Universidad de las Ciencias Informáticas

Facultad 3

Propuesta de un procedimiento para la seguridad de las bases de datos durante el diseño e implementación

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO INFORMÁTICO

Autora: María Antonia Lajús Marrero

Tutores: Ing. Rudel Cárdenas Díaz

Ing. Yosvany Márquez Ruiz

Consultante: Dr. Pascual Verdecia Vicet

Asesor: Dr. Gabriel Lajús Barrabeitg

Cuidad de la Habana

Junio del 2008

"La posibilidad de realizar un sueño es lo que hace que la vida sea interesante."

Paulo Coelho

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco 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 16 días del mes de Junio del año 2008.

María Antonia Lajús Marrero	**Yosvany Márquez Ruiz**	**Rudel Cárdenas Díaz**
_____	_____	_____
Firma del Autor	Firma del Tutor	Firma del Tutor

DATOS DE CONTACTO

Tutores

Yosvany Márquez Ruiz (Cuba, 1972).

Graduado de Ingeniería en Informática en el año julio 1995, en la CUJAE. Analista de Sistema “A” en TEICO, Casa del Software Villa Clara desde el año 1996 hasta el 2001. Especialista en Sistemas Informáticos en TEICO, Casa del Software Villa Clara, desde el año 2002 hasta el 2005. Jefe de Módulo Administración Financiera del Proyecto Registro y Notarias en la UCI desde el año 2006.

Rudel Cardenas Diaz (Cuba, 1980).

Graduado de Ingeniería en Informática desde el año 2003, en la CIJAE. Categoría Docente: Instructor. Jefe del Centro de Datos SAREN. Trabaja en el proyecto Registro y Notarias en la UCI desde el año 2005. Actualmente aspira al Grado Científico de Máster.

Consultante

Pascual Verdecia Vicet (Cuba, 1963)

Ingeniero de Minas, Ingeniero Civil (1994), Máster en Voladura con Explosivos, Doctor en Ciencias Técnicas (2002), 21 años profesor de Física, Categoría Asistente.

Asesor

Gabriel Lajús Barrabeitg (Cuba, 1956).

Doctor en Medicina, Especialista en Psiquiatría, Investigador Auxiliar, Máster en Psiquiatría Social, Diplomado en Psicoterapia y Diplomado en Gerencia Organizacional; Profesor de Análisis Transaccional del EastWind Institute de Canadá y de Cuba y, Profesor de la Universidad Médica de Ciudad de la Habana, impartiendo docencia en la Facultad Universitaria de Cuba “Victoria de Girón”, así como en otras instituciones médicas del país y del extranjero, como en la República Bolivariana de Venezuela en estos momentos. Actualmente aspira al Grado Científico de Doctor en Ciencias.

AGRADECIMIENTOS

¿Agradecer? Uff, resulta un poco difícil, porque hay tantas personas especiales que aparecen en nuestras vidas y a quienes no podemos dejar de agradecer en momentos como estos, por existir y estar ahí.

Por eso agradecer:

Mucho, mucho a mis padres que aunque están lejos, son participes de cada paso que doy en mi vida. A ustedes que me dieron la vida, que han sabido guiarme con gran amor y han sabido ser, con defectos y virtudes, los mejores padres del mundo; por estar conmigo en las buenas y malas, por confiar en mi con los ojos cerrados, por creer en mis sueños y apoyarme para lograr alcanzarlos, por su paciencia inagotable con mis malcriadeces, por vivir para mi, por ser un ejemplo a seguir, de sacrificio, entrega y compromiso. Gracias por formar en mí, los valores que me hacen ser hoy la persona que ven. A ti mamita gracias por tu apoyo cuando casi toque fondo, me hiciste ver la luz al final del túnel y recobrar la fe, te amo mucho. A ti papito muchas gracias por cada minuto dedicado a mi tesis, gracias por tus consejos, revisiones, por tu apoyo con la tesis, por tu paciencia, te amo. Sin ustedes nunca lo hubiese logrado. Este resultado es de ustedes.

Pascual, me ayudaste a arrancar. Gracias por guiarme cuando estaba perdida y no sabía por donde empezar con toda esta investigación, por apoyarme siempre incondicionalmente y por ser tan buen profesor y amigo, eres un ejemplo, te quiero BARTOLO.

A mis tutores muchas gracias. Rudel, tu que desde la distancia siempre respondiste cada duda que me surgió, a pesar de todo tu trabajo, gracias por confiar en mi, como lo hiciste siempre. Yosva, jejeje a ti ufff un gracias, especial, por ser mi conciencia optimista, por apoyarme y estar a mi lado, por aguantarme así de obsesiva como me puse con las cosas de la tesis, por confiar en mi, por tener siempre una salida y hacerme ver que las cosas siempre se pueden lograr y al final el resultado puede ser muy bueno. Gracias por ser amigo y hacerme creer en el poder que puedo llegar a tener.

A mis tías, por dedicarme un espacio algunos fines de semana haciéndome compañía y preocupándose por mis cosas.

A unas personas que han entrado en mi vida para convertirse en familia; Idalmis (viejuca), Carlito (*my brother*) y Carlos, muchas gracias por apoyarme, por cuidarme y acogerme en su familia como una más, por preocuparse y ocuparse de mi cada fin. A ti Pedro, no te olvido, has sido un amigo muy especial, al que admiro por los grandes valores que te acompañan, gracias por todo. A la china (mi abuelita de San Agustín) por preocuparte siempre por mi y por tus visitas los domingos. A Migdi, por estar en mí casa apoyándonos a mis padres y a mí.

A mis amigas de los camilitos, el Club de las Monjitas. Es lindo saber que aun, después de 5 años, estamos ahí las unas para las otras.

A dos nuevas amigas, Ainel y Lenia, por hacerme compañía tantas veces, sacarme de la monotonía, preocuparse por mis cosas y hacerme un lugar en sus corazones.

Gracias Suce, gracias Yami, por estar ahí, por darme espacio en sus vidas, por estar a mi lado es los momentos más complejos de mi vida, por compartir mis alegrías, mis tristezas, mis dolores, mis caprichos, mis sueños, mis pesadeces, por ser las mejores amigas siempre que las he necesitado y por estar ahí aun cuando no las he tenido en cuenta. Palabras que expresen todo los que quisiera decirles, se que existen pero ahora no las encuentro, solo decirle que las quiero mucho a las dos con todo mi corazón y que más que amigas son mis hermanas, y con sus defectos y virtudes, no quiero dos amigas mejores.

Gracias a Lidy por darme ánimos y fuerza para salir adelante, por alimentar mi vida con esperanza y llenarme de fe. Eres una magnifica amiga y una persona increíble, doy gracias a dios por haberte conocido. Te quiero amiga. Gracias Yuyi, por estar ahí siempre, por quererme tanto, eres un ser increíble, que te mereces muchas cosas buenas y las tendrás amiga, te quiero. Yamile a ti no pudo dejar de mencionarte pues a pesar de todo, te quiero y siempre tendrás un lugar en mi corazón, gracias por escucharme y darme aliento.

Gracias Mily y Aray por preocuparse siempre por mí, por dar el paso al frente siempre que las he necesitado y por quererme así como las quiero yo.

Gracias Rey, por cada tiempo que me has dedicado y por ser un amigo tan magnifico y oportuno, nunca te olvidare. Gracias July, por preocuparte por mi, darme siempre el cariño de amigo oportuno, por alarme a todas las reuniones y actividades de las FAR, serás muy buen oficial. Gracias Io, por esas conversaciones tan profundas y por todo el cariño que siempre me profesas, eres muy buen amigo. Yandry muchas gracias por creer en mi, me diste las palabras precisas antes de la pre defensa, nunca lo olvidare, me dio las fuerzas que necesitaba para lograrlo.

A mis nuevos amigos del proyecto: Ale, gracias por apoyarme siempre y liberarme de trabajo en el proyecto, para que pudiera adelantar en mi tesis, eres una persona muy noble, gracias por tu amistad. Yunier, gracias por decirme linda cada mañana, eso me da mucha fuerza cada día. Maykel es bueno siempre ver esa mirada tímida, gracias por dejarme entrar.

Gracias a Fidel, por siempre mi comandante, por crear una sociedad como la nuestra, llena de oportunidades para ser y hacer algo bueno de nuestras vidas.

Leo y todavía me faltan muchas personas que no quisiera dejar de mencionar, pero solo hay una hoja de agradecimiento para mí en la tesis y ya me tomé dos. Por eso, que nadie se sienta mal si su nombre no está, porque yo les agradezco a todas y todos los que forman parte de mi vida y que de alguna manera han contribuido para que hoy haya logrado este título. Gracias a todos los que me tienen presente, a los que están y veo a diario, y a los que aún lejos me tienen en sus corazones.

Muchas gracias.

DEDICATORIA

Porque amarlos es mi pasión.

Salir adelante, mi deber.

Enorgullecerlos, mi misión.

Ser feliz, es hacerlos felices.

Porque mi vida son ustedes, mitad y mitad, dedico este trabajo a ti mamita mía y a ti papito mío. Orgullos de mi vida, mis dos amores por siempre.

Mary

RESUMEN

El presente documento, refleja detalladamente cada paso de la investigación realizada, para la creación de un procedimiento que garantice, en cierto grado, la seguridad de una base de datos. En él figuran, los aspectos fundamentales que se deben probar en cuanto a seguridad de bases de datos: la confidencialidad, la integridad y la disponibilidad.

Constituye un aporte sugestivo, ya que reúne gran número de información relacionada con la seguridad de las bases de datos, como la integridad, los privilegios de acceso, la autenticación, la asignación de roles, la trazabilidad, la salvaguarda y recuperación de la información; haciendo alusión a las políticas de seguridad y recuperación, a tener en cuenta, en caso de producirse desastres que atentan contra la integridad y preservación de la información, que se maneja en cada instante de tiempo. Es una propuesta que recoge, los aspectos fundamentales, donde la mayoría de los diseñadores y desarrolladores se bases de datos, comenten errores. Todos estos elementos se encuentran detallados, con gran precisión y claridad, en el presente material.

El mismo fue revisado por varios especialistas, quienes ofrecieron sus criterios sobre el procedimiento planteado como solución, de manera que cada uno de ellos, avaló la propuesta. De esta forma, se brinda una solución de alto valor científico-técnico que puede ser puesta en práctica, con pocos esfuerzos, en cualquier proyecto productivo.

ÍNDICE

INTRODUCCIÓN

A lo largo de los años la información ha sido un elemento muy valorado. Sin embargo, la socialización de la misma ha alcanzado su mayor auge en las últimas décadas, con el desarrollo de las tecnologías de redes, la aparición de novedosos servicios, el almacenamiento de información, el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) y el aumento de las necesidades de intercambio y comunicación. Todo esto ha favorecido que la información se haya convertido en factor determinante para el desarrollo de cualquier sociedad.

Es real que en todo el mundo se generan a diario millones de bits de información, que son consumidos en la misma proporción. Esto hace necesario que se establezca una organización y control sobre los mimos, ya sea para poder ofrecerlos o administrarlos. De esta manera aparecen los sistemas de información.

Los sistemas de información, son un conjunto de elementos que interactúan entre si, con el fin de apoyar determinadas actividades y son manejados a nivel de organizaciones, entidades y empresas. Realizan cinco operaciones básicas, sobre la información: la entrada, el almacenamiento, el procesamiento, la seguridad y su presentación; sin embargo el soporte lógico que los hace operables son las bases de datos, que aseguran la consistencia de los datos que manejan, así como el acceso a los mismos.

La colección de archivos relacionados, que almacenan una representación abstracta de un problema del mundo real y, los datos de la información acerca del problema en cuestión, constituyen las bases de datos; las que representan el soporte fundamental de empresas e instituciones en sus cotidianos procesos de gestión; ya que manejan, no solo, gran cantidad de información, sino a su vez, diversa y valiosa, la que es necesario organizar y dejar disponible para el momento de la toma de decisiones y/o la prestación de servicios.

En el entorno de las bases de datos, se trata de garantizar que los datos almacenados sean veraces y no estén corruptos, que sólo acceda a cada dato aquél que le corresponde acceder, y que la información esté disponible siempre que se requiera su uso.

La gestión de la seguridad de los sistemas, se convierte en un problema de primer orden cuando los datos críticos, se exponen a las posibles vulnerabilidades tecnológicas. La situación ideal, es aquella

en la que se logra gestionar un nivel adecuado de los tres aspectos fundamentales de la seguridad de la información: la integridad, la confidencialidad y la disponibilidad.

Es por eso, que al hablar de seguridad, debemos centrarnos en la información misma aunque a menudo se hable de seguridad informática, de seguridad de los Sistemas de Información (SI) o de seguridad de las Tecnologías de la Información (TI). Particularmente los diseñadores de bases de datos toman acciones al crearlas, tomando en cuenta el volumen de las transacciones y las restricciones que tiene que especificar en el acceso a los datos.

En la República Bolivariana de Venezuela, las decisiones de política económica adoptadas por el Estado están basadas en los datos disponibles acerca de las actividades realizadas por los distintos agentes que intervienen en la economía, es decir, las familias, las administraciones públicas y las empresas. En este país el proceso de informatización es profundo, sobre todo en una sociedad donde se suceden vertiginosos cambios, con el objetivo de lograr una mayor eficiencia en la administración de los recursos con que cuenta el Estado.

En Venezuela, el Ministerio del Poder Popular para la Relaciones Interiores y Justicia, se encuentra bastante automatizado pero solo a nivel de organizacional, porque ninguno de sus entes adscriptos, como por ejemplo Servicios Autónomos de Registro y Notarias (SAREN), tiene un sistema automatizado integral a nivel de toda la organización; es por tal razón, que se autorizó la digitalización y almacenamiento documental de la información contenida en las oficinas del país, profundizando en un conjunto de acciones encaminadas a garantizar un nivel permisible de la seguridad jurídica de los servicios registrales y notariales de la nación.

El proyecto Registro y Notarías (RN), está orientado hacia la solución de los problemas que afectan a los registros públicos y mercantiles, por lo que se procesará información relacionada con los inmuebles y compañías. Para obtener un sistema que de respuesta a las necesidades descritas anteriormente; se fragmentó en varios módulos; modelados de manera tal de que cada uno diera respuesta a una situación especifica del negocio y, posteriormente, integrar cada uno a un sistema global. Los módulos planteados son los siguientes:

- Módulo de Público.
- Módulo de Mercantil.
- Módulo de Servicio Autónomo.
- Módulo de Administración Financiera.

Por tanto en la concepción de este sistema, cada módulo de los anteriormente mencionados necesita de la modelación del negocio a nivel de Objetos Relacional. La base de datos es vital en el correcto funcionamiento de cada módulo, debiéndose prestar mucha atención a la seguridad, lo cual implica fiabilidad, disponibilidad y consistencia de la información.

El último de los módulos antes mencionados, subsistema del sistema global, es el de Administración Financiera. Este sistema está encaminado a garantizar un conjunto de acciones financieras integradas para todos los órganos que conforman el gobierno actual, contemplando, de esta forma, los siguientes componentes:

- Módulo de Presupuesto.
- Módulo de Tesorería.
- Módulo de Fondos Rotatorios.
- Módulo de Contabilidad.
- Sistema de Gestión Financiera de Recursos Humanos.
- Módulo de Gestión de Bienes.
- Módulo de Compras y Servicios.
- Módulo de Recaudación

Todos estos procesos administrativos que se ejecutan en los organismos públicos, afectan a uno o a varios de los sistemas nombrados, simultáneamente. Por lo que el nivel de complejidad de las bases de datos se hace mayor y, aún más, la necesidad de mantener la información segura y confiable, teniendo en cuenta el nivel de intercambio de información y la integración entre las diferentes áreas claves de la organización desde el punto de vista registral y financiero; lo que reclama la necesidad de aplicar determinados mecanismos de seguridad en el diseño de de las bases de datos, que garanticen, plenamente, el propósito antes mencionado.

Esto no quiere decir que la base de datos existente hasta el momento, no sea segura, ni que dejen de aplicarse mecanismos de seguridad, sino que ésta se aplica en dependencia de la experiencia de los diseñadores de bases de datos y de ciertas normas, que se aplican a cada módulo en particular a partir del diseñador; todo lo cual afecta el desempeño de los diseñadores, pues deben esforzarse más para lograr garantizar la seguridad e integridad de los datos que se almacenan, cuando se habla de una base de datos financiera. Simultáneamente, surge la idea de realizar un procedimiento que garantice la seguridad dentro de un tipo de base de datos, con las características ya descritas,

sirviendo así como estándar para el trabajo de los diseñadores del grupo de desarrollo a partir del estudio de las diferentes tendencias que existen en el mundo, para aplicar seguridad a una base de datos y, de las técnicas más utilizadas en tal sentido. Los aspectos que se expusieron constituyen *la problemática de este trabajo.*

Problema

¿Cómo realizar el diseño de la base de datos de un sistema, que garantice en cierto grado la integridad y seguridad de la información, aspectos que se ven afectados durante la etapa de diseño e implementación?

Objeto de Estudio

Seguridad de las bases de datos.

Campo de Acción

Procedimientos que tributen a la seguridad en las bases de datos.

Objetivos Generales

Elaborar un procedimiento que contribuya al diseño e implementación de una base de datos, que cumpla con las regulaciones vigentes en la República Bolivariana de Venezuela, y garantice la seguridad, el almacenamiento, el procesamiento y la consulta segura de la información generada en los sectores públicos.

Hipótesis

Si se elabora un procedimiento capaz de garantizar en cierto grado la seguridad de la información en el diseño e implementación de las bases de datos, entonces, quedan resueltos, los problemas de disponibilidad, fiabilidad y seguridad de los datos almacenados en el sistema de Administración Financiera del Proyecto Registro y Notarias.

Tareas científicas a desarrollar

- Estudio del estado del arte de la temática.
- Análisis de los modelos de diseño de bases de datos.
- Realizar un estudio de las herramientas utilizadas para el diseño de bases de batos relacionales.

- Realizar un estudio de los Sistemas Gestores de Bases de Datos (SGBD).
- Elaborar la propuesta de procedimiento.
- Realizar un estudio de los estándares internacionales que existen para evaluar la seguridad de los sistemas de información.
- Evaluar el procedimiento propuesto mediante el criterio de varios especialistas en el tema.
- Realizar un análisis y discusión de los resultados obtenidos a partir de la evaluación realizada por los diferentes especialistas.

Posibles resultados

Un procedimiento que garantice en cierto grado la seguridad de las bases de datos, la persistencia y la consulta de la información acumulada durante los procesos realizados en los Registros Públicos de Venezuela, así como la disponibilidad y fiabilidad de los mismos.

Métodos de trabajo científico

Método teórico

- **Histórico:** Porque es el que me permite estudiar todo el proceso y obtener un conocimiento histórico de su desarrollo y comportamiento.
- **Método Analítico-Sintético:** Porque se hace necesario la investigación y estudio de documentos, para extraer los elementos necesarios que se relacionan con la gestión de la información en los Registros y Notarias de la República Bolivariana de Venezuela.

Métodos lógicos

- **Hipotético-Deductivo:** Porque es el que permite inferir conclusiones a partir del conocimiento precedente (Hipótesis) y, además permite comprobarla.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En el desarrollo del capítulo se expondrán diferentes conceptos que serán empleados a lo largo de la investigación, para lograr un entendimiento profundo del tema. Se realiza un recuento histórico del tema de las bases de datos, partiendo de su surgimiento hasta su aplicabilidad en la actualidad. Se reseñan diferentes metodologías que son empleadas para el diseño e implementación de bases de datos y, se introdujo además, el desarrollo de aplicaciones con bases de datos, revelando la necesidad de integrar los aspectos relacionados con la seguridad en los sistemas de datos. Además se hace un breve estudio, de las principales herramientas de diseño y administración de las bases de datos, haciendo énfasis sobre la suite de productos de Oracle, particularmente en su versión Oracle Database 10g. Finalmente se mencionan los aspectos más importantes de la seguridad de los sistemas de información, aludiendo criterios que se tienen en cuenta internacionalmente cuando se van a evaluar.

1.1 Definición de base de datos

El uso de los discos fue un adelanto muy efectivo, ya que por medio de este soporte se podía consultar la información directamente, ayudando a ahorrar tiempo. No era necesario saber exactamente donde estaban los datos en los discos, ya que en milisegundos era recuperable la información. Estos fueron los que dieron paso a las bases de datos, pues los programadores con su habilidad de manipulación de estructuras, junto a las ventajas de los discos, hicieron posible guardar estructuras de datos como listas y árboles.

Se dice que las bases de datos, tienen sus orígenes en la década de los años sesenta, más específicamente en el año 1963, durante un simposio celebrado en California EUA. Las bases de datos se convierten en una herramienta de vital importancia para los progresos en la adquisición y almacenamiento de información en diferentes ambientes como gobiernos, industrias y aplicaciones científicas, arrojando como resultado el surgimiento de enormes bases de datos, cuyo tamaño se incrementa rápidamente, tanto en número de registros como en la dimensionalidad de los mismos. A partir de aquí surge en la actualidad, una necesidad creciente de explorarlas y extraer información y conocimiento que sea de interés para los propietarios de las mismas.

Existen diferentes definiciones sobre el término bases de datos. Gran número de expertos han elaborado axiomas referentes a este tema, resumiendo sus conocimientos sobre el mismo, en

conceptos precisos; además de que aparecen bien especificados en la literatura que aborda tal temática. A continuación se citarán solo algunas, que son las que resumen de forma menos restringida este concepto:

- Una base de datos o banco de datos se puede definir como un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. (1)
- Una base de datos, es un conjunto de información estructurada en registros y almacenada en un soporte electrónico legible desde un ordenador. Cada registro constituye, una unidad autónoma de información que puede estar a su vez estructurada en diferentes campos o tipos de datos, que se recogen en dicha base de datos. (2)
- Una base de datos es un conjunto de datos relacionados entre sí. Por datos entendemos hechos conocidos que pueden registrarse y que tienen un significado implícito. (3)
- Base de datos es un conjunto exhaustivo no redundante de datos estructurados y organizados independientemente de su utilización y su implementación, en máquina accesible en tiempo real y compatible con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo. (4)

De tal forma se acota, que las bases de datos constituyen un conjunto de información que está almacenada en forma sistemática, de manera tal que los datos que la conforman puedan ser utilizados en forma fragmentada cuando sea necesario; forman un conjunto de registros interrelacionados que se almacenan con objeto de satisfacer las necesidades del proceso de información en una organización, y la base de datos, permite incluir información nueva o modificar la existente, eliminando toda posibilidad de redundancia e inconsistencia, además de que facilita compartir la información y mejorar los controles sobre la misma.

En la actualidad, y gracias al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, lo que ofrece un amplio rango de soluciones al problema de almacenar datos.

1.2 Modelos de datos

Algunos autores que han investigado sobre este tema y realizado estudios profundos han aportado definiciones precisas de los modelos de datos; se reflejan a continuación varias de ellas:

- Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información. (1)
- Con respecto al diseño de bases de datos, el modelado de datos puede ser descrito así: "dados los requerimientos de información y proceso de una aplicación de uso intensivo de datos (por ejemplo, un sistema de información), construir una representación de la aplicación que capture las propiedades estáticas y dinámicas requeridas para dar soporte a los procesos deseados (por ejemplo, transacciones y consultas). Además de capturar las necesidades dadas en el momento de la etapa de diseño, la representación debe ser capaz de dar cabida a eventuales futuros requerimientos" (Brodie 1984:20). (5)

De lo anterior se puede concluir que los modelos de datos son un conjunto de conceptos que sirven para describir la estructura de una base de datos; los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos. (6) No son elementos físicos: son abstracciones que permiten la implementación de un sistema eficiente de bases de datos; por lo general se refieren a algoritmos, y conceptos matemáticos. Se deduce entonces que son el punto clave en la construcción de la base de datos.

Los modelos de datos se pueden clasificar dependiendo de los tipos de conceptos que ofrecen para describir la estructura de la base de datos. Los modelos de datos de alto nivel, o modelos conceptuales, disponen de conceptos muy cercanos al modo en que la mayoría de los usuarios percibe los datos, mientras que los modelos de datos de bajo nivel, o modelos físicos, proporcionan conceptos que describen los detalles de cómo se almacenan los datos en el ordenador; los conceptos de los modelos físicos están dirigidos al personal informático, no a los usuarios finales. Entre estos dos extremos se encuentran los modelos lógicos, cuyos conceptos pueden ser entendidos por los usuarios finales, aunque no están demasiado alejados de la forma en que los datos se organizan físicamente. (6)

Los modelos de datos se dividen en tres grupos:

1. Modelos lógicos basados en objetos.
2. Modelos lógicos basados en registro.
3. Modelos físicos de datos.

1.2.1 Modelos lógicos basados en objetos

Son aquellos que, se usan para describir datos en el nivel conceptual, es decir, con este modelo se representan los datos de tal forma como son captados en el mundo real. Tienen una capacidad de estructuración bastante flexible, y permiten especificar restricciones de datos explícitamente. Existen diferentes modelos de este tipo, pero el más utilizado por su sencillez y eficiencia es el modelo Entidad-Relación.

Modelo Entidad-Relación

Denominado por sus siglas como: E-R; Este modelo representa a la realidad a través de entidades, que son objetos que existen y que se distinguen de otros por sus características, por ejemplo: un alumno se distingue de otro por sus características particulares como lo es el nombre, o el número de control asignado al entrar a una institución educativa; así mismo, un empleado, una materia, etc. Las entidades pueden ser de dos tipos:

Tangibles: Son todos aquellos objetos físicos que podemos ver, tocar o sentir.

Intangibles: Todos aquellos eventos u objetos conceptuales, que no podemos ver, aún sabiendo que existen, como por ejemplo la entidad materia, sabemos que existe, sin embargo, no la podemos visualizar o tocar.

Las características de las entidades en base de datos se llaman atributos, por ejemplo: el nombre, dirección, teléfono, grado, grupo, etc., son atributos de la entidad alumno y clave, número de seguro social, departamento, etc., son atributos de la entidad empleado. A su vez una entidad se puede asociar o relacionar con más entidades a través de relaciones.

1.2.2 Modelos lógicos basados en registro

Son los que se utilizan para describir datos en los niveles conceptual y físico. Estos modelos utilizan registros e instancias para representar la realidad, así como las relaciones que existen entre estos

registros (ligas) o apuntadores. A diferencia de los modelos de datos basados en objetos, se usan para especificar la estructura lógica global de la base de datos y para proporcionar una descripción a nivel más alto de implementación.

Los tres modelos de datos más ampliamente aceptados son:

- Modelo de datos jerárquico.
- Modelo de datos de red.
- Modelo de datos relacional.

1.2.2.1 Modelo de datos jerárquico

El modelo de datos jerárquico, como su nombre lo indica, almacena su información en una estructura jerárquica y surge en la década de 1960. Se organiza en una forma similar a un árbol, donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y los nodos que no tienen hijos se les conocen como hojas.

Es especialmente útil en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos, permitiendo crear estructuras estables y de gran rendimiento. Una de las principales limitaciones de este modelo, es su incapacidad de representar eficientemente la redundancia de datos. A continuación se mencionan los problemas típicos del modelo de bases de datos jerárquico, problemas que se derivan del hecho, de que el sistema gestor de bases de datos, no implementa ningún control sobre los propios datos, sino que queda en manos de las aplicaciones garantizar que se cumplan las condiciones invariantes que se requieran; estos son los siguientes:

a. **Duplicidad de registros**

   No se garantiza la inexistencia de registros duplicados, lo que también es cierto para los campos clave; es decir, no se garantiza que dos registros cualesquiera, tengan diferentes valores en un subconjunto concreto de campos. (1)

b. **Integridad referencial**

   No existe garantía de que un registro hijo esté relacionado con un registro padre válido. Por ejemplo, es posible borrar un nodo padre sin eliminar antes los nodos hijos, de manera que éstos últimos están relacionados con un registro inválido o inexistente. (1)

c. Desnormalización

Este no es tanto un problema del modelo jerárquico como del uso que se hace de él, sin embargo, a diferencia del modelo relacional, las bases de datos jerárquicas no tienen controles que impidan la desnormalización de una base de datos, por ejemplo, no existe el concepto de campos clave. (1)

1.2.2.2 Modelo de datos de red

El modelo de datos red surge en la década de 1960. Es ligeramente distinto pues en este modelo, un hijo puede tener varios padres; las entidades se representan como nodos y sus relaciones son las líneas que los unen y, en esta estructura cualquier componente puede relacionarse con cualquier otro.

Fue una gran mejora con respecto al modelo anterior, pues ofrece una solución eficiente al problema de redundancia de datos; pero, aún así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales. (1)

1.2.2.3 Modelo de datos relacional

El modelo de datos relacional es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. El éxito de este modelo se basa fundamentalmente, en que ofrece un sistema simple y eficaz para representar y manipular los datos, además de constituir un modelo relacional con sólidas bases teóricas. Tras ser postulados sus fundamentos en el año 1970 por Edgar Frank Codd, de los laboratorios International Business Machines Corporation (IBM) en San José, California, no tardó en consolidarse como un nuevo paradigma en los modelos de bases de datos. Su idea fundamental es el uso de relaciones, que pudiesen considerarse en forma lógica como conjuntos de datos llamados tuplas. La teoría de las bases de datos relacionales, la mayoría de las veces, se conceptualiza de una manera más fácil de imaginar, pues se piensa en cada relación como si fuese una tabla que está compuesta por registros, que representarían las tuplas y campos.

En este modelo, el lugar y la forma en que se almacenan los datos no tienen relevancia, a diferencia de otros modelos como el jerárquico y el de red; por lo que tiene las considerables ventajas de que resulta más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante consultas, que ofrecen una amplia

flexibilidad y la posibilidad de poder administrar la información. Algunas de las grandes ventajas del modelo relacional son las siguientes:

- Define un álgebra llamada, álgebra relacional a partir de la cual se realizan todas las manipulaciones posibles sobre las relaciones que se obtienen mediante el uso de operadores.
- Garantía de independencia de los datos.
- Conectividad garantizada con los lenguajes de programación estándar.
- Compatibilidad y estandarización.
- Favorece la normalización por ser más comprensible y aplicable.
- Garantiza la integridad referencial, así al eliminar un registro elimina todos los registros relacionados dependientes.
- Garantiza herramientas para evitar la duplicidad de registros, a través de campos claves o llaves. (1)

No obstante, se deben tener presentes, los aspectos negativos o más bien las limitaciones, que trae aparejada la adopción del modelo de datos relacional, donde las más conocidas son las siguientes:

- Imposibilidad de representar conocimiento en forma de reglas.
- Inexistencia de mecanismos de herencia de propiedades.
- Dificultad para gestionar datos no atómicos.

Las desventajas del modelo relacional, en términos de su poder expresivo semántico, hicieron surgir el interés por los modelos semánticos.

1.2.3 Modelo de datos orientado a objetos

A los modelos que describen el estado y comportamiento de los objetos se les denomina modelos de datos orientados a objetos.

Estos modelos de datos, poseen un poder de expresión similar al de los modelos semánticos, pero los rebasan, en el sentido de que modelan explícitamente el comportamiento. Se asegura que son más

fáciles de implementar y usar cuando se trata de crear aplicaciones, debido a la modularidad integrada, el encapsulamiento y la reutilización de códigos.

