```

Figura 14: Trigger SQL Server.

Este mecanismo de seguridad funciona de manera similar en las 3 base de datos en prueba, obteniendo resultados muy parejos al momento de analizar para el registro de acciones, en cambio para el análisis de seguridad surge una diferencia entre MySQL y PostgreSQL con SQL Server, debido a que en los primeros dos es necesario otorgar permisos sobre la tabla donde se van a registrar los acciones que se realizan en las demás tablas, en contraste a SQL Server donde no es necesario el conceder algún permiso, siendo esto una desventaja ya que el usuario que realice modificaciones va a tener permiso a la tabla donde se registran las acciones.

Dado los resultados obtenidos con cada uno de los mecanismos de seguridad utilizados, los SGBD que lograron una mayor cantidad de ventajas fueron PostgreSQL y SQL Server, debido a que PostgreSQL cuenta con las funcionalidades de creación de grupos y no permite eliminar usuarios o roles sin antes haber eliminados los permisos, en tanto que SQL Server mantiene la configuración de los mecanismo de seguridad de manera independiente de cada una de las bases de datos, evitando que la configuración de una base de datos no pueda ser utilizada en otra. A continuación, se presenta la lista de chequeo junto con los resultados obtenidos. (Ver figura 15)

METODOS DE SEGURIDAD	BASES DE DATOS		
1. GESTION DE USUARIOS Y ROLES	**MYSQL**	**POSTGRESQL**	**SQL SERVER**
1.1. Creación de usuarios	X	X	X
1.2. Creación de grupos		X	
1.3. Creación de roles	X	X	X
1.4. Asignación de usuario a grupos		X	
1.5. Asignación de roles a usuarios	X	X	X
2. GESTION DE PRIVILEGIOS			
2.1. Permisos globales	X		
2.2. Permisos a nivel de Base de Datos	X	X	-
2.3. Permisos a nivel de Tablas	X	X	X
2.4. Permisos a nivel de Columnas	X	X	X
2.5. Permisos específicos	X	X	X
3. METODOS ENCRIPTACION			
3.1. MD5	X	X	X
3.2. SHA	X	-	X
3.3. AES	X	X	-
4. CREACION DE TRIGGER			
4.1. Creación de disparadores para auditoria	X	X	X

Figura 15: Lista de chequeo con resultado de pruebas.

CAPÍTULO IV: DISCUSIÓN

Esta investigación tuvo como objetivo analizar las vulnerabilidades en los gestores de base de datos, tomando como protagonistas a MySQL, PostgreSQL y SQL Server, donde se les realizaron una serie de paso con el fin de fortalecer la seguridad que trae consigo predeterminada al momento de su instalación. Como lo detalla [6] en su investigación, una de las causas de la aparición de vulnerabilidades en los SGBD son la falta de configuración por parte de los administrador dado que para poder ser el encargado de un SGBD es necesario tener un vasto conocimiento conforme a los mecanismos de seguridad que se deben implementar para alcanzar un nivel de seguridad alto en sus sistemas. Esto concuerda con los resultados obtenidos, puesto que fue necesario la investigación más a profundidad sobre los métodos de seguridad para los SGBD debido a que la configuración que viene por defecto al momento de su instalación no es lo suficientemente robusta si lo que se pretende es proteger los datos almacenados en ellos.

Si bien la investigación de [5] afirma que "la seguridad absoluta no existe" también nos incita a utilizar los SGBD que contengan la mayor cantidad de métodos de seguridad y dado el caso de aquellos en que sus actualizaciones corrijan las vulnerabilidades a medida que van surgiendo, teniendo mucho sentido ya que [1] afirma que "vulnerar la seguridad en SGBD comerciales son uno de los objetivos de los criminales", dado que optar por utilizar un sistema que se encuentra a la vanguardia en cuanto a los mecanismos y actualizaciones de seguridad, nos va a otorgar una cierta ventaja frente a los criminales que intenten sustraer información de nuestro sistema.

Cabe resaltar que uno de los puntos más interesantes que se encontró fue que para evitar la sustracción o la filtración de información a personas no deseadas es recomendable la buena utilización de los privilegios de acceso con los que cuentan los SGBD, manteniendo la precaución con el nivel y tipo de acceso que otorgamos a los usuarios, concordando con lo que dice [8] donde detalla que "el conceder permisos excesivos o no revocar los privilegios a medida que transcurre el tiempo puede desencadenar en una gran vulnerabilidad en la seguridad " ya que no es necesario conceder permiso más de los que necesita un usuario ya que esto puede desencadenar en una posible vulnerabilidad a nuestra información.

