Universidad de las Ciencias Informáticas

Facultad 10

Título: Indicadores que determinan la seguridad de las bases de datos realizadas en PostgreSQL en cuanto a la configuración, diseño, arquitectura y codificación.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores: Niurisleidy Reyes Piloto
Raúl Cambar Martínez

Tutores: Ing. Yusleydi Fernández del Monte
Ing. Sonia Guerrero Lambert
MsC. Michael González Jorrín

Mayo de 2010

"El mundo camina hacia la era electrónica... todo indica que esta ciencia se constituirá en algo así como una medida de desarrollo: quien la domine será un país de vanguardia. Vamos a volcar nuestros esfuerzos en este sentido con audacia revolucionaria."

Ernesto Che Guevara

Marzo 1962

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo:" Indicadores que determinan la seguridad de las bases de datos realizadas en PostgreSQL en cuanto a la configuración, diseño, arquitectura y codificación" y autorizamos a la facultad 10 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autores:

_____

Niurisleidy Reyes Piloto

_____

Raúl Cambar Martínez

Tutor:

_____

Ing. Yusleydi Fernández del Monte

_____

Ing. Sonia Guerrero Lambert

_____

MsC. Michael González Jorrín

AGRADECIMIENTO

Niurisleidy:

Quiero agradecer a todas esas personas que me ayudaron a lo largo de mi paso por la universidad:

A mis padres por darme confianza en que podía graduarme.

Raúl Cambar por estar incondicionalmente a mi lado y quererme mucho.

Esteban Joel Navarro por ser mi mejor amigo.

Aliuska Calunga por apoyarme en muchas ocasiones.

Yanisleidy Cañete (la china) por ayudarme en una parte de la investigación de mi tesis.

Prof. Raymundo Chacón por ser el único profesor que se preocupó por mi salud.

Juan Carlos por aconsejarme cuando mi mundo se venía abajo.

A mis tutoras Yusleydi y Sonia por ayudarme en el desarrollo de mi tesis.

Y especialmente a Neris Marcilli Ruíz al cual agradeceré eternamente el haberme ayudado a realizar mi sueño de ser Ingeniera Informática.

Gracias a todos.

Raúl:

Agradezco a mis padres por todo el apoyo que me han dado no solo aquí sino durante toda mi vida, a todos mis compañeros que han compartido conmigo lo bueno y lo malo de la UCI, Aliuskita y Juan siempre al lado de nosotros, a los que me ayudaron sobre todo en la última etapa de la tesis a Yoemir que no puso reparos y nos tendió la mano siempre. A mi familia y vecinos que son como familia siempre atentos a mí. Las tutoras Yusleydi y Sonia, son un dolor de cabeza; a Joel. A mis suegros que han corrido conmigo como con su hija y me han ayudado tanto.

A todos gracias.

DEDICATORIA

Niurisleidy:

Estudiar en una universidad a más de 600km del hogar durante 5 años es un gran sacrificio, pero cuando se obtiene el triunfo de graduarse uno se siente realizado.

Todos esos años de sacrificio y dedicación, dieron paso a mi tesis, la cual quiero dedicar a 3 personas fundamentales que me forjaron durante toda mi vida:

Mi mamá, mi modelón, mi estrella que desde que salí de su vientre ha sido y será la mejor madre del mundo.

Mi papá, ejemplo a seguir, el más preocupado de todos los padres y el más querido por su hija.

Mi abuelita, mi abu, que sin ella no sé que me haría.

Gracias a los 3 por su cariño, su apoyo, dedicación y por sobre todas las cosas

Gracias por existir.

Raúl:

Dedico este trabajo a mis padres Inés y Raúl, que siempre han estado batallando para que yo llegara a donde llegué, gracias por estar siempre conmigo ustedes son mi inspiración, son especiales: se la dedico a mi hermanita, sigue luchando, que esto te sirva de inspiración, a mis abuelos y sus lágrimas: a esa personita especial: mi novia, amiga y compañera, siempre guiándome para que no fracase; por quererme tanto, son ustedes lo más especial de mi vida.

RESUMEN

Las bases de datos constituyen un pilar en el desarrollo de las sociedades. La Universidad de las Ciencias Informáticas, enmarcada en el desarrollo revolucionario es puntera en la construcción y utilización de bases de datos. En el proceso de construcción de bases de datos se trata la seguridad de manera superficial, en muchos casos por desconocimiento. Para dar solución a la problemática planteada, la investigación tiene como objetivo diseñar una Guía de indicadores que permita elevar la seguridad en bases de datos realizadas en PostgreSQL. Para el desarrollo del trabajo se utilizó la entrevista para identificar los principales problemas. Como resultado de la investigación se obtuvo una Guía de indicadores que permite a los desarrolladores de bases de datos en PostgreSQL obtener bases de datos con alto grado de seguridad, en la misma se definen una serie de indicadores y buenas prácticas para conducir al desarrollador por los procesos de desarrollo de una base de datos: arquitectura, diseño, configuración y codificación; para esto se investigó el estado del arte de los conceptos de seguridad y el progreso de las bases de datos del gestor de PostgreSQL. Para conocer el desempeño de la Guía se realizó una predicción a través del método Delphy, demostrando que es una útil solución para elevar la seguridad en bases de datos desarrolladas en PostgreSQL y que es aplicable a cualquier proyecto que utilice este gestor.

PALABRAS CLAVE:

Bases de datos, seguridad.

ÍNDICE

INTRODUCCIÓN

El mundo actual está regido por el conocimiento; este cada día genera más volúmenes de datos de ahí que las bases de datos hayan logrado ser un pilar para la gestión y almacenamiento de información.

A nivel mundial las bases de datos son componentes esenciales de cualquier sistema económico o social, no existe un renglón que para su desarrollo no utilice las bases de datos, las cuales han cambiado y modernizado la forma de gestionar la información y el conocimiento en el mismo nivel tecnológico de otros softwares. Nunca antes fue tan necesario el establecer un mecanismo para almacenar la información como lo es ahora y en este aspecto han surgido a nivel mundial una diversidad de gestores de bases de datos con características específicas dando paso a la elección de uno u otro según las necesidades y políticas de la organización.

Dentro del proceso de desarrollo socio-económico y de cambios en el país se hace indispensable la informatización social, esto conlleva a la aplicación exhaustiva de bases de datos con gestores adaptables a las necesidades existentes.

Cientos de bits de datos son generados, almacenados y destruidos en fracciones de segundos; dentro de este volumen existe información vital para las organizaciones y la sociedad; y su seguridad es de orden primordial. Para cualquier empresa la seguridad en sus bases de datos es ineludible, a nivel mundial y nacional, la Universidad de las Ciencias informáticas (UCI) no queda exenta de esto.

Las bases de datos son creadas mediante un gestor que sirve de interfaz entre el usuario y la base de datos. Las mismas se diseñan bajo tres aspectos, diseño conceptual, físico y lógico, quedando estructurado el contenido de la base de datos, relaciones lógicas y su implementación en memoria. Dichas bases de datos responden a una arquitectura, basada en algún estándar en específico, como puede ser el de la ANSI/SPARC de tres niveles. El mecanismo de acceso a estos datos se realiza a través de la codificación, en muchos casos el lenguaje utilizado como estándar es el SQL. La configuración es el último proceso y no menos importante, ya que aquí se definen las características operativas de las bases de datos.

Entre los proyectos productivos que se llevan a cabo en la Universidad de las Ciencias Informáticas está el Centro de Desarrollo de Tecnología de Datos (DATECD), en el cual usando el gestor PostgreSQL se desarrollan bases de datos para las cuales es imperioso mantener la seguridad. Se han detectado en DATECD una serie de problemas en la creación de bases de datos seguras que

atentan contra otros parámetros de calidad. Se conoce que no existe un mecanismo idóneo para medir el nivel de seguridad en bases de datos desarrolladas en PostgreSQL, no existen indicadores que permitan conocer los fallos de seguridad en estas bases de datos y hay un pobre conocimiento sobre buenas prácticas en los aspectos de configuración, diseño, arquitectura y codificación de las bases de datos en PostgreSQL. Aunque tienen conocimiento de que la seguridad afecta otros factores de calidad no saben en qué medida ni de qué forma se afectan las bases de datos por una elevada o baja seguridad, no son capaces de detectar de qué manera se manifiestan los errores. Entre los problemas más comunes que la falta de seguridad acarrea se han encontrado:

- ✓ Sentencias de SQL que entran a los datos sin pasar por la validación web: esto permite que los individuos logren acceder a la información, modificarla o copiarla, siendo esta en muchos casos información crítica.
- ✓ Acceso a datos críticos con cuentas que tienen más permiso del que necesitan: lo que trae como consecuencia que se borren o modifiquen datos o ficheros por usuarios que tienen demasiados permisos y sin necesidad de tenerlos.
- ✓ No hay conocimiento de medidas a tomar para elevar la seguridad lo que provoca que la seguridad se trate de manera mínima y superficial lo que conlleva a que otros factores de calidad como el rendimiento se vean afectados.

Por lo tanto, analizando la problemática anteriormente expresada se plantea resolver el siguiente **problema científico** ¿Cómo elevar la seguridad en bases de datos realizadas en PostgreSQL en cuanto a configuración, diseño, arquitectura y codificación?

El objeto de estudio que encierra el problema a tratar es: los procesos de obtención de seguridad en bases de datos, se tiene como **campo de acción**: **l**os procesos de obtención de seguridad en bases de datos realizadas en PostgreSQL.

El Objetivo de la investigación es diseñar una Guía de indicadores que permita elevar la seguridad en bases de datos realizadas en PostgreSQL.

Se ha identificado una serie de pasos para cumplir con el objetivo de la investigación como hilo conductor para la resolución satisfactoria del problema anteriormente planteado:

- ✓ Investigar las bases de datos realizadas en PostgreSQL para obtener conocimientos de la forma de creación de las mismas y su modo de funcionamiento, así como las variantes

existentes para configurar, diseñar, realizar la arquitectura y codificación en bases de datos teniendo en cuenta lo establecido en PostgreSQL.

- ✓ Investigar sobre indicadores que dicen hasta qué punto las bases de datos realizadas en PostgreSQL son seguras en cuanto a configuración, diseño, arquitectura y codificación para conocer las principales tendencias y antecedentes del tema.
- ✓ Investigar los factores de calidad entre los que se encuentra la seguridad, la relación que tienen y su influencia en la calidad para determinar el balance que debe tener la seguridad con otros factores de calidad para mantener esta última.
- ✓ Diseñar una guía compuesta por un sistema de indicadores que permita elevar la seguridad en bases de datos realizadas en PostgreSQL.
- ✓ Aplicar el método Delphy para pronosticar posibles resultados de la utilización de la Guía.

Se define como **idea a defender:**

La utilización de una Guía de indicadores de seguridad puede elevar la seguridad en las bases de datos realizadas en PostgreSQL

Para llevar a cabo la investigación los **métodos científicos** utilizados son:

Métodos teóricos:

Método histórico-lógico:

Se utiliza para comprender la seguridad como factor de la calidad, su evolución histórica hasta la seguridad informática analizando planteamientos de autores con conocimientos en el tema; las tendencias, evolución, utilización y funcionamiento buscando entender las bases de datos desarrolladas utilizando el gestor PostgreSQL.

Método Inductivo-Deductivo: Dada la predicción de un panel de expertos se deduce que la Guía de indicadores se puede ser aplicable en todas las bases de datos que se creen en PostgreSQL.

Analítico – Sintético: Se utiliza este método para descomponer el tema de Seguridad en sistemas informáticos, en sus partes y elementos, de modo que permita analizar su contenido más detalladamente. Además se establecieron relaciones entre los elementos que componen las bases de datos desarrolladas en PostgreSQL y seguridad informática, escogiéndose los elementos que más se adaptan a la problemática presente y que son más utilizados a nivel mundial y de mayor difusión.

Métodos empíricos:

La entrevista: Se realizaron entrevistas a líderes y desarrolladores del proyecto DATECD, específicamente al Jefe de la línea de Configuración y al Administrador de Bases de Datos de la Gestión de la Configuración, proporcionando conocimiento sobre los problemas que existen en las bases de datos realizadas en PostgreSQL, y su opinión sobre la utilidad de la Guía de indicadores.

La encuesta: Se realizan encuestas a personas que trabajan directamente en la elaboración de estas bases de datos. Con el objetivo de conocer hasta qué punto tienen conocimientos de seguridad en bases de datos realizadas en Postgre y si aplican medidas de seguridad en estas bases de datos.

Método estadístico:

Delphy:

El método Delphy se utiliza para recoger una serie de criterios de expertos en el campo de seguridad en bases de datos en cuanto a la propuesta de solucion para su pronóstico de utilidad; a través de este método se posibilita conocer la utilidad de la propuesta de solución.

Población: Aplicaciones realizadas en PostgreSQL.

Muestra: Una selección de aplicaciones realizadas en PostgreSQL del proyecto DATECD.

Muestreo: El muestreo es no probabilístico y se realiza de manera accidental según las condiciones.

El presente trabajo de diploma se compone de los siguientes apartados:

Capítulo 1 “Estado del arte de la seguridad de las bases de datos realizadas en PostgreSQL”: se expone la fundamentación teórica sobre la que se sustenta la investigación, analizando elementos teóricos como conceptos y tendencia de seguridad en bases de datos realizadas en PostgreSQL.

Capítulo 2 “Guía de indicadores para elevar la seguridad en bases de datos realizadas en PostgreSQL”: se presenta la Guía de indicadores para elevar la seguridad en bases de datos realizadas en PostgreSQL.

Capítulo 3 “Pronóstico sobre los resultados de la Guía de indicadores”: se muestran resultados estadísticos que indican hasta qué punto se eleva la seguridad con la Guía de indicadores.

CAPITULO I
Estado del arte de la seguridad de las bases de datos realizadas en PostgreSQL.

Introducción:

La seguridad como factor de calidad ha evolucionado a lo largo de la historia y su concepto varía en dependencia de la aplicación a la que se haga referencia; en este capítulo se presenta de forma histórica el desarrollo del término seguridad; su aplicación como seguridad informática y su relación con la disponibilidad, la integridad y la confidencialidad según algunos autores; sus conceptos y su aplicación. Se expone la tendencia de los gestores de bases de datos, en particular el progreso del gestor PostgreSQL y las mejoras de seguridad en sus diferentes versiones; la relación de la seguridad de PostgreSQL con otros factores de calidad que están presentes en cualquier base de datos.

1.1-Conceptos principales.

1.1.1-Seguridad.

El término seguridad proviene de la palabra “securitas” del latín. Cotidianamente se puede referir a la seguridad como la ausencia de riesgo o también a la confianza en algo o alguien. El concepto de riesgo está íntimamente relacionado al de incertidumbre, o falta de certeza, de que algo pueda acontecer y generar una pérdida del mismo.

El diccionario de la Real Academia Española define a la palabra seguridad como, unos mecanismos de control que evitan el uso no autorizado de recursos.

Según Henry Fayol en 1919 identifica la Seguridad como una de las funciones empresariales, luego de la técnica, comercial, financiera, contable y directiva. [1]

Al definir el objetivo de la Seguridad Fayol dice:

> *"...salvaguardar propiedades y personas contra el robo, fuego, inundación contrarrestar huelgas y felonías, y de forma amplia todos los disturbios sociales que puedan poner en peligro el progreso e incluso la vida del negocio. Es, generalmente hablando, todas las medidas para conferir la requerida paz y tranquilidad (Peace of Mind) al personal"*. [1]

Aunque lejos de la sofisticación actual, no hay duda de que Fayol puede ser considerado el pionero de los más recientes conceptos de seguridad.

Sin embargo, el término de seguridad puede tomar diversos sentidos según el área a la que haga referencia.

El Dr. Giovanni Manunta, Consultor y profesor de seguridad de Cranfield University plantea en su libro Seguridad Corporativa y Protección del Patrimonio, que el concepto de seguridad no debe ser definido de forma general, pues depende del contexto en que se enmarque. Giovanni expresa*:" El concepto de seguridad es multidimensional, significando cosas diferentes a diferentes personas en diferentes contextos. Debido a su complejidad, tanto la explicación como la atribución de responsabilidades y la medición de las actividades son frecuentemente poco fiables, o al menos discutibles."*. [1]

La existencia de riesgos es un factor que está presente en todo sistema de seguridad y está estrechamente relacionado con las amenazas y vulnerabilidades del sistema en cuestión.

Según el estudio y análisis realizado, los autores de este trabajo de diploma abogan que la seguridad es un proceso en el cual se elimina el riesgo y se salvaguardan las propiedades o activos de valor.

1.1.1.1-Riesgos.

De acuerdo a Ulrich Beck[1], a comienzos de la era industrial el concepto de riesgos permite fundamentalmente una forma de calcular consecuencias imprevistas, desarrollando formas y métodos de hacer previsible lo imprevisto. En la sociedad actual el concepto de riesgo es más expandido y estrechamente relacionado con las incertezas. [2]

Según el diccionario de la Real Academia Española "riesgo" no es más que: contingencia o proximidad de un daño. El riesgo debe tener cierto grado de incertidumbre y está estrechamente relacionado con el impacto o consecuencia.

Existen varias técnicas de manejos de riesgo, los mismos son:

Evitar. Es lograr cambios significativos para mejorar algo; de forma tal que se impida cualquier suceso desagradable en los procesos, siendo el resultado de adecuados controles y acciones realizadas.

[1] Alemán nacido en 1944, Sociólogo, filósofo, psicólogo. Estudió en la Universidad de Friburgo y Munich.

Reducir. Cuando el riesgo no puede evitarse por tener varias dificultades de tipo operacional, la alternativa puede ser su reducción hasta el nivel más bajo posible. Esta opción es la más económica y sencilla y se consigue optimizando los procedimientos y con la implementación de controles.

Retener. Cuando se reducen los riegos, se podría retener los riesgos residuales. Dentro de los planes del manejo de riegos de la empresa se debe plantear cómo manejar dichos riegos si ocurriesen.

Transferir. Es buscar un respaldo y compartir el riesgo con otros controles o entidades. Esta técnica se usa ya sea para eliminar un riego de un lugar y transferirlo a otro, o para minimizar el mismo, compartiéndolo con otras entidades.

1.1.1.2-Amenaza.

La **amenaza** representa el tipo de acción que tiende a ser dañina para un activo o el sistema, es un evento, individuo o entidad que representa un riesgo para el sistema. Es cualquier acción o evento que puede ocasionar consecuencias adversas, en su totalidad son conocidas las posibilidades de ocurrencias.

1.1.1.3-Vulnerabilidad.

La **vulnerabilidad** (conocida a veces como *falencias (flaws)* o *brechas (breaches)*) representa el grado de exposición a las amenazas en un contexto particular. Puede llamarse también debilidad y hace referencia a la parte débil o fisura del sistema. Muchos autores concuerdan en que una vulnerabilidad es la deficiencia que puede ser explotada por una amenaza, y en conjunto constituyen un riesgo.

1.1.2-Seguridad Informática.

La seguridad Informática es un conjunto de métodos y herramientas destinados a proteger los activos informáticos de una institución. Los **activos informáticos** son recursos del sistema de información o relacionado con éste, necesarios para que la organización funcione correctamente y alcance los objetivos propuestos. El término seguridad informática está estrechamente relacionado con tres aspectos fundamentales de cualquier sistema de computación:

Confidencialidad: es la propiedad de la información, garantiza que a esta información solo acceda el personal autorizado. La confidencialidad es una de las piedras angulares de la seguridad de la información.

Integridad: se refiere a la cualidad de íntegro. Aquello íntegro es algo que no carece de ninguna de sus partes. En la informática la integridad referencial es una propiedad deseable en las bases de datos que

garantiza que una entidad (fila o registro) siempre esté relacionada con otras entidades válidas, es decir, la integridad referencial garantiza que una fila se relaciona únicamente y solo con otra fila específica de otra tabla. La integridad referencial supone que todos los datos sean correctos, sin repeticiones, que no se pierdan y que no existan entre ellos relaciones sueltas, que no existan modificaciones sin consentimiento o alguna clase de alteración de los datos

Disponibilidad: los activos informáticos son accedidos por las personas autorizadas en el momento requerido. [3]

Entre los principales activos de un sistema informático se encuentra la información, por lo que muchas veces se utiliza seguridad informática o de la información para referirse a la seguridad de un sistema informático.

