Universidad de las Ciencias Informáticas

Facultad 15

Título: Desarrollo de Plug-ins de soporte para la herramienta AuditBD.

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

Autores: Danay Betancourt Quintanal

Dayron Abreus Ruiz

Tutor: Ing.Taymí Soledad Peña Quesada.

Co-Tutor: Ing. Pedro E. Castiñeiras Sánchez

Junio 2010

Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido...

Ché

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Danay Betancourt Quintanal
_____
Firma del Autor

Dayron Abreus Ruiz
_____
Firma del Autor

Taymí Soledad Peña Quesada
_____
Firma del Tutor

Pedro E. Castiñeiras Sánchez
_____
Firma del Co-Tutor

Agradecimientos

A la Universidad de las Ciencias Informáticas, por habernos formado como profesionales a la altura de estos tiempos.

A nuestra tutora y co-tutor (Taymí y Pedro) por la confianza que depositaron en nosotros y su ayuda para sacar adelante este trabajo.

DANAY:

A mis padres (Maribel e Iván) que con infinito amor y consagración han sabido educarme y mostrarme el camino correcto para alcanzar mis sueños. Para ellos todo el amor del mundo y espero estén orgullosos de mí.

A mis hermanitos (Daniela, Lázaro Daniel y Osvany) por servirme de motor impulsor, cuando la tristeza y la nostalgia se adueñaban de mí.

A mi segundo papá Viky, por darme siempre su apoyo.

A mi tía Nerta por ser como una madre para mí en estos 5 años.

A mi tío Isidro por servirme de ejemplo ante la superación, consagración y por apoyarme en todo momento.

A mi maravillosa familia que me ha dado todo su amor y ha confiado siempre en mí.

A mi compañero de tesis Dayron por el empeño y dedicación que pone en todo lo que hace, por su inmensa paciencia conmigo y por haberse convertido en el amigo que hoy es.

A todas las amistades que he cultivado durante estos 5 años, especialmente a mis amigas de toda la vida Marlen, Eileen, Nivia, Aleidy, las Daineris, Yailen, Nallelys, Yaily, Marlen F y Yuly, a todas ellas, todo el amor del mundo y mi eterno agradecimiento por ser las amigas incondicionales que son.

En fin gracias a todos los que de una forma u otra contribuyeron a mi formación profesional así como al desarrollo de este trabajo.

DAYRON:

Quiero agradecer primeramente al Comandante en Jefe por su idea de una escuela tan dinámica.

A mis padres y mi hermano por siempre estar junto a mí.

A los profesores de toda mi vida de estudiantes que me han legado parte de sus conocimientos.

A todos los que me han ayudado de una forma u otra a atravesar esta etapa de la vida tan hermosa y agitada.

A mi compañera de tesis por todos los trabajos compartidos.

Dedicatoria

DANAY

Dedico este trabajo a la persona que ha sido capaz de darlo todo sin recibir nada. De quererme con todo su corazón sin esperar nada a cambio. De invertir todo en mí sin medir la rentabilidad que le aporte su inversión. A ti que has mantenido la confianza en mí cuando todos los demás la han perdido y que has sabido darme el amor, el cariño y la educación que necesitaba para transitar por la vida como una persona de bien. A ti dedico este trabajo mamita.

Gracias por ser mi madre.

A mi papá por el inmenso amor que siempre me ha dado.

DAYRON:

A mi abuela.

A mi madre y a mi padre.

A mi hermano menor.

Resumen

Actualmente en la Universidad de Ciencias Informáticas se desarrollan una considerable cantidad de proyectos productivos, donde se utiliza el sistema gestor de bases de datos PostgreSQL.

Debido a la importancia que constituyen las bases de datos en las aplicaciones informáticas así como la necesidad de contar con un software capaz de verificar la seguridad de la información, específicamente, los datos manejados por el sistema gestor de bases de datos PostgreSQL y la carencia de la misma, se decide vincular el trabajo de diploma al proyecto desarrollado por el laboratorio de seguridad de la facultad 2 de dicha universidad, encargado de desarrollar la aplicación informática AuditBD, cuyo objetivo fundamental es realizar auditorías a diferentes sistemas gestores de bases de datos.

Debido a que AuditBD no brinda soporte para PostgreSQL, se plantea como objetivo de este trabajo desarrollar los plug-ins que le faciliten esta opción.

Basado es un profundo estudio sobre la seguridad de los datos, especialmente los manejados por PostgreSQL, se llegó a una propuesta de buenas prácticas en la seguridad de este gestor para convertirlas en plug-ins que brinden soporte a la aplicación informática AuditBD en este aspecto.

Palabras Claves

Plug-ins, Seguridad, Bases de Datos, PostgreSQL

Tabla de Contenido

Introducción 1
Capítulo 1: Fundamentación Teórica 5
Introducción 5
Calidad del Software 5
Bases de Datos. 7
Sistemas Gestores de Bases de Datos. 7
PostgreSQL. 8
Seguridad Informática. 11
Seguridad Física. 11
Seguridad Lógica. 12
Seguridad de los datos. ¿Por qué proteger los datos? 12
Seguridad en Bases de Datos. 13
Seguridad en PostgreSQL. 13
Seguridad de bases de datos en proyectos de la UCI. 14
Metodologías de desarrollo 15
Herramientas informáticas para probar la seguridad en Bases de Datos. 18
Herramientas utilizadas para desarrollar los plug-ins. 19
Conclusiones parciales. 20
Capítulo 2: Solución Propuesta 21
Introducción 21
Buenas prácticas 21
Estructura de PostgreSQL. 25
Aplicación de seguridad AuditBD. 27
Estructura AuditBD. 27
Plug-ins de AuditBD. 28
Plug-in para BD. 28
Plug-ins para S.O. 29
Fase de desarrollo 29
Fase de Exploración 29
Fase de Planificación de la Entrega 33
Fase de Iteraciones 34
Fase de Producción 42
Fase de Mantenimiento 48
Fase Muerte del Proyecto 48

Conclusiones parciales 48
Capítulo 3: Validación de la Solución 49
Introducción 49
Métodos de validación: 49
Validación por el método Delphi 51
Conclusiones del capítulo 65
Conclusiones 66
Recomendaciones 67
Bibliografía 68
Anexos 71
Glosario 79

Índice de tablas

Tabla 1: Diferencias entre metodologías ágiles y metodologías tradicionales 16
Tabla 2: Plantilla de Historia de Usuarios 30
Tabla 3: Comprobar permisos en los archivos de configuración y directorios de datos. 31
Tabla 4: Verificar Contraseñas en Blanco. 32
Tabla 5: Verificar la existencia de bases de datos con nombres de usuarios. 32
Tabla 6: Comprobar súper usuarios en el SGBD 33
Tabla 7: Verificar usuarios que actualizan System Catalag 33
Tabal 8: Estimación del esfuerzo por historia de usuarios 34
Tabla 9: Plan de duración de las iteraciones 35
Tabla 10: Plan de entrega. 37
Tabla 11: Modelos de tareas 37
Tabla 12: Tarea Comprobar permisos en los archivos de configuración. 37
Tabla 13: Tarea Comprobar permisos en las carpetas de datos. 38
Tarea 14: Devolver resultado de las comprobaciones a la configuración y directorios de datos. 38
Tarea 15: Verificar contraseñas vacías. 38
Tarea 16: Comprobar contraseñas almacenadas en texto claro. 39
Tarea 17: Verificar existencia de contraseñas triviales 39
Tarea 18: Comprobar que las contraseñas no sean los nombres de los usuarios. 39
Tarea 19: Informar vulnerabilidades de contraseñas 40

Tarea 20: Verificar la existencia de bases de datos con nombres de usuarios. 40
Tarea 21: Informar coincidencia de BD y nombre de usuarios. 40
Tarea 22: Comprobar el número de súper usuarios en el SGBD. 41
Tarea 23: Mostrar súper usuarios. 41
Tarea 24: Verificar usuarios que actualizan System Catalog. 41
Tarea 25: Mostrar modificadores de System Catalog. 42
Tabla 26: Pruebas 1: Permisos en los ficheros de configuración. 42
Tabla 27: Pruebas 2: Permisos en los ficheros de configuración. 43
Tabla 28: Pruebas 1: Permisos en las carpetas de datos. 43
Tabla 29: Pruebas 2: Permisos en las carpetas de datos. 43
Tabla 30: Pruebas 1: Contraseñas en blanco. 44
Tabla 31: Pruebas 2: Contraseñas en blanco. 44
Tabla 32: Pruebas 1: Bases de datos con nombre de usuarios. 45
Tabla 33: Pruebas 1: Súper usuarios en el SGBD. 45
Tabla 34: Pruebas 1: Permisos de actualización de System Catalog. 46
Tabla 35: Pruebas 1: Contraseñas comunes. 46
Tabla 36: Pruebas 2: Contraseñas comunes. 46
Tabla 37: Pruebas 3: Contraseñas comunes. 47
Tabla 38: Pruebas 1: Contraseñas almacenadas en claro. 47
Tabla 39: Pruebas 1: Nombre de usuario como contraseña. 47
Tabla 40: Coeficiente de competencia de los expertos seleccionados. 54
Tabla 41: Resumen de la Validación de Expertos. 62
Tabla 42: Cálculo de la Concordancia. 64

Introducción

La presencia del software y de internet se encuentra en casi todas las áreas de la sociedad, va desde el comercio hasta la salud y la educación, acelerando todo un conjunto de funciones diarias, por ejemplo la comunicación mediante el correo electrónico. A través de las aplicaciones informáticas, ya sean web o de escritorio se recogen, procesan, almacenan, recuperan o se obtienen nuevos datos a partir de los que están registrados, estos pueden variar desde las actividades financieras de una compañía hasta los gustos particulares de una persona común. Con el desarrollo de internet se crea un nivel superior en cuanto a las comunicaciones y relaciones entre personas, empresas y gobiernos, etc., donde cobra vital importancia la protección de los datos tanto en canales de comunicación así como en los sistemas de almacenamiento que pueden ser ficheros o Bases de Datos (BD), por ende el software que se desarrolle debe tener un alto nivel de seguridad.

La seguridad de los productos de software se ha convertido en un requisito obligatorio, desempeñando un papel muy importante los Sistemas Gestores de Bases de Datos (SGBD), quienes son los encargados de contener y procesar toda la información con la que se trabaja en los diferentes sistemas informáticos actuales.

Cuba se encuentra en un intenso proceso de preparación de profesionales de la especialidad de informática. Con vistas a soportar este reto se construye la Universidad de las Ciencias Informáticas (UCI), universidad que dentro de su alcance se encuentra la producción de software tanto para consumo nacional como para exportación, pero no solo la producción de software, sino también la producción de software con calidad.

Es muy importante destacar que antes de comercializar cualquier producto, la universidad se encarga de realizar diversas pruebas de calidad al software. Antiguamente dentro de estas pruebas de calidad, quedaban excluidos los tests de seguridad. Hoy, se están desarrollando un conjunto de actividades para erradicar

estos problemas, con el objetivo de producir software no solo con calidad, sino también con un elevado nivel de seguridad.

En la producción de software en la UCI es apreciable el uso de numerosos SGBD, pero la utilización de PostgreSQL en los diferentes proyectos productivos ha causado un gran impacto.

PostgreSQL es uno de los SGBD más usados y competentes que existen en la actualidad. El mismo, pese al esfuerzo de su equipo desarrollador no está exento de vulnerabilidades que podrían comprometer a un sistema que lo use como componente. Es por ello que en las aplicaciones que se desarrollan, uno de los aspectos más importantes a tener en cuenta respecto a la protección de los datos es el aseguramiento de la BD. Debido a la inserción de la UCI en el mundo del software se necesita desarrollar los productos con la mayor seguridad posible, y por tanto una forma de medir la misma para la retroalimentación.

En la actualidad la UCI se encuentra enfrascada en el desarrollo de la aplicación informática AuditBD, que tiene como objetivo fundamental, auditar BD manejadas por diferentes SGBD, pero esta no brinda soporte para PostgreSQL.

Dada la situación problemática antes descrita, se desprende el siguiente *problema científico* de la investigación:

¿Cómo auditar las BD manejadas por el SGBD PostgreSQL en la aplicación AuditBD, desarrollada en la UCI?

Objeto de estudio:

La seguridad de los datos manipulados por SGBD.

Campo de acción:

Buenas prácticas en la seguridad de los servidores de BD PostgreSQL.

El objetivo general es:

Desarrollar Plug-ins de soporte a la aplicación informática AuditBD, que permitan realizar auditorías a las BD manejadas por el SGBD PotgreSQL.

Objetivos específicos:

Realizar un estudio de los aspectos teóricos relacionados con el tema.

Investigar la existencia de aplicaciones informáticas para auditar BD controladas por SGBD, especialmente PostgreSQL.

Elaborar el listado de buenas prácticas para realizar auditorías a las BD manejadas por el SGBD PostgreSQL.

Desarrollar los Plug-ins que brinden soporte a la herramienta AuditBD.

Evaluar la efectividad de la solución.

Se plantea como *idea a defender* que el desarrollo de los Plug-ins para la herramienta AuditBD que le permitan realizar auditorías a las BD manejadas por el SGBD PostgreSQL, garantizará tener el conocimiento de en qué medida se encuentra protegida la información almacenada.

Para el desarrollo del trabajo se decidió definir 3 capítulos, quedando organizados de la siguiente manera:

Capítulo 1: Fundamentación Teórica

En este capítulo se abordan las definiciones y conceptos de términos usados en la investigación, así como las principales aplicaciones informáticas que existen para probar la seguridad en BD. Se tratan algunos temas de interés para el correcto entendimiento del trabajo como es: la calidad de un producto y dentro de esta la seguridad de los mismos, atendiendo principalmente a la seguridad en las BD que utilizan SGBD PostgreSQL. Se muestra la situación actual de la UCI en cuanto al tema de la seguridad de las BD, mediante los resultados arrojados por las encuestas realizadas en los diferentes proyectos productivos existentes en la universidad.

Capítulo 2: Solución propuesta

En este capítulo se muestra la solución que se buscó al problema planteado, para ello, se realiza un estudio de las buenas prácticas empleadas para comprobar la seguridad en las BD, se establecen los parámetros a utilizar para realizar las auditorías a las BD y se desarrollan los plug-ins con los mismos.

Capítulo 3: Validación de la Solución

En este capítulo se realiza la evaluación de la solución propuesta en el capítulo anterior, mediante criterios de los especialistas en el tema y los resultados arrojados tras su aplicación.

Capítulo 1: Fundamentación Teórica

Introducción

En el presente capítulo se ofrece una panorámica sobre los principales temas relacionados con la calidad de un producto, la seguridad informática, la seguridad de la información, las BD y los SGBD, haciendo especial énfasis en el SGBD PostgreSQL, así como en las aplicaciones informáticas existentes para la evaluación de vulnerabilidades y algunas metodologías de desarrollo, incluyendo la que se emplea para el desarrollo de la solución.

Calidad del Software

Cuando se piensa en desarrollar un software o herramienta informática, la condición indispensable y el principal objetivo que se persigue es producir un software con gran calidad, que cumpla y si es posible supere las expectativas de los usuarios.

Uno de los aspectos más importantes a tener en cuenta en el desarrollo de un software es que estén bien definidos sus requisitos, ya que constituyen la base de las medidas de calidad, además la utilización de las metodologías de desarrollo del software es muy útil si se desea obtener un producto con calidad, pues definen todo un conjunto de criterios de desarrollo y rigen a su vez la forma en que se aplica la ingeniería. De manera general, se dice que la calidad de un software consiste en la concordancia de los requisitos funcionales y no funcionales establecidos, con los estándares de desarrollo explícitamente documentados y con las características contenidas que se espera de todo software desarrollado profesionalmente (1). Para saber hasta qué punto un software cuenta con la calidad requerida, se puede hacer uso de algunos factores destinados a determinar cuan perfecto es el producto. Estos factores de pueden dividir en tres grupos (1):

Operaciones del producto. Características operativas.