Este modelo es bastante reciente y propio de los modelos informáticos orientados a objetos. Una base de datos orientada a objetos es aquella que incorpora todos los conceptos importantes del paradigma orientado a objetos, que son:

a. **Encapsulación**: Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
b. **Herencia**: Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
c. **Polimorfismo**: Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Las bases de datos orientadas a objetos proporcionan ventajas tales como:

- La cantidad de información que puede modelarse en una base de datos orientada a objetos se incrementa y es más fácil modelar esta información.
- Los sistemas de bases de datos orientados a objetos son capaces de tener mayores capacidades de modelado por medio de la extensibilidad.
- En una base de datos orientada a objetos, el manejo de versiones está disponible para ayudar a modelar cambios diversos a los sistemas.
- La reutilización de clases juega un rol vital en el desarrollo y mantenimiento más rápido de aplicaciones. Las clases genéricas son potentes, pero más importante aún es, que ellas pueden ser usadas nuevamente.

Las bases de datos orientadas a objetos, brindan beneficios más allá de los modelos tradicionales de bases de datos, pero poseen desventajas que no debieran ser pasadas por alto, como son las siguientes:

- Los sistemas entidad relación tradicionales han estado en el mercado por un largo tiempo y un cambio se apartaría de las ideas establecidas, requeriría una filosofía diferente de pensar, y en algunos casos los usuarios no tendrían los conocimientos orientados a objetos necesarios para

trabajar con sistemas de bases de datos orientadas a objetos. La educación de las personas en base a la filosofía orientada a objetos es un proceso muy minucioso, requeriría gran cantidad de tiempo, dinero y otros recursos. También y con motivo del cambio, se requeriría más tiempo para mover los datos a los nuevos sistemas manejadores de bases de datos orientadas a objetos.

- La necesidad de que los sistemas tradicionales y los sistemas manejadores de bases de datos orientados a objetos se comuniquen y trabajen juntos, debido a que las bases de datos orientadas a objetos frecuentemente son incompatibles entre ellas. Esto hace imposible migrar una aplicación desde una base de datos orientada a objetos a otra. (1)

1.2.4 Modelo de bases de datos distribuidas

En un modelo de bases de datos distribuidas, los datos se almacenan en varios computadores. Los computadores de un sistema distribuido se comunican entre sí, a través de diversos medios de comunicación, tales como cables de alta velocidad o líneas telefónicas. No comparten la memoria principal ni el reloj. (1)

Un sistema distribuido de bases de datos consiste en un conjunto de localidades, cada uno de las cuales puede participar en la ejecución de transacciones que accedan a datos de una o varias localidades. La diferencia principal entre los sistemas de bases de datos centralizados y distribuidos es que, en los primeros, los datos residen en una sola localidad, mientras que, en los últimos, se encuentran en varias localidades. (1)

1.3 Metodologías de diseño de bases de datos

El diseño de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de éstos independientemente, utilizando técnicas específicas. Así, el diseño de una base de datos se descompone en diseño conceptual, diseño lógico y diseño físico. (6)

1.3.1 Diseño conceptual

El diseño conceptual parte de las especificaciones de requisitos de usuario y su resultado es el esquema conceptual de la base de datos. Un esquema conceptual es una descripción de alto nivel de la estructura de la base de datos, independientemente del Sistema Gestor de Base de Datos (SGBD)

que se vaya a utilizar para manipularla. Un modelo conceptual es un lenguaje que se utiliza para describir esquemas conceptuales. El objetivo del diseño conceptual es, describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar esta información. (6)

En esta etapa se debe construir un esquema a partir de la información que se ha logrado capturar del entorno, independientemente de cualquier consideración física. A este esquema se le denomina esquema conceptual. Al construir el esquema, los diseñadores logran descubrir el significado de los datos, encontrando entidades, atributos y relaciones siendo su objetivo comprender:

- La perspectiva que cada usuario tiene de los datos.
- La naturaleza de los datos, independientemente de su representación física.
- El uso de los datos a través de las áreas de aplicación.

El esquema conceptual se puede utilizar para que el diseñador transmita lo que ha entendido sobre la información que se maneja. Para ello, ambas partes deben estar familiarizadas con la notación utilizada en el esquema. (1)

1.3.2 Diseño lógico

El diseño lógico parte del esquema conceptual y da como resultado un esquema lógico. Un esquema lógico es una descripción de la estructura de la base de datos, en términos de las estructuras de datos que puede procesar un tipo de SGBD. Un modelo lógico es un lenguaje usado para especificar esquemas lógicos (por ejemplo: modelo relacional, modelo de red, etc.). (6)

Es el proceso de construir un esquema de la información que utiliza la empresa, basándose en un modelo de bases de datos específico, independiente del SGBD concreto que se vaya a utilizar y de cualquier otra consideración física. En esta etapa, se transforma el esquema conceptual en un esquema lógico que utilizará las estructuras de datos del modelo de base de datos en el que se basa el SGBD que se vaya a utilizar, como pueden ser el modelo relacional, el modelo de red, el modelo jerárquico o el modelo orientado a objetos. Conforme se va desarrollando el esquema lógico, éste se va probando y validando con los requisitos de usuario. (1)

La normalización, es una técnica que se utiliza para comprobar la validez de los esquemas lógicos basados en el modelo relacional, ya que asegura que las relaciones (tablas) obtenidas no tengan datos

redundantes. El esquema lógico es una fuente de información para el diseño físico, que a su vez juega un papel importante durante la etapa de mantenimiento del sistema, ya que permite que los futuros cambios que se realicen sobre los programas de aplicación o sobre los datos, se representen correctamente en la base de datos. Tanto el diseño conceptual, como el diseño lógico, son procesos iterativos, tienen un punto de inicio y se van refinando continuamente. Ambos se deben ver como un proceso de aprendizaje en el que el diseñador va comprendiendo el funcionamiento de la empresa, y el significado de los datos que maneja. El diseño conceptual y el diseño lógico, son etapas claves para conseguir un sistema que funcione correctamente. (6)

1.3.3 Diseño físico

El diseño físico parte del esquema lógico y da como resultado un esquema físico. Un esquema físico, es una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos. Por ello, el diseño físico depende del SGBD concreto y el esquema físico se expresa mediante su lenguaje de definición de datos. (6)

Para llevar a cabo esta etapa, se debe haber decidido cuál es el SGBD que se va a utilizar, ya que el esquema físico se adapta a él, como se mencionó con anterioridad. Entre el diseño físico y el diseño lógico hay una realimentación, ya que algunas de las decisiones que se tomen durante el diseño físico para mejorar las prestaciones, pueden afectar a la estructura del esquema lógico.

En general, el propósito del diseño físico, es describir cómo se va a implementar físicamente, el esquema lógico obtenido en la fase anterior. Concretamente, en el modelo relacional, esto consiste en:

- Obtener un conjunto de relaciones (tablas) y las restricciones que se deben cumplir sobre ellas.
- Determinar las estructuras de almacenamiento y los métodos de acceso que se van a utilizar para conseguir prestaciones óptimas.
- Diseñar el modelo de seguridad del sistema.

1.4 Herramientas CASE para el diseño de bases de datos

Se pueden definir las herramientas *Computer-Aided Software Engineering* (CASE) como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores,

durante todos los pasos del ciclo de vida de desarrollo de un software. Como es sabido, los estados en el ciclo de vida de desarrollo de un software son: la investigación preliminar, el análisis, el diseño, la implementación y la instalación. (7)

Un CASE es una herramienta que ayuda al ingeniero de software a desarrollar y mantener el software. A continuación se presentan algunas definiciones dadas para el término CASE.

- Herramientas individuales para ayudar al desarrollador de software o administrador de proyecto durante una o más fases del desarrollo de software (o mantenimiento). (8)
- Una combinación de herramientas de software y metodologías de desarrollo. (9)
- Una innovación en la organización, un concepto avanzado en la evolución de tecnología con un efecto potencialmente profundo en la organización. Se puede ver al CASE como la unión de las herramientas automáticas de software y las metodologías de desarrollo de software formales. (10)

La realización de un nuevo software requiere que, las tareas sean organizadas y completadas en forma correcta y eficiente. Las herramientas CASE, fueron desarrolladas, para automatizar esos procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software.

Cuando se realiza la planificación de la base de datos, se puede en la primera etapa del ciclo de vida de las aplicaciones, escoger una herramienta CASE que permita llevar a cabo el resto de las tareas del modo más eficiente y efectivo posible. El uso de las herramientas CASE puede mejorar la productividad en el desarrollo de una aplicación de bases de datos.

1.4.1 Evolución histórica de las herramientas CASE

Las herramientas CASE tienen su inicio con el simple procesador de palabras que fue usado para crear y manipular documentación. La década de los setenta vio la introducción de técnicas gráficas y diagramas de flujo de estructuras de datos. Sobre este punto, el diseño y especificaciones en forma pictórica han sido extremadamente complejos y consumían mucho tiempo para realizar cambios. (10)

La introducción de las herramientas CASE para ayudar en este proceso ha permitido que los diagramas puedan ser fácilmente creados y modificados, mejorando la calidad de los diseños de software. Pronto se reemplazaron los paquetes gráficos por paquetes especializados que habilitan la edición, actualización e impresión en múltiples versiones de diseño. (10)

La primera herramienta comercial se remonta a 1982, aunque algunos especialistas indican que ciertos ejemplos de herramientas para diagramación ya existían. No fue sino hasta 1985 en que las herramientas CASE se volvieron realmente importantes en el proceso de desarrollo de software. El objetivo en 1985 para muchos vendedores fue producir software más rápidamente. (10)

Las herramientas del CASE serían una familia de métodos favorablemente estructurados para planeamiento, análisis y diseño. (10)

En las últimas décadas se ha estado trabajando en esta área; desarrollando sistemas para encontrar técnicas que permitan incrementar la productividad y el control de la calidad en cualquier proceso de elaboración de software y hoy en día, la tecnología CASE, reemplaza al papel y al lápiz por el ordenador, logrando transformar de esta forma, la actividad de desarrollar software, en un proceso automatizado.

1.4.2 Clasificación de las herramientas CASE

No existe una única clasificación de herramientas CASE y, en ocasiones, es difícil incluirlas en una clase determinada. Pudiera clasificarse atendiendo a:

- Las plataformas que soportan.
- Las fases del ciclo de vida del desarrollo de sistemas que cubren.
- La arquitectura de las aplicaciones que producen.
- Su funcionalidad.

Las herramientas CASE, en función de las fases del ciclo de vida que abarcan, se pueden agrupar de la siguiente forma: (10)

a. **Herramientas integradas**, **I-CASE (Integrated CASE, CASE integrado)** abarcan todas las fases del ciclo de vida del desarrollo de sistemas. Son llamadas también CASE workbench.

b. **Herramientas de alto nivel, U-CASE (Upper CASE-CASE superior)** o front-end, orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño.

c. **Herramientas de bajo nivel, L-CASE (Lower CASE-CASE inferior)** o back-end, dirigidas a las últimas fases del desarrollo: construcción e implantación.

d. **Juegos de herramientas o Tools-Case**, son el tipo más simple de herramientas CASE; automatizan una fase dentro del ciclo de vida. Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.

CASE es una combinación de herramientas software (aplicaciones) y de metodologías de desarrollo, donde las metodologías definen los procesos y las herramientas en el proceso de desarrollo del software

1.4.3 Herramientas CASE

Analizando la literatura revisada en Internet y, la aportada por estudiosos del tema, se pudo hacer un compendio de las diferentes herramientas CASE que se utilizan a nivel mundial, en diferentes empresas, compañías y procesos de desarrollo de software en general, fundamentalmente en las que se utilizan en la Ingeniería de bases de datos; con lo que se conformó la lista siguiente:

- **ASADAL**, herramienta CASE especializada en sistemas de tiempo real.
- **CASE GENEXUS Tool.**
- **System Architect**, herramientas CASE para Análisis y Diseño, incluye técnicas estructuradas y orientadas a objetos.
- **Win A&D**, herramientas CASE para análisis y diseño, incluye técnicas estructuradas y orientadas a objetos.
- **CRADLE**, conjunto de herramientas CASE integradas que dan soporte a la planificación estratégica, análisis y diseño.
- **PowerDesigner**, herramienta CASE de análisis y diseño, incluye capacidades de generación relacional y con orientación a objetos.
- **SilverRun**, conjunto integrado de de herramientas CASE para el modelado de negocios.
- **ERwin**, conjunto integrado de de herramientas CASE que brinda productividad en su diseño, generación, y mantenimiento de aplicaciones.
- **EasyCASE**, centro de productos para procesos, modelamiento de datos y eventos.
- **Oracle Designer**, es un juego de herramientas para guardar las definiciones que necesita el usuario y automatizar la construcción rápida de aplicaciones cliente/servidor flexibles y gráficas.

- **SNAP,** proporciona el ambiente integral de trabajo, brindando la posibilidad de construir sistemas de inmejorable calidad, adheridos a los estándares Arquitectura de Sistemas de Aplicaciones (SAA) de IBM.

A continuación se describen los principales componentes y las funcionalidades, de las herramientas CASE más utilizadas a nivel mundial.

System Architect

System Architect conocido por sus siglas (SA), es una herramienta CASE que provee soporte para variadas técnicas de desarrollo de los sistemas de información. Dichas técnicas están asociadas a las principales metodologías actualmente en uso. SA permite generar automáticamente plantillas de código en varios lenguajes de programación y también esquemas de implementación para gestores de bases de datos relacionales.

SA es considerado un Upper Case, que puede ser integrado a la mayoría de los generadores de código. Traduce modelos de entidades, a partir de la enciclopedia, en esquemas para Sybase, DB2, Oracle u Oracle 7, Ingress, SQL Server, RDB, XDB, Progress, Paradox, SQL Base, AS400, Interbase, OS/2, DBMS, Dbase 111, Informix, entre otros. Genera también Windows DDL, definiciones de datos para lenguaje C/C++ y estructuras de datos en Cobol. Posee esquemas de seguridad e integridad a través de contraseñas que posibilitan el acceso al sistema en diversos niveles, pudiéndose integrar a la seguridad de la red Novell o Windows/NT de ser necesario. Posee también con un completo sistema de ayuda sensible al contexto. Posee un módulo específico para ingeniería reversa o inversa. SA es una herramienta CASE de última generación, creada específicamente para la arquitectura.

EasyCASE

EasyCASE Profesional - el centro de productos para procesos, modelamiento de datos y eventos, e ingeniería de base de datos - es un producto para la generación de esquemas de base de datos e ingeniería reversa. Trabaja para proveer una solución comprensible para el diseño, consistencia y documentación del sistema en conjunto.

Esta herramienta permite automatizar las fases de análisis y diseño dentro del desarrollo de una aplicación, para poder crear las aplicaciones eficazmente – desde procesamiento de transacciones a la aplicación de bases de datos de cliente/servidor, así como sistemas de tiempo real -. Para un diseño legítimo y modelamiento de datos, procesos y eventos, permite crear y mantener diagramas de flujo de

datos, diagramas de entidad-relación, mapas de estructura y más. EasyCASE soporta una gama amplia de metodologías estructuradas, permitiendo escoger los métodos más apropiados para realizar las tareas.

El verdadero poder de EasyCASE se encuentra en el soporte comprensivo al modelamiento de datos, procesos y eventos. Posee desde el editor de diagramas flexible y un diccionario de los datos integrado en formato dBASE, así como una extensa cantidad de reportes y análisis. Porque EasyCASE Profesional, una herramienta multi-usuario, es ideal para aquellos que necesitan compartir datos y trabajar en un proyecto con otros departamentos. El equipo completo puede acceder a proyectos localizados en el servidor de la red concurrentemente. Para garantizar la seguridad de los datos, existe el diagrama y diccionario de los datos que bloquean por niveles al registro, al archivo y al proyecto, y niveles de control de acceso.

Oracle Designer

Oracle Designer es un juego de herramientas para guardar las definiciones que necesita el usuario y automatizar la construcción rápida de aplicaciones cliente/servidor flexibles y gráficas. Integrado con Oracle Developer, Oracle Designer provee una solución para desarrollar sistemas empresariales cliente/servidor de segunda generación.

Sofisticadas aplicaciones cliente/servidor pueden ser 100% generadas usando la lógica de la aplicación y el módulo de componentes reutilizables. Oracle Designer también habilita la captura del diseño de sistemas existentes, salvaguardando la versión actual.

Todos los datos ingresados por cualquier herramienta de Oracle Designer, en cualquier fase de desarrollo, se guardan en un repositorio central, habilitando el trabajo fácil del equipo y la dirección del proyecto.

Muchas metodologías diferentes para base de datos y desarrollo de aplicaciones existen actualmente. Oracle Designer no fuerza al uso de alguna metodología específica, pero en cambio proporciona un juego de herramientas que le permiten la utilización de la metodología de desarrollo que se elija.

PowerDesigner

PowerDesigner es una suite de aplicaciones de Powersoft para la construcción, diseño y modelado de datos a través de diversas aplicaciones. Es la herramienta para el análisis, diseño inteligente y construcción sólida de una base de datos y un desarrollo orientado a modelos de datos a nivel físico y

conceptual, que dan a los desarrolladores cliente/servidor la más firme base para aplicaciones de alto rendimiento.

Esta suite cuenta con los siguientes productos:

- PowerDesigner ProcessAnalyst.
- PowerDesigner DataArchitect.
- PowerDesigner AppModeler.
- PowerDesigner WarehouseArchitect.
- PowerDesigner MetaWorks.
- PowerDesigner Viewer.

ERwin

PLATINUM ERwin es una herramienta de diseño de base de datos, que brinda productividad en diseño, generación y mantenimiento de aplicaciones. Desde un modelo lógico de los requerimientos de información, hasta el modelo físico perfeccionado para las características específicas de la base de datos diseñada, ERwin permite visualizar la estructura, los elementos importantes y optimizar el diseño de la base de datos. Genera automáticamente las tablas y miles de líneas de procedimientos almacenados y *triggers* para los principales tipos de base de datos.

ERwin hace fácil el diseño de una base de datos. Los diseñadores de bases de datos sólo apuntan y pulsan un botón para crear un gráfico del modelo Entidad Relación (ER) de todos sus requerimientos de datos y capturar las reglas de negocio en un modelo lógico, mostrando todas las entidades, atributos, relaciones y llaves importantes.

Más que una herramienta de dibujo, ERwin automatiza el proceso de diseño de una manera inteligente. Por ejemplo, ERwin habilita la creación de un diccionario de atributos reutilizables, asegurando la consistencia de nombres y definiciones para sus bases de datos.

Se mantienen las vistas de la base de datos como componentes integrados al modelo, permitiendo que los cambios en las tablas sean reflejados automáticamente en las vistas definidas. La migración automática garantiza la integridad referencial de la base de datos.

ERwin establece una conexión entre una base de datos diseñada y una base de datos, permitiendo transferencia entre ambas y la aplicación de ingeniería reversa. Usando esta conexión, ERwin genera automáticamente tablas, vistas, índices, reglas de integridad referencial (llaves primarias, llaves foráneas), valores por defecto y restricciones de campos y dominios.

ERwin soporta principalmente bases de datos relacionales SQL y bases de datos que incluyen Oracle, Microsoft SQL Server, Sybase, DB2, e Informix. El mismo modelo puede ser utilizado para generar múltiples bases de datos, o convertir una aplicación de una plataforma de base de datos a otra.

1.5 Sistemas manejadores de bases de datos

Inicialmente, en los años cuarenta, los sistemas de archivos generados a través de lenguajes de programación no propietarios como Cobol y Fortran (vigentes en la actualidad), permitían almacenar los datos a través de archivos planos con funciones básicas de lectura y escritura sobre ellos. En el año 1964, se concibieron los primeros gestores de base de datos (Database Management System: DBMS), por medio de los cuales se pretende dar un viraje a los sistemas de archivos, los cuales se limitan a la estructuración del almacenamiento físico de los datos.

A lo largo de los últimos treinta años esta tecnología se ha estado continuamente actualizando. Los primeros sistemas de bases de datos estuvieron basados en el uso de archivos separados Método de Acceso Secuencial Indexado (ISAM[1]) y Método de Acceso Virtual de Almacenamiento (VSAM[2]), que son ejemplo de sistemas manejadores de archivos. Partiendo de esta tecnología, se pasó a la integración de los datos en una única colección (base de datos). La gestión o manipulación de éstos es llevada a cabo por los Sistemas Gestores de Bases de Datos (SGBD). (11)

Los SGBD son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un

[1] **ISAM** son siglas de Indexed Sequential Access Method se trata de un método para almacenar información a la que se pueda acceder rápidamente. ISAM fue desarrollado originalmente por IBM y en la actualidad forma parte del almacenamiento básico de muchos sistemas de bases de datos, tanto relacionales como de otros modelos.

[2] Virtual Storage Access Method (VSAM) es un esquema de almacenamiento de IBM del sistema operativo OS/VS2 utilizado también en la arquitectura.

lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes y acrónimos, respectivamente, de SGBD o *DataBase Management System* en su expresión inglesa.

Algunos autores han expuesto su criterio sobre este término, quedando definido algunos conceptos:

- DBMS consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos. (12)
- El SGBD es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma. (6)

Se puede decir que el propósito general de los sistemas de gestión de base de datos, es manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de los datos.

Un SGBD es un conjunto de programas, que se encargan de manejar la creación y todos los accesos a las bases de datos y está compuesto por:

a. ***Data Definition Language* (DDL):** Lenguaje de definición de datos. (1) Por medio de este, el DBMS identifica las descripciones de los elementos de los esquemas y almacena la descripción del esquema en el catálogo del DBMS, además especifica el esquema conceptual e interno (Base de datos Almacenada).
b. ***Data Manipulation Language* (DML):** Lenguaje de manipulación de datos. Permite la manipulación de las operaciones de inserción, eliminación y modificación. (1)
c. ***Structured Query Language* (SQL):** Lenguaje de consulta estructurado.

Algunos de los objetivos de los manejadores de bases de datos son:

- Crear una colección integrada de datos disponibles a una amplia variedad de usuarios.
- Proveer calidad e integridad de los datos.
- Asegurar la privacidad a través de medidas de seguridad.
- Permitir un control centralizado de la base de datos, para una administración más eficiente.

- Los DBMS más comunes son Oracle, SqlServer, Informix, Sysbase, MySQL, PostgreSql, Magic, Firebird. (1)

Con los DBMS se crea el concepto de administración de los datos, por medio de actividades integradas que permiten verlos físicamente en un solo almacenamiento pero lógicamente se manipulan a través de esquemas compuestos por estructuras donde se establecen vínculos de integridad, métodos de acceso y organización física sobre los datos, permitiendo así obtener valores agregados de utilización tales como: manejo de usuarios, seguridad, atomicidad e independencia física y lógica de los datos, entre otros.

Un SGBD se divide en módulos que tratan cada una de las responsabilidades del sistema general. Los componentes funcionales *(Ver anexo 1).*

En sí, un sistema manejador de base de datos es el corazón de la base de datos ya que se encarga del control total de los posibles aspectos que la puedan afectar.

1.5.1 Structured Query Language (SQL)

Los orígenes del SQL están ligados a los de las bases de datos relacionales. En el año 1970, E. F. Codd propone el modelo relacional y asociado a éste, un sublenguaje de acceso a los datos basado en el cálculo de predicados. Basándose en estas ideas, los laboratorios de IBM definen el lenguaje *Structured English QUEry Language* (SEQUEL) que más tarde sería ampliamente implementado por el SGBD experimental System R, desarrollado en el año 1977, también por IBM. Sin embargo, fue Oracle quien lo introdujo por primera vez en el año 1979 en un programa comercial.

El SQL pasa a ser el lenguaje por excelencia de los diversos SGBD relacionales surgidos en los años siguientes y es por fin estandarizado en el año 1986 por el Instituto Nacional Estadounidense de Estándares (ANSI[3]), dando lugar a la primera versión estándar de este lenguaje, el SQL-86 o SQL1. Al año siguiente este estándar es también adoptado por la Organización Internacional de Estandarización (ISO).

[3] **ANSI**, por sus siglas en inglés: American National Standards Institute es una organización sin ánimo de lucro que supervisa el desarrollo de estándares para productos, servicios, procesos y sistemas en los Estados Unidos. Miembro de la ISO y de la Comisión Internacional Electrónica (IEC)

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

En la actualidad el SQL es el estándar que usan la inmensa mayoría de los SGBD comerciales, y aunque la diversidad de añadidos particulares que incluyen las distintas implementaciones comerciales del lenguaje es amplia, el soporte al estándar SQL-92 es general y muy amplio.

1.5.2 Gestores de bases de datos

Existen diferentes gestores de bases de datos, que son utilizados en todo el mundo y que han surgido a lo largo de todos estos años como paradigmas dentro de proceso de desarrollo y perfeccionamiento del manejo de los datos, marcando pautas significativas en la calidad de los nuevos productos. En el mundo se utilizan gran variedad de ellos, fundamentalmente las principales potencias de desarrollo. Se listarán todos los SGDB existentes hasta el momento, clasificados de acuerdo a su poder adquisitivo en el mercado internacional de software y se realizará un análisis por regiones del grado de aplicabilidad de los SGBD, para determinar los más usados comúnmente.

Se pueden clasificar en tres grandes grupos:

1) **Sistemas Gestores de Bases de Datos libres**

   PostgreSQL Licencia BSD.

   MySQL Licencia Dual, depende del uso.

   Firebird basada en la versión 6 de InterBase.

   SQLite Licencia Dominio Publico.

   DB2-Express-C.

   Apache Derby.

2) **Sistemas Gestores de Bases de Gratuitos.**

   Microsoft SQL Server Compact Edition

   Sybase ASE Express Edition para Linux (Edicion gratuita para Linux)

3) **Sistemas Gestores de Bases de Comerciales.**

   Advantage Database.

Dbase.

FileMaker.

Fox Pro.

IBM DB2 Universal Database (DB2 UDB).

IBM Informix.

Interbase de CodeGear, filial de Borland.

MAGIC.

Microsoft Access.

Microsoft SQL Server.

NexusDB.

Open Access.

Oracle.

Paradox.

PervasiveSQL.

Progress (DBMS).

Sybase ASE.

Sybase ASA.

Sybase IQ.

WindowBase.

A partir de una extensa y profunda revisión de trabajos en Internet y de otras fuentes revisadas, la autora de la presente investigación, al hacer un análisis, fundamentalmente de las principales potencias en el desarrollo de software a nivel mundial que trabajan con el procesamiento de los datos, y que utilizan diversidad de SGBD como: los Estados Unidos, Europa, China y la India; arribó a la siguiente conclusión: a nivel mundial los SGBD más comúnmente utilizados son: Oracle, SqlServer, Informix, Sysbase, MySQL y PostgreSql.

1.5.2.1 Sistema gestor bases de datos SQL Server

SQL Server 7.0 fue la primera versión con verdadera interfaz gráfica para los usuarios. Fue sucedido por SQL Server 2000, que fue la primera edición que se lanzara con una variante para la arquitectura IA-64.

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Se caracteriza por:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Microsoft SQL Server 7.0

Microsoft SQL Server 7.0 constituye un lanzamiento determinante para los productos de bases de datos de Microsoft, continuando con la base sólida establecida por SQL Server 6.5; como la mejor base de datos para Windows NT. Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de productos significativos para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos. (13)

Microsoft SQL Server 2000

SQL Server 2000 es un potente motor de bases de datos de alto rendimiento capaz de soportar millones de registros por tabla con un interfaz intuitiva y con herramientas de desarrollo integradas como Visual Studio 6.0 o .NET, además incorpora un modelo de objetos totalmente programable (SQL-DMO) con el que podemos desarrollar cualquier aplicación que manipule componentes de SQL Server, es decir, hacer aplicación para crear bases de datos, tablas, DTS, *backups*, etc., todo lo que se puede hacer desde el administrador del SQL Server y permite hacerlo no sólo en Visual C++, sino también, en Visual Basic, ASP y por supuesto en .NET. (13)

Microsoft SQL Server 2000, es un sistema *Relational Database Management System* (RDBMS), que basado en el exitoso SQL Server 7, aporta todo lo necesario para facilitar la integración de sus datos en Internet; es un servidor de datos propiamente dicho, y por menos de lo que cuesta el servidor de otros fabricantes, SQL Server 2000 ofrece, además, herramientas de análisis y gestión de almacén de datos. (14)

Microsoft SQL Server 2005

SQL Server 2005, liberado en noviembre del 2005, es el sucesor de SQL Server 2000. Incluye soporte nativo para la gestión de datos XML, además de datos relacionales. También se ha mejorado con nuevos algoritmos de indexación y mejores sistemas de recuperación de errores. Se ha hecho más granular el permiso y el control de acceso, así como la consulta del procesador, se encarga de la ejecución simultánea de las solicitudes de información de manera más eficiente. (13)

SQL Server 2005 está creado para reducir el tiempo muerto tanto planeado como inesperado, provee soluciones para la recuperación de desastres y provee de mayor disponibilidad del sistema a usuarios de la base de datos a través de tecnologías de alta disponibilidad. (13) Lo nuevo de SQL Server 2005 es que tiene mejoras en la seguridad, mejoras en el motor relacional, integración con .NET Framework, tipo XML nativo y el lenguaje XQuery, *Service Broker* y Servicios Web XMI. Sus principales características son su facilidad de uso, su sólida base de datos cliente y su integración con Microsoft Visual Studio 2005.

Microsoft SQL Server 2008

La noticia de ésta versión de Microsoft SQL, se dio a conocer en noviembre del 2007. Propone la presentación de un nivel casi nulo en el tiempo de inactividad. Incluirá también el apoyo estructurado y semi-estructurado de datos, incluidos los formatos digitales de imágenes, audio, videos y otros datos multimedia. Incluye nuevos tipos de datos especializados de la fecha y la hora, tipos de datos especiales y un tipo de ubicación determinante de los datos especializados, entre ellos el archivo. SQL Server 2008 puede ser un almacenamiento de datos *backend*[4] de diferentes variedades de datos como, XML, correo electrónico, el tiempo/calendario, archivo, documento, espacial, etc., así como realizar la

[4] Este término se relaciona con el final de un proceso.

búsqueda, la consulta, el análisis, el intercambio y la sincronización a través de todos los tipos de datos.

Se puede concluir que Microsoft SQL Server, constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, Sybase ASE, PostgreSQL o MySQL.

1.5.2.2 Sistema gestor bases de datos MySQL

Se comenzó con la intención de usar MSQL para conectar ciertas tablas utilizando las propias rutinas rápidas de bajo nivel (ISAM). Sin embargo y tras algunas pruebas, se llegó a la conclusión que MSQL no era lo suficientemente rápido o flexible para las necesidades que existían. Esto provocó la creación de una nueva interfaz SQL para la base de datos pero casi con la misma interfaz de Programación de Aplicaciones (API[5]) que MSQL. Esta API fue diseñada para permitir código de terceras partes que fue escrito para poder usarse con MSQL y ser fácilmente portado para el uso con MySQL.

La derivación del nombre MySQL no está clara. El directorio base y un gran número de las bibliotecas y herramientas han tenido el prefijo "my" por más de 10 años. Sin embargo, la hija del co-fundador Monty Widenius también se llama My. Cuál de los dos dió su nombre a MySQL todavía es un misterio, incluso para sus creadores. (15)

- MySQL es un sistema de gestión de bases de datos.
- MySQL es un sistema de gestión de bases de datos relacionales.
- MySQL software *Open Source*.
- El servidor de bases de datos MySQL es muy rápido, fiable y fácil de usar.
- MySQL Server trabaja en entornos cliente/servidor o incrustados.
- Una gran cantidad de software esta disponible para MySQL.

Como se pudo apreciar MySQL, es el sistema de gestión de bases de datos SQL Open Source más popular, lo desarrolla, distribuye y soporta MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB es una

[5] Una **API** de sus siglas en ingles *Application Programming Interface*, es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos), que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

compañía comercial, fundada por los desarrolladores de MySQL, es una compañía Open Source de segunda generación que une los valores y metodología Open Source con un exitoso modelo de negocio. MySQL AB pertenece a Sun Microsystems desde el mes enero del año 2008.