El término seguridad informática es una generalización para un conjunto de tecnologías que ejecutan ciertas tareas relativas a la seguridad de los datos. ISO, en su norma 7498, define la seguridad informática como: una serie de mecanismos que minimizan la vulnerabilidad de bienes y recursos, donde un bien se define como algo de valor y la vulnerabilidad se define como la debilidad que se puede explotar para violar un sistema o la información que contiene. El bien más preciado por cualquier institución es la información y de ahí que se han desarrollado protocolos y mecanismos adecuados, para preservar su seguridad. Se puede hablar en este sentido de cinco conceptos principales de la seguridad de los sistemas: autentificación, autorización, auditoría, administración de perfiles y administración de conexiones. Basándonos en estos conceptos a la hora de implementar la seguridad se lograría cumplir con la confidencialidad, integridad y el no-repudio, aspectos fundamentales para cualquier sistema que gestione información. [4]

1.1.2.1- Conceptos principales de la seguridad de los sistemas:

Autentificación o autentificación: es el acto de establecimiento o confirmación de algo (o alguien) como auténtico, es decir que quien se autentifica sea quien realmente dice ser.

Autorización: Acto o documento a través del cual se permite a una persona realizar aquello que solicita, siempre y cuando cumpla con los requisitos exigidos.

Auditoría (informática): proceso de recoger, agrupar y evaluar evidencias para determinar si un Sistema de Información salvaguarda el activo empresarial, mantiene la integridad de los datos, lleva a cabo eficazmente los fines de la organización y utiliza eficientemente los recursos.

Administración de perfiles: Gestionar, asegurar y facilitar el contenido de los perfiles por parte del administrador del sistema.

Administración de conexiones: Gestionar, asegurar y facilitar las conexiones de forma sencilla, garantizando la seguridad de los datos.

Algunos estándares como la norma ISO 7498 incluye al no repudio como un principio de la seguridad; el no repudio **p**roporciona protección contra la interrupción, por parte de alguna de las entidades implicadas en la comunicación, de haber participado en toda o parte de la comunicación.

Definiendo la seguridad informática, se puede decir que es un sistema de herramientas y métodos para proteger los activos informáticos para que funcionen correctamente, hay que tener en cuenta varios aspectos fundamentales; los cuales son: la confidencialidad, integridad, disponibilidad y el no repudio.

1.1.3-Seguridad de la Información.

En muchos casos cuando se habla de la seguridad informática se tiene implícita la información que hace posible la existencia de los sistemas informáticos, es necesario aclarar que:

La información que está almacenada y procesada en computadoras, puede ser confidencial para algunas personas o a escala institucional; puede ser mal utilizada o divulgada, estar sujeta a robos, sabotaje o fraudes.

La norma UNE-ISO/IEC 17799 define la seguridad de la información como la preservación de su confidencialidad (sólo quienes estén autorizados pueden acceder a la información), su integridad (la información y sus métodos de proceso son exactos y completos), su disponibilidad (los usuarios autorizados tienen acceso a la información y a sus activos asociados cuando lo requieran).

La información es uno de los principales activos de las organizaciones, pudiendo llegar a poner en peligro su continuidad en caso de pérdida de la confidencialidad, integridad o disponibilidad de la información. Por este motivo, son necesarios unos sistemas de protección adecuados, así como una correcta gestión de la seguridad. [5]

1.2-Bases de datos.

1.2.1-Generalidades de las Bases de datos.

Hoy en día las bases de datos son componentes cardinales en cualquier aplicación o sistema informático, pudiendo almacenar un gran cúmulo de información. Debido a que la información considerablemente sensible o secreta es manejada por estos sistemas, se debe considerar seriamente proteger dicha información.

Una base de datos no es más que un conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. Una base de datos puede considerarse una colección de datos variables en el tiempo.

Las bases de datos han existido desde los comienzos de las civilizaciones y de hecho definen a las civilizaciones. Cuando el hombre necesita guardar conocimiento o seguir el rastro de la información, lo escribe, y lo cataloga usando índices de papel, así el libro fue el primer tipo de base de datos. Estos no eran bases de datos electrónicas, sin embargo servían para el mismo propósito. Eran usados para seguir el rastro de libros de contabilidad, conocimiento científico e histórico. Hoy cuando se piensa en bases de datos, lo primero que viene a la mente son bases de datos electrónicas, no en estos objetos que definieron a la civilización hace varios miles de años.

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego se pueda encontrar y utilizar fácilmente. [6]

1.2.2- Bases de datos relacionales.

Cuando la gente habla de Bases de Datos, regularmente se refieren a Bases de datos electrónicas más estructuradas tales como Relacionales, Objetos, OLAP (Procesamiento Analítico en Línea cuyo objetivo es agilizar la consulta de grandes cantidades de datos) o espaciales.

Las Bases de datos relacionales tienen sus orígenes en el año de 1970 cuando E.F. Codd de IBM introdujo la idea de un modelo relacional de Bases de datos en un documento titulado **“A Relational Model of data for Large Shared Banks”**, antes de eso la mayoría de las bases de datos estaban basadas en un modelo de red o una simple estructura de archivo plano. [7]

El modelo relacional está basado en una teoría de conjuntos matemáticos que servía para múltiples propósitos:

- Abstraer la representación de datos de su almacenaje físico y manipularlos.
- Minimizar la redundancia de datos, dividiéndolos en distintos grupos no duplicados que pueden ser relacionados en un infinito número de maneras para producir un infinito número de representaciones.
- Incrementar la consistencia de datos, por ejemplo si se cambia el nombre de un cliente, este cambiará en todos los reportes que se hagan acerca de ese cliente, porque esa parte es guardada en una sola parte, pero genera varias vistas o representaciones del dato.

Posteriormente un lenguaje llamado SQL (Lenguaje estructurado de consultas) también desarrollado por IBM, fue creado para generar reportes y actualizar datos en este nuevo modelo relacional.

Es entonces que nace el ***Sistema R***, pero fue ignorado por IBM, y poco después Oracle sacó su versión comercial de BD basada en la teoría relacional de Codd, y el Berkely Ingres.

Otros modelos relacionales de BD empezaron a brotar de estos modelos pioneros, ***Informix***, ***Sybase*** y el proyecto ***Ingres*** dieron nacimiento al ***Postgres,*** el cual consiste en agregar más características orientadas a Objetos al modelo relacional, después se transformó en PostgreSQL. El sistema R dió también nacimiento a ***DB2***.

1.3-PostgreSQL.

1.3.1-Surgimiento.

PostgreSQL es un sistema de gestión de bases de dato objeto-relacional, diseñado para administrar grandes cantidades de datos.

Los Sistemas de Gestión de Base de Datos (en inglés DataBase Management System) son un tipo de software específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

El proyecto PostgreSQL tal y como se conoce hoy en día empezó en 1996, aunque las bases y el trabajo en el que se asienta tienen sus comienzos en la década de los 70, los principales hitos del desarrollo del PostgreSQL se exponen a continuación.

- ✓ Ingres 1977-1985 - "El comienzo".

La década de los 70 fue una década de desarrollos y pruebas de nuevos conceptos en el nuevo mundo de los gestores de bases de datos.

IBM había estado trabajando desde 1973 con los primeros conceptos, ideas y teorías sobre bases de datos relacionales. Su proyecto "System R" fue entre otras cosas la primera implementación del lenguaje SQL. Este proyecto, sus decisiones de diseño y muchos de los algoritmos usados, influenciaron muchos de los sistemas de bases de datos relacionales que aparecieron posteriormente.

Por aquel entonces un profesor de la Universidad de Berkeley, Michael Stonebraker, leyó unos artículos publicados por IBM sobre "System R" que le hicieron interesarse en el tema. Utilizando el dinero de otro proyecto que ya tenía asignado, Ingres (INteractive Graphics REtrieval System), Stonebraker empezó a desarrollar sus ideas sobre bases de datos relacionales. Durante estos años Ingres mantuvo su código fuente abierto y permaneció en gran medida similar en conceptos a "System R".

A principio de los 80, Ingres estuvo compitiendo con Oracle por el liderazgo en el mundo de bases de datos relacionales y su código e implementación evolucionaron y fueron el origen de otras bases de datos relacionales, entre ellas podemos citar a Informix, NonStop SQL y Sybase (Microsoft SQL Server fue una versión licenciada de Sybase hasta su versión 6.0.

Michael Stonebraker dejó la Universidad de Berkeley en 1982 para comercializar Ingres pero volvió a la misma en 1985 con nuevas ideas.

- ✓ Postgres 1986-1994 - Después (post) de ingres.

Después de su vuelta a Berkeley en 1985, Michael Stonebraker lideró un nuevo proyecto llamado *Postgres* (después de Ingres) patrocinado por la *Defense Advanced Research Projects Agency (DARPA)*, la *Army Research Office (ARO)*, la *National Science Foundation (NSF)*, y *ESL, Inc*. Con este proyecto y basándose en la experiencia obtenida con Ingres, Stonebraker tenía como meta mejorar lo que habían conseguido y aprendido en el desarrollo de Ingres. Y aunque se basó en muchas ideas de Ingres, no se basó en el código fuente del mismo.

Los objetivos iniciales de este proyecto fueron:

- Proporcionar un mejor soporte para objetos complejos.
- Proporcionar a los usuarios la posibilidad de extender los tipos de datos, operadores y métodos de acceso.
- Proporcionar los mecanismos necesarios para crear bases de datos activas (triggers, etc.).
- Simplificar el código encargado de la recuperación del sistema después de una caída del mismo.
- Hacer cambios mínimos (preferiblemente ninguno) en el modelo relacional.
- Mejorar el lenguaje de consulta QUEL heredado de Ingres (POSTQUEL).

La última versión de Postgres en este proyecto fue la versión 4.2.

- ✓ Postgres95 1994-1995 - Nueva vida en el mundo open source.

En 1994, dos estudiantes de Berkeley, Andrew Yu y Jolly Chen, empezaron a trabajar con el código de Postgres (versión 4.2) y llamaron al proyecto Postgres95. Hicieron una limpieza general del código, arreglaron errores en el mismo, e implementaron otras mejoras, entre las que destacan:

- Sustitución de POSTQUEL por un intérprete del lenguaje SQL.
- Re-implementación de las funciones agregadas.
- PSQL fue creado para ejecutar consultas SQL.
- El interface de objetos grandes (large-object) fue revisado.
- Un pequeño tutorial sobre Postgres fue creado.
- Postgres se pudo empezar a compilar con GNU make y GCC sin parchear.

La versión 1.0 de Postgre95 vio la luz en 1995, el código era 100% ANSI C, un 25% más corto en relación con la versión 4.2 y un 30-50% más rápido. El código fue publicado en la web y liberado bajo una licencia BSD, más personas empezaron a utilizar y a colaborar en el proyecto.

- ✓ PostgreSQL 1996-actualidad - Proyecto PostgreSQL

En 1996, Andrew Yu y Jolly Chen ya no tenían tanto tiempo para dirigir y desarrollar Postgres95. Algunos de los usuarios habituales de las listas de correo del proyecto decidieron hacerse cargo del mismo y crearon el llamado *"PostgreSQL Global Development Team"*.

En un principio este equipo de desarrolladores a cargo de la organización del proyecto estuvo formado por Marc Fournier en Ontario, Canadá, Thomas Lockhart en Pasadena, California, Vadim Mikheev en Krasnoyarsk, Rusia y Bruce Momjian en Philadelphia, Pennsylvania. El nombre fue cambiado de Postgres95 a PostgreSQL y lanzaron la versión 6.0 en enero de 1997.

Inicialmente el Postgre provino de Ingres, pero a partir de esta etapa con la utilización del lenguaje SQL se decidió nombrarlo PostgreSQL, aunque para abreviar muchos autores y usuarios lo mencionan solamente como Postgre haciendo referencia al PostgreSQL. [8]

1.3.2- Actualidad de PostgreSQL.

Hoy en día el grupo central (core team) de desarrolladores está formado por 7 personas, existen 24 desarrolladores principales y más de 18 desarrolladores habituales. En total alrededor de 50 personas activas, contribuyendo con el desarrollo de PostgreSQL.

Existe también una gran comunidad de usuarios, programadores y administradores que colaboran activamente en numerosos aspectos y actividades relacionadas con el proyecto. Informes y soluciones de problemas, test, comprobación del funcionamiento, aportaciones de nuevas ideas, discusiones sobre características y problemas, documentación y fomento de PostgreSQL son solo algunas de las actividades que la comunidad de usuarios realiza.

No se puede olvidar que existen muchas empresas que también colaboran con dinero y/o con tiempo/personas en mejorar PostgreSQL. Muchos desarrolladores y nuevas características del PostgreSQL, están muchas veces patrocinadas por empresas privadas. Esto se debe a que el PostgreSQL es una excelente alternativa al software privativo en específico al ORACLE, resultando en un gestor de bases de datos realmente económico propiciando su extensión comercial por todo el mundo.

1.3.3-PostgreSQL en el mundo.

PostgreSQL es un poderoso sistema gestor de bases de datos, que tiene la fama de ser la base de datos de código abierto (Open Source) más avanzada del mundo, por estas y muchas otras características PostgreSQL se usa de una forma sorprendente.

En Japón, el uso del gestor PostgreSQL es amplio y existen muchas empresas importantes que usan esta base de datos en ese país.

En Norteamérica lo usan varias agencias gubernamentales como la fuerza armada y algunos proyectos de la Biblioteca del Congreso de los Estados Unidos. También son dignos de resaltar algunas iniciativas del Estado de California, de la Universidad de Oxford y del Laboratorio Nacional de Sandia (encargado de dar soluciones tecnológicas para resolver las amenazas nacionales y mundiales para la paz y la libertad).

En Latinoamérica son conocidos los casos de Loma Negra y Quilmas en Argentina, los casos de Entel (Empresa Nacional de Telecomunicaciones) y la Superintendencia de AFPs (Asociación de Administradoras de Fondos de Pensiones) en Chile y los casos de varias empresas de telecomunicaciones Brasileñas.

Pero el caso más paradigmático lo constituye el uso del PostgreSQL y otros productos Open Source por una de las corporaciones financieras más grandes del mundo: Deutsche Bank (Banco de inversión internacional líder). [9]

Se encuentran además, proyectos como:

- ✓ Armada Nacional de la República de Colombia[2].
  Todo funcionando con Linux+Apache+PHP+PostgreSQL
- ✓ SAPI - Servicio Autónomo de la Propiedad Intelectual, República Bolivariana de Venezuela.[3]
  Todo funcionando con Linux+Apache+Squid+Sendmail+PostgreSQL

[2] http://www.armada.mil.co

[3] http://www.cnti.ve/avances_sl4.html#

- ✓ Dirección Nacional de los Registros Nacionales de la Propiedad del Automotor y de Créditos Prendarios de la República Argentina[4]
  Funcionado replicación con PostgreSQL. [10]

1.3.4-PostgreSQL en Cuba

En Cuba existen más de 150 proyectos que utilizan PostgreSQL, entre ellos se encuentra el Centro de Desarrollo de Software de la UCI en Villa Clara, la Universidad Holguín, DATECD (Tecnologías de datos) centro que tiene como premisa desarrollar bases de datos utilizando PostgreSQL.

Además se encuentra la Comunidad Técnica Cubana de PostgreSQL que tiene como objetivos contribuir al desarrollo de tecnologías de bases de datos tomando como base el gestor PostgreSQL, proveer soluciones integrales y consultorías relacionadas con la migración y la explotación de PostgreSQL y contribuir a la formación de especialistas de alto nivel apoyando el desarrollo tecnológico cubano.

1.3.5-PostgreSQL en la Universidad de las Ciencias Informáticas (UCI).

En la Universidad de las Ciencias Informáticas (UCI) el uso de PostgreSQL es muy extendido, siendo ella la primera en usar este gestor en el país. En la universidad existen 16 proyectos productivos que trabajan con PostgreSQL de los muchos que se realizan hoy en la misma. Dentro de los proyectos que usan como gestor PostgreSQL se pueden mencionar:

1. Sistema de Gestión Fiscal (SGF).
2. Sistema Nacional Público para el Seguimiento de Inversiones y Sectores (SINAPSIS).
3. ERP Cuba (sus siglas en inglés Planificación de Recursos Empresariales).
4. Tribunales Populares Cubanos.
5. Filtro de contenido web (FCWEB) con su producto FILPACON.

Para ver la lista de proyectos que utilizan PostgreSQL en la UCI, ver anexo 1.

[4] http://rrii.sgp.gov.ar/jaiio/docs/postgresql_dnrpa.pps

El proyecto vanguardia en el uso del PostgreSQL es DATECD proyecto que con el objetivo de crear bases de datos en PostgreSQL, beneficia a entidades y ministerios tales como: MIC, MININT, MINFAR, MINBAS, MINSAP, MFP, ONE, Consejo de Estado, entre otros.

1.3.6-Características de PostgreSQL.

Está demostrado que PostgreSQL se ha expandido por todo el mundo y que cada día son más los usuarios que lo prefieren pero; ¿Por qué PostgreSQL? A continuación se presentan algunas características que hacen a PostgreSQL más atractivo para los usuarios globales. [11] La siguiente tabla muestras las características a partir de PostgreSQL 7.1.x.

Tabla 1: Características de PostgreSQL.

Características	Descripción
DBMS Objeto-Relacional	PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arrays.
Altamente Extensible	Soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario.
Soporte SQL Comprensivo	Soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
Integridad Referencial	PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
API Flexible	La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo con más facilidad para el RDBMS PostgreSQL. Estas interfaces incluyen Object

	Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.
Lenguajes Procedurales	PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.
Cliente/Servidor	PostgreSQL usa una arquitectura proceso-por-usuario y cliente/servidor. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectarse a PostgreSQL.
Costo	En realidad el único costo asociado a PostgreSQL es el de conocerlo ya que su código fuente está disponible bajo la más liberal de las licencias del Open Source: la licencia BSD. Bajo esta licencia tenemos la libertad de usar, modificar y distribuir Postgre, en productos comerciales o no comerciales, sin costo alguno.

1.3.6-Tipos de usuarios de PostgreSQL.

PostgreSQL posee niveles de usuarios para un mejor aseguramiento de los datos almacenados. En el momento en que el administrador de Postgre comienza a trabajar se crea el:

Superusuario de Postgres: es el usuario llamado Postgres que es dueño de los ficheros de la bases de datos y binarios de Postgres. Como superusuario de la base de datos, no le es aplicable ninguno de los mecanismos de protección y puede acceder a cualquiera de los datos de forma arbitraria. Además, al superusuario de Postgres se le permite ejecutar programas de soporte que generalmente no están disponibles para todos los usuarios. [12]

Luego se van creando los demás usuarios según el nivel que tengan de acceso a la información almacenada y a la base o bases de datos que se creen. Este sistema de creación de usuarios, dándole permisos según su nivel de acceso ayuda en gran medida no solo al aseguramiento de la información, sino que permite tener un mayor control de los entradas y modificaciones de cada usuario en la base de datos, permitiéndole así al administrador gestionar posibles errores que presente la base de datos.