La implementación de mecanismo de seguridad al momento de iniciar con el uso de alguno de los SGBD no basta si lo que se busca en mantener un nivel de seguridad alto, sino que es de vital importancia darle mantenimiento y rectificando la configuración con el fin de cerciorarse que nada haya sido modificado y de evitar configuraciones o permisos de acceso a la información obsoleto, si bien la implementación de privilegios a los usuarios es un método de mantener a la información segura, la supervisión de estos complementa el método utilizado, tal y como lo afirma [7] "la supervisión de los usuarios privilegiados, es requisito para la gobernabilidad de datos y cumplimiento de regulaciones como SOX (Ley Sarbanes-Oxley) y regulaciones de privacidad", esto complementa la detección de intrusos no deseados en nuestro sistema y evitar posibles vulnerabilidades.

Si bien la investigación de [9] establece que es necesario desarrollar una política de seguridad acorde que permita al personal encargado la correcta toma de decisiones en la configuración y administración de este servidor en caso de presentarse un incidente, siendo totalmente valida esta técnica de seguridad, debido dado el caso resulta útil el poder implementar políticas de seguridad donde se establezcan todas las precauciones y contra medidas en caso de un intento o sustracción de información de los SGBD.

CAPÍTULO V: CONCLUSIONES

- Una vez revisada la información disponible sobre los distintos SGBD juntos con artículos científicos y libros respecto a lo que seguridad se trata, se pudo evidenciar que la bibliografía pone énfasis en la utilización de control de acceso por usuarios, asignación de permisos y encriptación de información sensible, si lo que se desea es mantener cierto grado de confidencialidad e integridad de la información almacenada en los sistemas.
- Es indispensable conocer la documentación necesarios sobre los distintos gestores de base de datos, ya que, dependiendo de las necesidades, algunos de ellos cuentan con distintas versiones de sus sistemas.
- La ejecución de las pruebas con el fin de sustraer, visualizar y modificar la información, arrojaron diferentes resultados muy marcados entre antes y después de implementar los mecanismos, ya que una vez implementados, se pudo mermar algunas vulnerabilidades que existían referente a las personas que tenían acceso a la información.
- Mediante la aplicación de gestión de usuarios y privilegios, se pudo restringir el acceso a la base de datos a todos los usuarios, pudiendo solamente acceder los designados por el DBA, creando de esta manera un control a los datos en los SGBD.
- Se pudo apreciar que cada SGBD de datos responde de manera muy similar, realizando las acciones para las cuales fueron diseñados cada uno de los comandos ingresados, siendo la diferencia más notable que MySQL y SQL Server al contrario de PostgreSQL, no cuenta con la acción de crear de grupos de usuarios, lo cual resulta útil si lo que se pretende es agrupar varios usuarios y asignarle permisos, evitando de esta manera el realizarlo de manera unitaria.
- Los SGBD que obtuvieron mejores resultados y de los cuales se pudieron apreciar mayor nivel de seguridad, fueron en el caso de PostgreSQL y SQL Server, teniendo PostgreSQL la posibilidad de configurar los mecanismos establecidos de manera que engloba todas las bases de datos que tiene, en cambio SQL Server mantiene separadas las configuraciones de cada uno de sus bases de datos, de tal manera que no es posible utilizar los permisos creados en una base de datos para las demás.

CAPÍTULO VI: RECOMENDACIONES

Para otorgar al SGBD un nivel confiable de seguridad es recomendable una revisión de la documentación disponible en lo referente a cada uno de los sistemas que se van a utilizar y en específico lo correspondiente a seguridad, no es recomendable dejar la configuración predeterminara de manera que viene al momento de su instalación, ya que esta no cuenta con los mecanismos necesarios para proteger la información.

La revisión de la configuración de usuarios y privilegios de manera periódica es un complemento de seguridad que sirve para ratificar si la configuración no ha sido modificada sin el consentimiento del administrador.

Dado que las empresas dueñas de los SGBD se esfuerzan continuamente por dar un producto de calidad, estas constantemente liberan para los usuarios los denominados parches de seguridad que a su vez se encargan de resolver los fallos en la seguridad, por lo cual es recomendable utilizar un sistema que se encuentre en continua mejora.

