Universidad
Carlos III de Madrid

Escuela Politécnica Superior

Ingeniería Técnica en Informática de Gestión

LAS BASES DE DATOS, SU SEGURIDAD Y AUDITORÍA. EL CASO DE MYSQL

Leganés, 2011

AUTOR: Jessica Pérez Sandoval
TUTOR: Miguel Ángel Ramos González

PROYECTO DE FIN CARRERA

Las bases de datos, su seguridad y auditoría.
El caso de MySQL

Autor: Jessica Pérez Sandoval
Tutor: Miguel Ángel Ramos González

EL TRIBUNAL

Presidente: Ana Isabel González-Tablas Ferreres

Vocal: Harith Aljumaily

Secretario: Antonio García Carmona

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 8 de abril de 2011 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

A la memoria de mis abuelos

A la pequeña Sara,
porque algún día pueda sentirse orgullosa de mí

Agradecimientos

Me gustaría que estas líneas sirvieran para expresar mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado de alguna forma en la realización del presente trabajo.

En primer lugar quisiera agradecer a Miguel Ángel Ramos González, tutor de este proyecto, su profesionalidad, orientación, disponibilidad, seguimiento y buen trato recibido. Sin su apoyo este trabajo nunca hubiera sido posible.

A mis padres, por animarme, por creer en mí y por apoyarme siempre, no sólo en este proyecto, sino en todas las áreas de mi vida. A mi hermana Vanesa, por ser la mejor amiga que puedo tener. Por sus reprimendas y sus ánimos, por saber en todo momento qué es lo que necesitaba para seguir adelante. A Eduardo, por enseñarme a que nunca hay que rendirse y que con esfuerzo todo se consigue.

Gracias a ellos tengo la certeza de que las buenas personas existen, y son para mí el mejor ejemplo a seguir.

A mis compañeros de trabajo: José Manuel, Vanesa y María Jesús. Jamás pensé que tendría la suerte de trabajar con personas tan estupendas. A D. Juan Carlos Falcón por ofrecerme la posibilidad de compaginar estudios y trabajo, por su flexibilidad y comprensión.

A mis compañeros de universidad: por los momentos vividos, por las experiencias compartidas, por las lecciones aprendidas.

A la Universidad, porque en ella he vivido una de las experiencias más gratificantes y enriquecedoras. Y a los profesores: los de verdad, los que desean enseñar y lo hacen con todo su empeño.

A todos ellos, **GRACIAS**.

Índice general

BLOQUE II: LAS BASES DE DATOS Y SGBD

BLOQUE III: LA AUDITORÍA

BLOQUE IV: AUDITORÍA DE MySQL

Índice de figuras

Resumen

El presente Proyecto Fin de Carrera, perteneciente a la titulación de Ingeniería Técnica en Informática de Gestión de la Universidad Carlos III de Madrid, tiene como finalidad el estudio de la Auditoría de la seguridad en un Sistema Gestor de Base de Datos, siendo el caso concreto MySQL.

El auditor debe tener la capacidad y los conocimientos necesarios para revisar y evaluar el control interno del entorno en que se desarrolla la base de datos, capacidad para revisar riesgos y controles, evaluar y recomendar mejoras, etc. Como punto importante, la Información es un activo clave en toda organización, por ello, la labor del auditor es de suma importancia.

El estudio de la auditoría de la seguridad de MySQL comienza desde el conocimiento general del auditor tanto de la seguridad (políticas, estándares, normas…) como de las bases de datos, y más concretamente de la base de datos MySQL. Finalmente, se aborda la auditoría de un aspecto concreto de MySQL: su seguridad.

Abstract

This project, part of the Technical Engineering Degree in Computer Science by Carlos III University of Madrid, aims the study of the security and audit of Database Management Systems, being the case MySQL.

Auditors must have the ability and knowledge to review and evaluate the internal control of the database environment, the ability to review risks and controls, evaluate and recommend improvements, etc. As important point, the information is a key asset in any organization, therefore the auditor's work is of paramount importance.

The study of the security audit of MySQL starting from the auditor general knowledge of security (policies, standards, rules...) as databases, and more specifically the MySQL database. Finally, the audit tackle a specific aspect of MySQL: its security.

Introducción

El presente apartado de introducción consta de cinco apartados.

- El primero expone los motivos que justifican la realización de este PFC.
- El segundo recoge los objetivos a conseguir.
- En el tercer apartado se presenta en detalle la estructura de la memoria.
- El cuarto presenta un glosario de los términos utilizados.
- Dado que en este PFC se han empleado tecnologías de libre distribución, en el último apartado se hará una reflexión sobre el concepto de software libre.

1. Motivación

Actualmente la Base de Datos es el sistema de almacenamiento y gestión de la información más extendido y utilizado por las empresas, no sólo por la reducción del tiempo de proceso y por la mejora en actualización, sino también por su mejora en los costes que su implantación supone con respecto a otros sistemas.

2. Objetivos

Este Proyecto de Fin de Carrera, intentará servir como pequeña ayuda a la Auditoría genérica de la seguridad de los Sistemas Gestores de Bases de Datos, utilizando como ejemplo MySQL, así como el acercamiento de su estructura y algunas características particulares.

Con respecto a la cantidad de información que existe referente al SGBD MySQL: en relación a otros SGBD utilizados de una manera más generalizada (como Oracle, DB2 Universal Database de IBM, SQL SERVER…) hay una falta de información.

Se intentará explicar la necesidad de realizar auditorías, especialmente sobre la seguridad. Las necesidades, las utilidades y los objetivos éstas, así como resaltar políticas y estándares que ayuden a la práctica de dicha auditoría; orientado todo ello a las bases de datos.

3. Glosario

- **ACL:** Listas de Control de Acceso
- **AEPD:** Agencia Española de Protección de Datos
- **ANSI:** *American Nacional Standards Institute*
- **Antivirus**: Programas desafuero que se ejecutan en los PC para evitar y tratar las infecciones por virus.
- **Archivo:** Son un conjunto de datos que han sido "codificados" para ser manipulados por un ordenador. Usualmente los archivos tienen una 'extensión' después de un punto, que indica el tipo de data que contiene el archivo. También conocidos como "*file*".
- **Archivo Log:** Un *log* es un registro oficial de eventos durante un rango de tiempo en particular. Para los profesionales en seguridad informática, es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué (*who, what, when, where* y *why*) un evento ocurre para un dispositivo en particular o aplicación.
- **ASCII (*American Standard Code for Information Interchange*):** Código americano estándar para el intercambio de información. Éste código representa cada una de las teclas del teclado con un número binario de siete dígitos.
- **Backup:** Copia de seguridad.

- **Base de Datos:** Una **base de datos** o **banco de datos** (en ocasiones abreviada BB.DD.) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Es una colección estructurada de datos.

- **Bit**: Abreviatura en inglés de los términos en "*binary unit*" (unidad binaria); se trata de un solo dígito de información, que es un 0 o un 1.

- **Bloqueador de elementos emergentes**: Programa de software que se ejecuta en los PC para evitar la aparición de elementos emergentes.

- **Bloqueador de software espía y publicitario** : Programa de software que se ejecuta en los ordenadores y evita la instalación de software espía y publicitario en el equipo.

- **Byte**: Un "fragmento" estándar de información de red o de lenguaje informático. Un byte consta de 8 bits.

- **Caballo de Troya** : Programa oculto de otro programa (el portador) por un programa informático. Cuando el programa portador se abre, se inicia el programa oculto, proporcionando al pirata informático el control del equipo infectado o enviándole información personal del usuario del mismo.

- **Cable**: Puede hacer referencia a un cable con conectores para unir dos dispositivos o hacer referencia al tipo de servicio de banda ancha ofrecido por un proveedor de estos servicios.

- **Checklist**: Lista de comprobación

- **CISA**: Auditor certificado en Sistemas de Información de ISACA (*Certified Information Systems Auditor*)

- **Cliente DHCP:** Equipo que solicita el uso de una dirección IP.

- **COBIT:** *Control Objectives for Information and related Technology.*

- **Cookies:** Un pequeño archivo de texto colocado en un PC por un sitio Web para registrar la información del usuario y preferencias de configuración personal.

- **Correo electrónico:** Aplicación que se utiliza para intercambiar notas y archivos entre dos o más usuarios. La dirección de correo electrónico se identifica mediante el nombre de usuario y el proveedor de servicio. También conocido por E-Mail.

- **Cortafuegos:** Dispositivo físico o programa software que evita el acceso no deseado a redes privadas desde una ubicación externa. También se le conoce como *firewall.*

- **CPU (*Central Processing Unit):*** Unidad central de procesamiento. Es el "cerebro" del equipo, donde se llevan a cabo todos los cálculos.

- **DBA:** Administrador de Base de Datos.

- **DNS (*Domain Name System*):** es un sistema de nomenclatura jerárquica para ordenadores, servicios o cualquier recurso conectado a Internet o a una red privada. Este sistema asocia información variada con nombres de dominios asignado a cada uno de los participantes. Su función más importante, es traducir (resolver) nombres inteligibles para los humanos en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

- **ER:** Entidad/Relación.

- **Hacker:** Persona que entra ilegalmente en redes o equipos o que crea un software de virus. También conocido como "pirata informático".

- **Hardware:** Son los dispositivos físicos que conforman un ordenador: como la placa base, la CPU o el monitor.

- **Internet:** Es una combinación de hardware (ordenadores interconectados por vía telefónica o digital) y software (protocolos y lenguajes que hacen que todo funcione). Es una infraestructura de redes a escala mundial (grandes redes principales (tales como **MILNET**, **NSFNET**, y **CREN**), y redes más pequeñas que conectan con ellas) que conecta a la vez a todos los tipos de ordenadores.

- **ISACA:** Asociación de Auditoría y Control de Sistemas de Información.

- **ISO:** *International Organization for Standardization*. Organización Internacional para la estandarización.

- **LOG**: Registro cronológico de anotaciones.

- **LOPD**: Ley Orgánica de Protección de Datos de carácter personal

- **Servidor:** En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al equipo físico (ordenador) en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

- **Sistema de información:** Es un conjunto de elementos ordenadamente relacionados entre sí de acuerdo a ciertas reglas, que aportan a la organización a la que sirven información necesaria para el cumplimiento de sus fines.

- **Software:** Es el conjunto intangible de datos y programas del ordenador.

- **SQL:** El lenguaje estandarizado para pedir información de una base de datos. La versión original (llamada SEQUEL, por *Structured English Query Language*) (Lenguaje de Interrogaciones Estructuradas en Inglés) fue diseñada en un centro de investigación de IBM en 1974 y 1975.

- **SSH:** Desarrollada por SSH Communications Security Ltd., Shell Segura (*Secure Shell*) es un programa utilizado para conectarse en otro equipo en la red, para ejecutar órdenes en una máquina remota, y para mover archivos de una máquina a otra. Provee eficiente autenticidad y seguridad para las comunicaciones sobre canales que normalmente son inseguros. SSH protege una red de ataques como falsificación de IP, ruta de origen IP, y falsificación de DNS. Un intruso (atacante) que logra entrar en la red solo puede forzar SSH a que se desconecte, pero no puede controlar el tráfico o desviar la conexión cuando se habilita la codificación.

4. Estructura del PFC

Este trabajo se ha dividido en cuatro grandes bloques y una introducción inicial.

- En la introducción inicial se presentan los objetivos principales del PFC, el contenido de la memoria, un glosario de términos, la organización y estructura del PFC y una pequeña introducción al *Open Source* o Código Abierto.

- En el primer bloque se estudia la seguridad, en concreto la seguridad informática, los conceptos, los elementos que la componen, amenazas y tipos de seguridad, y algunas de las medidas para prevención y control, así como algunas ayudas a este control.

- En el segundo bloque -que a su vez se divide en dos apartados- se explica en el primero de sus apartados el concepto de "Base de Datos", las ventajas e inconvenientes de su implantación, su administración, los Sistemas Gestores de Bases de Datos y sus características principales, así como lenguajes y estructura. En el segundo apartado se hace una introducción al SGBD MySQL y sus características particulares, sus tipos de tablas, y cómo es su almacenamiento de datos. Asimismo, se explica la seguridad y administración en ésta.

- En el tercer bloque se estudia la auditoría y su concepto. El método auditor, así como sus características, alcance y objetivos.

- En el último bloque se estudia la auditoría de la seguridad de MySQL, estudiando las amenazas lógicas y físicas, copias de seguridad, etc.

5. Código abierto y Software Libre (*Open Source and Free software*)

Como se comentará a lo largo del presente proyecto*, MySQL es uno de los SGBD de código abierto más extendido actualmente, por ello es importante comprender, qué es realmente el código abierto. En este apartado se estudiará con más detalle esta característica.

El Software de *Open Source* exige la distribución libre y gratuita acompañada del código fuente. Código abierto (*open source* en inglés) es el término por el que se le conoce al software distribuido y desarrollado en una determinada forma. Este término empezó a utilizarse en 1998 por usuarios de la comunidad del software libre, tratando de usarlo como reemplazo del nombre original del software libre (*free software*) que era ambiguo.

En inglés, "*free software*" puede significar diferentes cosas. Por un lado, permite pensar en "software por el que no hay que pagar", y se adapta al término de forma igualmente válida que el significado que se pretende (software que posee ciertas libertades).

Sin embargo, el término no resultó apropiado como reemplazo para el ya tradicional *free software*, y en la actualidad es utilizado para definir un movimiento nuevo de software, diferente al movimiento del software libre, aunque no completamente incompatible con éste, de modo que es posible (como de hecho ocurre) que ambos movimientos trabajen juntos en el desarrollo práctico de proyectos.

** Para versiones 5.1 y anteriores*

El significado obvio del término “código abierto” es “se puede mirar el código fuente”, lo cual es un criterio más débil y flexible que el del software libre; un programa de código abierto puede ser software libre, pero también puede serlo un programa semi-libre o incluso uno completamente propietario.

El software de código abierto (OSS por sus siglas en inglés) es software para el que su código fuente está disponible públicamente, aunque los términos de licenciamiento específicos varían respecto a lo que se puede hacer con ese código fuente.

- Los Beneficios del *Open Source*

La clara ventaja monetaria es que no existe coste de licencia para el producto en sí mismo. La mayor diferencia es el que el usuario puede, además, obtener el código fuente. Esto le brinda independencia del proveedor (“contribuyente original” en el lenguaje de Código Abierto). De este modo el usuario no depende de su existencia y prioridades.

Toda la información (estado, errores o *bugs*, etc.) es abierta también, no existe política de ocultamiento corporativa ni censura. Si algo no funciona, no tendrá inconveniente en averiguarlo rápidamente. Como consecuencia, los proyectos de Código Abierto son muy rápidos para reaccionar si surgen problemas.

La comunidad de usuarios (y desarrolladores) hacen una notable diferencia. Debido a la diversidad de usuarios, los productos están usualmente muy bien probados y se puede obtener ayuda y consejo rápidamente.

- Problemas del *Open Source*

Los proyectos de Código Abierto funcionan bien cuando el alcance es el de herramientas básicas y donde los requerimientos están claramente definidos.

La prueba de funciones y rendimiento, requiere de un enfoque muy estructurado y recursos, usualmente limitados en los proyectos de Código Abierto. Lo mismo sucede con el empaquetado (*packaging*), actualizaciones y mejoras. Otro inconveniente puede darse por el requerimiento de licencias de terceros.

Según la OSI (*Open Source Initiative*) *Open Source* no significa únicamente acceso al código fuente. Los términos de distribución del Software de Código Abierto deben cumplir los siguientes criterios:

1. Libre redistribución. La licencia no restringe la venta o cesión del software como componente de una distribución que contenga varios códigos distintos. La licencia no requiere el pago de derechos ni cualquier otra forma de pago por la venta.

2. Código Fuente. El programa incluye el código fuente y debe permitir su distribución como tal así como en forma compilada. En el caso de que el producto distribuido no incluya el código fuente debe haber una forma suficientemente clara de obtenerlo, por un precio razonable no mayor del coste de reproducción. No se permite el código fuente deliberadamente complicado ni tampoco formas intermedias como salidas del preprocesador o traductor.

3. Trabajo derivado. La licencia debe permitir la distribución de trabajos derivados en los mismos términos que el software original.

4. Integrid ad del código fuente del autor. La licencia debe hacer explícito el permiso de distribución de software generado a partir de código fuente modificado. La licencia puede requerir que los trabajos derivados lleven un nombre o número de versión diferente al del software original.

5. No discriminación contra personas o grupos de person as. La licencia no puede discriminar a ninguna persona o grupo de personas.

6. No discr iminación c ontra áreas de trabajo . La licencia no puede restringir a nadie el uso del software en un campo específico de trabajo.

7. Distribución de la licencia. Los derechos vinculados al software deben ser de aplicación para todos aquellos a los que haya sido redistribuido, sin necesidad de ejecutar una nueva licencia para aquellas partes.

8. La licencia no debe ser específica para un producto . Si el software se distribuye de acuerdo a los términos de licencia del programa, todas las partes a las que se distribuye el programa deben tener los mismos derechos que los que están licenciados en la distribución de software original.

9. La licen cia no debe restringir otro software. La licencia no puede plantear restricciones en otro software que es distribuido acompañando al software licenciado.

10. La licencia debe ser tecnológ icamente neutral. Ninguna licencia puede estar dedicada a una tecnología individual.

BLOQUE I: LA SEGURIDAD

La ley limitará el uso de la informática para garantizar el honor y la intimidad personal y familiar de los ciudadanos y el pleno ejercicio de sus derechos"
Artículo 18.4 de la Constitución Española

1. SEGURIDAD INFORMÁTICA

1.1. INTRODUCCIÓN

En el momento histórico en el que nos hallamos inmersos, la denominada "era de la información", en la que cualquier persona, entidad o compañía puede acceder fácilmente a casi cualquier tipo de información, ésta debe tratarse como lo que realmente es, un arma muy poderosa, que si no se utiliza con cuidado, puede llegar a volverse muy peligrosa e incluso destructiva.

A lo largo de todos estos años y para prevenir estos hechos, las empresas han venido desarrollando planes de seguridad y contingencia para evitar y/o solventar posibles fallos, ataques o errores de seguridad que podrían hacer peligrar la estabilidad de la organización.

Por ello, una de las principales prioridades de cualquier compañía u organización debe ser la protección de los datos que almacenan sus sistemas, ya que cualquier defecto en ésta área podría acarrear graves consecuencias, no sólo por la pérdida económica que probablemente generaría, sino por la pérdida de confianza por parte de los usuarios y posibles clientes e interesados de la misma, que seguramente, recordarían cómo la empresa no pudo salvaguardar sus propios datos, creándose así, una espiral destructiva de la imagen corporativa de la empresa que llevaría a tomar medidas drásticas para solucionar estos problemas.

De acuerdo con lo anterior, la implementación de políticas de seguridad requiere un alto compromiso con la organización, agudeza, destreza y experiencia técnica así como un estudio profundo y conocimiento del tema, para detectar fallos y debilidades, además de una marcada constancia para renovar y actualizar dichas políticas en función del ambiente dinámico y cambiante que nos rodea en la actualidad.

1.2. CONCEPTOS Y DEFINICIONES

Para poder avanzar en éste ámbito tan importante, primero se ha de definir el término seguridad para poder comprender mejor su significado. Según el Diccionario de la Real Academia Española de la Lengua (RAE) está definido como:

Seguridad: "*Cualidad de seguro*"

Que nos lleva a la búsqueda del otro concepto importante, "seguro". Dentro del Diccionario de la RAE se encuentran recogidas varias acepciones de esta palabra que pueden ser útiles. Seguro: "*libre y exento de todo peligro, daño o riesgo. Lugar o sitio libre de todo peligro. Mecanismo que impide el funcionamiento indeseado de un aparato (…)"*.

Sin embargo, y a pesar de esta afirmación tan rotunda que se recoge dentro del Diccionario de la RAE, ningún sistema informático es seguro y fiable al cien por cien. Casi siempre cabe la posibilidad que exista una vulnerabilidad en el sistema, un fallo, error, o posibilidad no recogida, que por muy pequeña que sea puede ser atacada o vulnerada.

Seguridad es un concepto muy amplio que puede abarcar muchos aspectos, se podría decir que está dividida en diferentes áreas, en cada una de las cuales, se estudiará un ámbito u otro. Así, si se centra más en la seguridad lógica de la información, o se centra en la seguridad física, o en ambas, se tendrán que observar diferentes factores para cada uno de los casos. Por ejemplo, el término utilizado como "seguridad de los sistemas informáticos", estará orientado a la seguridad física de los elementos informáticos que almacenan los datos e informaciones, se podría decir que se centra más en el continente, dejando en un segundo plano el contenido - esto es, los datos y la información -. Asimismo, si hablamos de "seguridad de la información" tendremos el caso contrario, puesto que lo que primará será la seguridad lógica, la integridad de los datos.

Según José Luis Rivas López en su libro Protección de la información (2003) la "seguridad de la información" es:

> "(…) el estudio de los métodos y medios de protección de los sistemas y comunicaciones frente a revelaciones, modificaciones o destrucciones de la información, o ante fallos de proceso, almacenamiento o transmisión de dicha información, que tienen lugar de forma accidental o intencionada. La Seguridad de la Información se caracteriza como la protección frente a las amenazas de Confidencialidad, Integridad y Disponibilidad y pueden ser amenazas de fuerza mayor, fallos de organización, humanos o técnicos o actos malintencionados."

Si citamos[1] la norma ISO/IEC 27002 con respecto a la "seguridad de la información" afirma que:

> "La información es un activo que, como otros activos importantes del negocio, es esencial para la organización y por lo tanto las necesidades de la misma han de ser protegidas convenientemente. Esto es especialmente importante en un ambiente de negocio cada vez más interconectado. Como resultado de este aumento en la interconexión, actualmente la información está expuesta a un número creciente y a una variedad más amplia de amenazas y de vulnerabilidades.
>
> La información puede existir en muchas formas. Puede ser impresa o estar escrita en papel, almacenada electrónicamente, transmitida por correo o usando medios electrónicos, mostrada en películas, o hablada en una conversación. En cualquiera de las formas que pueda adoptar ésta debe ser protegida siempre apropiadamente."

---

1. *Traducción de la autora del PFC (fragmento What is information security?)*

Evidentemente, lo ideal sería aunar ambos aspectos de la seguridad, con lo que se podría obtener una seguridad más completa y activa, que tomaría en cuenta el conjunto global de la misma y no sólo una en particular, con lo que el sistema sería menos vulnerable a posibles amenazas internas o externas.

De nada servirá que la seguridad sea buena respecto a la protección de datos, si la seguridad física no está controlada, el resultado final será igualmente la inseguridad de un sistema que puede ser atacado en este aspecto básico.

De esta unión activa de ambos aspectos igualmente importantes surge el concepto "seguridad de los sistemas de información" que abarca los dos campos como uno solo y que permite un control más amplio de la situación. Desde un punto de vista general, se podría decir que la "**seguridad informática**" está contenida dentro de esta última clasificación, puesto que ésta tendrá en cuenta todos los factores que pueden alterar de una forma u otra un sistema o conjunto de sistemas de cualquier tipo teniendo en cuenta además su entorno. De hecho existe un viejo dicho en la seguridad informática que dicta que "todo lo que no está permitido debe estar prohibido".

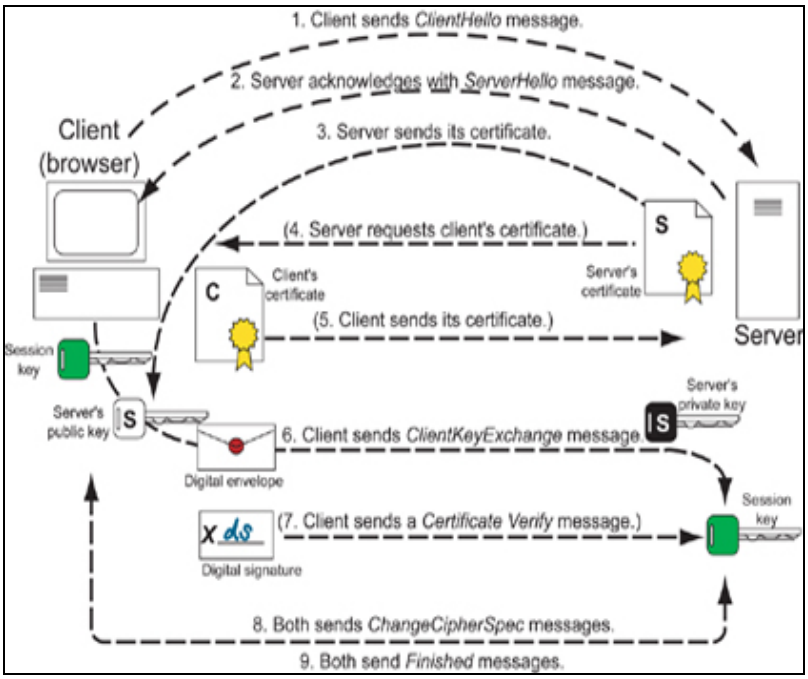


***Figura 1**. Uno de tantos ejemplos de seguridad informática*

1.3. ELEMENTOS DE LA SEGURIDAD INFORMÁTICA

Ya hemos visto lo importante que es la seguridad informática en cualquier ámbito relacionado con la informática. Ahora indagaremos más en cuáles son los principios y elementos necesarios para obtenerla adecuadamente.