1.3.7-Versiones de PostgreSQL vs Seguridad

A lo largo del desarrollo de PostgreSQL han surgido versiones que han revolucionado las características del gestor, dándole mayor seguridad. El 14 de Diciembre del 2009 se liberaron las versiones menores de todas las ramas activas del sistema gestor de bases de datos objeto-relacional PostgreSQL, incluyendo a las versiones 8.4.2, 8.3.9, 8.2.15, 8.1.19, 8.0.23 y 7.4.27. Estas nuevas versiones arreglan un problema de seguridad de "riesgo moderado" y uno de "bajo riesgo": un problema con la autenticación SSL, y un problema de escalado de privilegios con índices sobre expresiones. [12]

La versión 8.3 posee nuevas características en cuanto a la administración. La configuración GUC[5] por función permite a cada función tener configuración GUC propia, definida al momento de creación de la función. Es especialmente útil para definir la ruta de búsqueda de la función al tiempo de creación, lo cual mejora la seguridad. [13]

1.4 Seguridad en Bases de datos PostgreSQL.

La seguridad de base de datos consiste en las acciones que toma el diseñador de base de datos al momento de crear la base de datos, tomando en cuenta el volumen de las transacciones y las restricciones que tiene que especificar en el acceso a los datos; esto permitirá que el usuario adecuado sea quién visualice la información adecuada. [14] Existen básicamente cuatro niveles de seguridad para una base de datos fácilmente aplicables a una base de datos generada en PostgreSQL:

La seguridad de acceso: Se implementa de dos formas posibles, a nivel de sistema operativo, en cuyo caso el SGBD se apoya en la seguridad de entrada al sistema operativo para comprobar la validez del acceso a los datos almacenados; o bien lo que se llama modo mixto, en el cual la seguridad de entrada a la información la llevará a cabo el propio servidor de datos a partir de la definición de cuentas de usuario

[5] GUC Configuración Global de usuario por sus siglas en inglés.

La seguridad a nivel de objetos: Detalla el acceso a nivel de creación y administración de objetos de datos: tablas, vistas, índices, relaciones, reglas, etc. Es decir, las responsabilidades y acciones que puede hacer el usuario en el esquema de la base de datos.

La seguridad a nivel de datos: Se realiza en la capa de información, donde se indicará quién puede acceder a qué información para su consulta, actualización, inserción o borrado.

Seguridad a nivel de protección de los almacenamientos físicos de la información: Es tarea del sistema operativo, de los archivos de datos del sistema, y las políticas de copia de seguridad y restauración de los datos. [15]

1.4.1- Seguridad en el proceso de desarrollo de bases de datos desarrolladas en PostgreSQL.

Para elevar la seguridad en las bases de datos se debe analizar el proceso de desarrollo de estas, este se enmarca en cuatro partes fundamentales: la arquitectura, el diseño, la configuración y la codificación, esta investigación se basa fundamentalmente en estos cuatro procesos de construcción, donde la arquitectura por sus características engloba a los demás predefiniendo la arquitectura ANSI/SPARC por su amplia difusión y utilización en proyectos de la Universidad.

Para elevar la seguridad en el proceso de desarrollo de bases de datos se plantean una serie de indicadores, buenas prácticas y recomendaciones, los cuales se definen como:

Los **indicadores** son puntos de referencia conformados por uno o varios datos; en este caso buenas prácticas; opiniones o medidas que permiten conocer el desenvolvimiento del proceso de obtención de seguridad y su evaluación. En un formato amplio encierran un conjunto de buenas prácticas para dar cumplimiento al indicador referenciado. Los indicadores no son más que pasos que muestran qué, se debe hacer para elevar la seguridad en las bases de datos, en muchos casos son pasos simples que definen una acción en particular, otros indicadores tienden a ser más complejos y generales, dándole camino a una serie de pasos más concretos para dar solución al inicial, definidos como buenas prácticas.

Se define por **buenas prácticas** toda experiencia que se guía por principios, objetivos y procedimientos apropiados o pautas aconsejables que se adecuan a una determinada perspectiva normativa o a un parámetro consensuado, así como también toda experiencia que ha arrojado resultados positivos, demostrando su eficacia y utilidad en un contexto concreto.

Las **recomendaciones** se entienden como consejos o procesos que se sugieren para dotar al sistema de una seguridad adicional y que su desuso no demerita el nivel alcanzado con la aplicación de los indicadores. Acciones sencillas en su mayoría que el usuario debería aplicar para un mejor resultado, o simplemente un complemento practico que no conviene obviar.

1.4.1.1- Arquitectura en PostgreSQL.

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente dentro de la organización. [16]

Es el modelo inicial del sistema informático para el caso de una base de datos, define las tablas y disposición de las mismas, es un modelo y una descripción funcional de los requerimientos. Existen varios estándares de arquitectura; el más utilizado es el de tres capas; el comité ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Committee) define la descripción detallada de la arquitectura del sistema como la definición de un sistema de información. Esta arquitectura de un sistema de base de datos se divide en 3 niveles:

Nivel Interno: Es el más cercano al almacenamiento físico, es decir, es el que se ocupa de la forma como se almacenan físicamente los datos; (Físico).

Nivel Externo: Es el más cercano a los usuarios, es decir, es el que se ocupa de la forma como los usuarios reciben los datos; (aplicaciones).

Nivel Conceptual: Es el nivel de mediación entre los 2 anteriores; (modelo, (entidad/relación)).

La seguridad en la arquitectura está dada por la efectividad de representar o garantizar la seguridad en los otros procesos de desarrollo de las bases de datos, que es a su vez un mecanismo o herramienta que garantice la disponibilidad, integridad y confidencialidad de la información. Mediante la arquitectura se definen las conexiones con los datos y las medidas de seguridad en este ámbito definirán posteriormente en gran medida la seguridad de la base de datos. Es importante saber que una arquitectura que garantice la seguridad de la base de datos será aquella que sea capaz de garantizar la seguridad en cada nivel al que se hace referencia; como se muestra en la figura 1.

Figura 1: Arquitectura segura.

La arquitectura tres capas tiene:

- ✓ Un mayor grado de flexibilidad.
- ✓ Mayor seguridad, ya que la seguridad se puede definir independientemente para cada servicio y en cada nivel.
- ✓ Mejor rendimiento, ya que las tareas se comparten específicamente para cada nivel.

1.4.1.2- Diseño en PostgreSQL.

Erich Mario Gómez plantea en su tesis *Administración y optimización de un Sistema de Base de Datos Descentralizada, en PostgreSQL*, que el diseño debe dividirse en partes para su resolución, las partes son:

Diseño conceptual busca describir solo el contenido de la base de datos y no su estructura. Un esquema conceptual es una descripción de alto nivel de la estructura de la base de datos, independientemente del Sistema Gestor de Base de Datos (SGBD) que se vaya a utilizar para manipularla. [17]

Los modelos conceptuales deben tributar a la seguridad siendo buenas herramientas para representar la realidad, para esto deben poseer: **Expresividad[6], Simplicidad[7], Minimalidad[8], Formalidad[9].**

Diseño lógico parte del esquema conceptual y da como resultado un esquema lógico. [17] Este esquema es una descripción de la estructura de la base de datos que pueda ser procesado por un SGBD. El diseño lógico depende del tipo de SGBD que se vaya a utilizar. El esquema lógico es una fuente de información para el Diseño Físico. [18]

El Diseño físico parte del esquema lógico y da como resultado un esquema físico. Un esquema físico es una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos. Por ello, el diseño físico depende del SGBD concreto y el esquema físico se expresa mediante su lenguaje de definición de datos. [17] En esta etapa este diseño depende del SGBD que se va a utilizar y debe adaptarse a él.

El mayor problema de seguridad en el diseño es la dificultad para controlar que ciertos usuarios no accedan a contenido que no les está permitido. La información de toda empresa es importante, aunque unos datos lo son más que otros, por tal motivo se debe considerar el control de acceso a los mismos, no todos los usuarios pueden visualizar alguna información, por tal motivo para que un sistema de base de datos sea confiable debe de mantener un grado de seguridad que garantice la autentificación y protección de los datos. En un banco por ejemplo, el personal de nóminas solo necesita empleados del banco y no a otro tipo de información.

No existe un estándar de diseño por este motivo es difícil establecer pasos únicos que posibiliten un reconocimiento del nivel de seguridad; por ello se definieron indicadores de forma general adecuando una comprensión por cualquier desarrollador, en cada proceso de diseño se desarrollan una serie de acciones que posibilitan elevar la seguridad. Estas acciones son definidas como indicadores o buenas prácticas, entre los principales indicadores se encuentran:

---

[6] Suficientes conceptos para expresar perfectamente la realidad.

[7] Deben ser simples para que los esquemas sean fáciles de entender.

[8] Cada concepto debe tener un significado distinto.

[9] Todos los conceptos deben tener una interpretación única, precisa y bien definida.

Validar cada esquema mediante la normalización, Representación diseño físico y Definir las restricciones de integridad.

1.4.1.3- Configuración en PostgreSQL.

Configurar es adaptar una aplicación software o un elemento hardware al resto de los elementos del entorno y a las necesidades específicas del usuario. Es una tarea esencial antes de trabajar con cualquier nuevo elemento. La tendencia actual es a reducir las necesidades de configuración mediante sistemas que permiten al nuevo elemento detectar en qué entorno se instala, configurándose automáticamente sin requerir la participación del usuario. Para PostgreSQL el comportamiento en el sistema se controla con tres ficheros de configuración que se encuentran en el directorio de datos donde se inicializa el clúster, estos son tomados como indicadores necesarios a la hora de configurar la seguridad en el gestor:

pg_hba.conf: se utiliza para definir los diferentes tipos de accesos que un usuario tiene en el clúster; cómo, dónde y desde qué sitio un usuario puede utilizar el clúster PostgreSQL.

pg_ident.conf: se utiliza para definir la información necesaria en el caso que se utilice un acceso del tipo ident en pg_hba.conf.

postgresql.conf: En este fichero se pueden cambiar todos los parámetros de configuración que afectan al funcionamiento y al comportamiento de PostgreSQL en la máquina. Los cambios que se realicen en este fichero afectarán a todas las bases de datos que se hayan definido en el clúster de PostgreSQL.

Existen muchos parámetros que se pueden y con el tiempo se deberán ajustar, los más importantes y los cuales se deben cambiar antes de empezar a utilizar PostgreSQL son:

max_connections: Número máximo de clientes conectados a la vez a las bases de datos.

shared_buffers: Este parámetro es importantísimo y define el tamaño del buffer de memoria utilizado por PostgreSQL.

checkpoint_segments: Este parámetro es muy importante en bases de datos con numerosas operaciones de escritura. [19]

1.4.1.4- Codificación en PostgreSQL.

El código es el conjunto de instrucciones que permite la codificación y descodificación de la información que se transmite de manera que pueda ser intercambiada en forma comprensible entre la fuente y el destino. Entre los lenguajes más notorios en una base de datos se encuentra el SQL (Structured query language).

En el caso de PostgreSQL también utiliza PL/pgSQL (Procedural Language/PostgreSQL Structured Query Language) similar al PL/SQL de Oracle; es un lenguaje previsto para el gestor de base de datos PostgreSQL. Permite ejecutar comandos SQL utilizando un lenguaje de sentencias imperativas y uso de funciones, esto da un mayor control automático que las funciones SQL básicas.

El proceso de codificación es fundamental a la hora de garantizar la seguridad en una base de datos, puesto que aquí se definen las consultas y la forma de enlazar los datos con la aplicación, en esta etapa una mala práctica de codificación ocasionará el surgimiento de vulnerabilidades en la base de datos. En la codificación el principal riesgo es el ataque por inyección SQL que pertenece al tipo de ataques de validación de entradas del usuario. Probablemente esta sea la vulnerabilidad Web más importante de la historia. Una inyección SQL es una alteración de la sintaxis de código, o inyección de código. Es importante entender la necesidad para el programador del uso de buenas prácticas de programación para minimizar las vulnerabilidades.

Para la mejora de la seguridad, teniendo en cuenta que el principal problemas es la inyección SQL se tienen indicadores como: Auditoria de código SQL, Estandarización del código y Seguridad contra inyecciones SQL; los cuales denotan el hilo conductor del procedimiento de obtención de seguridad en la codificación.

1.5 Aplicaciones para la toma de decisiones.

Las bases de datos de ayuda a la toma de decisiones son las llamadas almacén de datos o *data warehouse* por su traducción en inglés; es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza. Se trata, sobre todo, de un expediente completo de una organización, más allá de la información transaccional y operacional, almacenado en una base de datos diseñada para favorecer el análisis y la divulgación eficiente de datos (especialmente OLAP[10]).

[10] procesamiento analítico en línea.

1.5.1-Características de un Data Warehouse.

Bill Inmon fue uno de los primeros autores en escribir sobre el tema de los almacenes de datos, define un **Data Warehouse** (**almacén de datos**) en términos de las características del repositorio de datos:

- ✓ **Orientado a temas**: los datos en la base de datos están organizados de manera que todos los elementos de datos relativos al mismo evento u objeto del mundo real queden unidos entre sí.
- ✓ **Variante en el tiempo**: los cambios producidos en los datos a lo largo del tiempo quedan registrados para que los informes que se puedan generar reflejen esas variaciones.
- ✓ **No volátil**: la información no se modifica ni se elimina, una vez almacenado un dato, éste se convierte en información de *sólo lectura*, y se mantiene para futuras consultas.
- ✓ **Integrado**: la base de datos contiene los datos de todos los sistemas operacionales de la organización, y dichos datos deben ser consistentes.

Definición de Ralph Kimball

Este es otro conocido autor en el tema de los data warehouse, define un almacén de datos como: "una copia de las transacciones de datos específicamente estructurada para la consulta y el análisis". También fue Kimball quien determinó que un data warehouse no era más que: "la unión de todos los Data marts[11] de una entidad". Defiende por tanto una metodología ascendente (bottom-up) a la hora de diseñar un almacén de datos.

Las definiciones anteriores se centran en los datos en sí mismos. Sin embargo, los medios para obtener y analizar esos datos, para extraerlos, transformarlos y cargarlos, así como las diferentes formas para realizar la gestión de datos son componentes esenciales de un almacén de datos. Muchas referencias a un almacén de datos utilizan esta definición más amplia. Por lo tanto, en esta definición se incluyen herramientas para la inteligencia empresarial, herramientas para extraer, transformar y cargar datos en el almacén de datos, y herramientas para gestionar y recuperar los metadatos. [20]

[11] almacén de datos especializado (contiene datos para dar apoyo solamente a un área específica de análisis de negocios), orientado a un tema, integrado, volátil (los usuarios pueden actualizar los datos e incluso, posiblemente, crear nuevas tablas para algún propósito) y variante en el tiempo para apoyar un subconjunto específico de decisiones de administración.

Existen 2 criterios bien identificados y que han marcado claramente su tendencia sirviéndole de guía a la comunidad mundial en cuanto a este tema.

Estas tendencias son conocidas como Metodología Kimball y Metodología de Inmon, en honor a sus creadores Ralph Kimball y William H. Inmon. Ralph Kimball y Bill Inmon son dos de las personalidades referentes y más influyentes en el área de data warehousing. El primero es un especialista reconocido a nivel mundial en el diseño de Data Warehouse y creador del enfoque Multidimensional; mientras que el segundo es el creador del término Data Warehouse y considerado como el padre de la disciplina.

La figura 2 muestra un la composición de un Data Warehouse.

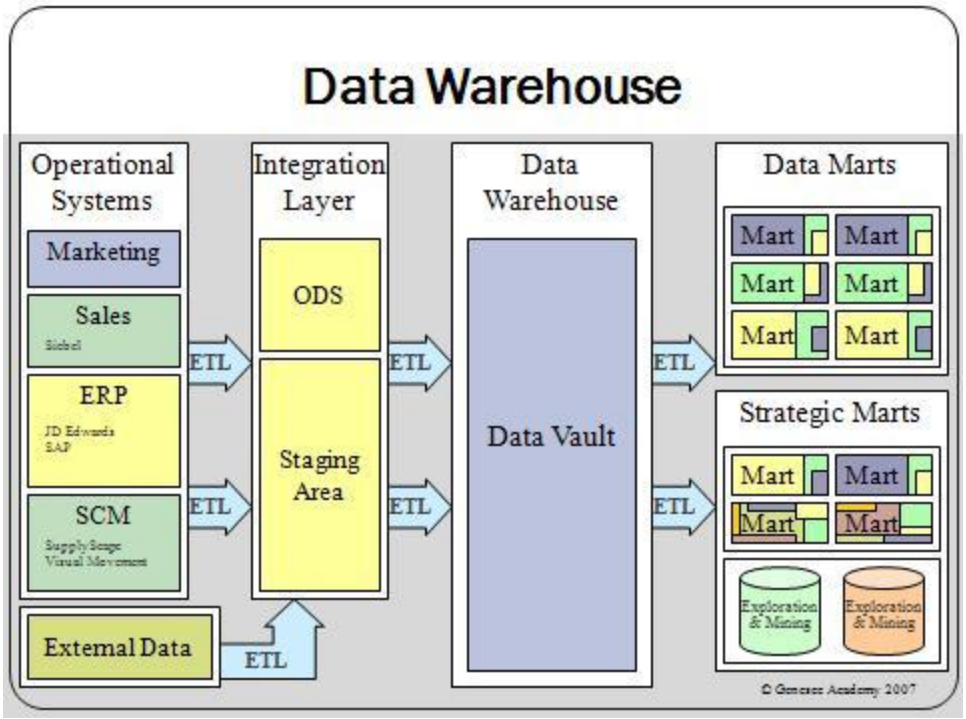


Figura 2: Data Warehouse.

Por lo general, la actualización que se da está limitada a operaciones de carga o actualizaciones periódicas y esas operaciones están dominadas a su vez por INSERTs, los DELETEs se realizan muy ocasionalmente y los UPDATEs casi nunca. También vale la pena señalar las siguientes características adicionales de las bases de datos de apoyo para la toma de decisiones.

- ✓ Se tiende a usar las columnas en combinación.
- ✓ Por lo general, no preocupa la integridad.
- ✓ Las claves incluyen frecuentemente un componente temporal.

- ✓ La base de datos tiende a ser grande (especialmente cuando se acumulan los detalles de las transacciones de negocios a lo largo del tiempo, y con frecuencia así sucede).
- ✓ La base de datos tiende a estar muy indexada[12]
- ✓ La base de datos involucra frecuentemente varios tipos de redundancia controlada. [21]

Este tipo de base de datos tiende a engendrar muchas complejidades de la escritura de consultas entre las que se encuentran:

- ✓ Complejidad de expresiones lógicas: las consultas de apoyo para la toma de decisiones involucran expresiones complejas en la cláusula WHERE; las cuales son difíciles de escribir, difíciles de comprender y difíciles de manejar adecuadamente por el sistema.

- ✓ Complejidad de joins: las consultas de apoyo para la toma de decisiones requieren frecuentemente acceso a muchas clases. Por consecuencia, en una base de datos diseñada adecuadamente dichas consultas involucran, por lo general a muchos joins.

- ✓ Complejidad de función: las consultas de apoyo para la toma de decisiones involucran frecuentemente funciones estadísticas y matemáticas. Pocos productos soportan tales funciones.

- ✓ Complejidad analítica: las preguntas de negocios rara vez son respondidas con una sola consulta. No sólo es difícil para los usuarios escribir consultas de gran complejidad, sino que las limitaciones que tienen las implementaciones de SQL pueden impedir el procesamiento de una de las consultas. [21]

1.6 -Herramientas de seguridad para PostgreSQL.

PaGoDump - PostgreSQL v1.0.0.16 es una utilidad creada para realizar copias de seguridad de una base de datos PostgreSQL. Hace copias de seguridad coherente, incluso si la base de datos se está utilizando al mismo tiempo. PaGoDump - PostgreSQL v1.0.0.16 no bloquea el acceso a otros usuarios de la base de datos (lectores o escritores), también trabaja con bases de datos con nombres (Unicode) y realiza volcados a cualquier archivo unicode de nuevo.

[12] Registrar ordenadamente información para elaborar su índice.

Bacula es un conjunto de programas para gestionar copias de seguridad, recuperar y verificar los datos de una máquina en una red de máquinas de diferentes tipos. El servicio Director de Bacula supervisa todas las copias de seguridad, la recuperación, la verificación y las operaciones de almacenamiento. Se puede ejecutar como un demonio o como un servicio en primer plano de modo que los administradores puedan usarlo para programar copias de seguridad y recuperar archivos. Esta versión almacena el catálogo Bacula en un servidor PostgreSQL y por lo tanto se recomienda para instalaciones grandes.