1.5.2.3 Sistema gestor bases de datos PostgreSQL

PostgreSQL es una base de datos relacional de objetos-sistemas de gestión (ORDBMS) que se ha desarrollado en diversas formas desde el año 1977. Comenzó como un proyecto denominado Ingres en la Universidad de California en Berkeley. Ingres fue más tarde desarrollado comercialmente por relacionales *Technologies/Ingres Corporation*. (16)

En 1986 otro equipo dirigido por Michael Stonebraker de Berkeley continúo el desarrollo del código de Ingres para crear un objeto de base de datos relacional llamado Postgres. En 1996, debido a un nuevo esfuerzo de código abierto y el aumento de la funcionalidad del software, Postgres fue renombrado a PostgreSQL y, después de un breve tiempo como Postgre95. PostgreSQL está todavía en fase muy activa de desarrollo en todo el mundo de un equipo de desarrolladores por fuente abierta y de los contribuyentes. (16)

PostgreSQL es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD[6]. Como otros proyectos open *source*, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PostgreSQL *Global Development Group* (PGDG). (17)

PostgreSQL esta altamente considerada como la más avanzada base de datos de código abierto en el mundo, tiene una gran cantidad de características que, normalmente, sólo se encuentran en las bases de datos comerciales tales como DB2 u Oracle. A continuación se muestra una breve lista de algunos de los rasgos esenciales, especialmente de PostgreSQL 7.1.x.

- Objeto-relacional DBMS.
- Altamente extensible.

[6] La licencia BSD es una licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Pertenece al grupo de licencias de software libre. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio publico. La licencia BSD al contrario de la GPL permite el uso del código fuente en software no libre.

- SQL completa de apoyo.
- Integridad referencial.
- API Flexible.
- Cliente /Servidor. (16)

Como se ha podido apreciar PostgreSQL es un proyecto de código abierto, que por definición esto nos ofrece la posibilidad de obtener su código fuente, usar el programa y modificarlo sin los límites de un software propietario. Esta posibilidad de ser una fuente abierta, permite que tengamos acceso a la honesta evaluación de rendimientos y estadísticas, cosa que empresas como Oracle prohíbe. PostgreSQL brinda la posibilidad de que se pueda modificar su código y que cada cual lo ajuste a sus necesidades.

1.5.2.4 Sistema gestor bases de datos SysBase

Sybase IQ es un motor de bases de datos altamente optimizado para inteligencia empresarial, desarrollado por la empresa Sybase; diseñado específicamente para entregar resultados más rápidos en soluciones de inteligencia empresarial analítica de misión crítica, almacenes de datos y generación de reportes. Combina velocidad y agilidad, con un bajo costo total de propiedad, lo que permite a las empresas llevar a cabo análisis de datos y generación de reportes antes impensables, imprácticos o costosos. La más reciente versión de Sybase IQ es la 12.6.

Entre sus principales características se pueden mencionar:

- Rapidez.
- Menor costo total de propiedad.
- Facilidad de uso
- Escalabilidad.
- Flexibilidad.

El SGBD Sybase soporta lo sistemas operativos: HP-UX, Microsoft Windows, AIX, Solaris y GNU/Linux. En la actualidad importantes proyectos de empresas de sector financiero utilizan este sistema gestor de base de datos, aprovechando todas sus funcionalidades.

1.5.2.5 Sistema gestor bases de datos Informix

Informix es una familia de productos RDBMS de IBM, adquirida en el año 2001 a una compañía (también llamada Informix o Informix Software) cuyos orígenes se remontan al año 1980.

El DBMS Informix fue concebido y diseñado por Roger Sippl a finales de los años setenta. La compañía Informix fue fundada en el año 1980, salió a bolsa en el año 1986 y durante parte de los años noventa fue el segundo sistema de bases de datos, más popular, después de Oracle. Sin embargo, su éxito no duró mucho y para el año 2000, una serie de tropiezos en su gestión había debilitado seriamente a la compañía desde el punto de vista financiero. (7)

En el año 2001, IBM, impulsada por una sugerencia de Wal-Mart (el mayor cliente de Informix) compró Informix; tenía planes a largo plazo tanto para Informix como para DB2, compartiendo ambas bases de datos tecnología de la otra. A principios del año 2005, IBM lanzó la versión 10 del *Informix Dynamic Server* (IDS). (7)

Con este software de gestión de datos y bases de datos autocargables de IBM, el usuario no sólo accede a una solución de almacenamiento, sino que también tiene la posibilidad de acceder y analizar datos al instante.

1.5.2.6 Sistema gestor bases de datos Oracle. Características de sus productos

El manejador de bases de datos Oracle, surgió a finales de los años setenta y a principio de los años ochenta. George Koch y su equipo, fueron los primeros en desembarcar en el terreno de Oracle en el año 1982, durante un proceso de evaluación de sistema de gestión de bases de datos para una importante aplicación comercial. Oracle, conocida entonces como Relational Software, tenía poco más de veinticinco empleados en aquel tiempo y solo unos pocos clientes importantes. Sin embargo, cuando se completó el estudio, Oracle fue declarada vencedora. George Koch afirmó que el SGBD Oracle era técnicamente el mejor producto del mercado. Actualmente es el mayor y más usado Sistema Manejador de Bases de Datos Relacionales (RDBMS) en el mundo. La Corporación Oracle ofrece este RDBMS como un producto incorporado a la línea de producción. Además incluye varias generaciones de desarrollo de aplicaciones, herramientas de reportes y utilitarios.

Oracle es básicamente una herramienta cliente-servidor para la gestión de bases de datos, es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio, hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general. Es un manejador de

bases de datos relacional que hace uso de los recursos de los sistemas informáticos en todas las arquitecturas de hardware, lo que permite garantizar su aprovechamiento en ambientes cargados de información, por su capacidad de almacenar y acudir a los datos de forma recurrente.

Oracle es soportado en computadoras personales (PC), microcomputadoras y computadoras con procesamiento paralelo masivo. Soporta unos diecisiete idiomas, funciona automáticamente en más de 80 arquitecturas de hardware y software distintos sin tener la necesidad de cambiar una sola línea de código. Esto es porque más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos. Oracle es un sistema comercial que aporta un SGBD que ofrece las particularidades básicas para trabajar en entornos multi-usuarios[7]. Como sistema gestor de bases de datos, es actualmente uno de los paquetes de software más ampliamente extendidos en todas las compañías que tienen que gestionar una cantidad importante de información. Oracle es uno de los sistemas más conocidos, que alcanza hoy en día un buen nivel de madurez y de profesionalidad gracias especialmente a:

- Su transportabilidad, funciona hoy en día sobre decenas de plataformas.
- La potencia de sus instrumentos de desarrollo de aplicaciones.
- La riqueza de su diccionario de datos.
- Los mecanismos encargados de la seguridad y la confidencialidad.
- Una experiencia probada sobre el terreno y una buena presencia de Oracle a nivel de formación, consejo y soporte técnico.

La arquitectura de bases de datos descentralizada tiene como herramienta fundamental para su implementación Oracle Database 10g release 2. Esta versión 10g de Oracle salió al mercado en febrero del año 2004, primero en su versión para UNIX y posteriormente en sus versiones para Linux y Windows. La novedad más llamativa de esta versión es que pone la "g" en el nombre de la versión. Es la capacidad de estos servidores de funcionar según el paradigma de "Grid[8]" o rejilla. La novedad

[7] En general se le llama multiusuario al sistema operativo o programa que es capaz de proveer servicio y procesamiento a múltiples usuarios simultáneamente (tanto en paralelismo real como simulado), Se puede utilizar en aplicaciones de base de datos.

[8] La computación en ***grid*** o en malla es un nuevo paradigma de computación distribuida en el cual todos los recursos de un número indeterminado de computadoras son englobados para ser tratados como un único superordenador de manera transparente.

principal de Oracle 10g descansa precisamente en la preparación de dicho software para poder encajar en este modelo. Además, como parte de esta manera de entender el negocio, Oracle también ha hecho cambios en el área de marketing. Así, la letra "i" que se asociaba desde hace seis años a la marca y que representaba la entrada en la era de Internet de la compañía ha sido sustituida por la "g" de Grid, de tal forma que la anterior versión Oracle 9i cambia a Oracle 10g. No obstante, ciertos usuarios valoran más las mejoras en la administración y la integración de algunos elementos que previamente no funcionaban correctamente juntos.

Al instalar la base de datos, también se instala el nuevo Oracle Enterprise Manager Database Control, basado en la Web, que será la herramienta primaria para el manejo de la base de datos y establece un nuevo estándar en cuanto a facilidad de uso, ya que es un entorno visual e intuitivo, sin necesidad de usar el texto como medio de comunicación con la base de datos.

La suite de productos de Oracle contempla las herramientas siguientes:

1) **Oracle Database 10g Standard Edition:** Esta versión fue conocida como Servidor de grupos de trabajo (*Workgroup*). Es una base de datos con características completas para pequeñas y medianas empresas que requieren el desempeño, la disponibilidad y la seguridad de la base de datos a un bajo costo. Es fácil de instalar y configurar; es la opción segura para desarrollar e implementar de manera económica las aplicaciones de la base de datos. Este producto está considerado una base de datos multiusuario pero con un número limitado de usuarios, y actualmente existe para Windows, Unix y Linux.

2) **Oracle Database 10g Enterprise Edition:** Es ideal para empresas que necesitan soportar grandes volúmenes de transacciones en línea y consulta intensiva de datos [(18)]. Ofrece confiabilidad, escalabilidad y desempeño de primer nivel para configuraciones en *cluster*[9] y en un solo servidor. Oracle Database 10g Enterprise Edition propone desempeño y escalabilidad de récord absoluto para el procesamiento de transacciones y depósitos de datos de gran escala en Windows, Linux y servidores UNIX. [(1)]

[9] El término ***cluster*** se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora. Hoy en día juegan un papel importante en la solución de problemas de las ciencias, las ingenierías y del comercio moderno.

3) **Oracle Database 11g Enterprise Edition:** Ofrece las más completas características para soportar el procesamiento de transacciones más exigentes, inteligencia de negocios, y aplicaciones para la administración de contenido. Brinda protección ante las fallas del servidor, fallas del sitio, errores humanos y reducción del tiempo de baja programado. Asegura la protección de datos con seguridad única en el nivel de filas, auditorías detalladas y encriptación transparente de datos. Incluye data *warehousing* de alto desempeño, procesamiento analítico online y características de extracción de datos.

4) **Oracle Database Lite:** Es una solución completa para el desarrollo, despliegue y gestión de aplicaciones para entornos móviles. [19] Proporciona a los usuarios la posibilidad de aumentar su eficiencia, su productividad y su capacidad de respuesta de la fuerza de trabajo móvil, al tiempo que reduce los costos y mejora la satisfacción de los clientes. Basado en la infraestructura de Oracle Grid Computing, Oracle Database Lite 10g brinda a los clientes una arquitectura de tecnología confiable y segura que permite un mayor desempeño y escalabilidad incremental para soportar miles de usuarios concurrentes. [20]

5) **Oracle TimesTen In Memory Database:** Es parte de los productos de Oracle Database, ofrece un rendimiento en tiempo real mediante el cambio de la hipótesis en torno a donde residen los datos en tiempo de ejecución para dar soporte a las necesidades de gestión de datos de las aplicaciones con desempeño clave. Mediante la gestión de datos en la memoria y, la optimización de las estructuras de datos y algoritmos de acceso en consecuencia, la ejecución de operaciones de base de datos con la máxima eficacia [18]. Se planea que las versiones futuras incluyan mayor compatibilidad e interoperabilidad con Oracle Database. [1]

6) **Oracle Berkeley DB:** Familia de tres bases de datos embebidas, de origen abierto y alto desempeño. [1] Ofrece muy alto desempeño, confiabilidad, escalabilidad y disponibilidad en las aplicaciones. Completamente integrado en la aplicación e invisible a los usuarios finales [19]. Nuevas versiones de Berkeley DB y Berkeley DB Java Edition fueron lanzadas en septiembre del año 2006, y Berkeley DB XML 2.3 fue lanzada en diciembre del mismo año. [1]

7) **Oracle Enterprise manager 10g Grid Control:** Oracle Enterprise manager 10g Grid Control es la consola central de administración y el entorno que automatiza las tareas administrativas para el conjunto de sistemas implicados en un entorno grid. Con OEM Grid Control, Oracle 10g automatiza la instalación, configuración y clonación de servidores de aplicación y de bases de

datos sobre múltiples nodos. Este entorno puede utilizarse tanto para la adición de nuevos sistemas como para aplicar parches o añadir utilidades a sistemas ya existentes. También mantiene la sincronía entre los nodos. (1)

8) **Oracle Application Server 10g:** Oracle Application Server 10g proporciona una plataforma para desarrollar y ejecutar aplicaciones empresariales, integrando muchas funciones, por ejemplo, un entorno de ejecución para Web Services y J2EE, complementos de Business Intelligence o una Web caché, entre otras; aparte de características especialmente enfocadas al grid. (1) Application Server 10g, puede ser muy diferente dependiendo de la manera en que ha instalado y configurado el software. (21)

9) **Real Application Cluster (RAC):** Oracle Real Application *Clusters* permite que una única base de datos se expanda por múltiples nodos en un grid o red, uniendo los recursos de varias máquinas. Esto que requería un proceso en versiones anteriores del servidor se puede hacer inmediatamente en Oracle 10g y, se puede empezar a balancear el flujo de trabajo hacia la nueva máquina que se incorpora al grid, a la vez que abandonarla cuando ya no es necesario. Otros sistemas de bases de datos no pueden hacer ésto dinámicamente cuando la base de datos se encuentra ejecutándose. El software de cluster en Oracle 10g simplifica el proceso eliminando la necesidad de adquirir, instalar y configurar estas herramientas de terceros. Se pueden añadir servidores a la vez que eliminarlos en un cluster Oracle sin tiempo de inactividad, es decir, sin detener la base de datos, sin importar tampoco la plataforma donde se encuentra instalado el servidor. (1) Oracle RAC es un conjunto de bases de datos, con una arquitectura de memoria en cache que supere las limitaciones del tradicional compartido-nada-disco compartido y enfoques para proporcionar la alta escalabilidad y soluciones de base de datos disponibles para todas sus aplicaciones de negocio.

1.6 Seguridad de base de datos

La información es uno de los activos más importantes de las entidades. Es indudable que cada día las entidades dependen en mayor medida de la información y de la tecnología, y que los SI están más soportados por la tecnología, frente a la realidad de hace pocas décadas.

Por otra parte, hace unos años la protección era más fácil, con arquitecturas centralizadas y terminales no inteligentes, pero hoy en día los entornos son realmente complejos, con diversidad de plataformas y proliferación de redes, no sólo internos sino también externos, incluso con enlaces internacionales.

Entre las plataformas físicas (hardware) pueden estar: ordenadores grandes y medios ordenadores departamentales y personales, solos o formando parte de red, e incluso ordenadores portátiles. Esta diversidad acerca la información a los usuarios y hace mucho más difícil proteger los datos, especialmente porque los equipos tienen filosofías y sistemas operativos diferentes, incluso, a veces, siendo del mismo fabricante.

Al hablar de seguridad se prefiere centrarse en la información misma, aunque a menudo se hable de seguridad informática, de seguridad de los SI, o de seguridad de las tecnologías de la información.

La protección de los datos deberá llevarse a cabo contra fallos físicos, fallos lógicos y fallos humanos (intencionados o no). Estos fallos alteran indebidamente los datos, los corrompen con lo que la base de datos ya no puede servir a los fines para los que fue creada.

El SGBD facilita normalmente mecanismos para prevenir los fallos (subsistema de control), para detectarlos una vez que se han producido (subsistema de detección) y para corregirlos después de haber sido detectados (subsistema de recuperación).

Aspectos fundamentales de la seguridad:

- Confidencialidad. No desvelar datos a usuarios no autorizados. Comprende también la privacidad (protección de datos personales).
- Accesibilidad. La información debe estar disponible.
- Integridad. Permite asegurar que los datos no han sido falseados.

La seguridad en las bases de datos abarca varios temas:

- Cuestiones éticas y legales relativas al derecho a tener acceso a cierta información.
- Cuestiones de política en el nivel gubernamental, institucional o corporativo relacionadas con la información que no debe estar disponible para el público.
- Cuestiones relacionadas con el sistema.
- Necesidad en algunas organizaciones de identificar múltiples niveles de seguridad y de clasificar los datos y los usuarios según estos niveles.

El SGBD debe proveer técnicas que permitan a ciertos usuarios tener acceso a porciones selectas de una base de datos, sin tener acceso al resto. Por lo regular, un SGBD cuenta con un subsistema de seguridad de autorización de la base de datos, que se encarga de garantizar la seguridad de porciones de la base de datos, contra el acceso no autorizado.

Pudiera decirse que es necesario mantener en mente una simple regla: protección en profundidad. Entre más acciones se tomen para incrementar la protección de una base de datos, menor será la probabilidad de que un atacante tenga éxito exponiendo o abusando de cualquier información almacenada. Un buen diseño del esquema de la base de datos y de la aplicación, basta para lidiar con sus mayores temores.

1.6.1 Criterios de evaluación de seguridad

Es innegable el hecho de que el sistemas de bases de datos que decidamos utilizar en una aplicación determinada, deberá ser valorado, fundamentalmente por la seguridad que brinde. Existen actualmente, criterios de evaluación de seguridad, con validez internacional que permiten clasificar cada sistema de bases de datos en distintas categorías, pudiendo determinarse en cierto grado el nivel de seguridad e integridad de los datos almacenados. En la actualidad no existe ningún estándar internacional a través del cual regirse a la hora de asegurar las bases de datos, pero si diferentes valoraciones, realizadas por grupos de expertos en el tema, que han emitidos criterios sobre su aplicación en diferentes empresas.

Se han expuesto, con anterioridad, los motivos por los cuales la seguridad es uno de los aspectos más importantes en los actuales SI, haciéndose evidente la necesidad de crear criterios y metodologías que evalúen el grado de seguridad disponible, en un determinado sistema. El hecho de que exista un estándar que arroje cierta objetividad sobre la seguridad de los productos se hace sumamente necesario.

A continuación se reseñan, brevemente, dos políticas y estándares internacionales existentes.

- **Common Criteria (CC)**
  Es un conjunto internacionalmente aprobado de normas de seguridad, que ofrece una clara y fiable evaluación de las capacidades de seguridad de los productos; al proporcionar una evaluación independiente de la capacidad de un producto para cumplir con las normas de seguridad. El propósito del criterio común, es identificar y evaluar características en productos y sistemas, que se encuentran ratificados por el estándar ISO 15408. Ofrece a los clientes más confianza en la seguridad de los productos de tecnología de la información y, conduce a decisiones más informadas. En definitiva CC proporciona un conjunto de estándares de seguridad, instaurando un lenguaje común entre fabricantes y usuarios.

- **ISO 17799**

  Es un estándar para la seguridad de la información y desarrollado para proveer una coherencia de los controles comprometidos en las mejores prácticas en información de seguridad y donde la única fuente de información es la compañía C & A *Systems Security* LTD. ISO/IEC 17799, proporciona recomendaciones de las mejores prácticas en la gestión de la seguridad de la información a todos los interesados y responsables en iniciar, implantar o mantener sistemas de gestión de la seguridad de la información. La seguridad de la Información se define en el estándar, como la preservación de la confidencialidad (asegurando que sólo quienes estén autorizados puedan acceder a la información), integridad (asegurando que la información y sus métodos de proceso sean exactos y completos) y disponibilidad (asegurando que los usuarios autorizados tengan acceso a la información y a sus activos asociados cuando lo requieran).

Se puede concluir que, para evaluar el grado de seguridad alcanzado por un SI, en necesario tener en cuanta una serie de elementos importantes, que han de ser definidos y aplicados. En primer lugar, hay que establecer criterios para poder determinar si los diferentes requisitos son cubiertos o no. También hay que tener claro que el trabajo de evaluación, no sólo debe ser objetivo, sino también repetitivo. Por lo tanto es necesario detallar una metodología de evaluación, que sea capaz de medir el trabajo realizado en pos de garantizar cierto grado de seguridad.

Es importante resaltar, que en casos en que el SI requiere la identificación o autenticación del usuario, esto lleva a dos importantes aspectos relacionados con la evaluación de la seguridad; debiendo existir una metodología y criterios para que pueda ser evaluado el grado de seguridad alcanzado.

Conclusiones del Capítulo

En este capítulo, se realizó una revisión de los diferentes temas y conceptos relacionados con el objetivo de la investigación, tratando de conformar los fundamentos teóricos necesarios, que contribuyeran a una toma de decisión en la propuesta final de una solución. Se analizaron conceptos como el de bases de datos, quedando claro a la hora de analizar otros como modelo de datos y las diferentes metodologías de diseño que existen. Además se realizó el estudio de las herramientas CASE para el diseño de bases de datos, pudiendo determinarse las más usadas a nivel internacional. Se realizó el estudio de los SGBD, haciendo un poco de historia y determinando todos los que se utilizan hoy en día, clasificándolos de acuerdo a su disponibilidad ente los desarrolladores, si son libres o propietarios; además de concluir en una explicación de los más usados a nivel mundial, resaltado

con una explicación de los mismos. Finalmente se puntualizó en algunos conceptos relacionados con la seguridad de las bases de datos, que son de especial interés en la búsqueda de una solución, al problema planteado en la presente investigación. De tal forma, quedo claro que, la seguridad nunca puede ser considerada como un objetivo que ha de alcanzarse de forma absoluta, e incluso, no se puede hablar del mismo concepto de seguridad en todos los casos.

CAPÍTULO 2: DESCRIPCION DE LA SOLUCION. PROPUESTA DEL PROCEDIMIENTO.

Introducción

En el presente capítulo se parte del concepto de seguridad de los sistemas de información, dando una breve explicación sobre algunos temas que giran alrededor del tema seguridad, haciendo especial énfasis en los relacionados con la seguridad de las bases de datos, que es el fundamento del presente trabajo. Se presentan algunos elementos, que deben tenerse en cuenta para lograr un diseño eficaz, que garantice en alguna medida la seguridad de los datos que se almacenan. Se explican cada uno de los elementos considerados por la autora, como los más significativos a tener en cuenta, para garantizar seguridad a la hora de diseñar bases de datos; puntualizando en algunos aspectos importantes a destacar como: la integridad, la seguridad, la trazabilidad y la salva guarda de la información. Finalmente, se propone el procedimiento que da solución a la problemática que motivó la investigación.

2.1 ¿Qué es seguridad?

Se puede entender como seguridad, una característica de cualquier sistema (informático o no) que nos indica que ese sistema está libre de todo peligro, daño o riesgo, y que es, en cierta manera, infalible. Particularizando, para el caso de los SI y más específicamente las bases de datos, es muy difícil conseguir un grado absoluto de seguridad (según la mayoría de expertos, casi imposible), se suaviza la definición y se pasa a hablar de fiabilidad (probabilidad de que un sistema se comporte tal y como se espera de él) más que de seguridad. Por tanto, generalmente se habla de sistemas fiables en lugar de hacerlo de sistemas seguros.

A un nivel más internacional, se puede citar un concepto de seguridad ofrecido por la ISO y por la Comisión Electrotécnica Internacional (IEC) en el año 1999 y que es el siguiente: “Seguridad es la capacidad de un producto software, de proteger los datos y la información para que personas no autorizadas, no puedan leerlos o modificarlos y que el acceso no sea denegado a personal autorizado” (22)

La calidad del software enmarca dentro de los aspectos más importantes a tener en cuenta la funcionalidad que tenga el producto, estando simplemente determinada por la seguridad que se logre sobre los datos que se almacenan.

A grandes rasgos cuando se habla se seguridad (o fiabilidad) de los sistemas de información, consiste básicamente en garantizar tres aspectos fundamentales: confidencialidad, integridad y disponibilidad.

¿Qué implica cada uno de los tres aspectos antes mencionados? La confidencialidad indica que los objetos de un sistema han de ser accedidos únicamente por elementos autorizados a ello, y que esos elementos autorizados no van a convertir esa información en disponible para otras entidades. Generalmente la confidencialidad, se refiere a la protección de los datos implicados en entornos altamente protegidos Por otro lado, la integridad significa que los objetos sólo pueden ser modificados por elementos autorizados y de una manera controlada. De manera general la integridad consiste en prevenir, detectar e impedir la modificación inadecuada de la información. Finalmente la disponibilidad indica, que los objetos del sistema tienen que permanecer accesibles a elementos autorizados, siendo lo contrario de la negación de servicio; es el proceso de prevenir, detectar e impedir la denegación inadecuada de acceso a servicios ofrecidos por un sistema, estando relacionada, además, con los mecanismos de recuperación de las bases de datos ante caídas del sistema.

El problema de la seguridad consiste en lograr que los recursos de un sistema sean, bajo toda circunstancia, utilizados para los fines previstos; para eso se utilizan mecanismos de protección.

Los Sistemas Operativos (SO) proveen algunos mecanismos de protección para poder implementar políticas de seguridad. Las políticas definen qué hay que hacer (qué datos y recursos deben protegerse de quién; es cuestión de administración), y los mecanismos determinan cómo hay que hacerlo. Esta separación es importante en términos de flexibilidad, puesto que las políticas pueden variar en el tiempo y de una organización a otra. Los mismos mecanismos, si son flexibles, pueden usarse para implementar distintas políticas.

Se hace necesario entonces, aplicar diferentes mecanismos que aseguren en cierto grado la seguridad de los sistemas de información, vistos algunos mecanismos; los que brindan los SO, que sirven para evitar las deficiencias en la seguridad, pues a partir de aquí se pudieran generar diferentes problemas en el ámbito de las empresas (pérdidas económicas, pérdidas de imágenes y pérdidas de clientes, además de problemas legales) y a nivel individual (privacidad). Particularmente es necesario buscar algún mecanismo que nos permita asegurar, en cierta medida, la seguridad en las bases de datos, ya que éstas almacenan información importante en todos los ámbitos: comerciales, militares, médicas, administrativas, judiciales, entre otros.

Hoy en día, las bases de datos son componentes cardinales de cualquier aplicación, dígase aplicaciones de escritorio o basadas en la web, pues permiten alcanzar dinamismo referente a la información que se maneja. Se ve evidenciado, pues para recuperar o almacenar cualquier información se necesita conectarse a una base de datos, enviar una consulta válida, recoger el resultado y cerrar la conexión. Actualmente, el lenguaje de consultas más utilizado en estas interacciones es el Lenguaje de Consultas Estructurado (SQL por sus siglas en inglés), y es a través de este mismo, que un atacante es capaz de hacer cualquier intromisión a un sistema. Se evidencia entonces, la realidad de que la información almacenada puede encontrarse amenazada, a pesar de ser cuantiosamente sensible o secreta, por lo que se debe tener en cuanta seriamente la protección de los datos que se almacenan en las bases de datos.

Después de haberse realizado un estudio, de los aspectos que se tienen en cuenta para asegurar la seguridad de las bases de datos, se listan, a continuación, los que comúnmente se tienen en cuenta en diferentes trabajos y por gran número de expertos en el tema; quedando claro, de manera general, que si se persigue lograr la protección de las bases de datos, garantizando en alguna media la seguridad de los datos almacenados, es necesario que se tenga presente:

- Privilegios de acceso (usuarios con contraseña).
- Control de concurrencia (sistemas de bloqueo).
- Sistema de recuperación de datos (copias de seguridad).
- Control de acceso.
- Autenticación.
- Integridad.
- Seguridad.
- Criptografía.
- Trazabilidad.
- Especificar restricciones de seguridad.

En ésta investigación en particular, se decidió tener en cuenta algunos elementos de los que anteriormente se mencionaron, para conformar el procedimiento que de respuesta a la problemática propuesta, sobre la base de las características del negocio de RN, de la organización de la base de datos que presenta una estructura que se modela a partir del modelo relacional. Al realizar la selección de los elementos a tener en cuenta para realizar un estudio y llegar a conclusiones, que

permitan establecer un procedimiento estándar; se seleccionaron, como ya se dijo, varios de los mencionados, pues se consideró que algunos tocaban los temas de manera más general, pudiendo enmarcar unos dentro de otros.

Finalmente los elementos, que se determinaron tener en cuenta para el desarrollo de este trabajo; (cuando se quiere garantizar la seguridad en una base de datos), son los que la autora consideró a partir del estudio, la revisión realizada y la experiencia personal, como aspectos determinantes a tener en cuenta, en la etapa de diseño de las bases de datos. Estos elementos son: la *integridad*, en ella podemos incluir el hecho de especificar restricciones de seguridad; la *seguridad*, donde se pueden tener en cuenta los privilegios de acceso, la autenticación, la asignación de roles y la criptografía; la *trazabilidad*, se puede llevar un control de concurrencias y la *salvaguarda de la información;* que tiene presente todo el proceso de recuperación de los datos.

Después de todo el análisis realizado anteriormente, se puede llegar a la conclusión de que es importante mantener una simple regla: la protección en profundidad. Entre más acciones se tomen para incrementar la protección de una base de datos, menor será la probabilidad de que existan huecos de seguridad en los sistemas, y se reducen las oportunidades, para que un atacante tenga éxito exponiendo o abusando cualquier información almacenada. Un buen diseño del esquema de la base de datos, teniendo en cuenta cada uno de estos elementos, basta para lidiar con sus mayores amenazas.

2.2 Integridad

Cuando se modela una realidad (toda el área del negocio) en el nivel conceptual, junto con identificar los datos que se desea manejar y cómo éstos se relacionan, surgen 'reglas' que deben cumplir dichos datos para representar lo más exactamente posible esa realidad. Es decir, el cumplimiento de estas reglas, ayuda a mantener la integridad de los datos. El término integridad de los datos, se refiere a la corrección y completitud de los datos, en una base de datos; cuando los contenidos que ésta almacena se modifican, puede perderse la integridad de muchas maneras diferentes. La mayor parte de las bases de datos están sujetas a un gran número de reglas de integridad. Estas reglas se reflejan en la especificación de la base de datos (esquema) a través de las Restricciones de Integridad (RI) de distintas maneras, como por ejemplo: definición de dominios para los datos, especificación de cardinalidad en los tipos de interrelación, relacionada con las claves primarias, relacionadas con las claves foráneas, etc.

Existen ciertas restricciones, propias de los datos, que son difíciles de representar utilizando los modelos tradicionales. Esto produce una pérdida de expresividad en los esquemas y una falta de portabilidad de los mismos, ya que en las etapas posteriores, aquellas restricciones son manejadas por las aplicaciones software que usan dichos datos, o por el administrador de bases de datos, mediante procedimientos almacenados y disparadores (*triggers*), los que deben ser programados por los desarrolladores y quedan documentados separadamente del modelo conceptual, pero se especifican durante el diseño.

La importancia de considerar las restricciones que afectan a los datos que se están modelando, radica en que la presencia de éstas conforma una garantía de la consistencia de la base de datos en el tiempo. Ahora bien, mientras más previamente se consideren las restricciones en el desarrollo; precisamente durante la etapa de diseño, más tempranamente se podrán detectar errores, depurar y modificar los esquemas de la base de datos, para así asegurar su desempeño. La atención de las restricciones en etapas posteriores puede inducir a modificaciones costosas del diseño.

Este hecho justifica todos los esfuerzos por incluir formalmente las restricciones en el diseño en los momentos iniciales del desarrollo, ya que de otra forma quedarán, en el mejor de los casos, expresadas sólo en lenguaje natural, con el consiguiente riesgo de que no sean consideradas en la implementación final.

Como el objetivo de la investigación es, elaborar y proponer un procedimiento que asegure en cierta medida la seguridad de una base de datos partiendo del diseño de la misma; se hace necesario tener en cuenta que las operaciones realizadas por los usuarios sean correctas y mantengan la consistencia de la base de datos; elementos que comprende la integridad. El hecho de garantizar la integridad de los datos avala, además, la calidad de la información que se almacena y para lograrlo es necesario aplicar restricciones de integridad.