Según Jorge Ramió Aguirre, profesor de la Universidad Politécnica de Madrid, explica en su "Curso de Seguridad Informática" (2002) que la seguridad informática está compuesta por tres principios fundamentales:

> "*Primer principio: El intruso del sistema utilizará cualquier artilugio que haga más fácil su acceso y posterior ataque.*"

Advierte de la gran diversidad de frentes desde los que puede producirse el ataque que dificultará en gran medida el análisis de riesgos, puesto que el atacante intentará encontrar el punto del sistema más vulnerable.

> "*Segundo principio: Los datos deben protegerse sólo hasta que pierdan su valor*."

Según este principio que se plantea, existe caducidad del sistema de protección. Por lo que se tendrá que valorar adecuadamente en qué momento concreto una información deja de ser relevante para el sistema.

> "*Tercer principio: Las medidas se control se implementan para ser utilizadas de forma efectiva. Deben ser eficientes, fáciles de usar y apropiadas al medio*."

Por último, establece que las medidas de control deben funcionar en el momento oportuno, así como que se realicen de forma que optimicen los recursos del sistema.

En cualquier caso los pilares básicos en los que ha de sustentarse la seguridad informática para que ésta sea eficiente son cinco: la **confidencialidad**, la **integridad,** la **disponibilidad,** la **autenticidad** y la **imposibilidad de rechazo (no repudio)**.

1.3.1 Confidencialidad

La confidencialidad es la característica que hace que los componentes del sistema sean accesibles sólo por los usuarios autorizados del mismo. Con esto se consigue que los datos e informaciones recogidas dentro de un sistema no puedan ser reveladas ni mostradas a ningún usuario o personal no autorizado que pudiera hacer un uso fraudulento o incorrecto de éstos.

Así según el Reglamento de desarrollo de la Ley Orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal[*] se establecen varios niveles de seguridad, en base a los cuales se necesitará un grado de protección más alto atendiendo a la naturaleza de la información:

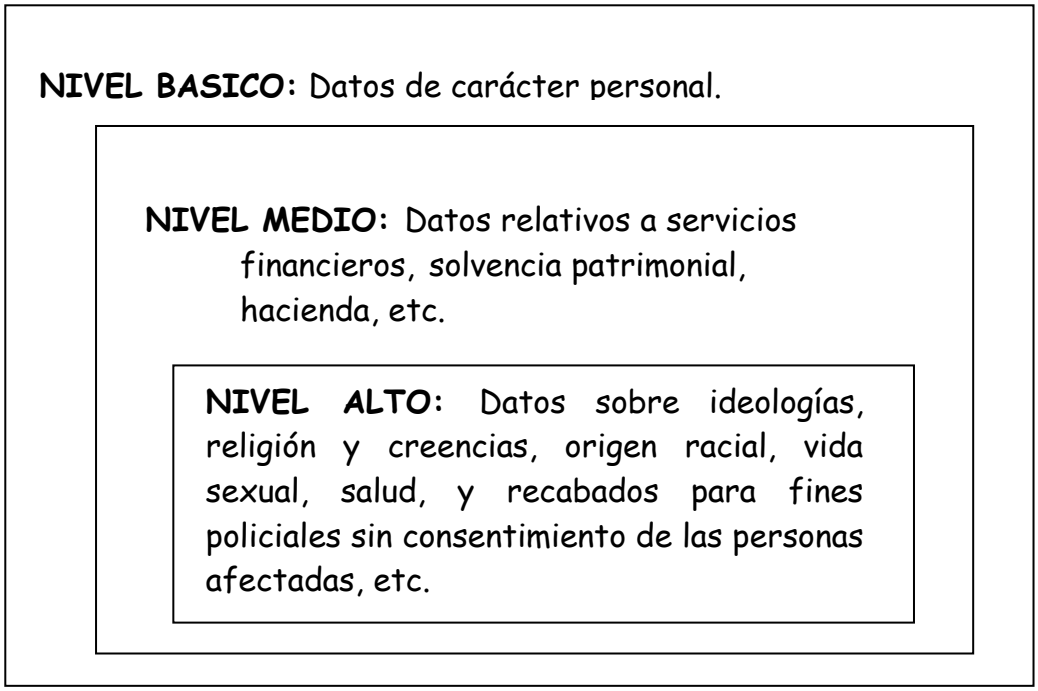


Figura 2. *Niveles de seguridad según la naturaleza de la información*

[*]Boe Num.17. Sábado 19 de enero de 2008

Para controlar este aspecto, se desarrollaron sistemas mediante transformación de la información, que intentaron proteger la confidencialidad de los datos, actualmente ha derivado en lo que hoy en día se conoce como criptografía (protección de los datos encubriéndolos mediante algoritmos que dependerán de un parámetro o clave)[2].

1.3.2. La integridad

La integridad en la información es la característica que asegura que la información de que se dispone es completa y exacta en todo momento. Además que la misma sólo puede ser modificada o gestionada por los usuarios autorizados para ello, sin que ningún otro usuario externo a la misma pueda acceder e ella (en el caso de que los datos sean modificados por personal autorizado, esta modificación deberá ser registrada para posteriores controles y auditorías).

Así la integridad de un sistema será un aspecto clave, puesto que se tendrá la confianza de que los datos que se están manejando tienen la fiabilidad necesaria y no han sido manipulados.

1.3.3. La disponibilidad

No serviría de nada tener una información y unos datos almacenados de forma segura a los que no se pudiera acceder en el momento en que se necesiten. Por ello esta característica asegura que los datos que están almacenados dentro del sistema sean accesibles por los usuarios autorizados para ello independientemente de la situación en la que éstos se encuentren.

En caso de que suceda algún fallo, error, pérdida o ataque en el sistema, éste tiene que tener la capacidad y autonomía suficiente para poder recuperar el control lo antes posible, y así seguir ofreciendo las funcionalidades de las que dispone.

---

2. *Extraído de Seguridad y Protección de la Información de UC3M (2006)*

1.3.4. La autenticidad

La autenticidad es la manera de asegurar el origen y el destino de la información, asegurando que la entidad no es falsa, ya sea utilizando la firma electrónica o digital, la validez del correo electrónico, mediante biometría, etc.

1.3.5. Imposibilidad de rechazo (no repudio)

A través de esta imposibilidad de rechazo se asegura que cualquier entidad o usuario que haya enviado o recibido información, niegue el haberlo hecho para eximirse de responsabilidades o con cualquier otro fin.

Esto es, cualquier usuario que haya enviado información nunca podrá negar ante otros que no lo hizo, puesto que sus acciones han sido registradas de forma que sin lugar a dudas pueda ser identificado como el causante de su acción. Al igual que sucede con el destinatario de esa información que tampoco podrá repudiar el hecho de que la recibió. Cualquiera de las dos partes implicadas en la transmisión (no repudio de origen) y recibo de la información (no repudio de recepción) tiene en su poder las pruebas irrefutables a partir de las cuales se puede demostrar que la comunicación efectivamente sí existió, independientemente de que lo nieguen las partes involucradas.

1.3.6. Conclusión

Lógicamente, para que exista esta seguridad informática de la que se ha venido comentando en apartados anteriores, todos los procesos de gestión de seguridad y políticas de control deberán tener muy en cuenta dichos conceptos e intentar cumplirlos en la medida de lo posible, puesto que sin ellos, no se podría tener la seguridad de que el sistema sea realmente seguro.

1.4. TIPOS DE SEGURIDAD INFORMATICA

Como se ha dicho anteriormente, se puede entender como seguridad una característica de cualquier sistema, ya sea informático o no, que indica que ese sistema está libre de peligro o riesgo. Sin embargo, esta característica es extremadamente complicada de conseguir, lo que se deberá intentar es que el sistema sea lo más fiable posible.

El estudio de la seguridad puede hacerse dependiendo de las fuentes de amenazas a los sistemas, de ahí surge la necesidad de diferenciar entre: la seguridad física y la seguridad lógica.

1.4.1. Seguridad Lógica

La seguridad lógica protege la información mediante el uso de herramientas de seguridad. Podría definirse como “el conjunto de operaciones y técnicas orientadas a la protección de la información contra la destrucción, la modificación, la divulgación indebida, etc.”.

La seguridad lógica consiste en la aplicación de barreras y procedimientos que resguarden el acceso a los datos, de tal manera que sólo puedan acceder a ellos las personas autorizadas para hacerlo.

Los objetivos que se plantean son:

- Que la información que se transmita sea recibida por su destinatario y no por otro.
- Que la información que se recibe lo haga inalterada, es decir, que la información enviada sea la misma que la recibida.
- Que se utilicen los datos, programas y archivos correctos por el procedimiento adecuado.
- Restringir el acceso a programas y archivos.

- Que los operadores puedan trabajar sin una supervisión exhaustiva y no puedan modificar los programas ni los archivos que no les corresponden.
- Que se disponga de pasos alternativos de emergencia para la transmisión de información.

La seguridad lógica está estandarizada de acuerdo a unos niveles determinados de seguridad. El estándar más utilizado internacionalmente es el que ofrece *Trusted Computer System Evaluation* (en adelante TCSEC), desarrollado en 1982. Los niveles describen diferentes tipos de seguridad de un sistema operativo y se enumeran desde el mínimo grado de seguridad al máximo.

Nivel D: Sistemas que no cumplen con ninguna especificación de seguridad. No hay protección para el hardware, el sistema operativo es inestable y no hay autentificación.

Nivel C1: Protección discrecional. El acceso a distinta información se realiza mediante identificación de usuarios. Cada usuario maneja por tanto, una información privada y distingue entre usuarios y el administrador del sistema. Hay por tanto, acceso de control discrecional e identificación y autentificación de usuario.

Nivel C2: Protección de acceso controlado. Se debe llevar una auditoría de accesos e intentos fallidos de acceso a objetos. Tiene capacidad de restricción para que los usuarios ejecuten ciertos comandos o tengan accesos a determinados archivos; también deniega o permite datos a usuarios en base no sólo a los permisos sino también a los niveles de autorización. Requiere que se audite el sistema.

Nivel B1 : Seguridad etiquetada. A cada objeto del sistema se le asigna una etiqueta con nivel de seguridad estipulado con respecto a una jerarquía (reservado, secreto, alto secreto, etc.) y con unas categorías. Cada usuario que accede a un objeto debe poseer el permiso expreso para hacerlo (cada usuario tiene sus objetos asociados).

Nivel B2: Protección estructurada. Requiere que se etiquete cada objeto de nivel superior por ser padre de un objeto inferior. La protección estructurada es la primera que empieza a referirse al problema de un objeto a un nivel más elevado de seguridad y comunicación con otro objeto a un nivel inferior.

Nivel B3: Dominios de seguridad. Refuerza a los dominios con la instalación de hardware. Todas las estructuras de seguridad deben ser lo suficientemente pequeñas como para permitir análisis y comprobaciones ante posibles violaciones. Este nivel requiere que el usuario se conecte al sistema por medio de una conexión segura.

Nivel A: Protección verificada. Es el nivel más elevado, incluye un proceso de diseño, control y verificación, mediante métodos formales (matemáticos) para asegurar todos los procesos que realiza un usuario sobre el sistema.

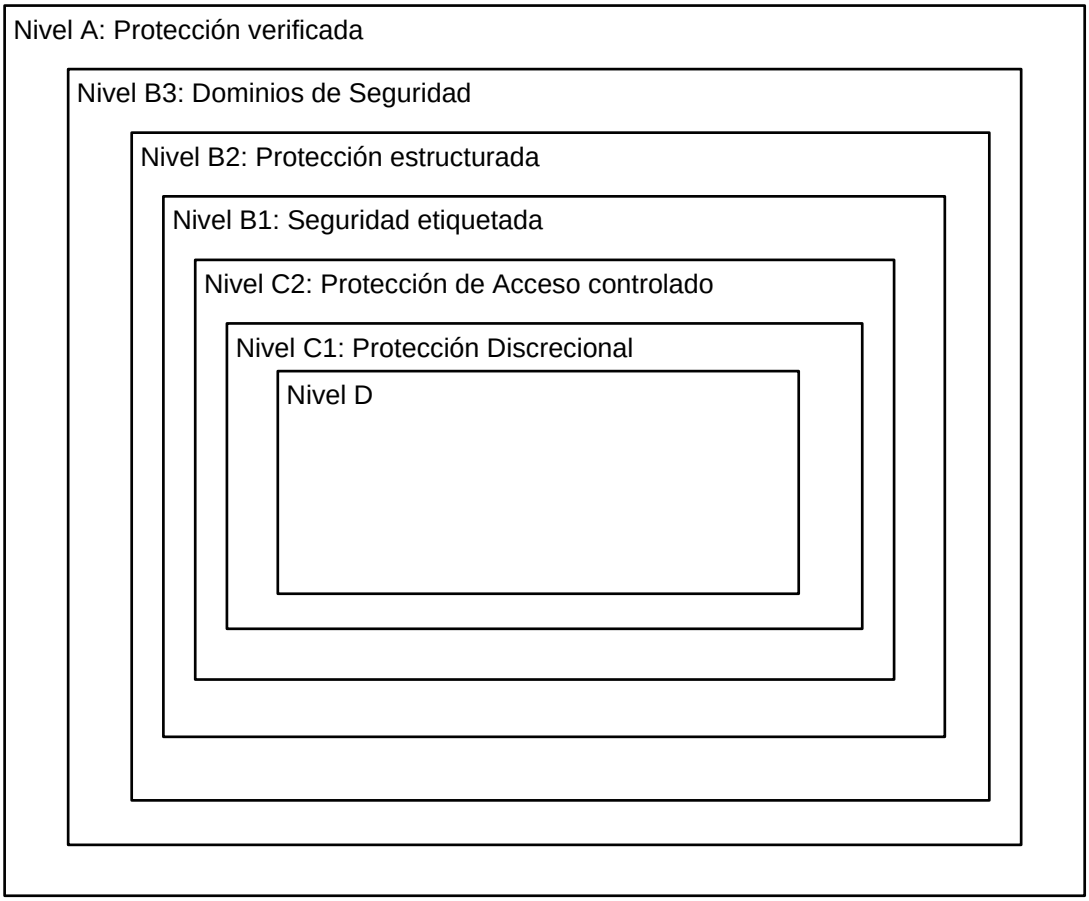


Figura 3. *Niveles de seguridad lógica. Cada nivel requiere de todos los anteriores*

1.4.2. Seguridad Física

Generalmente, cuando se habla de seguridad informática siempre se piensa en el software, sin embargo, la seguridad informática también implica otro aspecto muy importante: la seguridad física.

En muchas ocasiones la seguridad física es uno de los aspectos más olvidados a la hora del diseño de un sistema informático. Por ello, deben de tomarse medidas teniendo en cuenta también a las personas que trabajan con los equipos informáticos.

En el plan de seguridad física se debe contemplar los siguientes apartados:

- Enumeración de los recursos físicos a proteger.
- Estudio del área donde se encuentran los recursos.
- Descripción del perímetro y de los problemas potenciales o desventajas de colocar equipos en él.
- Relación de amenazas de las que hay protegerse.
- Informe de las defensas y cómo mejorarlas.
- Presupuesto que contemple el valor de la información que se está protegiendo, los costes que conlleva la recuperación de todo el sistema ante cada tipo de desastre y evaluación de las probabilidades existentes de un ataque físico, accidente o catástrofe.

Un plan de seguridad correcto deberá contener además de todos los puntos anteriores, otras medidas de seguridad particulares adaptadas a cada situación que dependerán del entorno en el que se localiza el sistema. Este plan, deberá distribuirse entre el personal responsable de su operación, y por precaución es recomendable tener una copia fuera de la dirección de informática. Además, es conveniente que la información esté tan actualizada como sea posible.

Los desastres que pueden suceder se pueden clasificar en:

- Destrucción del centro de cómputo: completa o parcial.
- Destrucción o mal funcionamiento de los equipos auxiliares del centro de cómputo.
- Destrucción parcial o total de los equipos descentralizados.
- Pérdida del personal clave.
- Huelga o problemas laborales.
- Pérdida (total o parcial) de información, manuales o documentación.

Cuando el plan sea requerido debido a una emergencia se deberá asegurar de que todos los miembros sean notificados, así como de informar al director de informática. Posteriormente hay que cuantificar el daño o pérdida del equipo, archivos y documentos, para definir qué parte del plan debe ser activado; determinar el estado de todos los sistemas en proceso debe ser el siguiente paso, así como notificar a los proveedores del equipo de cual fue el daño. Por último, habría que establecer la estrategia para llevar a cabo las operaciones de emergencias.

1.5. AMENAZAS A LA SEGURIDAD

Se podría definir como amenaza a la seguridad aquellos hechos o acciones que tarde o temprano pueden atacar un sistema. Con frecuencia se suele identificar a los atacantes únicamente como usuarios o personas, sin embargo, para globalizar y generalizar estas amenazas se hablará de “elementos” y no de personas (a pesar de que a veces resulte difícil recordar que un sistema puede verse perjudicado por múltiples entidades aparte de personas, como por ejemplo programas, catástrofes naturales, etc). Al fin y al cabo poco importa cuál haya sido la causa, puesto que el resultado será siempre el mismo, la pérdida de información relevante que debería haber sido controlada. Un ataque es la realización de una amenaza.

A continuación se presenta una relación de los elementos que potencialmente pueden amenazar un sistema, y que por tanto, atacarán los pilares sobre los que se fundamenta la seguridad informática:

- Ataque a la disponibilidad (Interrupción) : un recurso del sistema es destruido o se torna no disponible. Ejemplos de este ataque son la destrucción de un elemento hardware, cortar una línea de comunicación o deshabilitar algún sistema de gestión de información.

- Ataque a la confidencialidad (Intercepción) : una entidad no autorizada consigue acceso a un recurso. La entidad no autorizada podría ser cualquiera: una persona, un programa o un ordenador… Ejemplos de este ataque es pinchar una línea para hacerse con datos que circulen por la red y la copia ilícita de datos ya sea de ficheros, bases de datos o programas o bien la lectura de las cabeceras de paquetes para desvelar la identidad de uno o más de los usuarios implicados en la comunicación observada de forma ilegal, denominada intercepción de la identidad.

- Ataques a la integridad (Modificación): una entidad no autorizada no sólo consigue acceder a un recurso, sino que es capaz de manipularlo. Ejemplos de este ataque son el cambio de valores en un archivo, o de los datos que están almacenados en una base de datos, la alteración del código fuente de un programa para que funcione de forma diferente o modificar el contenido de los mensajes que se transfieran por la red.

- Ataques contra la autenticid ad (Fabricación) : una entidad no autorizada inserta objetos falsificados en el sistema. Ejemplo de este ataque es la inserción de registros en un archivo de datos, o en una base de datos.

Hay muchas maneras de clasificar las posibles amenazas a la seguridad a las que cualquier entidad en un momento dado puede estar expuesta. Otra forma de clasificación bastante acertada la realiza Gonzalo Álvarez Marañón del CSIC (2002) que clasifica las amenazas en dos grandes grupos: atacantes activos y atacantes pasivos.

1.5.1. Atacantes pasivos

En los ataques pasivos el atacante no altera la comunicación, sino que únicamente realiza una "escucha" o "espionaje", para obtener información que está siendo transmitida. Sus objetivos son la intercepción de datos y el análisis de tráfico, una técnica más sutil para obtener información de la comunicación, que puede consistir en:

- Obtención del origen y destinatario de la comunicación, leyendo las cabeceras de los paquetes monitorizados.

- Control del volumen de tráfico intercambiado entre las entidades monitorizadas, obteniendo así información acerca de actividad o inactividad inusuales.

- Control de las horas habituales de intercambio de datos entre las entidades de la comunicación, para extraer información acerca de los períodos de actividad.

Los ataques pasivos son muy difíciles de detectar, ya que no provocan ninguna alteración de los datos. Sin embargo, es posible evitar su éxito mediante el cifrado de la información y otros mecanismos que se verán más adelante.

1.5.2 Ataques activos

Estos ataques implican algún tipo de modificación del flujo de datos transmitido o la creación de un falso flujo de datos, que puede subdividirse a su vez en cuatro grandes grupos:

- Suplantación de identidad: el intruso se hace pasar por una entidad diferente. Normalmente incluye alguna de las otras formas de ataque activo. Por ejemplo, secuencias de autenticación pueden ser capturadas y repetidas, permitiendo a una entidad no autorizada acceder a una serie de recursos privilegiados suplantando a la entidad que posee esos privilegios, como al robar la contraseña de acceso a una cuenta de usuario.

- Reactuación: alguna información o datos son capturados y repetidos para producir un efecto no deseado, como por ejemplo ingresar dinero repetidas veces en una cuenta dada.

- Modificación de mensajes: una porción del mensaje o de los datos es alterada, también puede pasar que los mensajes sean retardados o reordenados, para producir un efecto indeseado.

- Degradación fraudulenta del servicio: impide o inhibe el uso normal de la gestión de recursos informáticos y de comunicaciones. Por ejemplo, el intruso podría suprimir todos los mensajes dirigidos a una determinada entidad o se podría interrumpir el servicio de una red. Entre estos ataques se encuentran los de denegación de servicio, consistentes en paralizar temporalmente el servicio de un servidor.

1.5.3 Otras clasificaciones

También hay que añadir que existen vulnerabilidades en el sistema que son sus puntos débiles, los aspectos en los que la seguridad es menor y por tanto, puede ser fácilmente atacada.

Además, como se ha dicho anteriormente se suele identificar al causante de la amenaza de la seguridad como una persona física, y esto no es cierto completamente, puesto que existen factores ambientales, desastres naturales (incendios, terremotos, inundaciones, condiciones climatológicas, etc.) que pueden ser tan dañinos como los humanos.

Con respecto a los ataques de personas, cabe destacar aquellos que han sido provocados de manera consciente (sabotaje realizado por personal de la empresa con algún fin, o externo, como los hackers informáticos) como los que se han producido de manera inconsciente (algún usuario autorizado que ha realizado alguna acción prohibida por desconocimiento), etc.

Otra clasificación puede ser la que se hace con respecto a la seguridad lógica y física, que se analizará con más detalle en los siguientes dos apartados.

1.5.4. Amenazas, riesgos y ataques en la Seguridad Lógica

Algunas definiciones importantes son:

- Amenazas lógicas: son todo tipo de programas que de una forma u otra pueden dañar el sistema, creados de forma intencionada para ello (software malicioso) o simplemente los ocurridos por errores (*Bugs* o agujeros). Una amenaza es la posibilidad de la ocurrencia de algún evento que afecte el buen funcionamiento de un sistema.

- Riesgo: es la proximidad o posibilidad de daño sobre un bien.

-Vulnerabilidad: es la característica del sistema o del medio ambiente que facilita que la amenaza tenga lugar.

- Ataque: es el evento que atenta sobre el buen funcionamiento de un sistema, sin importar si es intencionado o accidental.

Algunos de los tipos de ataques más importantes son:

- Ingeniería social. Esta técnica es una de las más usadas y efectivas a la hora de averiguar nombres de usuarios y *passwords*. Es la manipulación de las personas para convencerlas de que ejecuten acciones o actos que normalmente no realizan para que revelen todo lo necesario para superar las barreras de seguridad.

- Ingeniería social inversa. En este caso el intruso da a conocer de alguna manera que es capaz de brindar ayuda a los usuarios, y estos llaman ante algún imprevisto. El intruso aprovechará la oportunidad para pedir información necesaria para solucionar el problema consiguiendo información útil.