- Corrección: El grado en que una aplicación satisface sus especificaciones y consigue los objetivos.

- Fiabilidad: El grado que se puede esperar que una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida.
- Eficiencia: La cantidad de recursos hardware y software que necesita una aplicación para realizar sus tareas con los tiempos de respuesta adecuados.
- Integridad: El grado con que puede controlarse el acceso al software o a los datos a personal no autorizado.
- Facilidad de uso: El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.

Revisión del producto. Capacidad para soportar cambios.

- Facilidad de mantenimiento: El esfuerzo requerido para localizar y reparar errores.
- Flexibilidad: El esfuerzo requerido para modificar una aplicación en funcionamiento.
- Facilidad de pruebas: El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos.

Transición del producto. Adaptabilidad a nuevos entornos.

- Portabilidad: El esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo.
- Reusabilidad: Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones.
- Interoperabilidad: El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos.

Cuando se habla de calidad del software todos imaginan un producto que cumpla con los requisitos funcionales establecidos previamente por el cliente, pero el problema radica en que los clientes en su gran mayoría no solicitan explícitamente un mecanismo de seguridad determinado a no ser que lo requiera su sistema, sin embargo, esperan que su aplicación tenga todos los mecanismos necesarios para su correcto funcionamiento.

Existen varios aspectos que influyen en la calidad de un software, uno de ellos y quizás uno de los más importantes es la seguridad, requisito indispensable para que un software cuente con la calidad necesaria. Para comprobar cuan seguro es un software, se emplean varios métodos, uno de ellos es mediante la realización de auditorías, no solo al software de manera general, sino también a la información almacenada en las BD y es en esta dirección que está enfocado este trabajo.

Bases de Datos.

En la actualidad existen varias definiciones de las BD, todas muy claras y de manera general convergen en la siguiente definición:

Una base de datos es un conjunto de datos persistentes que es utilizado por los sistemas de aplicación (2).

Según C. J. Date entre los principales beneficios que ofrecen las bases de datos se encuentran: un alto grado de independencia lógica y física de la información almacenada, que la redundancia entre los datos sea mínima, que los usuarios puedan acceder de forma concurrente a la información, teniendo la posibilidad de realizar consultas optimizadas y complejas y que se logre conservar la integridad de los datos, así como su seguridad.

Sistemas Gestores de Bases de Datos.

Un Sistema Gestor o Manejador de Bases de Datos (SGBD) es un conjunto de programas que permite a los usuarios crear y mantener una BD, por lo tanto, el SGBD es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. Pueden ser de propósito general o específico (3).

En el mundo de la informática, se ha desatado un ascendente desarrollo de los SGBD, contando hoy en día con numerosos sistemas dedicados a manejar la creación de todos los accesos a la BD, pudiendo clasificarlos en dos grupos:

Software bajo licenciamiento libre: Es aquel en el que los usuarios tienen la posibilidad de copiar, modificar, distribuir, ejecutar y mejorar el software, algunos ejemplos son:

- PostgreSQL: Posee una licencia TPL (The PostgreSQL License), de código abierto que permite a cualquier usuario la ejecución, modificación y distribución tanto del software como de la documentación para cualquier propósito (4).
- MySQL: Tiene un modelo dual de distribución del software y una de sus vertientes es bajo licenciamiento GPL (5), como el MySQLCommuntity 5.1.47.

Software bajo licenciamiento comercial: Su uso, redistribución o modificación están prohibidos, requieren que se solicite una autorización para hacerlo.

- Sybase ASE: Es el sistema de gestión de datos críticos, desarrollado por la compañía especializada en soluciones de manejo de información Sybase Inc.
- Microsoft SQL Server: SQL Server es un SGBD relacional que usa Transact-SQL para enviar peticiones entre el cliente y el servidor SQL (6).
- Oracle: Es uno de los SGBD más populares. Es multiplataforma y provee variadas y numerosas funcionalidades (7). Cuenta con gran soporte tanto de documentación como de herramientas informáticas.
- DB2: Es el sistema de BD de IBM, provee soporte para GNU\Linux, UNIX, Windows.

En la UCI:

Para conocer el SGBD más utilizado en la UCI, se contactó con el Director del Departamento de la Dirección Técnica de la Producción: Pedro Yobanis Piñero Pérez, quien asegura que el gestor más utilizado en la universidad es PostgreSQL (8).

PostgreSQL.

PostgreSQL, desarrollado originalmente en el Departamento de Ciencias de la Computación de la Universidad de California en Berkeley. Ofrece soporte a algunas variantes del lenguaje SQL, integridad de transacciones, y extensibilidad de tipos de datos. PostgreSQL es un descendiente de dominio público y código abierto del código original de Berkeley.

Concepto:

Según el grupo de desarrollo de PostgreSQL (9), este gestor es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD, fue pionera en muchos conceptos que sólo estuvieron disponibles en algunos sistemas de BD comerciales mucho más tarde.

Características de PostgreSQL (10):

Soporte SQL92/SQL99: PostgreSQL implementa un subconjunto extendido de los estándares SQL92 y SQL99. En cada versión implementa el soporte para los estándares más avanzados de ANSI SQL que se encuentra disponible.

- Transacciones: Permiten el paso entre dos estados consistentes manteniendo la integridad de los datos.
- Integridad referencial: PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la BD.
- Bloqueos de tabla y filas: PostgreSQL ofrece varios modos de bloqueo para controlar el acceso concurrente a los datos en tablas. Algunos de estos modos de bloqueo los adquiere PostgreSQL automáticamente antes de la ejecución de una declaración, mientras que otros son proporcionados para ser usados por las aplicaciones.
- Constraints y triggers: Tienen la función de mantener la integridad y consistencia en la BD. Ejecución de acciones antes o después de un evento de BD.
- Múltiples tipos de datos predefinidos: Como todos los manejadores de BD, PostgreSQL implementa los tipos de datos definidos para el estándar SQL3 y aumenta algunos otros.
- Soporte de tipos y funciones de usuario: PostgreSQL soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario, además, Incorpora una estructura de datos Array.
- Conectividad TCP/IP, JDBC y ODBC.
- Interfaz con diversos lenguajes: C, C++, Java, Delphi, Python, Perl, PHP, Bash.

Ventajas de PostgreSQL:

Dentro de las facilidades que brinda PostgreSQL se encuentran su estabilidad, alto rendimiento y gran flexibilidad (11), soporte para sistemas operativos como Windows, FreeBSD, GNU\Linux, Mac OS X y Solaris, la documentación que se instala con el núcleo del SGBD, varias herramientas informáticas de línea de comandos y lenguajes procedurales, entre otras, además de la posibilidad de descargar y compilar el código fuente.

Actualidad de PostgreSQL:

Después de un año y medio de desarrollo, el equipo de PostgreSQL lanza al mercado la versión 8.4.0 del producto el pasado 1 de julio de 2009. Esta cuenta con 293 características entre las que se encuentran (12):

- Funciones Windows.
- Expresiones comunes de tablas.
- Joins recursivos.
- Restauración de Backups en paralelo.
- Permisos para usuarios a nivel de columnas.
- Asignación automática de espacio para el "Free Space Map".
- Uso más fácil del sistema de replicación "Warm Standby".
- Uso de certificados SSL para autentificar a usuarios.
- Estadísticas de ejecución para las funciones creadas.
- Editado fácil de funciones en psql.

El 29 de abril del 2010 se anuncia oficialmente la primera versión beta de la familia 9.0 de PostgreSQL. Dentro de sus nuevas funciones y mejoras se incluyen (13):

- Nueva replicación de binarios.
- Soporte para Windows de 64 bits.
- Mejora de la función LISTEN/NOTIFY que permite el envío rápido de mensajes dentro de la base de datos.
- Bloqueos de procedimientos anónimos con el comando DO.

- Disparadores condicionales y “SQL-compliant” por columnas.
- Soporte para Python 3 en PL/Python y numerosas mejoras en PL/Perl.
- Restricciones solo para datos no escalares (restricciones de exclusión).
- Soporte mejorado para valores de datos clave.

Seguridad Informática.

Seguridad se define en el diccionario de la real academia española como la cualidad de seguro, la certeza (conocimiento seguro y claro de algo). De “seguro” se pueden destacar las acepciones: libre de todo peligro daño o riesgo, seguridad, certeza, confianza, mecanismo que impide el funcionamiento indeseado de un aparato, utensilio, máquina o arma, o que aumenta la firmeza de un cierre (14).

Informática en el mismo diccionario se lee: conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores.

Uniendo un concepto con otro se puede deducir la definición prematura de seguridad informática como un conjunto de técnicas que dan certeza y confianza al tratamiento automático de la información.

Como es muy difícil asegurar que un sistema está seguro, por no decir imposible, algunos autores plantean la seguridad como un proceso constante e incremental (15), que incluye un conjunto de técnicas y herramientas informáticas para la protección de la información. Esta última, se recoge, procesa y almacena en programas (software), que a su vez están contenidos en equipos físicos (hardware) que están ubicados en edificaciones; o sea, que para proteger la información habría que defender el local, el hardware y el software.

Seguridad Física.

Se encarga de la protección del local, donde se encuentra instalado el equipamiento, de las amenazas externas tales como: incendios, inundaciones, altas temperaturas, caídas de voltaje y demás fallas eléctricas, hurto, etc.

Seguridad Lógica.

Constituye una barrera defensiva más allá de la Seguridad Física. Según Aneiro (16), la seguridad física se encarga de la protección mediante software y dispositivos digitales o eléctricos del software y los datos. Se toman todas las precauciones necesarias para garantizar la confidencialidad, fundamentalmente.

Seguridad de los datos. ¿Por qué proteger los datos?

Cada vez que se haga referencia a datos, debe entenderse como datos digitalizados.

Si se puede decir que el dinero se encuentra en algún lugar en la red, este se encuentra en un servidor de BD. Cuando se dice que las economías modernas dependen de las computadoras, lo que realmente se quiere decir es que las economías modernas dependen de los sistemas de BD (7).

Los datos constituyen uno de los bienes más difíciles de recuperar en cualquier entidad debido a que se necesita tiempo para recogerlos. Gran cantidad de ellos son el reflejo de la vida y accionar de una entidad, por lo que tendrán un elevado valor para la misma.

Los datos pueden ser ordinarios, o sea, su publicación no implica ningún riesgo, pero si son sensibles, dígase información valiosa para la competencia de una empresa, constituyen secreto de un gobierno, o sencillamente direcciones o datos personales como número de cuenta bancaria, tarjeta de crédito, etc., estos necesitan una protección especial debido a que el solo hecho de su divulgación puede tener repercusiones inesperadas, pudiendo resultar en daños a la entidad involucrada.

Cuando alguien usa una BD para guardar o procesar sus datos, espera que estos estén disponibles siempre que los necesite, que solo los que estén autorizados tengan acceso a dicha información, y que esta no sea modificada o eliminada de ninguna forma que no sea la que se definió.

Otro motivo por el cual proteger los datos, es la pérdida económica que implica un ataque, intencionado o no, que logre comprometer la base de datos, importante es también la pérdida del prestigio del proveedor del servicio o producto inseguro.

Debido a que el objetivo de la producción de software en la UCI es evolucionar para bien y proporcionar un producto de alta calidad, los autores de este trabajo consideran importante prestarle más atención al tema de la seguridad de los datos en las aplicaciones de software desarrolladas en la UCI, específicamente en las desarrolladas con el SGBD PostgreSQL.

Seguridad en Bases de Datos.

La seguridad en las BD depende del SGBD que es el que controla las configuraciones. Entre las funciones de los SGBD no solo se encuentra la administración de tablas, tuplas, etc., sino que sirven como herramienta en la toma de medidas para asegurar la información almacenada o por almacenar.

A pesar de esta seguridad abonada por los SGBD se suelen emplear ORM (Mapeo Objeto - Relacional) para el aseguramiento de la misma, llegando en algunos casos a ignorar en cierto punto la seguridad ofrecida por estos SGBD.

El SGBD tiene que ser un bastión en la protección de los datos, ya que lógicamente es la última barrera entre el atacante y su objetivo, la información.

Dentro de las funciones que tienen las BD para combatir las intrusiones están (17):

1. El control de acceso.
2. La autentificación.
3. Los roles y permisos.
4. Las abstracciones de datos, que son las consultas y los procedimientos almacenados.
5. Las auditorías de datos como: Los logs (registros) y auditorías de tablas y los Frameworks (marcos de trabajo) de seguridad avanzada.

Seguridad en PostgreSQL.

Desde el momento de su instalación PostgreSQL está tomando medidas de seguridad como por ejemplo: no permite conexiones de red al servidor, no se ejecuta como root

en los sistemas Unix (18), el instalador para Windows chequea la fortaleza de la contraseña y sugiere parámetros para una nueva si es necesario.

Los desarrolladores de PostgreSQL publican las vulnerabilidades del gestor en el sitio, proveen versiones con fallos eliminados y otras mejoras además de las de seguridad.

Actualmente PostgreSQL cuenta con una gran cantidad de documentación en inglés y en español, acerca de seguridad y con un abanico de aplicaciones incorporadas que tienen impacto en el control de la seguridad del SGBD. Tampoco es menos cierto que las principales herramientas comerciales que tienen una colección de utilidades necesarias para realizar exámenes de seguridad más profundos, no brindan soporte a PostgreSQL, en cambio, se inclinan por Oracle, IBM DB2, y MS SQL Server principalmente.

Seguridad de bases de datos en proyectos de la UCI.

En sus inicios la UCI tenía como uno de sus principales objetivos, la formación de profesionales capaces de desarrollar software. Actualmente, tras haber desarrollado varios proyectos y adquirido experiencia durante el proceso, se encuentra enfocada a la producción de software con una mayor calidad, prestando atención a la seguridad como parte de la calidad de sus productos. En las entrevistas a jefes de proyectos se pudo constatar que el trabajo que se está realizando en materia de seguridad de BD es aún insuficiente.

Basado en entrevistas que se realizaron para la elaboración de este trabajo a líderes de proyectos, arquitectos, desarrollares, responsable de BD, se pudo apreciar que aunque es un tema de preocupación en los proyectos, las pruebas a la seguridad de las BD se realizan en muy pocos proyectos. Tampoco se brinda asesoría a los desarrolladores responsables de montar, controlar y mantener las BD. Incluso una propuesta de procedimiento para realizarle pruebas a la Capa de Datos durante el proceso de desarrollo de software fue terminada en junio del 2009, o sea, que este no existía o al menos no se había trabajado en serio sobre este tema.

Dentro de la producción se pueden encontrar proyectos donde se implantan las buenas prácticas, partiendo de estudios previos o siendo definidas localmente. Estas prácticas son implementadas y luego no se prueban, dando lugar a que los proyectos

en ocasiones sean liberados solo con la ejecución de las pruebas de calidad, donde no han sido incluidas las pruebas de seguridad.

En varios proyectos de la universidad se trabaja en la línea de establecer un mínimo de seguridad de la información en sus BD y posteriormente realizar pruebas. La información arrojada tras estas actividades, no está publicada. Esto provoca que varios proyectos estén realizando una misma investigación en paralelo y sin compartir la información, contribuyendo así al desperdicio de tiempo y recursos.

Hoy la UCI cuenta con un laboratorio de seguridad, creado desde el 2009. Entre las funciones que realizan se encuentra la investigación y la realización de pruebas, vinculadas a los SGBD. Este se encuentra localizado en la Facultad 2, y en el mismo se tiene soporte para Oracle, MySQL y SQL Server. El equipo de desarrollo del laboratorio se encuentra desarrollando un proyecto en convenio con ETECSA, que consiste en el desarrollo de la aplicación informática AuditBD, herramienta para evaluar las vulnerabilidades en las BD. La misma consiste en una aplicación de software la cual cuenta con extensibilidad mediante plug-ins, que implementan las políticas de seguridad a implantar en la BD, en pocas palabras mediante los plug-ins se pueden evaluar las vulnerabilidades que afecten las políticas de seguridad previamente definidas.