Hasta este punto quedan claros algunos elementos. En el momento en que se va a desarrollar una base de datos, se encuentra que los modelos conceptuales son cada vez más ricos semánticamente hablando y, esto implica, que sean capaces de recoger con mayor precisión las especificaciones del dominio. El problema surge cuando queremos mantener esta semántica en todas las fases del diseño, donde la mayoría de las metodologías coinciden en la utilización del modelo relacional para transformar un esquema conceptual en uno más cercano a la implementación. Para ello, se pudieran aplicar buenas prácticas de diseño de base de datos, enfocadas a aspectos precisos, cuando se

persigue mantener segura la información almacenada. Por tal razón se aplican un conjunto de reglas, que llevan al diseñador a controlar y chequear las RI de la base de datos.

2.2.1 Restricciones de integridad

En el mundo real existen ciertas restricciones que deben cumplir los elementos en él existentes; por ejemplo, una persona sólo puede tener un número de Carnet de Identidad (CI) y una única dirección particular oficial. Cuando se diseña una base de datos se debe reflejar fielmente el Universo del Discurso (UD) que estamos tratando, o lo que es lo mismo, reflejar las restricciones existentes en el mundo real.

Los componentes de una restricción son los siguientes:

- La operación de actualización (inserción, borrado o eliminación), cuya ejecución ha de dar lugar a la comprobación del cumplimiento de la restricción.
- La condición que debe cumplirse, la cual es en general una proposición lógica definida sobre uno o varios elementos del esquema, que puede tomar uno de los valores de verdad (ciertos o falsos).
- La acción que debe llevarse a cabo dependiendo del resultado de la condición.

Una restricción de integridad es una propiedad que la base de datos debe satisfacer en cualquier instante. Existen dos tipos de restricciones: las estáticas y las dinámicas.

Las restricciones estáticas, limitan los estados permitidos de la base de datos para que reflejen exactamente la situación del UD. Comprenden desde simples restricciones de dominio (por ejemplo, la edad de una persona debe ser ente 0 y 120 años) hasta complejas relaciones entre piezas de información (por ejemplo, los proyectos dirigidos por un investigador deben ser aquellos en los que el investigador tiene asistentes de investigación).

Las restricciones dinámicas en cambio, restringen las posibles transacciones de estado de la base de datos (por ejemplo, los salarios no pueden bajar). La mayoría de los modelos semánticos no tratan restricciones dinámicas.

Existen diferentes Restricciones de Integridad (RI) provistas por distintos modelos, pero en el caso de esta investigación, se centra la atención en las restricciones que se utilizan más comúnmente en el modelo relacional En general, se pueden mencionar tres tipos de RI elementales en el modelo relacional:

- Integridad de dominio: restringe los valores que puede tomar un atributo respecto a su dominio, por ejemplo, que la edad siempre sea mayor que 18 años y menor que 65 (EDAD >= 18 - 65).
- Integridad de entidad: la clave primaria de una entidad no puede tener valores nulos y siempre deberá ser única, por ejemplo el CI.
- Integridad referencial: los valores de las claves ajenas de una tabla se tienen que corresponder con el valor de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos. Por ejemplo, si se tiene una tabla PERSONA que tiene como llave primaria IDPERSONA y otra tabla GRUPO que tiene como llave primaria el IDGRUPO. La relación entre estas tablas es de uno a muchos, por lo que IDGRUPO, pasa para la tabla PERSONA como llave foránea (FK). De esta manera se conocerían los estudiantes de un grupo.
- Otro ejemplo, se tiene la misma tabla PERSONA y una tabla PROFESOR, que hereda de ésta: entonces el IDPERSONA pasa a la tabla PROFESOR como llave primaria (PK) y como llave foránea (FK).

Las restricciones se clasifican en:

1. Inherentes.
    - Están impuestas por el modelo.
    - No tienen que ser definidas por el usuario, ya que se encuentran en el propio modelo.
    - Se activan en el momento de la definición del esquema, cuando se produce un intento de violación.
    - Se rechaza todo esquema que no cumple estas restricciones.
    - Introducen rigidez en el modelo.

2. Semánticas.
    - Impuestas por el UD.
    - Tienen que ser definidas por los diseñadores.
    - Se activan en el momento de la actualización de la base de datos.
    - Se rechaza todo ejemplar que no cumpla estas restricciones (o se ponen en marcha otros medios, a fin de que no se produzca un estado de inconsistencia).
    - Ayudan a capturar la semántica de los datos y a conseguir su consistencia.

Queda claro que los datos que se almacenan, deben cumplir ciertas reglas de integridad para lograr garantizar que sean correctos. Una vez especificada la clasificación de las RI y el objetivo que cumplen dentro del diseño de una base de datos, se prosigue con el estudio de cada una de ellas.

Al definir cada atributo sobre un dominio se impone una restricción sobre el conjunto de valores permitidos para cada atributo; a este tipo de restricción se le denomina restricción de dominio. Hay además dos reglas de integridad muy importantes (las reglas deben cumplirse todo el tiempo), que son restricciones que se deben cumplir en todas las bases de datos relacionales y en todos sus estados e instancias (las reglas deben cumplirse todo el tiempo); por lo que se tendrán en cuenta para la solución que se propone en este trabajo.

2.2.1.1 Reglas de integridad de dominio

Las reglas de integridad de dominio, son un dominio de valores posibles que pueden estar asociados con cada atributo. Los límites de dominio son la forma más elemental de restricciones de integridad; son fáciles de probar por el sistema siempre que se introduce un nuevo dato en la base de datos.

Se pueden identificar diferentes tipos de dominios. Es posible que varios atributos tengan el mismo dominio. Se puede ver que una definición adecuada de restricciones de dominio, no sólo permite probar consultas para asegurar que la comparación que se hace tiene sentido. El principio que hay detrás de los dominios de atributo, es similar al que hay detrás de la asignación de tipos a variables en los lenguajes de programación.

Al aplicar la integridad de dominio, se tiene como requerimiento que se establezca una columna que tenga un valor no nulo. Esto se logra definiendo en la declaración de una columna que es NOT NULL, cuando la tabla que contiene dichas columnas se crea por primera vez, como parte de la sentencia que se utilice en dependencia del SGBD que se utilice.

Cuando se crea una tabla, cada columna que la conforma tiene un tipo de dato especificado en el momento en que se diseñó, para que posteriormente el SGBD, asegure que solamente los datos del tipo especificado sean ingresados en la tabla. Este proceso necesario, se conoce como chequeo de validez de los datos almacenados.

Hasta aquí se puede concluir que, la integridad de dominio viene dada por la validez de las entradas para una columna determinada. Existe para restringir el tipo mediante tipos de datos, el formato

mediante reglas y restricciones de chequeo. Es positivo tener en cuenta este tipo de reglas en el momento en que se realiza el diseño, pues de esta manera, se garantiza un control sobre las entidades, los tipos de datos y se asegura que no existan valores nulos, todo esto aplicando sentencias de chequeo y apoyándose en las funcionalidades que brinde el SGBD que se utilice.

2.2.1.2 Reglas de integridad de entidad

Esta regla se aplica a las claves primarias de las relaciones base: ningún atributo que forme parte de una llave primaria puede aceptar valores nulos. Como se definió antes los valores nulos son como un valor indefinido o como ningún valor; usados en las columnas en las que se desconozca su valor, sin significar un espacio en blanco. Por definición, una clave primaria es irreducible y se utiliza para identificar de modo único los registros. Irreducible significa que ningún subconjunto de la clave primaria, sirve para identificar las tuplas de modo único. Si se permite que parte de la clave primaria sea nula, se está diciendo que no todos sus atributos son necesarios para distinguir las tuplas, con lo que se contradice la irreductibilidad. Esta regla sólo se aplica a las relaciones base y a las claves primarias, no a las claves foráneas ni alternativas.

La integridad de entidad define una fila como entidad única para una tabla determinada. La integridad de entidad exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índices y restricciones que identifiquen que ese campo es único (UNIQUE) o restricciones que determinen que es una clave primaria (PRIMARY KEY).

Queda claro que las reglas de integridad de entidad establecen que, ningún atributo de una llave primaria puede tener el valor NULL. Establece además, que la clave primaria de una tabla debe tener un valor único para cada fila de la tabla, si no la base de datos perderá su integridad. Se especifica en las sentencias de creación de una tabla y el SGBD comprueba automáticamente la unicidad del valor de la clave primaria con cada sentencia de inserción y actualización, es decir para cualquier modificación. Un intento de insertar o actualizar una fila con un valor de la clave primaria ya existente, fallará.

Cuando se está diseñando una base de datos y para cumplir con la normalización, se empiezan a crear las entidades y relaciones. El problema se presenta, cuando se tienen dos tablas, TBA y TBB, cada una con IDTBA e IDTBB, como atributos llaves respectivamente. Entonces si estas tablas tienen una relación, de muchos a muchos, surgiría una nueva tabla TBAB, con IDTBA e IDTBB como llaves. Hasta este punto todo bien, pero si tenemos otra tabla TBC que se relaciona con TBAB de muchos a

muchos también, entonces surge una nueva tabla con tres atributos como llave y como es de imaginar el modelo crecería mucho, pues en algún momento se tendrán tablas de hasta diez atributos llaves, y eso no es óptimo. A partir de aquí surge el concepto de llaves sobregadas (ALTERNATE KEY), como solución al problema. Los ALTERNATE KEY, o llaves secundarias, son cualquier llave candidata pero que no es llave primaria, y sirve como complemento, de manera que se pueda insertar el mismo elemento.

Aplicando estas reglas en particular, se asegura que ninguno de los atributos que componen la clave primaria pueda ser nulo, de esta forma se evitan inconsistencias e incongruencias en los datos almacenados. Quedó claro que a menos que exista una sola PRIMARY KEY se puede crear ALTERNATE KEY, lo que da garantía en la modelación de todo tipo de relaciones Además resalta el hecho de que, los diseñadores, deben tener en cuenta este elemento si quieren lograr un diseño que garantice la seguridad, porque partiendo de aquí, se logra que si en cualquier momento se hace referencia a una tabla, la misma esté identificada únicamente por su clave primaria, pues ésta tendrá un valor y éste será único dentro del negocio.

2.2.1.3 Reglas de integridad referencial

La integridad referencial, se aplica a las claves ajenas o extranjeras. Si en una relación hay alguna clave ajena, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos. De manera más sencilla se puede decir que, las claves ajenas de una tabla, deben corresponderse con las llaves primarias en la tabla a la que hacen referencia.

Es importante resaltar que se enmarca en términos de estados de la base de datos, es decir indica lo que es un estado ilegal, pero no dice como puede evitarse. Existen dos acciones a tener en cuenta ante esta situación, una, rechazar la operación; porque ha ocurrido alguna violación de las reglas de integridad establecidas y la otra, aceptar la operación, ya que se cumple con la debida referencia entre las claves primaria y extranjera; realizando las operaciones adicionales compensatorias que conduzcan a un estado legal.

Antes de definir las reglas de integridad referencial, se deben especificar las reglas de las claves primarias. Lo que se persigue con la aplicabilidad de estas reglas, es garantizar que la clave primaria sea única y que además la clave primaria no cambie.

1. La clave primaria debe ser única.

2. La clave primaria no debería cambiar.

Esto último es muy discutido, pues en realidad no existe una regla específica que normalice esta operación, pero a partir del estudio realizado se determinó como norma, pues garantizando ésto no se tendrá problemas con la integridad referencial, cuando se realicen modificaciones en la base de datos.

Se ha podido apreciar, a partir del estudio realizado, la importancia de tratar con rigor el tema de las claves ajenas y aplicar reglas de integridad referencial para las claves extranjeras, a partir de lo cual se resumen tres elementos determinantes en el diseño. Primeramente es importante tener en cuenta que los valores nulos solo se pueden permitir en claves que sean extranjeras y nunca en valores que constituyan claves primarias. En segundo lugar se menciona el borrado de las tuplas de una tabla, donde es de vital importancia tener en cuenta que al borrar la clave primaria las tuplas que contengan ese mismo valor como clave ajena se pondrán a nulo. Finalmente es necesario tener en cuenta alguna regla que normalice el tema de las modificaciones, es por eso que se recomienda que al modificar un valor de la clave primaria de una tupla, dicha modificación debe propagarse a todas las claves ajenas de las tuplas que corresponda en otras tablas.

Hasta aquí, resulta necesario hacer un alto, para exponer dos buenas prácticas y tendencias que existen de manera general entre los diseñadores, para trabajar todo el proceso de borrado y actualización en las tablas, pero que garantice la integridad referencial. El actualizar o eliminar, son procesos que se definen durante el diseño y cuando se definen las claves foráneas, lo que le indicara al SGBD qué acciones tomar en los casos comentados con anterioridad.

Cuando se quiere actualizar registros en cascada, se determina que en el momento de cambiar un valor del campo clave de la tabla principal, automáticamente cambiará el valor de la clave foránea de los registros relacionados en la tabla secundaria.

Si no se tiene definida esta opción, no se pueden cambiar los valores de la clave principal de la tabla principal, es decir si no se produce el cambio, el sistema devuelve un error o un mensaje informando, que los registros no se han podido modificar por infracciones de clave.

Cuando se quiere eliminar registros en cascada, se determina que en el momento en que se elimina un registro de la tabla principal automáticamente se borran también los registros relacionados, es decir a los que se hace referencia en la tabla secundaria.

En realidad consideran de manea general dos tipos de transacciones para el borrado en una base de datos: restrictivo y en cascada. El borrado restrictivo solo incluye la operación de borrado de la entidad, a diferencia del borrado en cascada que además de la operación de borrado de la entidad, incluye el borrado de las concurrencias de la relación en las que interviene dicha entidad. Cuando se aplican restricciones de integridad; lo que se recomienda siempre, el borrado siempre tendrá que ser en cascada, pues aseguran la integridad.

Se determina, que si no se tiene definida esta opción de borrado en cascada, no se podrán borrar registros de la tabla principal, si éstos tienen registros relacionados en la tabla secundaria. En este caso, si no se cumple con la regla establecida entonces no se produce el borrado y el sistema devuelve un error o un mensaje, informando que los registros no se han podido eliminar por infracciones de clave.

Establecidas estas reglas el SGBD sabrá qué hacer en cada momento.

La integridad referencial, es una propiedad deseable en las bases de datos. Gracias a la integridad referencial se garantiza que una entidad (tupla, fila o registro) siempre se relaciona con otras entidades válidas, es decir, que existen en la base de datos. Implica que en todo momento dichos datos sean correctos, sin repeticiones innecesarias, datos perdidos y relaciones mal resueltas.

Todas las bases de datos relacionales gozan de esta propiedad gracias a que el SGBD vela por su cumplimiento, pero para lograr que éste garantice que se cumplan las reglas de integridad, es importante establecerlas durante el diseño. De manera diferente ocurre en las bases de datos jerárquicas, pues éstas requieren que sean los programadores quienes aseguren que se mantenga tal propiedad en sus programas.

Particularmente, en la presente investigación se determina que con la regla de borrado, hay que tener especial cuidado, porque no se considera la mejor opción, pues muchas veces ocurre que un campo, es clave ajena, pero a su vez es clave primaria o parte de una clave primaria en otra tabla, por lo que si se aplica siempre y fielmente esta regla, se estaría rompiendo una regla aún mayor, la de no tener valores nulos, ni repetidos en una clave primaria. Entonces se cree que la regla de borrado pudiera aplicarse, pero únicamente con ciertas restricciones. De forma que se propone, que un registro de una tabla no debe borrarse, si su clave primaria es clave ajena en otra tabla y pertenece además a la clave primaria de esa otra tabla. En caso de que ésto no ocurra entonces se puede aplicar fielmente la regla de borrado mencionada anteriormente.

Hasta este punto se puede llegar a conclusiones parciales sobre el tema integridad, partiendo de que la integridad es un elemento muy importante; lo que ha quedado evidenciado después de todos los elementos expuestos, por lo que es primordial tener en cuenta durante el diseño de la base de datos todos los aspectos que tocan las reglas de integridad, pues éstas garantizan que los datos sean correctos y relevantes. La integridad nos asegura la consistencia de la información, pues basándonos en el manejo de las claves; ya sean primarias o foráneas, se garantiza que los registros a los que se hace referencia existan realmente en la base de datos, y que además no exista duplicidad de elementos, en los que coincida que dos tablas tengan como llave primaria el mismo valor, logrando un diseño más robusto y que represente fielmente los elementos del negocio que se quiera modelar, alcanzando un mayor acercamiento al mundo real.

En fin, las RI garantizan que, el contenido de la base de datos es conforme con las reglas establecidas para presentar el UD. La integridad de una base de datos significa la existencia de dos componentes importantes que son la exactitud y la completitud; es decir, que la integridad de una base de datos garantiza que todos los datos sean correctos (válidos) y relevantes.

2.3 Seguridad

Se ha repetido de manera reiterada, que los datos constituyen un recurso valioso que debe ser estrictamente controlado y gestionado al igual que cualquier otro recurso corporativo. El término seguridad hace referencia a la protección de la base de datos frente a accesos no autorizados, ya sean intencionados o accidentales, es decir, hace referencia a cualquier situación o suceso, provocados de forma intencionada o accidental, que pueda afectar adversamente a un sistema y consecuentemente a la organización.

Las bases de datos, en particular, deben tener un sistema de seguridad sólido para controlar las actividades que pueden realizarse y determinar qué información puede verse y cuál puede modificarse. Un sistema de seguridad sólido, asegura la protección de datos sin tener en cuenta cómo los usuarios obtienen el acceso a la base de datos.

Los conceptos de seguridad en bases de datos son los mismos que se plantean en cualquier otro sector de tecnologías de la información, pero en su aplicación práctica poseen muchas particularidades. Cada dato tiene consideraciones distintas. Dentro de una misma tabla, no es lo mismo conocer un campo determinado que represente un aspecto específico del mundo real, como

puede ser el "apellido" de una persona, que el campo "salario"; y de igual forma, no es lo mismo conocer el del registro que sea "bedel" que el del registro que sea "consejero delegado".

Lo que se comentó previamente, conlleva a que la seguridad sea una de las principales preocupaciones de los administradores de sistemas, redes y bases de datos. Mientras un administrador está implementando la seguridad, es natural que pueda estar preocupado por los ataques externos. Pero existe más que eso, es esencial primero implementar la seguridad dentro de la organización, para asegurar que sólo las personas autorizadas tengan acceso a los datos. Si no se establecen estas medidas de seguridad, lo más seguro es encontrar a alguien destruyendo sus datos o vendiendo los secretos de la compañía a los competidores o alguien accediendo a la información privada de otros. Es importante hacer un plan de seguridad, el cual identifique qué usuarios en la organización pueden ver determinados datos y qué actividades deben realizar en la base de datos.

Es recomendable además, en un primer nivel de control de acceso lógico implementar tecnología *firewall* (cortafuegos) como primer filtro en el que se establecen qué redes de confianza pueden acceder a la base de datos. Posteriormente, es importante analizar, la seguridad de las aplicaciones que acceden a la propia base de datos, ya que normalmente el usuario no se introduce directamente a la base de datos, sino que utiliza un cliente o una aplicación web que es realmente quien se identifica y accede al motor de la base de datos. La seguridad del código es necesaria para evitar ataques del tipo SQL- *Injection*[10], originados por la falta de validación de los datos de entrada en la aplicación, para lograr que la base de datos nunca sea atacada.

Partiendo de este análisis, se pudiera clarificar la seguridad de las bases de datos en dos categorías: seguridad del sistema y seguridad de los datos. La seguridad del sistema, cubre el acceso y uso de la base de datos a nivel del sistema, como el nombre de usuario y password, el espacio en disco destinado a los usuarios y las operaciones del sistema que los usuarios pueden ejecutar. La seguridad de los datos, cubre el acceso y uso de los objetos de la base de datos y las acciones que estos usuarios, pueden tener en los objetos.

[10] **Inyección SQL** es una vulnerabilidad informática en el nivel de la validación de las entradas a la base de datos de una aplicación. El origen es el filtrado incorrecto de las variables utilizadas en las partes del programa con código SQL. Es, de hecho, un error de una clase más general de vulnerabilidades que puede ocurrir en cualquier lenguaje de programación o de script que esté incrustado dentro de otro.

En el desarrollo de este epígrafe se aporta una visión general de cómo se deber realizar adecuadamente la gestión de la seguridad en sistemas de bases de datos; partiendo del estudio de las diferentes tendencias que se ponen en práctica a nivel mundial y de manera particular, de la experiencia personal, lográndose evitar en la medida de lo posible, los detalles técnicos concretos que puedan perder valor en el tiempo. Se da respuesta a interrogantes como: ¿Para qué sirven los roles y grupos de usuarios? ¿De qué manera afectan y ayudan a la seguridad de las aplicaciones existentes sobre bases de datos? Esto se consigue mediante: un control sobre los usuarios que acceden a la base de datos y los tipos de operaciones que están autorizados a realizar. Este control, por llamarlo de alguna manera, gestión de autorizaciones, permite crear o borrar usuarios y conceder o retirar derechos a efectuar determinados tipos de operaciones (por ejemplo: crear objetos, borrar objetos, modificar datos de un objeto, etc.).

Hasta este punto se puede decir que, seguridad es la protección frente a intentos maliciosos de robar o modificar los datos; particularmente en las bases de datos, donde estos se almacenan. Definir la seguridad se puede hacer de muchas maneras, se pueden encontrar incluso aplicaciones que tienen definida hasta el rango de horas en las que un usuario (dependiendo del rol que tenga) puede entrar a los diferentes formularios de una aplicación. Pero todo esto se hace en dependencia de los objetivos que se persigan y del grado de seguridad que se quiera garantizar.

Tratando de hacerlo de manera poco compleja, lo primero que corresponde hacer es definir todos los roles. Luego, a cada rol, definirles las pantallas a las que tendrán acceso, es decir los privilegios que tendrá dicho rol. Y al momento de crear un usuario deberá asignársele un rol, el que tendrá predefinido determinados privilegios, que le darán acceso a determinadas funciones dentro de la base de datos, para cuando este usuario se autentifique se realice la búsqueda en la base de datos y se muestren todos los elementos a los cuales se le dio acceso (según su rol) y de esa manera se le construirá el menú correspondiente para que trabaje en las áreas a que se le dio acceso. Todo con un único fin: que el acceso a unos datos concretos se produzca por quien debe y desde donde debe, y que ese alguien, que no debe, no tenga permiso, ni derecho a acceder o modificar lo que no le corresponde *(Ver anexo 2).*

Existe además, complejidad inherente en el control de accesos en bases de datos, con múltiples usuarios, con distintos permisos de acceso sobre multitud de objetos como tablas, vistas, procedimientos almacenados, etc. Por esta razón resulta difícil aplicar correctamente el conocido

principio de ”mínimo privilegio” en la configuración de accesos y por demás se hace sumamente complicado garantizar la seguridad.

A continuación se enumeran algunas de las prácticas más notables de los últimos años, tratando de no particularizar sobre diseñadores, proyectos, compañías o fabricantes; pues al fin y al cabo en esta ciencia unos aprenden de los otros y al cabo de un tiempo, todos aplican los mismos descubrimientos; estas prácticas son:

1. Las vistas.
2. Permisos de usuarios.
3. Roles.

2.3.1 Vistas

Una vista es una forma de proporcionar al usuario un modelo personalizado de la base de datos. Aunque es imposible impedir que un usuario tenga acceso directo a una relación, puede permitírsele acceso a parte de esa relación por medio de una vista. En una vista pueden implementarse controles que restrinjan los valores de entrada o salida al dominio válido de los atributos, mejorando así el nivel de integridad de la base de datos. De igual manera, el nivel de seguridad se incrementa al incluir en la vista sólo los elementos que sean considerados al alcance del usuario.

En el modelo relacional, el término “vista” tiene un significado un tanto diferente. En lugar de ser todo el esquema externo de un usuario, una vista es una relación virtual, una relación que en realidad no existe como tal. Vendría conformando una especie de tabla virtual; es decir no existe físicamente sino que se forma mediante la selección y/o filtrado de los componentes de otras tablas. Esto significa que pueden crearse dependencia entre las vistas; cuando una vista es definida en base a otra, se dice que es dependiente de ésta, por lo tanto, se suprimirá automáticamente la vista dependiente si se suprime la vista original. La eliminación de una tabla provoca también la eliminación automática de todas las vistas que se hayan definido haciendo referencia a ella.

Las vistas tienen la misma estructura que una tabla: filas y columnas. La única diferencia es que sólo se almacena de ellas la definición, no los datos. Los datos que se recuperan mediante una consulta a una vista se presentarán igual que los de una tabla. De hecho, si no se sabe qué se está trabajando con una vista, nada hace suponer que es así. Al igual que sucede con una tabla, se pueden insertar, actualizar, borrar y seleccionar datos en una vista. Aunque siempre es posible seleccionar datos de

una vista, en algunas condiciones existen restricciones para realizar el resto de las operaciones sobre ellas.

La forma en la que se acceda y consulte la información que se encuentra almacenada en la base de datos, a través del SGBD que se utilice, va a tener una gran importancia cuando haya pasado un tiempo y se encuentre que en la base de datos hay miles (o cientos de miles) de registros. Se necesitará que la herramienta proporcione vías de búsqueda y de consulta de la información, de una forma flexible y ágil. Lógicamente, se quiere que las vistas sean flexibles, y parametrizables tanto por los administradores del sistema como por el propio usuario, de forma que cada uno de los miembros del equipo de soporte de primer nivel.

Una vista es el resultado dinámico de una o varias operaciones relacionales realizadas sobre las relaciones base. Como ya se mencionó, una vista es una relación virtual que se produce cuando un usuario la consulta. Al usuario le parece que la vista es una relación que existe y la puede manipular como si se tratara de una relación base, pero la vista no está almacenada físicamente. El contenido de una vista está definido como una consulta sobre una o varias relaciones base. Cualquier operación que se realice sobre la vista se traduce automáticamente a operaciones sobre las relaciones de las que se deriva. Las vistas son dinámicas porque los cambios que se realizan sobre las tablas base que afectan a una vista, se reflejan inmediatamente sobre ellas. Cuando un usuario realiza un cambio sobre la vista (no todo tipo de cambios están permitidos), este cambio se realiza sobre las relaciones de las que se deriva.

Es importante que quede claro que, las vistas deben ser dinámicas, que no deben proporcionar una información estática, sino que deben actualizarse automáticamente (ya sea periódicamente o en base a eventos de forma “on-line”) de tal forma que se tenga la certeza de que el usuario ve la información actualizada.

Las vistas son una excelente forma de dar al usuario la información que necesita y sólo ésa. Son simples consultas a través de las cuales el usuario final únicamente ve determinadas columnas, filas o campos que cumplen un criterio. Se crea así un esquema conceptual a partir de lo que el usuario solicita, esto evitará tener información redundante. Se ha desarrollado bastante y mejorado el tipo de vistas, lo que supone la creación de vistas materializadas, vistas multinivel, vistas fragmentadas, subvistas, etc.

Las vistas materializadas fueron implementadas por primera vez, en el gestor de base de datos Oracle. Son vistas que presentan la particularidad de que almacenan la información que se consulta, para que cuando se quiera acceder nuevamente a esta información lo único que se necesite sea acceder a la consulta guardada en la memoria. Evidentemente, si una vista utiliza muchas tablas base, enlazadas de forma compleja, y dicha vista va a ser utilizada frecuentemente, será muy conveniente definirla como una vista materializada.

Las vistas materializadas también admiten índices, esta funcionalidad resulta muy útil a la hora de mejorar el rendimiento de las sentencias PL/SQL[11] o SQL que utilicen vistas materializadas. Cuando una sentencia SQL o PL/SQL accede a una vista materializada el servidor de la base de datos, en este caso en particular, Oracle, transforma la sentencia dirigiéndose directamente a los datos de la vista que están ya almacenados, en lugar de utilizar los datos de las diferentes tablas utilizadas en la definición de dicha vista.

Por otro lado, está el inconveniente de que la vista materializada va a tener que reutilizarse en el futuro, entonces se necesita de un mecanismo para actualizar o refrescar dicha vista materializada, ya que las tablas base de la vista pueden haber sufrido modificaciones desde la creación de la misma. Se plantea además, que a la hora de determinar si una vista debe definirse como vista o es mejor definirla como vista materializada, es muy importante valorar los costes de tener que ejecutar la sentencia SQL base de una vista normal siempre que se acceda a dicha vista, frente a los costes de almacenamiento y actualización de una vista materializada.

Una vista materializada utiliza una aproximación diferente: el resultado de la consulta se almacena en una tabla cache real, que será actualizada de forma periódica a partir de las tablas originales. Esto proporciona un acceso mucho más eficiente, a costa de un incremento en el tamaño de la base de datos y a una posible falta de sincronía, es decir, que los datos de la vista pueden estar potencialmente desfasados con respecto a los datos reales. Esta es una solución muy utilizada en entornos de

[11] Lenguaje de programación embebido en Oracle y PostgreSQL. El PL/SQL soporta todas las consultas y manipulación de datos que se usan en SQL, pero incluye nuevas características.

almacenes de datos (*datawarehousing*[12]), donde el acceso frecuente a las tablas básicas resulta demasiado costoso.

Ahora, para dar solución a este problema, que se puede ocasionar con la utilización de las vistas materializadas en la consistencia de los datos almacenados, surge la idea de aplicar la técnica del ”refresco” de la información. Consiste en refrescar, es decir actualizar la información que contiene la vista, con los posibles cambios que pudieran producirse sobre las tablas que relaciona. El tipo de refresco que debemos elegir dependerá de la frecuencia de actualización de las tablas base y de las necesidades que se tenga de disponer de datos exactos.

Por otro lado se encuentran las vistas multinivel, que son las que pueden presentar información de varios tipos o entidades relacionadas entre sí. Por ejemplo, una vista en la que se muestra el árbol de categorías de elementos de configuración para cada categoría, se muestran los elementos de configuración presentes en dicha categoría, para cada elemento, se muestran las llamadas de servicio que se han recibido relativas a ese elemento, para cada llamada los problemas relacionados y para cada problema, los cambios relacionados con ese problema.

Hasta este punto se ha evidenciado, que cuando se actualiza una relación base, el cambio se tiene que reflejar automáticamente en todas las vistas que la referencian. Del mismo modo, si se actualiza una vista, las relaciones base de las que se deriva, deberían reflejar el cambio. Las consultas sobre las vistas se tratan de igual modo que sobre las tablas.

De manera que se pueden identificar tres ventajas fundamentales de la utilización de las vistas:

- Perspectivas directas**:** Proporcionarse diversos modelos de información basados en los mismos datos, enfocándolos hacia distintos usuarios con necesidades específicas. El mostrar la información desde distintos ángulos ayuda a crear ambientes de trabajo y operación acordes a los objetivos de cualquier empresa. Debe evaluarse el perfil y requerimientos de información de los usuarios destino de la vista.

[12] En el contexto de la informática, un almacén de datos (del inglés *datawarehouse*) es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza.

- Transparencias en las modificaciones**:** El usuario final no se verá afectado por el diseño o alteraciones que se realicen en el esquema conceptual de la base de datos. Si el sistema requiere una modificación en su funcionamiento interno, podrán afectarse diversas estructuras que proveen el desempeño de éste; se pretende que los usuarios finales no adviertan tales alteraciones.
- Seguridad**:** Las vistas proporcionan de manera natural un medio para ocultar y proteger datos, dado que sólo se presenta al usuario una selección de lo atributos existentes.