- *Decoy*. Son programas diseñados con la misma interfaz que el original. Imitan la solicitud de *login* y el usuario creyendo que es el programa original, lo hace; dicho programa esa información obtenida y deja paso a las actividades normales del sistema. La información recopilada será utilizada por el atacante para futuras "visitas".

- *Scanning*. La idea es escanear tantos puertos de escucha como sea posible, y guardar información de aquellos que sean receptivos o de utilidad para cada necesidad en particular. Muchas utilidades de auditoría también se basan en este paradigma.

  Existen diversos tipos de *scanning* entre los que destaca *TCP Connect Scanning* que es la forma básica de escaneo de puertos TCP.

- *EavesDropping* o *Packet Sniffing*. Es la interceptación pasiva del tráfico de red. Esto se realiza con *Packet Sniffers,* programas que controlan los paquetes que circulan por la red. Los *sniffers* pueden ser colocados tanto en estaciones de trabajo conectadas a la red como a un equipo *Router* o un *Gateway* de Internet y esto puede ser realizado por un usuario con legítimo acceso o por un intruso que ha ingresado por otras vías.

- *Spoofing-Looping*. "*Spoofing*" puede traducirse como "hacerse pasar por otro". Una forma común de "*Spoofing*" es conseguir el nombre y *password* para una vez ingresado al sistema, tomar acciones en nombre de él. El intruso usualmente utiliza un sistema para obtener información e ingresar en otro y luego utiliza éste para entrar en otro y así sucesivamente. Este proceso llamado "*Looping*", tiene la finalidad de ocultar la identificación y ubicación del atacante.

- *Web Spoofing*. El atacante crea un sitio web completo (falso) similar al que la víctima desea visitar. Los accesos a este sitio están dirigidos por el atacante permitiéndole controlar todas las acciones de la víctima, desde sus datos hasta las contraseñas, número de tarjetas de crédito, etc.

- Utilización de puertas traseras (*backdoors*). Las puertas traseras son trozos de código en un programa que permiten a quien las conoce saltarse los métodos usuales de autenticación para realizar ciertas tareas. Habitualmente son insertados por los programadores del sistema para agilizar la tarea de probar el código durante la fase de desarrollo. Esta situación se convierte en una falla de seguridad si se mantiene, involuntaria o intencionalmente, una vez terminado el producto ya que cualquiera que conozca el agujero o lo encuentre en su código podrá saltarse los mecanismos de control normales.

- Denegación de servicio (DoS). Los ataques de negación de servicio tienen como objetivo saturar los recursos de las víctimas de forma que se inhabiliten los servicios brindados por la misma.

- Vulnerabilidades en los navegadores. Un fallo común ha sido el denominado "*Buffer Overflow*" que consiste en explotar una debilidad relacionada con los *buffers* que la aplicación usa para almacenar las entradas de usuario. Por ejemplo, cuando el usuario escribe una dirección en formato URL ésta se guarda en un buffer para luego procesarla. Si no se realizan las oportunas operaciones de comprobación, un usuario podría manipular estas direcciones.

- Virus Informáticos. Un virus informático es un pequeño programa (invisible para el usuario) de funcionamiento específico cuyo código incluye información suficiente y necesaria para que utilizando los mecanismos de ejecución que le ofrecen otros programas puedan reproducirse formando réplicas de sí mismos susceptibles de mutar, resultando de dicho proceso la modificación, alteración y/o destrucción de los programas, información y/o hardware afectado.

Los tipos de virus más importantes son:

- Archivos Ejecutables: El virus se adosa a un archivo ejecutable y desvía el flujo de ejecución a su código, para luego retornar al huésped y ejecutar acciones esperadas por el usuario. Al realizarse esta acción el usuario no se percata de lo sucedido. Una vez que el virus es ejecutado se aloja en memoria y puede infectar otros archivos ejecutables que sean abiertos en esa máquina.

- Virus del Sector de arranque: Se guarda la zona de arranque original en otro sector del disco. Luego el virus carga la antigua zona de arranque. Al arrancar el disquete ejecuta el virus (que obligatoriamente debe tener 512 bytes o menos) quedando residente en memoria, luego ejecuta la zona de arranque original, salvada anteriormente.

- Virus Residente: El objetivo de residir en memoria es controlar los accesos a disco realizados por el usuario y el sistema operativo. Cada vez que se produce un acceso, el virus verifica si el disco o archivo objeto al que se accede está infectado y si no lo está procede a almacenar su propio código en el mismo. Este código se almacenará en un archivo, tabla de partición o en el sector de arranque dependiendo del tipo de virus de que se trate.

- Virus de Macros: Estos virus infectan archivos de información generados por aplicaciones de oficina que cuentan con lenguajes de programación de macros. Su funcionamiento consiste en que si una aplicación abre un archivo infectado, la aplicación (o parte de ella) se infecta y cada vez que se genera un nuevo archivo o se modifique uno existente contendrá el virus de macros

- Virus de Mail: Al usuario le llega vía mail un mensaje con un archivo comprimido, el usuario lo descomprime y al terminar esta acción, el contenido del archivo se ejecuta y comienza el daño. Este tipo de virus tomó relevancia con la explosión masiva de Internet y virus tipo *Melissa* y *I Love You*. Generalmente estos virus se auto-envían a algunas de las direcciones de la libreta.

- Gusanos: Un gusano es un programa capaz de ejecutarse y propagarse por sí mismo a través de redes, en ocasiones portando virus o aprovechando errores de los sistemas a los que conecta para dañarlos.

- Caballos de Troya: Programa que aparentemente realiza una función útil a la vez que una operación que el usuario desconoce y que generalmente beneficia al autor del troyano o daña el sistema huésped.

1.5.5. Riesgos y amenazas en la Seguridad Física

Hay que tener en cuenta que un sistema informático, está continuamente amenazado por los siguientes elementos:

- Empleados o ex-empleados
- Proveedores o clientes
- Competidores o gobiernos extranjeros
- Espías industriales
- Hackers, curiosos o vándalos
- La propia naturaleza

Las cinco primeras amenazas intentarán acceder al sistema informático generalmente para robar información, y raramente su finalidad será la de destruir el propio sistema, al contrario de lo que sucede con la última amenaza, que siempre ocasionará un daño físico.

Si el objetivo es evitar el robo de información, se estaría hablando de la seguridad lógica (recogida en el apartado anterior). Sin embargo, si la finalidad es evitar el daño físico, se estaría hablando de la seguridad física. No se puede intentar separar un concepto de otro, ya que ambos están estrechamente ligados.

Algunos de los riesgos físicos más comunes son los que se enumeran a continuación:

1. Fuego: Es uno de los factores que más daño puede causar, ya que podría destruir completamente el sistema. El ordenador no soportaría las altas temperaturas existentes en un incendio, y si lo hace, aún tendría que hacer frente al agua, que con

certeza, se usaría para apagar éste. Hay varios sistemas que se utilizan actualmente entre los que destacan Inergen y FE-13 (o trifluorometano) como sustitutos del Halon.

Existe una clasificación del fuego (de entre muchas otras), que figura en tratados sobre la materia, que clasifica éste en razón del material combustible que lo origina:

TIPO A: Fuego producido por materiales sólidos (papel, madera, fibra…), todos aquellos que durante su ignición producen brasas y como residuo dejan ceniza.

TIPO B: Fuego producido por gases, líquidos o sólidos inflamables, y que desprenden gases, vapores o partículas.

TIPO C: Los que tienen su origen en equipos, dispositivos o conductores eléctricos (se denominan fuegos eléctricos). En realidad son fuegos que, aunque producidos por la electricidad al originar calentamientos, se producen en los materiales aislantes y no en los conductores.

TIPO D: Los que tienen su origen en cierto tipo de metales combustibles, tales como el zinc en polvo o el aluminio en polvo, el magnesio, el litio, el sodio, el potasio, etc.

Categoría	Tipos de combustible	Agentes extintores	Acción a llevar a cabo
A	Papel, madera, etc.	Espuma - Soda-Acido-Agua-HALOTRON I	Eliminación del calor, por el agua
B	Líquidos inflamables, aceites, etc.	Gas carbónico - Polvo químico seco- Espuma	Neutralización del comburente con sustancia no inflamable
C	Equipos eléctricos encendidos	HALOTRON I- Polvo químico seco-espuma	Neutralización del comburente con sustancia no inflamable
D	Metales combustibles	Gas carbónico- Polvo químico seco	Neutralización del comburente con sustancia no inflamable

Figura 4. *Cuando descriptivo de categorías de incendios*

2. Humo: El humo es un poderoso abrasivo y tiende a almacenarse en los cabezales de las unidades de disco (ópticas y magnéticas) y de las unidades de cinta. Uno de los problemas más habituales que tiene que ver con el humo, es el que se produce por el tabaco, ya que éste se acumula muy fácilmente en partes muy delicadas del ordenador, siendo una de las más afectadas el teclado.

3. Polvo: La mayor parte del polvo es conductor de la electricidad, lo que puede inducir a fallos de difícil detección. Además, el polvo, así como el humo, es un abrasivo que tiende a acumularse en las cabezas de lectura de los sistemas de almacenamiento.

4. Terremotos: El mayor problema que aparece asociado a los terremotos lo ponen las personas del centro. Mientras que los expertos saben que no es así, se considera que una zona de escasa actividad sísmica no debe preocuparse por los terremotos. Éstos, al igual que otras muchas catástrofes, de las consideradas naturales, responden a un factor estadístico. Realmente, cuando se dice que una zona es más o menos estable, lo que realmente dice es que la probabilidad de que se produzca un gran terremoto es muy baja, lo que no quiere decir que no deban tomarse las medidas de precaución necesarias para minimizar las posibles consecuencias.

5. Explosiones: Las explosiones pueden producirse por muchos factores: porque existan conductos de gas en la proximidad, porque se almacenen productos inflamables o por acciones terroristas, etc. El edificio donde se encuentran los equipos que forman el sistema informático, es por tanto, susceptible de sufrir este tipo de desastre.

6. Insectos: Los ordenadores siguen siendo uno de los refugios favoritos de muchas clases de insectos, en especial las fuentes de alimentación. No sería la primera vez que un ordenador resultase dañado porque una polilla se ha quedado atrapada en la fuente de alimentación y produjese alteraciones en el suministro eléctrico. En la actualidad, se ha descubierto un hongo que destruye los discos compactos si se dan unas condiciones de temperatura y humedad altas.

7. Ruido eléctrico: Los motores, los grandes equipos y los ordenadores, pueden generar ruido eléctrico, siendo ésta la causa de algunos problemas inminentes. Este ruido se transmite a través del aire o a través de las líneas eléctricas cercanas. Asimismo, los equipos de transmisión por radio pueden causar errores en el funcionamiento de los

ordenadores cuando están transmitiendo. En especial, los transmisores de alta potencia pueden dañar permanentemente los equipos.

8. Tormentas: Las tormentas entrañan un riesgo para los equipos eléctricos, por lo que es aconsejable la instalación de un pararrayos en la zona, así como la desconexión de los equipos en las grandes tormentas eléctricas.

9. Vibraciones: Existen vibraciones que causan daños en los sistemas informáticos. Éstas, principalmente, pueden hacer que algunos conectores que no estén bien colocados dejen de hacer conexión. Además, los dispositivos de los equipos informáticos como los discos duros son muy sensibles a las vibraciones y, una vibración más fuerte de lo normal puede causarles daños irreversibles.

10. Humedad: Aunque la humedad relativa, previene de las descargas eléctricas que pueden llegar a producirse por la electricidad estática acumulada en los circuitos del ordenador, una humedad excesiva puede ser un problema para los equipos, por lo que es conveniente tener sensores de humedad que detecten cuando ésta está cercana al límite establecido.

11. Agua: El agua puede provocar cortocircuitos, ya que, a pesar de que el agua no es un buen conductor de la electricidad, sí lo es la sal y la suciedad que rodea el equipo, que puede provocar un daño irreparable en algunos componentes del ordenador. Las principales fuentes de agua que suponen una amenaza para los ordenadores suelen ser las lluvias y las inundaciones.

12. Vandalismo y robos: Los actos de vandalismo hacia los sistemas informáticos son mayores de lo que en un principio cabría esperar. Los propios trabajadores (un trabajador despedido que quiera vengarse, un golpe de violencia, etc.) pueden poner en peligro los equipos. Otro factor importante, es la posibilidad de robos en el sistema. Los ordenadores son piezas que son susceptibles de ser robadas, La primera medida de protección sería la de fijar los equipos a las mesas o escritorios, usando dispositivos existentes para tales efectos, a pesar de que no se evita el robo, sí se reduce el riesgo.

13. Presencia de comidas y bebidas: Los accidentes no se pueden prevenir, pero sí se puede prevenir el riesgo de que se produzcan. Uno de los mayores factores de riesgo

para causar accidente son la comida y la bebida. Al igual que el agua, se corre riesgo de derramar el líquido sobre el equipo, especialmente sobre el teclado, provocando su inoperatividad. Con la comida sucede lo mismo. La grasa de la comida se acumula en los dedos de las personas y las migas son residuos que se acumulan sobre todo en los teclados de los equipos. La mejor solución, es mantener alejada la comida de los ordenadores.

14. Acceso físico: Otro de los problemas de seguridad que se encuentran en los sistemas informáticos es el acceso físico al recinto. No sirve de nada tener el mejor cortafuegos si cualquier persona puede acceder físicamente al equipo principal. Los servidores deberían estar en una sala en la que sea precisa una llave para poder acceder a ella, además de reducir los conductos de ventilación para evitar que una persona pudiera acceder a las salas a través de dichos conductos, así como tener bien ancladas las rejillas de acceso.

Por todo ello, hay unos factores inherentes a la sala de ordenadores donde están ubicados éstos, que tienen gran importancia. Es necesario considerar las características físicas que deben tener las instalaciones para proporcionar seguridad: como el cableado, las paredes y el techo, las puertas de acceso, la iluminación, los filtros, el espacio, el aire acondicionado y la calefacción, etc.

1.6. MEDIDAS DE PREVENCIÓN Y CONTROL

Como acertadamente se recoge en el marco referencial de COBIT (*Control Objetives Management Guidelines Maturity Models*) muchas organizaciones reconocen los beneficios potenciales que la tecnología puede proporcionar. Las organizaciones exitosas, sin embargo, también comprenden y administran los riesgos asociados con la implementación de nueva tecnología.

Según la normativa ISO/IEC 27002 (antes ISO 17799-2005) establece varios tipos de seguridad preventiva y de control, en base a los cuales, cualquier organización debería orientar sus procesos para la obtención de una buena seguridad.

Se podrían clasificar los dos más relevantes y diferenciados como (dejando en un segundo plano para esta primera clasificación el resto):

1.6.1 Seguridad de los recursos humanos o ligados al personal

Establece que su objetivo es reducir los riesgos de errores humanos, robos, fraudes o mal uso de las instalaciones y de los servicios.

La seguridad debería contemplarse desde las etapas de selección de personal, incluirse en los contratos y seguirse durante el desarrollo de la relación laboral. Todo personal relacionado con el sistema deberá estar debidamente informado y ser consciente en todo momento de los fallos que pueda cometer y por ello sería conveniente la firma de un acuerdo de responsabilidad y de confidencialidad.

Igualmente, la formación de los usuarios debería ser primordial, puesto que de esta manera, se asegura que los usuarios son conscientes de las amenazas y riesgos en el ámbito de la seguridad de la información, y que por tanto, están preparados para sostener la política de seguridad que esté llevando a cabo la organización.

Si llegado el caso, el daño llegara a producirse, éste debería ser tratado de manera que se llegasen a minimizar dichos daños – provocados por incidencias de seguridad- de manera que se controlase más y se pudiese aprender de ellos. Debería informarse de las incidencias que afectasen a la seguridad de la información a través de los canales de la Organización más adecuados, asignando un punto de contacto con el que poder comunicarse en caso de necesidad.

1.6.2 Seguridad física y del entorno

Establece que su objetivo es evitar los accesos no autorizados, daños e interferencias contra los locales y la información de la organización. Los recursos para el tratamiento de la información crítica deberían ubicarse en áreas seguras, protegidas por un perímetro de seguridad o barrera de seguridad y controles de entrada apropiados. Se debería dar protección física contra el acceso, daño, e interferencia no autorizados.

La seguridad en los equipos sería primordial, puesto que ahí es donde estará almacenada gran parte de la información y de los datos con los que cuenta una organización. Por tanto uno de los objetivos sería evitar las pérdidas, daños, etc. que pudieran producirse en éstos. Una de las preguntas que cabe hacerse es: ¿cómo se pueden evitar o adoptar medidas necesarias contra un robo, incendio, polvo, agua, interferencias en el suministro eléctrico…? Mantenimientos, seguridad del cableado y otros sistemas de prevención no deberían faltar.

En definitiva todo lo que afectase físicamente al entorno en el que desarrolla su función una organización, ya que como se ha explicado con anterioridad tan importantes son los datos, como el soporte en el que estos se almacenan y su entorno.

Puesto que hay muchas áreas que se deben proteger (hay que recordar que el concepto de seguridad es clasificable de muchas maneras dependiendo de en base a qué característica se esté agrupando), se podría establecer otra clasificación de la seguridad que se explicará en los apartados siguientes.

1.6.3 Seguridad interna

Establece que su objetivo primordial es la seguridad de la información mientras ésta esté recogida dentro del sistema, como un Sistema de Gestión de Bases de Datos, Sistema de Ficheros, Sistema Operativo, etc. estableciéndose las políticas adecuadas para ello.

Para comprender mejor este concepto se propone como ejemplo –que anecdóticamente coincide con el tema del que trata el proyecto, esto es, la auditoría de la seguridad en una base de datos concreta - Si nosotros fuéramos el administrador de una base de datos éste debería controlar los temas involucrados en el nivel del sistema de archivo, que consiste principalmente en proteger los directorios donde se encontrasen los datos.

1.6.4 Seguridad externa

También denominada “Seguridad en las comunicaciones” que establece que su objetivo es la protección de la información cuando ya ha salido del sistema y está siendo transmitida a otro sistema. Se establecerán las políticas adecuadas para que la información llegue completa, exacta y sin alteraciones a su destino.

1.7. IMPLEMENTACIÓN SEGURIDAD INFORMÁTICA

Antes de la implementación y/o auditoría de cualquier solución de seguridad para la Organización es indispensable la realización de un paso previo orientado a la identificación y medición de los riesgos que tiene cada entidad en aspectos de seguridad informática, siguiendo para ello alguna línea o norma que nos indique cuáles son los alcances o criterios a tener en cuenta para la especificación de la solución requerida, y que ayude al auditor a realizar una auditoría que sea de calidad evaluando aquellos elementos que sean necesarios. Para ello una de las posibles normas de las que el auditor puede extraer algunas ideas puede ser la ISO-IEC 27002, que es una de las normas más extendidas en esta área.

Los procedimientos para la gestión de los riesgos deben implementarse como una metodología sistemática que permita definir e implantar políticas de seguridad adecuadas para cada entidad, para lo cual, y como primera medida, se deben identificar cuáles son los activos que realmente requieren mayor protección. Entre estos activos principales los más comunes son:

- Elementos de hardware: como pueden ser servidores, estaciones de trabajo, impresoras, equipos para procesos de información, etc.
- Elementos de software: programas fuente, históricos, programas objeto, programas de diagnóstico, programas de operaciones, sistemas operativos, aplicaciones, etc.
- Datos: la información almacenada y accesible en línea o fuera de línea, copias de seguridad, registros de auditoría, bases de datos e información de tránsito entre canales de información, ficheros, etc.
- Personas: usuarios de la información, administradores de la red, etc.
- Documentación: documentación de auditoría, evaluaciones internas de la infraestructura tecnológica de la organización, metodologías y procedimientos administrativos.

Una vez que se han identificado los elementos a evaluar, se deben determinar las posibles “amenazas” a los mismos, y las posibilidades reales de que esto suceda en el entorno de la propia organización que se está auditando.

Se podría entender como amenaza los accesos no autorizados a los recursos del sistema, la manipulación indebida de los datos, la apropiación indebida de la información almacenada, etc.

Una vez determinados los posibles riesgos y ataques de los que la organización puede ser objetivo, la organización debe determinar qué nivel de riesgo está dispuesta a aceptar. Una vez se ha llevado a cabo el "análisis de riesgos" y se conoce la naturaleza y la prioridad que debe concederse a su gestión, se deben establecer las "políticas de seguridad".

1.7.1 Ayuda al control de la seguridad: La norma ISO-IEC 27002

En apartados anteriores ya se han comentado de manera muy esquemática algunos de los objetivos y utilidades que se consiguen aplicando esta normativa. Es materia de desarrollo de este apartado profundizar un poco más en ella.

Desde su publicación por parte de la Organización Internacional de Normas (*International Standard Organization*) en diciembre de 2000 la norma ISO surge como la norma técnica de seguridad de la información con mayor reconocimiento internacional. La norma ISO 27002 (antes ISO 17799) se puede definir como

> "(…) un completo conjunto de controles que incluye las prácticas exitosas de seguridad de la información (…)"

La norma técnica fue redactada intencionalmente para que fuera flexible y nunca indujo a las personas que la cumplían para que prefirieran una solución de seguridad específica. Las recomendaciones que ofrece la normativa son neutrales en cuanto a la tecnología, y no ayudan a entender ni a evaluar las medidas de seguridad existentes. Así cuando en la norma se discute la necesidad de utilizar un *firewall* como medida de protección, ésta no profundiza en los tipos de *firewalls* existentes en el mercado, sino que la propia organización será la responsable en la toma de decisiones encargada de elegir aquellos productos o servicios que más le convengan.

Este hecho ha llevado a algunos detractores de esta normativa a discutir sobre la estructura irreal e imprecisa de ésta, alegando que es demasiado general para poder ser de alguna utilidad. Sin embargo, la generalidad de la norma es intencional, puesto que la aplicación de una norma que funcione con toda la variedad de entornos de tecnología de la información y que sea capaz de desarrollarse en el cambiante mundo de la tecnología es sumamente complicado, precisamente, por la complejidad y rapidez con la que se modifican estos entornos tecnológicos.

La normativa ISO 27002 define once áreas de control (algunas de ellas ya se han comentado en apartados anteriores), las cuales se deberán tener en cuenta a la hora de diseñar las soluciones de seguridad informática:

1. **Política de seguridad**

   Se necesita una política que refleje las expectativas de la organización en materia de seguridad a fin de suministrar administración con dirección y soporte.

   El principal objetivo es la elaboración de un documento de políticas de seguridad publicado por el máximo nivel directivo de la organización con una declaración apoyando sus principios y objetivos

   Se debe asegurar que se comunica esta política a todos los usuarios del sistema y que es fácilmente comprensible por todos ellos.

2. **Organización de la seguridad**

   Sugiere diseñar una estructura de administración dentro de la organización que establezca la responsabilidad de los grupos en ciertas áreas de la seguridad y un proceso para el manejo de respuesta a incidentes.

   Deben definirse los responsables de cada recurso de la organización y de su protección, siendo conveniente la delimitación del área de responsabilidad de cada persona para que no existan huecos ni solapamientos; habitualmente habrá un responsable de seguridad que

delegará la responsabilidad de seguridad en otras personas, pero en último término será él el responsable tanto del recurso como de validar la correcta implementación de las medidas de seguridad.

En el caso de contratación de terceros que desarrollarán su labor en el sistema en cuestión o de externalización total es esencial tener un modelo de contrato, validado por el departamento jurídico –en caso de hacer falta-, que contenga todos los requerimientos de seguridad para asegurar el cumplimiento de las políticas y normas de la organización, con especial énfasis en la Ley Orgánica de Protección de Datos. Además debe ser lo suficientemente claro para que no puedan surgir malentendidos entre la organización y el proveedor.

3. **Gestión de activos**

Necesita un inventario de los recursos de información de la organización y en base a ello asegurar que proporcione un nivel adecuado de protección.

En primer lugar hay que contar con un exhaustivo inventario de activos que deberá incluir los recursos de información de todo tipo: recursos hardware, software, servicios informáticos y de comunicaciones, así como de todos aquellos recursos que afecten en alguna medida como climatización, suministro eléctrico, etc.

4. **Seguridad ligada al personal**

Establece la necesidad de formar e informar a los empleados actuales y potenciales sobre lo que se espera de ellos en materia de seguridad. Se debe implementar un plan para el reporte de incidencias.