GreenSQL v1.2 es capaz de proteger PostgreSQL. GreenSQL es diseñado para proteger bases de datos contra ataques de inyección SQL y otros cambios no autorizados, en una manera similar a un cortafuegos que protege una red TCP/IP contra ataques. También proporciona un interfaz de usuario gráfico para supervisar el cortafuego de base de datos.

Camouflage: Permite encriptar archivos y ocultarlos de forma que se adjunten a el tipo de archivo de su selección, se comportarán como el tipo de archivo que se decidió puede manipularse como tal y luego se extrae necesitando para esto una contraseña.

Guardium: Es una firma de **IBM** (actualmente incorporados al catálogo de Information on Demand (Información bajo demanda)**;** Guardium está especializada en la fabricación de tecnología para la monitorización en tiempo real de la actividad de las bases de datos, permitiendo a las empresas que la utilizan detectar fraudes, ataques externos y otras actividades ilícitas.

DataVantage: DataVantageXBR es una herramienta de la compañía DataVantage, es un sistema de prevención de pérdidas; con 17 años de experiencia esta compañía es el proveedor #1 en Software de Prevención de Pérdidas.

Kerberos: Kerberos es un programa de autenticación tiene tres niveles:

Autenticación: Prueba que el usuario es quien dice ser. Puede ser que la autenticidad se establezca al inicio de la conexión de red y luego se asuma que los siguientes mensajes de una dirección de red determinada se originan desde la parte autenticada.

Integridad de datos: Asegura que los datos no se modifican en tránsito. Se requiere autenticación de cada mensaje, sin importar el contenido del mismo. Esto se denomina mensaje seguro.

Privacidad de datos: Asegura que los datos no son leídos en tránsito. En este caso, no sólo se autentica cada mensaje, sino que también se cifra. Estos mensajes son privados.

1.7- Factores de calidad vs Seguridad.

Uno de los aspectos a tener en cuenta en la creación de las bases de datos es la repercusión de una elevada seguridad en la calidad, muchos son los autores que han determinado cuales son los factores de calidad, tales como McCall y Hewlett-Packard – FRUPS; para esta investigación se utilizan los factores establecidos por la **ISO 9126**.

Funcionalidad. El grado en que el software satisface las necesidades indicadas por los siguientes atributos: idoneidad, corrección, interoperatividad, conformidad y seguridad.

Confiabilidad. Cantidad de tiempo que el software estará disponible para su uso. Está referido por los siguientes atributos: madurez, tolerancia a fallos y facilidad de recuperación.

Usabilidad. Grado en que el software es fácil de usar. Viene reflejado por los siguientes atributos: facilidad de comprensión, facilidad de aprendizaje y operatividad.

Eficiencia. Grado en que el software hace óptimo el uso de los recursos del sistema. Está indicado por los siguientes atributos: tiempo de uso y recursos utilizados.

Facilidad de mantenimiento. La facilidad con que una modificación puede ser realizada. Está indicada por los siguientes atributos: facilidad de cambio y estabilidad.

Portabilidad. La facilidad con que el software puede ser llevado de un entorno a otro. Está referido a los siguientes atributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

La siguiente imagen muestra el equilibrio de los factores de calidad.

Figura 3: Equilibrio de la calidad y sus factores.

Para obtener una aplicación con calidad se debe equilibrar los factores de calidad entre los que esta la seguridad; y analizar la forma de interactuar con los otros factores de calidad. La siguiente tabla ilustra algunos factores de calidad y su relación con la seguridad.

Otros factores de calidad	Relación con la Seguridad	Descripción del factor
Facilidad de uso.	Una aplicación difícil de usar provoca que se cometan violaciones de seguridad a causa de errores de usuarios. Una aplicación segura no tiene porque ser una aplicación difícil de usar.	La facilidad de uso se mide por la complejidad de aprender a usar en software; una aplicación con facilidad de uso es aquella que sea sencilla para el usuario.
Portabilidad	Un software debe ser capaz de	Portabilidad es la

	adaptarse al cambio y de tener facilidad de instalación sin violar su seguridad es necesario que los desarrolladores en conjunto con los clientes sean capaces de llegar a un equilibrio.	capacidad de adaptación al entorno, de un software y su facilidad de traslado e instalación.
Facilidad de mantenimiento	Un sistema de bases de datos con buena facilidad de mantenimiento siempre será más seguro en la medida que el equipo de desarrollo sea capaz de detectar con facilidad los errores y posibilidad de corrección. La seguridad no tiene por qué afectar la facilidad de mantenimiento puesto que la seguridad se obtiene a través de pautas predeterminadas y recomendadas que permiten establecer pasos registrados y documentados correctamente.	La facilidad de mantenimiento es el esfuerzo requerido para localizar y arreglar un error en un programa. La pregunta asociada a este factor sería: ¿Puedo corregirlo?
Eficiencia	La eficiencia denota el buen trabajo del desarrollador, a mayor eficiencia menor probabilidad de que ocurran fallos del sistema y mayor será la probabilidad de detectar o impedir un ataque por lo que elevará la seguridad. La	Grado en que el software hace óptimo el uso de los recursos del sistema. Está indicado por los siguientes atributos: tiempo de uso y recursos utilizados. En la

	capacidad de procesamiento y velocidad de un software tapa vulnerabilidades como desbordamiento de buffer y cuellos de botellas.	actualidad está llamado a la correlación gráfica con la realidad y la eficiencia de portabilidad del software.

Figura 4: Factores de calidad y su relación con la seguridad.

1.8-Generalidades del capítulo.

- ✓ De forma general se entiende por seguridad en una base de datos de PostgreSQL las herramientas, métodos o técnicas para asegurar de forma sistemática la integridad, disponibilidad y confidencialidad de la información almacenada en la base de datos.
- ✓ Del PostgreSQL se analiza el desarrollo y evolución del mismo; las mejoras de seguridad incluidas son epígrafes de interés en este capítulo.
- ✓ Se definieron los procesos de desarrollo de bases de datos: arquitectura, configuración, diseño y codificación.
- ✓ Existe un número considerable de herramientas y programas de software que ayudan a mejorar la seguridad de las bases de datos.
- ✓ La calidad debe estar presente en cualquier software y sin dudas la seguridad es un factor de peso en este sentido, así como otros factores de calidad.

CAPITULO II
Guía de indicadores para elevar la seguridad en bases de datos realizadas en PostgreSQL.

Introducción

Aunque muchos piensen lo contrario, no existe una base de datos invulnerable; este asunto aunque se cree maduro, la realidad que se observa es que la 'seguridad del dato' presenta todavía numerosas lagunas, por errores de concepto más que por carencias tecnológicas de los propios sistemas de base de datos. Los sistemas de base de datos comerciales son enormemente complejos en la actualidad. Todo sistema complejo tiene defectos, los defectos derivan en vulnerabilidades, y a través de estas vienen los ataques y uso indebido. Un ejemplo de los principales problemas de seguridad en bases de datos se muestra en la figura 5.

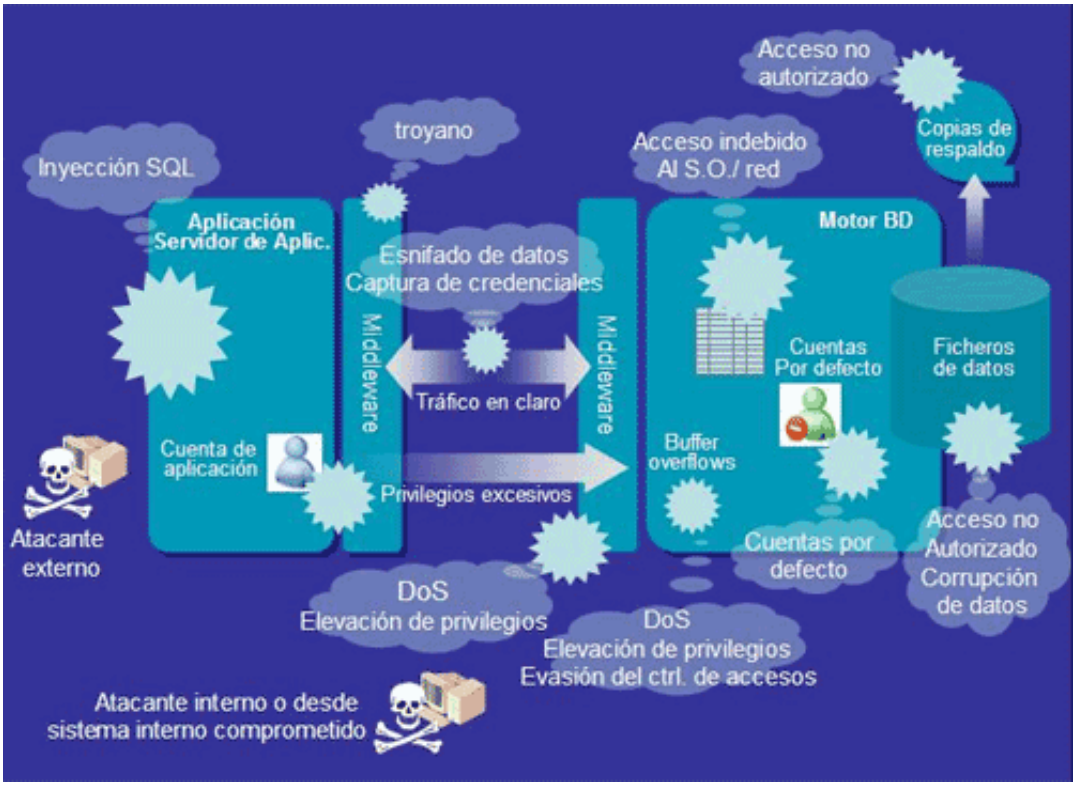


Figura 5: Problemas de seguridad en bases de datos [21]

En este capítulo se presenta la Guía de indicadores que tiene por finalidad obtener seguridad en bases de datos desarrolladas en PostgreSQL en cuanto a: Arquitectura, Diseño, Configuración y Codificación; proporciona una indicación de los elementos que posibilitan al desarrollador obtener una elevada seguridad en estas bases de datos. Desarrollada a partir de buenas prácticas que proponen usuarios de experiencias y autores de artículos reconocidos a nivel internacional en el campo en el que se desarrolla la Guía, además tiene su base en los manuales de usuarios que proponen los desarrolladores de PostgreSQL, fundamentos y metodologías asentadas en años de explotación para el diseño y desarrollo de bases de datos. Está dirigida fundamentalmente para desarrolladores de bases de datos, administradores y clientes o cualquier usuario con conocimientos medios en el desarrollo de bases de datos. Esta Guía está diseñada para el gestor de PostgreSQL aunque algunas de sus partes son fácilmente utilizables en otros SGBD.

El objetivo fundamental de la Guía es conducir al desarrollador en los procesos de construcción de una base de datos de forma tal que se le brinde información precisa, posibilitando que se eleve la seguridad de la base de datos, no se propone una metodología de diseño o arquitectura, ni se tiene en cuenta ninguna metodología o modelo de desarrollo de software, se trató de ser lo más flexible posible para permitir al usuario[13] comodidad sin establecer una estructura rígida.

A través de la experiencia de un grupo de desarrolladores y científicos reconocidos en esta área del conocimiento se dan un conjunto de recomendaciones necesarias para alcanzar ciertos niveles de seguridad, recomendaciones que muchas veces son obviadas por desconocimiento de desarrolladores novatos.

El presente capítulo solo contiene una síntesis de la Guía de indicadores, para entrar en detalles se debe acceder directamente a la Guía que es el producto resultado de la investigación.

2.1- Indicadores de seguridad

No es objetivo del presente trabajo el enseñar a los desarrolladores a realizar bases de datos; estos indicadores solo muestran el camino a tomar para tener una base de datos con determinados niveles de seguridad, aunque en muchos casos se redirige el estudio a algún indicador en concreto para conocimiento del desarrollador. Tómese la Guía de indicadores como el primer escalón para elevar el conocimiento básico del desarrollo de un sistema de bases de datos.

[13] Se le llamará usuario a los desarrolladores de bases de datos que utilicen la guía; y no a los usuarios de las bases de datos.

En términos generales, se entiende por **guía** aquello que tiene por objetivo y fin conducir, encaminar y dirigir algo para que se llegue a buen puerto en la cuestión que se trate. En un caso más concreto portador de pasos o mecanismos de ejecución, indicativos de la vía correcta a seguir para obtener un resultado satisfactorio.

2.1.1- Estructura de la Guía.

La Guía de indicadores se compone por 5 epígrafes: arquitectura, diseño, configuración y codificación estos epígrafes están compuestos por indicadores, buenas prácticas y recomendaciones; como se muestra en la figura 6 y otro epígrafe donde se propone la utilización de herramientas para elevar la seguridad.

Figura 6: Composición de la Guía de indicadores para la mejora de la seguridad de bases de datos desarrolladas en PostgreSQL.

Los **indicadores** son puntos de referencia para elevar la seguridad; en este caso, compuestos por buenas prácticas para dar cumplimiento al indicador referenciado; si se cumplen estos indicadores se obtiene cierto nivel de seguridad en el proceso.

Se define por **buenas prácticas** toda experiencia que ha arrojado resultados positivos y se guía por principios; pautas aconsejables que se adecuan a acciones correctas en la construcción de bases de datos.

Las **recomendaciones** son acciones sencillas en su mayoría que el usuario debería aplicar para un mejor resultado, o simplemente un complemento práctico que no conviene obviar.

La estructura de la Guía se compone por los cuatros procesos de desarrollo de una base de datos y el epígrafe de herramientas. Se expone además como complemento una lista de chequeo para posibilitar al desarrollador conocer hasta qué grado da solución a los indicadores definidos en la Guía, y conocer el nivel de seguridad de las bases de datos según estos indicadores. En la figura 7, se representa la composición de los epígrafes con los indicadores de cada etapa de desarrollo de bases de datos.

Figura 7: Estructura de la Guía de indicadores.

2.2- Epígrafes de la Guía de indicadores para obtener seguridad en las bases de datos realizadas en PostgreSQL.

2.2.1- Arquitectura:

En la guía de indicadores se trata la arquitectura de tres capas o ANSI/SPARC sirviendo para explicar la estructura de los epígrafes; sin afirmar que toda estructura pueda coincidir con esta infraestructura totalmente, ni imponer esta arquitectura como única.

Para adentrarse en la arquitectura ANSI/SPARC primero se debe conocer que PostgreSQL se basa en una arquitectura Cliente-Servidor, los clientes son las diversas aplicaciones que se ejecutan sobre el DBMS, tanto escritas por el usuario como acopladas; el servidor no es más que el propio DBMS que soporta todas las funciones de los niveles externo, conceptual e internos.

La arquitectura de tres capas incluye en sí la codificación, el diseño y la configuración ya que en la arquitectura se garantiza la seguridad dependiendo de estos factores.

Contribuyen a elevar la seguridad desde la arquitectura los siguientes indicadores:

Existencia de Independencia física de los datos.

La independencia física es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos). Por ejemplo, puede ser necesario reorganizar ciertos ficheros físicos con el fin de mejorar el rendimiento de las operaciones de consulta o de actualización de datos. Se utiliza para separar las aplicaciones y las estructuras físicas de almacenamiento.

Existencia de Independencia lógica de los datos.

La independencia lógica es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se puede modificar el esquema conceptual para ampliar la base de datos o para reducirla.

Su utilidad está dada en que aunque se modifique el esquema conceptual, la vista que poseen las aplicaciones (los esquemas externos) no será afectada. [26]

Protección física de los datos externa al SGBD.

Debe existir un mecanismo de protección exterior a los SGBD, el almacenamiento físico hace que los datos puedan ser accesibles si se tiene acceso al dispositivo de almacenamiento por lo que es importante un sistema que garantice que no se acceda a estos dispositivos sin autorización.

La protección externa se utiliza como una condición de seguridad adicional al gestor, para impedir el acceso no autorizado a través del sistema operativo que contiene la base de datos.

2.2.2- Diseño:

El diseño de una base de datos sin importar el SGBD está dado por tres etapas fundamentales; si en cada una de ellas se logran los objetivos de asegurar la seguridad posteriormente se asegurará la seguridad para la base de datos en construcción.

No se propone una metodología de diseño, ya que cada cual tiene su propia forma de diseñar y algunos proyectos incluso tienen sus propios métodos de diseño ajustados a sus propias necesidades; por esto se hace difícil proponer una serie de pasos sin violar estos preceptos, por lo que proponen pasos recomendados en estas metodologías y que se pudieran incluir para un desarrollo acertado de una base de datos sin esquematizar; si llegado a un punto del diseño alguno de estos indicadores entra en contraposición a lo que se debe realizar, es necesario escoger entonces el camino más acertado para la solución propuesta.

2.2.2.1- Diseño conceptual

El modelo conceptual no depende de un gestor específico y es importante contar con una buena definición de los datos, las entidades y sus atributos; aunque no parezca relacionado con la seguridad un mal diseño conceptual conlleva a un mal diseño general de la base de datos y una base de datos mal diseñada acarrea problemas de mala manipulación de datos y de tablas repetidas.

De un buen diseño conceptual depende el diseño lógico, la Universidad de Cincinnati propone un resumen de una metodología de diseño para las bases de datos relacionales. Los factores de éxito críticos en el diseño de base de datos son tratados a continuación:

Construcción del modelo conceptual de datos local para cada vista de usuario.

Construir un modelo conceptual de datos locales de una empresa para cada vista de usuario específico. Proporciona crear una vista de usuario a cada rol específico, para mejorar la otorgación de permisos y restricciones.

Entre las buenas prácticas que componen este indicador están:

- ✓ Identificar los tipos de entidad.
- ✓ Identificar los tipos de relación.

- ✓ Identificar atributos.
- ✓ Determinar los dominios de atributos.
- ✓ Determinar el candidato principal a atributo clave.
- ✓ Revisión del modelo conceptual de datos locales con el usuario.

Construcción y validación del modelo lógico local de datos para cada usuario.

Construir un modelo lógico de datos basado en el modelo conceptual de datos para cada vista de usuario de la empresa, y luego validar el modelo mediante la técnica de normalización y en contra de las transacciones requeridas.

Esto se hace para establecer las relaciones lógicas de los datos para cada vista de usuario, o sea se ejecuta primeramente para cada rol antes de crear un modelo lógico global, lo que posibilita un mejor control de las transacciones de los usuarios.

Para una correcta ejecución del indicador se expone como buena práctica realizar el Modelo conceptual de datos para el modelo lógico de datos locales. Aquí se eliminan características indeseables y se asigna este modelo a un modelo lógico de datos locales.

2.2.2.2- Diseño lógico:

Las reglas de diseño lógico no dependen del uso que se pretenda dar a la base de datos, ya que se aplican las mismas reglas sin tomar en cuenta los tipos de aplicaciones. Por lo tanto, no debe haber diferencia si esas aplicaciones son operacionales (OLTP) o de apoyo para la toma de decisiones; de cualquier forma, es necesario seguir el mismo procedimiento de diseño. [27]
En la investigación realizada en la línea de desarrollo de almacenes del proyecto DATECD se conoce que la metodología fundamental utilizada por el equipo de desarrollo tiene su base en la metodología propuesta por Kimball y la Metodología para el Diseño Conceptual de Almacenes de Datos de Leopoldo Zenaido Zepeda Sánchez[14], adicionando nuevos componentes que surgen por las experiencias adquiridas en estos proyectos.

[14] Doctor en computación Universidad Politécnica de Valencia

Se consideran una serie de indicadores específicos para las aplicaciones de ayuda a la toma de decisiones puesto que estas aplicaciones presentan características especiales y aunque Pérez y Fernandeze exponen que no existen grandes diferencias; a partir de estas características planteadas en general por estos mismos autores se establecieron una serie de indicadores para este tipo de diseño, sin olvidar que los indicadores propuestos para las OLTP siguen aplicándose a las base de datos almacenes o Data Warehouse.

Validación de cada esquema mediante la normalización.