2.3.2 Permisos de usuario

La seguridad debería conservarse por el mantenimiento de los usuarios de la bases de datos. Dependiendo del tamaño de un sistema de bases de datos y de la cantidad de trabajo requerido para administrar los usuarios de bases de datos, el administrador de seguridad debería ser el único usuario con los privilegios requeridos para crear, alterar o borrar un usuario de base de datos. Por otro lado, debería haber un número de administradores con privilegios para administrar los usuarios de bases de datos. Sólo personas que gozan de confianza deberían tener privilegios totales para administrar los usuarios de bases de datos.

El usuario de una base de datos es una entidad de seguridad de la base de datos. Cada usuario de una base de datos es miembro de la función pública. Se podría definir a los usuarios, como toda persona que tenga todo tipo de contacto con el sistema de base de datos desde que éste se diseña, se elabora, se termina y hasta que se pone en uso.

Una de las tareas más básicas que tienen los administradores de las bases de datos es identificar los usuarios. Como principio se toma en casi todos los sistemas, que cada usuario que se conecta a una base de datos, debe tener una cuenta. Cuando uno se conecta con una instancia de la base de datos, la cuenta de usuario debe se estar autenticada; esto quiere decir, que cuando cualquier usuario se conecta a una base de datos, ésta verifica que este usuario y la contraseña introducida estén almacenadas en la base de datos y, que además sean correctas. Las contraseñas se guardan encriptadas en la base de datos y cuando un usuario se conecta con la base de datos, se verifica globalmente, cuando la información pasa por una opción avanzada de seguridad, que confirma el nombre de usuario registrado en la base de datos junto con la contraseña que lo identifica, de manera tal que ese usuario se pueda diferenciar de cualquier otro que esté almacenado y que éste reciba la

información a la que tiene acceso y sólo ésta, garantizando así, la seguridad de los datos almacenados.

De manera general existe la tendencia de utilizar roles, que serán otorgados a los usuarios según sus funciones. Es decir se hace una asignación de nombre de usuarios para cada usuario en particular con su respectiva clave de acceso (*password*) y los perfiles asociados. Los perfiles son un conjunto de recursos agrupados bajo un mismo nombre que permiten al Administrador de la Base de Datos (DBA) gestionar de manera eficaz, rápida y racional recursos como; el consumo de recursos del sistema, el consumo de recursos de la base de datos y poner restricciones relativas a las contraseñas.

Queda claro que, una cuenta válida dentro de una base de datos es requerida para acceder a esta base de datos. Las cuentas de usuario son específicas a una base de datos. Todos los permisos y los objetos son controlados por la cuenta de usuario; quiere esto decir que asociado a cada usuario de la base de datos, existe una cuenta o esquema, como lo nombra Oracle, con el mismo nombre, y que es un conjunto lógico de objetos (tablas, vistas, índices, procedimientos, funciones, etc.). Por defecto cada usuario crea y tiene acceso a todos los objetos en su correspondiente cuenta o esquema, en el caso particular de Oracle, como se mencionó anteriormente.

Para que cada usuario pueda acceder a un objeto, se le deben conceder los privilegios apropiados. Los privilegios sirven para determinar qué tipo de acceso puede tener un usuario a los objetos de otro usuario, o bien, qué derecho tiene a realizar determinadas sentencias SQL. Los usuarios con los privilegios adecuados, pueden otorgar privilegios a otros usuarios a su criterio; lo que se conoce como seguridad discrecional.

A un usuario se le pueden otorgar una serie de privilegios. Un privilegio permite a un usuario acceder a ciertos objetos o realizar ciertas acciones:

- Privilegios sobre objetos (*object privileges*), son permisos sobre vistas, tablas, secuencias, procedimientos, paquetes.
- Privilegios del sistema (*system privileges*), son permisos sobre “niveles de la base de datos” como pueden ser conexión a la base de datos, creación de usuarios, limitar cuentas.
- Privilegios sobre roles (*role privileges*), son muchos permisos otorgados mediante roles agrupando un conjunto de privilegios.

Es importante destacar que los privilegios de autorización pueden ser leer, escribir, ejecutar, seleccionar, insertar, actualizar, referenciar o indexar; son los tipos de operaciones que se pueden ejecutar sobre los objetos del sistema.

A partir de aquí surge la necesidad de utilizar los permisos de usuarios. Con los permisos de usuarios, se trata de definir qué puede hacer un usuario concreto, o qué tipo de operaciones no le están permitidas. Los permisos, se suelen distinguir entre autorizar, no decir nada, o denegar explícitamente. A diferencia de la vida real, donde cada persona en particular, le puede dar las llaves de su casa a quien desee, los usuarios pueden no ser propietarios de sus objetos, y no tener permiso para delegar nada. Entonces se definen permisos sobre tablas, vistas, procedimientos, y casi todo lo que se le pueda ocurrir. La complejidad puede ser casi infinita, con un sinfín de jerarquías que se pueden contradecir, pues no sólo es, a qué se da permiso, sino también quién lo da, y qué ocurre si simultáneamente se otorga un permiso por un usuario y otro lo deniega explícitamente. Con idea de simplificar ésto se idearon los roles, concepto que se explicará más adelante en este trabajo.

En dependencia del negocio que se esté desarrollando, se pueden crear usuarios con diferentes privilegios o derechos, como pueden ser crear, borrar y modificar objetos, y que por demás tengan la posibilidad de conceder privilegios a otros usuarios sobre los objetos que ha creado. También se pueden crear usuarios con derecho a consultar, o actualizar, pero sin derecho a crear o borrar objetos. En fin en dependencia de las características propias del negocio es que se crean los usuarios y es que se le otorgan los privilegios, creando los diferentes tipos de usuarios dentro de una misma base de datos.

De manera general se ha podido constatar a partir de la investigación realizada que se pueden crear diferentes usuarios; como se ha mencionado en repetidas ocasiones con anterioridad, para cada aspecto de su aplicación con derechos muy limitados sobre los objetos de la base de datos. Tan solo deben otorgarse los privilegios estrictamente necesarios, y evitar que el mismo usuario pueda interactuar con la base de datos en diferentes casos de uso. Esto quiere decir que si un intruso gana acceso a su base de datos, usando las credenciales de sus aplicaciones, él solo puede efectuar tantos cambios como su aplicación se lo permita, garantizando de esta forma un nivel mayor de seguridad en la base de datos.

Después de estudiar varios ejemplos y, además, por la experiencia de trabajo de la autora, se han determinado algunos grupos de usuarios, que a partir de su comportamiento en el mundo real se pudieran enmarcar dentro de estas clasificaciones.

- Programadores de aplicaciones: Son profesionales en computación que interactúan con el sistema por medio de llamadas en el Lenguaje de Manipulación de Datos (DML), las cuales están incorporadas en un programa escrito en un lenguaje de programación (por ejemplo, C#, C, C++, Java, etc.)
- Usuarios sofisticados: Los usuarios sofisticados interactúan con el sistema sin escribir programas. En cambio escriben sus preguntas en un lenguaje de consultas de base de datos.
- Usuarios especializados: Algunos usuarios sofisticados escriben aplicaciones de base de datos especializadas que no encajan en el marco tradicional de procesamiento de datos.
- Usuarios ingenuos: Los usuarios no sofisticados que interactúan con el sistema invocando a uno de los programas de aplicación permanentes que se han escrito anteriormente en el sistema de base de datos. Se puede mencionar al usuario ingenuo como el usuario final que utiliza el sistema de base de datos sin saber nada del diseño interno.

Se puede concluir de manera parcial, que la seguridad de los datos incluye, los mecanismos que controlan el acceso y el uso de la base de datos en el nivel de objeto. La política de seguridad de los datos determina qué usuarios tienen acceso a objetos de un esquema específico, y los tipos específicos de acciones permitidos para cada usuario sobre el objeto. También se deberían definir las acciones para cualquiera, y para cada objeto de esquema.

Pudiendo destacar de manera particular que un elemento a tener en cuenta como mecanismo de seguridad, basándose en los privilegios de los usuarios, que es necesario tener en cuenta la sensibilidad de los datos; si la información no es sensible, entonces la política de seguridad de datos puede ser más flexible. Sin embargo, si los datos son sensibles, una política de seguridad debe ser desarrollada para mantener un fuerte control sobre el acceso a los objetos. Estos son elementos determinantes si se persigue lograr un buen diseño que garantice en alguna medida la seguridad.

Los administradores, por su parte, deben definir también una política para la seguridad de usuario final. Si una base de datos es grande y con muchos usuarios, el administrador de seguridad puede decidir qué grupos de usuarios pueden ser categorizados, crear roles de usuarios para esos grupos de usuarios, otorgar los privilegios necesarios o roles de aplicación para cada rol de usuario y, asignar los

roles de usuarios, a los usuarios. En excepciones, el administrador de seguridad, debe también decidir, qué privilegios deben ser explícitamente otorgados a usuarios individuales.

Por otro lado se plantea que, los administradores, quienes se encargan de administrar una base de datos con muchos usuarios, aplicaciones u objetos, deben tomar ventajas de los beneficios ofrecidos por los roles. Los roles simplifican en gran medida la tarea de la administración de privilegios, en entornos complejos.

Ahora bien, por último se pudiera concluir sobre este aspecto, que para los administradores de seguridad sería bueno considerar los temas relacionados a la administración de privilegios para todos los tipos de usuarios. Por ejemplo, en una base de datos con muchos usuarios, pudiera ser beneficioso utilizar roles; tema que se abordará más adelante, para administrar los privilegios disponibles a usuarios. Por otro lado, sería bueno agregar en este punto, que a partir de la experiencia personal de la autora, en una base de datos con un número de usuarios reducido, pudiera ser más fácil otorgar privilegios explícitamente a los usuarios y anular el uso de roles; elemento que se pudiera tener en cuenta en la propuesta del procedimiento, pues es un aspecto que brinda flexibilidad, ya que de esta manera, se estaría ofreciendo la posibilidad de adaptación a diferentes entornos y la aplicabilidad de igual manera.

2.3.3 Roles

Los roles son un grupo de privilegios que se conceden a los usuarios o a otro rol. Estos no son propiedad de nadie, ni se enmarcan; por ejemplo en el caso de Oracle, en un esquema determinado. Oracle define un esquema como un objeto que contiene tablas, índices, vistas, procedimientos almacenados, etc. Una regla importante que se plantea sobre la declaración de los roles, es que se puede dar acceso a cualquier usuario a un rol, excepto a uno mismo; es decir ésto se ve de manera reflexiva. Pueden ser activados y desactivados por usuarios autorizados. Un rol puede además determinar el acceso de un usuario a un objeto, pero no puede permitir la creación de objetos.

Se puede decir entonces, que los roles son simplemente, un conjunto de permisos que se unen para mayor comodidad. Los sistemas suelen traer algunos predefinidos, “administrador” o “usuario” son dos de ellos, pero realmente los roles están destinados a ser personalizados. Dentro de un rol se puede incluir, por ejemplo, el acceso a tres vistas, la ejecución de seis procedimientos concretos, y la escritura en una tabla. De esta forma se acota el alcance de un usuario concreto, y por tanto el daño

que puede llegar a hacer. Existen roles específicos para la función de asignación de roles, siendo éstos los que deben controlarse de forma más estricta.

Queda conceptualizado que los roles son, agrupaciones de privilegios que reciben un nombre; se conceden y revocan, de igual forma que los privilegios; pueden tener contraseña; se pueden activar y desactivar dinámicamente en una sesión de consulta. Además, son un conjunto de privilegios, que pueden concederse de golpe a un usuario sobre objetos de inserción o creación, de borrado o eliminación, de modificación o actualización, de consulta, de ejecución o elementos propiamente del sistema, como son la creación de tablas, vistas, etc. Es decir, un rol, es el nombre de un grupo de privilegios asociados que pueden ser otorgados a los usuarios, y este método hace más fácil administrar los privilegios.

A partir de la experiencia personal de la autora, se determina que un usuario puede tener acceso a varios roles, y varios usuarios pueden ser asignados al mismo rol. Esto es un elemento a tener en cuenta para la propuesta del procedimiento. Además sería bueno resaltar, que más que dejar que las aplicaciones de base de datos sean quienes creen los roles de forma automática, sean los diseñadores, quienes conozcan el negocio y sean capaces de prever posibles huecos de seguridad, los que establecieran los roles de ante mano, quedando éstos predefinidos en el momento de la implementación; asegurando de esta forma la seguridad de este aspecto específico, ya que los roles dan una base, para la especificación de requisitos de protección que pueden ser necesarios en situaciones reales.

En síntesis un rol no es más que un grupo al cual los usuarios individuales pueden ser adicionados, para que los permisos puedan ser aplicados a un grupo en lugar de aplicar los permisos a todos los usuarios en forma individual. Cada base de datos tiene un conjunto definido de roles de base de datos, al cual los usuarios de la base de datos pueden ser adicionados. Estos roles de base de datos son únicos dentro de la base de datos; mientras los permisos de los roles de base de datos no pueden ser alterados, nuevos roles de base de datos pueden ser creados.

Al realizar un análisis de todos los elementos que han sido abordados con anterioridad, se relacionan a continuación algunas propiedades de los roles; que constituyen ventajas en sí, pues facilitan la administración de los privilegios en una base de datos:

- Reducida asignación de privilegios: En lugar de otorgar explícitamente el mismo conjunto de privilegios a muchos usuarios, el administrador de la base de datos puede asignar los privilegios a un rol y éste a un grupo de usuarios.
- Administración dinámica de los privilegios: Cuando los privilegios de un grupo deben cambiar, sólo los privilegios del rol necesitan ser modificados. Los dominios de seguridad de todos los usuarios a los que asignó dicho rol, reflejarán automáticamente los cambios hechos en el rol.
- Selectiva disponibilidad de los privilegios: Los roles asignados a los usuarios pueden ser selectivamente activados o desactivados. Esto permite el control específico de los privilegios de los usuarios en cualquier situación.
- Consciencia de aplicación: Una aplicación de la base de datos puede ser diseñada para habilitar o inhabilitar roles automáticamente cuando un usuario intenta usar la aplicación.

Es necesario dejar claro que son los roles, éstos son un grupo de privilegios que pueden ser concedidos a un grupo de usuarios o a otro rol, es decir son sencillamente un conjunto de permisos que se unen para lograr una mayor comodidad. Con la utilización de los roles, se ha reflejado que se acota el alcance de un usuario concreto, y por tanto el daño que puede llegar a hacer con la información que se almacena en la base de datos. Sin más ni menos, un rol es el nombre de un grupo de privilegios asociados que pueden ser otorgados a los usuarios haciendo más fácil la administración de privilegios. Como aspecto positivo, vale destacar que, un usuario puede tener acceso a varios roles, y varios usuarios pueden ser asignados al mismo rol, reflejando con esta norma la realidad en la que se trabaja. Finalmente se establece que los roles en una base de datos son únicos dentro de la misma.

Sería bueno antes de concluir con este tema, hacer algunas recomendaciones sobre el trabajo con los roles dentro de una base de datos. Partiendo de los elementos que se han aportado en la investigación realizada, es importante definir los privilegios que se quieren dar, y estos privilegios otorgarlos a un rol específico, es decir crear varios roles; que serían los que contendrían privilegios determinados; no un gran número de privilegios, sólo los que represente una acción específica a realizar sobre un objeto de la base de datos; que estará determinado por el negocio en que se desarrolle, logrando de esta forma que cada rol represente un privilegio específico, de manera que después se creen otros roles que tengan dentro de ellos a estos pequeños roles que fueron creados con privilegios específicos, y finalmente sean estos roles los que se les asignen a los usuarios. Esto crea una mayor independencia entre los privilegios asignados a los usuarios.

Todo lo que se ha propuesto, crea un nivel de acceso superior, pues se tendría un acceso a los objetos de la base de datos, más seguro, ya que para un atacante se volvería más complicado acceder a los elementos que se encuentran almacenados. Todo esto, constituye un aspecto de vital importancia que se recomienda tener presente para conformar el procedimiento y llegar a la solución.

2.4 Trazabilidad

Una de las grandes preocupaciones en los contextos de redes y bases de datos son los ingresos y alteraciones no deseadas de la información. Algunas personas se plantean la idea, pregunta, inquietud, de cómo hacer para monitorear, auditar las bases de datos y su contenido. Cómo saber qué hacen los usuarios, qué cambios realizan, qué borran, qué agregan, y bueno todo lo concerniente al monitoreo de la base de datos; partiendo del hecho, que los desarrolladores de las mismas no tienen en cuenta la implementación de esos aspectos. La acción que se acomete para dar solución a la problemática que se describió, es la trazabilidad.

El término trazabilidad viene del inglés "*trace ability*", es la habilidad de rastrear. Establecer qué significa la trazabilidad, implica revisar definiciones y reflexiones que previamente se han establecido alrededor de la auditoría de las Tecnologías de Información (TI). Si bien, el registro de las operaciones electrónicas, es un factor fundamental en los procesos de las organizaciones, la reconstrucción de eventos obedece tanto a la formalidad de los registros de auditoría, como a las características técnicas y administrativas que las base de datos deben incorporar, si quieren contar con estrategias y escenarios para rastrear situaciones particulares que se puedan producir. En este ámbito se define reconstrucción de los eventos, como la posibilidad de reencontrar los datos, los antecedentes y la locación de una acción determinada, mediante la identificación de registros almacenados.

Es decir, se define trazabilidad como la capacidad que tiene una organización o sistema para rastrear, reconstruir o establecer relaciones entre objetos monitoreados, para identificar y analizar situaciones específicas o generales en los mismos. [23] Para aclarar esta definición, se procede a profundizar en las palabras señaladas, brindándole un sentido práctico de aplicación que será revisado más adelante en la presente investigación:

- Capacidad: Esta palabra sugiere la definición de acciones y estrategias específicas por parte de la organización o sistema, que permita, bajo directrices establecidas desarrollar una actividad específica.

- Rastrear, reconstruir o establecer: Este conjunto de verbos, hacen referencia a la esencia misma del concepto que especifican. Son las acciones que se buscan efectuar cuando de trazabilidad se habla.
- Objetos monitoreados: La trazabilidad sin una adecuada definición de pistas de auditoría no logra alcanzar sus objetivos (rastrear, reconstruir o establecer). El monitoreo es un prerrequisito para darle sentido a la trazabilidad como capacidad en un sistema.

Es importante aclarar que es posible no tener monitoreo definido formalmente, pero sí, registros propios de los sistemas o procesos que pueden ser útiles para rastrear, reconstruir o establecer relaciones. Así mismo, es fundamental la capacidad de relacionar esa información de auditoría, entre los diversos niveles y componentes de una infraestructura informática.

Entonces, si se logra mantener una traza sobre la información que se tiene almacenada en la base de datos, se podrán conocer diferentes aspectos como:

1. El usuario que realizó una determinada operación.
2. El objeto o los objetos a los que accedió un usuario determinado.
3. Fecha y hora en la que ocurrieron las acciones que se decidan monitorear.
4. Código de la acción.

Es importante dejar claro, que en estos casos no se refleja la información que se modificó, sólo datos referentes a ella, de manera que se puedan registrar todas las acciones realizadas sobre la misma.

Hasta este punto, es importante que quede claro, todo lo que se gana cuando mantenemos un monitoreo de las transacciones que se realizan en la base de datos y además quedan las trazas de las operaciones realizadas. Es inminente que a partir de aquí, se permite medir, asegurar, demostrar, monitorear y registrar los accesos a la información almacenada en la base de datos; controlar y registrar las acciones de determinados usuarios de la base de datos. La auditoría es además, la facilidad al administrador de la base de datos, de vigilar el uso y los privilegios de la base de datos. Se pudiera llevar el registro histórico de todos los cambios (inserciones, borrados o actualizaciones) de la base de datos junto con información sobre el usuario que realizó el cambio y en qué momento. Los SGBD proporcionan mecanismos para crear trazas de auditoría, pero esto varía de un SGBD a otro.

El hecho que muchas aplicaciones de bases de datos seguras requieren que se mantenga una traza de auditoría se hace incuestionable después de todos los elementos expuestos. Una traza de

auditoría, es un registro histórico de todos los cambios, como se advirtió con anterioridad; inserciones, borrados o actualizaciones de la base de datos, junto con la información que permita identificar al usuario que realizó el cambio y en qué momento. Obviamente, la traza de auditoría, ayuda grandemente a mantener la seguridad de la información almacenada en la base de datos.

Ahora bien, cada sistema de bases de datos realiza la traza de auditoria de forma diferente; en caso de que la tengan. Después de la revisión realizada, la autora, ha podido identificar algunas tendencias. Primero, cada diseñador pudiera implementar una estrategia particular, que cubra de acuerdo a las necesidades de su negocio todos los elementos de trazabilidad. Por otro lado, se pudiera apoyar en las funcionalidades que al respecto brindan algunas herramientas de control de versiones, como el *Subversion* (SVN[13]). Siempre, cuando se diseña un software, es muy importante mantener trazas del sistema, tanto para ver que pasó en cada momento, como para poder decir, que uno u otro paso se ejecuto mal; pero para poder decirlo con elementos contundentes, son necesarias las trazas.

La manera de persistir las trazas, siempre, cada desarrollador la define particularmente basado en las necesidades de su negocio y a partir de su experiencia personal. Se parte de la primera de las tendencias mencionadas; que persigue modelar el flujo de acciones que se realizan en una aplicación sobre la información que se almacena en la base de datos. De esta forma se plantea que las aplicaciones pudieran modelar un flujo, es decir como generalmente en toda aplicación todas las operaciones que se realizan pasan por varios flujos de acciones, pues entonces sería conveniente modelar una tabla en la que persistan los flujos por los que pasa la aplicación. Dicha tabla se nombraría de la forma que los diseñadores determinen. La tabla contendría cada modificación que se realice sobre el flujo de la aplicación. Contendría como llave el identificador de la acción específica que se está realizando, es decir el flujo determinado que se inició y, además, los diferentes cambios que se fueron produciendo sobre esa acción, a lo largo de todo el flujo; incorporando el usuario que trabajó

[13] Sistema de control de versiones de código abierto (free/open-sourse). Es un software de sistema de control de versiones. Una característica importante de Subversion es que, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

sobre esta acción, la fecha en que se realizó, pudiéndose incluir además la hora en que se inició la transacción como tal o de cada modificación en particular.

Lo que se persigue al aplicar esta estrategia, es mantener un control de todas las acciones que se realizan en las aplicaciones, monitoreando en cada momento del flujo de la aplicación las acciones realizadas sobre los objetos de la base de datos, de manera tal que en cualquier momento que se necesite conocer quién realizó un cambio sobre algún objeto de la base de datos, o en qué momento se produjo un error determinado, tener toda la información sobre lo ocurrido; lo que satisface todos los aspectos que se persiguen con la trazabilidad.

Esta solución es factible como se pudo apreciar, pues satisface todo lo que persigue el monitoreo y auditoría de las bases de datos. Ahora, es indiscutible que establecer todo este proceso resulta bastante engorroso para los diseñadores y los desarrolladores, ya que se tendría que establecer que todo este control se active a partir de que comience cualquier proceso desde la aplicación, o de otra manera a partir de *trigger*, que activen todo este proceso de creación de trazas sobre las tablas que se determinen en el diseño, supuestamente las más significativas para el negocio, de las que sería importante tener un historial. Debiéndose tener en cuenta la limpieza de esta tabla, por cuestiones de rendimiento de las operaciones realizadas en la base de datos; elemento que se explicara más adelante en este documento.

Los *trigger* o disparadores, en una base de datos son una rutina asociada con una tabla o vista que automáticamente realiza una acción cuando una fila en la tabla o la vista se inserta (INSERT), se actualiza (UPDATE), o borra (DELETE), permiten vigilar y registrar acciones especificas según las condiciones propias del negocio; admitiendo la creación de ficheros de auditoría a la medida. Son usados para mejorar la administración de la base de datos, sin necesidad de contar con que el usuario sea quien ejecute la sentencia SQL.

Por otro lado, se prosigue analizando la segunda opción mecionada; donde se aprovecharían las funcionalidades que brindan los sistemas controladores de versiones, particularmente en este caso el SVN. El *Subversion* lo que hace es que modela, es decir brinda una tabla que contiene un identificador (ID), un nombre para la tabla (NOMBRETABLA), el campo de la tabla que se está registrando (CAMPO), un valor anterior (VALORANTERIOR); que representa si anteriormente se realizó en algún momento alguna acción sobre este campo, un valor nuevo (VALORNUEVO); donde se escribe el nuevo valor que toma ese campo, la acción (ACCION); que puede ser identificada numéricamente, por

ejemplo, cuando se inserta se asume que el valor será 1, si se actualiza será 2 y si se elimina entonces será 3, ésta es solo una manera en que se pudiera representar. Además, el SVN nos permite determinar la fecha en que se realizó la acción en cuestión (FECHA) y finalmente el usuario de la base de datos que la realizó (USUARIO).

¿Como funciona? Pues bien, él a cada tabla de la base de datos se le crea un *trigger*, que tiene como función insertar en la tabla que se describió con anterioridad cada vez que se realice alguna acción sobre la tabla desde donde él se dispara. Es decir por cada campo de la tabla, se insertan los valores anteriores y los nuevos, que se van tomando. Con ésto se tiene una traza completa de qué pasa con todos los campos de una tupla, con todas las tuplas de una tabla y finalmente con todas las tablas de la base de datos.

Esta última opción está indudablemente muy bien pensada, lo único que tiene que hacerse bien es la creación de los índices, pues la tabla que almacena todos estos datos, crecerá mucho, y ésto enlentecerá las consultas dentro de la base de datos. El hecho de que crezca mucho, afectaría a nivel de motor de base de datos y no a nivel de dispositivo de almacenamiento; pues realmente se ha estimado un cálculo, sobre qué capacidad en disco ocuparían alrededor de 260 000 tuplas; conteniendo datos simples y no imágenes, un total de 32 MB de memoria. Esto claramente no representa nada para los dispositivos de almacenamiento, pero si para el motor de la base de datos pues las consultas se pondrían muy lentas, como se mencionó con anterioridad, dada la gran cantidad de información almacenada en la base de datos. Para resolver esta problemática se propone entonces, garantizar la limpieza de esta tabla cada cierto tiempo; lo que establecería cada diseñador, dependiendo de su negocio, pero creando además, un historial en el que se haga una copia de toda la información que está contenida en la tabla que registra las trazas, de manera tal que no se pierdan todos los cambios realizados y que en el momento requerido se puedan hacer consultas de todos los sucesos ocurridos, desde la aplicación y que puedan afectar la base de datos, permitiendo así tomar decisiones.

Los métodos descritos son sólo dos ejemplos de qué se puede hacer, porque realmente el diseñador determina si necesita utilizar alguno que ya ha sido probado en otra empresa o si aplica alguno que personalmente haya probado con anterioridad, si no establece uno nuevo, a partir de su experiencia. En cualquier caso, lo importante es que se garantice la sincronización, el control, la integridad de archivos y la confiabilidad en la generación de los registros de auditoría, que son elementos requeridos

para darle vida a la trazabilidad o a la capacidad para rastrear, reconstruir o establecer relaciones entre los objetos monitoreados de un sistema. Realmente con la trazabilidad se hace referencia a un concepto sistémico, en la necesidad de establecer relaciones y observar el todo.

El constante avance de la tecnología y los procedimientos internos de las organizaciones, sugieren elementos nuevos que deben ser revisados para enriquecer la temática de la trazabilidad. Tema de numerosas investigaciones y en el que se seguirá innovando, gracias a la alta prioridad que tiene dentro del funcionamiento de las empresas.

2.5 Salvaguarda y recuperación

Debido a que la información almacenada sobre los medios de cómputo, está sujeta a la pérdida o la corrupción ocasionada por la amplia gama de procesos, es importante proporcionar medios para restaurar los datos adecuadamente, al mismo estado en que se encontraban en el momento del fallo de sistema, pero los procedimientos de recuperación de la base de datos, pueden restablecerla al estado en que se encontraba, pero tiempo antes del fallo e identificar el estado del procesamiento de la transacción en el momento del fallo. Con esta capacidad, las transacciones no procesadas podrán ejecutarse contra la base de datos restablecida para regresar a un estado completo.

Las técnicas de recuperación son otra función esencial del administrador de la base de datos. A pesar de que el SGBD lleva a cabo una parte del proceso de recuperación, los usuarios determinan en forma crítica la operatividad de esos sistemas de protección. El administrador de la base de datos debe anticipar fallas y definir procedimientos estándares de operación; los usuarios deben saber qué hacer, cuando el sistema está caído y qué es lo primero que debe realizarse, cuando el sistema sea puesto en marcha nuevamente. El personal que trabaja directamente con la base de datos, deberá saber como iniciar el proceso de recuperación de la base de datos, qué copias de seguridad utilizar; cómo programar la reejecución del tiempo perdido y las tareas pendientes.

Se han podido identificar las siguientes fuentes de fallo:

1. **Errores el sistema**. El sistema ha entrado en un estado indeseable, como el interbloqueo, que evita al programa continuar con el procesamiento normal. Este tipo de fallo puede o no provocar corrupción en los archivos de datos.
2. **Fallos de hardware**. Dos de los tipos más comunes de fallos de hardware son el fallo del disco y la pérdida de la capacidad de transmisión en un enlace de transmisión. En el primer caso, la

causa está normalmente, en que la cabeza de lectura/escritura del disco entra en contacto físico con la superficie del disco.

3. **Errores lógicos**. Las condiciones más comunes que pudieran impedir que un programa continúe con su ejecución normal son los datos erróneos o inexistentes.

Seguidamente en esta investigación se describen algunos procedimientos de recuperación, que son apropiados para diferentes tipos de fallos.

Antes es importante definir, que la integridad de los datos exige que una transacción esté en uno de los dos estados siguientes:

- **Abortada**. Una transacción puede no siempre completar su proceso exitosamente. Para asegurar que la transacción incompleta no afecte el estado consistente de la base de datos, dichas transacciones tienen que abortarse, restaurando la base de datos al estado en que estaba antes de que la transacción en cuestión comenzara su ejecución. Tal restauración se logra mediante la operación de *rollback*.
- **Cerrada**. Una transacción que completa exitosamente su procesamiento se dice que está cerrada. Una transacción cerrada deja la base de datos en un estado consistente.

Aclarados estos elementos, ya se puede comenzar a comentar algunos de los mecanismos de recuperación.

El diario de operaciones *log,* juega un papel clave en la recuperación de los fallos. El diario de operaciones, es un historial de todos los cambios hechos a la base de datos, así como también del estado de cada transacción. Obviamente, es importante que los datos del diario de operaciones no se pierdan ni se destruyan. Por consiguiente, la información del diario de operaciones se guarda en un almacenamiento estable; al menos lo más estable posible, para que sobreviva a todos los fallos. En la práctica, ésto se logra manteniendo varias copias en disco.

Una estrategia de recuperación puede desarrollarse utilizando uno de los dos enfoques siguientes, registrar las actualizaciones diferidas o registrar las actualizaciones inmediatas y los puntos de chequeo proporcionan eficiencia adicional; elemento que se abordará más adelante, en esta investigación.

Las actualizaciones diferidas proponen el registro en el diario de operaciones del momento en que comienza la transacción. Durante la ejecución de la transacción, se almacenan todas las

modificaciones que se hagan sobre el objeto que se trabaja. Finalmente si la transacción se ejecutó exitosamente se cierra parcialmente y se escribe en el registro diario de operaciones.

Al utilizar el diario de operaciones, el SGBD puede gestionar cualquier fallo que no provoque la pérdida de la información del propio diario de operaciones. La prevención de la pérdida del diario de operaciones se resuelve mediante su duplicación en más de un disco. Como la probabilidad correspondiente a la pérdida del diario es muy pequeña, este método se considera comúnmente como un método de almacenamiento estable.