Metodologías de desarrollo

Las metodologías de desarrollo de software abarcan todo el ciclo de vida del software, y se definen como “un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software (19). Hoy en días en el desarrollo de software resulta muy importante el empleo de una metodología de desarrollo debido a las ventajas que estas ofrecen, aunque en ocasiones resulte complicado seleccionar cual es la que más se ajusta a las características del proyecto a desarrollar.

Según Patricio Letelier en (20), las metodologías se pueden clasificar en:

- Metodologías Tradicionales: Hacen mayor énfasis en la planificación y control del proyecto, en la especificación precisa de requisitos y modelado y generan un gran cúmulo de información.

- Metodologías Ágiles: Están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos de desarrollo pequeños, hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso.

En la siguiente tabla se establecen las diferencias entre las Metodologías Ágiles y las Metodologías Tradicionales, analizando algunas de sus características (21).

Metodologías Ágiles	**Metodologías Tradicionales**
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 1: Diferencias entre metodologías ágiles y metodologías tradicionales

A continuación se presentan algunos ejemplos de metodologías tradicionales como es el caso de RUP y metodologías ágiles tales como SCRUM y XP.

Proceso Unificado del Software (RUP):

El proceso unificado de desarrollo de software (RUP por sus siglas en inglés Rational Unified Process), es un proceso que orienta y provee varios medios que dotan al proceso de desarrollo de software de una gran organización y control. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software (22).

SCRUM

Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, con una duración de 30 días y como resultado de cada iteración se obtiene un incremento ejecutable que se muestra al cliente. La segunda característica importante es, las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración (21).

Extreme Programming (XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (21).

Analizando las metodologías anteriormente mencionadas y considerando las características del trabajo que se pretende llevar a cabo, se decidió hacer uso de la metodología XP para el desarrollo del mismo. Esta metodología es la que más se acerca a las necesidades del proyecto que se pretende desarrollar, analizando que el equipo de trabajo está compuesto por 2 programadores, existe un constante vínculo con el cliente que en este caso es el Director del proyecto encargado de desarrollar la aplicación informática AuditBD y el período de tiempo con el que se dispone es corto, aproximadamente 5 meses.

Herramientas informáticas para probar la seguridad en Bases de Datos.

Una vez implementada la seguridad de la aplicación a nivel de datos. Se necesita probar la efectividad de la misma. Estas comprobaciones pueden realizarse utilizando listas de chequeo, algunos scripts y herramientas informáticas.

NGSSQuirrel: Es un producto de la empresa NGSS (Next Generation Security Software). Es un scanner de evaluación de vulnerabilidades para BD. Dentro de sus principales funciones se encuentra la vista y edición de pruebas para el núcleo, la creación de chequeos para usuarios, referencias de usuarios y tipos de referencia de usuarios. Este producto tiene versiones para MS SQL Server, Oracle, IBM DB2, entre otros (23).

AppDetectivePro: Dentro de sus funcionalidades está la identificación en una red de BD disponibles, escaneo de vulnerabilidades tanto desde fuera de la BD, como desde el interior, así como el chequeo de los permisos de los usuarios. Este producto brinda soporte fundamentalmente para varias versiones de Oracle, de IBM DB2 y de Microsoft SQL Server (24).

DB Audit Expert de la empresa Soft Tree Technologies: Es una solución con funcionalidades para la seguridad y la realización de auditorías a las BD. Brinda soporte para Oracle, IBM DB2, MS SQL Server, Sybase y MySQL (25).

Hay otras herramientas como la Secure Auditor de la Security Byte, pero el acuerdo para la obtención de sus productos excluye a Cuba por prohibiciones del gobierno norteamericano.

Aparte de las aplicaciones para evaluar la seguridad se considera conveniente mencionar otras como JMeter, Data Test Professional, Forecast for Databases, que brindan funcionalidades para evaluar aspectos como el rendimiento y desempeño, estos aspectos, en algún momento pueden influir en la seguridad de la base de datos, aunque no se especializan en la comprobación de vulnerabilidades en privilegios, contraseñas, etc. como las mencionadas anteriormente.

Debido a que las algunas de las herramienta**s mencionadas anteriormente son** privativas y las que no lo son, no realizan las funciones necesarias para evaluaciones

de vulnerabilidades en PostgreSQL, es que se decide realizar un software que brinde este servicio.

Herramientas utilizadas para desarrollar los plug-ins.

La elección de las herramientas que se exponen a continuación tiene su justificación en que son funciones que se encuentran en un SO básico GNU\Linux, una vez instalado el servidor PostgreSQL, proveen mecanismos fáciles de utilizar e interpretar al ver las funciones definidas con los mismos. Los lenguajes utilizados y principales herramientas en la elaboración de los plug-ins fueron SQL para la comunicación con la BD, AWK, XML y scripts de Shell, específicamente para BASH.

Structured Query Language (SQL): Es un lenguaje estándar de comunicación con BD, que permite trabajar con cualquier tipo de lenguaje como ASP o PHP en combinación con cualquier tipo de BD como pueden ser MySQL, PostgreSQL y SQL Server entre otras. Tiene sus inicios en los años 1974 por parte de Donald Chamberlin y de otras personas que trabajaban en los laboratorios de investigación de IBM. SQL cuenta con dos tipos de comandos, los DDL que permiten crear y definir nuevas BD, campos e índices y los DML que posibilitan generar consultas para ordenar, filtrar y extraer información de las BD (26).

Bourne Again Shell (Bash): Es uno de los tipos de shell desarrollados para interpretar comandos. Tiene sus inicios en SH (Bourne Shell) desarrollado a mediados de 1970 por Stephen R. Bourne en los Laboratorios de AT&T. Como parte del proyecto GNU se desarrolla en 1987 (27). Bash que combina características de shells como csh y tcsh.

Shell o terminales: Son interfaces que les facilitan a los usuarios interactuar con el sistema operativo. Constituyen un entorno para la interpretación de órdenes.

AWK: Es un lenguaje diseñado para trabajar con archivos estructurados y patrones de texto. En los SO usualmente se instalan uno o más intérpretes de este lenguaje mawk y gawk. Estas herramientas permiten su combinación con diferentes comandos de la shell para capturar y manipular salidas de los comandos lo que les proporciona comodidad para el desarrollo.

Conclusiones parciales.

Con la investigación realizada se sentaron las bases para la solución del problema en cuestión.

Tras el estudio realizado de las principales herramientas para auditar la seguridad en las BD, se concluye que existen pocas aplicaciones informáticas para la evaluación de la seguridad, en su mayoría son privativas y no ofrecen soporte para PostgreSQL, ofreciéndoselo a SGBD como DB2, SQL Server y Oracle principalmente.

Producto de entrevistas realizadas a líderes de proyectos, arquitectos, desarrolladores y responsables de BD se conoció que en la UCI existe un avance incipiente en cuanto al tema de seguridad.

Mediante una entrevista al Director del Departamento de la Dirección Técnica de la Producción se concluye que el SGBD más utilizado en la UCI es PostgreSQL.

Capítulo 2: Solución Propuesta

Introducción

En el capítulo 2 se abordarán los temas necesarios para la explicación y la reseña de la solución propuesta al problema en cuestión que se plantea a continuación. Se abordan temas como buenas prácticas, algunas nociones de la estructura de PostgreSQL 8.3, versión con la que se pretende trabajar, permitiendo el entendimiento del proceso para llegar a la solución.

El proyecto AuditBD, desarrollado por la UCI en convenio con ETECSA, el cual consiste en el desarrollo de una herramienta informática para evaluar las vulnerabilidades de las BD manejadas por diversos SGBD, no brinda soporte para PostgreSQL.

AuditBD se desarrolla en el laboratorio de seguridad perteneciente a la universidad ubicado en la facultad 2. Este software tiene como objetivos implementar de manera automatizada pruebas para varias BD manejadas por diferentes SGBD. La aplicación cuenta con plug-ins que son los que implementan las políticas a evaluar, o sea, configuración correcta que evita una vulnerabilidad en una BD.

Con vistas a implementar los plug-ins para la mencionada aplicación, es necesario tener una propuesta de la política de seguridad para la base de datos a probar, la cual consiste en un conjunto de buenas prácticas, producto de una investigación previa.

Buenas prácticas

Las buenas prácticas son procesos o metodologías que representan la manera más efectiva de obtener un objetivo. Constituye un modelo a seguir que es válido hasta que se replantea basándose en los avances del campo donde se aplica.

Cuando se toca el tema de las BD varias empresas y personalidades, recomiendan el empleo de las buenas prácticas para la implementación de la seguridad, pero más que buenas prácticas podrían ser consideradas como necesidades, debido al impacto e importancia que ha tenido históricamente la protección de la información.

A continuación se muestra un listado de buenas prácticas para el aseguramiento de la seguridad de las BD manejadas por SGBD como Oracle, MySQL, SQL Server y PostgreSQL, haciendo especial énfasis en este último.

- Aplicaciones innecesarias: Solo se ejecutan en el servidor las aplicaciones necesarias para el servicio que ofrece.
- Cuenta del servicio: Minimizar los privilegios a la cuenta encargada de ejecutar el servicio de PostgreSQL.
- Ficheros de las BD: Los ficheros con la información de las BD, no deben estar ubicados en particiones del SO, debido a que su crecimiento puede provocar denegación de servicio.
- Histórico de comandos: El histórico de comandos debe estar deshabilitado, para evitar el acceso a través de este a información que haya quedado guardada, como contraseñas.
- Inicio de sesión local: Evitar que la cuenta del servicio pueda crear sesión localmente.
- Autenticación mixta: Usar autenticación exclusiva de PostgreSQL o exclusiva del sistema operativo. Una combinación de ambos podría devenir en más complejidad, y no necesariamente más seguridad.
- Uso de firewalls: Se aconseja la implementación de muros de seguridad (Firewalls) configurándolos manualmente para evitar configuraciones por defecto inseguras elevando así la seguridad del SO.
- Permisos sobre los directorios de datos: Solos los usuarios de PostgreSQL acceden a los directorios de datos. En Debian y Ubuntu están en /var/lib/postgresql/8.x/main. En estos directorios postgres guarda las BD en archivos binarios identificados por un número único. El autorizado a crear y modificar estos archivos es el usuario postgres, encargado del servicio.
- Permisos sobre los ficheros de configuración: Los ficheros deben pertenecer y ser accedidos solo por el usuario postgres. Los ficheros de configuración se encuentran usualmente en /etc/postgresql/8.x/
- Permisos sobre ficheros de logs: El dueño de los ficheros de logs debe ser el usuario del servicio de postgresql.

- Permisos sobre ficheros SSL: Los ficheros para la configuración de SSL deben estar bien protegidos principalmente server.crt y server.key
- Los logs habilitados: Los logs de errores, y de autenticación deben estar habilitados, puede incluir otros con logs de conexión y desconexión, lo necesario para tener un buen diario del funcionamiento e interacción del servidor.
- Ubicación de directorios de logs: Ubicados fuera de los directorios del SO y de la BD, para almacenar una gran cantidad y no perjudicar el funcionamiento de ambos sistemas.
- Uso de estándar syslog: Se recomienda el uso del estándar de syslog, debido a su popularidad en sistemas GNU\Linux, aunque no es el único estándar que soporta PostgreSQL. Se recomienda en caso de cargar logs a programas utilizar entonces el formato csv.
- Actualización de SGBD: En cada versión superior de PostgreSQL, se arreglan y mejoran funciones por lo que es aconsejable usar la última versión estable. Instalar parches y utilidades de seguridad.
- Bases de datos en desuso o de pruebas: La eliminación de BD genéricas o en desuso. Conduce a una mejor seguridad, debido a que se desactivan posibles vulnerabilidades, ya que son menos los recursos a proteger.
- Complejidad de las contraseñas: Se recomienda poner límite de complejidad de las contraseñas para prevenir el descifrado en caso de que sea robada. PostgreSQL no soporta esta característica pero si se puede usar con un método de autenticación externo como GSSAPI o LDAP.
- Contraseñas triviales: El uso de contraseñas triviales como postgres, 123456789, admin, etc., que se utilizan a veces para facilitar el acceso a la BD durante pruebas o desarrollo, son vulnerabilidades que no se deben dejar pasar, debido a que comprometen el acceso al servidor de BD.
- Contraseñas en blanco: Las contraseñas en blanco favorecen que cualquier usuario se autentique, y no se puede garantizar la identificación del mismo.
- BD con nombre de usuario: El uso de BD con el mismo nombre que usuarios, debido a que PostgreSQL reserva algunas configuraciones como sameuser,

samerole que en combinación con trust por ejemplo, pueden comprometer la confidencialidad de algunas BDs.

- sameuser samerole: Los valores sameuser y samerole en combinación a veces son usados para hacer conexiones a BD con el nombre de user o role, por lo que se exime al cliente, que inicia la conexión, de la especificación de la BD involucrada. Esta oportunidad puede ser a aprovechada por atacantes para acceder a alguna BD en caso de comprometer una cuenta.
- Contraseñas en claro: No se recomienda el envío de contraseñas en texto plano para la autenticación por el alto riesgo que representa. Es necesario evitar este método por su debilidad ante sistemas pasivos de captura de información.
- Métodos de autenticación: Se recomienda el uso de métodos de autenticación como md5, krb5, ldap, pam, gss, los demás métodos tienen características que los hacen menos seguros que estos.
- Lenguajes confiables: Se recomienda usar lenguajes confiables, que son los lenguajes que no tienen acceso a elementos fuera del dominio de las BD definido por PostgreSQL.
- Uso de reject: Usar la sentencia reject, para controlar el acceso a las BD. Esta sentencia permite controlar mejor quien no debe acceder a determinadas BD del servidor.
- Uso de vistas: Usar vistas como abstracciones de las tablas para que no se acceda directamente a ellas.
- Un solo administrador de BD: Un solo administrador en cada BD, el propietario.
- Acceso a la BD postgres: Solo el administrador podrá tener acceso a la BD postgres. Usar un súper-administrador, que no se llame postgres.
- Otorgar permisos: Es necesario minimizar el número de usuarios con el derecho a otorgar permisos, debido a que se puede utilizar esa característica para escalar privilegios.
- Permiso para actualizar el System Catalog: Solo el usuario administrador de la BD tiene permiso para actualizar el System Catalog (Catalogo del sistema). El

System Catalog provee información de los metadatos en el cluster de BD; por lo que no basta con ser súper usuarios para manipularlo.

- Creación de roles: Solo los súper usuarios pueden crear roles. A menos que la aplicación necesite crear roles.
- Uso de pgcrypto: pgcrypto es un módulo de postgresql que provee funciones de cifrado, que permiten una mejor protección a la información.

Estructura de PostgreSQL.

Primeramente las BD se guardan en el disco duro, en las carpetas global y base, definidas previamente en los tablespaces, pg_global y pg_base, donde cada BD tiene un identificador numérico, también se pueden definir otros tablespaces.

Cada tabla tiene una colección de filas, cada fila de una tabla dada comparte el mismo conjunto de nombres de columnas, cada columna usa un tipo específico de datos. Las tablas se encuentran agrupadas en BD y al conjunto de estas administradas por una sola instancia de PostgreSQL se le llama cluster de BD.

Los principales archivos de configuración de PostgreSQL son postgresql.conf, pg_hba.conf, pg_ident.conf, en ellos se guardan las configuraciones generales y de acceso.

Para la organización en el cluster y las tablas, así como para la configuración dentro de estas, el SGBD PostgreSQL tiene un catálogo donde se definen la forma en que se va a organizar la información en el mismo. Este se llama System Catalog (catálogo del sistema).

System catalog es donde se encuentran los metadatos de esquemas de las BD, con informaciones acerca de tablas, columnas, tipos, etc. Los catálogos de PostgreSQL son tablas regulares que se pueden personalizar aunque no es recomendable hacerlo ya que se puede perturbar el funcionamiento esperado a partir de la lectura de los documentos del proveedor.

Vistas del Sistema: Algunas de estas vistas sirven para acceder a consultas en el catálogo del sistema, y otras para hacerlo al estado interno del servidor.