5. **Seguridad física y ambiental**

   Establece la necesidad de proteger las áreas donde se encuentren los equipos y controles generales de la organización, así como la propia seguridad de los equipos.

6. **Manejo de las comunicaciones y las operaciones**

   Algunos de los objetivos son la minimización del impacto si fallan los sistemas, protección de la integridad del software y de la información, garantizar la protección de la información en las redes, evitar interrupciones en las actividades de la organización, evitar la pérdida, modificación o uso incorrecto de la información susceptible de ser intercambiada entre organizaciones. Gestión de seguridad de redes, manipulación de soportes, intercambios de información, etc.

7. **Control de acceso**

   Establece la importancia de monitorear y controlar el acceso a la red y los recursos de aplicación para proteger contra los abusos internos e intrusos externos.

8. **Desarrollo y mantenimiento de los sistemas**

   Reitera la importancia de mantener el control y mantenimiento de los sistemas, así como el uso de controles de la seguridad durante todas las etapas del proceso.

9. **Gestión de incidentes de la seguridad de la información**

   Aconseja estar preparado para las posibles interrupciones en las actividades de la organización y para proteger los procesos importantes de la empresa en caso de algo falle.

10. Gestión de la continuidad del negocio

En el que se tratan los aspectos de la seguridad de la información en la gestión de la continuidad del negocio. Evaluación del riesgo, pruebas de mantenimiento de los planes de continuidad, etc.

11. Cumplimiento

Imparte instrucciones a las organizaciones para que verifiquen si el cumplimiento de la normativa técnica concuerda con otros requisitos jurídicos o legales a los que pueda estar sujetos.

1.7.2 Enfoque conceptual del IT Governance Institute (COBIT)

Los controles COBIT son un modelo autoritativo internacional para el uso diario en esquemas de control y auditoría. Es el modelo actualmente utilizado por los miembros del ISACA (*Information Systems Audit and Control Association & Foundation*) a nivel mundial.

COBIT fue desarrollado como un estándar generalmente aplicable y aceptado para la práctica del control de la Tecnología Informática (TI).

COBIT está basado en los Objetivos de Control existentes de la ISACA mejorados con los estándares internacionales existentes y emergentes técnicos, profesionales, reguladores y específicos de la industria. Los Objetivos de Control resultantes, aplicables y aceptados en forma generalizada, han sido desarrollados para ser aplicados a los sistemas de información de toda la empresa

COBIT ayuda a salvar las brechas existentes entre necesidades de control, aspectos técnicos y riesgos de negocio. Proporciona prácticas útiles a través de un Marco Referencial de procesos y dominios, además de presentar actividades en una estructura manejable y lógica.

Las organizaciones deben cumplir con requerimientos de calidad, de seguridad, tanto para su información, como para sus activos. La administración deberá obtener un balance adecuado en el empleo de recursos disponibles, los cuales incluyen: personal, instalaciones, tecnología, sistemas de aplicación y datos.

Para cumplir con lo anteriormente expuesto, así como para alcanzar las expectativas, la administración deberá establecer un sistema adecuado de control interno. Por lo tanto, este sistema de control deberá existir para proporcionar soporte a los procesos de negocio y debe ser preciso en la forma en la que cada actividad individual de control satisface los requerimientos de información y puede impactar en los recursos de TI.

El impacto de los recursos de TI es resaltado en el Marco Referencial de COBIT conjuntamente a los requerimientos de información del negocio que deben ser alcanzados: Efectividad, Eficiencia, Confidencialidad, Integridad, Disponibilidad, Cumplimiento y Confiabilidad.

El control, que incluye estructuras, prácticas, políticas y procedimientos organizacionales, es responsabilidad de la administración. La administración, mediante este gobierno corporativo, que es el sistema que establece la alta gerencia para asegurar el logro de los objetivos de una Organización, debe asegurar que la debida diligencia sea ejercitada por todos los individuos involucrados en la administración, empleo, diseño, desarrollo u operación de sistemas de información. Un Objetivo de Control es una definición del resultado o propósito que se desea alcanzar implementando procedimientos de control específicos dentro de una actividad de TI.

La orientación a negocios es el tema principal de COBIT. Está diseñado no sólo para ser utilizado por usuarios y auditores, sino que en forma más importante, está diseñado para ser utilizado como una lista de verificación detallada para los propietarios de los procesos de negocio.

En forma incremental, las prácticas de negocio requirieren de una mayor delegación y apoderamiento de los dueños de procesos para que éstos posean total responsabilidad de todos los aspectos relacionados con dichos procesos de negocio.

En forma particular, esto incluye el proporcionar controles adecuados. El Marco Referencial de COBIT proporciona herramientas al propietario de procesos de negocio que facilitan el cumplimiento de esta responsabilidad. El Marco Referencial comienza con una premisa simple y útil:

Con el fin de proporcionar la información que la empresa necesita para alcanzar sus objetivos, los recursos deben ser administrados por un conjunto de procesos agrupados en forma natural.

Continúa con un conjunto de 34 Objetivos de Control de alto nivel, uno para cada uno de los Procesos de TI, agrupados en cuatro dominios:

- Planificación y organización
- Adquisición e implementación
- Entrega y soporte (de servicio)
- Monitoreo

Estos dominios contemplan la totalidad de los procesos típicos de la función de la tecnología en prácticamente cualquier organización de negocios.

Dirigiendo esos 34 Objetivos de Control de alto nivel, el propietario de procesos de negocio podrá asegurar que se proporciona un sistema de control adecuado para el ambiente de tecnología de información. Adicionalmente, correspondiendo a cada uno de los 34 objetivos, existe una guía de auditoría que permite la revisión de los procesos de TI contra los 302 objetivos de control recomendados por COBIT para proporcionar a la Gerencia la certeza de su cumplimiento y/o una recomendación para su mejora.

COBIT contiene un conjunto de herramientas de implementación que proporciona lecciones aprendidas por empresas que rápida y exitosamente aplicaron COBIT en sus ambientes de trabajo. Incluye un resumen ejecutivo para el entendimiento y la sensibilización de la alta gerencia sobre los principios y conceptos fundamentales de COBIT. La guía de implementación cuenta con dos útiles herramientas: diagnóstico de sensibilización gerencial y diagnóstico de control en TI, para proporcionar asistencia en el análisis del ambiente de control en una organización.

El Marco Referencial COBIT otorga especial importancia al impacto sobre los recursos de TI, así como a los requerimientos de negocios en cuanto a Efectividad, Eficiencia, Confidencialidad, Integridad, Disponibilidad, Cumplimiento y Confiabilidad, que deben ser satisfechos. Además, el Marco Referencial proporciona definiciones para los requerimientos de negocio que son derivados de objetivos de control superiores en lo referente a calidad, seguridad y reportes fiduciarios en tanto se relacionen con Tecnología de Información.

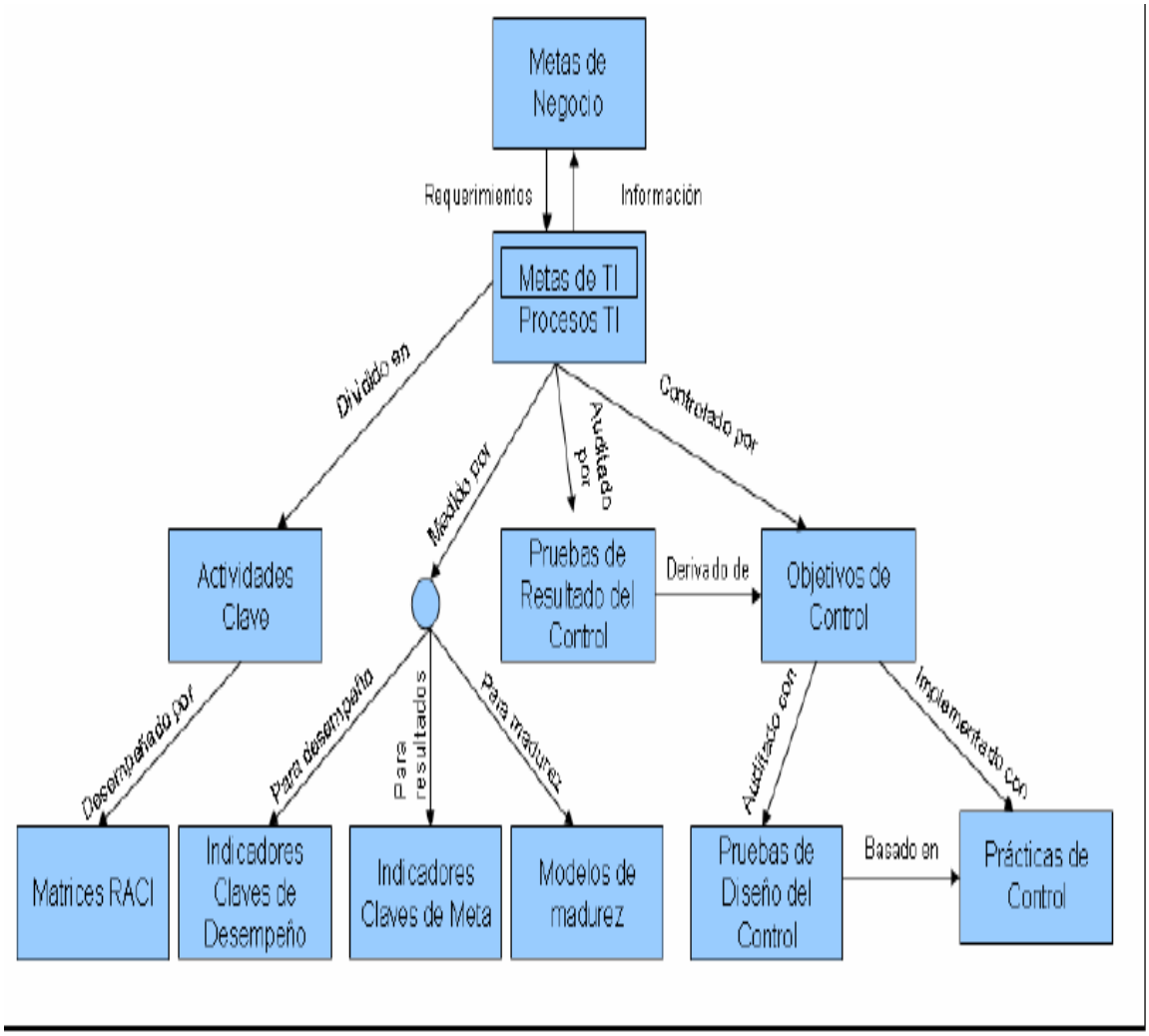


Figura 5. *Interrelación entre los componentes de COBIT 4.1*

La administración de una empresa requiere de prácticas generalmente aplicables y aceptadas de control y gobierno en TI para medir en forma comparativa tanto su ambiente de TI existente, como su ambiente planeado.

El desarrollo de COBIT ha resultado en la publicación de:

- Un Resumen Ejecutivo, el cual, adicionalmente a esta sección de antecedentes, consiste en una síntesis ejecutiva -que proporciona a la alta gerencia entendimiento y conciencia sobre los conceptos clave y principios de COBIT- y el Marco Referencial -el cual proporciona a la alta gerencia un entendimiento más detallado de los conceptos clave y principios de COBIT e identifica los cuatro dominios de COBIT y los correspondientes a los 34 procesos de TI-.

- El Marco Referencial que describe en detalle los 34 objetivos de control de alto nivel e identifica requerimientos de negocio para la información y los recursos de TI que son impactados en forma primaria por cada objetivo de control.

- Directrices de Auditoría, las cuales contienen los pasos de auditoría correspondientes a cada uno de los 34 objetivos de control de TI de alto nivel para proporcionar asistencia a los auditores de sistemas en la revisión de los procesos de TI con respecto a los 302 objetivos detallados de control recomendados para proporcionar a la gerencia certeza o una recomendación de mejora.

- Un Conjunto de Herramientas de Implementación, las cuales proporcionan lecciones aprendidas por organizaciones que han aplicado COBIT rápida y exitosamente en sus ambientes de trabajo.

El conjunto de herramientas de implementación incluye la síntesis ejecutiva, proporcionando a la alta gerencia eficiencia y entendimiento de COBIT. También incluye una guía de implementación con dos herramientas muy útiles: Diagnóstico de la Conciencia de la Gerencia y el Diagnóstico de Control de TI – para proporcionar asistencia en el análisis del ambiente de control en TI de una organización.

El objetivo principal del proyecto COBIT es el desarrollo de políticas claras y buenas prácticas para la seguridad y el control de Tecnología de Información, con el fin de obtener la aprobación y el apoyo de las entidades comerciales, gubernamentales y profesionales en todo el mundo.

La meta del proyecto es el desarrollar estos objetivos de control principalmente a partir de la perspectiva de los objetivos y necesidades de la empresa. Esto concuerda con

la perspectiva COSO (*Committee of Sponsoring Organitations*), que constituye el primer y mejor marco referencial para la administración en cuanto a controles internos. Posteriormente, los objetivos de control fueron desarrollados a partir de la perspectiva de los objetivos de auditoría (certificación de información financiera, certificación de medidas de control interno, eficiencia y efectividad, etc.).

1.7.2.1. Audiencia

COBIT está diseñado para ser utilizado por tres audiencias distintas:

1. Administración: Para ayudar a lograr un balance entre los riesgos y las inversiones en control en un ambiente de tecnología de información frecuentemente impredecible.

2. Usuarios: Para obtener una garantía en cuanto a la seguridad y controles de los servicios de tecnología de información proporcionados internamente o por terceras personas.

3. Auditoría de Sistemas de Información: Para dar soporte a las opiniones mostradas a la administración sobre los controles internos.

Además de responder a las necesidades de la audiencia inmediata de la Alta Gerencia, a los auditores y a los profesionales dedicados al control y seguridad, COBIT puede ser utilizado dentro de las empresas por el propietario de procesos de negocio en su responsabilidad de control sobre los aspectos de información del proceso, y por todos aquéllos responsables de TI de la empresa.

1.7.2.2. Principios del marco referencial

Existen dos clases distintas de modelos de control disponibles actualmente: aquéllos de la clase del “modelo de control de negocios” (por ejemplo COSO) y los modelos más enfocados a la Tecnología de Información (por ejemplo DTI). COBIT intenta cubrir la brecha que existe entre las dos, debido a esto, COBIT se posiciona como una herramienta más completa para la Administración y para operar a un nivel superior que los estándares de la tecnología de la administración de sistemas de información. Por lo tanto, COBIT es el modelo para el gobierno de TI.

El concepto fundamental del marco referencial COBIT se refiere a que el enfoque del control en TI se lleva a cabo visualizando la información necesaria para dar soporte a los procesos de negocio y considerando la información como el resultado de la ampliación combinada de recursos relacionados con la Tecnología de Información que deben ser administrados por procesos de TI.

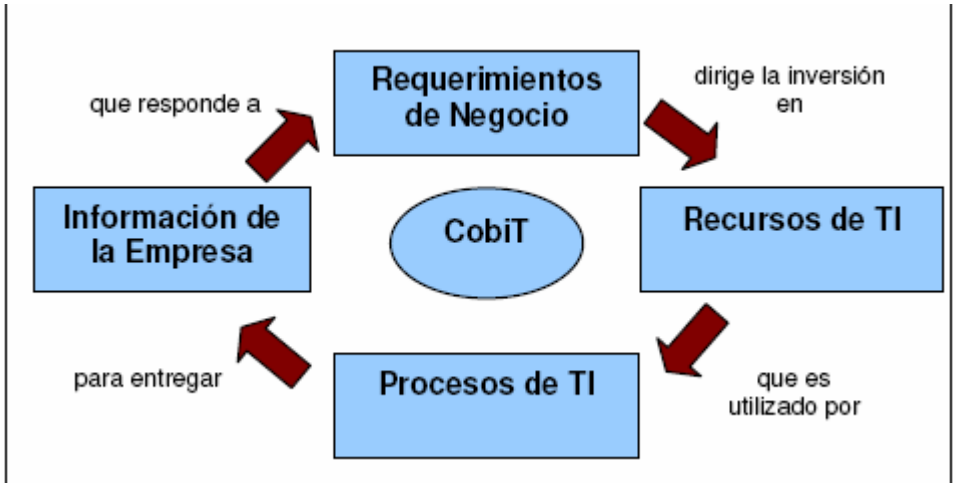


***Figura 6**. Principios básicos de COBIT*

Para satisfacer los objetivos del negocio, la información necesita concordar con ciertos criterios a los que COBIT hace referencia como requerimientos de negocio para la información. Al establecer la lista de requerimientos, COBIT combina los principios contenidos en los modelos referenciales existentes y conocidos:

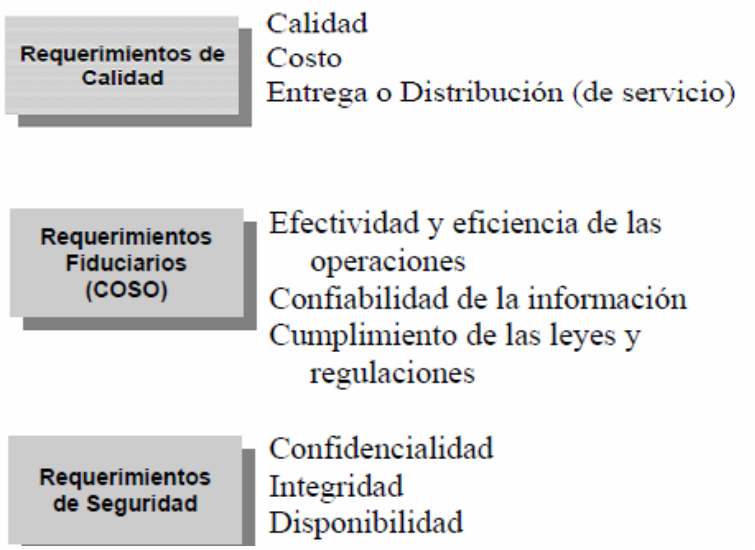


***Figura 7**. Principios*

La Calidad ha sido considerada principalmente por su aspecto 'negativo' ausencia de fallos, confiabilidad, etc.), lo cual también se encuentra contenido en gran medida en los criterios de Integridad. Los aspectos positivos, pero menos tangibles, de la calidad (estilo, atractivo, desempeño más allá de las expectativas, etc.) no fueron, por un tiempo, considerados desde un punto de vista de Objetivos de Control de TI. La premisa se refiere a que la primera prioridad deberá estar dirigida al manejo apropiado de los riesgos al compararlos contra las oportunidades.

El aspecto utilizable de la Calidad está cubierto por los criterios de efectividad. Se consideró que el aspecto de entrega o distribución del servicio, de la Calidad se traslapa* con el aspecto de disponibilidad correspondiente a los requerimientos de seguridad y también en alguna medida, con la efectividad y la eficiencia. Finalmente, el Costo también es considerado, siendo cubierto por la Eficiencia.

Para los requerimientos fiduciarios, COBIT no intentó reinventar la rueda – se utilizaron las definiciones de COSO para la efectividad y eficiencia de las operaciones, confiabilidad de información y cumplimiento con leyes y regulaciones. Sin embargo, confiabilidad de información fue ampliada para incluir toda la información – no sólo información financiera.

Con respecto a los aspectos de seguridad, COBIT identificó la confidencialidad, integridad y disponibilidad como los elementos clave se encontró que estos mismos tres elementos son utilizados a nivel mundial para describir los requerimientos de seguridad.

Comenzando el análisis a partir de los requerimientos de Calidad, Fiduciarios y de Seguridad más amplios, se extrajeron siete categorías distintas, ciertamente superpuestas. A continuación se muestran las definiciones utilizadas por COBIT:

** Según definición de la RAE: "Cubrir total o parcialmente algo con otra cosa"*

-**Efectividad**: Se refiere a que la información relevante sea pertinente para el proceso del negocio, así como a que su entrega sea oportuna, correcta, consistente y de manera utilizable.

- **Eficiencia**: Se refiere a la provisión de información a través de la utilización óptima (más productiva y económica) de recursos.

- **Confidencialidad** : Se refiere a la protección de información sensible contra divulgación no autorizada.

- **Integridad**: Se refiere a la precisión y suficiencia de la información, así como a su validez de acuerdo con los valores y expectativas del negocio.

- **Disponibilidad**: Se refiere a la disponibilidad de la información cuando ésta es requerida por el proceso de negocio ahora y en el futuro. También se refiere a la salvaguarda de los recursos necesarios y capacidades asociadas.

- **Cumplimiento**: Se refiere al cumplimiento de aquellas leyes, regulaciones y acuerdos contractuales a los que el proceso de negocios está sujeto, por ejemplo, criterios de negocio impuestos externamente.

- **Confiabilidad de la i nformación**: Se refiere al suministro de información apropiada para la administración de las operaciones del negocio y para ejercer sus responsabilidades de reportes financieros y de cumplimiento.

Los recursos de TI identificados en COBIT pueden explicarse o definirse como se muestra a continuación:

- **Datos**: Son objetos en su más amplio sentido, (por ejemplo, externos e internos), estructurados y no estructurados, gráficos, sonido, etc.

- **Aplicaciones**: Se entiende como sistemas de aplicación la suma de procedimientos manuales y programados.

- **Tecnología** : La tecnología cubre hardware, sistemas operativos, sistemas de administración de bases de datos, redes, multimedia, etc.

- **Instalaciones**: Recursos para alojar y dar soporte a los sistemas de información.

- **Personal**: Habilidades del personal, conocimiento, sensibilización y productividad para planear, organizar, adquirir, entregar, soportar y monitorear servicios y sistemas de información.

El dinero o capital no se tuvo en cuenta como un recurso para la clasificación de objetivos de control para TI debido a que puede considerarse como la inversión en cualquiera de los recursos mencionados anteriormente.

Es importante hacer notar también que el Marco Referencial no menciona, en forma específica para todos los casos, la documentación de todos los aspectos "materiales" importantes relacionados con un proceso de TI particular. Como parte de las buenas prácticas, la documentación es considerada esencial para un buen control y, por lo tanto, la falta de documentación podría ser la causa de revisiones y análisis futuros de controles de compensación en cualquier área específica en revisión.