De esta manera la base de datos puede haberse corrompido, pero el procesamiento de la transacción se completa y los nuevos valores para los elementos de los datos correspondientes están contenidos en el diario de operaciones. Esto excluye la necesidad de reprocesar la transacción.

Por otro lado, el diario de operaciones con actualizaciones inmediatas. Un método alternativo que utiliza un diario de operaciones para la recuperación, consiste en hacer todas las actualizaciones de la base de datos inmediatamente y llevar un registro de todos los cambios en el diario. Al igual que en el método anterior, si ocurre un fallo, el diario se utiliza para restablecer el estado de la base de datos a un estado previo consistente. Similarmente, cuando comienza una transacción, en el diario de operaciones se escribe el registro. Durante la ejecución de la transacción cualquier operación que se haga está precedida por la escritura de un nuevo registro en el diario de operaciones (recuérdese que en la estrategia previa, no se aplicaba ninguna operación de escritura a la base de datos, hasta que la transacción cerrara parcialmente). Cada registro diario se escribe de la misma forma que en la anterior y, cuando la transacción se cierra parcialmente, se escribe en el diario de operaciones.

En el caso de los puntos de comprobación (*cheekpointing*) de los procedimientos precedentes, se pudiera concluir que, la recuperación solamente requiere analizar el diario de operaciones para las entradas correspondientes a la transacción más reciente o a algunas transacciones recientes. En principio, no debe haber límite con relación a cuanto debe mirar el sistema hacia atrás en el diario, ya que los errores pueden haber comenzado con la primera transacción. Esto puede provocar un gran consumo y despilfarro de tiempo. Una mejor manera consiste en encontrar hacia atrás un punto suficientemente alejado como para asegurar que cualquier elemento antes de ese punto se haya escrito correctamente y se haya almacenado sin riesgo. Este método se denomina el de puntos de control.

Oracle por su parte, propone también una técnica para realizar las copias de seguridad y recuperación. El es modo de trabajo ARCHIVELOG. Este modo, permite guardar una copia de los ficheros *redolog* utilizados (antes de ser reutilizados). Realizar un proceso cíclico del fichero *redolog*, desde el último al primero; en caso de pérdida podemos recuperar hasta la última transacción. Este modo tiene como ventajas, que permite realizar una recuperación completa y en un determinado momento del tiempo. Está especialmente indicado para aplicaciones críticas, donde no se pueda tolerar una mínima pérdida de información. Se requiere un espacio adicional en disco. Una cosa si posee, que el administrador tiene trabajo adicional (gestión de espacio y seguimiento de los ficheros de log).

Para realizar las copias de seguridad, en caso de error o pérdida por error físico permiten recuperar la información. Si se está en modo ARCHIVELOG podremos recuperar hasta la última transacción efectuada si no hasta la última copia de seguridad.

Para realizar la recuperación, se depende en gran medida del tipo de copia de seguridad que se haya hecho. Será de gran importancia, si se ha realizado en modo ARCHIVELOG o no. Existe la recuperación incompleta, que con ella podemos recuperar pérdidas de ficheros *redolog*, de control o de datos. Por otro lado la recuperación manual, que se realiza hasta la hora en la que un fichero *redolog* se estropeó; se utiliza sobre todo para pérdidas de ficheros *redolog* o de control. Por último la recuperación en un punto del tiempo, donde se recupera hasta una fecha y hora determinada. Se puede utilizar para ficheros de *redolog* o de datos y existen dos tipos, la automática y la manual.

Hasta aquí se puede concluir, después de ver diferentes métodos que se utilizan para realizar la salvaguarda de la información que se almacena en una base de datos. Son varias las estrategias que se pudieran trazar para realizar esta operación, todo está en dependencia de las necesidades en particular de cada negocio, lo que si es importante lograr garantizar es, que no se pierda la información almacenada, que en todo momento exista una copia de la información con la que se trabaja, para que ante cualquier fallo, exista respuesta rápida y confiable.

Toda gestión de una plataforma de software, necesita la gestión de la continuidad y para ello se planifican las tareas que garanticen continuar la operación en momentos de desastre. Los elementos de software en los servidores que pueden ser objeto de daño o avería y generar incidentes, son los que hay que tener en cuenta a la hora de planificar el proceso que garantizará la salva y recuperación de la información. En dependencia del software que se dañe y en qué servidor, se deberán trazar procedimientos diferentes para solucionarlo. Quizá su última copia de seguridad válida sea de hace

unos días, quizás, también puede ser que tengas un sistema un poco más complejo de copias, pero en cualquier caso, estás en un problema y la forma de solucionarlo es poner en practica cualquiera de las opciones que se mencionaron o alguna, que como diseñador, seas capaz de idear.

Antes de concluir con este tema se quiere dejar claro que, dependiendo de la cantidad de datos que se manejen, se podrá establecer el tipo de respaldo, pues hacer un respaldo consume recursos del servidor de base de datos, por lo que no se debe hacer cuando se tienen todos los usuarios conectados, pero sí cuando todos se van a su casa o a la hora del almuerzo, ésto se puede valorar. Ahora bien, es riesgoso hacer un solo respaldo diario, pues qué pasaría si se acostumbra a sacar respaldos al final de cada día. Por ejemplo, si se saca el respaldo el lunes en la noche y el martes a las 15:00 horas ocurre algún error y se daña la base de datos, correspondería entonces restaurar el respaldo del lunes en la noche, pero todo lo hecho el martes hasta las 15:00 horas se va a perder. Para que ésto no ocurra, el respaldo sacado al final de cada jornada debe ser un tipo de respaldo total, un respaldo general. Esta propuesta será abordada de forma amplia más adelante en esta investigación, pues se dará una propuesta, de solución con el procedimiento.

3.6 Propuesta de procedimiento

En este trabajo de diploma, se propone un procedimiento que ha sido creada con el objetivo de garantizar, en cierta medida, la seguridad de una base de datos; sea cual sea el negocio en el que se trabaje. Partiendo de los diferentes elementos que se han analizado, que se deben tener en cuenta en la etapa de diseño, se han establecido pasos a seguir, alcanzando flexibilidad en cada uno de los aspectos que propone el procedimiento, logrando así que la aplicabilidad no sea solamente en el negocio de Registro y Notarias; que es en el negocio que se validará y a partir del cual surgió la idea, sino en cualquier base de datos en la que sea necesario asegurar la información que se almacena.

El procedimiento ofrece una serie de pasos, a través de los cuales se propone ir pasando a lo largo de la etapa de diseño. Cada uno de estos pasos gira alrededor de los cuatro elementos que fueron abordados en este capítulo, y que garantizan la seguridad de la base de datos. Se nombran estos pasos como fases; en las que se irán desarrollando todos los pasos a tener en cuenta para garantizar la integridad, la seguridad, establecer la trazabilidad y determinar todo el proceso de salvaguarda y recuperación. Cada una de ellas irá dando paso a la siguiente. Es preciso puntualizar, que se recomienda, no violar ninguno de los pasos que se establecen, quizás pasándolos por alto porque representen cosas muy obvias para el que las aplique o comunes por su experiencia, ya que cada uno

ellos está elaborado estratégicamente para lograr el resultado que se persigue: la seguridad de la base de datos.

La primera fase está relacionada con la integridad: donde se establecen los pasos que se deben seguir para garantizar lo que a seguridad en base de datos se refiere. Después, la fase siguiente, es la que aborda los aspectos relacionados con la seguridad; es decir, el trabajo con los roles, usuarios y vistas; proponiéndose los pasos a seguir para asegurar la información almacenada a partir de estos elementos. Posteriormente se establece una estrategia para la trazabilidad; ésta sería la tercera fase, que cuenta con una propuesta flexible y configurable. Finalmente, se propone la fase que se refiere a todo lo relacionado con la salvaguarda y recuperación de la información, planteándose los pasos que aseguran la disponibilidad.

Hasta aquí, se han brindado los elementos con los que contará el procedimiento. Se determinó que para poder poner en práctica el procedimiento que se propone en esta investigación, deben quedar garantizadas ciertas cuestiones asociadas al diseño de base de datos y, algunos requisitos, que son de vital importancia tener en cuenta para la seguridad física de la base de datos, quedando establecidas como precondiciones.

Se señala de esta forma, porque se conoce que el diseño de una base de datos puede hacerlo prácticamente cualquiera, simplemente se requiere conocer la herramienta que se va a utilizar, y ya se puede proceder. No obstante, cuando se diseña una base de datos, por lo general posteriormente se encuentran diversos problemas relacionados con cosas que en el inicio no se pensaron o que quizás, debieron haberse hecho, de manera diferente.

Con el tiempo, una persona dedicada a realizar este tipo de tareas, irá ganando experiencia que le permitirá anticipar los futuros problemas que se le puedan presentar y logrará diseñar bases de datos mejores y más óptimas. No obstante, es posible diseñar una base de datos bien hecha desde el principio, gracias a diversas reglas y recomendaciones que se van aprendiendo. Se trata de los fundamentos para el diseño de una base de datos relacional; los que establecen las reglas para diseñar, que garantizan un mínimo de problemas futuros aún sin tener experiencia.

Siguiendo estas sencillas reglas, el diseño mejorará, aunque es indudable que la experiencia y la práctica siguen teniendo un lugar preponderante en ésta y en la mayoría de las tareas de la administración de sistemas. Sin embargo, son un buen punto de partida para iniciarse en la aventura de realizar un buen diseño de una base de datos, por lo que es muy provechoso estudiarlos y

aprenderlos, pues pueden en el futuro, ahorrar muchos dolores de cabeza. Teniendo asegurado un buen diseño, correspondería asegurar la seguridad física de la base de datos, por lo que se definen los requisitos que son de vital importancia tener en cuenta. Estos serán abordados más adelante en esta investigación, como precondición establecida para aplicar el procedimiento.

Después de tener claros todos los elementos explicados y ya puntualizados previamente, las condiciones que deben quedar establecidas antes de aplicar este procedimiento, se hace finalmente el planteamiento de la propuesta.

3.6.1 Procedimiento

Se parte del estudio exhaustivo del negocio que se persigue modelar, de manera tal que se puedan identificar todos los aspectos del mundo real que en él se relacionan. A partir de aquí, ya se debe tener una idea bastante clara de la manera en que será persistida la información en la base de datos. Se comenzará realizando el diseño y antes de empezar a aplicar el procedimiento, se establece la necesidad de tener garantizadas las condiciones que se mencionan a continuación.

Precondiciones

Paso 1

Deben estar garantizados los fundamentos para el diseño de una base de datos relacional; los que establecen las reglas para diseñar de forma tal que se garantice un mínimo de problemas.

- El modelo de bases de datos relacionales, comprende conceptos basados en la lógica así como en la teoría y básicamente, nos indica la manera correcta de hacer las cosas.
- Una base de datos creada de acuerdo a las reglas del modelo relacional, tiende a ser más eficiente, predecible y bien formada. Además, debido a que su estructura corresponde con algo ya esperado, requiere un menor esfuerzo para entenderla, y por ende, se facilitará mucho su mantenimiento.
- Cada una de las tablas, deberá tener una llave primaria, la cual identifique de manera única a cada renglón, es decir, a cada registro de la tabla.
- Las claves extranjeras o llaves foráneas, son columnas o campos que hacen referencia a una llave primaria desde otra tabla.

- Es posible definir tres tipos de relaciones entre las tablas de una base de datos relacional: relación uno a uno, uno a muchos y muchos a muchos. Cabe mencionar que la relación muchos a muchos, no es directa ya que requiere de una tabla intermedia, que realice el enlace.
- El proceso de normalización consiste en la simplificación del diseño de la base de datos, de tal manera que se optimice la estructura lo más posible.
- Una base de datos bien formada, cumple con las reglas de normalización.
- Mantener la integridad de los datos a lo largo del tiempo.
- Las reglas de integridad de las entidades, prohíben la existencia de valores nulos dentro de las columnas declaradas como campo llave.
- Las reglas de integridad referencial indican que, la base de datos, no debe tener ni una sola llave foránea inexistente o perdida.
- Las reglas de negocios, son una parte importante de la integridad de una base de datos.
- Un buen diseño de bases de datos requiere de nociones de negocios, dedicarle tiempo y experiencia.
- En raras ocasiones es posible que surja la necesidad de romper alguna regla de normalización, con el objetivo de aumentar el desempeño.

Paso 2

Deben estar garantizados los requisitos que son de vital importancia para la seguridad física de la base de datos.

- La base de datos debe ser protegida contra el fuego, el robo y otras formas de destrucción.
- Los datos deben ser reconstruibles, porque por muchas precauciones que se tomen, siempre ocurren accidentes.
- Los datos deben poder ser sometidos a procesos de auditoría. La falta de auditoría en los SI ha permitido que se cometan grandes delitos.
- El sistema debe diseñarse a prueba de intromisiones. Los programadores, por ingeniosos que sean, no deben poder pasar por alto los controles.
- Ningún sistema puede evitar, de manera absoluta, las intromisiones malintencionadas, pero es posible hacer que resulte muy difícil eludir los controles. El sistema debe tener capacidad para verificar que sus acciones han sido autorizadas. Las acciones de los usuarios deben ser supervisadas, de modo tal, que pueda descubrirse cualquier acción indebida o errónea.

Fase 1: Integridad

Después de establecidas las condiciones elementales para un buen diseño y estando asegurada físicamente la base de datos, es necesario proseguir a determinar las RI que quedarán establecidas.

Paso 1

Los límites de dominio son la forma más elemental de restricciones de integridad. Son fáciles de probar por el sistema después que se introduce un nuevo dato en la base de datos. Se establecen las siguientes RI de dominio:

- A cada atributo estará asociado un dominio de valores posibles. Quiere esto decir que, se restringirán los valores que cada atributo puede tomar respecto a su dominio; por ejemplo, que el atributo edad de una base de datos tendrá el siguiente dominio: (EDAD <= 1-120).
- Se establece además, que puede darse el caso de que varios atributos tengan el mismo dominio. En esta situación se establece un dominio para ese conjunto de atributos. En principio lo que hay detrás de los dominios de atributos, es similar a la asignación de tipos de variables en los lenguajes de programación.
- Trate de usar campos que no puedan tener valores nulos (*Not Null*). Los valores nulos ralentizan las lecturas.

Paso 2

Después de establecidas las restricciones de dominio, se prosigue a establecer las de entidad. La integridad de entidad define una fila como entidad única para una tabla determinada. La integridad de entidad exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índices y restricciones. Por lo que se establecen las siguientes restricciones:

- Se debe establecer una clave que identifique cada tupla en la relación. Es decir un atributo de una tabla que será distintivo entre los demás y que será único en toda la relación.
- Se determina que puede darse el caso de que sea un conjunto de atributos y no un solo atributo el que conforme la llave primaria. Para este caso en particular se establece que como llave primaria, se tome el valor de un conjunto de atributos que determina unívocamente el valor de otro conjunto de atributos.

- Cuando se modele la relación de muchos a muchos, la nueva tabla que se genera contendrá como llave primaria (PK) un auto numérico y se creará una llave alterna (ALTERNATE KEY ó AK) con las llaves foráneas (FK) que migraron de las tablas que se relacionan.
  NOTA: Es el administrador quien establecerá como auto numérico el valor de la llave primaria (PK) de la nueva relación. De esta forma se garantiza que la combinación de las tablas padres nunca se puedan repetir.
- Se establece que la máxima cantidad de atributos llaves de una tabla será de tres. Si se incrementa esta cantidad entonces se utilizarán las llaves alternas (AK), como se explicó en el punto anterior, para simplificar la cantidad de atributos llaves de la relación.
  NOTA: Este último elemento es una buena práctica de diseño, establecida en esta investigación como una restricción de entidad.
- Hacer que los atributos clave primaria no puedan ser nulos (NOT NULL).
- Para mejorar una consulta (SELECT), hay que crear un índice sobre los campos que son utilizados en las búsquedas (los que aparecen en las cláusulas WHERE o JOIN), por lo que se recomienda utilizar índices esencialmente sobre campos con valores únicos, ya que los índices son menos efectivos, si el campo tiene valores duplicados.
  NOTA 1: Se recomienda crear índices sobre las llaves foráneas de las tablas hijas, que se relacionan con la tabla padre a través de una relación de borrado en cascada.
  NOTA 2: No cree índices innecesarios, pues estos se actualizan con cada cambio en la tabla asociada y pueden ralentizar las modificaciones de la misma.

Con la creación de las claves primarias por cada una de las tuplas de la base de datos se logra de manera rápida y eficiente la búsqueda de los datos en una tabla, además de que se preserva la integridad de los datos.

Paso 3

Ya establecidas las RI de dominio y de entidad, se pasan a establecer las RI referencial.

Estas restricciones estarán establecidas sobre las claves ajenas: es decir si en una relación hay alguna llave foránea, sus valores deben coincidir con los valores de la llave primaria a la que hace referencia, o bien deben ser completamente nulos. A partir de aquí quedan establecidas las siguientes normas de integridad referencial:

- Para cada clave ajena de la base de datos, tendrá que existir la clave primaria a la que ésta hace referencia. Es decir por cada llave foránea debe existir la respectiva llave primaria a la que ésta se refiere.
- En caso de que la llave foránea haga referencia a un elemento que no se encuentra ya en la base de datos, porque sufrió alguna modificación, entonces el valor que ocupe debe ser completamente nulo.
- Se establece que el borrado de los elementos de una tabla de la base de datos, se realizará en cascada. De manera tal que, no exista inconsistencia entre los datos que están almacenados.
- Se tendrán que realizar tres preguntas para cada llave foránea de la base de datos:
  1. Reglas de los nulos: ¿Tiene sentido que la clave ajena acepte nulos?
  2. Regla de borrado: ¿Qué ocurre si se intenta borrar la tupla referenciada por la clave ajena?
     - Restringir: no se permite borrar la tupla referenciada.
     - Propagar: se borra la tupla referenciada y se propaga el borrado a las tuplas que la referencian mediante la clave ajena.
     - Anular: se borra la tupla referenciada y las tuplas que la referenciaba ponen a nulo la clave ajena (sólo si acepta nulos).
  3. Reglas de modificación: ¿Qué ocurre si se intenta modificar el valor de la clave primaria de la tupla referenciada por la clave ajena?
     - Restringir: no se permite modificar el valor de la clave primaria de la tupla referenciada.
     - Propagar: se modifica el valor de la clave primaria de la tupla referenciada y se propaga la modificación a las tuplas que la referencian mediante clave ajena.
     - Anular: se modifica la tupla referenciada y las tuplas que la referenciaban ponen a nulo la clave ajena (sólo si acepta nulos).

Fase 2: Seguridad

Después de establecidas las restricciones de integridad, quedan garantizados parte de los elementos a tener en cuenta para la seguridad dentro de una bases de datos.

La acción que sigue, se refiere a la protección frente a accesos malintencionados. Es importante tener claro que lograr la protección absoluta es imposible, pero se puede garantizar en alguna medida, por lo que se propone seguir los siguientes pasos.

Paso 1

El primer elemento que será analizado en este apartado es, el de los roles, ya que es importante establecer quién tendrá autorización para modificar los esquemas, índices y recursos de la base de datos, además de la autorización para alterar los atributos de una relación y eliminar las relaciones como tal, es decir determinar quién trabajará sobre ciertos objetos de la base de datos y quién no (privilegios). De esta forma se propone:

- Crear un rol para cada aplicación que acceda a la base de datos (rol de aplicación). Esto puede variar en dependencia de los intereses del diseñador y de las características que tenga su negocio.
- Crear un rol para cada tipo de usuario que se conecta con la base de datos a través de la aplicación (rol de usuario). Esto puede variar en dependencia de los intereses que tenga el diseñador y de las características que tenga su negocio.
- Para el caso en que el SGBD que se defina utilizar en el proyecto soporte el uso de paquetes, se recomienda, agrupar todas las funciones de búsqueda, es decir las consultas de selección (SELECT) en estos paquetes; lo que contribuye a realizar agrupaciones por funcionalidades, para que cada aplicación que se conecte con la base de datos, tenga asociada a ellas el paquete donde estén todas las funciones que esa aplicación necesite y no tengan acceso a otras, que no tienen nada que ver con ella, logrando establecer de esta forma niveles de seguridad. La manera en que se conformen los paquetes dependerá estrictamente del negocio en el que se trabaja.

  NOTA: En dependencia del interés de los diseñadores se le puede dar permiso a todos los roles sobre los paquetes, ya que éstos sólo tendrán acciones de lectura, y sólo podrán realizar inserción y modificación a aquellos que, puntualmente, se autorice trabajar, sobre procedimientos de este tipo.
- Las inserciones y actualizaciones (INSERT, DELETE, UPDATE) deberán estar encapsuladas en procedimientos apartes, no se establecerán dentro de paquetes. De manera tal que, a los roles se les de acceso sobre procedimientos puntuales. Los diseñadores pueden establecer los procedimientos de la manera que prefieren, lo que si debe quedar claro es, que no se puede violar la independencia entre los paquetes y los procedimientos.

NOTA: Con este paso se logra controlar el acceso a los objetos de la base de datos, garantizando la confidencialidad, ya que los privilegios estarán otorgados de forma concreta a cada rol, estableciendo la seguridad a través de niveles de acceso.

- Se establece que cada funcionalidad que se realiza en el sistema, es un rol, y ese rol es indivisible, es decir, ese rol ya no se puede subdividir más. Los roles pequeños, serán las unidades más chicas e indivisibles, por lo que en la base de datos existirán cientos de estos roles a nivel físico. Establecer las funcionalidades, queda por parte del diseñador en dependencia de las características específicas que tenga su negocio.
- Se establecerán procedimientos, funciones, vistas, etc., que conformarán en su totalidad a un rol determinado, por lo que no se puede quitar nada, si no entonces este rol no funciona. La asignación de cada una de estas funciones a un rol determinado, será responsabilidad del diseñador, quien lo establecerá en dependencia de sus intereses y las particularidades de su negocio.
- Se establecen tablas a nivel lógico, es decir en el diseño que representarán los roles grandes; así se llamará a los roles que tendrán asignados todos estos roles pequeños que se mencionaron con anterioridad; todo ésto en dependencia de los privilegios que se le quiera dar a este rol grande.

  NOTA: Para no tener que saber el nombre de todos los roles pequeños y sea más fácil referirse a un rol especifico, lo que se propone es crear estos roles grandes a nivel lógico, de manera tal, que cada rol grande contenga las funcionalidades de muchos roles pequeños, pero que en sí los roles pequeños serían los que persistirían de manera física en la base de datos.

Paso 2

Después de establecidas las acciones a tomar con los roles de la base de datos, el segundo elemento que será analizado en este apartado es el de los usuarios. Ya concedidos los privilegios a los roles se impone asignar a los usuarios el rol o los roles que van a desempeñar dentro de la base de datos. La opción que se propone es la más difícil, pero al final la más satisfactoria, puesto que, se deberá conformar un sistema de seguridad para la integridad de los datos almacenados y la propia aplicación o aplicaciones. A este punto del procedimiento se dará la posibilidad de gestionar los usuarios de la manera más óptima posible. De esta manera se establece que:

- Las aplicaciones nunca deberían conectarse a la base de datos bajo el usuario correspondiente al administrador de la base de datos, lo que se puede llamar también súper-usuario, ya que estos usuarios tienen acceso a todos los objetos de la base de datos, y a realizar cualquier operación sobre ellos.

  NOTA: Es importante garantizar que desde la aplicación no se pueda conectar ningún usuario con el rol de administrador, éste sólo tendrá acceso a la base de datos conectándose directamente al SGBD.
- Los usuarios de la aplicación serán manejados de la misma forma que los roles, es decir a nivel de tablas en la base de datos y físicamente. El diseñador establecerá los diferentes usuarios de la aplicación, en dependencia de sus intereses particulares y las necesidades de su negocio.
- Cuando se crea un usuario, a éste se le puede asignar cualquier rol de la base de datos. Ahora como se estableció anteriormente, se tendrán roles solamente lógicos (los llamados roles grandes), que contendrían a roles físicos (los llamados roles pequeños) y/o lógicos. Antes de asignarle un rol a un usuario, si el rol es lógico, se buscarán todos los roles físicos que lo conforman, para poder hacer un GRANT a nivel de base de datos.

  NOTA: La sentencia GRANT es una sentencia de Oracle para otorgar privilegios.
- En el momento en que se crea un usuario, se le asignará un *password* desde la aplicación, que será encriptado por la misma, de manera tal que a nivel de aplicación el *password* sea uno y a nivel de base de datos sea ese mismo pero encriptado.

  NOTA: Con ésto se logra, si alguien quisiera conectarse directamente a la base de datos, no va a poder hacerlo, pues a nivel de base de datos su *password* no es el mismo con que se autentica desde la aplicación, al menos no en texto plano, sino encriptado.
- Es importante establecer las direcciones IP desde las que se puede conectar una persona a la base de datos. Esto lo puede establecer el diseñador en dependencia de sus intereses y las características de su negocio, y el administrador sería quien se encargaría a partir del SGBD de limitar la direcciones IP específicas desde las que se puede establecer una conexión a la base de datos.

  NOTA: De esta forma se asegura la base de datos, por si algún día una persona cualquiera puede evadir la seguridad de la red, las limitaciones establecidas de las direcciones IP, restrinjan la conexión, intromisión y pérdida de la información.

- No se permitirá conceder privilegios de acceso de unos usuarios de aplicación a otro usuario de la aplicación. Esto sólo le estará permitido al administrador de la base de datos, que es el encargado de crear todos los usuarios y asignarle en dependencia de sus características, los roles que necesita para desempeñar su función.
- Se establece una política de seguridad por contraseña, para mantener la seguridad de acceso a la base de datos. Las contraseñas deben contar con más de siete caracteres, algún carácter especial y la combinación de letras y números. Los usuarios estarán forzados a modificar su contraseña cada cierto periodo regular de tiempo, lo que será notificado por los administradores. NOTA 1: Al momento de establecer la contraseña se puede evaluar desde la aplicación cuan fuerte es; en caso de que no cumpla con los parámetros fijados en las políticas se seguridad, el nivel de fortaleza será menor, por lo que se deniega la confirmación de la contraseña de un usuario determinado.
  NOTA 2: De esta manera se limitan los accesos a bases de datos sin autorización.

Paso 3

Para establecer los roles y los usuarios, se pasa a determinar el trabajo con las vistas. Crear medidas de seguridad sobre las vistas es de suma importancia para garantizar la integridad de los datos que se almacenan, ya que las vistas nos permiten mostrar la información de una tabla a determinados usuarios, estableciendo así privilegios, de manera tal que no todos tengan acceso a todos los objetos de la base de datos, sino a aquellos que se les permita. Queda claro que, sirven como mecanismo de compartimentación de la información almacenada, permitiendo presentar a diferentes usuarios parte del universo, según se considere necesario. De esta manera:

- Se establece la política de seguridad de que los usuarios de la aplicación no tendrán acceso directo a las tablas completas, sino que lo harán a través de las vistas, las cuales, por ser un objeto, están sujetas a otras medidas de seguridad. La creación de las vistas y los privilegios asignados a través de los roles a éstas, será una función que realizará el administrador de la base de datos y, que por demás, estará condicionada por las características particulares del negocio.
- Cuando se actualice una relación base, el cambio se refleja automáticamente en todas las vistas que la referencian. Del mismo modo, si se actualiza una vista, las relaciones base de las

que se deriva deben reflejar el cambio. Esto está sujeto, a determinadas restricciones que se establecen, para determinar los tipos de modificaciones que se pueden realizar sobre las vistas.

- Las vistas, deben contener de forma específica los periodos en los que debe considerarse la información que representa como vigente. Posibles fechas de caducidad o actualización, así como fechas en que se genera la información presentada. Esto se establece en dependencia de las particularidades de cada negocio.
- Las vistas, deben contener información precisa de las fuentes utilizadas para generar la información; de los responsables directos o indirectos de su generación y de los medios utilizados para ellos.
- Las vistas, en casos especiales, pueden incluir valores o datos, que permitan la toma de decisiones o la aplicación de un criterio sobre la información contenida en la vista. Esto se establece en dependencia de las particularidades de cada negocio.
  NOTA: Estos datos adicionales pueden ser utilizados para completar procesos, cálculos o delimitar áreas de acción.
- La vista, debe incluir para quién es útil la información que ella presenta, es decir, se deben especificar destinatarios indirectos o directos de la información presentada en la vista. Esto está sujeto a las particularidades de cada negocio.

Fase 3: Trazabilidad

Hasta este punto, se han mencionado, los elementos relacionados con la integridad y la seguridad, se impone continuar con la manera que se establecerá, para mantener trazas de la información.

Como fue señalado, anteriormente en la presente investigación, siempre cuando se diseña un software, es muy importante mantener trazas del sistema, para poder ver qué pasó en cada momento y poder decir que una u otra acción se ejecutó mal o de forma incorrecta. La creación de trazas garantiza poder realizar estos análisis con conocimiento de causa, es decir, con los elementos en la mano que brindan toda la información sobre lo ocurrido. La manera de persistir las trazas, siempre cada diseñador las establece de manera particular, pero a continuación, se propone, una muy eficaz y que ha sido analizada con detenimiento.

Las trazas que se van a mantener serán subdivididas en trazas a nivel de objetos y trazas a nivel de datos, de forma tal que no se cambien ninguno de los elementos de la base datos relacionados con la seguridad, teniendo todos los elementos que garanticen la auditoría.

Paso 1

Se parte estableciendo las trazas a nivel de datos, para poder garantizar la auditoría sobre la información que se almacena, es decir, tener el control total sobre las funciones (INSERT, DELETE Y UPDATE).

Después del análisis de métodos que garanticen la trazabilidad, se concluye que la solución que brinda el *SubVersion* (SVN) para controlar las trazas de su base de datos. El SVN modela una tabla que contiene ID, NOMBRETABLA, CAMPO, VALORANTERIOR, VALORNUEVO, ACCION, FECHA y USUARIO. A partir de aquí se toma esta idea y se adecua, para brindar esta solución. Se establece que el administrador de la base de datos debe:

Para cada tabla de la base de datos que se quiera auditar se crea un *trigger*, que inserta en la tabla que se propone llamar AUDITORIA, cada vez que se realice alguna acción sobre la tabla desde donde él se dispara. La tabla AUDITORIA tendrá una estructura idéntica a la que se describió que proponía el SVN para llevar sus trazas.

> NOTA 1: Esta es la manera en que se aplicó la idea tomada a partir de la forma en que trabaja el SVN para mantener sus trazas. Esta es la solución propuesta y los pasos a seguir para garantizar el registro de las trazas.

Ejemplo

Se tiene una tabla PERSONA. Tiene como atributos IDPERSONA, NOMBRE, SEXO. Cuando se inserta en esa tabla la siguiente tupla lo siguiente (1, Maria Antonia, F) inmediatamente se escriben en la tabla AUDITORIA lo siguiente:

> 1, TABLAPERSONA,IDPERSONA,null,1,1,12/5/2008,malajus
>
> 1, TABLAPERSONA,NOMBRE,null,´Maria Antonia´,1,12/5/2008,malajus
>
> 1, TABLAPERSONA,SEXO,null,´F´,1,12/5/2008,malajus

Cada campo de la tabla es el que inserta los valores anteriores y nuevos que van tomando.

> NOTA 2: Es importante destacar que después del valor nuevo, se pone un 1 porque así determinó el autor, es decir que establece como 1 a la operación insertar (INSERT), 2 podría ser actualizar (UPDATE) y 3 eliminar (DELETE). Por tanto si en cualquier momento se quisiera actualizar el nombre de ´Maria Antonia´ por ´Maria´, se ejecutará una sentencia (UPDATE), entonces se escribiría en la tabla AUDITORIA lo siguiente:

1,TABLAPERSONA,NOMBRE,´Maria Antonia´,´Maria´,2,13/5/2008,malajus

De igual manera si en cualquier momento lo que se quisiera es eliminar a persona se ejecutará la sentencia (DELETE), y se escribirá en la tabla AUDITORIA lo siguiente:

1, TABLAPERSONA,IDPERSONA,1,null,3,12/5/2008,malajus

Aplicando esta solución se logra una total trazabilidad del sistema, pero se incurre en costos de rendimiento, por lo que la organización debe valorar qué hacer al respecto.