La persona encargada en su administración tendrá que actualizarse sobre los nuevos métodos que van surgiendo para fortalecer su seguridad y de los que ya se encuentran disponibles para su uso. Los Sistemas Gestores de Base de Datos (SGBD) administrarán la información ingresada en su Base de Datos, por lo que virtualizar la información conlleva a un ahorro de espacio físico y monetario para la empresa.

REFERENCIAS

[1] J. Villalobos, A. Guevara, A. Reyes, E. De Leon Guerrero, G. Cruz, and J. Borbon, "Hacktivismo y DDoS : Tendencias actuales de ataque," vol. 12, 2012.

[2] O. Pérez Mora, C. Martín Escofet, M. Gibert Ginestà, D. Costal Costa, L. A. Casillas Santillán, and R. C. Paré, *Bases de datos*. 2005.

[3] J. Pérez, "Las bases de datos, su seguridad y auditoría. el caso de MySQL," 2011.

[4] A. M. Rubinos Carvajal, H. Alina, and N. León, "Seguridad en bases de datos Security Database," *Rev. Cuba. Ciencias Informáticas*, vol. 5, no. Sistema de bases de datos, p. 16, 2011.

[5] P. López Herrera, "Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL. Tesis para obtener el título de Ingeniera en Computación.," p. 72, 2016.

[6] I. Armendariz and D. Cuesta, "Analisis de los principales sistemas de gestion de base de datos ante ataques basicos," 2016.

[7] J. Domínguez Chávez, "Principios Básicos de Seguridad en Bases de Datos," 2015.

[8] M. Malik and T. Patel, "DATABASE SECURITY-ATTACKS AND CONTROL METHODS," *Int. J. Inf. Sci. Tech.*, vol. 6, no. 1, 2016.

[9] M. J. García, "Database Main Threats Analisys Using MS SQL Server."

[10] A. Cobo Yera, *Diseño y programación de bases de datos*. Visión Libros, 2007.

[11] M. G. Piattini Velthuis and E. del. Peso Navarro, *Auditoría informática : un enfoque práctico*. Ra-Ma, 1997.

[12] A. Moreno, *DISEÑO E IMPLEMENTACIÓN DE UN LEXICÓN COMPUTACIONAL PARA LEXICOGRAFÍA Y TRADUCCIÓN AUTOMÁTICA*. Universidad Autónoma de Barcelona, 1999.

[13] Universidad Murcia, "Sistemas de Gestión de Bases de datos y SIG," *Sist. Gestión Base Datos Y Sig*, vol. 9, no. 3, pp. 167–180, 2013.

[14] C. J. Date, *An introduction to database systems*. Addison-Wesley, 1990.

[15] D. María Del Carmen and G. Fuentes, *Bases De Datos*. 2013.

[16] Mc Graw Hill, "Sistemas gestores de bases de datos," pp. 1–15, 2016.

[17] Telefonica, "Bases de datos y sus vulnerabilidades más comunes," *AcensTechnologies*, 2010.

[18] G. Cardenal, "Ataques de inyección SQL," *HostaliaWhitepapers*, vol. 1, pp. 48008–902, 2013.

[19] H. D. H. Juárez, "Hackeo mediante Ataque de fuerza Bruta Ataque de fuerza bruta," *Dinámica Comun. Integr.*, no. Hackeo mediante Ataque de fuerza Bruta, 2016.

[20] "LEY ORGANICA DE TRANSPARENCIA Y ACCESO A LA INFORMACION PUBLICA DEL ECUADOR."

[21] "CÓDIGO ORGÁNICO INTEGRAL PENAL DEL ECUADOR."

[22] "LEY ORGANICA DE SERVICIO PUBLICO DEL ECUADOR."

ANEXOS

ANEXO 1: Lista de Chequeo

METODOS DE SEGURIDAD	BASES DE DATOS		
1. GESTION DE USUARIOS Y ROLES	**MYSQL**	**POSTGRESQL**	**SQL SERVER**
1.1. Creación de usuarios			
1.2. Creación de grupos			
1.3. Creación de roles			
1.4. Asignación de usuario a grupos			
1.5. Asignación de roles a usuarios			
2. GESTION DE PRIVILEGIOS			
2.1. Permisos globales			
2.2. Permisos a nivel de Base de Datos			
2.3. Permisos a nivel de Tablas			
2.4. Permisos a nivel de Columnas			
2.5. Permisos específicos			
3. METODOS ENCRIPTACION			
3.1. MD5			
3.2. SHA			
3.3. AES			
4. CREACION DE TRIGGER			
4.1. Creación de disparadores para auditoria			