Esquema de información: Provee un conjunto alternativo de vistas que solapa las funcionalidades de las consultas del sistema, con la diferencia que este esquema es SQL mientras que las vistas del sistema son específicamente PostgreSQL, es preferible usar el esquema de información si este provee toda la información necesitada, pero este no ofrece información de las características de PostgreSQL.

Dentro de las tablas que usaremos que se encuentran en la sección antes mencionada se encuentran pg_shadow, pg_database, pg_user, pg_group, pg_authid,estas vistas y tablas como se puede imaginar, gestionan información acerca de BD, roles y grupos.

Para el desarrollo de la solución se hizo un análisis de las funciones de las tablas existentes y se llegó a la conclusión que las siguientes tablas estaban en el dominio de datos que se necesita para la implementación de la misma. Debido a su importancia en la seguridad de la BD.

pg_database: Almacena información acerca de las BD, usar el campo "datdba" donde está el usuario dueño de la BD.

pg_role: Es una vista pública de pg_authid donde se almacena información acerca de los identificadores de autorización, más conocidos como roles. Un rol engloba los conceptos de usuario y grupo. Un usuario es esencialmente un rol con la bandera “rolcanlogin" activada. Cualquier rol puede tener otros roles como miembros.

pg_shadow: Muestra las propiedades de los roles que pueden iniciar sesión. Existe un catálogo de este tipo en cada cluster.

pg_tables: Es una vista que muestra información acerca de las tablas existentes en la BD.

pg_user: Es una vista simple y publica de pg_shadow, no muestra contraseñas, siempre muestra “*******” en el campo contraseña.

pg_proc: Contiene información acerca de los procesos almacenados.

pg_language: Tiene detalles de los lenguajes procedimientos contenidos que permiten el registro y ejecución de los mismos. Cada BD tiene su copia de pg_language.

pg_triggers: Contiene los disparadores almacenados en tablas. Cada tabla tiene una copia de este catálogo.

Aplicación de seguridad AuditBD.

AuditBD es una herramienta informática para auditar BD. Se lleva a cabo su desarrollo en el laboratorio de seguridad de la facultad 2 y se encuentra en su primera versión operativa.

Surge en colaboración con ETECSA y telemáticos por la necesidad de automatizar el proceso de auditoría de BD llevado a cabo por profesionales de esta empresa, de forma manual.

El proceso que siguen los expertos en ETECSA es la confección de una matriz de buenas prácticas que tenían definidos el nombre, una descripción, y un resultado ideal.

Esta herramienta implementa funcionalidades extensibles mediante plug-ins.

Dentro de sus principales componentes en el desarrollo se encuentran Python, librerías para trabajo con XML, MARA, JAXML.

Extensible Markup Language (XML): Es un estándar de etiquetado de documentos, lo suficientemente flexible y legible para emplearse en dominios tan diferentes como sitios web, intercambio de datos, gráficos vectoriales, sistemas de correo de voz y más (28).

AMARA: Es una librería o módulo para Python que provee un conjunto de funciones para el procesamiento de XML.

JAXML: Es un módulo de Python para generar automáticamente documentos legibles.

Estructura AuditBD

AuditBD es una herramienta informática que se compone de un cargador de plug-ins y el núcleo de la aplicación encargado de ejecutar las auditorías definidas en los mismos, también cuenta con un componente que se encarga de generar informes. Tanto para los plug-ins como para la generación de reportes se utilizan librerías XML.

Plug-ins de AuditBD.

AuditBD utiliza 2 tipos de plug-ins: los de buenas prácticas de Sistema Operativo (SO), que son las configuraciones especiales para un SO que vaya a soportar un servidor de BD, y las buenas prácticas para los SGBD.

Plug-in para BD.

Los plug-ins de BD de AuditBD tienen la siguiente estructura:

- Identificador: Corresponde a una combinación única de caracteres alfanuméricos que identifican al plug-in.
- Nombre: En este lugar se pone el nombre del gestor de BD.
- Descripción: Descripción del método a seguir en esta comprobación.
- Aclaración: Comentario extra, para un mejor entendimiento del funcionamiento del plug-in.
- Dependencias: Elementos necesarios para realizar la auditoría al SGBD.
- Conexiones al gestor: Cómo se conecta al gestor.
- Excepciones: Una especificación para controlar errores.
- Plug-in: Listado de plug-ins para hacer las pruebas.
- Creador: En esta sección figura(n) el/los autor(es) del mismo.

Plug-in de buenas prácticas para BD.

Los plug-ins de BP listan los plug-ins necesarios para realizar la comprobación de las buenas prácticas definidas. Estos plug-ins tienen una estructura parecida a los de BD, ya que coinciden en los campos Identificador, Nombre, que en este caso sería el nombre de la buena práctica, Descripción, Aclaración, Creador. También se le agregan las secciones Sentencia, donde se encuentran las órdenes principalmente escritas en SQL para implementar la buena práctica y por último el campo resultado esperado, que no es más que la coincidencia con la buena práctica.

Plug-ins para S.O.

Los plug-ins para S.O. Tienen una estructura idéntica a los de BD, con las variaciones correspondientes, para su ajuste al sistema operativo, lo mismo ocurre con los complementos de buenas prácticas para S.O.

Concluyendo se puede decir que esta aplicación de software funciona con 4 tipos de plug-ins, aunque en la práctica son 2 tipos: los del elemento a comprobar y los que implementan esta comprobación.

Los elementos a comprobar son los SGBD y los SO, mientras que las implementaciones de comprobación no son más que las buenas prácticas especificas previamente.

Fase de desarrollo

Elaborado el listado de buenas prácticas y con el objetivo de obtener un producto con alta calidad y en un corto período de tiempo se decide hacer uso de la metodología de desarrollo XP para el desarrollo de los plug-ins. Esta metodología basa su funcionamiento en el desarrollo de 6 faces, las cuales se explican a continuación.

Fase de Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología (29).

Historias de Usuarios

Las HU constituyen la técnica empleada por la metodología XP para que los clientes expresen las características que desean tenga el sistema. Estas HU, son elaboradas mediante el lenguaje de clientes, sin hacer uso de lenguajes técnicos y cada una constituye la descripción de las características principales. Las HU, son la base para las pruebas de unidad, presiden la creación de pruebas de aceptación y cada una se

descompone en tareas que les son asignadas a los programadores, para ser implementadas durante una iteración.

Para la elaboración de las HU no existe una planilla específica, de ahí que decida utilizar el siguiente modelo:

Historia de Usuario	
Nombre Historia de Usuario: Comprobar permisos en los archivos de configuración.	
Número:	**Iteración Asignada:**
Prioridad:	**Puntos estimados:**
Cliente:	**Tipo de actividad:**
Descripción:	
Observaciones:	

Tabla 2: Plantilla de Historia de Usuarios

Leyenda:

Nombre de Historia de Usuario: Constituye el nombre de la funcionalidad que se desea implementar.

Numero: Es el identificador de la HU, único para cada una.

Prioridad: Puede ser alta, media o baja, en dependencia del orden en que se deba implementar.

Descripción: En este campo se da una explicación detallada de la HU.

Observaciones: Se especifican aspectos que deben tenerse en cuanta para desarrollar la HU.

Usuario: Este campo contiene el nombre del creador de la HU.

Iteración Asignada: Se especifica en que iteración se desarrollará la HU.

Puntos de Estimación: Son asignados por los programadores y constituyen el esfuerzo asociado a la implementación de las HU, constituyen el equivalente a una semana de programación y por lo general las HU tienen de 1 a 3 puntos.

Historia de Usuario	
Nombre Historia de Usuario: Comprobar permisos en los archivos de configuración y directorios de datos.	
Número: 1	**Iteración Asignada:** 1
Prioridad: Alta	**Puntos estimados:** 2
Cliente: AuditBD	**Tipo de actividad:** Nueva
Descripción: Se verifican los permisos asignados a los archivos de configuración y los directorios de datos del servidor de PostgreSQL.	
Observaciones: El dueño de los ficheros, así como de los directorios, es el usuario encargado de ejecutar el servidor de PostgreSQL y solo este tiene permisos de escritura.	

Tabla 3: Comprobar permisos en los archivos de configuración y directorios de datos.

Historia de Usuario	
Nombre Historia de Usuario: Verificar contraseñas vulnerables.	
Número: 2	**Iteración Asignada:** 2
Prioridad: Alta	**Puntos estimados:** 2
Cliente: AuditBD	**Tipo de actividad:** Nueva
Descripción: Se verifican que no existan usuarios con contraseñas en blanco,	

triviales, que sean el nombre del usuario o que se almacenen en texto claro.
Observaciones: No deben existir usuarios que posean contraseñas con las características mencionas anteriormente.

Tabla 4: Verificar Contraseñas en Blanco.

Historia de Usuario	
Nombre Historia de Usuario: Verificar la existencia de bases de datos con nombres de usuarios.	
Número: 3	**Iteración Asignada:** 2
Prioridad: Alta	**Puntos estimados:** 1
Cliente: AuditBD	**Tipo de actividad:** Nueva
Descripción: Se verifica que no existan bases de datos con igual nombre que usuarios.	
Observaciones: No debe existir ninguna base datos con nombre igual al de un usuario, si hay alguna excepción es postgres.	

Tabla 5: Verificar la existencia de bases de datos con nombres de usuarios.

Historia de Usuario	
Nombre Historia de Usuario: Comprobar súper usuarios en el SGBD.	
Número: 4	**Iteración Asignada:** 3
Prioridad: Alta	**Puntos estimados:** 1
Cliente: AuditBD	**Tipo de actividad:** Nueva

Descripción: Se verifica que exista la menor cantidad de súper usuarios.
Observaciones: Se aceptan dos súper usuarios como máximo, el administrador del servidor y el usuario de prueba.

Tabla 6: Comprobar súper usuarios en el SGBD

Historia de Usuario	
Nombre Historia de Usuario: Verificar usuarios que actualizan System Catalog.	
Número: 5	**Iteración Asignada:** 3
Prioridad: Alta	**Puntos estimados:** 1
Cliente: AuditBD	**Tipo de actividad:** Nueva
Descripción: Se verifica que exista la menor cantidad de usuarios que actualicen el System Catalog.	
Observaciones: El privilegio de modificar el System Catalog es tan crítico que ser súper usuario no lo garantiza. Se acepta como máximo un súper usuario, al administrador del servidor.	

Tabla 7: Verificar usuarios que actualizan System Catalag

Fase de Planificación de la Entrega

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

Las estimaciones de esfuerzo asociado a la implementación de las historias, son establecidas por los programadores, utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la

"velocidad" de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación (29).

Estimaciones de esfuerzo por historia de usuarios

No.	Historias de Usuarios	Puntos de Estimación (semanas)
1.	Comprobar permisos en los archivos de configuración y directorios de datos.	2
2.	Verificar contraseñas vulnerables.	2
3.	Verificar la existencia de bases de datos con nombres de usuarios.	1
4.	Comprobar súper usuarios en el SGBD.	1
5.	Verificar usuarios que actualizan System Catalog.	1

Tabal 8: Estimación del esfuerzo por historia de usuarios

Fase de Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser

utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fuercen la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración. Al final de la última iteración el sistema estará listo para entrar en producción.

Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores (29).

Plan de iteraciones

Para la implementación de los plug-ins se decidió hacer tres iteraciones:

Primera Iteración: Se implementan las HU 1 y 2.

Segunda Iteración: Se implementan las HU 3 y 4.

Tercera Iteración: Se implementan las HU 4 y 5.

Plan de duración de las iteraciones

Historias de Usuario a implementar	Iteraciones	Duración total de las iteraciones
- Comprobar permisos en los archivos de configuración y directorios de datos.	1	2 semanas
- Verificar contraseñas vulnerables. - Verificar la existencia de bases de datos con nombres de usuarios.	2	3 semanas
- Comprobar súper usuarios en el SGBD. - Verificar usuarios que actualizan System Catalog.	3	2 semanas

Tabla 9: Plan de duración de las iteraciones

Plan de Entrega

El Plan de Entrega le permite a los programadores desglosar las HU en tareas, facilitando una menor complejidad a la hora desarrollar funcionalidades complejas.

Número de HU	Historias de Usuario	Tareas
1.	Comprobar permisos en los archivos de configuración y directorios de datos.	1. Comprobar permisos en los archivos de configuración. 2. Comprobar permisos en los directorios de datos. 3. Devolver resultado de las comprobaciones a la configuración y directorios de datos.
2.	Verificar contraseñas vulnerables.	4. Verificar contraseñas vacías. 5. Comprobar contraseñas almacenadas en texto claro. 6. Verificar existencia de contraseñas triviales. 7. Comprobar que las contraseñas no sean los nombres de los usuarios. 8. Informar vulnerabilidades de contraseñas.
3.	Verificar la existencia de bases de datos con nombres de usuarios.	9. Verificar la existencia de bases de datos con nombres de usuarios. 10. Informar coincidencia de BD y nombre de usuarios.

4.	Comprobar súper usuarios en el SGBD.	11. Comprobar el número de súper usuarios en el SGBD. 12. Mostrar súper usuarios.
5.	Verificar usuarios que actualizan System Catalog.	13. Verificar usuarios que actualizan System Catalog. 14. Mostrar modificadores de System Catalog.

Tabla 10: Plan de entrega.

Para la descripción de cada una de las tareas se utilizó el siguiente formato de tabla:

Tarea	
No. Tarea:	**No. Historia de Usuario:**
Nombre de la tarea:	
Descripción:	

Tabla 11: Modelos de tareas.

A continuación se presentan con un mayor número de detalles las tareas que conforman las HU.

Tarea	
No. Tarea: 1	**No. Historia de Usuario:** 1
Nombre de la tarea: Comprobar permisos en los archivos de configuración.	
Descripción: El sistema captura los permisos y propietarios de los archivos y los compara con una propuesta previa.	

Tabla 12: Tarea Comprobar permisos en los archivos de configuración.

Tarea

No. Tarea: 2	**No. Historia de Usuario:**1
Nombre de la tarea: Comprobar permisos en las carpetas de datos.	
Descripción: El sistema captura los permisos y propietarios de las carpetas de datos y los compara con una propuesta previa.	

Tabla 13: Tarea Comprobar permisos en las carpetas de datos.

Tarea	
No. Tarea: 3	**No. Historia de Usuario:** 1
Nombre de la tarea: Devolver resultado de las comprobaciones a la configuración y directorios de datos.	
Descripción: El sistema muestra un mensaje informando si los resultados de las comprobaciones son correctos o no apropiados.	

Tarea 14: Devolver resultado de las comprobaciones a la configuración y directorios de datos.

Tarea	
No. Tarea: 4	**No. Historia de Usuario:** 2
Nombre de la tarea: Verificar contraseñas vacías.	
Descripción: El sistema comprueba la tabla pg_shadow de System Catalog la existencia de usuarios con contraseñas vacías.	

Tarea 15: Verificar contraseñas vacías.

Tarea	
No. Tarea: 5	**No. Historia de Usuario:** 2

Nombre de la tarea: Comprobar contraseñas almacenadas en texto claro.
Descripción: El sistema verifica en la tabla pg_shadow de System Catalog la existencia de usuarios con contraseñas almacenadas en texto claro.

Tarea 16: Comprobar contraseñas almacenadas en texto claro.

Tarea	
No. Tarea: 6	**No. Historia de Usuario:** 2
Nombre de la tarea: Verificar existencia de contraseñas triviales.	
Descripción: El sistema verifica en la tabla pg_shadow de System Catalog la existencia de usuarios con contraseñas triviales.	

Tarea 17: Verificar existencia de contraseñas triviales.

Tarea	
No. Tarea: 7	**No. Historia de Usuario:** 2
Nombre de la tarea: Comprobar que las contraseñas no sean los nombres de los usuarios.	
Descripción: El sistema verifica en la tabla pg_shadow de System Catalog la existencia de usuarios con contraseñas iguales al nombre.	

Tarea 18: Comprobar que las contraseñas no sean los nombres de los usuarios.