Otra forma de ver la relación de los recursos de TI con respecto a la entrega de servicios se describe a continuación:

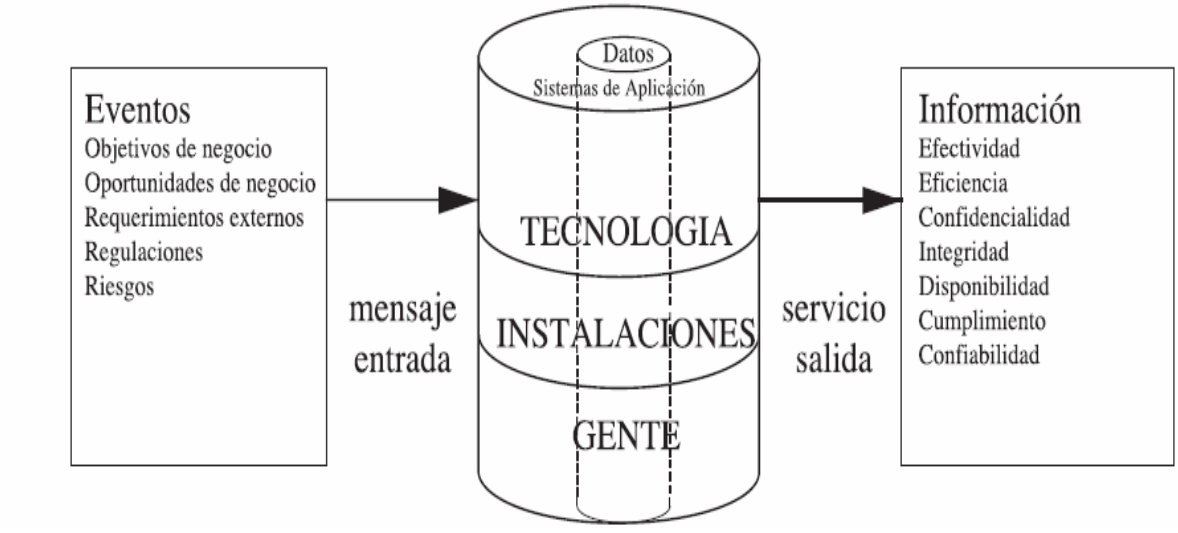
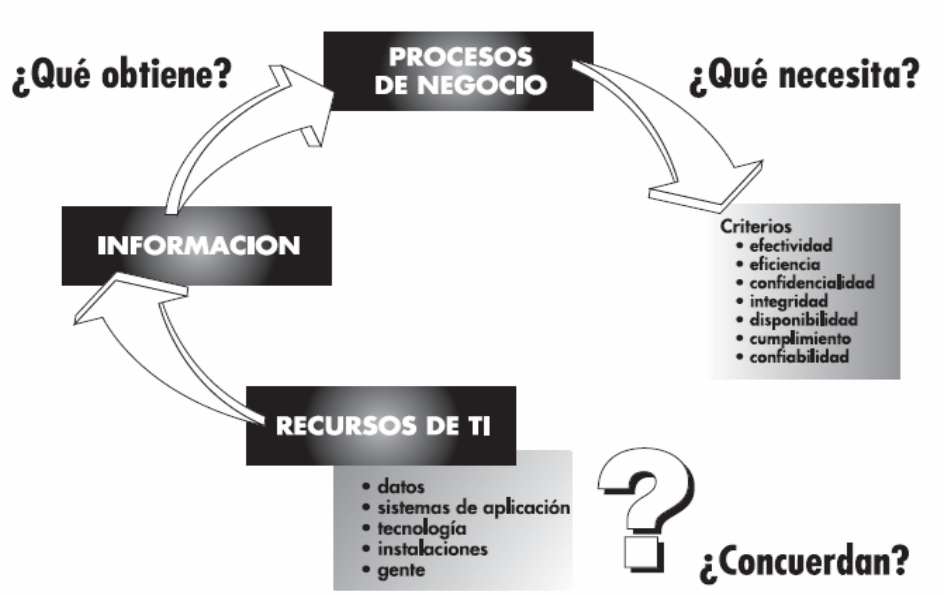


***Figura 8**. Otra forma de ver la relación de recursos TI con respecto a la entrega de servicios*

Con el fin de asegurar que los requerimientos de negocio para la información son satisfechos, deben definirse, implementarse y monitorearse medidas de control adecuadas para estos recursos. ¿Cómo pueden entonces las empresas estar satisfechas respecto a que la información obtenida presente las características que necesitan? Es aquí donde se requiere de un robusto marco referencial de Objetivos de Control para TI. El diagrama mostrado a continuación ilustra este concepto:



F***igura 9***. *Objetivos de control para TI (relación entre procesos de negocio, información y recursos TI)*

El *Marco de Referencia de COBIT* consta de Objetivos de Control de TI de alto nivel y de una estructura general para su clasificación y presentación. La teoría subyacente para la clasificación seleccionada se refiere a que existen, en esencia, tres niveles de actividades de TI al considerar la administración de sus recursos.

Comenzando por la base, encontramos las actividades y tareas necesarias para alcanzar un resultado medible. Las actividades cuentan con un concepto de ciclo de vida, mientras que las tareas son consideradas más discretas. Los procesos se definen entonces en un nivel superior como una serie de actividades o tareas conjuntas con “cortes” naturales (de control).

En el nivel más alto, los procesos son agrupados de manera natural en dominios. Su agrupamiento natural es denominado frecuentemente como dominios de responsabilidad en una estructura organizacional, y está en línea con el ciclo administrativo o ciclo de vida aplicable a los procesos de TI.

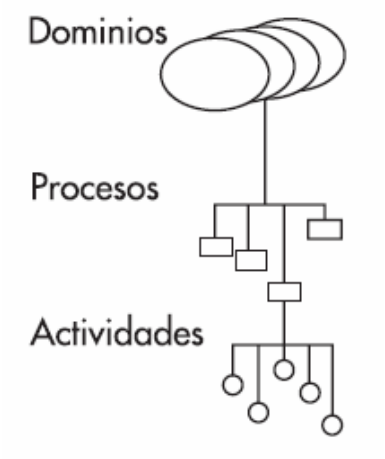


Figura 10. *Marco de referencia*

Por lo tanto, el *Marco de Referencia* conceptual puede ser enfocado desde tres puntos estratégicos: (1) Criterios de información, (2) recursos de TI y (3) procesos de TI. Estos tres puntos estratégicos son descritos en el Cubo COBIT que se muestra a continuación:

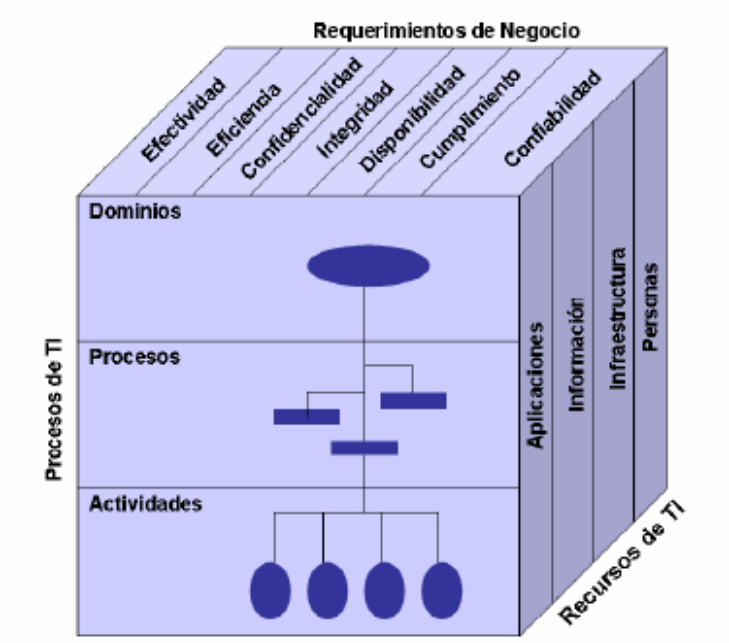


Figura 11*. Cubo COBIT y los 3 puntos estratégicos*

Con lo anterior como marco de referencia, los dominios son identificados utilizando las palabras que la gerencia utilizaría en las actividades cotidianas de la organización –y no la "jerga4" o terminología del auditor -. Por lo tanto, cuatro grandes dominios son identificados: planeación y organización, adquisición e implementación; entrega y soporte y monitoreo.

Las definiciones para los dominios mencionados son las siguientes:

- **Planeación y organización** : Este dominio cubre las estrategias y las tácticas y se refiere a la identificación de la forma en que la tecnología de información puede contribuir de la mejor manera al logro de los objetivos del negocio. Además, la consecución de la visión estratégica necesita ser planeada, comunicada y administrada desde diferentes perspectivas. Finalmente, deberá establecerse una organización y una infraestructura tecnológica apropiadas.

- **Adquisición e implementación** : Para llevar a cabo la estrategia de TI, las soluciones de TI deben ser identificadas, desarrolladas o adquiridas, así como implementadas e integradas dentro del proceso del negocio. Además, este dominio cubre los cambios y el mantenimiento realizados a sistemas existentes.

- **Entrega o soporte** : En este dominio se hace referencia a la entrega o distribución de los servicios requeridos, que abarca desde las operaciones tradicionales hasta el entrenamiento, pasando por la seguridad en los sistemas y la continuidad de las operaciones así como aspectos sobre entrenamiento. Con el fin de proveer servicios, deberán establecerse los procesos de soporte necesarios.

- **Monitoreo**: Este dominio incluye el procesamiento de los datos el cual es ejecutado por los sistemas de aplicación, frecuentemente clasificados como controles de aplicación.

Todos los procesos necesitan ser evaluados regularmente a través del tiempo para verificar su calidad y suficiencia en cuanto a los requerimientos de control.

Este dominio también advierte a la Administración sobre la necesidad de asegurar procesos de control independientes, los cuales son provistos por auditorías internas y externas u obtenidas de fuentes alternativas.

Es importante tener en cuenta que estos procesos de TI pueden ser aplicados en diferentes niveles de la organización. Por ejemplo, algunos de los procesos serán aplicados al nivel de la empresa, otros al nivel de la función de TI, otros al nivel del propietario de los procesos del negocio, etc.

Debe notarse además, que el criterio de efectividad en los procesos que planean o distribuyen soluciones para los requerimientos del negocio cubrirá algunas veces los criterios de disponibilidad, integridad y confidencialidad— en la práctica, éstos se han convertido en requerimientos del negocio. Por ejemplo, el proceso de "identificar soluciones" tiene disponibilidad, integridad y confidencialidad.

Es claro que todas las medidas de control no necesariamente satisfarán los diferentes requerimientos del negocio para la información en el mismo grado.

- **Primario:** es el grado en el cual se definen objetivos de control que impactan directamente los criterios de información considerados.

- **Secundario:** es el grado en el cual se definen objetivos de control que solo satisfacen una extensión pequeña o satisfacen indirectamente al criterio de información considerado.

- **En blanco:** podría ser aplicable. Sin embargo los requerimientos son satisfechos de una forma más apropiada por otro criterio en este proceso y/o en otro proceso.

En forma similar, todas las medidas de control no necesariamente impactarán a los diferentes recursos de TI en el mismo grado. Por consiguiente, el Marco de Referencia de COBIT indica específicamente la aplicabilidad de los recursos de TI que

son específicamente administrados por el proceso bajo consideración (no solamente los que toman parte en el proceso).

Esta clasificación se realiza con el Marco de Referencia de COBIT, basado sobre un riguroso proceso de recolección de ideas proporcionadas por investigadores, expertos y revisores, usando estrictas definiciones previamente indicadas.

En resumen, con el fin de proveer la información que la organización necesita para lograr sus objetivos, el Gobierno de TI debe ser entrenado por la organización para asegurar que los recursos de TI serán administrados por una colección de procesos de TI agrupados naturalmente.

Dentro de los procesos COBIT es muy importante analizar cada uno de los ofrecimientos de los proveedores o fabricantes de sistemas de seguridad informática y verificar que se adapten al modelo de la entidad. La selección tecnológica debe estar de acuerdo con las políticas de seguridad que se estén llevando a cabo por parte de la organización y a su organización de IT. Es sumamente importante documentar todos y cada uno de los procesos.

Quizá el punto más importante dentro de los procesos COBIT es la medición del riesgo informático, donde el riesgo se puede medir teniendo en cuenta el costo de la información, la amenaza que puede sufrir esta información (en su confiabilidad, disponibilidad e integridad) y las vulnerabilidades que aquejan a esta información.

De esta manera se puede caracterizar el riesgo para cada uno de los procesos y/o aplicaciones del negocio permitiendo:

Eliminar el riesgo: Evitando la amenaza o eliminando la vulnerabilidad.

Minimizar el riesgo: Implantando esquemas de seguridad adecuados.

Aceptar el riesgo permanente : Aceptar y manejar el riesgo intrínseco que no puede ser eliminado o minimizado.

Delegar el riesgo: Delegando a terceros la administración de un riesgo latente.

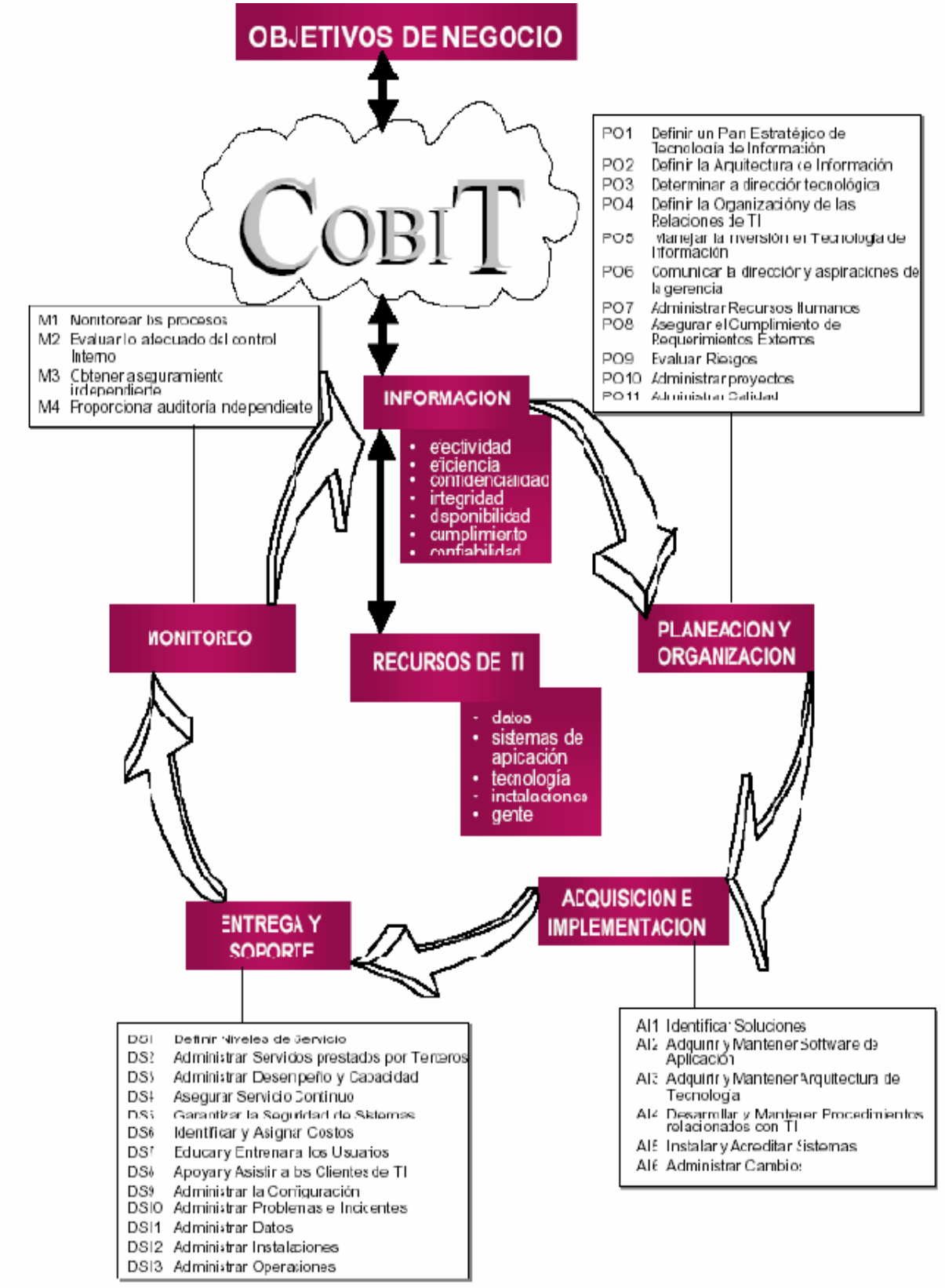


***Figura 12**. Metodología y objetivos de negocio (según COBIT)*

Para la implementación de esquemas y soluciones de seguridad informática es recomendable implementar un modelo de seguridad que integre las mejores prácticas de la norma ISO 27002 y del enfoque de ISACA (COBIT). Este modelo debe permitir a las entidades ajustarse adecuadamente a cualquiera de las metodologías que se establezcan y complementarlas de manera que se obtenga el mayor beneficio posible, y que la implantación se realice de manera gradual en todos sus procesos.

El modelo se encuentra claramente definido por el marco conceptual de la seguridad informática, la estructura de gestión de dicha área, y de un compromiso con la alta gerencia para ayudar en un programa de concienciación de todas las entidades.

En la siguiente figura que muestra el diagrama conceptual de la seguridad informática se pueden observar todo esto:

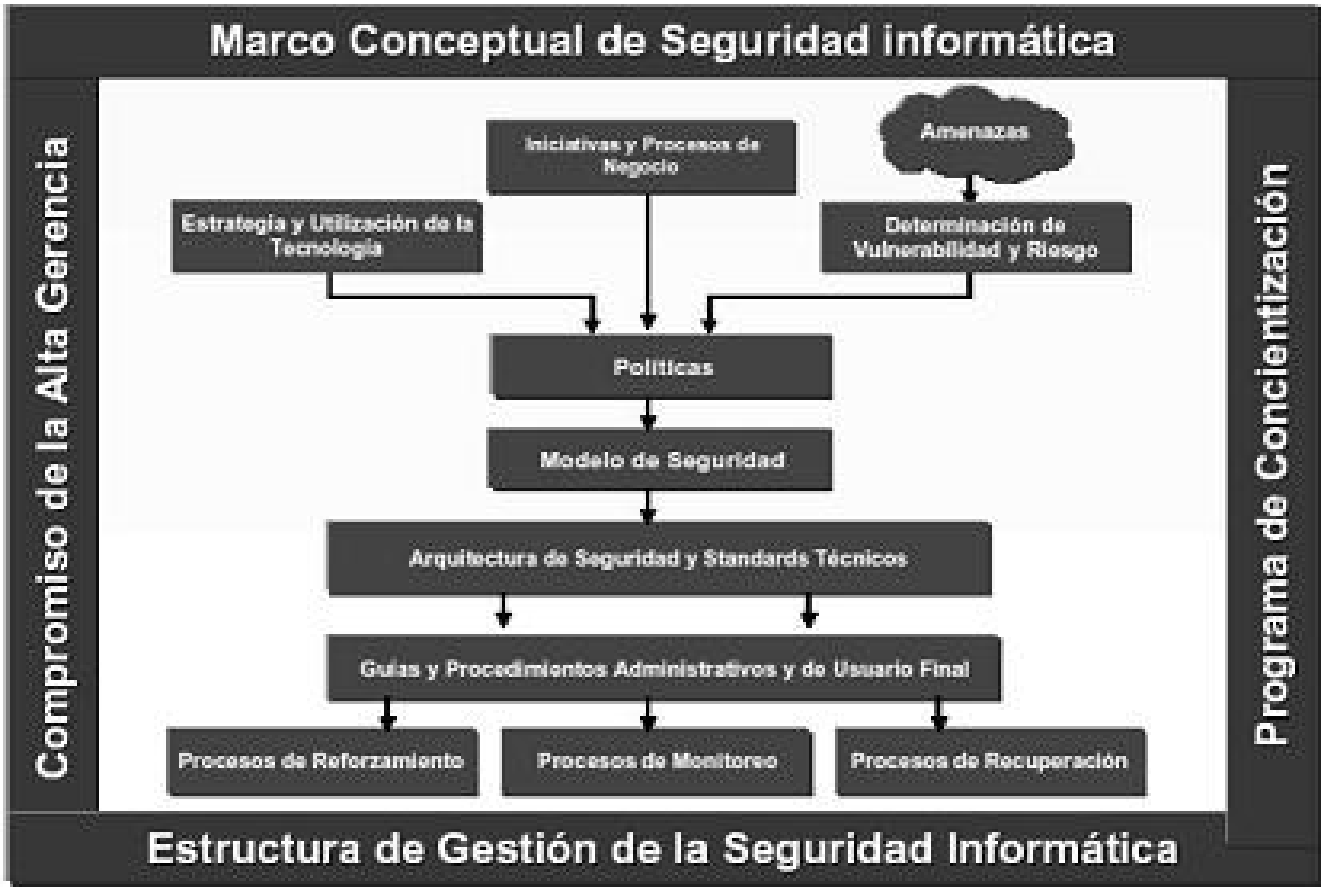


Figura 13. *Diagrama conceptual de seguridad informática*

En resumen, el contar con un sistema de gestión de seguridad informática, permite obtener, entre otros, los siguientes beneficios:

- Establecer un esquema preventivo ante los incidentes de seguridad.
- Mayor rapidez de respuesta ante un problema o situaciones de peligro para la organización
- Conocimiento de los puntos débiles de la infraestructura de IT y metodología para solventarlos.
- Distribución adecuada de los recursos humanos y técnicos para la protección de los procesos cruciales para la organización.
- Mantenimiento de los niveles de seguridad requeridos.
- Facilidad a la hora de obtener certificaciones de calidad y de seguridad.

BLOQUE II: LAS BASES DE DATOS

"El verdadero progreso es el que pone la tecnología al alcance de todos"
Henry Ford

2. LAS BASES DE DATOS

2.1 INTRODUCCIÓN

En la actualidad, en un mundo que cada vez avanza más deprisa, en el que se necesita la información de una manera rápida, confiable y segura se necesita alguna herramienta que ayude a este propósito. De esta necesidad de abstraer la realidad y extraer los datos más importantes, surgen las bases de datos. A partir de datos, atributos, tablas y las relaciones que se establecen entre ellos se puede obtener un diseño esquemático de una realidad concreta con la que poder trabajar de forma lo más eficiente posible asignando aquellas características y atributos que se crean necesarios.

En un principio para solventar este problema surgieron los sistemas de ficheros, que aunque fueron de gran utilidad, se tornaron a la larga obsoletos, puesto que su coste, lentitud y pocas funcionalidades con respecto a un gran volumen de información, hacía que gran parte de las necesidades por las que se habían creado quedasen descubiertas (como búsquedas, mantenimiento…), por tanto, las empresas y organizaciones buscaron otra solución para suplir estas deficiencias.

Así si por ejemplo se tiene una empresa de recursos humanos y se necesita tener un control exhaustivo de todos los datos relacionados con los empleados, una forma eficiente y útil de controlar dichos datos, podría ser la de contar con una base de datos que almacene todos los datos relativos a los mismos, evitando posibles fallos o errores, como duplicación de información, lentitud en búsquedas, seguridad (asignando privilegios o contraseñas), etc.

Actualmente las bases de datos son uno de los sistemas de gestión de la información más extendido, puesto que al reducir especialmente el tiempo de proceso, el coste en implantación de la misma, mejora en actualización, etc., las empresas han visto en éstas un sistema con el que mejorar notablemente la manera de gestionar la información que almacenan sus sistemas.

Para poder comenzar con la auditoría de la seguridad de una de base de datos concreta MySQL, que es la finalidad de este proyecto, en primer lugar, se necesitan tener unos conocimientos previos sobre esta área, puesto que si no conocemos el sistema será imposible realizar una auditoría de calidad.

Por ello este punto será una breve introducción a las bases de datos y a los sistemas de gestión de bases de datos, que ayudarán al auditor a conocer el entorno a auditar, puesto que tan importante es conocer la metodología necesaria para desarrollar la auditoría como el conocimiento profundo del entorno. ¿De qué serviría conocer los procedimientos necesarios para auditar si no sabemos qué áreas es importante auditar?

Asimismo, se darán nociones básicas sobre el entorno MySQL más generales (cada versión tiene sus especificidades) y su estructura genérica.

2.2. CONCEPTOS Y DEFINICIONES

El término “base de datos” aparece por primera vez en 1963 en un Congreso en Santa Mónica que incluía en el título “Data Base”.

Desde ese momento los avances en el campo teórico y en realizaciones de programas concretos con bases de datos han ido aumentando a lo largo de los años, hasta el momento actual, en el que es uno de los sistemas de gestión de la información más utilizado.

El concepto de Base de Datos ha ido variando a lo largo de los años, puesto que se ha ido avanzando y mejorando sus utilidades y funcionalidades a razón de las exigencias de la demanda. Teniendo en cuenta este hecho, se ha creído apropiado exponer la definición Base de Datos realizada por De Miguel y Piattini en su libro “Fundamentos y modelos de Bases de Datos” (1992):

> “Una base de datos es una colección o depósito de datos integrados, con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real: los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas, y su definición y descripción, únicas para cada tipo de datos han de estar almacenadas junto con los mismos. Los procedimientos de actualización y recuperación, comunes y bien determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de datos.”

Es decir, que una base de datos es un sistema formado por un conjunto de datos almacenados en un soporte no volátil lógicamente relacionados entre sí de manera que se controla el almacenamiento de datos redundantes.

Los datos son independientes de los programas que los usan, la definición y las relaciones entre los datos se almacenan junto a éstos y se puede acceder a los datos de diversas formas. También es un modelo o representación del mundo real o de una parte de éste y debe servir para un mejor conocimiento de una organización o actividad y para la adopción de decisiones.

Debido a la importancia que tienen las interrelaciones entre los datos para poder realizar un diseño de la realidad en la que poder recoger todos los supuestos que puedan darse, es necesario que la base de datos sea capaz de almacenar correctamente todas estas relaciones, así como los atributos de los datos, entidades y restricciones al modelo que se habrán de definir previamente al realizar el diseño y posteriormente al introducir los datos en la base de datos.

Como ya se comentó en la introducción de este apartado, esta característica es la diferencia fundamental entre las bases de datos y los ficheros, que no eran capaces de recoger estas interrelaciones.