El objetivo de este paso es asegurar que cada relación que se derive del modelo lógico de datos, aunque se plantea que es óptimo llevarlo hasta FNBC, hay veces que en 3ra FN es una buena transformación. La primera forma normal no admite valores que se comportan como conjunto dentro de otros valores, es decir, atributos multievaluados o compuestos, en la segunda forma normal se eliminan las dependencias parciales y en la tercera forma normal se eliminan las dependencias transitivas.

En la mayoría de las ocasiones, después de la normalización se realiza lo que se conoce como depuración de las relaciones, que es cuando se analiza si el resultado obtenido luego de la normalización es óptimo, o si hay otro mejor, ya sea por unir atributos a otros o cambiar algún elemento, sin embargo, el objetivo ahora es conseguir una base de datos normalizada por las siguientes razones:

- ✓ Un esquema normalizado es robusto y carece de redundancias, por lo que está libre de ciertas anomalías que pueden provocar cuando se actualiza la base de datos.
- ✓ Los equipos informáticos de hoy en día son mucho más potentes, por lo que en ocasiones es más razonable implementar bases de datos fáciles de manejar, a costa de un tiempo adicional de proceso.
- ✓ La normalización produce bases de datos con esquemas flexibles que pueden extenderse con facilidad. [28]

Validación de cada esquema frente a las transacciones del usuario.

La finalidad es validar cada esquema lógico local para garantizar que puede soportar las transacciones requeridas por los correspondientes usuarios. Estas transacciones se encontrarán en las especificaciones de requisitos de usuario.

Definición de las restricciones de integridad.

Las restricciones de integridad son reglas que se imponen para proteger la base de datos, de modo que no pueda llegar a un estado inconsistente.

Este es el indicador de máxima prioridad y el que más concienzudamente debe hacerse puesto que en ocasiones el diseñador para acomodar su trabajo deja todo el trabajo de validación al programador y trae como resultado que se alteren valores de entrada o exista inconsistencia en el diseño de las entidades. Todas las restricciones de integridad establecidas en este indicador se deben reflejar en el diccionario de datos para que puedan ser tenidas en cuenta durante la fase del diseño físico. [25]

Construcción y validación del Modelo Global de Datos Lógico.

Este indicador consiste en el agrupamiento y acoplamiento de los modelos locales de datos lógico en un solo modelo general según la parte del negocio que se está tratando.

Se usa para combinar los modelos locales lógico de datos en un único modelo global de datos lógicos que se pueda utilizar para representar a la parte de la empresa que se desea modelar.

Este indicador posee una serie de buenas prácticas las cuales son:

- ✓ Combinar los modelos lógicos de datos locales en el modelo universal.
- ✓ Validar modelo general de datos lógicos.
- ✓ Verificar el crecimiento futuro.

Son buenas prácticas además:

- ✓ Traducir Modelo Global de Datos Lógico para el DBMS de destino.
- ✓ Diseño de relaciones base para el DBMS de destino.

2.2.2.3- Diseño Físico

En el diseño físico se analiza a los Data Warehouse o almacenes de datos; en el trabajo **"APOYO PARA LA TOMA DE DECISIONES"** de los autores Santiago Pérez y Néstor Fernandeze se plantea que estas bases de datos tienden a ser grandes y fuertemente indexadas e involucran diversos tipos

de redundancia controlada; para contrarrestar los inconvenientes que esto pueda traer los profesores, Santiago Pérez y Néstor Fernandeze recomiendan el uso del “partido” (también conocido como fragmentación).

Algunos de los errores comunes a los que los desarrolladores pueden someterse son:

- ✓ Filas duplicadas. Los diseñadores de apoyo para la toma de decisiones dicen con frecuencia que sus datos simplemente no tienen un identificador único y que por lo tanto, tienen que permitir duplicados. Esto surge debido a que el esquema físico no deriva a partir de un esquema lógico (el cual probablemente nunca se creó).
- ✓ Nulos. Los diseñadores tratan frecuentemente de ahorrar espacio permitiendo nulos en las columnas. Sin embargo, por lo general dichos intentos son erróneos.
- ✓ No se diseñan tablas de resumen. La cuestión del diseño lógico de tablas de resumen es con frecuencia ignorada, lo que da lugar a una redundancia no controlada y a dificultades para mantener la consistencia.

Representación del diseño físico.

Para la representación del diseño físico se debe determinar las organizaciones de archivos y métodos de acceso que se utilizarán para almacenar las relaciones base; que es la forma en que se realiza el almacenamiento secundario de las relaciones y tuplas. Un esquema de base de datos físico es un plan de cómo almacenar los datos en un sistema particular.

Este indicador se emplea para determinar la organización de los datos físicos y como se almacenan en la base de datos. Además define la funcionalidad de las transacciones que se ejecutan en la base de datos y analiza las transacciones importantes.

Las buenas prácticas de dicho indicador se agrupan en:

- ✓ Analizar las transacciones.
- ✓ Elegir organizaciones de base.
- ✓ Elegir índices secundarios.

- ✓ Estimar las necesidades de espacio en disco.
- ✓ Considerar la introducción de redundancia controlada.
- ✓ Monitorear y ajustar el sistema operativo.

2.2.3- Configuración:

La configuración es uno de los procesos de más peso en la ejecución de las bases de datos, aquí se definen las características que tendrán y su desempeño. En la etapa de configuración se definen los indicadores que ayudan a elevar la seguridad y junto a ellos las buenas prácticas y recomendaciones.

Creación de contraseñas fuertes para administrador.

Se plantea cómo debe ser una contraseña para garantizar la seguridad tanto administrativa como de usuarios comunes, se busca obtener garantías a partir de la creación de claves con características específicas que protejan la base de datos de posibles ataques.

Creación de usuarios y permisos según los roles.

Define la forma de crear los usuarios en PostgreSQL, los permisos que pueden tener y las formas de autorización según el nivel de acceso, asegurando la información; permitiendo un mayor control de las entradas y modificaciones de cada usuario en la base de datos, para gestionar posibles errores que presente la base de datos.

PostgreSQL posee niveles de usuarios para un mejor aseguramiento de los datos almacenados. En el momento en que el administrador de Postgre comienza a trabajar se crea el: Superusuario de Postgres: este es dueño de los ficheros de la bases de datos y binarios. Como superusuario de la base de datos, no se le aplica ninguno de los mecanismos de protección y puede acceder a cualquiera de los datos de forma arbitraria. Además, al superusuario de Postgres se le permite ejecutar programas de soporte que generalmente no están disponibles para todos los usuarios. [12]

Las buenas prácticas son:

- ✓ Autorizar las Vistas.
- ✓ Controlar acceso.

- ✓ Mecanismos de seguridad discrecionales: se usan para otorgar privilegios a los usuarios, incluida la capacidad de tener acceso a archivos, registros o campos de datos específicos en un determinado modo. [26]
- ✓ Mecanismos de seguridad obligatorios: sirven para imponer igualdad de múltiples niveles clasificando los datos y los usuarios en varias clases (o niveles) de seguridad e implementando después la política de seguridad apropiada de la organización. [26]

Registro de auditoría.

El registro de auditoría es un archivo o base de datos especial en el que el sistema lleva automáticamente la cuenta de todas las operaciones realizadas por los usuarios sobre lo datos, estos deben dar la posibilidad de consultarlo con un lenguaje de consulta normal; siempre y cuando se tengan los permisos.

Configuración de copias de seguridad (Backups).

Las copias de seguridad en un sistema informático tienen por objetivo mantener cierta capacidad de recuperación de la información ante posibles pérdidas. Esta capacidad puede llegar a ser muy importante, incluso crítico, para las empresas. Se han dado casos de empresas que han llegado a desaparecer ante la imposibilidad de recuperar sus sistemas al estado anterior después de producirse un incidente grave de seguridad.

Configuración segura del pg_hba.conf.

El fichero pg_hba.conf de Postgre SQL se utiliza para definir cómo, dónde y desde qué sitio un usuario puede utilizar el clúster de PostgreSQL. De una buena configuración o no dependerá la seguridad de las bases de datos que se generen con este SGBD, ya que aquí se define quién tiene acceso; cómo se validan las contraseñas y hasta qué usuario tiene permitido el acceso en la base de datos.

Para una acertada configuración del indicador se exponen en la Guía una serie de buenas prácticas que poseen pasos que se deben configurar:
Uno de los pasos más importante es:


Quitar los permisos de acceso por default: para ejecutarlo hay que tener en cuenta elementos como:

- ✓ Type: Tipo de conexión que se usará, son de dos formas: local y remota. También en la parte del host está la opción con soporte para SSL **"hostssl"**.
- ✓ Database: Es el nombre de la base de datos a la que se puede conectar.
- ✓ User: Es el usuario que tendrá acceso a la(s) base(s) de datos que le sea(n) permitida(s).
- ✓ Cidr-address: Es la columna de la IP y la netmask que puede acceder al servidor. [29]


Configuración segura del postgresql.conf.

El archivo **postgresql.conf** define las características de las bases de datos que se generan con el SGBD. Con la edición de este archivo se puede definir el número máximo de conexiones en correspondencia con los usuarios, esto toma carácter delicado cuando se comprende que estos ficheros no solo involucran la seguridad si no otros factores como el rendimiento, y se debe tomar un nivel balanceado para no desproporcionar la calidad de las bases de datos.


Buenas prácticas:

- ✓ Determinación del número máximo de clientes conectados a la vez a las bases de datos utilizando el Max_connections.
- ✓ Definición el tamaño del buffer de memoria utilizando el parámetro Shared_buffers. [30]
- ✓ Definición del tipo de mantenimiento utilizando el parámetro Maintenance_work_mem.
- ✓ Optimización de la lectura de datos utilizando el parámetro Effective_cache_size.
- ✓ Configuración del parámetro Checkpoint_segments para operaciones de escritura (Insert, Update, Delete).
- ✓ Definición del tiempo máximo de autentificación con el parámetro Authentication_timeout.

- ✓ Configuración de los parámetros Krb_server_keyfile, krb_srvname, krb_server_hostname y krb_caseins_users: estos parámetros definen el uso del servidor Kerberos en caso de su utilización. Kerberos permite la autentificación segura, aunque su uso puede ser muy complejo es potente.

- ✓ Utilización de SSL: PostgreSQL tiene soporte nativo para utilizar conexiones SSL para cifrar comunicaciones cliente / servidor logrando una mayor seguridad. Esto requiere que OpenSSL esté instalado en ambos sistemas cliente y servidor. Por último se debe activar el fichero SSL (Enable) en postgresql.conf

El funcionamiento de la encriptación por SSL se define en profundidad en la Guía dada su importancia y gran utilidad a la hora de enviar datos encriptados por la red. La figura 8 muestra cómo funciona el empaquetado de red.

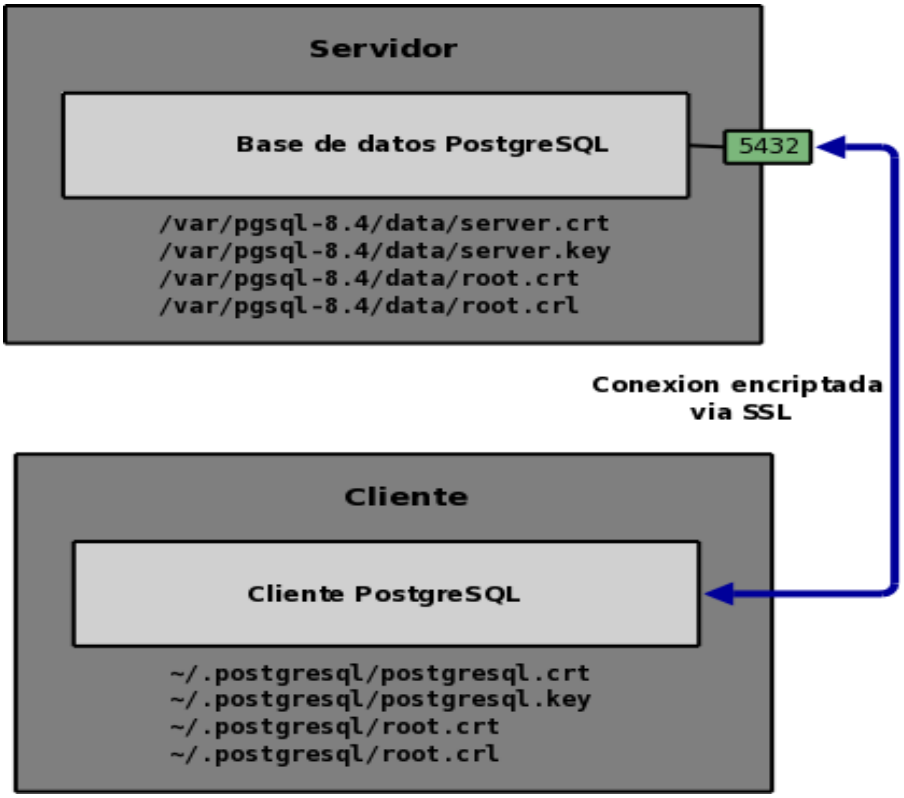


Figura 8: Empaquetado SSL.

Entre las recomendaciones de más importancia de la propuesta de solución se encuentra la configuración de Pgcrypto; puesto que este es un tema novedoso y que exiten pocos conocimientos al respecto, se propone como recomendación solamente y no como buena práctica.

PostgreSQL posee un módulo llamado pgcrypto, el cual permite varias funciones de encriptado de datos a través de funciones Hash: digest(), hmac()y funciones PGP además de métodos md5, la encriptación en la base de datos proporciona seguridad adicional puesto que evita en caso de acceso no autorizado a la base de datos que estos datos puedan ser leidos por el atacante.

2.2.4- Codificación:

La codificación de un sistema, por lo general se realiza por un equipo de trabajo lo que conlleva a que existan problemas típicos de seguridad provocados por el no entendimiento del código, funciones vulnerables o poco conocimiento sobre buenas prácticas para una mejor confección del código.

En la propuesta de solución se expone el enfoque al principal problema de seguridad que tiene el código de una base de datos; tanto PostgreSQL como otros gestores utilizan el estándar SQL; su principal problema de seguridad está dado por las inyecciones SQL, las cuales no son más que una alteración de la sintaxis del código o como suele llamarse también inyección de código, un código SQL no está exento de un ataque y tampoco es invulnerable; aunque el programador sea un experto y tenga experiencia en la codificación, nunca tendrá la certeza de un código invulnerable, dada esta situación es necesario que en la codificación exista una revisión constante, esta propuesta está basada sobre la primicia de que la revisión debe hacerse de forma interna en el proyecto.

Auditoria de código SQL.

Consiste en la revisión del código SQL en busca de vulnerabilidades. Neerumalla[15] propone algunas técnicas para detectar problemas en instrucciones SQL a la hora de revisar el código.

Buenas prácticas.

- ✓ Detectar inyecciones SQL de primer o segundo orden.

[15] Bala Neerumalla es desarrollador de software de seguridad en Microsoft. Es especialista en descubrir posibles vulnerabilidades en los sistemas de seguridad de las aplicaciones.

- ✓ Detectar modificación de SQL por problemas de truncamiento.
- ✓ Detectar inyecciones SQL por problemas de truncamiento.
- ✓ Detectar problemas de inyección SQL.
- ✓ Detectar problemas de truncamiento.
- ✓ Detectar modificación de SQL por problemas de truncamiento. [22]

Si se detecta una inyección SQL se corrige la vulnerabilidad que la permitió.

 **Uso de métodos de pruebas de caja negra.**

Esta recomendación se basa en la existencia de múltiples herramientas capaces de probar aplicaciones por el método de cajas negras y en específico en búsqueda de inyecciones SQL.

Estandarización del código.

La estandarización de código busca la uniformidad de trabajo interno de un proyecto, con el objetivo de facilitarle al equipo de desarrollo la detección de errores de código o vulnerabilidades con mayor facilidad y su mejor corrección asegurando una alta mantenibilidad del código.

El estándar condiciona la comodidad de trabajo del equipo y mucha más facilidad de auditoría para quien esté familiarizado con el código; existen un conjunto de buenas prácticas definidas por Pinal Kumar Dave[16], que se toman de inicio para una elaboración acertada de una propuesta de estándar de codificación.

Buenas prácticas.

Entre las Buenas prácticas presentadas para obtener un estándar de código se propone una guía de codificación descrita por Pinal Kumar Dave entre las que se encuentran:

[16] MVP (Most Valuable Professional) de Microsoft SQL Server. Autor de cientos de artículos de SQL Server. Seis años de experiencia como principal administrador de base de datos en MSQL Server 2008/2005, .NET (C #), y ColdFusion MX. Máster en Ciencias en Redes de Computadores.

- ✓ Utilizar un prefijo para todo un gran grupo de tablas.

  Por ejemplo:

  Page_UserDetails

  Page_Emails

- ✓ Utilizar las siguientes convenciones para nombrar (stored procedures) procedimientos almacenados, triggers, índices, foreign keys y primary keys.

  - ➢ Stored procedures: sp<Nombre de Aplicación> [_<grupo>_] <tipo de accion><nombre de tabla>. Donde acción es un verbo como Get, Delete, Update, Write, Archive, etc. Por ejemplo: spUserAdministration_Page_UpdateEmails
  - ➢ Triggers: TR_<nombre de tabla>_<action><description>. Por ejemplo:
  - ➢ TR_Emails_LogEmailChanges
  - ➢ Indices: IX_<nombre de tabla>_<columnas separadas por _>. Por ejemplo: IX_UserDetails_UserID
  - ➢ Claves primarias: PK_<nombre de tabla>. Por ejemplo: PK_UserDetails
  - ➢ Claves externas: FK_<tabla 1>_<tabla_2> [23]

La propuesta de Dave comprende lo más básico de la codificación y sin embargo no deja de ser importante la necesidad de un código estructurado y bien organizado, para más profundidad en estas buenas prácticas referirse a la Guía de indicadores donde se muestran en toda su magnitud.

Seguridad contra inyecciones SQL.

La inyección SQL es el principal problema de seguridad del código SQL, por ello asegurar el código es de vital importancia; existen varias tecnologías o buenas prácticas para minimizar este riesgo potencial. Los ataques intencionados a bases de datos son a través de las inyecciones SQL, cada día aparece una nueva tecnología de ayuda a los atacantes, por esto es necesario darle la importancia necesaria al código SQL.

A continuación se muestran un conjunto de buenas prácticas que están más ampliadas en la Guía de indicadores, con una serie de casos de estudio que demuestran cómo puede ocurrir una inyección SQL y las vulnerabilidades que utilizan.

Las buenas prácticas son:

- ✓ Utilización de Consultas SQL parametrizadas
- ✓ Conocer que es una consulta no parametrizada o concatenada
- ✓ Delimitar siempre los valores en las consultas
- ✓ Verificar siempre los datos que introduce el usuario [24]

Control y registro de excepciones

Las excepciones pueden contener información específica sobre la aplicación o el origen de datos, puede ayudar a proteger mejor estos últimos elementos exponiendo al cliente solamente la información necesaria. El usuario no tiene por qué saber cómo funciona la aplicación solo se debe mostrar lo que sea estrictamente necesario, esto sirve para toda la aplicación en general.

2.2.5- Herramientas de seguridad para PostgreSQL

El uso de herramientas para elevar la seguridad tiene una marcada importancia para las bases de datos de PostgreSQL, puesto que estas proporcionan funcionalidades que ayudan a perfeccionar los mecanismos de seguridad. En la Guía se sugieren un conjunto de herramientas que ayudan a obtener seguridad en las bases de datos, las más relevantes se muestran en la tabla presentada a continuación:

Tabla 2: Descripción de las herramientas de seguridad.

Nombre de la herramienta: Bacula 

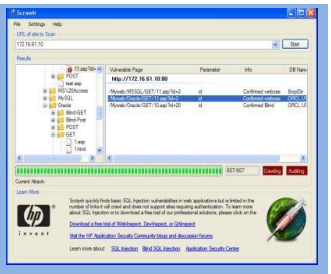
Descripción	Bacula es una amplia colección de herramientas de respaldo, capaz de cubrir eficientemente las necesidades de respaldo de equipos bajo redes IP. Está basada en una arquitectura cliente/servidor que resulta muy eficaz y fácil de manejar.