Tarea	
No. Tarea: 8	**No. Historia de Usuario:** 2
Nombre de la tarea: Informar vulnerabilidades de contraseñas.	
Descripción: El sistema muestra en el caso:	

Contraseñas vacías: El número de usuarios que presentan contraseñas vacías y sus nombres. Contraseñas almacenadas en texto claro: La cantidad de contraseñas sin cifrar. Contraseñas triviales y usuarios con igual nombre que contraseña: El nombre del usuario.

Tarea 19: Informar vulnerabilidades de contraseñas.

Tarea	
No. Tarea: 9	**No. Historia de Usuario:** 3
Nombre de la tarea: Verificar la existencia de bases de datos con nombres de usuarios.	
Descripción: El sistema busca en la tabla pg_database el nombre de la BD y en la tabla pg_shadow, el nombre del usuario para establecer comparaciones.	

Tarea 20: Verificar la existencia de bases de datos con nombres de usuarios.

Tarea	
No. Tarea: 10	**No. Historia de Usuario:** 3
Nombre de la tarea: Informar coincidencia de BD y nombre de usuarios.	
Descripción: El sistema muestra una tabla con el listado de los usuarios que tienen igual nombre que la base de datos.	

Tarea 21: Informar coincidencia de BD y nombre de usuarios.

Tarea	
No. Tarea: 11	**No. Historia de Usuario:** 4

Nombre de la tarea: Comprobar el número de súper usuarios en el SGBD.
Descripción: El sistema busca en la tabla pg_shadow la cantidad de súper usuarios existentes en la BD.

Tarea 22: Comprobar el número de súper usuarios en el SGBD.

Tarea	
No. Tarea: 12	**No. Historia de Usuario:** 4
Nombre de la tarea: Mostrar súper usuarios.	
Descripción: El sistema muestra un listado con el nombre de todos los súper usuarios encontrados.	

Tarea 23: Mostrar súper usuarios.

Tarea	
No. Tarea: 13	**No. Historia de Usuario:** 5
Nombre de la tarea: Verificar usuarios que actualizan System Catalog.	
Descripción: Se busca en la tabla pg_authid los usuarios que tienen permisos para actualizar el System Catalog.	

Tarea 24: Verificar usuarios que actualizan System Catalog.

Tarea	
No. Tarea: 14	**No. Historia de Usuario:** 5
Nombre de la tarea: Mostrar modificadores de System Catalog.	
Descripción: El sistema muestra la cantidad de usuarios y sus nombres en caso de	

coincidencias.

Tarea 25: Mostrar modificadores de System Catalog.

Fase de Producción

Con el objetivo de entregar un mejor producto al cliente, que cumpla correctamente las funciones para las que se concibió, se hace necesario realizarle pruebas a las funcionalidades implementadas, por lo que se decidió aplicar pruebas de aceptación.

Prueba 1	
Plug-in: duso1	**Nombre:** Permisos en los ficheros de configuración.
Descripción: Los archivos de configuración son de la propiedad del usuario postgres encargado del servicio, el postgresql.conf tiene permisos de escritura solo del propietario, y el resto permisos de lectura; el pg_hba.conf tiene permisos de escritura para el propietario y el grupo tiene permisos de lectura, el resto no tiene permisos.	
Resultado esperado: Se imprime "Correcto, los permisos son aceptables"	**Resultado Obtenido:** Se imprime "Correcto, los permisos son aceptables"
Observaciones: Anexo 1	
Evaluación de la prueba: Satisfactorio	

Tabla 26: Pruebas 1: Permisos en los ficheros de configuración.

Prueba 2	
Plug-in: duso1	**Nombre:** Permisos en los ficheros de configuración.
Descripción: El archivo postgresql.conf se le han cambiado los permisos de "-rw-r--r--" a "-rwxrwxrwx" dándole una configuración errónea.	
Resultado esperado: Se imprime "Hay algunos permisos no recomendados"	**Resultado Obtenido:** Se imprime "Hay algunos permisos no recomendados"

Observaciones: Anexo 2
Evaluación de la prueba: Satisfactorio

Tabla 27: Pruebas 2: Permisos en los ficheros de configuración.

Prueba 3	
Plug-in: duso2	**Nombre:** Permisos en las carpetas de datos.
Descripción: Los directorios tienen los permisos recomendados: Los directorios "main", "global" y "base" son propiedad del usuario encargado del servicio postgres y solo el mismo tiene permisos de escritura y lectura.	
Resultado esperado: Se imprime "Correcto, los permisos son aceptables"	**Resultado Obtenido:** Se imprime "Correcto, los permisos son aceptables"
Observaciones: Anexo 3	
Evaluación de la prueba: Satisfactorio	

Tabla 28: Pruebas 1: Permisos en las carpetas de datos.

Prueba 4	
Plug-in: duso2	**Nombre:** Permisos en las carpetas de datos.
Descripción: Al directorio “base” se le ha cambiado el propietario.	
Resultado esperado: Se imprime "Hay algunos permisos no recomendados"	**Resultado Obtenido:** Se imprime "Hay algunos permisos no recomendados"
Observaciones:	
Evaluación de la prueba: Satisfactorio	

Tabla 29: Pruebas 2: Permisos en las carpetas de datos.

Prueba 5	
Plug-in: pgbd1	**Nombre:** Contraseñas en blanco
Descripción: No hay ningún usuario con contraseñas en blanco.	
Resultado esperado: blankpass = 0, usuario = null	**Resultado Obtenido:** blankpass = 0, usuario = null
Observaciones:	
Evaluación de la prueba: Satisfactorio	

Tabla 30: Pruebas 1: Contraseñas en blanco.

Prueba 6	
Plug-in: pgbd1	**Nombre:** Contraseñas en blanco
Descripción: Se ha creado un usuario llamando: usarioprueba, sin contraseña que debe ser identificado por la consulta.	
Resultado esperado: Blankpass = 1, usuario = usuarioprueba	**Resultado Obtenido:** Blankpass = 1, usuario = usuarioprueba
Observaciones: Anexo 4	
Evaluación de la prueba: Satisfactorio	

Tabla 31: Pruebas 2: Contraseñas en blanco.

Prueba 7	
Plug-in: pgbd2	**Nombre:** Bases de datos con nombre de usuarios
Descripción: Solo se deja la BD postgres y el usuario postgres como BD con nombre igual a usuario del SGBD.	

Resultado esperado: usuario = postgres, cantidad = 1	**Resultado Obtenido:** usuario = postgres, cantidad = 1
Observaciones: Anexo 5	
Evaluación de la prueba: Satisfactorio	

Tabla 32: Pruebas 1: Bases de datos con nombre de usuarios.

Prueba 8	
Plug-in: pgbd3	**Nombre:** Súper usuarios en el SGBD
Descripción: Hay un solo súper usuario además del usuario de pruebas "usuarioprueba".	
Resultado esperado: cant_superusuarios = 2 usuario = usuarioprueba	**Resultado Obtenido:** cant_superusuarios = 2 usuario = usuarioprueba
Observaciones: Anexo 6	
Evaluación de la prueba: Satisfactorio	

Tabla 33: Pruebas 1: Súper usuarios en el SGBD.

Prueba 9	
Plug-in: pgbd4	**Nombre:** Permisos de actualización de System Catalog.
Descripción: Ningún usuario tiene el permiso para actualizar el System Catalog.	
Resultado esperado: super_act = 0 , nosuper_act = 0	**Resultado Obtenido:** super_act = 0 , nosuper_act = 0
Observaciones: Anexo 7	

Evaluación de la prueba: Satisfactorio

Tabla 34: Pruebas 1: Permisos de actualización de System Catalog.

Prueba 10	
Plug-in: pgbd5	**Nombre:** Contraseñas comunes.
Descripción: Se le ha colocado al usuario postgres la contraseña “postgres”.	
Resultado esperado: usuarios_passwd_triviales = postgres	**Resultado Obtenido:** Error en la ejecución.
Observaciones: psql informa de un error de sintaxis. Anexo 8	
Evaluación de la prueba: Insatisfactorio	

Tabla 35: Pruebas 1: Contraseñas comunes.

Prueba 11	
Plug-in: pgbd5	**Nombre:** Contraseñas comunes.
Descripción: Se le ha colocado al usuario postgres la contraseña “postgres”.	
Resultado esperado: usuarios_passwd_triviales = postgres	**Resultado Obtenido:** passwd_faciles = postgres
Observaciones: Anexo 9	
Evaluación de la prueba: Insatisfactorio	

Tabla 36: Pruebas 2: Contraseñas comunes.

Prueba 12	
Plug-in: pgbd5	**Nombre:** Contraseñas comunes.
Descripción: Se le ha colocado al usuario postgres la contraseña “postgres”.	

Resultado esperado: usuarios_passwd_triviales = postgres	**Resultado Obtenido:** usuarios_passwd_triviales = postgres
Observaciones: Anexo 10	
Evaluación de la prueba: Satisfactorio	

Tabla 37: Pruebas 3: Contraseñas comunes.

Prueba 13	
Plug-in: pgbd6	**Nombre:** Contraseñas almacenadas en claro.
Descripción: Se ha agregado mediante un usuario con una contraseña no cifrada, para que sea reconocido. El usuario es “usuarioprueba”	
Resultado esperado: usuario = usuarioprueba, en_claro = 1	**Resultado Obtenido:** usuario = usuarioprueba, en_claro = 1
Observaciones: Anexo 11	
Evaluación de la prueba: Satisfactoria	

Tabla 38: Pruebas 1: Contraseñas almacenadas en claro.

Prueba 14	
Plug-in: pgbd7	**Nombre:** Nombre de usuario como contraseña.
Descripción: Se le ha puesto a usuario prueba el password usuarioprueba, para que sea reconocido.	
Resultado esperado: usuario = usuarioprueba	**Resultado Obtenido:** usuario = usuarioprueba
Observaciones: Anexo 12	
Evaluación de la prueba: Satisfactoria.	

Tabla 39: Pruebas 1: Nombre de usuario como contraseña.

Antes de presentarle cada plug-in al cliente, jefe del equipo de desarrollo de AuditBD, se le realizaron las pruebas de aceptación anteriormente expuestas, para verificar las funcionalidades. Luego de llegar a un resultado satisfactorio, se le entregaron los complementos para que los avalara, mostrado en (Anexo 13).

Fase de Mantenimiento

En esta fase se desarrollan concurrentemente 2 tareas: mantener el sistema en funcionamiento y el desarrollo de nuevas iteraciones, propiciando a su vez, una disminución considerable en el tiempo de desarrollo del proyecto. En este caso, el cliente luego de analizar el listado de BP, solicitó la implementación de 9 de estas, sin incorporar ninguna otra durante el desarrollo del sistema.

Fase Muerte del Proyecto

La muerte del proyecto viene dada por varias causas, el cliente no tiene nada más que agregarle al sistema, el mismo no genera los beneficios esperados por el cliente o el costo se vuelve insostenible. En el primer caso, se procede a la documentación del sistema y no se realizan más cambios. En el caso de un proyecto de seguridad como el que se maneja la muerte del mismo debe llegar de manos de una de las 2 últimas razones, debido a que constantemente se crean o descubren nuevas vulnerabilidades y amenazas, por lo que es difícil que los clientes o desarrolladores se satisfagan completamente con lo desarrollado.

Conclusiones parciales

Como resultado de una profunda investigación y en la estructura de PostgreSQL específicamente, se obtuvo un listado conformado por un total de 30 BP para auditar la seguridad en el SGBD PostgreSQL.

Como implementación de algunas de las BP que conforman el listado, se obtuvieron un total de 9 plug-ins, 2 de ellos son de BP para el SO, los restantes 7 plug-ins, son de BP para la BD.

Con la implementación de los plug-ins se le ofrece soporte a la herramienta AuditBD para auditar las BD manejadas por el SGBD PostgreSQL.

Capítulo 3: Validación de la Solución

Introducción

En el presente capítulo, con el objetivo de obtener valoraciones sobre la investigación, se realiza una evaluación mediante encuestas a expertos en el tema. Las encuestas se realizaron a personas conocedoras del tema o vinculadas directamente con él. Partiendo de los resultados del trabajo: el listado de buenas prácticas, así como los plug-ins que son la implementación de estas buenas prácticas, se hace necesaria la validación del mismo, para ello se muestran los criterios de los expertos encuestados.

Métodos de validación:

Existen varios métodos de validación, por lo que se hace necesario un estudio de cada uno de ellos para seleccionar el método a emplear. Los métodos que se tuvieron en cuenta fueron.

- Comparación de los resultados de salida del modelo con los del sistema real.
- Método Delphi.
- Test de Turing.

Comparación de los resultados de salida del modelo con los del sistema real

Para poder hacer uso de este método, el sistema real debe existir, además se comparan las salidas del modelo con las del sistema, mediante algún método estadístico. El mayor inconveniente que tiene este método es que la mayoría de los procesos de salida no son estacionarios, provocando que estos tests no sean directamente aplicables (30).

Método Delphi

El método de validación Delphi consiste en seleccionar un grupo de expertos en el tema y circular entre ellos de manera individual un cuestionario que contenga las preguntas que se consideren más importantes sobre el tema en cuestión. Basándose

en las respuestas dadas a las cuestiones planteadas, se elaboran nuevos cuestionarios que van centrándose en temas más específicos, estos nuevos cuestionarios, son enviados a los expertos junto con las respuestas obtenidas en rondas anteriores. Estos pasos pueden ir repitiéndose hasta conseguir por parte del equipo de expertos una predicción de la respuesta del sistema.

En este método se excluyen las discusiones cara a cara de los miembros del equipo de expertos y su mayor desventaja es que consume mucho tiempo, por eso, en la mayoría de los casos no se realizan muchas iteraciones en cuanto a la elaboración de nuevos tests y su análisis.

Test de Turing

Alan Turing sugirió este método como un test de inteligencia artificial. En este test, a un experto, o grupo de expertos, se le presentan resúmenes o informes de resultados de ejecución del sistema y del modelo, a los que se les ha dado el mismo formato. Estos informes se reparten aleatoriamente a los ingenieros y administradores del sistema, para ver si son capaces de discernir cuáles son los reales del sistema y cuales la imitación resultado de la simulación. Si los expertos no son capaces de distinguir entre ambos, se puede concluir que no hay evidencias para considerar inadecuado al modelo. Si descubren diferencias las respuestas sobre lo que encuentran inconsistente se puede utilizar para realizar mejoras en el modelo.

Se puede considerar que este método es el inverso al método de Delphi. En el test de Turing se consulta a los expertos para ver si son capaces de identificar las respuestas del sistema, mientras que en el de Delphi se pregunta a los expertos para que predigan las respuestas del sistema.

Aunque este test parece muy intuitivo, hay muy pocos informes de su uso, ya que requiere un esfuerzo considerable para formatear las medidas de ejecución del sistema a la hora de crear el informe que se da a los expertos. Otra dificultad está en ajustar las medias del sistema real ya que en ellas intervienen elementos que no se han considerado en el modelo. Por último, este test requiere un análisis estadístico por parte del grupo de expertos para determinar si hay diferencias significativas entre el informe real y el simulado (30).

Validación por el método Delphi

Delphi o experto como también se le conoce es el método que más se ajusta a las necesidades del trabajo, puesto que en él, el panel de expertos, mediante las encuestas que le son realizadas, de forma anónima, son quienes deben predecir los resultados a obtener con el trabajo desarrollado.

Este método puede tener tantas iteraciones como sean necesarias, pero en este caso, debido a la dinámica de la UCI se hace engorroso y largo el proceso de hacer más de una ronda de preguntas, es por ello que el análisis de los resultados se realiza en una primera ronda.

1. Formulación del problema

Primeramente de definen los criterios (C) a evaluar por los especialistas, estos deben ser, medibles, claros e independientes. Los criterios que se tuvieron en cuenta fueron los siguientes:

C1: Importancia de la seguridad de los datos.

C2: Importancia de las buenas prácticas.

C3: Influencia de la seguridad de los datos en la calidad.