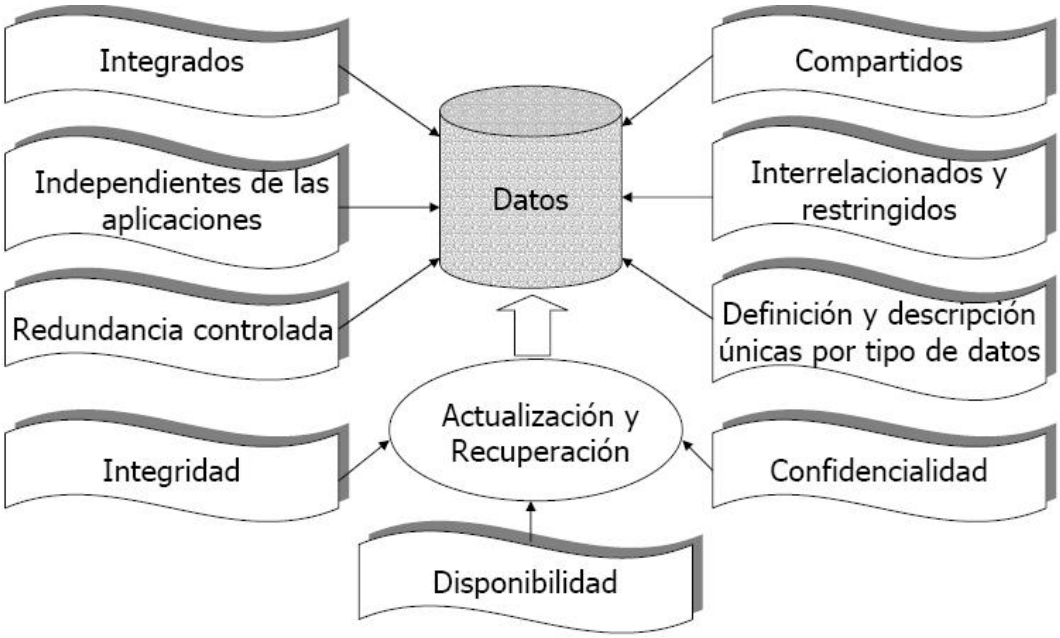


Figura 14. *Concepto y objetivos de las bases de datos*

2.2.1. Ventajas de las bases de datos

Son muchas las ventajas que ofrecen las bases de datos y los sistemas de gestión de bases de datos. Es materia de estudio en este apartado profundizar en cada una de ellas desde los diferentes prismas desde los que se puede observar**.**

Con respecto a las ventajas que se puede obtener en el uso de una base de datos desde el punto de vista de los datos es la independencia de los datos respecto a su tratamiento. Esto supone:

- La inclusión, eliminación o modificación de informaciones y datos no obliga a alterar los programas puesto que están almacenadas en estructuras independientes de éstos.

- Desaparece la redundancia, puesto que dejan de existir duplicidades de información que son inútiles e incluso pueden llegar a tornarse perjudiciales, puesto que puede haber incoherencia al ser tratados por el sistema como datos distintos. A todo esto hay que añadir a la lista de inconvenientes que la existencia de redundancia hace que el tamaño de la base de datos aumente de manera considerable. Por tanto la actualización de un dato será única, eliminando el inconveniente de tener que modificar la información en cada fichero o cada estructura múltiples veces. (Se hace necesario añadir que sí puede existir cierta redundancia física a efectos de eficiencia, pero no debería existir redundancia lógica entre los datos).

- En relación al punto anterior se obtiene por tanto, mayor eficacia en la recogida de datos. Al no existir redundancia los datos sólo se recogen una vez. Menor probabilidad de cometer errores y generar inconsistencias

- Mejor disponibilidad de los datos para los usuarios. Cada aplicación ya no es propietaria de los datos. Se comparten por el conjunto de aplicaciones.

- La descripción de los datos va almacenada junto a éstos con lo que se obtiene una mayor información acerca de la estructura.

- Reducción del espacio de almacenamiento. Almacenamiento único de cada dato.

Las ventajas con respecto a los resultados se podrían resumir en los siguientes puntos:

- Coherencia de los resultados: al recogerse los datos una sola vez en todos los tratamientos los resultados son homogéneos y comparables.

- Mayor valor informativo puesto que la base de datos no sólo recoge los datos de determinadas entidades sino las relaciones entre ellos. Permite tener una visión de conjunto de éstos y por tanto se tiene una mejor representación del mundo real.

Las ventajas respecto al punto de vista de los usuarios se podrían resumir en los siguientes puntos:

- Mayores facilidades para compartir datos por el conjunto de los usuarios. Al almacenarse conjuntamente, todos los usuarios tienen todos los datos disponibles (si tienen los permisos o contraseñas necesarios para ello concedidos por el administrador de la base de datos).

Esta característica es especialmente importante, puesto que los usuarios pueden manejar la base de datos actualizándola y bajo diferentes aplicaciones. Si tenemos en cuenta que los ficheros estaban especialmente diseñados para atender a una demanda concreta en una determinada aplicación, esta nueva posibilidad que ofrecían las bases de datos, ayudaba especialmente al conjunto de la organización en el desarrollo de su trabajo.

- Mayor flexibilidad para atender las demandas cambiantes en la organización puesto que la independencia entre datos y programas permite cambiar o ampliar la estructura de los datos sin modificar el programa o la aplicación que utiliza dicha base de datos.

- Se puede tener una interacción sencilla y más rápida a través de una interfaz de alto nivel y acceso a los datos de forma conversacional (como en Access que se utilizan cuadros de diálogo y menús sin tener que conocer perfectamente el lenguaje SQL, en el que el usuario de una manera sencilla puede representar las tablas e interrelaciones de una forma muy visual que le ayudará a su propósito).

- Al tener un control centralizado de la base de datos, el administrador de la misma, puede garantizar la observación de todas las normas aplicables para la representación de los datos. La normalización de formatos de los datos almacenados es deseable sobre todo como apoyo para el intercambio de información o migración de datos a otros sistemas. Del mismo modo, las normas para nombrar y documentar los datos son muy convenientes como ayuda para la compartición y compresión de la información.

2.2.2. Inconvenientes de las bases de datos

En todos los sistemas de gestión de información existen inconvenientes, y cómo no, las bases de datos también las tienen.

Sin embargo, en la utilización de una base de datos como almacenamiento y gestión de datos y relaciones, los inconvenientes que éstos acarrean casi siempre estarán muy por debajo de las ventajas y beneficios que podemos obtener al utilizarlas con respecto a cualquier sistema anterior.

También hay que añadir que en la actualidad se sigue investigando la manera de mejorar los inconvenientes tanto de las bases de datos como de los sistemas de gestión de bases de datos, y que muchos de los inconvenientes que aquí se recogen están en gran medida resueltos. Asimismo éstos pueden quedar obsoletos o no ser del todo exactos en un breve espacio de tiempo. No hay que olvidar que la tecnología avanza a pasos agigantados y que cada vez hay más personas que se encargan de estudiar la manera de mejorar las funcionalidades de las aplicaciones que manejan.

Se hace necesario desmentir la idea generalizada de personas poco versadas en el manejo de base de datos, que achacan los errores e incoherencias de información, baja utilidad, etc., a la base de datos, puesto que en la mayoría de los casos estos errores o fallos son la consecuencia directa de un mal diseño del problema, de los atributos, de los campos, entidades e interrelaciones de los datos y no de la propia base de datos, que lo "único que hace" es recoger la estructura que el administrador de la base de datos ha introducido.

A continuación se muestran algunos de estos inconvenientes:

- Implantación costosa en hardware y software, que puede venir derivada de la implantación de nuevos programas o aplicaciones, instalación de nuevos equipos de procesos de información, etc.

- La implantación de las bases de datos o de los sistemas de gestión de bases de datos puede llegar a tornarse larga y laboriosa, puesto que los usuarios y personal que van a utilizar esa base de datos necesitan unos conocimientos para administrar dicha base de datos. Por ello se necesitará personal cualificado. Esta necesidad puede hacer que el tiempo de implantación del sistema aumente. Además, las pruebas y detección de errores puede extenderse también en el tiempo.

- Necesidad de personal especializado (instalaciones complejas), por las razones anteriormente comentadas.

- Con respecto a las bases de datos y SGBD existe cierta falta de estándares con los que poder controlar la normalización para poder unificar los criterios de manejo.

- La rentabilidad es únicamente a medio-largo plazo, puesto que a corto plazo, es posible que los costes de implantación, desarrollo del sistema y formación del personal sean más altos que la rentabilidad que se obtiene en ese momento.

2.3. CONCEPTOS NECESARIOS EN UNA BASE DE DATOS

Como en todo sistema se necesitan unos conceptos básicos necesarios para poder comprender de manera completa y correcta lo que se ha venido comentando hasta este momento.

En una base de datos hay tres conceptos básicos necesarios imprescindibles. A saber:

- **Campo**. Cada campo contiene un dato individual acerca de la entidad de que se trate. Este debe ser único, y además tener un tipo de datos asociado (numérico, alfanumérico, alfabético, fecha…)

- **Registro**. Estructura formada por una serie de elementos que contienen información relativa a un mismo ente (o entidad; una persona, un objeto, un hecho). Los registros están formados por campos que son los que almacenan dicha información.

- **Tabla**. Es un conjunto de información homogénea, es decir, es el tipo de modelamiento de datos, en que se guarda en él los datos recolectados que previamente se habrán introducido en la base de datos. Las tablas están formadas por registros, que se corresponden con cada fila de la tabla (también denominada tupla) y por campos, que se corresponden con las columnas en una tabla.

Estos tres conceptos son básicos para comprender la manera en la que están almacenados los datos. Si por ejemplo se estuviese trabajando con una base de datos en una biblioteca, el siguiente gráfico podría mostrar una manera visual de ver cómo se relacionarían los campos, los registros y la tabla que almacena toda esa información[3].

***3**. Gráfico obtenido de la página http://www.elcuaderno.info/amanda/mysql/1.png*

Figura 15. *Concepto de base de datos (datos, registros y tabla).*

Después de haber resumido los tres pilares sobre los que se sustenta la información y los datos en las bases de datos, se pasará a desarrollar brevemente otros conceptos igualmente importantes, sin los cuales, las bases de datos no podrían existir:

2.4 ADMINISTRADOR DE LA BASE DE DATOS (DBA)

El DBA es la persona encargada de definir y controlar las bases de datos, además proporciona asesoría a los usuarios y ejecutivos que la requieran.

Las principales funciones del administrador son:

- El DBA deberá considerar qué información será necesario almacenar en la estructura de la base de datos después de haber analizado los requerimientos precisos y en consonancia con las funcionalidades requeridas por los usuarios.

- Los estándares por los que se va a regir la organización en cuanto a documentación de la base de datos y metodologías de diseño de la misma.

- La estrategia de transición del sistema en caso de tener que portarse a un nuevo sistema. El DBA deberá decidir sobre la posible puesta en marcha en paralelo del nuevo sistema con el antiguo, las fases de implantación del mismo, los controles necesarios, etc. Todas estas decisiones habrán de tomarse en función de los objetivos marcados y de forma que se cause el mínimo trastorno a los usuarios.

- Los permisos de explotación y uso, es decir, establecer la normativa necesaria para la utilización de la base de datos: solicitudes de acceso, actualizaciones, etc.

- Los aspectos relativos a la seguridad, incluidos los procedimientos de control y las auditorías.

- Mantenimiento rutinario. Algunos ejemplos de actividades rutinarias que el administrador de la base de datos debe revisar son:

    - Copia de seguridad periódica de la base de datos, bien sobre cinta o sobre servidores remotos, para prevenir la pérdida de datos en caso de desastres o imprevistos.

    - Asegurarse de que exista suficiente espacio libre en el disco duro para las operaciones normales y aumentar el espacio en el disco en caso de ser necesario.

    - Supervisión de los trabajos que se ejecuten sobre la base de datos y sobre todo asegurarse que el rendimiento no se degrade por tareas muy costosas realizadas por algunos usuarios.

2.5. SISTEMAS GESTORES DE BASES DE DATOS

En un sistema de base de datos tiene que existir una capa intermedia entre los datos que se almacenan en la base de datos, las aplicaciones y los usuarios que las utilizan. A este sistema se le denomina Sistema de Gestión de la Base de Datos (SGBD), actuando como intermediario entre los usuarios, los datos y las aplicaciones proporcionando medios para describir, almacenar y manipular los datos. De igual manera proporciona herramientas al administrador para gestionar el sistema, entre las que destacan las herramientas de desarrollo de aplicaciones, generador de informes, lenguajes específicos de acceso a los datos, como SQL (*Structured Query Language*) o QBE (*Query By Example*) (en bases de datos relacionales).

Un SGBD se puede definir como una agrupación coordinada de programas, procedimientos, lenguajes, etc. que suministra los medios necesarios para describir, recuperar y manipular los datos almacenados en la base de datos, manteniendo su integridad, confidencialidad y seguridad.

El principal objetivo de un SGBD es el de proporcionar el entorno adecuado para extraer, almacenar y manipular información de la base de datos. El SGBD gestiona de forma centralizada todas las peticiones de acceso a la base de datos, por lo que este sistema funciona como interfaz entre los usuarios y la base de datos. Además, el SGBD gestiona la estructura física de los datos y su almacenamiento, y es por ello que dicho sistema evita al usuario al necesidad de conocer exactamente la organización física de los datos y de crear algoritmos para almacenar, actualizar o consultar dicha información que está contenida en la bases de datos.

2.5.1 Características de un SGBD

De manera general todos los Sistemas Gestores de Bases de Datos (S.G.B.D.) presentan unas características comunes.

Algunas de estas características son:

- Mantener la independencia de los programas y la estructura de la base de datos. Así se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

- Asegurar la coherencia de los datos (no debe existir redundancia en los datos).

- Permitir a los usuarios almacenar datos, acceder a ellos y actualizarlos. El SGBD debe hacerlo de tal manera que sea transparente al usuario, ocultando la estructura física de los datos y su forma de almacenamiento.

- Contener un catálogo accesible por los usuarios en el que se almacenen las descripciones de los datos de forma centralizada.

- Garantizar que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna. Una transacción es un conjunto de acciones que cambian el contenido de la base de datos. Si la transacción falla durante su realización, la base de datos quedará en un estado inconsistente. Algunos de los cambios se habrán hecho y otros no, por lo tanto, los cambios realizados deberán ser deshechos para devolver la base de datos a un estado consistente.

- Permitir que varios usuarios tengan acceso al mismo tiempo a los datos, es decir, se debe permitir el acceso concurrente, impidiendo que haya datos inconsistentes.

- Garantizar la recuperación de la base de datos en caso de que algún suceso la dañe. El SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos llevándola a un estado consistente.

- Garantizar la seguridad de la base de datos. Solamente los usuarios autorizados pueden acceder a la base de datos (impidiendo los accesos no autorizados tanto accidentales como intencionados), permitiendo diferentes niveles de acceso.

- Garantizar la integridad de la base de datos.

- Mantener la disponibilidad continua: la base de datos debe estar siempre disponible para su acceso.

- Proporcionar herramientas de administración de la base de datos. Estas herramientas permiten entre otras funcionalidades: importar y exportar datos, monitorizar el funcionamiento y obtener estadísticas de utilización contra la base de datos, reorganizar índices y optimizar el espacio liberado para reutilizarlo.

- Integrarse con algún software gestor de comunicaciones. Muchos usuarios acceden a la base de datos desde terminales remotos, por lo que la comunicación con la máquina que alberga al SGBD se debe hacer a través de una red. Todas estas transmisiones de mensajes las maneja el gestor de comunicaciones de datos. Aunque este gestor no forma parte del SGBD, es necesario que el SGBD se pueda integrar con él.

- Garantizar la escalabilidad y elevada capacidad de progreso. El SGBD debe aprovechar todos los recursos de máquina disponibles en cada momento, aumentando su capacidad de proceso, conforme disponga de más recursos.

- Poseer un lenguaje de manipulación de datos, que permita la inserción, eliminación, modificación y consulta de los datos de la base de datos, de la forma más eficiente y conveniente posible.

- Permitir el almacenamiento de enormes cantidades de datos sin que el usuario perciba una disminución del rendimiento global del sistema. Para ello el SGBD debe utilizar: índices, partición de tablas, etc.

2.5.2 Lenguajes de los SGBD

Para proporcionar a los usuarios las diferentes facilidades, los SGBD deben ofrecer lenguajes especializados e interfaces apropiadas para cada tipo de usuario: administradores de la base de datos, diseñadores, programadores de aplicaciones y usuarios finales.

La interacción del usuario con la base de datos debe efectuarse a través de alguna técnica que haga fácil la comunicación, y que permita al usuario centrarse en el problema que desea solucionar, más que en la forma de expresarlo. La mejor forma de alcanzar este objetivo, es darle un lenguaje parecido al lenguaje natural, que le permita expresar de forma sencilla los requerimientos.

Los lenguajes que interactúan con los SGBD, se pueden clasificar en dos grandes grupos:

1. Los lenguajes de definición, manipulación y control: orientados hacia la función.

2. Orientados a los diferentes tipos de usuarios o de procesos.

Los SGBD deben ofrecer lenguajes e interfaces apropiados para cada tipo de usuario (diseñadores, programadores, usuarios, administradores de la base de datos, etc.). Estos lenguajes se pueden clasificar en tres:

1. DDL: Lenguaje de Definición de Datos, que define y mantiene la estructura de datos; es decir, creación, borrado y mantenimiento de base de datos, tablas, índices, claves, etc.

2. DML: Lenguaje de Manipulación de Datos, que obtiene, inserta, elimina y modifica los datos de la base de datos.

3. DCL: Lenguaje de Control de Datos, que sirve para trabajar en un entorno multiusuario, donde es importantísima la protección y seguridad de los datos, así como la compartición de datos entre usuarios.

2.5.3. Estructura de un SGBD

Los SGBD son paquetes de software muy complejos que deben proporcionar los servicios comentados anteriormente. Los elementos que componen un SGBD varían mucho unos de otros. El sistema operativo proporciona servicios básicos al SGBD, que están construidos sobre él.

Una de las características más importantes de los SGBD es la independencia entre programas y datos. Para asegurar esta independencia es necesario separar la representación física y lógica de los datos. Esta distinción fue reconocida oficialmente en 1978, cuando el comité ANSI/X3/SPARC (grupo de estudio del *Standard Planning and Requirements Committee* –SPARC- del ANSI -*American National Standards Institute*-, dentro del Comité X3, que se ocupa de ordenadores e informática) propuso una arquitectura en 3 niveles:

- Nivel interno o esquema físico: Es la representación del nivel más bajo de abstracción, en el que se describe en la estructura física de la base de datos: índices, estrategias de acceso, dispositivos de almacenamiento físico, etc. También se elige una implementación para cada una de las estructuras definidas en el esquema lógico.

- Nivel conceptual: Este nivel describe qué datos son almacenados realmente en la base de datos y las relaciones que existen entre los mismos.

- Nivel externo: Es el nivel más alto de abstracción, es lo que el usuario final puede visualizar del sistema terminado. En él se definen las vistas parciales de la base de datos para distintos grupos de usuarios. Cada vista parcial, a la que se denomina esquema externo, consiste en un conjunto de estructuras (estructuras derivadas) definidas a partir de las estructuras del esquema lógico.

El SGBD debe asegurar que estos niveles sean independientes entre sí; es decir, que los cambios realizados en cualquiera de ellos no afecten a los niveles superiores.

Los principales módulos del SGBD son:

- El compilador del Lenguaje de Definición de Datos (LDD). Actualiza las tablas del diccionario de datos o catálogo que contienen los metadatos, y chequea la sintaxis de las sentencias del LDD.

- El precompilador del Lenguaje de Manipulación de Datos (LMD). Convierte las sentencias del LMD en sentencias listas para su procesamiento por parte del compilador de lenguaje anfitrión, extrayendo dichas sentencias para que puedan ser procesadas de manera independiente por el compilador del LMD.

- El compilador del LMD. Chequea la sintaxis de las sentencias del LMD y se las pasa al procesador de consultas.

- El procesador de consultas. Realiza la transformación de las consultas en un conjunto de instrucciones de bajo nivel que se dirigen al gestor de la base de datos.

- El gestor de la base de datos. Es la interfaz con los programas de aplicación y las consultas de los usuarios. El gestor de la base de datos acepta consultas y examina los esquemas externo y conceptual para determinar qué registros se requieren para satisfacer la petición. Entonces el gestor de la base de datos realiza una llamada al gestor de ficheros para ejecutar la petición.

Los principales componentes del gestor de la base de datos son los siguientes:

- El gestor de transacciones. Realiza el procesamiento de las transacciones.

- El gestor de *buffers*. Transfiere los datos entre memoria principal y los dispositivos de almacenamiento secundario.

- El gestor de ficheros: Gestiona los ficheros en disco en donde se almacena la base de datos. Este gestor establece y mantiene la lista de estructuras e índices definidos en el esquema interno. Para acceder a los datos pasa la petición a los métodos de acceso del sistema operativo que se encargan de leer o escribir en los ficheros físicos que almacenan la información de la base de datos.

2.6 TIPOS DE ARQUITECTURAS DE BASES DE DATOS

2.6.1. SGBD centralizados

Un sistema de base de datos centralizado es aquel que se ejecuta en un único sistema computacional sin tener, para tal efecto, que interactuar con otros ordenadores. El rango de estos sistemas comprende desde los sistemas de bases de datos monousuario ejecutándose en ordenadores personales hasta los sistemas de bases de datos que se ejecutan en sistemas de alto rendimiento.

Normalmente los sistemas de base de datos monousuario no suelen proporcionar muchas de las facilidades que ofrecen los sistemas multiusuario. En particular no tienen control de concurrencia y tienen precarios o inexistentes sistemas de recuperación.

Dado que las máquinas en las cuales se utilizan los sistemas monousuario son comúnmente ordenadores de propósito general, la arquitectura de estas máquinas es siempre parecida (de 1 a 2 procesadores que comparten la memoria principal) por tanto los sistemas de base de datos que se ejecutan sobre estas máquinas no intentan dividir una consulta simple entre los distintos procesadores, sino que ejecutan cada consulta en un único procesador posibilitando así la concurrencia de varias consultas.

Este tipo de sistemas provocan la sensación de una mayor productividad (puesto que pueden ejecutar un mayor número de transacciones por segundo) a pesar de que cada transacción individualmente no se ejecute más rápido. Por el contrario las máquinas paralelas tienen un gran número de procesadores y los sistemas de base de datos que ahí se ejecutan siempre tenderán a paralelizar las tareas simples (consultas) que solicitan los usuarios.

2.6.2. Cliente/Servidor

Para poder desarrollar esta arquitectura se necesita un cliente inteligente que pueda solicitar servicios de un servidor en red. A una aplicación cliente / servidor se le puede pedir que realice validaciones o que muestre listas de opciones válidas, pero la mayor parte de las reglas de integridad de los datos y de negocio se imponen en la propia base de datos: relaciones, índices, valores predeterminados, rangos, disparadores, procedimientos almacenados, etc.

En el lado del servidor se encuentra un motor de servidor de bases de datos inteligentes. El servidor está diseñado para aceptar consultas SQL desde la aplicación frontal, generalmente en forma de llamadas a procedimientos almacenados que devuelven conjuntos de resultados claramente definidos y de ámbito limitado.

Generalmente, la aplicación cliente es responsable, al menos, de la administración de la conexión, la captura de los datos, la presentación de datos y la administración de los errores, mientras que el servidor, es el responsable de la administración inteligente de los recursos, la administración de la seguridad, la administración de los datos, de las consultas y sobre todo de la integridad de los datos.

Con el crecimiento de ordenadores personales (*Personal Computer* o PC) y de las redes de área local (LAN), se ha ido desplazando hacia el lado del cliente la funcionalidad de la parte visible al usuario de la base de datos (interfaces de formularios, gestión de informes, etc.), de modo que los sistemas servidores provean la parte subyacente que tiene que ver con el acceso a las estructuras de datos, evaluación y procesamiento de consultas, control de concurrencia y recuperación.

Los sistemas servidores pueden dividirse en dos tipos:

- **Los servidores transaccionales** : que sirven para agrupar la lógica del negocio en un servicio aparte. Proveen una interfaz a través de la cual los clientes pueden enviar peticiones como lo son los ODBC[4]).