Funcionalidades Principales:

- ✓ Planificación interna para ejecución automática de tareas.
- ✓ Planificación para múltiples tareas al mismo tiempo.
- ✓ Ejecución de una tarea o múltiples al mismo tiempo
- ✓ Secuencias de tareas usando prioridades.
- ✓ Independencia del sistema operativo en el formato de los volúmenes.

Disponibilidad	Se puede descargar gratuitamente desde: http://www.bacula.org//es

Nombre de la herramienta: Scrawlr



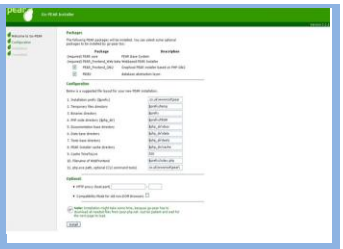
Descripción	Desarrollada por HP, Scrawlr es una herramienta gratuita que revisa sitios web sugeridos en busca de posibles debilidades que puedan facilitar la inyección de SQL.

Funcionalidades Principales:

- ✓ Auditor de código SQL para aplicaciones web.
- ✓ Busca posibles fallas a través del método de prueba de caja negra.

Disponibilidad	Se puede descargar gratuitamente desde: http://www.kriptopolis.org/scrawlr-detecta-sql-injection.

Nombre de la herramienta: PEAR MDB2



Descripción	Proporciona una API común para todos los RDBMS. Se puede utilizar opcionalmente para construir instrucciones SQL portátiles y con más seguridad contra inyecciones SQL.

Funcionalidades Principales:

- ✓ API orientada a objetos.
- ✓ Un DSN (nombre de la fuente de datos de formato) o la matriz para especificar los servidores de base de datos.
- ✓ Tipo de datos y la abstracción en la conversión de tipo de datos de la demanda.
- ✓ Varios modos de buscar para arreglar problemas de portabilidad.
- ✓ Los códigos de error portátil.
- ✓ Capacidad para hacer consultas con buffer y sin buffer.
- ✓ Preparar / execute (enlazar) el nombre y la emulación de marcador de posición sin nombre.
- ✓ Secuencia / autoincrement emulación.
- ✓ Reemplazar la emulación.
- ✓ Fila límite de emulación.
- ✓ Transacciones / punto de retorno de apoyo.
- ✓ Soporte de objeto grande.

Disponibilidad	Se puede descargar gratuitamente desde:

	http://pear.php.net/package/MDB2/download/

Nombre de la herramienta: GreenSQL

Descripción	GreenSQL es un cortafuego diseñado para brindar protección a Bases de Datos frente a ataques del tipo SQL inyección. GreenSQL trabaja en modo proxy interviniendo las conexiones a la base de datos y evaluando los comandos SQL que se envían.

Funcionalidades Principales:

- ✓ Función de Firewall bloqueando inyecciones de comandos utilizados para tareas administrativas de bases de datos.
- ✓ Calcula el riesgo de cada consulta SQL bloqueando las consultas que comprometan la seguridad de la aplicación Web o base de datos.

Disponibilidad	Se puede descargar gratuitamente desde: http://www.greensql.net/download

Nombre de la herramienta: Camouflage

Descripción	Camuflaje permite ocultar archivos de cifrado y luego adjuntarlo. Este archivo camuflado se comporta como un archivo normal, y se puede almacenar, utilizar o enviar por correo electrónico sin llamar la atención.

Funcionalidades Principales:

- ✓ Ocultar archivos dentro de documentos Word u otros tipos de archivos.
- ✓ Establece contraseñas al archivo camuflado.

Disponibilidad	Se puede descargar gratuitamente desde: http://camouflage.unfiction.com/Download.html

Nombre de la herramienta: Kerberos

Descripción	Kerberos es un protocolo de autenticación de red creado por el MIT (Massachusetts Institute of Technology), y utiliza una criptografía de llave simétrica y un tercero, un KDC, para autenticar a los usuarios de servicios de red.

Funcionalidades Principales:

- ✓ Bloqueo de usuarios no autorizados que intenten averiguar las contraseñas monitoreando el tráfico de red (sniffers).
- ✓ El objetivo primario del diseño de Kerberos es eliminar la transmisión de contraseñas encriptadas en la red.

Disponibilidad	Se puede descargar gratuitamente desde: http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/s1-kerberos-works.html.

Nombre de la herramienta: Guardium

Guardium®
SAFEGUARDING DATABASES® | AN IBM® COMPANY

Descripción	La herramienta Guardium provee una solución para garantizar la integridad de la información empresarial y prevenir fugas de información de las bases de datos.

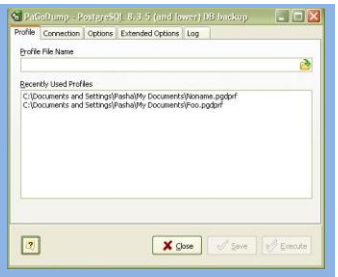
Funcionalidades Principales:

- ✓ Localizar y clasificar información sensible.

- ✓ Evaluar las vulnerabilidades de la base de datos.
- ✓ Asegurar que las configuraciones estén bloqueadas.
- ✓ Proveer 100% de visibilidad y granularidad a todas las transacciones de la base de datos.
- ✓ Supervisar y hacer cumplir las políticas de acceso a datos sensibles.
- ✓ Automatizar todo el proceso de cumplimiento de auditoría.
- ✓ Crear un único y centralizado repositorio de auditoría.
- ✓ Fácilmente escalable.

Disponibilidad	Se puede descargar gratuitamente desde**:** http://www.guardium.com/

Nombre de la herramienta: PaGoDump



Descripción	PaGoDump - PostgreSQL es una utilidad creada para realizar copias de seguridad de una base de datos PostgreSQL.

Funcionalidades Principales:

- ✓ Hace copias de seguridad coherentes, incluso si la base de datos se está utilizando al mismo tiempo.
- ✓ Proporciona una flexible transferencia de archivos.

Disponibilidad	Se puede descargar gratuitamente desde: http://gratis.portalprogramas.com/PaGoDump.html

Nombre de la herramienta: Nessus 	
Descripción	Nessus es una aplicación desarrollada para optimizar las opciones de seguridad en una red o PC, es un escáner remoto de seguridad.
Funcionalidades Principales: ✓ Genera reportes en HTML, XML, LaTeX, y texto ASCII. ✓ Realiza pruebas de seguridad remotas. ✓ Sugiere soluciones para los problemas de seguridad.	
Disponibilidad	Se puede descargar gratuitamente desde: http://www.nessus.org/download/

2.3- Generalidades del Capítulo.

- ✓ Se describe la propuesta de solución y su estructura de forma general.
- ✓ Se resume y expone un número de indicadores que proporcionan cierto nivel de seguridad en bases de datos desarrolladas en PostgreSQL.
- ✓ Se realiza un análisis de la propuesta de solución, basada en la investigación y práctica de un número de autores de nivel internacional.
- ✓ Se muestra la composición de los epígrafes fundamentales de la propuesta de solución, dejando claro por cada epígrafe los principales indicadores y buenas prácticas que lo componen.
- ✓ Se expuso los principales indicadores y buenas prácticas fundamentales por cada epígrafe, con una explicación de los primeros.
- ✓ Se muestra un resumen de las herramientas más utilizadas para ayudar a elevar la seguridad en bases de datos, haciendo referencias de sus características y funcionalidades.

CAPITULO III
Pronóstico sobre los resultados de la Guía de indicadores

Introducción:

Toda nueva propuesta debe pasar un proceso de prueba y afinamiento; los métodos de predicción posibilitan predecir el comportamiento de un evento específico de la propuesta de solución para lo que se utiliza el método Delphy, mediante el criterio de expertos.

3.1- El método Delphy y la evaluación de expertos.

El método Delphy es una técnica de investigación social, tiene como centro la opinión de un conjunto de expertos o especialistas en un campo específico. Olaf Helmer, Nicholas Rescher, Dalkey Normando, y otros en la RAND [17] de Santa Mónica, California, desarrollaron el Método Delphy. Siendo un método de carácter anónimo y con respuestas estadísticas; para la ejecución del método se necesita considerar dos cuestiones importantes:

- ✓ Selección del panel de expertos.
- ✓ Confección del cuestionario.

3.1.1- Proceso de selección de los especialistas.

La selección de especialistas se hace a un grupo de persona con conocimientos, y capaces de ofrecer un criterio, aportar ideas o valoraciones concluyentes sobre el tema a tratar.

Se seleccionan especialistas en el área de:

- ✓ Desarrollo de bases de datos.
- ✓ Desarrollo de bases de datos con PostgreSQL.
- ✓ Seguridad de bases de datos.

[17] Proyecto de Investigación y Desarrollo (RAND) por sus siglas en ingles **R**esearch **and D**evelopment, con más de 1000 investigaciones en la actualidad

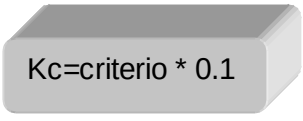
Dadas las áreas de conocimiento se seleccionan un total de 8 especialistas vinculados a la producción y a la docencia dentro de la UCI, donde 4 son profesores de bases de datos, y 4 pertenecen a las líneas de producción de DATECD.

Una de las necesidades del método es conocer las competencias de los especialistas, la cual consiste en el nivel de calificación en la esfera de conocimiento tratada; en algunas ocasiones se tiende a pensar que la competencia está dada por el nivel científico o el cargo que ocupan, sin embargo no siempre esto determina el grado de conocimiento.

Para determinar las competencias de los especialistas se calcula el coeficiente k; que se determina a su vez por el conocimiento o información que tiene el especialista del tema, valorado por el propio especialista representado por el coeficiente kc. Se representa una tabla con un rango del 0-10 donde se define el nivel de conocimiento, y se multiplica por 0.1.

Tabla 3: Autovaloración de conocimiento de los especialistas con respecto al tema.

	Grado de conocimiento o información sobre el tema.								
Especialistas	0	1	2	3	4	5	6	7	8
1								X	
2						X			
3						X			
4					X				
5									X
6						X			
7							X		
8								X	



El resultado para todos los especialistas de kc se muestra a continuación:

Tabla 4: Resultado de kc.

Especialista	1	2	3	4	5	6	7	8
kc	0.7	0.5	0.5	0.4	0.8	0.6	0.7	0.7

Se pasa a calcular Ka coeficiente de argumentación o fundamentación de los criterios del especialista, obtenido como resultado de la suma de los puntos alcanzados a partir de una tabla patrón.

Tabla 5: Tabla patrón para calcular el coeficiente de argumentación

FUENTES DE ARGUMENTACION	Grado de influencia de cada una de las fuentes en sus criterios.		
	A (alto)	M (medio)	B (bajo)
Análisis teóricos realizados por usted	0.3	0.2	0.1
Su experiencia obtenida	0.5	0.4	0.2
Trabajos de autores nacionales	0.05	0.05	0.05
Trabajos de autores extranjeros	0.05	0.05	0.05
Su propio conocimiento del estado del problema en el extranjero	0.05	0.05	0.05
Su intuición	0.05	0.05	0.05

A continuación se muestran los resultados para esta tabla.

Tabla 6: Resultado de la tabla patrón para el coeficiente de argumentación.

FUENTES DE ARGUMENTACION	Grado de influencia de cada una de las fuentes en sus criterios.		
	A (alto)	M (medio)	B (bajo)
Análisis teóricos realizados por usted	4	1,2,3,5,6,8	7
Su experiencia obtenida	1,2,4,7	5,6,8	3
Trabajos de autores nacionales	3,5,6,7	4	1,2,8
Trabajos de autores extranjeros	1,3,5,6,7,8	2,4	
Su propio conocimiento del estado del problema en el extranjero	1	4,6,7,8	2,3,5
Su intuición		3,4,5,6,8	1,2,7

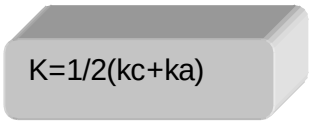
$$Ka=\sum Pa$$

El valor de Pa son los puntos alcanzados por cada especialista.

Tabla 7: Valores del coeficiente de argumentación para cada especialista.

Especialista	1	2	3	4	5	6	7	8	9	10
Ka	0.9	0.9	0.6	1	0.8	0.8	0.8	0.8		

Luego se pasa a calcular los valores de k a través de los coeficientes kc y ka, con la siguiente fórmula.



Donde el coeficiente de competencia se encuentra en el rango $0.25 \leq k \leq 1$.

En la siguiente tabla se muestran los resultados generales para el conjunto de especialistas.

Tabla 8: Resultados generales para el panel de especialistas.

Especialistas	Coeficiente de conocimiento kc	Coeficiente de argumentación Ka	Coeficiente de competencia K	Grado de influencia del coeficiente de competencia
1	0.7	0.9	0.8	Alto
2	0.5	0.9	0.7	Medio
3	0.5	0.6	0.55	Medio
4	0.4	1	0.7	Medio
5	0.8	0.8	0.8	Alto
6	0.6	0.8	0.7	Medio
7	0.7	0.8	0.75	Medio
8	0.7	0.8	0.75	Medio

3.1.2- Proceso de confección del cuestionario.

En esta etapa se confecciona el cuestionario, a través del cual se interactúa con los especialistas donde se exponen sus criterios y valoraciones, el cuestionario se centra en metas o indicadores que debe cumplir la propuesta para que sea válida, se definen un conjunto de preguntas abiertas y cerradas para facilitar la interacción de los especialistas y se expone un resumen de la información que contiene la Guía de indicadores, lo cual permite la comprensión de la solución por parte del panel posibilitando la resolución de la encuesta. Para acceder al cuestionario referirse al anexo 2.

3.2- Análisis de los resultados.

El análisis de los resultados está basado en metas a cumplir por parte de la propuesta de solución, para guiar el análisis se establecen estas metas divididas por preguntas, la siguiente tabla muestra la relación objetivo-pregunta.

Tabla 9: Relación de los objetivos y preguntas.

	Preguntas						
Objetivos	1	2	3	4	5	6	7
1-Importancia de la creación y utilización de la Guía de indicadores.	X						
2- Necesidad del empleo de la Guía.		X			a)		
3- Alcance de la Guía.				X			
4- Complejidad de uso de la Guía.			X		b)		
5-Adaptabilidad a los proyectos.					c)		
6-Eficacia en la utilización de la Guía de indicadores.					d)		
7-Cumplimiento del objetivo de creación.						X	X

Importancia de la propuesta.

Para la valoración de este objetivo se tiene en cuenta la respuesta de los especialistas a la pregunta 1, las respuestas posibles eran: Si, No, No sé.

Como se muestra en el gráfico.

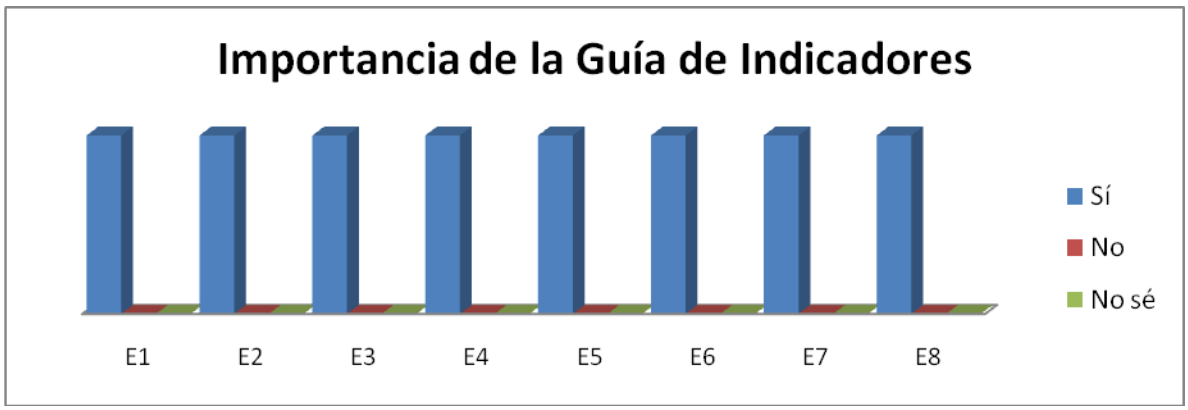


Gráfico 1: Importancia de la Guía de indicadores.

Como se puede observar el 100 % de los especialistas están de acuerdo en que es importante la creación de una Guía de indicadores para elevar la seguridad en bases de datos desarrolladas en PostgreSQL, argumentando que:

- ✓ Da una vista panorámica de los principales aspectos a tener en cuenta para elevar la seguridad a nivel de RDBMS.
- ✓ Muchos proyectos ni siquiera llevan la seguridad de la base de datos, solo llegan hasta la seguridad de software, sin embargo la seguridad de los datos es muy importante.
- ✓ Es de vital importancia la confección de la guía, porque en muchas ocasiones por falta de conocimiento se descuida la protección de la información almacenada en bases de datos y la guía servirá como apoyo a los proyectos de la universidad que lo necesiten.
- ✓ En ocasiones los desarrolladores saben y comprenden la necesidad de implementar la seguridad en las BD, el no contar con una guía de elementos que rigen donde y como se implementa la seguridad, no la implementan o solo lo hacen de manera superficial.
- ✓ Permite a los desarrolladores de bases de datos hacer un análisis más profundo sobre la seguridad, la integridad y robustez de los datos. Además se proporciona más conocimientos sobre conceptos claves de seguridad en bases de datos.
- ✓ Es necesario contar con la guía ya que la mayoría de los casos no se implementa la seguridad por no contar con los pasos necesarios.

Necesidad de uso de la Guía de indicadores.

Las preguntas 2 y 5 inciso a) responden el objetivo, necesidad del empleo de la propuesta, el cual está encaminado a pronosticar si la Guía debe ser aplicada o no; la gráfica muestra las respuestas de los especialista donde se puede apreciar que un 87.5 % está de acuerdo en la necesidad de uso de la Guía.

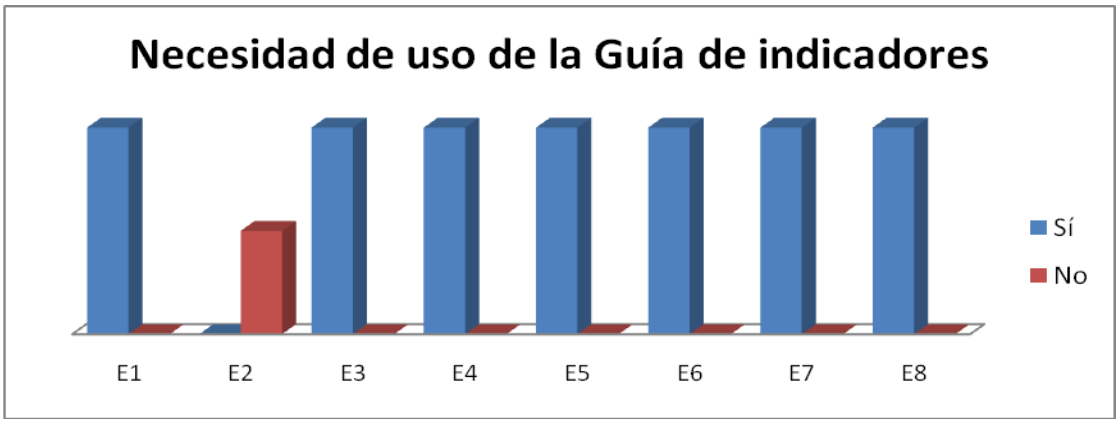


Gráfico 2: Necesidad de uso de la Guía.