Paso 2

Se continúa estableciendo la forma de crear las trazas a nivel de objetos, para poder garantizar la auditoría de los cambios de estructura que puedan presentar los objetos de la base de datos. De la misma forma que se realiza sobre las funciones (INSERT, DELETE Y UPDATE), poderla establecer a partir de *trigger*, para poder saber cuándo se hace algo con los valores de una tabla, o cuándo cambia un procedimiento, etc., de tal manera que se puedan auditar tanto objetos, como datos.

De igual forma a como quedó establecida la manera en la que se realizarán las trazas a nivel de datos en el Paso 1, se realizarán a nivel de objetos, con la diferencia de que los *trigger* se crearán a partir de los eventos que ocurran sobre los objetos. Se establece que el administrador de la base de datos debe:

Para cada objeto (procedimientos, tablas, etc.) de la base de datos que se quiera auditar se crea un *trigger*, que inserta en la tabla que se propone llamar AUDITORIA, cada vez que se realice alguna acción sobre la tabla desde donde él se dispara. La tabla AUDITORIA tendrá una estructura idéntica a la que se describió que proponía el SVN para llevar sus trazas.

NOTA: De esta manera quedan garantizados otros aspectos de seguridad, pues no solo se tienen presentes los datos sino también los objetos de la base de datos. Si en algún momento, por ejemplo, un procedimiento se cambia y ésto no se actualiza en el repositorio, entonces existiría inconsistencia, porque lo que se esperaba que este hiciera o devolviera, cambió. Esto es importante prevenirlo a tiempo porque después que la base de datos se explora nadie puede cambiar ningún objeto, sino la aplicación asociada a ella no funcionara.

Paso 3

Hasta este momento con la utilización de esta herramienta se tiene una traza completa de qué pasa con todos los campos de una tupla, qué pasa con todas las tuplas de una tabla y finalmente con todas las tablas de la base de datos.

Surge una contradicción, porque a pesar de que esta solución está muy bien pensada, se tiene que tener cuidado con la creación de los índices definidos, pues esta tabla crecerá mucho y puede traer problemas a nivel de motor de base de datos. Para solucionar ésto en el presente procedimiento se propone la creación una tabla que represente un historial, porque contiene una copia de la tabla que propone el SVN, de manera que se pueda limpiar cada cierto tiempo esta tabla y que no se pierda la información que contenía; información importante para la toma de decisiones.

- La tabla que propone el SVN deberá limpiarse cada cierto tiempo, por el problema que se expuso anteriormente, de lo que representa a nivel de motor de base de datos que la tabla crezca tanto. La periodicidad con que se produzca la limpieza de esta tabla estará determinada por las necesidades que se presenten, a partir de las características del negocio en que se trabaje.
- Cada vez que se realice la limpieza, dependiendo del tiempo que se haya establecido es responsabilidad determinar si la información se almacena otra tabla que representa un historial o se desecha, borrándose totalmente. Esta es una decisión tomada en dependencia de las características puntuales del negocio y los intereses que a partir de aquí tenga el diseñador.
- En caso de que se decida almacenar la información en un historial, se creará una tabla llamada HITORIAL, que tendrá una estructura idéntica a la tabla que quieres guardar para este historial y cuando se cumpla el tiempo establecido todas las tuplas de esta tabla que cumplan ese tiempo pasan al historial. Es decir es una limpieza de la tabla que contiene las trazas, lo que no se desecha esta información sino que se guarda en esta tabla HISTORIAL, de manera que se puedan tomar decisiones a partir de esta información en el momento que se requiera.

  NOTA: De esta manera se logra que el procedimiento sea más configurable, pues si usted determina que es necesario mantener almacenadas las trazas, para tomar decisiones ante posibles problemas que se puedan presentar, pues se establece la tabla HISTORIAL, que garantiza que en cualquier momento se podrá monitorear todo lo que ocurre y quién realiza cada operación en la base de datos.

Fase 4: Salva y recuperación

Dependiendo de la cantidad de datos que se manejen se podrá establecer el tipo de respaldo. Hacer un respaldo consume recursos del servidor de base de datos, por lo que no se lo debe de hacer cuando tienes a todos los usuarios conectados, pero sí cuando todos se van a su casa o a la hora del almuerzo, por ejemplo. Es riesgoso hacer sólo un respaldo diario, pues se acostumbra a sólo sacar un respaldo al final de cada día, se perdería información, es decir quedarían márgenes de tiempo en los que podría ocurrir una catástrofe y entonces, ser perdería la información que de manera inmediata fue almacenada en la base de datos. Para que esto no ocurra se establece:

Paso 1

Se podrían realizar varios respaldos a lo largo de todo el día, pero teniendo en cuenta que existen bases de datos con mayor volumen de información que otras y que manejan información más crítica, se propone realizar un solo *backup* de la información almacenada durante el día, ya que asumir la primera postura menciona, conlleva a que se produzca una sobrecarga en los servidores. La manera de realizar este respaldo, estará determinada por las particularidades de cada negocio y de los requisitos de seguridad que se tengan. Los respaldos pudieran efectuarse automáticamente, utilizando los dispositivos correspondientes y programándolos de manera tal que se ejecuten en el tiempo establecido.

En el caso particular de RN, por ejemplo, la salva esta planificada por días de la semana, de manera se establecen periodos de salva, donde los lunes se realiza una salva de ese día, el martes de igual forma y el miércoles se realiza la salva de toda la información procesada del lunes, el martes y el miércoles. El jueves se repite el mismo proceso, al hacer la salva de este día y el viernes de igual forma, para el sábado repetir el proceso en que realiza la salva de la información de esos tres días, desde el jueves hasta el sábado. Y finalmente el domingo se realiza la salva de todos los días de la semana.

Este es solo un ejemplo de cómo pudiera planificarse este proceso. Ya que se pueden establecer cualquiera de los tipos de copias de seguridad que existen; es decir el backup diferencial, backup incremental, backup completo o backup espejo; que son los diferentes tipos que existen. La manera en que se planifique estará en dependencia de cada especialista, que deberá plantearlo a partir de las particularidades del negocio.

Para garantizar la salva de cada uno de los servidores se propone, la selección de una herramienta que integre los mecanismos necesarios para realizar la recuperación. Es importante que dicha herramienta sea capaz de de administrar el respaldo y arquitecturas de almacenamientos de múltiples servidores. Existen distintas herramientas que garantizan cada uno de estos elementos, algunas privativas y otras libres, pudiendo mencionarse:

1. *Backup Watcher* para MySQL se ha pensado para hacer copias de seguridad de bases de datos de MySQL Server. El almacenamiento se puede hacer en un ordenador local o en un servidor remoto en Internet incluso tras un firewall.
2. *Handy Backup* facilita un modo rápido y eficaz para hacer copias de seguridad de *Microsoft SQL Server*, que es un sistema para administrar bases de datos diseñado por *Microsoft*, realmente se utiliza mucho.
3. Finalmente la herramienta *HP OpenView Storage Data Protector 6.0*, la misma integra en un mismo producto la recuperación basada en disco y en cinta

La administración de la herramienta seleccionada estará alojada en uno de los servidores pertenecientes al centro en que se encuentre almacenada la base de datos; sitio que se hará llamar, para un mejor entendimiento, centro de datos, pues será el lugar donde esté almacenada toda la información relacionada con el proyecto en cuestión, y donde deberán quedar establecidas todas las medidas de protección física que fueron mencionadas. Por esta razón la configuración y administración de esta herramienta, solamente puede ser implementada por el personal que se designe; preferiblemente el personal será seleccionado por la dirección del órgano que se este informatizando o la representación directiva que se beneficia con el proceso que se automatiza.

Paso 2

Para realizar la planificación de *backup* y del periodo de tiempo en que éstas se realizarán se tuvo en cuenta las prestaciones del HP OpenView Storage Data Protector 6.0, que es el software de salva que se propone usar por la experiencia de la autora. También se tuvo en cuenta, algunos conceptos relacionados como, nivel de los *backups* y retención de los *backups*. A partir de aquí se establece:

Estructura general de salva

Al concluir la instalación y configuración de cada servidor, se realiza una imagen de todo el Sistema de Archivos y es a lo que se denomina Software Base, esto garantiza la recuperación en el menor tiempo posible, en caso de un error grave.

Se define como Software Base el conjunto de aplicaciones que se instalan como parte de los servicios que aloja el servidor: (24)

- Sistema Operativo
- Aplicaciones de Negocio.
- Antivirus.
- Motores de Datos.
- Otros.

Para lograr una organización lógica, se define, que los nombres de cada una de las imágenes realizadas a cada uno de los servidores seguirán la siguiente nomenclatura en su nombre:

"Imagen"+NombreServidor

En el caso de los servidores con sistema operativo GNU/Linux se tienen que hacer 2 imágenes por servidor, una del Sistema de Archivo y la otra de una de las particiones que no tiene el mismo sistema de archivos que las demás; en este caso al nombre de la imagen se le adiciona FS o B.

Ejemplo:

Servidor	Nombre de la Imagen
svbdsar01	**Imagensvbdsar01FS**
svbdsar01	**Imagensvbdsar01B**
svunimij01	**Imagensvunimij01**

El sistema operativo que posee el servidor, definirá la manera de realizar las salvas, cada tipo de sistema operativo provee herramientas para realizar dichas imágenes. El resultado de dicho proceso

genera un fichero, que es almacenado en otro servidor que se denominará Área de Almacenamiento, y se le realiza una salva utilizando la herramienta HP Open View Data Protector 6.0. Cada vez que se realicen actualizaciones al Sistema Operativo o al Software Base que contiene el servidor, se deberá realizar una actualización de dicha salva, a este tipo de configuración se le denomina *Salvas en Demanda.* (24)

Paso 3

Estructura general de recuperación

Para que la recuperación sea efectiva deberán cumplirse los siguientes requerimientos.

1. La configuración del hardware del servidor no puede sufrir cambios.
2. El requisito de mantener el mismo hardware está relacionado principalmente con las particiones que tienen discos duros.
3. Debe estar establecida la configuración de la red. Cómo y desde dónde se tendrá acceso al centro de datos.
4. Disponer de las contraseñas administrativas locales existentes en el momento en que se creó el *backup*. Si no se tiene esta información no se podrá ingresar al sistema una vez restaurado. Dicha información no se expone en este documento por razones de seguridad y porque además, serán establecidas de acuerdo a las políticas del negocio en cuestión, y será responsabilidad del administrador del centro de datos donde se encuentre alojada toda la base de datos.

Recuperación de todo el sistema de archivo del servidor

Todas las fallas que se clasifican como graves (los dos discos duros fallan, fallas graves del sistema) son las que se proceden a recuperar, a partir de la salva realizada a la imagen del Software Base y después se recupera cada uno de los servicios que aloja el servidor mediante HP Open View Data Protector 6.0 las cuales se especifican de acuerdo a los servicios de cada servidor.

La recuperación a partir de la imagen Software Base al igual que la salva, se realiza según el sistema operativo instalado en el servidor. (24)

Recuperación de cada servicio

Se selecciona el *backup* más reciente para obtener la configuración precisa antes de la falla.

Después de realizada esta restauración, los administrador(es) de la aplicación procederá a ejecutar todos los servicios y verificar si cada uno de éstos funciona de manera correcta.

A continuación se especifican los procedimientos a seguir, teniendo en cuenta los distintos tipos de fallas que pueden afectar el funcionamiento de los servicios. (24)

Recuperación ante fallas de hardware

Lo primero que se debe hacer, es listar los elementos de hardware en los servidores que pueden ser objeto de daño o avería y generar un incidente, para a partir de ahí, se puedan establecer las medidas a tomar, en dependencia de los que se tengan. Ejemplo de estos elementos, son las fuentes, los discos duros, los procesadores, la Motherboard, las tarjetas de fibra óptica, las tarjetas de red, la memoria RAM, etc.

La solución que se propone de forma general, para cuando ocurra alguna falla en cualquiera de estos elementos de hardware, es llamar a soporte técnico. Ahora en el caso particular en que tengamos dos fuentes y se avería una sola, pudiera retirarse la dañada y trabajar con la que está en buen estado.

Procedimiento para cuando hay un disco defectuoso

Se recomienda que cada servidor, cuente con dos discos duros configurados y que la información se encuentre redundante en los dos. De manera tal que al fallar uno de los dos discos, se extrae el disco con problemas y se levanta el sistema con el otro disco disponible. El sistema estaría trabajando con el arreglo degradado hasta que se reemplace el disco defectuoso. Al reemplazar el mismo, se reconstruye el arreglo de discos con las herramientas que provee HP Open View Data Protector 6.0. (24)

Paso 4

Finalmente se deben establecer los horarios en los que se efectuarán las salvas de cada servidor o medio de almacenamiento con que se cuente, a partir de las características en particular que tenga cada negocio y de las necesidades puntuales que existan. De este modo se propone la creación del plan de salvaguarda. En el que se establecerán:

- En este plan constará el horario y el período en que se realizará la salva de cada uno de los medios de almacenamiento conque se cuente en el centro de datos, es decir en el lugar donde estén los medios que contienen la información.

- Se propone establecer los medios en que será salvada la información, es decir los dispositivos de almacenamiento (CD, DVD, Discos duros, TAPE, etc.). La selección de los dispositivos, así como la distribución de la información exacta que será almacenada en cada cual será establecida por el responsable de este proceso dentro del grupo de desarrollo.
- NOTA: Se debe diseñar una política que establezca el tiempo de retención de los dispositivos de almacenamiento que se consideran reutilizables.
- Se propone establecer las normas que regulen el proceso de salvaguarda. Estas normas estarán determinadas por las características del negocio en que se trabaje.

De igual manera se propone la creación de un plan para el proceso de recuperación, donde deberá establecerse que:

- Se propone crear normas que regulen el proceso de recuperación, las que estarán determinadas por el negocio en que se trabaje.

Se establece que debe planificar para cada servidor teniendo en cuenta los servicios y aplicaciones, la configuración de las salvas y la recuperación a partir del procedimiento que se brindó de manera general para cualquier servidor.

De cualquier forma, este método debe probarse periódicamente para asegurar que funcionen completamente y que sean fiables.

1. Aspecto a considerar

Han quedado establecidos todos los pasos a tener en cuenta para lograr garantizar la seguridad de una base de datos, a partir de los elementos de seguridad que se tuvieron en cuenta en el procedimiento propuesto. Teniendo en cuenta que no todas las bases de datos presentan la misma complejidad, el mismo volumen de información, las mismas características y fundamentalmente los mismos requerimientos de seguridad, se proponen dos niveles de seguridad:

Nivel 1

En el que estarán todos los elementos que se relacionan dentro del procedimiento en la fase que garantiza la integridad, la fase que asegura la seguridad y la fase relacionada con el proceso de salvaguarda y recuperación.

Nivel 2

Este nivel contendrá todos los elementos que plantea el primer nivel, es decir estará a un nivel 1. Además se incluyen los aspectos relacionados con la fase tres, en la que se establece todo lo relacionado con la trazabilidad.

De esta forma, cada especialista responsable de la seguridad, seleccionará cual de estos niveles incluir en el diseño e implementación de la base de datos, partiendo de las necesidades de cada negocio en particular.

A pesar de estos niveles, es importante aclarar que ninguno de los pasos propuestos es obligatorio, pues están estructurados de forma flexible, para que se puedan aplicar a partir de las necesidades particulares de cada negocio.

Poscondiciones

Una base de datos asegurada en cierta medida, capaz de garantizar los elementos de la seguridad de los sistemas de información; aspectos que no se deben violar, como la integridad, disponibilidad y confidencialidad.

Conclusiones

El conocimiento de los elementos administrativos y de control es necesario para la gestión efectiva de un sistema de base de datos. El mantenimiento de la seguridad y de la integridad de la base de datos es a la vez complejo y esencial, debido al acceso de múltiples usuarios en forma concurrente a los sistemas de base de datos.

Este capítulo se dedicó primero, a realizar un análisis de los diferentes aspectos que se tienen en cuenta para garantizar la seguridad en los sistemas de información, realizando una selección de los que serían tomados en cuenta en la investigación, a partir de la experiencia e intereses de la autora. Los elementos que se tuvieron en cuenta fueron: la integridad, la seguridad, la trazabilidad y la salva y recuperación de la información.

A partir de aquí se realizó un análisis de cada uno de estos aspectos, comentándose criterios y consideraciones particulares al respecto, de manera tal, que se pudo arribar a conclusiones parciales de cada uno de estos temas; que posteriormente serían elementos tomados en cuenta para conformar el procedimiento, que es la propuesta de solución final.

En términos de integridad, se reflejó los tipos de RI que se pueden crear en el modelo relacional. Definiéndose que las reglas de integridad de entidades, son una restricción que dice que ninguno de los atributos que forman la clave primaria puede ser nulo. Las reglas de integridad referencial, donde los valores de las claves ajenas deben coincidir con alguno de los valores de la clave primaria a la que hacen referencia, o bien ser completamente nulos.

Por otro lado, se concluye con relación a la seguridad de la base de datos; que en principio está relacionada con la determinación de quién tiene acceso legítimo a qué datos, entonces, con la seguridad se refuerza la legitimidad. La autenticación se refiere, a los métodos para restringir el acceso al sistema. La autorización se refiere, a los métodos de control de los recursos, a los que se puede tener acceso una vez que se haya obtenido acceso al sistema, así como qué se puede hacer con esos recursos.

Se vio la importancia de mantener trazas de los flujos que se mueven dentro de una base de datos, de manera tal que se sepa en todo momento qué y quién realizó una operación, de manera tal que si en cualquier momento es necesario hacer alguna toma de decisiones, tener todos los elementos en la mano. De aquí la importancia de que exista un sistema que permita auditar la base de datos.

A lo largo de la presente investigación, se han mencionado las directrices generales cuya aplicación en la práctica puede destrozar un diseño perfecto, por ejemplo, fallos en el software (de funcionamiento o específicos de seguridad), incompatibilidades entre las diferentes aplicaciones, la incorrecta dimensión de los recursos de hardware y comunicaciones, los malos hábitos de las personas (sobre todo cuando no son conscientes del por qué de los controles o no se les ha transmitido la importancia de mantener la seguridad en la información que manejan).

Por tal motivo es importante, que todos los sistemas de bases de datos tengan procedimientos seguros de copias de seguridad, es decir, establecido un proceso para la salvaguarda y recuperación de la información, para evitar ineficiencias y, aun más, pérdidas catastróficas. Los diarios de operaciones de transacciones con actualizaciones diferidas o inmediatas, los puntos de comprobación y las propias copias de seguridad (*backup*) de la base de datos, son elementos esenciales para su seguridad y recuperación.

Finalmente, después de un análisis profundo de todos estos elementos, se realizó la propuesta del procedimiento. Este fue dividido en cuatro fases, cada una representa a los elementos de la seguridad que se persigue garantizar. Las fases incluyen pasos a seguir y con cada uno de estos pasos se van

estableciendo normas que aseguran el cumplimiento de estos elementos durante el diseño, implementación y administración de una base de datos. Se estableció además, una precondición, pues antes de comenzar a desarrollar cada fase del procedimiento, es importante haber asegurado, dos elementos: un buen diseño de la base de datos y las medidas necesarias para la seguridad física de la misma. Al aplicar este procedimiento, se obtiene cierta garantía de que la base de datos conque se trabaja es segura, lo que será validado en el siguiente capítulo.

CAPÍTULO 3: ANÁLISIS Y DISCUSION DE LOS RESULTADOS.

Introducción

En el presente capítulo se realizará un estudio de los estándares internacionales encontrados para la evaluación de los sistemas de información. La investigación contribuirá a tomar decisiones en cuanto a que método, estándar o técnica utilizar para realizar la validación de la solución. Se analizarán las características y normas de cada uno de los estándares que son: *Common Criteria* y la ISO 17 799, para arribar a conclusiones. Se describe la técnica que se aplicará para evaluar el procedimiento, explicándose cada uno de los pasos que se efectuarán. Finalmente se realizará el análisis y discusión de los resultados, a partir de los criterios emitidos por cada especialista contactado.

3.1 Common Criteria

Common Criteria, como señala Héctor Sánchez, responsable de Seguridad en Microsoft Ibérica, “es una norma que permite especificar y medir la seguridad de un producto”.

Common Criteria es el resultado final de importantes esfuerzos en el desarrollo de criterios de evaluación unificados para la seguridad de los productos de la Tecnología de la Información (TI) y ampliamente aceptado por la comunidad internacional. A principios de los años 1980, se desarrollaron en Estados Unidos, los criterios de seguridad recogidos bajo el nombre de TCSEC (*Trusted Computer System Evaluation Criteria*) y editados en el famoso "libro naranja". [(25)]

En las décadas posteriores, varios países tomaron como base el TCSEC americano y evolucionaron las especificaciones para hacerlas más flexibles y adaptables a la constante evolución de los sistemas de TI. De ahí la comisión europea, en el año 1991 publicó el ITSEC *(Information Technology Security Evaluation Criteria*), desarrollado conjuntamente por Francia, Alemania, Holanda y el Reino Unido. En Canadá, igualmente se desarrollaron en 1993 los criterios CTCPEC (Canadian *Trusted Computer Product Evaluation*) uniendo los criterios americanos y europeos. En ese mismo año, el gobierno americano publicó los *Federal Criteria* como una aproximación de unificar los criterios europeos y americanos. [(25)]

Tal escenario comienza a aclararse, con la decisión de estandarizar internacionalmente estos criterios para uso general, y en esa labor ISO comienza a trabajar a principios de los años 1990 dando como resultado la certificación *Common Criteria* (o ISO-IEC 15408)

¿Cuales son los beneficios de *Common Criteria*?

Los catorce países signatarios de los acuerdos de *Common Criteria*, llegaron a este arreglo porque permitiría establecer un único criterio con el que evaluar la seguridad de los productos de TI, contribuyendo a aumentar la confianza de los usuarios en los mismos. Esto es beneficioso porque habilita a los usuarios la posibilidad de tomar decisiones con información y criterio, por encima de otras consideraciones como son: [25]

- Los usuarios pueden comparar sus requerimientos específicos frente a los estándares de *Common Criteria* para determinar el nivel de seguridad que necesitan.
- Los usuarios pueden determinar más fácilmente cuando un producto cumple una serie de requisitos. Igualmente, *Common Criteria* exige, a los fabricantes de los productos certificados, publicar una documentación exhaustiva sobre la seguridad de los productos evaluados.
- Los usuarios pueden tener plena confianza en las evaluaciones de *Common Criteria* por no ser realizadas por el vendedor, sino por laboratorios independientes. La evaluación de *Common Criteria* es cada vez más utilizada como condición necesaria para concurrir a concursos públicos. Por ejemplo, el Departamento de Defensa Americano ha anunciado planes para utilizar, exclusivamente, productos certificados por *Common Criteria*.
- Debido a que *Common Criteria* es un estándar internacional, proporciona un conjunto común de estándares que los usuarios con operaciones internacionales pueden utilizar, para escoger productos que se ajusten localmente a las necesidades de seguridad.
- Proporcionando un conjunto de estándares en seguridad como los recogidos por *Common Criteria*, se crea un lenguaje común entre los fabricantes y los usuarios, que ambos pueden entender. Los fabricantes utilizarán este lenguaje para contar a sus clientes potenciales las características de sus productos evaluadas en *Common Criteria*, e igualmente habilita a los usuarios a identificar y comunicar, adecuadamente sus necesidades de seguridad.
- Finalmente proporcionan medios y mecanismos objetivos, que permiten tomar decisiones en base algo más sólido que las meras percepciones.

Se mencionó este estudio sobre el criterio de evaluación de la seguridad, con el objetivo de analizar si este estándar internacional serviría para evaluar la efectividad del procedimiento realizado, es decir de la solución propuesta. Ahora bien, se llegó a la conclusión de que el *Common Criteria* realiza sus evaluaciones sobre productos para la seguridad informática y los sistemas. La evaluación que realiza

sirve para validar las alegaciones sobre el objetivo, es decir para lograr un uso práctico, la evaluación debe verificar los elementos de seguridad.

Common Criteria, es un estándar internacional que proporciona un conjunto común de criterios, que los usuarios con operaciones internacionales pueden utilizar, para escoger productos que se ajusten localmente a sus necesidades de seguridad. Incluye, pero no está limitado, a la protección de la información en los aspectos de: confidencialidad, integridad y disponibilidad; el *Common Criteria* reconoce que no hay evaluación de la seguridad de TI, si no que los resultados se alcanzan, mediante la aplicación de criterios que contienen elementos objetivos y subjetivos, por ello disponer de índices o parámetros precisos y universales para la seguridad en TI, no es posible.

Después de revisado y analizado este estudio, la autora determinó no utilizar el criterio común para la evaluación de la seguridad de la tecnología de la información, para realizar la validación de la solución propuesta, ya que el *Common Criteria* describe el conjunto de acciones generales que el evaluador debe llevar a cabo, no indica los procedimientos a seguir, por lo que necesita una metodología conjuntamente, para poder ofrecer los resultados esperados. A pesar, fue un elemento a tener en cuenta, por ser el estándar más utilizado a nivel mundial para evaluar la seguridad de los sistemas de información.

3.2 ISO 17 799

El ISO 17 799 es un estándar internacional de alto nivel para la administración de la seguridad de la información, fue publicado por la ISO en diciembre del año 2000, con el objeto de desarrollar un marco de seguridad sobre el cual trabajen las organizaciones. [25]

El ISO 17 799, se define como una guía en la implementación del sistema de administración de la seguridad de la información y se orienta a preservar los siguientes principios de la seguridad informática: [25]

Confidencialidad. Asegurar que únicamente personal autorizado tenga acceso a la información.

Integridad. Garantizar que la información no será alterada, eliminada o destruida por entidades no autorizadas.

Disponibilidad. Asegurar que los usuarios autorizados tengan acceso a la información cuando la requieran.

Estos principios en la protección de los activos de información, constituyen las normas básicas deseables en cualquier organización, sean instituciones de gobierno, educativas e investigación; no obstante, dependiendo de la naturaleza y metas de las organizaciones, éstas mostrarán especial énfasis en algún dominio o área del estándar. Por tal razón, el estudio de este estándar se convierte en un elemento de la investigación, pues tiene presente los aspectos antes mencionados, que están estrechamente relacionados con la seguridad.

El objetivo que se persigue, al garantizar la seguridad de los datos, es asegurar la continuidad de las operaciones de la organización, reducir al mínimo los daños causados por una contingencia, así como optimizar la inversión en tecnologías de seguridad, y todos estos elementos están incluidos en esta norma.

Como todo buen estándar, el ISO 17 799 da la pauta en la definición sobre cuáles metodologías, normas o estándares técnicos pueden ser aplicados en el sistema de administración de la seguridad de la información, pudiendo entenderse que estos estándares son auxiliares y serán aplicados en algún momento al implementar el mismo.

La aplicación de un marco de referencia de seguridad, basado en el ISO 17 799, proporciona beneficios a toda organización que lo implemente al garantizar la existencia de una serie de procesos, que permiten evaluar, mantener y administrar la seguridad de la información.

Las políticas, estándares locales y los procedimientos se encuentran adaptados a las necesidades de la organización debido a que, el proceso mismo de su elaboración integra mecanismos de control y, por último, la certificación permite a las organizaciones, demostrar el estado de la seguridad de la información.

En cada una de las áreas, se establecen una serie de controles que serán seleccionados dependiendo de los resultados obtenidos en los análisis de riesgos, además, existen controles obligatorios para toda organización, como es el de las políticas de seguridad, cuyo número dependerá, más de la organización que del estándar, el cual no establece este nivel de detalle.

Se tomó en cuenta en la presente investigación la norma ISO 17 799, puesto que, es un estándar que está estrechamente relacionado con la seguridad de los SI y que en particular establece que, la organización debe disponer de procedimientos, que garanticen la calidad y seguridad de los sistemas desarrollados. Después del análisis realizado a partir de los pasos que propone y tiene en cuenta esta

norma, se determinó no utilizarla para evaluar o validar la solución propuesta, porque en realidad, los aspectos que contempla, esta norma van más allá del marco de la seguridad de la información, pues se enfoca, mayormente, en la seguridad que a nivel organizacional.

La norma ISO 17 799 deja claro que, toda organización que haga uso de las TI debe implementar buenas prácticas de seguridad, pues en muchas ocasiones, el no seguir un proceso de implementación adecuado como el que ella establece, puede generar huecos de seguridad; por la misma complejidad de las organizaciones, y en ese sentido aumenta la posibilidad de riesgos. De manera que, se pueden establecer puntos de contacto, entre los elementos que establece la norma, como necesarios para garantizar la seguridad y los que se tuvieron en cuenta en la propuesta de solución.

Finalmente se puede decir que, aunque es un estándar teórico se lee como un sistema de recomendaciones. Contornea medidas de seguridad que las organizaciones deben tener, pero no especifica cómo ponerlas en ejecución. Fija simplemente la expectativa y los procesos para proteger la información contra aberturas, uso erróneo y abuso, tanto interno como externo. Su propósito es establecer una serie de lineamientos que, cualquier empresa, pueda seguir para construir su arquitectura de seguridad.

3.3 Evaluación por criterio de especialistas

Después de realizar un estudio sobre los dos estándares internacionales encontrados, que sirven para evaluar la seguridad de los sistemas de información, se concluyó que ninguno de los estándares analizados satisfacía totalmente los intereses perseguidos para realizar la evaluación del procedimiento a partir de los elementos encontrados, que fueron mencionados en el epígrafe anterior.

El *Common Criteria* y la ISO 17 799, plantean una serie de aspectos a tener en cuenta para realizar la evaluación, pero a nivel organizacional; de forma que establece los puntos que no se deben violar, si un sistema es seguro. Sin embargo, lo que se persigue es la evaluación de la solución planteada, por lo que estos estándares no ofrecen lo elementos necesarios para realizar esta tarea.

De esta forma se decidió aplicar la técnica de evaluación por criterios de especialistas. Esta técnica contempla: la selección de un grupo de especialistas en una materia determinada, a los que se les presenta una panorámica de la investigación y la solución que se propone, de tal manera que se la puedan estudiar y emitan sus criterios y consideraciones al respecto. Se dejó como constancia cada

aval *(Ver anexo 3)* firmado por dichos especialistas, quedando de esta forma conformada la evaluación. Esta es una técnica adaptada a las condiciones particulares del entorno en que se enmarca la investigación, pero no deja de ser una fuente que arroja muy buenos y enriquecedores resultados.

De esta manera, se realizó la selección de cinco especialistas dentro de la Universidad de las Ciencias Informáticas (UCI); de ellos dos Ingenieros Informáticos pertenecientes al proyecto RN, un Ingeniero Informático especialista de la Dirección Técnica de la UCI, un Ingeniero Informático perteneciente al proyecto Identidad y un Ingeniero Informático, con años de experiencia, perteneciente al proyecto Aduanas.

De los criterios emitidos por los especialistas, se extrajeron muchos aspectos importantes en relación con la solución propuesta; pues ellos resaltaron aspectos positivos y relevantes de la investigación y, además, realizaron algunas recomendaciones.