- **Los servidores de datos** : los cuales envían datos a más bajo nivel y que descansan en la capacidad de procesamiento de datos de las máquinas clientes.

Existen 2 arquitecturas dominantes en la construcción de motores de base de datos cliente-servidor: los **motores multiprocesos** y los **motores multihilos**.

2.6.2.1. Motores de Bases de Datos Multiprocesos

Algunos motores de base de datos confían en múltiples aplicaciones para realizar su trabajo. En este tipo de arquitectura, cada vez que un usuario se conecta a la base de datos, ésta inicia una nueva instancia de la aplicación de base de datos.

---

4. Open DataBase Connectivity *que hace posible el acceso a cualquier dato de cualquier aplicación, sin importar qué SGBD se utilice*

Con el fin de coordinar a muchos usuarios que acceden a los mismos conjuntos de datos, estos ejecutables trabajan con un coordinador global de tareas que planifica operaciones para todos los usuarios.

La mayoría de los motores de base de datos multiprocesos fueron desarrollados antes de que los sistemas operativos soportaran características tales como hilos o planificación de tareas (*scheduling*). Como resultado de esto, el hecho de descomponer una operación significaba escribir un ejecutable distinto para manejar esta operación. Esta característica proporciona el beneficio de la fácil escalabilidad a través de la adición de más CPUs.

En un ambiente de multitarea el sistema operativo divide el tiempo de procesamiento entre múltiples aplicaciones asignándoles una porción de tiempo de CPU ("*slice*") a cada una. De esta manera siempre hay una sola tarea ejecutándose a la vez, sin embargo, el resultado es que múltiples aplicaciones aparenten estar corriendo simultáneamente en una sola CPU. La ventaja real viene cuando el sistema operativo cuenta con múltiples CPUs.

2.6.2.2. Motores de Bases de Datos Multihilos

Los motores de base de datos multihilos abordan el problema del acceso multiusuario de una manera distinta, pero con principios similares. En lugar de confiar en que el sistema operativo comparta los recursos de procesamiento, el motor toma la responsabilidad por sí mismo, lo que en la práctica se asocia a una mejor portabilidad del sistema. Las ventajas de este tipo de motores radican en una mayor eficiencia en el uso de recursos para determinadas plataformas y que no hay necesidad de un mecanismo de comunicación de interprocesos; de esta manera, la base de datos utiliza un elemento finito de trabajo, (el hilo) para una variedad de operaciones (instrucciones de usuarios, bloqueos de datos, administración del caché, etc.) en vez de utilizar aplicaciones especializadas para cada tarea.

2.6.3. SGBD Paralelos

Los sistemas paralelos de base de datos constan de varios procesadores y varios discos conectados a través de una red de conexión de alta velocidad.

Un sistema que procese un gran número de pequeñas transacciones puede mejorar su productividad realizando muchas transacciones en paralelo. Un sistema que procese transacciones más largas puede mejorar tanto su productividad como sus tiempos de respuesta realizando en paralelo cada una de las subtareas de cada transacción. Las ganancias en este tipo de SGBD se pueden dar en términos de velocidad (menor tiempo de ejecución para una tarea dada) y de la capacidad de procesar tareas más largas en el mismo tiempo.

Existen varios modelos de arquitecturas para máquinas paralelas, los más mencionados son:

• Memoria compartida: Todos los procesadores comparten una memoria común.

• Disco Compartido: Todos los procesadores comparten una disposición de discos común.

• Sin Compartimiento: Los procesadores no comparten ni memoria ni disco.

• Jerárquico: Se comparte tanto memoria como disco.

2.6.4. SGBD Distribuidos

En un SGBD distribuido, la base de datos se almacena en varios equipos que se pueden comunicar a su vez por distintos medios (desde redes de alta velocidad a líneas telefónicas). No comparten memoria ni discos y sus tamaños pueden variar tanto como sus funciones pudiendo abarcar desde un único equipo hasta grandes sistemas. (Se

denomina con el término de “emplazamientos” o “nodos” a todos aquellos equipos que pertenecen a un sistema distribuido.)

Las principales diferencias entre las bases de datos paralelas y las bases de datos distribuidas son las siguientes:

- Las bases de datos distribuidas se encuentran normalmente en varios lugares geográficos distintos, se administran de forma separada y poseen una interconexión más lenta.

- En un sistema distribuido se dan dos tipos de transacciones, las locales y las globales. Una transacción local es aquella que accede a los datos del único emplazamiento en el cual se inició la transacción. Por otra parte una transacción global es aquella que, o bien accede a los datos situados en un emplazamiento diferente de aquel en el que se inició la transacción, o bien accede a datos de varios emplazamientos distintos.

Un sistema de base de datos distribuido se conoce por:

• Los distintos emplazamientos están informados de los demás. Aunque algunas relaciones pueden estar almacenadas sólo en algunos emplazamientos, éstos comparten un esquema global común.

• Cada emplazamiento proporciona un entorno para la ejecución de transacciones tanto locales como globales.

2.6.5. SGBD Relacionales

Para entender qué son los SGBD Relacionales primero hay que comprender qué es el modelo de datos relacional.

El modelo de datos relacional fue presentado por E. F. **Codd** en 1970 y se basa en la representación del universo del discurso mediante el álgebra relacional. Codd, que era experto matemático, utilizó la terminología perteneciente a las matemáticas (teoría de conjuntos y lógica de predicados).

Las características principales del modelo son las siguientes:

1. Estructura los datos en forma de relaciones que se modelan mediante tablas de dos dimensiones. La estructura denominada "relación", permite representar tanto los objetos como las relaciones entre ellos.

2. Permite la incorporación de aspectos semánticos del universo del discurso mediante el establecimiento de reglas de integridad. Estas reglas permiten trasladar al esquema conceptual restricciones o comportamientos de los datos presentes en el universo del discurso que no se podrían modelar exclusivamente con tablas.

3. Está basado en un modelo matemático con reglas y algoritmos algebraicos establecidos, lo que permite el desarro0llo de lenguajes de acceso y manipulación potentes.

El esquema de una base de datos relacional consiste en la definición de una o más relaciones. Una base de datos relacional está constituida por una extensión para cada una de las relaciones de su esquema.

Para cada relación del esquema, se pueden especificar las siguientes propiedades:

- *Clave primaria*, que permite expresar la restricción de identificación.
- *Claves ajenas*, que permiten representar relaciones de "uno a muchos".
- *Valores no nulos*, para los atributos.

Los SGBD relacionales se caracterizan por las estructuras de datos, los operadores asociados y los aspectos semánticos. Los conceptos principales son:

1. Estructura de datos: Relaciones y Claves

- **Relación:** Es un subconjunto de un producto cartesiano entre conjuntos formados por atributos.

- **Atributos**: Son las columnas de la tabla. Corresponden a las propiedades de las entidades presentes en el universo del discurso que interesa almacenar en la base de datos. Cada uno de estos atributos puede tomar valores dentro de un rango determinado.

- **Dominio:** Rango de valores aceptable para un atributo dado. Dicho rango depende del atributo y condiciona los valores posibles dentro de cada celda de la tabla. Los valores que forman el dominio han de ser del mismo tipo e indivisibles.

- **Tuplas:** Es el nombre que recibe cada una de las filas de la tabla. Corresponden a cada una de las ocurrencias de la relación que representa la tabla.

- **Cardinalidad de la relación :** es el número de tuplas de la relación.

- **Grado:** Es el número de atributos que intervienen en la relación.

2.7. EL LENGUAJE SQL

El lenguaje SQL (inicialmente denominado SEQUEL) es el lenguaje estándar actual para los sistemas de bases de datos relacionales. Fue desarrollado originalmente por IBM (*International Business Machines*) a mediados de la década de los años setenta, e implementado por primera vez en un prototipo de IBM, el System R.

En el año 1986, el lenguaje SQL fue propuesto por ANSI (*American National Standards Institute*) como lenguaje relacional, y fue aceptado en 1987 por ISO como lenguaje estándar.

En sus orígenes fue un lenguaje de tipo lógico, basado en el Cálculo Relacional o de tuplas de E. F. Codd. Posteriormente fue incorporando operadores algebraicos.

El SQL actual en algunos sectores se considera un híbrido entre el Cálculo Relacional y el Álgebra Relacional. Así permite tanto el uso explícito de operadores algebraicos como el uso de expresiones lógicas de cualquier complejidad.

En la actualidad SQL es uno de los lenguajes con mayor aceptación dentro del mundo de la Informática. El motivo por el cual el lenguaje SQL se ha convertido en el estándar más utilizado en las bases de datos ha sido, por su parte, que dos de las principales compañías de software, IBM y Oracle, impulsaron firmemente su utilización, y por otra parte, por sus ventajas sobre otros estándares (proximidad al lenguaje natural, carácter declarativo, etc.).

El lenguaje SQL contiene instrucciones para la definición, la manipulación y el control o gestión de las bases de datos relacionales.

Las instrucciones SQL permiten al usuario interactuar con la base de datos. Estas instrucciones pueden ser:

- Instrucciones de manipulación de datos, que permiten la consulta y la actualización de la base de datos.

- Instrucciones de definición de datos, que permiten la definición y creación de objetos de la base de datos, así como la modificación de la definición de estos objetos.

- Instrucciones de gestión, que permiten la gestión de la sesión, de las transacciones, etc.

2.7.1. Conexión, sesión y transacción

Una conexión es la asociación entre un cliente SQL y un servidor de la base de datos SQL.

Los conceptos de cliente SQL y servidor SQL son similares a los conceptos de la arquitectura cliente/servidor. El cliente SQL está dedicado a proporcionar la interfaz de usuario (mediante la ejecución de los programas de aplicación de usuario final en algún entorno), y cuando es necesario solicita la ejecución de instrucciones SQL a un servidor SQL, que posteriormente le devolverá los resultados de su ejecución para que las procese.

Una transacción es una unidad lógica de procesamiento formada por una secuencia de instrucciones SQL. El procesamiento de las transacciones debe cumplir las propiedades de consistencia, atomicidad, persistencia y aislamiento.

Una sesión es el contexto en el cual un usuario ejecuta una secuencia de instrucciones SQL a través de una conexión.

2.7.2. Usuario y privilegio

La seguridad es un aspecto primordial en cualquier sistema. Para ello, para proporcionar seguridad, es importante la posibilidad de identificar a los usuarios que interactúan con él, de manera que sea posible saber si están autorizados a realizar la operación que se está solicitando.

En SQL las operaciones que realiza un usuario van acompañadas de un *identificador de autorización*, que permite al sistema distinguir al usuario de cualquier otro.

Los privilegios pueden estar asociados a componentes de objetos (como la modificación de una columna de una tabla), a objetos completos (como el privilegio de inserción en una tabla) o a la base de datos (como el privilegio de conexión a la base de datos).

3. SGBD MySQL

3.1 INTRODUCCIÓN

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de doce millones de instalaciones*.

MySQL en los últimos años ha tenido un crecimiento vertiginoso. Es una de las bases de datos de código abierto[5] más popular del mundo. Esta característica hace que todas aquellas personas que lo deseen pueden contribuir con ideas, elementos, mejoras o sugerir optimizaciones.

Y así es como MySQL se ha transformado de una pequeña base de datos a una completa herramienta. Su rápido desarrollo se debe en gran medida a la contribución de mucha gente al proyecto, así como la dedicación del "antiguo" equipo de MySQL (MySQL AB, que desde enero de 2008 fue adquirida por la empresa *Sun Microsystems*, que a su vez forma parte desde abril de 2009 de *Oracle Corporation*) .

A diferencia de los proyectos propietarios, en los que el código fuente es desarrollado por un número reducido de personas y se protege atentamente, los proyectos de código abierto no excluyen a nadie interesado en aportar ideas, si disponen de los conocimientos necesarios.

MySQL es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un gestor de bases de datos capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización. De hecho hay quien comenta que MySQL compite con sistemas RDBMS propietarios SQL Server, DB2 y antes con Oracle.

***.** *Dato extraído del seminario proporcionado en la página de MySQL (año 2009)*

5. *Todo el mundo puede acceder al código fuente, es decir, al código de programación de MySQL. (Véase el apartado "código abierto" para más información sobre éste)*

MySQL incluye todos los elementos necesarios para instalar el programa, reparar diferentes niveles de acceso de usuario, administrar el sistema y proteger los datos. También existe la posibilidad de crear la base de datos de una manera más intuitiva a través de la herramienta *Workbench* (cuya descarga está licenciada bajo GPL) que se ofrece en la página Web de MySQL, de tal manera que no es necesaria la utilización de la consola, que es mucho menos visual que la otra utilidad. Por otro lado, puede desarrollar sus propias aplicaciones de bases de datos en la mayor parte de lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos. MySQL utiliza el lenguaje de consulta estructurado (SQL).

Antes MySQL se consideraba como la opción ideal de sitios Web, sin embargo, ahora incorpora muchas de las funciones necesarias para otros entornos y conserva su gran velocidad. Además dispone de un sistema de permisos potente que actualmente incluye un motor de almacenamiento *InnoDb*[6] compatible con *ACID*[7]

También dispone de un sistema de asistencia eficiente y a un precio razonable, y, como ocurre con la mayor parte de las comunidades de código abierto, se puede encontrar una gran cantidad de ayuda en la Web, ya sea en foros de consulta o en páginas especializadas.

Con respecto a la licencia se dice que es un sistema dual: por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. MySQL es patrocinado por una empresa privada, que posee el *copyright* de la mayor parte del código.

Además de la venta de licencias privativas, la compañía ofrece soporte y servicios.

---

6. *INNODB es un tipo de tabla de MySQL que permite trabajar con transacciones, y definir reglas de integridad referencial. Ver las últimas modificaciones de la versión MySQL 5.5.*

7.*ACID son las propiedades que una base de datos debe cumplir para que el Sistema administrador de base de datos (DBMS) maneje correctamente la transaccionalidad. El término ACID viene de Atomicidad, Consistencia, Aislamiento, Durabilidad.*

3.2. PRINCIPALES CARACTERÍSTICAS DE MySQL

Según el manual que proporciona la página oficial de MySQL[8] las principales características del sistema de la base de datos son:

• Escrito en C y en C++

• Probado con un amplio rango de compiladores diferentes

• Funciona en diferentes plataformas.

• Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (via dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby,Gambas, REALbasic (Mac y Linux), FreeBASIC, Tc, etc.

• Proporciona sistemas de almacenamiento transaccional y no transaccional.

• Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.

• Relativamente sencillo de añadir otro sistema de almacenamiento

• Un sistema de reserva de memoria muy rápido basado en *threads*.

• Utilización de tablas temporales.

• Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.

• El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (*linkado*) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.

• Registros de longitud fija y longitud variable.

• Existe una interfaz ODBC (MyODBC) que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

• DELETE, INSERT, REPLACE, y UPDATE devuelven el número de filas que han cambiado (han sido afectadas).

---

8. http://www.mysql.com

• El comando específico de MySQL SHOW puede usarse para obtener información acerca de la base de datos, el motor de base de datos, tablas e índices. El comando EXPLAIN puede usarse para determinar cómo el optimizador resuelve una consulta.

• Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el *host*. Las contraseñas son seguras porque todo el tráfico de contraseñas está cifrado cuando se conecta con un servidor.

• Soporte a grandes bases de datos y para alias en tablas y columnas como lo requiere el estándar SQL.

En el manual que proporciona la página webWeb de MySQL también se relata brevemente la historia de cómo surge MySQL: en un principio sesus creadores empezaron con la intención de usar mSQL (también conocido como mini-SQL) para conectar a sus tablas utilizando sus propias rutinas rápidas de bajo nivel (ISAM). Sin embargo y tras algunas pruebas, los desarrolladores llegaron a la conclusión que mSQL no era lo suficientemente rápido o flexible para las necesidades que tenían. Esto provocó la creación de una nueva interfaz SQL para su base de datos pero casi con la misma interfaz API que mSQL. Esta API fue diseñada para permitir código de terceras partes que fue escrito para poder usarse con mSQL para ser fácilmente portado para el uso con MySQL.

La derivación del nombre MySQL no está clara. El directorio base de la empresa que desarrollaba la base de datos (MySQL AB) y un gran número de sus bibliotecas y herramientas tenían el prefijo “my”; por otro lado, la hija del de uno de los fundadores de la compañía también se llamaba My, por lo tanto saber cuál de los dos dio su nombre a MySQL todavía es un enigma.

---

* Mas información en: *http://en.wikipedia.org/wiki/MSQL*

A pesar de las obvias similitudes entre MySQL y SQL también existen diferencias. Algunas de ellas son las que se detallan a continuación:

- Para columnas VARCHAR, los espacios finales se eliminan cuando el valor se guarda. (Arreglado en la versión MySQL 5.0.3).

- En algunos casos, las columnas de tipo CHAR se convierten en columnas VARCHAR cuando define una tabla o altera su estructura. (Arreglado en la versión MySQL 5.0.3).

- Los privilegios para una tabla no se eliminan automáticamente cuando se borra una tabla. Debe usarse explícitamente un comando REVOKE para quitar los privilegios de una tabla.

- La función CAST( ) no soporta conversión a REAL o BIGINT.

- SQL estándar necesita que las cláusulas HAVING en un comando SELECT puedan referirse a columnas en la cláusula GROUP BY. Esto no se permite antes de la versión MySQL 5.0.2.

3.3. TIPOS DE TABLAS

Existen dos tipos de tablas de transacción segura Innodb y BDB. El resto ISAM, MyISAM, MERGE y HEAP son de transacción no segura. La elección del tipo de tabla adecuado puede afectar enormemente al rendimiento.

3.3.1 Tablas ISAM (Método de Acceso Secuencial Indexado)

Las tablas de tipo de *Index Sequential Access Method* o Método de Acceso Secuencial Indexado (ISAM) era el estándar antiguo de MySQL. Éstas fueron sustituidas por las tablas MyISAM en la versión 3.23. La principal diferencia entre las dos, es que el índice de las tablas MyISAM es mucho más pequeño que el de las tablas ISAM, de manera que un SELECT con un índice sobre una tabla MyISAM utilizará muchos menos recursos del sistema.

3.3.2 Tablas MyISAM

Las tablas MyISAM no son transaccionales ni permiten bloqueos de muy bajo nivel, pero incluyen la indexación por texto completo, la compresión y otras características.

El almacenamiento se gestiona en este tipo de tablas a través de dos archivos: un archivo de datos (con extensión .MYD) y un archivo de índices (con la extensión .MYI). El formato MyISAM es independiente de plataforma, por lo que podrá copiar tanto los datos como los índices desde un servidor en diferentes plataformas.

Las tablas MyISAM pueden contener tanto filas dinámicas como estáticas. MySQL decide cuál es el formato a emplear dependiendo de la definición de la tabla. El número de filas que puede haber en una tabla MyISAM está limitado principalmente por el espacio en disco disponible en el servidor de la base de datos y por el tamaño máximo de archivo que se pueda crear en el sistema operativo.

3.3.2.1. Tablas estáticas

Las tablas estáticas tienen longitud fija. Cada registro tiene asignado exactamente 10 Bytes.

Este tipo de tablas se caracterizan por:

- Ser muy rápidas.
- Resultan sencillas de almacenar en caché.
- Resultan sencillas para reconstruir tras un fallo.
- Requieren más espacio de disco.

3.3.2.2. Tablas Dinámicas

Las columnas de las tablas dinámicas tienen diferentes tamaños. Aunque este tipo de dato ahorra espacio, resulta sin embargo más complejo.

Las tablas de tipo dinámico presentan las siguientes características:

- Todas las columnas de cadena son dinámicas.
- Por regla general, ocupan mucho menos espacio de disco que las tablas fijas.
- Las tablas requieren un mantenimiento regular para evitar su fragmentación.
- No resulta tan sencillo de reconstruir tras un fallo, especialmente si las tablas están muy fragmentadas.

3.3.2.3 Tablas comprimidas

Las tablas comprimidas son tablas de solo lectura que utilizan mucho menos espacio en disco. Son ideales para su uso con datos comprimidos que son sólo de lectura y donde no exista mucho espacio disponible.

Las tablas comprimidas presentan las siguientes características:

- Las tablas son mucho más pequeñas.
- La carga de acceso es reducida, puesto que cada registro se comprime de forma separada,
- Cada columna se podría comprimir de forma diferente, utilizando distintos algoritmos de compresión.
- Se pueden comprimir formatos de tabla fija y dinámica.

3.3.3 Tablas Merge

Las tablas Merge son la fusión de las tablas MyISAM iguales. Por lo general se usa cuando las tablas MyISAM comienzan a resultar demasiado grandes.

Entre las ventajas de estas tablas se pueden mencionar las siguientes:

- Resultan más rápidas en determinadas situaciones.
- El tamaño de la tabla es más pequeño.

Con respecto a las desventajas este tipo de tabla destacan:

- Resultan mucho más lentas en búsquedas.
- El comando REPLACE (reemplazo) no funciona sobre ellas.

3.3.4 Tablas Heap

Las tablas *Heap* son el tipo de tabla que más rapidez proporciona puesto que se almacena en memoria y utilizan un índice asignado; sin embargo, en caso de un fallo de memoria se perderían todos los datos.

3.3.5 Tablas Innodb

Las tablas Innodb son tablas de transacción segura. En una tabla MyISAM, la tabla entera se bloquea (`LOCK TABLE`) al realizar funciones de inserción. Durante esa fracción de segundo, no se puede ejecutar ninguna otra instrucción sobre la tabla.

3.4 ALMACENAMIENTO DE DATOS EN MySQL

Una de las características de MySQL es el gran número de opciones disponibles dependientes del entorno del usuario. Desde el punto de vista del servidor, la configuración predeterminada se puede modificar para que funcione de manera correcta en gran variedad de plataformas hardware. Para el desarrollo de aplicaciones MySQL proporciona multitud de tipos de datos entre los que escoger al crear tablas para almacenar registros. Pero lo más importante, es que se puede escoger el tipo de tabla en el que se almacenan los registros. Se pueden incluso unir y enlazar tablas de distintos tipos en la misma base de datos.

La eficiente arquitectura de MySQL provee beneficios para el tipo de aplicación que se necesite como procesamiento de transacciones, situaciones de alta disponibilidad, etc. Todo para que el motor de la base de datos funcione de forma útil. Esto va de la mano con las ventajas de utilizar un conjunto de interfaces y servicios.

El programador de aplicaciones y el administrador de la base de datos pueden interactuar con la base de datos a través de los conectores Api’s y capas de servicios que están disponibles para los tipos de almacenamiento. Es posible cambiar entre los tipos de almacenamiento sin necesidad de modificar mucho código o procesos durante el cambio.