C4: Mejoras de seguridad utilizando buenas prácticas.

C5: Contribución de las buenas prácticas a la seguridad de los datos.

C6: Contribución de los plug-ins al incremento de la calidad de las auditorías.

C7: Evaluación del resultado de la investigación.

Estos criterios constituyen la base para la confección del cuestionario que se les aplica a los expertos.

2. Proceso de selección de expertos

Primeramente se debe conocer que los expertos son aquellas personas que tiene un elevado dominio del tema que se está analizando, en este caso las buenas prácticas para el SGBD PostgreSQL. Además, son capases de ejercer criterios concluyentes del trabajo, realizar recomendaciones que consideren, sean de ayuda para el enriquecimiento del mismo y estar dispuestos a participar en la validación del trabajo.

En principios se tienen en cuenta un total de 7 candidatos a expertos, los cuales desempeñan los siguientes roles en la UCI:

- Experto 1: Jefe del Proyecto de Personalización de PostgreSQL del Dpto. de PostgreSQL del Centro de Tecnologías de Gestión de Datos (DATEC).
- Experto 2: Líder del proyecto LabSI (Laboratorio de Seguridad Informática).
- Experto 3: Jefe del grupo de Replicación de Datos del Dpto. de PostgreSQL.
- Experto 4: Arquitecto de Datos del Proyecto ERP.
- Experto 5: Jefa del Departamento de PostgreSQL del Centro y profesora de la asignatura Sistemas de Bases de Datos.
- Experto 6: Subdirector de Formación del Centro Gobierno Electrónico y profesor de las asignaturas Sistemas Operativos y Seguridad Informática.
- Experto 7: Jefe del grupo de migración en PostgreSQL en el centro de BD DATEC.

Para el proceso de selección de los expertos se ha analizado un aspecto de interés por parte de los autores de este trabajo de diploma, el nivel de competencia de los encuestados en el tema que se analiza.

Para determinar cuáles de los candidatos participará en la evaluación de la solución, se calculó el coeficiente de competencia K, haciendo uso de la siguiente fórmula matemática.

$$K = \frac{Kc + Ka}{2}$$

Donde:

Kc: Es el coeficiente de conocimiento.

Ka: Es el coeficiente de argumentación.

Para calcular el Kc y Ka se realiza una encuesta (Anexo 14) a los candidatos a expertos.

Para calcular el Kc se le solicita al posible experto que de su criterio sobre los conocimientos que cree posee sobre el tema. Para esto se utiliza un rango del 1 al 10, considerando que 1 es no tener ningún dominio del tema y 10 es tener pleno dominio del tema. Posteriormente este valor obtenido se multiplica por 0.1 para obtener el coeficiente en un rango de 0 a 1.

Para el cálculo del Ka, el candidato debe clasificar el grado de competencia que posee sobre los aspectos o bibliografías consultadas en cuanto al tema de la investigación. Cada nivel de clasificación tiene un valor y la suma de los valores marcados por cada criterio será el Ka del candidato a experto. Para realizar el cálculo de Ka se hace uso del cuestionario definido por las autoras Yilena Borrero Luzúa y Yaima Viltres Cisnero en su trabajo de diploma “Propuesta de un Proceso de Selección de Roles y Personal con sus Niveles de Competencia para Proyectos Multimedia”. (Anexo 15: tabla auxiliar para el cálculo del grado de argumentación del experto (Ka)).

Después de haber calculado el Ka, se evalúa cuales de los candidatos pueden pasar a ser expertos, analizando los siguientes aspectos:

- Si $0.8 < K < 1.0$, el coeficiente de competencia es alto.
- Si $0.5 < K < 0.8$, el coeficiente de competencia es medio.
- Si $K < 0.5$ el coeficiente de competencia es bajo.

Es recomendable incluir en el grupo de expertos aquellos que su coeficiente de competencia sea medio o alto.

A continuación se muestra una tabla donde se relacionan los expertos con sus respectivos valores de Kc, Ka y K.

	Expertos	Kc	Ka	K
1	Ing. Marcos Ortiz Valmaseda	0.7	0.95	0.82
2	Ing. Rogfel Thompson Martínez	1	0.88	0.94
3	Ing. Leonel Fuentes Marrero	0.8	0.86	0.83
4	Ing. Ariel Torres Gálvez	0.7	0.98	0.84
5	Ing. Yudisney Vázquez Ortiz	0.8	0.77	0.78
6	Ing. Carlos Y. Hidalgo García	0.8	0.74	0.77
7	Ing. Héctor Miguel Beltrán Lugo	0.5	0.79	0.64

Tabla 40: Coeficiente de competencia de los expertos seleccionados.

Después de analizado el coeficiente de competencia de cada uno de los candidatos a expertos, se obtuvo como resultado que ninguno de ellos tiene su coeficiente de competencia bajo, por lo que los 7 están aptos para conformar el panel de expertos. De los seleccionados 4 tienen su coeficiente de competencia alto y 3 lo tienen medio.

Los nombres y cargos desempeñados por los expertos seleccionados para conformar el panel, se muestran a continuación:

- Experto 1: Ing. Marcos Ortiz Valmaseda: Jefe del Proyecto de Personalización de PostgreSQL del Dpto. de PostgreSQL del Centro de Tecnologías de Gestión de Datos (DATEC).
- Experto 2: Ing. Rogfel Thompson Martínez: Líder del proyecto LabSI (Laboratorio de Seguridad Informática).
- Experto 3: Ing. Leonel Fuentes Marrero: Jefe del grupo de Replicación de Datos del Dpto. de PostgreSQL.
- Experto 4: Ing. Ariel Torres Gálvez: Arquitecto de Datos del Proyecto ERP.
- Experto 5: Ing. Yudisney Vázquez Ortiz: Jefa del Departamento de PostgreSQL del Centro y profesora de la asignatura Sistemas de Bases de Datos.
- Experto 6: Ing. Carlos Y. Hidalgo García: Subdirector de Formación del Centro Gobierno Electrónico y profesor de las asignaturas Sistemas Operativos y Seguridad Informática.

- Experto 7: Ing. Héctor Miguel Beltrán Lugo: Jefe del Grupo de Migración en PostgreSQL en el Centro de BD DATEC.

3. Encuesta elaborada

La estructura de la encuesta (Anexo 16) elaborada fue diseñada por los autores de la tesis en dependencia de sus necesidades.

En la encuesta, primeramente se solicitan los datos personales de los expertos y posteriormente se originan 9 preguntas, ya sean de tipo contable o abierto, permitiendo estas últimas que los expertos emitan sus criterios y hagan recomendaciones con el objetivo de mejorar los resultados de la investigación.

Las preguntas de tipo contable se crearon con el objetivo de sacar estadísticas y mostrar gráficamente los resultados arrojados con las encuestas, además, en este tipo de preguntas la escala para sus respuestas se dividió en 5, sin importar si esta división está dada en forma cualitativa (_Adecuada, _Poco Adecuada) o cuantitativa (%).

Análisis de las encuestas

Para analizar las encuestas realizadas a los expertos se tuvieron en cuenta 2 criterios de evaluación, los criterios cualitativos, quedando identificados por (Muy alta, alta, media, baja y muy baja) y los criterios cuantitativos expresados en (100%, 75%, 50%, 25%, 0%). A cada uno de estos criterios se les otorgó una puntuación entre 1 y 5 en dependencia de los valores asignados por los expertos a las preguntas contestadas en el cuestionario para su posterior análisis.

Criterios de Evaluación			Puntuación
	Criterios Cualitativos	*Criterios Cuantitativos (%)*	
1	Muy Alta	100	5
2	Alta	75	4
3	Media	50	3
4	Baja	25	2

5	Muy baja	0	1

Criterios de Evaluaciones

Opiniones de los expertos encuestados

Experto 1: Ing. Marcos Ortiz Valmaseda

El experto considera que es sumamente importante que se desarrollen soluciones como ésta para darle un uso eficiente al SGBD PostgreSQL, aprovechando al máximo sus características en materia de seguridad, además considera que la seguridad de los datos en las aplicaciones informáticas tiene una elevada repercusión (5 puntos) y su influencia en la calidad de los proyectos que se desarrollan en la UCI es 100% (5 puntos).

Con respecto a la propuesta de buenas prácticas, opina que recoge en un 75% (4 puntos) el mínimo de elementos necesarios para contribuir a la seguridad de los datos, analizando el software de BD y del sistema operativo y en cuanto a la aplicación de esta propuesta, considera que disminuirá en un 75 % (4 puntos) el esfuerzo en el momento de corregir brechas de seguridad en el SGBD PostgreSQL, por tanto, su importancia es alta (4 puntos), pero esto depende fundamentalmente del nivel de experiencia que tengan los administradores de BD.

El experto cree que la puesta en práctica de los plug-ins que se desarrollaron en el trabajo, tendrá un alto impacto (4 puntos) en la calidad de las auditorías que se le realizan al SGBD PostgreSQL.

De manera general el experto, basándose en la posibilidad de aplicación que él considera tengan los plug-ins y el soporte que le puedan proporcionar a la herramienta AuditBD, otorga una evaluación de 5 puntos al resultado del trabajo, recomendando profundizar un poco más en temas como:

Evitar ataques de Inyección SQL (Revisar documento Advanced SQL Injection)

1. Encriptación de datos (pgcrypto)
2. Seguridad a nivel de registros (Veil)
3. Auditoría (table_log)

Experto 2: Rogfel Thompson Martínez

El experto le confiere una importancia muy alta (5 puntos) a la seguridad en los sistemas informáticos así como a las buenas prácticas en la seguridad de los datos y cree que la seguridad informática tiene una influencia de un 100% (5 puntos) en la calidad del software que se desarrolla en la UCI.

Es del criterio que con la aplicación de las buenas prácticas obtenidas se disminuirá en gran medida el esfuerzo al momento de corregir brechas de seguridad en el SGBD PostgreSQL, a 75% (4 puntos) aproximadamente. También opina que el listado de buenas prácticas propuestas, recoge en un 75 % (4 puntos) los elementos básicos para la seguridad de los datos, aunque siempre se queda abierta una brecha a nuevas formas de seguridad informática.

Piensa que con la propuesta de plug-ins desarrollados, se incrementará en un 100% la calidad de las auditorías realizadas a PostgreSQL y califica, en la escala de 1 a 5, con un 5 el resultado del trabajo en cuanto a posibilidad de aplicación y complemento de la herramienta AuditBD.

El experto opina que el uso de buenas prácticas en BD y en este caso específico de PostgreSQL permite a los administradores de una forma disciplinada eliminar y minimizar potenciales ataques al sistema y malos funcionamientos de este. Esta es una rama de investigación de la seguridad informática no tratada en la universidad, la cual le permite mejorar la seguridad de las BD de PostgreSQL. Se ha realizado una buena investigación en cuanto a la seguridad y el uso de buenas prácticas en este gestor. Estos temas de investigación de Seguridad Informática requieren un seguimiento debido a que este campo posee una gran evolución en sus temas.

Experto 3: Leonel Fuentes Marrero

El experto tiene el criterio: El trabajo posee una importancia considerable debido a que el SGBD PostgreSQL es el líder de los SGBD libres, además de que existe en el país una marcada intención de usar el mismo en la mayoría del software que se construye y proyecta a diario.

El experto considera que la seguridad de los datos en las aplicaciones informáticas tiene muy alta repercusión (5 puntos) y opina que las buenas prácticas tienen una elevada (4 puntos) importancia en la seguridad de los datos. Cree que la seguridad de los datos tienen una influencia de 75% (4 puntos) en la calidad del software que se desarrolla en la UCI. Considera que las buenas prácticas obtenidas como resultado de la investigación reducirán en al menos un 75% (4 puntos) el esfuerzo en el momento de corregir brechas de seguridad en un SGBD PostgreSQL, y recogen en un 75% (4 puntos) los elementos necesarios para contribuir a la seguridad de los datos en el mencionado gestor.

El experto cree que el resultado de la investigación incrementará la calidad de las auditorías realizadas a PostgreSQL en la universidad a un nivel alto (4 puntos), y le otorgó una evaluación máxima a la solución en una escala de 5 en cuanto a posibilidad de aplicación y complemento de la herramienta AuditBD.

Experto 4: Ariel Torres Gálvez

El experto opina que con la lectura del presente trabajo se puede lograr percibir la profundidad de la investigación sobre PostgreSQL y además conocer los distintos niveles de seguridad que se puede tener para poder afirmar que nuestra base de datos se encuentra segura ante malas acciones. La investigación desarrollada puede aportar grandes beneficios puestos que, abarca los fundamentales aspectos a tener en cuenta para todo administrador de base de datos sobre PostgreSQL en cuanto a seguridad de la información. Considera que la seguridad de los datos tiene una muy alta (5 puntos) importancia, además cree que este factor influye en un 75% (4 puntos), en la calidad del software desarrollado en la UCI

El experto considera que las buenas prácticas tienen una importancia muy alta (4 puntos) en la seguridad de los datos y que la puesta en práctica de ellas, disminuirá en un 75% (4 puntos), el esfuerzo en el momento de corregir brechas de seguridad en el SGBD PostgreSQL. Además, opina que el listado de buenas prácticas recoge en un 75% (4 puntos), los elementos mínimos necesarios para contribuir a la seguridad de los datos, analizando el software de BD y del sistema operativo. En este punto, el experto considera que sería bueno agregar todos los niveles de seguridad, ya que solamente se llega hasta nivel de base de datos, pero de ahí hacia abajo no se hace

alusión en el trabajo presentado, por ejemplo nivel de esquemas, nivel de objetos de base de datos entiéndase esto como: (tablas, funciones, secuencias,…), y por últimos el acceso a la información.

Con la propuesta de los plug-ins, el experto piensa que se incrementará en un término alto (4 puntos), la calidad de las auditorías que se realizan al SGBD PostgreSQL y en un rango de puntuación del 1 al 5 otorga una puntuación de 5 al trabajo realizado basándose en los criterios: Posibilidad de Aplicación y Complemento a la herramienta AuditBD.

Experto 5: Yudisney Vázquez Ortiz

La experta considera que el trabajo desarrollado constituye un resultado positivo que favorece el incremento de la seguridad en las BD. Además, tiene una gran utilidad ya que incluye al gestor PostgreSQL, entre los soportados por el Sistema AuditBD, encargado de la realización de auditorías a los datos; constituye un paso de avance en el proceso de difundir su utilización y explotación en el país.

Desde su punto de vista, la seguridad de los datos tiene una importancia muy alta (5 puntos) en las aplicaciones informáticas. Le otorga una clasificación de muy alta (5 puntos) a las buenas prácticas en la seguridad de los datos y considera que con la propuesta de buenas prácticas se disminuirá en un 75% (4 puntos) el esfuerzo en el momento de corregir brechas de seguridad en el SGBD PostgreSQL y piensa que el listado de buenas prácticas obtenido como resultado de la investigación, recoge en un 75% (4 puntos), los elementos mínimos necesarios para contribuir a la seguridad de los datos, analizando el software de BD y del sistema operativo.

La experta opina que la seguridad de los datos influye en un 75% (4 puntos), en la calidad del software desarrollado en la UCI y considera que con la propuesta de los plug-ins se incrementará en gran medida (4 puntos), incrementará la calidad de las auditorías que se le realizan al SGBD PostgreSQL.

La experta basándose en criterios como la posibilidad de aplicación y el complemento de la aplicación informática AuditBD, otorga una puntuación de 5 puntos al resultado del trabajo.

Experto 6: Carlos Y. Hidalgo García

El experto considera que el trabajo ofrece al administrador de BD o al especialista que se desempeñe en un rol afín, la posibilidad de contar con una guía para la revisión, control y mantenimiento de los aspectos básicos necesarios para garantizar el correcto funcionamiento del SGBD así como le permite ofrecer una seguridad adecuada en el tratamiento de la información que se maneje en el SGBD para las aplicaciones que se desarrollen.