Para ampliar la respuesta se exponen los siguientes criterios:

- ✓ Se debe aplicar por el hecho, de que muchos olvidan estos importantes aspectos: seguridad, restricciones, consistencia de los datos, integridad, etc.
- ✓ El hecho de que el desarrollador no conozca los indicadores no quiere decir que deba ignorar la seguridad o tratarla de manera superficial aplicando solo conocimientos muy básicos. No ver el peligro no significa que no esté ahí. Es por eso que debe ser recomendada.
- ✓ No todo los desarrolladores están capacitados en estos temas de seguridad en bases de datos, además es una forma de controlar, más eficiente y organizada la protección de los datos en una base de datos.
- ✓ Es recomendado la aplicación de la seguridad ya que los sistemas puedan ser objetos de ataques o ser vulnerables o fallos.

Alcance de la Guía.

El alcance de la Guía se responde en la pregunta 4 donde los especialistas podían seleccionar un rango del 1 al 5, 1 representaba que la Guía no era abarcadora y 5 que lo era en su totalidad, el resultado se muestra en la gráfica siguiente:

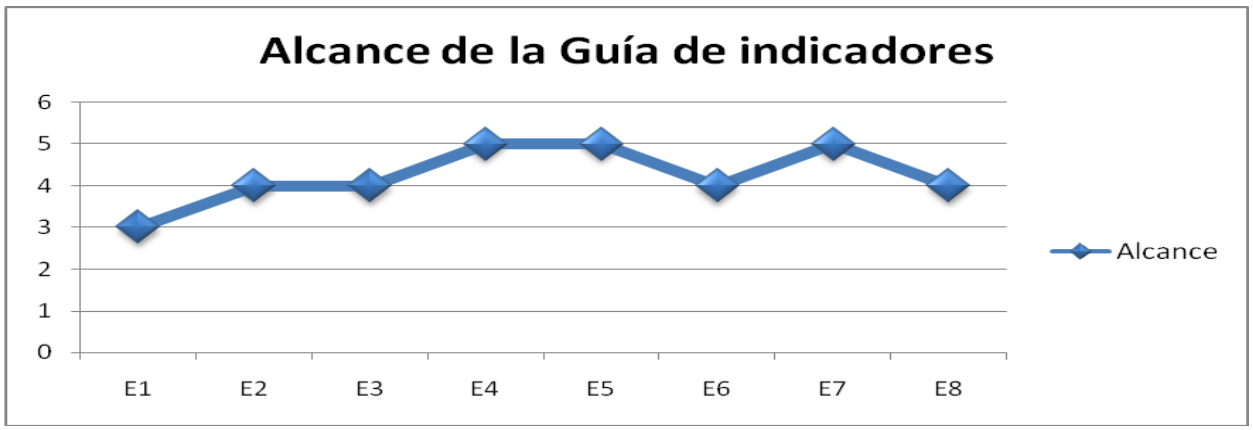


Gráfico 3: Alcance de la Guía de indicadores.

Para este objetivo hubo una concordancia del 87.5 % en que la Guía tiene un alcance alto o es abarcadora, emitiéndose criterios como:

- ✓ Se debe aplicar en un laboratorio de pruebas o en un proyecto real para observar los resultados.
- ✓ Pudieran considerarse elementos como el compromiso del personal responsable de las bases de datos, las características del entorno, etc.

Complejidad de uso de la de la Guía.

La complejidad de la utilización de la propuesta busca pronosticar la dificultad de empleo o entendimiento de la Guía de indicadores; este objetivo se responde en las preguntas 3 y 5 b), representada en el siguiente gráfico.

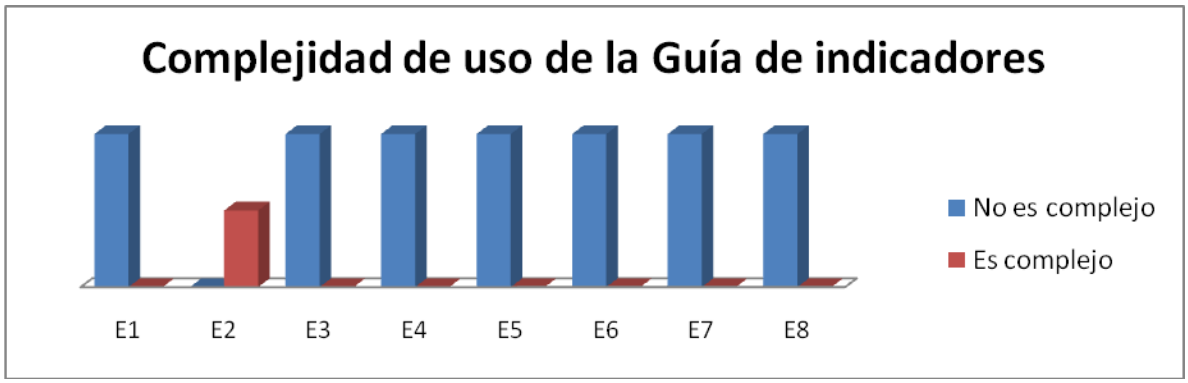


Gráfico 4: Complejidad de uso.

Donde el 87.5 % coincide en que la aplicación de la Guia de indicadores no representa ninguna complegidad. Los especialistas exponen los siguientes criterios:

- ✓ Recomendar tratar la etapa de despliegue la cual es importante en estos aspectos.
- ✓ Se debe detallar paso a paso la guía como aplicar la guía.
- ✓ No existe dificultad alguna, será un material de consulta para los desarrolladores y cada cual utilizará lo que necesite.
- ✓ Aplicar la guía no representa un problema, si los pasos son la bastante descriptivos y detallados como para aplicarlos de manera elemental. El problema reside más bien en la magnitud de la BD, en la concepción que se tenga de la seguridad que requiere y la manera que se vincula con la aplicación de la guía en sí.
- ✓ Esta guía en vez de ocasionar dificultades, facilita el trabajo. Ya que le permitirá al desarrollador trabajar más organizado y más eficiente, garantizando obtener un producto final con mayor calidad.
- ✓ El nivel de detalles de la guía define su complejidad a la hora de aplicarla. Mientras más detallada o mejor elaborada menos complejo es aplicarla.

Adaptabilidad a los proyectos.

La adaptabilidad de la Guía a los proyectos productivos se responde en el inciso c) de la pregunta 5, donde se desea conocer el grado de adaptación que se pronostica para la propuesta de solución, observándose un 100% de concordancia por parte de los especialistas en la adaptabilidad de la Guía de indicadores, representado en la gráfica que aparece a continuación.

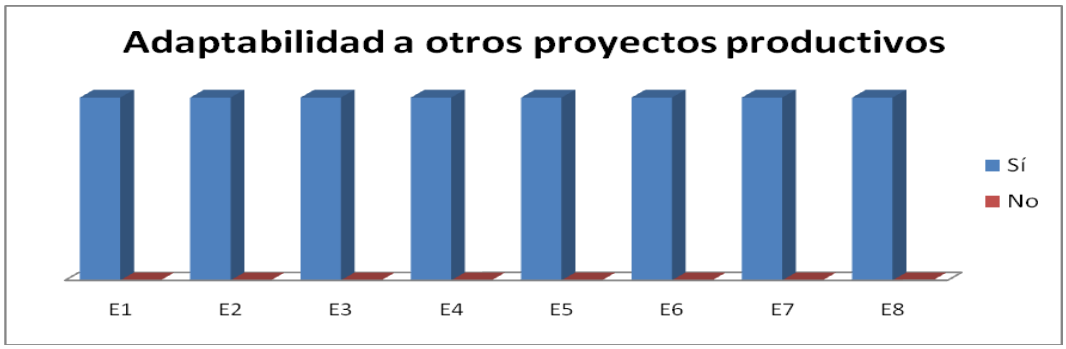


Gráfico 5: Adaptabilidad a otros proyectos.

Eficacia en la utilización de la Guía de indicadores.

La eficacia de creación busca pronosticar si la Guía representa una solución potencial a la problemática de elevar la seguridad en bases de datos, dándosele respuesta en el inciso d) de la pregunta 5, se representan los resultados a continuación:

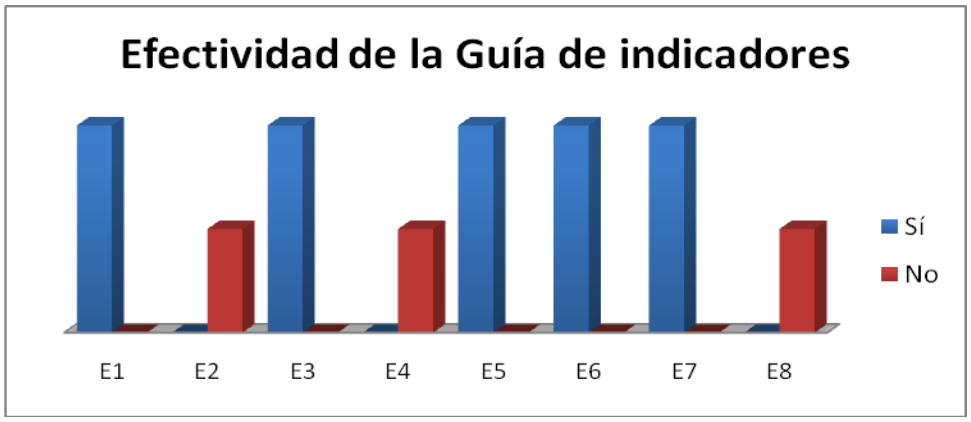


Gráfico 6: Efectividad de la Guía de indicadores.

El resultado muestra la concordancia de 5 especialista de 8 encuestados donde los resultados positivos marcan un 62,5% exponiéndose los siguientes criterios:

- ✓ La Guía está bastante completa y estable. Para los proyectos productivos sería de gran factibilidad.
- ✓ De cierta manera ya sea máxima o mínima se debe implementar la seguridad en las BD y será fácil de hacerse en la medida dependiendo de si se cuenta con una guía para ello y su nivel de detalles. La efectividad de su aplicación se verá representada por el nivel de seguridad.
- ✓ Es necesario e importante que desde la misma etapa de concepción del sistema se tenga en cuenta los aspectos de seguridad.
- ✓ La necesidad está dada ya que posibilita tener un sistema robusto capaz de soportar fallos y es adaptable a varios proyectos, debido a su flexibilidad, la efectividad. La efectividad de su utilización depende del nivel de seriedad con el que se tome.
- ✓ Es necesario que esta guía se emplee y se adapte a los proyectos productivos, teniendo en cuenta que nos resolverá varios problemas presentes hoy con la seguridad en base de datos. La guía será fácil de usar ya que cuenta con una estructura bien definida.

Cumplimiento del objetivo de creación de la Guía de indicadores.

Este objetivo describe si realmente la solución resolvería la problemática de elevar la seguridad en las bases de datos donde se encuestan a los especialistas en las preguntas 6 y 7, los resultados se muestran a continuación:

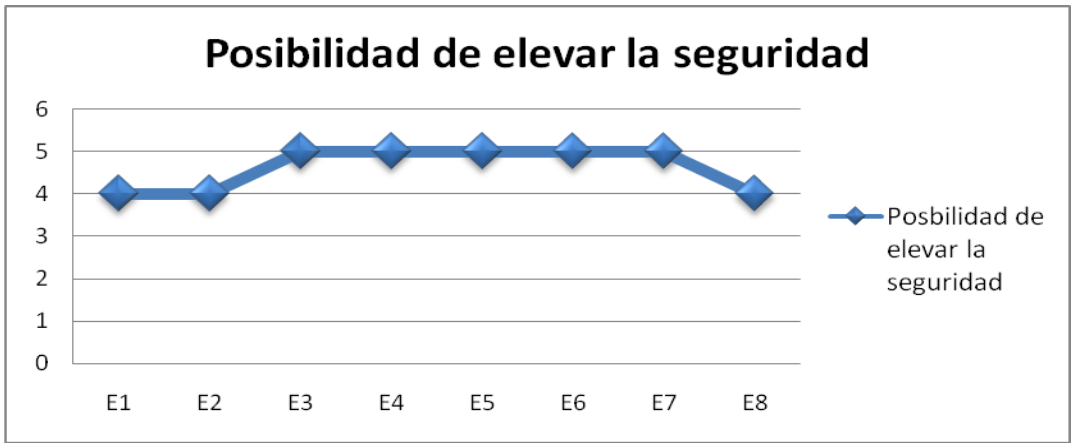


Gráfico 7: Posibilidad de elevar la seguridad.

La gráfica muestra que el 100 % de los especialistas plantea que la Guia de indicadores eleva la seguridad en las bases de datos o ayuda a elevar mucho la seguridad, exponiéndose los siguientes criterios:

- ✓ La guía cumple con el objetivo por el hecho de que los indicadores si están centrados en esos aspectos, por lo que al final de su aplicación se puede determinar que la aplicación o base de datos es medianamente segura.
- ✓ La guía cumple un 80% solo agregando las recomendaciones, creo que está completa.
- ✓ Dado que el objetivo es elevar la seguridad de las BD y que la guía provee una gama de indicadores, su correcta aplicación contribuye a elevar el nivel de seguridad de las BD.
- ✓ Cumple con el objetivo de elevar la seguridad si se aplica de la manera correcta.
- ✓ Depende de la configuración que se realice y el modo en que esté concebido el diseño.

Fase final del método deplphy.

El grado de aceptación de la Guía de indicadores por parte de los especialistas fue alta así como el porcentaje de respuestas fue positivo, destacando la importancia de la creación de la Guía de

indicadores, en el caso de la eficacia el resultado está dado en que algunos especialistas tuvieron en cuenta factores ajenos a la Guía de indicadores para definir la eficacia aunque el objetivo de creación de la Guía demuestra que cumple con el propósito de ayudar a elevar la seguridad en bases de datos desarrolladas en PostgreSQL. El grado de concordancia de los especialistas a todas las preguntas se calculó mediante el coeficiente de concordancia de Kendall utilizando el software SPSS 13.0 con un resultado de 0.914 con un valor cercano a 1.00, lo que demuestra un alto grado de concordancia por parte de los especialistas, representados a continuación en el siguiente gráfico:

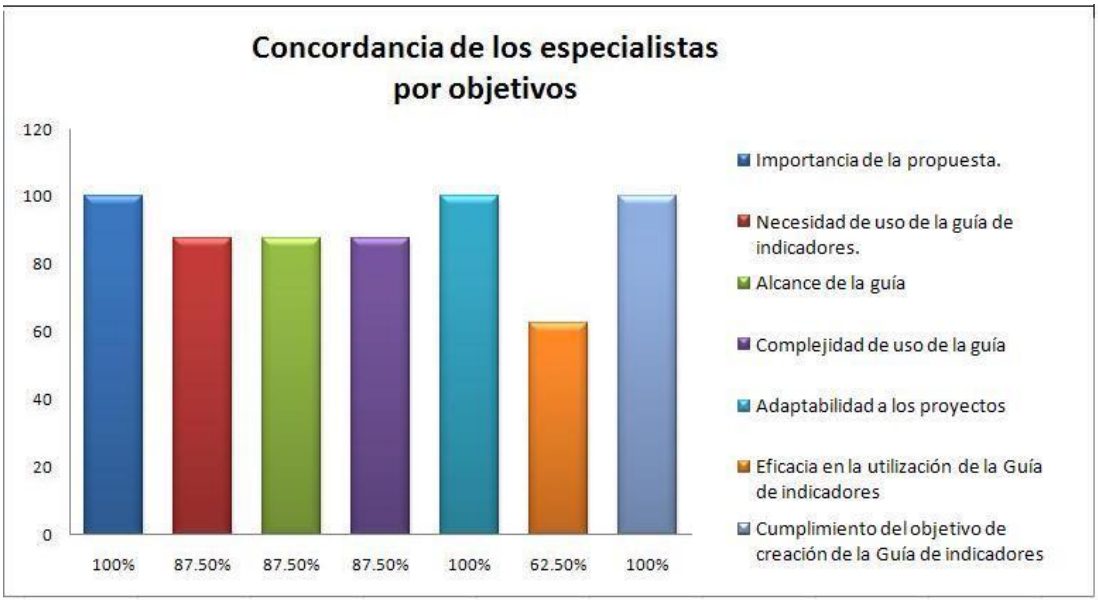


Gráfico 8: Concordancia de los especialistas por objetivos.

3.3- Generalidades del Capitulo.

- ✓ Se realiza un pronóstico de los posibles resultados de la propuesta de solución a través del método Delphy.
- ✓ Se tabularon las respuestas llegándose a la conclusión de que la propuesta de solución tiene un gran impacto positivo en el desarrollo de bases de datos.
- ✓ Algunas de las recomendaciones aportadas fueron:

  Profundizar en aspectos como seguridad a nivel de columnas, vistas y la etapa de despliegue.

  Profundizar en otros mecanismos que permitan elevar la seguridad en bases de datos desarrolladas en PostgreSQL.

Conclusiones Generales

- ✓ Se realizó un estudio del gestor PostgreSQL en el cual se constató su funcionamiento y principales características.
- ✓ Durante el estudio de los principales problemas de seguridad en bases de datos se identificaron una serie de pasos que permiten elevar la seguridad en bases de datos realizadas en PostgreSQL.
- ✓ Se definieron los procesos de desarrollo de bases de datos: arquitectura, configuración, diseño y codificación.
- ✓ Se confeccionó una Guía de indicadores enfocada a elevar la seguridad en los cuatro procesos de desarrollo de una base de datos: arquitectura, diseño, configuración y codificación.
- ✓ Se realizó un resumen de las herramientas más utilizadas para ayudar a elevar la seguridad en bases de datos, haciendo referencias de sus características y funcionalidades.
- ✓ Se realizó un pronóstico para comprobar la veracidad de la Guía y el impacto de la misma en el proceso de desarrollo de software.
- ✓ Se tabularon las respuestas llegándose a la conclusión de que la propuesta de solución tiene un gran impacto positivo en el desarrollo de bases de datos.

Recomendaciones.

- ✓ Aplicar la Guía de indicadores.
- ✓ Profundizar en los indicadores y sus respectivos pasos para crear una metodología.
- ✓ Tener en cuenta y profundizar en la etapa de despliegue.
- ✓ Agregar indicadores para la seguridad física de la base de datos llevando a cabo la seguridad a nivel de File System (sistema de ficheros), así como velar también por la seguridad del entorno, incluyendo por supuesto a los responsables de la BD.

REFERENCIAS BIBLIOGRÁFICAS

1- MANUNTA, D. G., *Seguridad: Una introducción.* [En línea], Madrid: Belt Ibérica, 2004, [noviembre, 2009], Disponible en: <http://www.belt.es/bibliografia/articulo.asp?id=130>

2- PAULUS, Nelson, *Del concepto de riesgo: Conceptualización del riesgo en Luhmann y Beck,* [electronico], Departamento de antropología de Chile, Mayo 2004, [noviembre, 2009], Disponible en: <http://www.revistamad.uchile.cl/10/paper07.pdf>

3- Teleformación, G. d.[Electrónico], noviembre, 2009, *Entorno Virtual de Aprendizaje,* [noviembre 2, 2009], Disponible en: http://eva.uci.cu/file.php/237/Introduccion/1_Conferencia_1_V2.0.pdf

4- RIVERO PINO, Noel Jesús, A. M, *Sistema de Gestión de Seguridad (SIGIS)*, [En línea], Ciudad de la Habana 2009, [noviembre 2009], Disponible en: < http://bibliodoc.uci.cu/TD/TD_2044_09.pdf >

5- Consultoría ISO 27001/UNE 71502, [En línea], [noviembre, 2009], Disponible en: <http://www.isecauditors.com/es/consultoria-iso-17799-iso-27001-une-71502.html>

6- PÉREZ VALDÉS, Damián**,** *¿Qué son las bases de datos?,* [En línea], publicado: octubre 26, 2007, [febrero 3, 2010], Disponible en: < http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>

7- ANZALDO, Juan**.** *Breve Historia de Bases de Datos, [En línea],* Diciembre,6 2005, [febrero, 2010], Disponible en: <http://janzaldo.wordpress.com/2005/12/06/breve-historia-de-las-bases-de-datos/>

8- Portal en español sobre PostgreSQL, *Historia,* [En línea], publicado: marzo 22, 2009, [febrero 9, 2010], Disponible en: <http://www.postgresql-es.org/sobre_postgresql#historia>

9- Proveedor de Tecnologías de información, *Sistema gestor de bases de datos* PostgreSQL, [En línea], publicado: 2007, [febrero 9, 2010.], Disponible en: <http://www.http-peru.com/postgresql.php>

10- ESPINOZA, Humberto, *PostgreSQL Una Alternativa de DBMS Open Source.* Publicado: 2005, Disponible en: <http://www.lgs.com.ve/pres/PresentacionES_PSQL.pdf>

11- GONZÁLEZ, Carlos D, *Curso Base de Datos PostgreSQL, SQL avanzado y PHP,* [En línea], actualizado marzo 2010, [febrero, 2010], Disponible en: <http://www.usabilidadweb.com.ar/postgre.php>

12- LOCKHART, Thomas, *Manual del usuario de PostgreSQL,* [En línea], 2006, [marzo, 18, 2010], Disponible en: <http://www.postgresql.org/docs/8.2/interactive/index.html>

13- CASANOVA, Jaime, *Nuevas actualizaciones de seguridad de PostgreSQL*, [Electrónico], Guayaquil-Ecuador, año 2008, [marzo, 18, 2010].