Dentro de las recomendaciones emitidas por estos especialistas se encuentra: la propuesta de tener presente las cualidades de mantenibilidad y flexibilidad del software, refiriéndose de forma general, al procedimiento de configuración de los permisos como una tarea que resulta realmente bastante difícil de mantener, cuando el flujo de tablas, vistas y procedimientos almacenados, crecen; el especialista consideró, que se debe velar por los aspectos mencionados con antelación y buscar un balance con respecto a la seguridad, ya que no siempre, es deseable sacrificar flexibilidad y mantenibilidad en las aplicaciones.

Otro de los especialistas realizó algunos señalamientos sobre ciertos elementos, donde él consideraba necesario dar más claridad y generalidad al tema que se investigó, por ser aspectos fundamentales cuando se habla de seguridad dentro de una base de datos.

Uno de los especialistas realizó recomendaciones, referentes a la utilización de herramientas y funcionalidades que proporciona puntualmente el SGBD Oracle; las que se analizaron, pero no fueron incluidas, porque la solución está encaminada a ser utilizada independientemente del SGBD conque se trabaje. De manera que, cada uno de los términos abordados, no están vinculados directamente a un tipo de gestor específico, sino se brinda una solución flexible y configurable, capaz de adaptarse a las necesidades de cualquier negocio y que se puede implementar sobre cualquier SGBD.

De manera general, los especialistas resaltaron los siguientes aspectos positivos, a partir de la solución propuesta:

"Acerca del trabajo presentado, se considera que representa una guía muy detallada e importante para los diseñadores de bases de datos. El trabajo tiene gran valor teórico-práctico y, se considera, que tiene muy alta calidad desde el punto de vista científico-técnico, pues abarca los puntos más importantes a tener en cuenta, para obtener una base de datos segura y confiable, como son: la seguridad en la base de datos, el estricto control de los usuarios y sus permisos, la integridad referencial de la base de datos y otros no menos importantes" *(Ver anexo 3)*.

"El trabajo presenta un elevado nivel científico acorde con el estado del arte y, su creadora esboza, con mucha claridad el problema. Nos brinda una solución de alto valor intelectual que puede ser puesta en práctica con pocos esfuerzos en nuestros proyectos productivos" *(Ver anexo 3)*.

"Es un material que mejora, en gran medida, los mecanismos de diseño e implementación de las bases de datos, pues reúne un gran número de elementos necesarios para lograr establecer, en cierta medida, un nivel de seguridad de la información que se almacena" *(Ver anexo 3)*.

"Es un aporte interesante, pues en él se encuentra reunido un gran volumen de información relacionada con la seguridad de la base de datos, como son: la integridad, la seguridad, la trazabilidad, la salvaguarda y recuperación de la información, aspectos que no aparecen recogidos en ningún otro material, ni con el grado de precisión que éste ofrece" *(Ver anexo 3).*

"En manos de los diseñadores y administradores de las bases de datos, este material fortalecerá el diseño, disminuirá el esfuerzo que se realiza para garantizar la seguridad y contribuirá a su experiencia" *(Ver anexo 3)*.

"La propuesta, recoge los elementos fundamentales, donde la mayoría de los diseñadores y desarrolladores de bases de datos, comenten errores" *(Ver anexo 3)*.

"Se plantean recomendaciones generales de diseño como: restricciones de dominio, limitación en el uso de nulos, uso de llaves; el uso de índice para aumentar el rendimiento de las consultas y se establecen fases, para la definición de procedimientos que concuerden con el estado del arte en la temática. De manera general consideramos que el procedimiento es factible y que debería aplicarse en condiciones reales para valorarlo en la práctica" *(Ver anexo 3)*.

Todo lo expuesto con anterioridad fueron, de manera general, las opiniones ofrecidas por los especialistas, que en su mayoría felicitaron la propuesta y la consideraron un aporte importante y novedoso, que agrupa los elementos principales a tener en cuenta cuando se persigue asegurar la base de datos; porque reúne y organiza gran cantidad de información, que sólo se adquiere con la experiencia, con el paso de los años, o a partir de artículos y recomendaciones que se encuentran en la Web.

3.4 Análisis y discusión de los resultados

En la actualidad es una realidad tangible, la imperiosa necesidad de garantizar la seguridad dentro de las bases de datos, ya que la información se ha convertido en uno de los activos más importantes de cualquier organización o empresa; de tal manera que es importante tener mecanismos que aseguren los datos almacenados, sean críticos o no, pues en dependencia de las particularidades de cada negocio, lo común es que en cualquier situación, es vital mantener aseguradas la confidencialidad, la integridad y la disponibilidad de la información.

Mediante el procedimiento propuesto, se pudieran llegar a obtener diseños de bases de datos robustos y seguros, garantizando mecanismos de seguridad en la implementación, independientemente del SGBD conque se trabaje; contribuyendo, de esta forma, a la calidad de los productos de software desarrollados.

Según los criterios de varios especialistas que evaluaron la solución; es una contribución muy atractiva, ya que es capaz de reunir gran cantidad de información relacionada con la seguridad de las bases de datos. Es un aporte importante, ya que a pesar de recoger ciertos temas que por experiencia personal se conocen, o que han sido estudiados por varios especialistas durante todos sus años de trabajo, no existe un solo documento en el que se ofrezcan tantos elementos y de manera tan clara y precisa, con un nivel de flexibilidad en cada uno de los aspectos propuestos.

Los elementos que se tuvieron en cuenta en la propuesta son la integridad, la seguridad, la trazabilidad, la salvaguarda y recuperación de la información. Cada uno de ellos constituye un aspecto fundamental si se persigue asegurar la información. Alrededor de ellos se definieron pasos que están encaminados a orientar a los diseñadores y administradores de bases de datos; quienes con este material es sus manos deberán esforzarse menos cuando quieran garantizar la seguridad dentro de la base de datos, pues esta guía permite adaptarse a cada situación en particular y les aportará nuevos elementos a su conocimiento y experiencia profesional.

A partir de recomendaciones ofrecidas por los especialistas, la autora del presente trabajo, consideró necesario reelaborar y ser más específica aún, sobre el tema de la estrategia y salvaguarda de los datos, conforme a que en la forma en que se presentó inicialmente, podía ser interpretado con otra visión, quedando de esta forma claro, para el lector, la verdadera intención que se perseguía al abordarse el tema que se trabajó.

Se tuvo muy en cuenta los sabios consejos aportados por los especialistas, dada su experticia y experiencia en el tema tratado. El planteamiento de uno de ellos consideró, que en alguna medida se descuidaron otros elementos importantes como son la mantenibilidad y la flexibilidad. No obstante, señala la autora, como lo ha hecho en reiteradas ocasiones a lo largo de la investigación, que la misma estuvo orientada a garantizar la seguridad de la información que se almacena en las bases de datos, objetivo principal de la investigación y que fue cumplimentado. Sin embargo teniendo en cuenta su criterio, que menciona aspectos relacionados con la calidad del software, la autora considera muy importante tomarlo en cuenta, para imprimirle un toque refinado a la solución, de manera que no se cambian unas por otras, sino que se llevan todas a la par.

Finalmente el objetivo principal de la investigación, el que se quería garantizar con la propuesta de solución y que era en cierto grado, la seguridad de las bases de datos, fue alcanzado, pues el procedimiento planteado reúne los aspectos fundamentales a tener en cuenta, para garantizar los tres pilares fundamentales de la seguridad de los sistemas de información.

Conclusiones

El presente capítulo, es una referencia de las características fundamentales que se identifican en cada uno de los estándares internacionales encontrados para la evaluación de los sistemas de información. Referente al *Common Critera*, se demostró el alto valor que tiene una evaluación de este tipo, para cualquier producto desde el punto de vista de los usuarios finales. Se realizó el mismo análisis de la ISO 17 799. Finalmente, después de estudiados cada uno de ellos, se expuso los elementos por lo que no servían para realizar la evaluación de la solución propuesta.

Se realizó una explicación de la técnica de evaluación utilizada. A partir de los criterios emitidos por varios especialistas en el tema de las bases de datos, se determinó que el trabajo realizado cumplió con el objetivo previsto y, además, se ofrecerán recomendaciones, que posteriormente quedarán reflejadas. Finalmente se realizó el análisis y discusión de los resultados, a partir de los criterios emitidos en cada uno de los avales proporcionados por los especialistas.

CONCLUSIONES

- Se realizó un estudio del estado del arte de los temas fundamentales, relacionados con la investigación, para lograr crear una fundamentación de temas teóricos referentes a las bases de datos.
- Se estudiaron los diferentes modelos de diseño de las bases de datos, haciendo especial énfasis, en el estudio del modelo de datos relacional, a partir del que se implementó la propuesta.
- Se estudiaron las diferentes herramientas CASE para el diseño de las bases de datos, fundamentalmente las más usadas internacionalmente para el desarrollo de software. Se puntualizaron lo aspectos más significativos del Edwin, por ser la herramienta con que se trabajó.
- Se estudiaron las diferentes SGBD que existen, clasificándolos de acuerdo a su accesibilidad, en libres y propietarios. Se especificaron los más usados en el mundo. Se realizó una explicación más detallada de la suite de productos de Oracle, por ser el gestor con el que se trabaja, en el proyecto Registro y Notarias.
- Se elaboró un procedimiento, que fue el que dio solución al problema de esta investigación. Se conformó una propuesta que recoge los elementos fundamentales, donde la mayoría de los diseñadores y desarrolladores de bases de datos, comenten errores.
- Se elaboró un procedimiento que reúne un gran volumen de información relacionada con la seguridad de las bases de datos, como la integridad, la seguridad, la trazabilidad, la salvaguarda y recuperación de la información, aspectos que no aparecen recogidos en ningún otro material, ni han sido explicados con el grado de precisión, que en esta investigación se ofrece.
- Este trabajo en manos de los diseñadores y administradores de las bases de datos, constituye un material que fortalecerá el diseño y disminuirá el esfuerzo realizado para garantizar la seguridad, además de contribuir con la experiencia de cada persona que lo utilice.
- Se realizó un estudio sobre los diferentes estándares para la evaluación de los sistemas de información, encontrando el *Common Criteria* y la ISO 17 799. Se realizó el análisis necesario para poder arribar a conclusiones sobre cada uno de ellos.

- Se efectuó la evaluación del procedimiento a partir del criterio de varios especialistas de la UCI, que fueron seleccionados de acuerdo a su experiencia en el tema de las bases de datos.
- A partir de las consideraciones arrojadas por cada uno de los especialistas entrevistados, se realizó el análisis y discusión de los resultados. De manera general se discutieron aspectos relacionados con la investigación.

Finalmente, se consideró que todos los temas abordados a lo largo de la presente investigación pueden contribuir al desarrollo futuro de bases de datos seguras, que se realicen en la Universidad de las Ciencias Informáticas independientemente del SGBD utilizado, ya que la propuesta fue elaborada de forma flexible y configurable. Además se pretende que el documento, constituya una referencia para el diseño e implementación de cualquier base de datos, principalmente para el proyecto Registro y Notarias, que es el negocio a partir del cual se diseñó la investigación.

RECOMENDACIONES

- Realizar un estudio sobre cómo usar tecnología *Storage Area Network* (SAN[14]) y *Network Attached Storage* (NAS[15]), para aumentar el valor de la solución en una posible versión futura.
- Realizar un estudio sobre otras herramientas que contribuyan con el proceso de salvaguarda, sobre todo si éstas son consideradas como software libre.
- Tener en cuenta para refinar la solución, otros temas como la mantenibilidad y la flexibilidad, de forma tal que no se deteriore la calidad del software por garantizar aspectos de la seguridad.
- Que se realice un estudio sobre los diferentes estándares internacionales para la evaluación de los sistemas de información, a fin de que la universidad pueda utilizarlos, para evaluar la seguridad de sus productos y dar una mayor garantía a los usuarios.
- Dar continuidad a este procedimiento, de manera que continúe enriqueciéndose con las experiencias de los especialistas.

[14] Una red SAN se distingue de otros modos de almacenamiento en red por el modo de acceso a bajo nivel. El tipo de tráfico en una SAN es muy similar al de los discos duros como ATA, SATA y SCSI. Esta estructura funciona principalmente en computadoras de Microsoft.

[15] Generalmente, los sistemas NAS son dispositivos de almacenamiento específicos a los que se accede desde los equipos a través de protocolos de red (normalmente TCP/IP). También se podría considerar que un servidor Windows que comparte sus unidades por red es un sistema NAS, pero la definición suele aplicarse a sistemas específicos.

REFERENCIAS Y BIBLIOGRAFÍA

1. **Fernández Arencibia, Daniel and Fábregas Santos, Yunieski.** *Administracion, configuracion y optimizacion de un sistema de Bases de Datos Desentralizado en Oracle Database 10g release 2.* Cuidad de la Habana : UCI, 2007.

2. *Bases de datos documentales: estructura y uso.* **Yunta, luis Rodríguez.** s.l. : La informacion especializada en Intertet, 2001.

3. **Meléndez, Raquel, Laurens, Yenifer and Betancourt, Dayan.** *Introduccion a los sistemas.* Carabobo : Universidad de Carabobo. Facultad esperimental de Ciencia y Tecnología, 2007.

4. *Bases de Datos.* **Ballester, Jose Luis Domenech.** 2007.

5. **Ortiz, Antonio Moreno.** *Diseño e implementacion de un lexico computacional para la lexicografía y traducción automática.* Málaga : s.n., 2000.

6. *Modelo de datos.* **Andréz, María Mercedes Márquez.** 2001.

7. **Computacion, Departamento de Sistemas Informaticos y.** *Introduccion a Herramientas CASE y System Architect.* Valencia : Metodología y Tecnología de la Programacion.

8.**Terry, B and Logee, D.** *Terminology for Sftware Engineering and Computer-aided Software Engineering.* s.l. : Software Engineering Notes, 1990.

9. **McClure, Carma.** *The CASE Experience .* s.l. : BYTE, 1989.

10. **Informática, Instituto Nacional de Estadística.** *Herramienta Case.* 1999.

11. **Bertino, Elisa.** *Traduccion de Object-oriented database system: concepts and architectures.* s.l. : Diaz de Santos, 1995.

12. **KORTH.** 1995.

13. **Microsoft, SQL Server de.** MSDN. [Online] 2008..

14. **Ojeda, Francisco Charte.** SQL Server 2000. [Online] 2001.

15. **MySQL.** MySQL. [Online] [Cited: Febrero 20, 2008.] http://dev.mysql.com/.

16. **Worsley, John and Drake, Joshua.** *Utiles PostgreSQL.* s.l. : Commandprompt, Inc, 2001-2002.

17. *PostgreSQL affiliates .ORG domain.* **Cameron, Nadia.** s.l. : COMPUTERWORLD, 2003.

18. *Oracle presenta Database Lite 10g.* **Corporation, Oracle.** s.l. : MasterMagazine, 2005.

19. **10g, Oracle Database.** ORACLE DATABASE 10G ENTERPRISE EDITION. 2004.

20. **Corporation, Oracle.** Oracle. [Online] [Cited: Febrero 19, 2008.] http://www.oracle.com

21. *Oracle Application Server 10g arquitectura y administración.* **Garmany, John and Burleson, Donald K. Oracle Application Server 10g Manual de Administración.** 2004.

22. **ISO/IEC.** 1999.

23. **Dictionary, IEEE Standard Computer.** *A complilation of IEEE Standard Computer Glossaries.* New York : s.n., 1990.

24. **Albet, Elaborado por Ingeniería y Sistemas.** Salvaguarda y Recuperacion. s.l. : Notarias, Registro y, 2007.

25. **Esteban, CALORE and ROLDAN, Natalia.** *Criterios de evaluacion de la seguridad en sistemas de informacion.* s.l. : Universidad Nacional del Comahue , 2004.

26. *PostgreSQL affiliates .ORG domain.* **Cameron, Nadia.** s.l. : COMPUTERWORLD, 2003.

27. **Ramirez, Luis, Escobar, Javier y Asprino, Omar.** *Integridad y Seguridad.*

28. *Integridad de datos relacionales.* **Decker, Hendrik.** s.l. : Revista del Instituto Tecnológico de Informática.

29. **Carrasco, Jorge y Varas, Marcela.** *Especificación de Restricciones de Integridad Condicionales en Esquemas Conceptuales de Bases de Datos.* Chile : Departamento de Ingeniería Informática y Ciencias de la Computación, 1999.

30. **Moreno, Iván Tapia.** *Modelo de Datos Relacional. Fundamentos de Base de Datos.*

31. *Administración de Bases de Datos ORACLE.* **(URJC), Basadas en el trabajo original de la Dra. Belén Vela.** Ciudad Real : E.S.Informática – UCLM.

32. SEGURIDAD EN UNIX Y REDES. 2002.

33. **Costa, Dolors Costal.** *Introducción al diseño de bases de datos.*

34. **Ramírez, Rubén José Álvarez.** *Bases de datos seguras.* s.l. : Modelos Avanzados de bases de datos, 2003/2004.

35. **Gómez, J. Galindo.** *Diseño de Bases de Datos: Modelo EER y Normalización.* s.l. : E.T.S.I. Informática.

36. **Fernández-Medina, Dr. Eduardo.** *SEGURIDAD EN EL DISEÑO DE BASES DE DATOS Y SISTEMAS DE INFORMACIÓN.* Castilla : Grupo de Investigación Alarcos.

37. **Jiménez, José Samos.** *Definición de Vistas en Bases de Datos Orientadas a Objetos.* Barcelona : Universitat Politècnica de Catalunya, 1993.

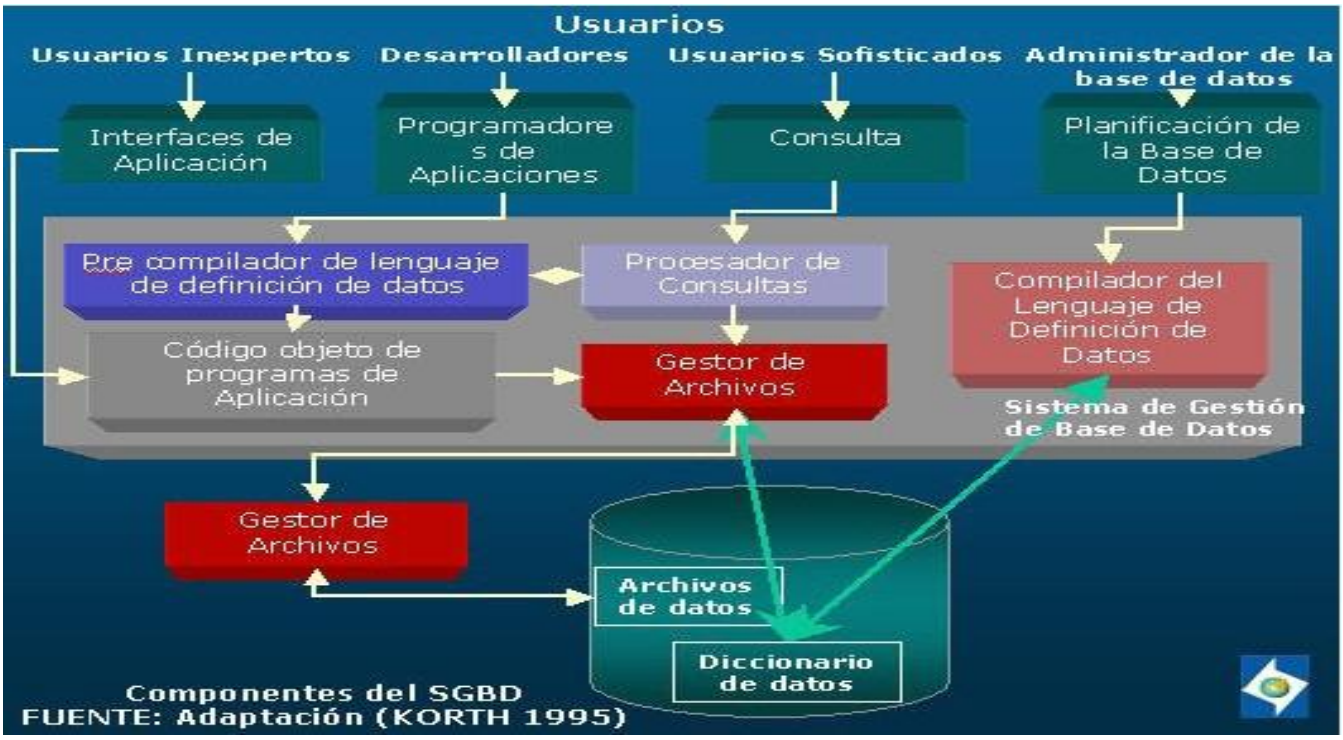
38. **Silva, Josep, Belenguer, Jorge y Celma, Matilde.** *Materialización de vistas Multi-Origen: Vistas Multinivel.* Valencia : s.n.

39. **Mota, Soraya Abad.** *Lineamientos de Diseño Físico.* 2007.

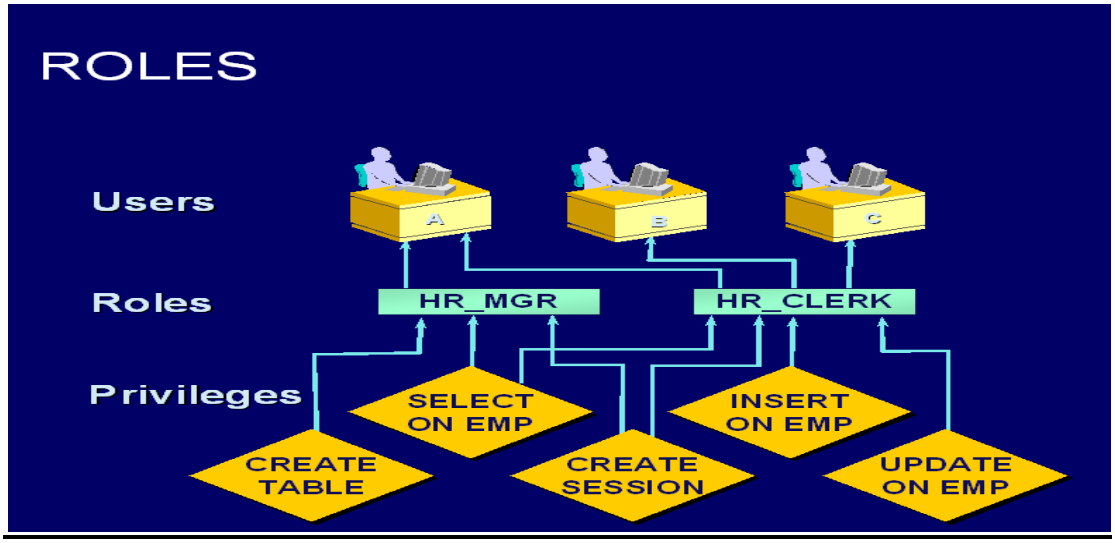
40. **Gómez, J. Galindo.** *Conceptos sobre el SGBD Oracle.* s.l. : E.T.S.I. Informática.

ANEXOS

Anexo 1 Componentes de un Sistema Gestos de Base de Datos



Anexo 2



Anexo 3

Caracas, Venezuela, 31 de Mayo del 2008

Criterio de especialista:

La investigación con titulo: ¨Propuesta de procedimiento para la seguridad de bases de datos durante el diseño¨, del autor Maria Antonia Lajús Marreo aborda temas fundamentales a tener en cuenta en el diseño y desarrollo de bases de datos presentando una guía que puede ser utilizada por analistas en futuros proyectos productivos.

Es un material que mejora en gran medida los mecanismos de diseño e implementación de las bases de datos, pues reúne un gran número de elementos necesarios para lograr establecer, en cierta medida, la seguridad de la información que se almacena.

El trabajo aborda un conjunto de prácticas que se aplican en diferentes proyectos productivos, sintetizándolas en un documento que puede ser utilizado como material de consulta. Esta propuesta, de cierta forma, permitirá crear una base para la estandarización de los procedimientos empleados con el fin de lograr un adecuado nivel de seguridad, tanto en el desarrollo, como en la puesta en marcha de bases de datos.

Teniendo en cuenta lo antes expuesto considero que la propuesta presentada es un documento de un gran valor científico por lo que recomiendo su publicación en eventos investigativos y su aplicación en nuestra Universidad.

Jofman Pérez Tarancón.
Facultad 1.
Universidad de las Ciencias Informáticas

Universidad de las Ciencias Informáticas

Cuidad de la Habana, 30 de Mayo del 2008

Criterio de especialista:

La investigación con titulo: ¨ Propuesta de procedimiento para la seguridad de bases de datos durante el diseño e implementación ¨, del autor Maria Antonia Lajús Marreo aborda uno de los temas fundamentales a tener en cuenta en la creación de una base de datos, el tema de la seguridad. Temática que se debe garantizar para lograr una mayor calidad en las bases de datos que se crean, a partir de la importancia que tiene en la actualidad la información para toda empresa u organización.

El trabajo presenta un elevado nivel científico acorde con el estado del arte, su creador esboza con claridad el problema. Nos brinda una solución de alto valor intelectual que puede ser puesta en práctica con pocos esfuerzos en nuestros proyectos productivos.

Es un material que mejora en gran medida los mecanismos de diseño e implementación de las bases de datos, pues reúne un gran número de elementos necesarios para lograr establecer en cierta medida un nivel de seguridad de la información que se almacena.

Es un aporte interesante, pues en el se encuentra reunida gran numero de información relacionada con la seguridad de la base de datos, como la integridad, la seguridad, la trazabilidad, la salva guarda y recuperación de la información, aspectos que no viene recogidos en ningún otro material, ni con el grado de precisión que este ofrece.

En manos de los diseñadores y administradores de las bases de datos, este material fortalecerá el diseño, disminuirá el esfuerzo realizado para garantizar la seguridad y contribuirá a su experiencia.

Por lo antes expuesto que el trabajo diploma cumple con los objetivos propuestos. Recomiendo su publicación en eventos investigativos y su aplicación práctica en nuestra Universidad.

Ing. Daniel Fernandez Arencibia
Donde pertenece (DPTO, FAC, etc.)
Universidad de las Ciencias Informáticas

Universidad de las Ciencias Informáticas

Cuidad de la Habana, 30 de Mayo del 2008

Criterio de especialista:

La investigación con titulo: "Propuesta de procedimiento para la seguridad de bases de datos durante el diseño e implementación", del autor Maria Antonia Lajús Marreo aborda uno de los temas fundamentales a tener en cuenta en la creación de una base de datos: la seguridad. Temática que se debe garantizar para lograr una mayor calidad en las bases de datos que se crean, a partir de la importancia que tiene en la actualidad la información para toda empresa u organización.

Acerca del trabajo presentado se considera que representa una guía muy detallada e importante para los diseñadores de bases de datos que usen el gestor Oracle. El trabajo tiene gran valor teórico-práctico y se considera que tiene una muy alta calidad desde el punto de vista científico-técnico, pues abarca los puntos más importantes a tener en cuenta para obtener una base de datos segura y confiable, como son: la seguridad en la base de datos, el estricto control de los usuarios y sus permisos, la integridad referencial de la base de datos y otros no menos importantes.

El procedimiento presentado es fruto de una larga y ardua investigación, avalada por los demás miembros del proyecto, dentro de los que se encuentra el autor de esta valoración. Se considera que debe ser extendido a todo el resto de la universidad para que sea de conocimiento de los distintos diseñadores de bases de datos existentes en la misma.

Se recomienda que se reelabore o se especifique más el tema de la estrategia de salva guarda de los datos, debido a que en la forma que se presenta actualmente puede ser interpretada desde varios puntos de vista, no quedando claro al lector la verdadera intención del autor.

Por lo antes expuesto que el trabajo diploma cumple con los objetivos propuestos. Recomiendo su publicación en eventos investigativos y su aplicación práctica en nuestra Universidad.

Ing. Yunieski Fabregas Santos
Facultad 3.
Universidad de las Ciencias Informáticas

GLOSARIO

Aplicaciones de Escritorio: Aplicaciones que se ejecutan en un escritorio (o portátil) informático en contraste con la web de aplicaciones basadas en Internet. El software que simula un objeto que normalmente se encuentran en una oficina de escritorio, como una calculadora, bloc de notas y el nombramiento de calendario.

Auto Numérico: es un término que se refiere a la utilización de números secuenciales exclusivos (con incremento de una unidad).

Base de Datos: recopilación de datos que puede organizarse de forma que pueda sus contenidos puedan accederse, gestionarse y actualizarse fácilmente. Representan aspectos del mundo real.

Sistemas de Información: conjunto de ficheros automatizados, programas, soportes y equipos empleados para el almacenamiento y tratamiento de datos de carácter personal.

Bits: es el acrónimo de *Binary digit*. (Dígito binario). Un bit es un dígito del sistema de numeración binario.

Dominio: Límite que se establece alrededor de los objetos creados dentro del mismo ámbito de aplicación (es decir, cualquier lugar de la secuencia de activaciones de objetos que empieza en el punto de entrada de la aplicación). Los dominios ayudan a aislar los objetos creados en una aplicación de los creados en otras aplicaciones, de forma que se pueda predecir el comportamiento en tiempo de ejecución. En un único proceso pueden existir varios dominios.

Campo: El término campo es frecuentemente intercambiable con el de columna, aunque muchos consideran más correcto usar el término campo (o valor de campo) para referirse específicamente al simple elemento que existe en la intersección entre una fila y una columna.

Interfaz: La interfaz de usuario es la forma en que los usuarios pueden comunicarse con una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.

Llave Foránea: conjunto de atributos común a dos entidades que sirve como relación entre las dos entidades. No es un atributo de la entidad relacionada, pero es la llave primaria de la entidad con la cual ésta se relaciona.

Llave Primaria: Conjunto de atributos que distingue cada ocurrencia de una entidad de forma inequívoca a las demás. Es una llave con valores únicos, es decir, no ocurren más de una vez en el atributo. Puede ser un atributo o una combinación de atributos.

Mínimo Privilegio: Una estrategia de defensa en profundidad, con capas de seguridad superpuestas, es la mejor manera de contrarrestar las amenazas de software malintencionadas. Consiste en limitar el acceso de los usuarios sobre determinados objetos de la base, de manera que solo tengan visibilidad sobre los elementos básicos para trabajar en una determinada área.

Persistencia: es la misma acción de almacenar información en la base de datos.

Repositorio: un repositorio, depósito o archivo es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.

Repositorio Central: un sitio centralizado donde se decide que estará almacena toda información digital, generalmente bases de datos.

Sistema Gestor de Bass de Datos: Herramienta Software que proporciona una interfaz entre los datos almacenados y los programas de aplicación que acceden a éstos y que se caracteriza fundamentalmente por permitir una descripción centralizada de los datos y por la posibilidad de definir vistas parciales de los mismos para los diferentes usuarios.

Sistemas Operativos (SO): es el software que viene con el ordenador antes de que se instale ninguna aplicación. Según esta definición, orientada al usuario final, las herramientas de desarrollo no son necesarias y, sin embargo, se consideran elementos imprescindibles un amplio número de programas orientados a diferentes tareas, como editores de texto, administradores de archivos, navegadores, etc. Debe ser de fácil uso y acceso y permitir además múltiples procesos y usuarios.

Suite: son un conjunto de productos de software, pertenecientes a una sola empresa y orientados a un tema específico, capaces de ofrecer una flexibilidad sin precedentes

Tecnología Firewall: es un sistema diseñado para prevenir acceso no autorizado hacia o desde una red privada. Provee un punto de defensa entre dos redes – protege una red de otra. Un firewall puede ser tan simple como un ruteador que filtra paquetes o tan complejo como varias computadoras o varios ruteadores que combinan el filtrado de paquetes con servicios proxy a nivel aplicación. Puede ser una poderosa herramienta para utilizarla como barrera básica de acceso.

Tecnologías de Información: son un conjunto de servicios, redes, software, aparatos que tienen como fin el mejoramiento de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario.

Tuplas: es la representación de una fila, en una de las tablas que se esta almacenando datos. Y las cuales serán utilizadas en el tiempo de ejecución de un sistema.