El experto opina que la seguridad de los datos tiene una importancia muy alta (5 puntos) en los sistemas informáticos y considera que las buenas prácticas tienen una importancia muy alta (5 puntos) en la seguridad de los datos y que el listado de buenas prácticas obtenido como resultado de la investigación disminuirá en un 75% (4 puntos) el esfuerzo en el momento de corregir brechas de seguridad, porque el otro 25% depende de otras medidas que se pueden tomar en cuanto a la protección de los accesos a los datos y a la utilización de principios y reglas a seguir para lograr una implementación segura en cuanto al código de las aplicaciones que se desarrollen (capa de acceso a datos) y su comunicación con el SGBD.

Considera además que la seguridad de los datos influye en un 75% (4 puntos) en la calidad del software desarrollado en la UCI, porque el otro 25% depende de la arquitectura, el IDE de desarrollo, la calidad en la captura de requisitos así como de las buenas prácticas en cuanto a la seguridad en la programación.

Desde su punto de vista el listado de buenas prácticas recoge en un 100% (5 puntos), los elementos mínimos necesarios para contribuir a la seguridad de los datos, analizando el software de BD y del sistema operativo. Considera que la propuesta de los plug-ins desarrollados incrementa en gran medida (4 puntos) la calidad de las auditorías que se le realizan al SGBD PostgreSQL y basándose en los criterios de posibilidad de aplicación y complemento al software AuditBD, otorga una puntuación de 4 puntos a la solución obtenida.

Experto 7: Héctor Miguel Beltrán Lugo

El experto considera que en todos los sistemas es de vital importancia la seguridad, una investigación sobre buenas prácticas y plug-ins que puedan ayudar a mejorar la

seguridad de los SGBD siempre es importante debido a que beneficia nuestras aplicaciones, garantizando la integridad y disponibilidad de la información que manejan nuestros sistemas.

La importancia que le concede el experto a la seguridad en los sistemas informáticos es alta (4 puntos).

Desde su punto de vista las buenas prácticas tienen una importancia alta (4 puntos) en la seguridad de los datos y con la propuesta de buenas prácticas que se obtuvo como resultado de la investigación, considera que disminuirá en un 75% (4 puntos), el esfuerzo en el momento de corregir brechas de seguridad en el SGBD PostgreSQL, además opina que este listado de buenas prácticas recoge en un 75% (4 puntos), los elementos mínimos necesarios para contribuir a la seguridad de los datos, analizando el software de BD y del sistema operativo.

Para el experto la seguridad de los datos influye en un 75% (4 puntos) en la calidad del software desarrollado en la UCI y para él, la propuesta de los plug-ins incrementará en alta medida (4 puntos), la calidad de las auditorías que se le realizan al SGBD PostgreSQL.

El experto basándose en criterio como la posibilidad de aplicación y el nivel de complemento a la herramienta AuditBD, le otorga una puntuación de 5 puntos al resultado del trabajo.

Resultados del análisis de las encuestas a los expertos.

En el presente capítulo, luego de recogidas y analizadas cada una de las opiniones de los expertos, se muestran los resultados de la validación de la solución. Los resultados se muestran en forma de gráficos de barras y tablas.

Criterios	Exp1	Exp2	Exp3	Exp4	Exp5	Exp6	Exp7	PP	MP	PA
ISD	5	5	5	5	5	5	4	4.8	5	97.1
IBP	4	4	4	5	5	5	4	4.4	5	88.5
ISDC	5	5	4	5	4	4	4	4.4	5	88.5
MSBP	4	4	4	4	4	4	4	4	4	100

CBPSD	4	4	4	4	4	5	4	4.1	5	82.8
CPCA	4	4	4	4	4	4	4	4	4	100
EI	5	5	5	5	5	4	5	4.8	5	97.1

Tabla 41: Resumen de la Validación de Expertos.

Leyenda:

Exp: Experto

PP: Promedio de puntuación

MP: Máxima puntuación

PA: Por ciento de aceptación

ISD: Importancia de la seguridad de los datos.

IBP: Importancia de las buenas prácticas.

ISDC: Influencia de la seguridad de los datos en la calidad.

MSBP: Mejoras de seguridad utilizando buenas prácticas.

CBPSD: Contribución de las buenas prácticas a la seguridad de los datos.

CPCA: Contribución de los plug-ins al incremento de la calidad de las auditorías.

EI: Evaluación del resultado de la investigación.

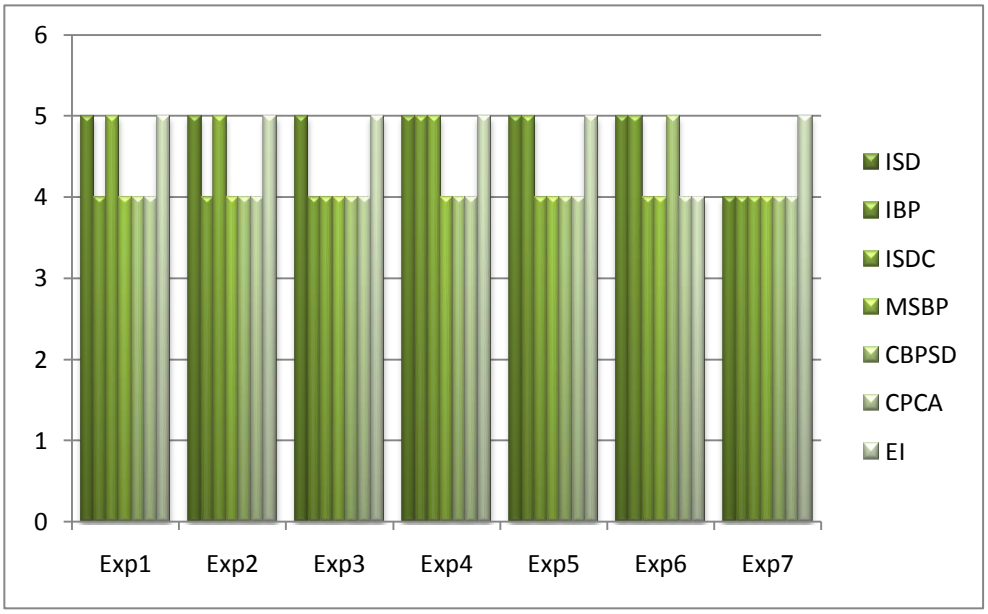


Gráfico 2: Resultados de la Validación

Analizando la tabla # 3 donde se muestran los resultados de la validación, se concluye que las evaluaciones otorgadas por los expertos son satisfactorias, encontrándose en la puntuación de 4 y 5 puntos. El promedio de puntuación está siempre por encima de los 4 puntos y analizando el por ciento de aceptación de cada uno de los criterios, se concluye que el trabajo cuenta con un 93.4 % de aceptación.

Evaluación de la concordancia entre expertos

Con el objetivo de aportarle un mayor peso al resultado de la validación se decidió determinar la concordancia de criterios entre los expertos, haciendo uso del cálculo del coeficiente de concordancia Kendall (W) (33).

Para el cálculo del W se emplea el programa estadístico Statistical Product and Service Solutions (SPSS), considerando como valores de entrada, los datos obtenidos como resultado de las encuestas de los expertos.

Los valores del W deben oscilar entre 0 y 1 ($0 < W < 1$), mientras más próximos se encuentren los valores a 1, mayor será el nivel de concordancia entre los criterios de los expertos. La concordancia se considera aceptable cuando los valores obtenidos como resultado del cálculo del W están por encima de 0.5 ($W > 0.5$).

Luego de obtenido el valor resultante del W, no basta con conocer si es significativamente distinto de 0, se debe hacer unos de las pruebas de hipótesis para determinar que tan significativa es la concordancia.

Hipótesis:

H0: No hay concordancia (Hipótesis nula).

H1: Hay concordancia (Hipótesis alternativa).

Luego de obtenido el resultado del W debe analizarse el tamaño de la muestra (N). En dependencia de N, será el proceso que se aplique para saber cuan significativa es la concordancia entre los criterios. Si N<=7, las muestras son pequeñas y se calcula el Nivel de Significación (s), en caso de que N>7, se utiliza el cálculo de Chi Cuadrado (x^2).

Si se hace uso del "s" y su valor obtenido con el empleo del SPSS es menor que 0.05, se rechaza H0 y se considera el valor de W como significativo. En caso de ser necesario el cálculo de x^2 por presentar más de 7 criterios, debe tenerse en cuanta que si el valor de x^2 es mayor que 0.05, entonces se rechaza la H0 y se considera significativo el valor del W.

Cálculo del Kendall

N	7
Kendall's W(a)	,511
Chi-Square	21,450
df	6
Asymp. Sig.	,002

Tabla 42: Cálculo de la Concordancia

El valor N=7 constituye el número de criterios definidos previamente. El resultado obtenido del W se encuentra en el rango de (0.5 < W < 1), por lo que se considera aceptable y el nivel de significación (s) es de 0.02 rechazándose la hipótesis nula, por lo que se concluye que existe una concordancia significativa entre los criterios de los expertos.

Conclusiones del capítulo

Obtenido el listado de plug-ins se validó su funcionamiento mediante pruebas de caja gris, permitiendo comprobar las funciones, con algunas vistas al código que facilitando una revisión y rectificación rápida de los errores encontrados.

El listado de buenas prácticas obtenidas como solución se validó haciendo uso del método Delphi, obteniéndose como resultado un 93.4 % de aceptación de la solución.

Conclusiones

Para el desarrollo del trabajo se analizaron con profundidad diferentes temas como la seguridad de los datos, los SGBD más usados, se analizó detalladamente la estructura del SGBD PostgreSQL y las principales herramientas informáticas que existen en la actualidad para auditar los datos, obteniéndose como resultado, que existen varias aplicaciones de software destinadas a este fin, pero ninguna de ellas brinda soporte para PostgreSQL.

Los principales resultados obtenidos con la culminación del trabajo son:

El listado de buenas prácticas, conformado por un total de 30 buenas prácticas, basadas fundamentalmente en las configuraciones del SO y del SGBD PostgreSQL, haciendo mención también a algunas de uso genérico.

La elaboración de este listado de buenas prácticas proporciona considerables mejoras de seguridad para el SGBD PostgreSQL y constituye a su vez una guía para la revisión, control y mantenimiento de los aspectos básicos necesarios para garantizar el correcto funcionamiento del SGBD.

La obtención de los plug-ins que brindarán soporte para PostgreSQL a la herramienta AuditBD encargada de desarrollar auditorías a los SGBD.

El desarrollo de los plug-ins está basado en algunas de estas buenas prácticas mencionadas anteriormente y se desarrollaron un total de 7 plug-ins 5 destinados a las BD y 2 de ellos al SO.

Recomendaciones

Debido a que la seguridad constituye un proceso en constante evolución se recomienda el desarrollo periódico de nuevos plug-ins y la actualización de los ya elaborados para la herramienta AuditBD.

Profundizar en el estudio de las diferentes funcionalidades de PostgreSQL con el objetivo de proponer nuevas buenas prácticas.

Desarrollar los plug-ins de las BP propuestas en este trabajo y que no se encuentran concretadas en complementos para AuditBD.

Bibliografía

1. *Calidad del Software.* **Lovelle, Juan Manuel Cueva.** 1999.

2. **Date, C. J.** *Introducción a sistemas de bases de datos.* s. l. : Pearson Educación, 2001.

3. **Asenjo, Jorge Sánchez.** *Sistemas Gestores de Bases de Datos.* 2005.

4. **Aguirre, Jorge Ramió.** *Libro Electrónico de Seguridad Informática y Criptografía.* 2006.

5. **García, Rosa María Mato.** *Diseño de Bases de Datos .* 1999.

6. *Conferencia 1: Introducción a la asignatura de sistemas de bases de datos.Introducción a los Sistemas de Bases de Datos.El Modelo Entidad-Relación (ER).* 2009.

7. The PostgreSQL Licence (TPL). [En línea] http://www.opensource.org/licenses/postgresql.

8. MySQL Legal Policies. [En línea] http://www.mysql.com/about/legal .

9. **Corporation, Microsoft.** *Microsoft SQL Server 7 System Administration Training kit.* 1999.

10. **Litchfield, David.** *The Database Hacker's Handbook: Defending Database Servers .* 2005.

11. **Grupo Asesoría Técnica.** *Reporte 2010-6, Resumen del reporte de tecnologías más usadas en la UCI.* 2010.

12. **Group, The PostgreSQL Global Development.** *PostgreSQL 8.4.0 Documentation.* 2009.

13. *PostgreSQL, Una Alternativa de DBMS Open Source.* **Espinoza, Humberto.** 2005.

14. **Denzer, Patricio.** PostgreSQL. [En línea] 2002. http://profesores.elo.utfsm.cl/~agv/elo330/2s02/projects/denzer/informe.pdf.

15. **PostgreSQL Global Development Group.** http://www.postgresql.org/docs/8.4/static/release-8-4.html. [En línea]

16. **Berkus, Josh.** PostgreSQL. *PostgreSQL 9.0 Beta 1 Now Available.* [En línea] http://www.postgresql.org/about/news.1198.

17. **Española, Real Academia.** DICCIONARIO DE LA LENGUA ESPAÑOLA. [En línea] 2009. http://buscon.rae.es/draeI/.

18. **Aneiro Rodríguez, Lázaro Orlando.** *Elementos de Arquitectura y Seguridad Informática.* 2001.

19. *Safe data is happy data.* **Berkus, Josh.** 2008.

20. **Martinez, Rafael.** PostgreSQL-es.org. [En línea] http://www.postgresql-es.org/node/218/840.

21. **Barzanallana, Rafael.** *Metodologías de desarrollo de software.* 2006.

22. **Letelier, Patricio.** Laboratorio Docente de Computación - Universidad Simón Bolívar. [En línea] http://www.ldc.usb.ve/~abianc/materias/ci4712/ProcesoSW-Letelier.pdf.

23. **Ltd, Next Generation Security Software.** NGS Software. [En línea] 2009. http://www.ngssoftware.com/services/software-products/Database-Security.aspx.

24. Application Security Inc. [En línea] http://www.appsecinc.com/products/appdetective/.

25. DB Audit Home Page. [En línea] http://www.softtreetech.com/dbaudit/.

26. **Casares, Claudio.** *Tutorial de SQL.*

27. **Veeraraghavan, Sriranga.** *Sams Teach Yourself Shell Programming in 24 Hours.* 1999.

28. **W3C.** Extensible Markup Language (XML). [En línea] 2010. http://www.w3.org/XML/.

29. **Labañino Griñan, Daysel y Sánchez Enrique, Yusniel.** *Procedimiento para la Evaluación de la Usabilidad en los Softwares de Gestion sobre Plataforma Web en la Facultad 2.* 2009.

30. **Pilone, Dan y Miles, Russ.** *Head First Software Development.*

31. *Conferencia 8: Dócimas no Paramétricas. Bondad de Ajuste. Homogeneidad de la muestra. Análisis para evaluar concordancia.* **Aplicada, DDC Matemática.** 2010.

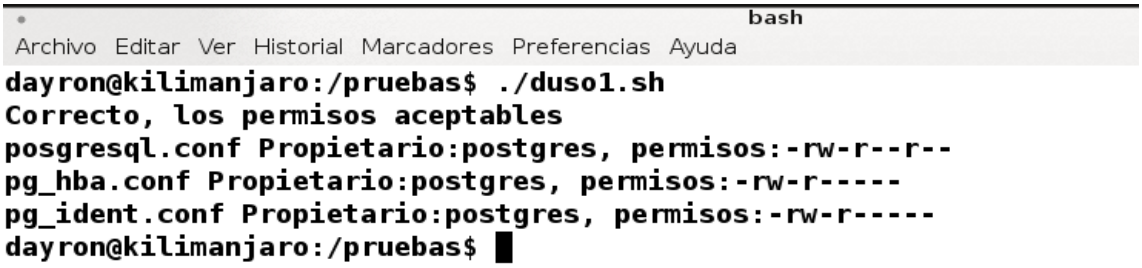
32. The GNU Operating system. [En línea] http://www.gnu.org/philosophy/free-sw.es.html.