14- PostgreSQL. *Lista de Características de PostgreSQL 8.3.* [Electrónico] 2009. [febrero 12, 2010]

15- *Concepto Seguridad Base Datos,* [En línea]*, publicado:* 2009, [noviembre 2 2009], Disponible en: <http://www.mitecnologico.com/Main/ConceptoSeguridadBaseDatos>

16- GÓMEZ PÉREZ, Erich Mario, Torres Gálvez, Ariel, *Administración y optimización de un Sistema de Bases de Datos Descentralizado, en PostgreSQL,* [En línea], Ciudad Habana mayo, 2008, [febrero 2010], Disponible en: <http://bibliodoc.uci.cu/TD/TD_1358_08.pdf >

17- GÓMEZ PÉREZ, Erich Mario, Torres Gálvez, Ariel, *Administración y optimización de un Sistema de Bases de Datos Descentralizado, en PostgreSQL,* [En línea], Ciudad Habana mayo, 2008, [febrero 2010], Disponible en: <http://bibliodoc.uci.cu/TD/TD_1358_08.pdf >

18- FÁBREGAS, Yuniesky, Fernández, Daniel**,** *Administración, configuración y optimización de un Sistema de Bases de Datos Descentralizado, en Oracle Database 10g,* [En línea], Ciudad Habana 2007, [febrero 2010], Disponible en: <http://bibliodoc.uci.cu/TD/TD_0289_07.pdf>

19- Manual de PostgreSQL 8.2.15 Chapter 17. Server Configuration, [En línea], Disponible en: <http://www.postgresql.org/docs/8.2/interactive/runtime-config.html>

20- RODRÍGUEZ, Luis, *Seguridad en bases de datos: Ficciones y fricciones*, [En línea], Consultor de ALS, Publicado en Revista SIC, Noviembre 2006, [4 marzo, 2010], Disponible en: < http://www.alses.com/home.php?location=recursos/articulos/seguridad-en-bases-de-datos >

21- RODRÍGUEZ, Luis, *Seguridad en bases de datos: Ficciones y fricciones*, [En línea], Consultor de ALS, Publicado en Revista SIC, Noviembre 2006, [4 marzo, 2010], Disponible en: <http://www.als-es.com/home.php?location=recursos/articulos/seguridad-en-bases-de-datos>

22- NEERUMALLA, Bala, *Seguridad en SQL. Nuevos ataques de truncamiento SQL y cómo evitarlos*, [En línea], 2006, [3 marzo, 2010], Disponible en:< http://msdn.microsoft.com/es-es/magazine/cc163523.aspx >

23- KUMAR, DAVE, Pinal, *Guidelines and Coding Standards*, [En línea], publicado por: Journey to SQL Authority with Pinal Dave, Septiembre 24, 2008, [10 noviembre, 2009], Disponible en: < http://blog.sqlauthority.com/author/pinaldave/ >

24- LLINÁS, MARTÍNEZ, Juan, *Inyección de código SQL*, [En línea], publicado en Epistemowikia Cala, Universidad de Extremadura 2009, [12 febrero, 2010], Disponible en: <http://campusvirtual.unex.es/cala>

25- Rodríguez, Luis, *Seguridad en bases de datos: Ficciones y fricciones*, [En línea], Consultor de ALS, Publicado en Revista SIC, Noviembre 2006, [4 marzo, 2010], Disponible en: <http://www.als-es.com/home.php?location=recursos/articulos/seguridad-en-bases-de-datos>

26- SÁNCHEZ, Jorge, *Diseño Conceptual de Bases de Datos*, [En línea], 2004, [8 marzo, 2010], Disponible en: < http://www.jorgesanchez.net >

27- PÉREZ, SANTIAGO, Fernandeze, Néstor, *Apoyo para la toma de decisiones,* [electrónico], Cátedra: Gestión de Datos**,** 2006, [marzo, 2010].

28- CORDINA, Lluis, *Metodología de diseño lógico en el modelo relacional,* [En línea], 12 febrero, 2001, [marzo, 2010], Disponible en: < http://www3.uji.es/~mmarques/f47/apun/node89.html >

29- GONZÁLEZ, MARTÍNEZ, Eduardo, NUÑEZ, SANDOVAL, Alejandro, *Implantación de Bases de Datos Seguras en PotgreSQL V 8.2.6*,[En línea], Departamento de Seguridad en Computo/UNAM-CERT,15 de Febrero de 2008 [febrero 2010], Disponible en:< http://www.seguridad.unam.mx/doc/?ap=tutorial&id=204**>**

30- VALLEJO, RODRÍGUEZ, *David, Optimizando el acceso a BBDD con un pool de conexiones*, [Electrónico], Consultor de ALS, 2007, [18 de marzo, 2010].

BIBLIOGRAFÍA:

- Portal en español sobre PostgreSQL, *Demasiadas soluciones de replicación incompletas,* [En línea], publicado: noviembre 20, 2009, [febrero 9, 2010], Disponible en: <http://www.postgresql-es.org/node/381>

- GONZÁLEZ, HERNÁNDEZ, Yanisbel, Medina, Rodríguez, Mabel, *Metodología para soluciones de integración y análisis de datos,* [Electrónico], Universidad de las Ciencias Informáticas, [abril, 2010].

- RODRÍGUEZ, Luis, *Seguridad en bases de datos: Ficciones y fricciones*, [En línea], Consultor de ALS, Publicado en Revista SIC, Noviembre 2006, [4 marzo, 2010], Disponible en: < http://www.alses.com/home.php?location=recursos/articulos/seguridad-en-bases-de-datos >

- RODRÍGUEZ, Luis, *Seguridad en bases de datos: Ficciones y fricciones*, [En línea], Consultor de ALS, Publicado en Revista SIC, Noviembre 2006, [4 marzo, 2010], Disponible en: <http://www.als-es.com/home.php?location=recursos/articulos/seguridad-en-bases-de-datos>

- C.J.Date, *Introducción a los Sistemas de Bases de Datos,*[Electrónico], Editorial Félix Varela, La Habana, 2003, [marzo, 2010]

- HERNÁNDEZ, LEÓN, Rolando Alfredo, Coello, González, Sayda. EL PARADIGMA CUANTITATIVO DE LA INVESTIGACION CIENTIFICA. Ciudad de la Habana: s.n., [En línea], noviembre del 2002, [noviembre.2009], Disponible en: <http://octavitos.uci.cu/descargas/5to/Libro_I_Rolando.pdf?option=com_content&task=view&id=894&Itemid=191>

- OSORIO, Víctor Larios, *¿Cómo hacer una encuesta?*, [En línea] Depto. de Matemáticas de la Fac. de Ingeniería de la U.A.Q. (México) : s.n, noviembre, 2009], Disponible en: <http://www.rrppnet.com.ar/comohacerunaencuesta.htm.>

- MATO GARCÍA, Rosa María, *Diseño de Bases de Datos,* [Electrónico], octubre 1999, [enero, 2010]

- JEROME, C. Glenn**,** *Futures Research Methodology,* [Electrónico], American Council for the United Nations University, Washington, USA, 1999, [abril, 2010].

- KUROKI, Christian, *Migración a PostgreSQL desde otras bases de datos,* [En línea], publicado: 2005, [febrero 2010], Disponible en: <http://www.dbrunas.com.ar/postgres/migrapg.pdf>

- VILLALÓN, HUERTA, Antonio. *Gestión de la seguridad de la información: UNE 71502, ISO 17799*, [En línea], publicado: Junio 2004, [noviembre 2009], Disponible en: <http://www.s2grupo.es/website/s2grupo/seguridad_informatica/descargas/parrafo/06/document/campusti.pdf>

- ALONSO, Lamí, Leiny Evelin, Aragón, Pérez, Yanetsy, *Sistema para gestionar las lecciones aprendidas en proyectos productivos*, [Electrónico], Ciudad de La Habana, Junio 2008, [febrero, 2010].

ANEXOS

Anexo: 1 Lista de proyectos que usan PostgreSQL en la Universidad de las Ciencias Informáticas.

1. Sistema de Gestión Fiscal (SGF).
2. Sistema Nacional Público para el Seguimiento de Inversiones y Sectores (SINAPSIS).
3. ERP Cuba (sus siglas en inglés Planificación de Recursos Empresariales).
4. Tribunales Populares Cubanos.
5. Filtro de contenido web (FCWEB) con su producto FILPACON
6. Gestión Universitaria de la Dirección de Informatización.
7. Sistema de Video Vigilancia (SURIA)
8. Comités Militares
9. Centro de inmunología molecular (CIM)
10. IPTV (Televisión por IP)
11. KAINOS
12. Señal ACN (Video y Sonido Digital)
13. RAP(Registro de Antecedentes Penales)
14. Convenio Cuba Venezuela (CCV)
15. Frente de Proyectos.
16. Atención Primaria de Salud (ASP)

Anexo: 2 Cuestionario

CRITERIO DE EXPERTO

La presente encuesta tiene como finalidad la evaluación, de la Guía de indicadores para elevar la seguridad en bases de datos creadas en PostgreSQL, es necesario que responda todas las preguntas; la encuesta tiene carácter anónimo. Gracias por su colaboración.

Categoría docente: _____

Cargo que ocupa:_____

Años de experiencia en la materia:_____

Nombre de la Tesis: Indicadores que determinan la seguridad de las bases de datos realizadas en PostgreSQL en cuanto a la configuración, diseño, arquitectura y codificación.

Autores: Niurisleidy Reyes Piloto
Raúl Cambar Martínez
Tutores: Ing. Yusleydi Fernández del Monte
Ing. Sonia Guerrero Lambert
MsC. Michael González Jorrín

Introducción

Valore su conocimiento en el campo de la seguridad en bases de datos marque con una X en una escala del 0 al 10

0	1	2	3	4	5	6	7	8	9	10

Marque con una X el grado que usted crea que han influenciado las siguientes fuentes de conocimiento.

FUENTES DE ARGUMENTACION	Grado de influencia de cada una de las fuentes en sus criterios.		
	A (alto)	M (medio)	B (bajo)
Análisis teóricos realizados por usted			
Su experiencia obtenida			
Trabajos de autores nacionales			
Trabajos de autores extranjeros			
Su propio conocimiento del estado del problema en el extranjero			
Su intuición			

Resumen de la propuesta de solución de la Guía de indicadores para elevar la seguridad en bases de datos desarrolladas en PostgreSQL.

El desarrollo de bases de datos en la Universidad de las Ciencias Informáticas está presente en todos los proyectos, y cada día es más utilizado el gestor PostgreSQL, los desarrolladores de estas bases de datos tratan la seguridad de forma superficial y en muchos casos; por falta de tiempo, por desconocimiento o porque no se le da la importancia que llevan ciertos pasos dentro de la construcción la seguridad es obviada; esto conlleva a que existan problemas de seguridad en las bases de datos, por ello se hace necesario la confección de una solución que garantice elevar la seguridad en las bases de datos desarrolladas en PostgreSQL.

La solución es una Guía de indicadores de los factores elevadores de la seguridad en bases de datos desarrolladas en PostgreSQL en la: Arquitectura, Configuración, Diseño y Codificación; esta proporciona una indicación de los elementos que posibilitan al desarrollador obtener una elevada seguridad en estas bases de datos. Desarrollada a partir de buenas prácticas que proponen usuarios de experiencia y autores reconocidos en el trabajo con bases de datos en especial de PostgreSQL, además tiene su base en los manuales de usuarios que proponen los desarrolladores de PostgreSQL, y fundamentos y metodologías asentadas en años de explotación para el diseño y desarrollo de bases de datos.

El objetivo de la Guía de indicadores es exponer de forma sintetizada un conjunto de indicadores que permiten elevar la seguridad en las bases de datos. Además de ilustrar casos de estudio de manera tal que el usuario comprenda la necesidad de la aplicación de estos indicadores.

Los **indicadores** no son más que pasos que muestran, qué se debe hacer para elevar la seguridad en las bases de datos, en muchos casos son pasos simples que definen una acción en particular, otros indicadores tienden a ser más complejos y generales dándole camino a una serie de pasos más concretos; para dar solución a los indicadores; definidos como buenas prácticas.

La composición de la propuesta de solución está dada por epígrafes los cuales representan los factores de desarrollo de una base de datos, cada epígrafe tiene una constitución similar dada por los indicadores, buenas prácticas y recomendaciones. Las buenas prácticas darán solución a los indicadores más importantes o complejos.

Cada epígrafe trata un elemento constructivo de la base de datos por cada uno se muestran los indicadores principales:

En la Arquitectura entre los indicadores más importantes se encuentran los siguientes:

Independencia física de los datos. Que es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos).

Protección física de los datos, externa al SGBD: Debe existir un mecanismo de protección exterior a los SGBD, el almacenamiento físico hace que los datos puedan ser accesibles si se tiene acceso al dispositivo de almacenamiento por lo que es importante un sistema que garantice que no se acceda a estos dispositivos sin autorización.

En la codificación se presentan:

Auditoria de código SQL: Consiste en la revisión del código SQL en busca de vulnerabilidades.

Estandarización del código: Crear un estándar tal que el equipo de desarrollo se sienta identificado y se haga más sencillo la detección de problemas de código o posibles vulnerabilidades.

Para dar solución a estos indicadores por su complejidad es necesario poner en práctica ciertos pasos, definidos en las buenas prácticas que están enmarcadas dentro de los indicadores; por ejemplo para dar solución o correcto cumplimiento al indicador **Auditoria de código SQL** debe trabajarse sobre la base de la: **Detección de inyecciones SQL de primer o de segundo orden, Detección de problemas de inyección SQL;** definidas como buenas prácticas, siendo las más importantes.

En el diseño se dan un número mayor de indicadores ya que es este mucho más complejo por la cantidad de pasos que debe tenerse en cuenta entre los más importantes están:

Las restricciones de integridad en general: Se debe tener en cuenta las restricciones de los datos para la base o sea validar el tipo de dato que se desea en específico, sí se define un tipo de dato como carácter debe validarse que se entre por parámetro un carácter; para las bases de datos de apoyo a la toma de decisiones que son principalmente de sólo lectura, la integridad de los datos se verifica al cargar (o actualizar) la base de datos. Por lo que declarar las restricciones de los datos ayuda a los usuarios a la hora de formular las consultas, además que proporciona información adicional al optimizador. En la Guía se trata; sobre todo en el diseño; un enfoque especial a las bases de apoyo a la toma de decisiones, dada sus características particulares.

En la configuración hay indicadores tales como **Registro de auditoría y Configuración de copias de seguridad (Backups);** además de exponerse la necesidad de una correcta configuración para elevar la seguridad de las bases de datos.

En los diferentes epígrafes se recomienda bibliografía complementaria para la profundización del conocimiento además de recomendaciones de acciones y herramientas para ayudar a elevar la seguridad dejando un apartado especial para estas.

Está dirigida fundamentalmente a desarrolladores de bases de datos, administradores y clientes o cualquier usuario con conocimientos medios en el desarrollo de bases de datos. Puede aplicarse en cualquier proyecto en el cual se utilice el gestor PostgreSQL.

Teniendo en cuenta el resumen anterior responda las siguientes preguntas.

1- Cree usted que es de gran importancia la creación y utilización de la Guía de indicadores, para ayudar a los desarrolladores de bases de datos a elevar la seguridad en las bases de datos que construyen.

____Si ____No ____No sé

¿Por qué?

_____
_____
_____
_____
_____
_____

2- Mencione usted si existen razones o cuestiones en el trabajo de los desarrolladores de bases de datos por las cuales el uso de la Guía de indicadores.

——Se debe recomendar ——No se debe recomendar ——Se recomienda una parte

¿Por qué?

_____
_____
_____
_____
_____
_____

3- Ve usted dificultades o alguna complejidad en el proceso de aplicación de la Guía para el desarrollador de bases de datos.

——Si ——No

¿Por qué?

_____
_____
_____
_____
_____
_____

4- Cree usted que es abarcadora la Guía de indicadores. En una escala del 1 al 5 indique que tan abarcadora es, teniendo en cuenta que1 quiere decir que la Guía no es abarcadora y 5 que lo abarca todo, en caso de su respuesta ser diferente de 5, diga las razones, por las que la Guía está incompleta.

1	2	3	4	5

_____
_____
_____
_____
_____
_____

5- Diga si la Guía de indicadores está facultada para cumplir las siguientes expectativas, para ello marque con una x las que usted cree que pueda cumplir.

——a) Necesidad del empleo de la propuesta.

——b) Fácil de usar.

——c) Adaptabilidad a proyectos productivos.

____d) Efectividad de su utilización.

Escriba el por qué de su selección:

_____
_____
_____
_____
_____
_____

6- Cree que la utilización de la Guía cumple con el objetivo para la cual fue creada (ayudar a elevar la seguridad en bases de datos). Indíquelo en una escala del 1 al 5, teniendo en cuenta que 1 quiere decir que la Guía no cumple el objetivo y 5 que lo cumple, en caso de su respuesta ser diferente de 5, diga las razones, por las que la Guía no cumple el objetivo.

1	2	3	4	5

_____
_____
_____
_____
_____
_____

7- ¿Cree que los indicadores posibilitan que en el área donde se aplican se eleve la seguridad?

Si____ No_____ En cierta medida_____

¿Por qué?

_____
_____
_____
_____
_____
_____

GLOSARIO DE TÉRMINOS

ADO: en inglés ActiveX Data Object, componentes de acceso a Datos.

ANSI/SPARC: literalmente, ANSI/ Comité de Planeación y Requerimientos de Sistemas; se usa para referirse a la arquitectura de sistemas de DB de tres niveles.

API: interfaz de programación de aplicaciones.

CODASYL: literalmente, Conferencia sobre Lenguajes de sistemas de Datos; se usa para referirse a determinados sistemas prerrelacionales (de red) tales como IDMS.

CPU: unidad central de procesamiento.

DB: Base de datos.

DBA: administrador de base de datos.

DBMS: sistema de administración de bases de datos.

E/S: entrada/salida.

ER: entidad/relación.

FRUPS: factores de calidad**,** Functionality, Reliability, Usability, Performance, and Supportability.

IDMS: Sistema Integrado de Administración de Bases de Datos.

IP: El protocolo de Internet (**Internet Protocol**). Es un protocolo enrutable responsable del direccionamiento **IP** y de la fragmentación y ensamblado de los paquetes, no confiable y sin conexión.

ISO: Organización Internacional de Estándares.

LOOP: También denominado lazo o ciclo. Es una estructura de control que permite la repetición de una serie determinada de sentencias.

OLAP: procesamiento analítico en línea.

OLTP: procesamiento de transacciones en línea.

RDBMS: DBMS relacional.

RM/T: modelo relacional Tasmania.

SGBD: Sistema Gestor de Base de Datos

SQL: originalmente, Lenguaje Estructurado de Consultas.

SSL: en inglés Secure Socket Layers, estándar de encriptación de datos a través de la red.