33. Filosofía del Proyecto GNU. [En línea] http://www.gnu.org/philosophy/categories.es.html#ProprietarySoftware.

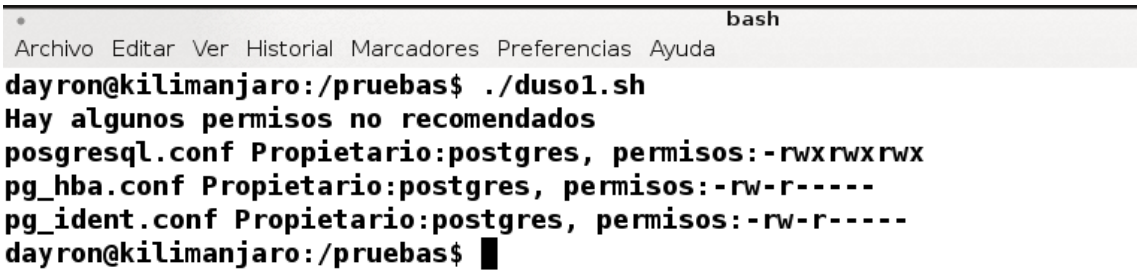
34. Productos de inteligencia empresarial: software de administración de bases de datos. [En línea] http://www.sybase.es/products/.

Anexos

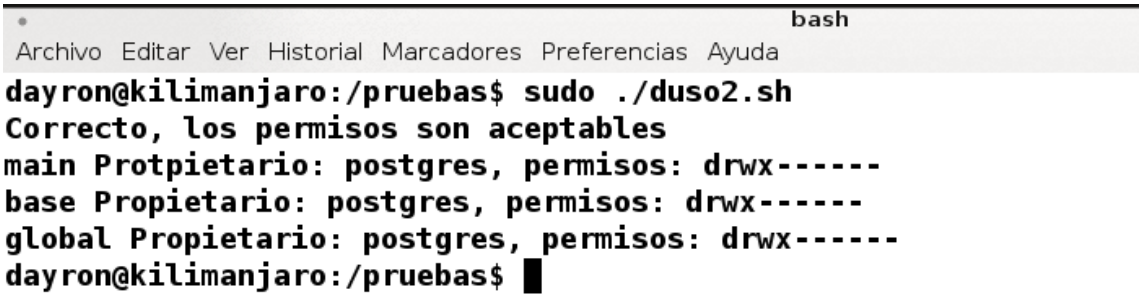
Anexo 1: Prueba 1 del plug-in duso1“Permisos en los ficheros de configuración”.



Anexo 2: Prueba 2 del plug-in duso1 “Permisos en los ficheros de configuración”.



Anexo 3: Prueba 1 duso2 “Permisos en las carpetas de datos”.



Anexo 4: Prueba 2 del plug-in pgbd1 “Contraseñas en blanco”.

```
bash
Archivo  Editar  Ver  Historial  Marcadores  Preferencias  Ayuda
dayron@kilimanjaro:/pruebas$ psql -d test -U dba -h localhost -f pgbd1.sql
 blankpass
-----------
         1
(1 fila)

    usuario
--------------
 usuarioprueba
(1 fila)

dayron@kilimanjaro:/pruebas$
```

Anexo 5: Prueba 1 del plug-in pgbd2 “Bases de datos con nombre de usuarios”.

```
bash
Archivo  Editar  Ver  Historial  Marcadores  Preferencias  Ayuda
dayron@kilimanjaro:/pruebas$ psql -d test -U dba -h localhost -f pgbd2.sql
 usuario
----------
 postgres
(1 fila)

 cantidad
----------
        1
(1 fila)

dayron@kilimanjaro:/pruebas$
```

Anexo 6: Prueba del plug-in pgbd3, “Súper usuarios en el SGBD”.

```
bash
Archivo  Editar  Ver  Historial  Marcadores  Preferencias  Ayuda
dayron@kilimanjaro:/pruebas$ psql -d test -U dba -h localhost -f pgbd3.sql
 cant_super_usuario
--------------------
                  2
(1 fila)

 super_usuario
--------------
 dba
 usuarioprueba
(2 filas)

dayron@kilimanjaro:/pruebas$
```

Anexo 7: Prueba 1 del plug-in pgbd4 “Permisos de actualización de System Catalog”.

```
bash
Archivo  Editar  Ver  Historial  Marcadores  Preferencias  Ayuda
dayron@kilimanjaro:/pruebas$ psql -d test -U dba -h localhost -f pgbd4.sql
 super_act
-----------
         0
(1 fila)

 nosuper_act
-------------
           0
(1 fila)

dayron@kilimanjaro:/pruebas$
```

Anexo 8: Prueba 1 del plug-in pgbd5 “Contraseñas triviales”.

```
bash
Archivo  Editar  Ver  Historial  Marcadores  Preferencias  Ayuda
dayron@kilimanjaro:/pruebas$ psql -d test -U dba -h localhost -f pgbd5.sql
psql:pgbd5.sql:1: ERROR:  error de sintaxis en o cerca de «passwd»
LÍNEA 1: ...R passwd = 'md5049b3b350621f8ff1fdd85c0f4eb1bba'  passwd = '...
                                                             ^
dayron@kilimanjaro:/pruebas$
```

Anexo 9: Prueba 2 del plug-in pgbd5 “Contraseñas triviales”.

```
bash
Archivo  Editar  Ver  Historial  Marcadores  Preferencias  Ayuda
dayron@kilimanjaro:/pruebas$ psql -d test -U dba -h localhost -f pgbd5.sql
 passwd_faciles
----------------
 postgres
(1 fila)

dayron@kilimanjaro:/pruebas$
```

Anexo 10: Prueba 3 del plug-in pgbd5 “Contraseñas triviales”.

```
bash
Archivo  Editar  Ver  Historial  Marcadores  Preferencias  Ayuda
dayron@kilimanjaro:/pruebas$ psql -d test -U dba -h localhost -f pgbd5.sql
 usuarios_passwd_triviales
---------------------------
 postgres
(1 fila)

dayron@kilimanjaro:/pruebas$
```

Anexo 11: Prueba 1 del plug-in pgbd6 “Contraseñas almacenadas en claro”.

```
bash
Archivo  Editar  Ver  Historial  Marcadores  Preferencias  Ayuda
dayron@kilimanjaro:/pruebas$ psql -d test -U dba -h localhost -f pgbd6.sql
    usuario    | passwd
---------------+---------
 usuarioprueba | usuario
(1 fila)

 en_claro
----------
        1
(1 fila)

dayron@kilimanjaro:/pruebas$
```

Anexo 12: Prueba 1 del plug-in pgbd7 “Nombre de usuario como contraseña”.

```
bash
Archivo  Editar  Ver  Historial  Marcadores  Preferencias  Ayuda
dayron@kilimanjaro:/pruebas$ psql -d test -U dba -h localhost -f pgbd7.sql
    usuario
---------------
 usuarioprueba
(1 fila)

dayron@kilimanjaro:/pruebas$
```

Anexo 13:

Validación de plug-ins de buenas prácticas para la herramienta informática Audit BD

Proyecto: Laboratorio de Seguridad Informática

Fecha: D 23 M 06 A 2010

En el trabajo de diploma para optar por el título de Ingenieros en Ciencias Informáticas, que tiene como tema: "Desarrollo de plug-ins de soporte para la herramienta auditBD", de los autores Danay Betancourt Quintanal y Dayron Abreus Ruiz, se desarrollan varios plug-ins para la herramienta auditBD, que le permitirán a la misma realizar auditorías a servidores de bases de datos PostgreSQL. Se hace necesario una validación y aceptación de estos plug-ins por parte del proyecto que desarrolla la herramienta informática auditBD, por lo que se le solicita al jefe del proyecto su evaluación y aceptación de los mismos.

Mediante el presente documento expresa su acuerdo con la validación y aceptación de los plug-ins desarrollados en el trabajo, para aplicarlos a la herramienta auditBD.

Observaciones: _____

Rafael Thompson Martínez

J´de Proyecto

Firma J´ Proyecto

Anexo 14: Encuesta aplicada a los expertos para conocer su coeficiente de conocimiento (Kc) y su coeficiente de argumentación (Ka).

Encuesta aplicada a los expertos para conocer su coeficiente de conocimiento (Kc) y su coeficiente de argumentación (Ka).

Usted ha sido seleccionado como posible experto para la validación del trabajo. Para calcular su coeficiente de competencia y certificarlo como experto de la solución, se le solicita responda las siguientes preguntas en la mediad que le sea posible. Muchas Gracias.

1. En una puntuación del 1 al 10, ¿qué conocimientos crees usted que posee sobre el tema? ___

   Nota: Tenga en cuenta que 0 significa no tener ningún conocimiento sobre el tema y 10 significa tener pleno conocimiento sobre el tema.

1. Clasifique su grado de competencia sobre los aspectos o fuentes de argumentación sometidos a consideración.

   Análisis teóricos realizados por usted.____ Alto____ Medio___ Bajo

   Su experiencia obtenida. ____ Alto____ Medio___ Bajo

   Trabajos de autores nacionales.____ Alto____ Medio___ Bajo

   Trabajos de autores extranjeros. ____ Alto____ Medio___ Bajo

   Su propio conocimiento del estado del problema.

   ___ Alto____ Medio___ Bajo

   Su intuición. ___ Alto____ Medio___ Bajo

Anexo 15: Tabla auxiliar para el cálculo del grado de argumentación del experto (Ka).

Fuentes de argumentación	Grado de influencia de cada una de las fuentes en sus criterios.		
	Alto	Medio	Bajo
Análisis teóricos realizados	0.3	0.2	0.1
Su experiencia obtenida	0.5	0.4	0.3
Trabajos de autores	0.05	0.04	0.03
Trabajos de autores	0.05	0.04	0.03
Su propio conocimiento del estado del problema	0.05	0.04	0.03
Su intuición	0.05	0.04	0.03
Total	1	0.76	0.52

Anexo 16: Cuestionario Aplicado a Expertos sobre el trabajo realizado Desarrollada.

Cuestionario Aplicado a Expertos sobre la Investigación Desarrollada.

Usted ha sido seleccionado como experto, basándonos en su aval, desempeño y conocimientos en el campo objeto de estudio. Se le solicita que responda en la medida que le sea posible las interrogantes planteadas a continuación con el objetivo de llevar a feliz término la investigación. Se les agradece de antemano. Muchas Gracias.

Nombre y Apellidos: _____

Centro Laboral: _____

Grado Científico: _____ **Categoría Docente:** _____

Años de experiencia laboral: _____**Asignatura Impartida:** _____

1. ¿Qué beneficios cree que tenga el desarrollo de este trabajo?

2. ¿Qué importancia usted le concede a la seguridad de los datos en las aplicaciones informáticas?

___Muy Alta ____Alta ____Media ___Baja ___Muy Baja

3. ¿Qué importancia usted le concede a las buenas prácticas en la seguridad de los datos?

___Muy Alta ____Alta ____Media ___Baja ___Muy Baja

4. ¿En qué medida cree usted que influya la seguridad de los datos en la calidad del software desarrollado en la UCI?

__100% ___75% ___50% ___25% ___0%

5. Con la propuesta de buenas prácticas, obtenidas como resultado de la investigación ¿En qué medida cree usted que se disminuirá el esfuerzo en el momento de corregir brechas de seguridad en el SGBD PostgreSQL?

__100% ___75% ___50% ___25% ___0%

6. ¿En qué medida considera usted que el listado de buenas prácticas recoge los elementos mínimos necesarios para contribuir a la seguridad de los datos, analizando solo el software de BD y del sistema operativo? Si cree preciso eliminar o poner alguna, menciónela y explíquela brevemente.

__100% ___75% ___50% ___25% ___0%

7. Con la propuesta de los plug-ins desarrollados, como resultado del trabajo ¿En qué medida cree usted que se incrementará la calidad de las auditorías que se le realizan al SGBD PostgreSQL?

___Muy Alta ____Alta ____Media ___Baja ___Muy Baja

8. En la escala del 1 al 5 (1 es bajo, 5 es alto) otorgue una evaluación al resultado del trabajo (Plug-ins), basándose en los siguientes criterios.

- Posibilidad de Aplicación.

- Complemento a la herramienta AuditBD.

___1 ___2 ___3 ___4 ___5

Haga un comentario o aporte sobre la labor desarrollada. (El comentario es libre y debe reflejar algún elemento de interés que aporte elementos para la investigación).

Glosario

Awk: Lenguaje de programación diseñado especialmente para el trabajo con ficheros y textos basado en patrones.

base y global: Son carpetas donde postgres guarda los archivos de bases de datos en el disco duro.

BD: Base(s) de dato(s)

Cluster: Es una colección de BDs administradas por una sola instancia de un servidor de PostgreSQL.

CSV: Comma Separated Values, formato de logs soportado por PostgreSQL.

Directorio: Es, en sistemas GNU\Linux, el equivalente a carpeta en Windows.

Firewall: También conocido como cortafuegos, es un tipo de herramienta informática o lógica que restringe o permite la entrada o salida de información por varios puertos de un equipo.

Gawk y Mawk: Son intérpretes del lenguaje awk.

GSSAPI: En este documento se refiere a uno de los métodos de autenticación que soporta PostgreSQL, basado en el protocolo de autenticación de igual nombre.

LDAP: Un método de autenticación soportado por PostgreSQL, basado en Lightweight Directory Access Protocol, el cual es un protocolo de red para acceder a un directorio de manera ordenada utilizado para gestionar información y recursos acerca de usuarios, así como políticas de seguridad.

Log: Son registros de diferentes acciones definidas previamente que permiten tener un historial de la actividad de aplicaciones.

MD5: Es un algoritmo de cifrado unidireccional que se usa en la comprobación de integridad de los datos o para proteger contraseñas como es el caso en PostgreSQL.

Plug-in: Es una herramienta informática que provee funciones extra a una aplicación núcleo.

postgres: Es el nombre del usuario encargado de correr el servicio de PostgreSQL en el equipo, el nombre del súper usuario por defecto en PostgreSQL, una base de datos que se genera por defecto en el cluster.

Role (rol): Es un grupo o usuario que tiene influencia en todo el cluster de BD.

Samerole: Es un valor que toma el campo "database" en el archivo de acceso pg_hba.conf, que indica una conexión del usuario a una BD cuyo nombre coincide con el del rol a donde pertenece el mismo.

Sameuser: Indica una conexión a una BD con el mismo nombre del usuario que se conecta.

SBD: Seguridad en bases de datos, o seguridad en la base de datos, según el contexto.

Schema: Los schemas son formas de organización que permiten que varios usuarios accedan a una BD sin interferir entre ellos, organizar la BD en grupos lógicos para manejarlos mejor.

SGBD: Sistema gestor de bases de datos.

Shell: Es un intérprete de comandos, provee una interfaz entre el sistema operativo y el usuario.

Sniffing: Tipo de ataque informático donde se captura parte del tráfico de red para su posterior análisis.

SO: Sistema operativo.

SQL: Es un lenguaje de consultas utilizado en bases de datos.

SSL: Secure Shell Layer, tipo de protocolo para comunicaciones seguras en red.

STDERR: Valor por defecto del parámetro log_destination de la configuración de PostgreSQL. Indica una ubicación para los logs.

Súper usuario: Es el mayor privilegio que puede alcanzar un usuario en PostgreSQL; aunque también se usa en otros contextos como Sistemas Operativos GNU\Linux.

SYSLOG: Valor del parámetro log_destination que indica salvar los logs a la ubicación especifica de syslog. Ubicación en el sistema operativo donde se registran logs.

System Catalog: El System Catalog provee información de los metadatos en el cluster de BD.

Tablespace: Es una colección de bases de datos manejada por una instancia del servidor de PostgreSQL.

Autenticación: Acción o función para comprobar la autenticidad de algo, en este documento, se usa para la autenticidad de los usuarios.

Lenguajes procedurales: Lenguaje en el que se implementan funciones definidas por el usuario en PostgreSQL.

HU: Historias de Usuarios.