La capa superior se compone de los servicios que no son únicos de MySQL. Se trata de servicios que necesitan la mayoría de servidores y herramientas basadas en red, de arquitectura cliente/servidor: Administración de conexiones, autenticación, seguridad, etc. En la segunda capa reside la evaluación y optimización de sentencias, su almacenamiento para posterior reutilización y todas las funciones internas al sistema, como gestión de fechas, horas, funciones matemáticas o cifrado (la funcionalidad se encuentra en esta capa). Por último, la tercera capa

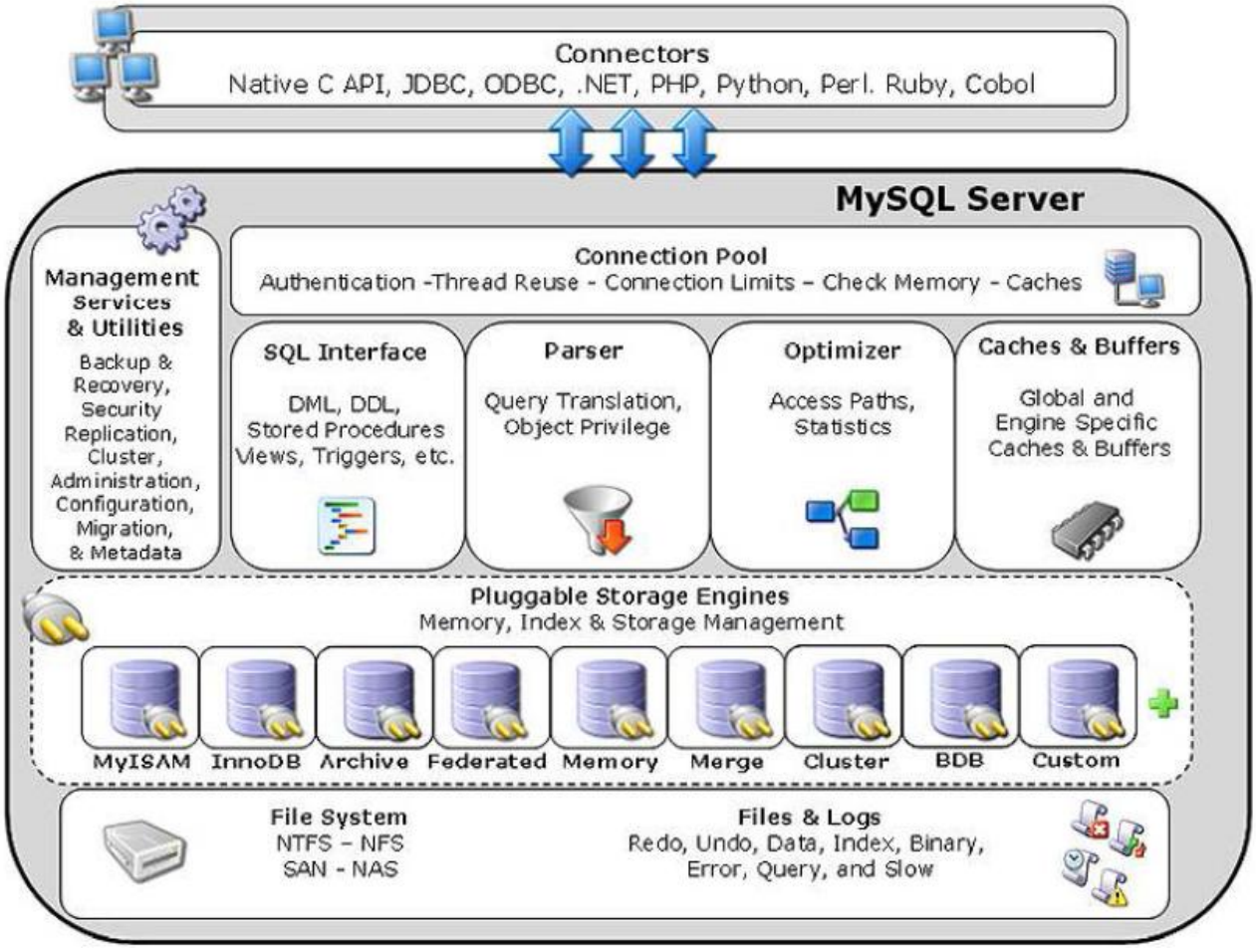


Figura 16. *Arquitectura de almacenamiento en MySQL (fuente: manual de MySQL)*

3.5. ESTRUCTURA INTERNA DE MySQL

Los datos en MySQL son almacenados en la base de datos utilizando tablas. Cada tabla está compuesta por un número de columnas. Las tablas se pueden relacionar entre sí a través de las columnas que las componen.

Para evitar que se incumplan las relaciones, utiliza la integridad referencial, que es la manera de comprobar y asegurar que todas las relaciones se cumplen.

MySQL utiliza la restricción de tabla a través de asignar claves a aquellos campos que sean prioritarios: claves primarias y claves ajenas (`PRIMARY KEY` y `FOREIGN KEY`). La clave primaria estará compuesta por las columnas que hacen a cada tupla de la tabla distinta y única; mientras que las claves ajenas se utilizan para especificar las relaciones entre tablas, de manera que un conjunto de columnas declaradas como clave ajena de una tabla, deben tener valores tomados de la clave primaria de otra tabla.

En versiones posteriores de MySQL (4.1, 5.0, 5.1,…) se utilizan también disparadores o *triggers* que son procedimientos de control que son ejecutados cuando se produce un determinado evento en la base de datos. Estos disparadores pueden utilizarse para reforzar la seguridad de la base de datos, así como su integridad.

MySQL –como se verá más detalladamente en el apartado de auditoría- puede otorgar privilegios a aquellos usuarios que los necesiten, así si un usuario quiere ejecutar la sentencia INSERT, SELECT, UPDATE, etc. , debe tener asignado ese privilegio.

Existe un diccionario de datos, que puede ser muy útil para el administrador y auditores del sistema, denominado `INFORMATION_SCHEMA`, que contiene datos relevantes de la base de datos.

A partir de la versión 5.0.10. de MySQL, se ofrece una nueva funcionalidad que es muy útil para el auditor y para el administrador de la base de datos: a través de `INFORMATION_SCHEMA` se puede extraer mucha información:

De la tabla INFORMATION_SCHEMA TRIGGERS se puede observar la tabla TRIGGERS que proporciona información acerca de disparadores. (Se debe tener el permiso SUPER para ver esta tabla.).

De la tabla INFORMATION_SCHEMA VIEWS se puede observar la tabla VIEWS que proporciona información acerca de las vistas en las bases de datos.

De la tabla INFORMATION_SCHEMA KEY_COLUMN_USAGE se puede observar la tabla KEY_COLUMN_USAGE que describe qué columnas clave tienen restricciones.

De la tabla INFORMATION_SCHEMA TABLE_CONSTRAINTS se puede observar la tabla TABLE_CONSTRAINTS que describe qué tablas tienen restricciones.

De la tabla INFORMATION_SCHEMA COLUMN_PRIVILEGES se puede observar la tabla COLUMN_PRIVILEGES que proporciona información acerca de permisos de columnas. Esta información viene de la tabla de permisos mysql.columns_priv.

De la tabla INFORMATION_SCHEMA TABLE_PRIVILEGES se puede observar la tabla TABLE_PRIVILEGES que proporciona información de permisos de tablas. Esta información viene de la tabla de permisos mysql.tables_priv.

De la tabla INFORMATION_SCHEMA SCHEMA_PRIVILEGES se puede observar la tabla SCHEMA_PRIVILEGES que proporciona información acerca del esquema de permisos (base de datos). Esta información viene de la tabla de permisos mysql.db.

De la tabla INFORMATION_SCHEMA USER_PRIVILEGES se puede observar la tabla USER_PRIVILEGES que proporciona información acerca de permisos globales. Esta información viene de la tabla de permisos mysql.user

De la tabla `INFORMATION_SCHEMA STATISTICS` se puede observar la tabla `STATISTICS` que proporciona información acerca de los índices de las tablas.

Como se puede observar, a partir de estas tablas, se pueden comprobar multitud de factores que influyen de forma definitiva en la administración de la base de datos, configuración y seguridad.

3.5.1. Administración de MySQL

Las tareas administrativas más importantes de MySQL deberían incluir como mínimo el inicio y desconexión del servidor, el mantenimiento de cuentas de usuario, el mantenimiento del archivo del registro, el ajuste del servidor, el uso de múltiples servidores y las actualizaciones de MySQL.

El directorio de datos: Se llama directorio de datos en MySQL al directorio que contiene todas las bases de datos y tablas manejadas por el servidor.

Una de las características particulares de MySQL es el hecho de que cada base de datos se corresponde con un directorio bajo el directorio de datos, y las tablas dentro de una base de datos se corresponden con los archivos en el respectivo directorio relativo a la base de datos.

La localización del directorio de datos en Windows suele ser ***c:\mysql\data.*** En UNIX/LINUX para una distribución fuente suele ser ***/usr/local/data/var.***

Múltiples servidores: Normalmente suele ejecutarse un solo servidor MySQL en una máquina determinada, pero en algunas ocasiones puede ser necesaria la ejecución de varios servidores. Los casos habituales aparecen cuando se quiere probar una nueva versión del servidor mientras sigue ejecutándose el servidor actual en producción. Un volumen de trabajo alto también puede hacer necesaria la presencia de varios servidores en ejecución simultánea. Por otro lado, cuando se trabaja en Internet, los proveedores de servicios suelen ofrecer a sus clientes su propia instalación de MySQL, lo que involucra servidores separados.

Para ejecutar diferentes versiones de servidor es necesario instalarlas en distintas localizaciones. Para las distribuciones binarias, lo que suele hacerse es instalarlas bajo un nombre de directorio que incluye la versión del servidor, con lo que diferentes versiones irán a distintos directorios.

Informes de bugs: La utilidad ***mysqlbug*** permite crear y enviar un informe a la lista de correo de MySQL cuando se descubre un problema con MySQL (informe de bug). Su sintaxis es la siguiente:

mysqlbug [dirección]

Si se especifica una dirección de e-mail, el informe es enviado a esa dirección, y si no se especifica la dirección, el informe es enviado a la lista de correo de mysql.

La utilidad *mysqldump* escribe el contenido de las tablas en la base de datos en ficheros de texto que pueden ser utilizadas como copias de seguridad de bases de datos, para mover bases de datos a otro servidor o para crear una base de datos de pruebas basada en una ya existente.

3.5.2. Seguridad en MySQL

Las tareas administrativas más importantes de MySQL incluyen también las relativas a la seguridad e integridad de la base de datos.

Evidentemente, el administrador de la base de datos es el responsable de que el contenido de ésta sólo sea accesible para aquellos que estén autorizados a utilizarla y hasta el nivel de autorización de que disponen:

- La seguridad interna normalmente se refiere a tareas de protección del directorio de datos de MySQL de los ataques de los titulares de otras cuentas en el mismo servidor.

- La seguridad externa es referente a la protección contra los ataques al servidor MySQL procedentes de las conexiones de red que realizan los clientes que conectan desde fuera intentando acceder a la base de datos.

3.5.2.1. Seguridad externa

En cualquier sistema es necesaria la seguridad y esta necesidad se acentúa cuando el sistema es multiusuario. Es necesario -como mínimo- establecer la autentificación y administración de usuarios, la administración de privilegios y funciones, la administración de contraseñas de usuario y el establecimiento de límites de recursos de la base de datos.

Cualquier usuario que intente conectarse a la base de datos debe hacerlo con un nombre de usuario determinado para que MySQL autentique que dicha persona está autorizada a usar la cuenta. Cada usuario debe obtener acceso a MySQL a través de una cuenta de inicio de sesión que establece su capacidad para conectarse.

A continuación, esta cuenta de inicio de sesión se asigna a una cuenta de usuario de MySQL que se utiliza para controlar las actividades realizadas en la base de datos (validación de permisos). Por tanto, se asigna un único inicio de sesión a cada cuenta de usuario creada en cada base de datos a la que el inicio de sesión tiene acceso. Los permisos de acceso y los privilegios de cada usuario los regula MySQL en las tablas de concesión. Existen 5 tablas de concesión, cuyos nombres son:

user (contiene los usuarios que pueden conectar con el servidor con cualquier privilegio que tengan), ***db*** (contiene los privilegios de nivel de base de datos), ***tables_priv*** (contiene los privilegios de nivel de tabla), ***columns_priv*** (contiene los privilegios de acceso de columna) y ***host*** (se usa en combinación con *db* para controlar los privilegios de acceso de *hosts* particulares al mejor nivel que esa posible sólo en la tabla *db*).

Cualquier privilegio de la tabla *user* es global, es decir, se aplica a todas las bases de datos. Esta tabla contiene la lista de usuarios a los que les está permitido conectarse con el servidor y sus contraseñas.

Cualquier privilegio de la tabla *db* es un privilegio de base de datos, es decir, se aplica a todas las tablas de la base de datos. La tabla *db* contiene bases de datos y la lista de usuarios a los que les está permitido acceder a los privilegios.

a) Privilegios administrativos: Los privilegios administrativos se aplican a las operaciones que administran el servidor o a la capacidad de un usuario de conceder privilegios. La relación siguiente resume los tipos de privilegios administrativos:

FILE: Permite leer y escribir archivos en el servidor.

PROCESS: Permite revisar información sobre los hilos ejecutados o eliminados en el servidor.

RELOAD: Permite recargar las tablas cedidas o vaciar los registros, la caché host o la caché de tablas.

SHUTDOWN: Permite cerrar el servidor.

ALL (u ALL PRIVILEGES): Asigna todos los permisos.

USAGE: Privilegio especial "sin privilegios".

b) Gestión de los privilegios: Las sentencias esenciales en MySQL para la gestión de privilegios son GRANT y REVOKE. La sentencia GRANT crea usuarios y regula sus privilegios y la sentencia REVOKE elimina los privilegios. Ambas sentencias afectan a las tablas de concesión. Cuando se usa GRANT para un usuario, se crea una entrada para ese usuario en la tabla *user*, y si la sentencia especifica cualquier privilegio global (administrativo o aplicable a todas las bases de datos) también se registran en *user*. Si se especifica una base de datos, tabla o privilegios de columna, se registran en las tablas *db, tables_priv y columns_priv*.

La sintaxis de GRANT:

```
GRANT tipo_priv [(lista_col)] [, tipo_priv [lista_col]... ON {*.* | * | nombre_db.* | nombre_db.nombre_tabla | nombre_tabla} TO usuario IDENTIFIED BY "contraseña" [, usuario IDENTIFIED BY "contraseña"]... [WITH GRANT OPTION]
```

El argumento *tipo_priv* especifica los privilegios que se concederán y *lista_col* especifica las columnas (separadas por comas) a las que se concederán estos privilegios en su caso.

La cláusula ON especifica el nivel (amplitud) a que se aplican los privilegios.

Los privilegios pueden ser globales o aplicables a todas las bases de datos y a todas las tablas, específicos de una base de datos *(nombre_bd.nombre_tabla* o *nombre_tabla*). Cuando en una tabla se utiliza cláusula ON, los privilegios se pueden hacer por columna específica nombrando una o más columnas separadas por comas en la cláusula *lista_col*.

La cláusula TO especifica el usuario para quien son los privilegios. Cada usuario consiste en un nombre de usuario y nombre de host mediante la especificación *nombre_usuario@nombre_host* y opcionalmente una cláusula IDENTIFYED BY que asigna una contraseña al usuario y que debe especificarse en texto plano.

```
REVOKE tipo_priv [(lista_col)] [, tipo_priv [lista_col]... ON {*.* | * | nombre_db.* | nombre_db.nombre_tabla | nombre_tabla} FROM usuario [, usuario]...
```

La sentencia GRANT concede privilegios insertando valores en las tablas de concesión, por lo tanto será posible asignar esos privilegios realizando las inserciones adecuadas en las tablas correspondientes mediante la sentencia INSERT, lo que exige conocer perfectamente la estructura de todas las tablas de concesión.

A veces, determinadas versiones de MySQL tienen más de 17 columnas en la tabla USER, por lo que es conveniente utilizar SHOW COLUMNS FROM USER antes de

utilizar INSERT, para estar seguros del número de columnas de la tabla y de los privilegios que se insertan.

Pero los privilegios pueden afectar a otras tablas distintas de *user*. Por ejemplo, cuando los privilegios a asignar a un usuario no son globales, se almacenarán en otras tablas de concesión, aunque tendremos que crear una entrada en la tabla *user* para que el usuario pueda conectar.

Ante la importancia de conocer con precisión la estructura de las tablas de concesión, es conveniente utilizar SHOW TABLES y SHOW COLUMNS para cotejarlas.

Las tablas de concesión pertenecen a la base de datos de nombre *mysql* que se genera por defecto en la instalación. Por lo tanto, la primera tarea será poner en uso la base de datos y mostrar todas sus columnas a través de: USE mysql; SHOW TABLES. A continuación se utilizará SHOW COLUMNS para algunas tablas de concesión: SHOW COLUMNS FROM db; SHOW COLUMNS FROM user; SHOW COLUMNS FROM columns_priv; SHOW COLUMNS FROM host.

3.5.2.2. Seguridad interna

Como se comentaba en un apartado anterior la seguridad interna era la que hacía referencia a las tareas de protección del directorio de datos de MySQL de los ataques de los titulares de otras cuentas del mismo servidor. Por lo tanto, hay que impedir que otros usuarios del *host* servidor tengan acceso de escritura o lectura a los archivos del directorio de datos. Por otra parte, los *logs* de seguimiento generales y de actualización deben ser seguros, ya que contienen los textos de las consultas.

Protección del directorio de datos: El directorio de datos de MySQL contiene todas las bases de datos y tablas manejadas por el servidor. Una de las características particulares de MySQL es el hecho de que cada base de datos corresponde con un directorio bajo el directorio de datos y las tablas dentro de una base de datos se corresponden con los archivos en el respectivo directorio relativo a la base de datos.

Por tanto, dada la información que contiene, es crucial la seguridad del directorio de datos.

Mantenimiento de archivos de log: Cuando se comienza el trabajo en el servidor MySQL hay que asegurarse de que se activan los archivos de *log* de rastreo que permitirán hacer un seguimiento posterior de toda la actividad en el servidor y a la vez de todos los usuarios que han realizado cada actividad.

En MySQL existe un archivo general *log* que informa de las conexiones del cliente, las consultas y otras acciones y que permite seguir la pista de la actividad del servidor, de los usuarios conectados, de su origen y de su actividad.

También existe un log de actualización que hace el seguimiento de las consultas que modifican la base de datos y que contiene un registro para cada consulta del tipo DELETE, INSERT, REPLACE, CREATE TABLE, DROP TABLE, GRANT y REVOKE. Estas consultas aparecen en el fichero log como sentencias SQL, lo que permitirá restaurar las tablas de la base de datos en caso de que una caída del sistema con pérdida de información. Se pueden usar los log de actualización como entrada para MySQL para restaurar la base de datos.

La sintaxis para habilitar el log general al iniciar el servidor MySQL es la siguiente:

mysqld --log [=nombre_ruta]

La sintaxis para habilitar el log de actualización al iniciar el servidor MYSQL es la siguiente:

mysqld --log-update [=nombre_ruta]

La opción *nombre_ruta* permite especificar un nombre y ruta en el sistema para el fichero log (distinto del nombre por defecto).

Estas opciones también se pueden especificar con sintaxis semejante cuando se inicia el servidor MySQL mediante ***safe_mysql*** o ***mysql.server.*** Es muy conveniente habilitar los archivos log siempre que se inicie el servidor.

Copias de seguridad**:** En todo momento se debe disponer de una copia de seguridad de las bases de datos en MySQL. Pueden existir colapsos del sistema u operaciones administrativas poco prudentes como DROP DATABASE o DROP TABLE que accidentalmente eliminen o dañen una base de datos. Es necesario estar preparados para este tipo de contingencias habilitando copias de seguridad que permitan la recuperación de la información si es necesario.

En MySQL hay dos métodos para hacer copias de seguridad de las bases de datos. La más conveniente es usar la utilidad "*mysqldump*". La otra opción es realizar una copia directa de los archivos de la base de datos mediante las órdenes de copia del sistema operativo *copy*, *cp, tar* o *cpio.*

La sentencia *mysqldump* es más lenta que la copia directa, pero es mucho más segura porque opera en cooperación con el servidor MySQL evitando los problemas de la realización de copias mientras se está produciendo una operación en la base de datos. Además, genera archivos de texto transportables a otras máquinas que harán posible la recuperación de la información en otro sistema distinto del nuestro.

Copias de seguridad con MYSQLDUMP (backups): La utilidad *mysqldump* escribe el contenido de las tablas de la base de datos en ficheros de texto que pueden usarse para copias de seguridad de bases de datos, para mover bases de datos a otro servidor o para crear una base de datos de pruebas basada en una ya existente. Su sintaxis es la siguiente:

mysqldump [opciones] nombre_base_de_datos [nombre_tabla]...

Cuando no se especifica el nombre de tabla o tablas, el proceso afecta a todas las tablas de la base de datos.

La sintaxis más habitual para volcar la base de datos completa en un fichero de backup es la siguiente:

mysqldump nombre_base_de_datos > fichero_de_backup

No obstante, suele utilizarse la opción `--opt,` que permite afinar la copia mediante el uso de instrucciones DROP TABLE, LOCK TABLE y otras previas a la copia para que ésta sea más segura.

Recuperación de una base de datos: La primera cuestión a tener en cuenta ante la recuperación de una base de datos es el nivel de recuperación que se desea. Para recuperar la base de datos incluyendo las tablas de concesión es necesario comenzar arrancando el servidor con la opción `--skip.grant-tables` como sigue:

```
mysqld --skip-grant-tables
```

Además, después de restaurar las tablas será necesario indicar al servidor que cargue las tablas de concesión y empiece a usarlas mediante la sentencia:

```
mysqadmin flush-privileges
```

Para restaurar las tablas se pueden utilizar los archivos de backup generados por *mysqldump* como entrada para *mysql* mediante la sintaxis:

```
mysql nombre_base_datos < fichero_de_backup
```

Si la copia de seguridad de la base de datos se ha realizado con órdenes de sistema operativo como *copy*, *cp* o *tar*, se realizará la recuperación copiando los ficheros a los directorios adecuados. En este caso es necesario conocer muy bien la estructura de directorios de las bases de datos de MySQL para volver a situar cada fichero en su sitio. Además, el servidor ha de desconectarse antes de iniciar la copia y reiniciarlo al finalizar la tarea de copia.

Reparación de tablas de una base de datos: Hay circunstancias externas como los apagones de luz que pueden provocar inconsistencias en las tablas de las bases de datos. Los fallos de hardware y los cierres indebidos de servidor también pueden provocar problemas en las tablas de las bases de datos MySQL.

Ante estas circunstancias son necesarios métodos y estrategias que solucionen estos problemas. El camino general a seguir comienza con la verificación de las tablas de la base de datos en busca de errores. En MySQL existen las utilidades *myisamchk* e *isamchk* que verifican el estado de las tablas de las bases de datos.

Una vez que se ha detectado una tabla con errores, será necesario hacer copia de sus archivos antes de repararla, por si el proceso de reparación nos condujese a un estado peor. Las copias se hacen con instrucciones de copia del sistema operativo (copy, cp, tar...) o con la utilidad *mysqldump*.

Posteriormente, se intenta reparar la tabla utilizando las opciones adecuadas de las utilidades *myisamchk*.

Por último, si falla la recuperación, será necesario restaurar la tabla desde las copias de seguridad o desde los logs de actualización:

Verificación de tablas con MYISAMCHK e ISAMCHK: Estas dos utilidades de verificación de tablas funcionan de modo parecido y únicamente se diferencian en el tipo de tablas para las que se aplican. La utilidad *myisamchk* se usa para verificación de tablas MyISAM, mientras que *isamchk* se usa para tablas ISAM. Para distinguir un tipo de tablas del otro se observa la extensión de los ficheros. En el caso de tablas MyISAM los ficheros tienen como extensión .MYI, mientras que los ficheros relativos a las tablas ISAM tienen como extensión .ISM. La sintaxis es la siguiente:

```
myisamchk [opciones] nombre_tabla …
```

```
isamchk [opciones] nombre_tabla …
```

Se puede especificar una tabla o múltiples tablas para verificar. También se pueden especificar los archivos de índice de las tablas. También se pueden utilizar patrones de nombre de archivo (*,?) para especificar múltiples tablas.

También se puede especificar la ruta en que se encuentran los ficheros de las tablas, en caso de que éstos no estuvieran en el directorio que se está utilizando.

Entre las opciones posibles suele utilizarse *–extend-check*, que provoca una verificación lenta, pero rigurosa. La sintaxis sería entonces:

Reparación de tablas: Una vez que se ha detectado que una tabla tiene errores, será necesaria su reparación. En caso de dificultades en su reparación habría que recuperarla de una copia de seguridad o de un fichero log de actualización.

El primer intento de reparación de una tabla suele hacerse:

```
myisamchk –recover –quick nombre_tabla
```

A través de la utilización de la opción --*quick* intenta realizar la reparación basándose sólo en los archivos de índice (sin tocar la información de las tablas).

Si el intento anterior falla se utiliza la opción:

```
myisamchk –safe-recover nombre_tabla
```

Existen otras opciones pero son menos utilizadas.

Bloqueo de tablas: El bloqueo de tablas evita que otros clientes escriban en la tabla mientras está siendo verificada o reparada. Será necesario cerrar los archivos de la tabla (a través de la sentencia FLUSH) para limpiar los cambios no escritos que todavía puedan estar situados en la memoria caché. El bloqueo de sólo lectura se realiza como sigue:

```
mysql> LOCK TABLE nombre_tabla READ;
mysql> FLUSH TABLES;
```

Una vez realizado el bloqueo, se realizan adecuadamente *myisamchk* e *isamchk* con las opciones correspondientes.

Una vez finalizadas las tareas de verificación o recuperación, será necesario volver a desbloquear la tabla mediante la sentencia UNLOCK con la siguiente sintaxis:

```
mysql> UNLOCK TABLE;
```

- CODIGOS DE ERROR: Cuando se producen errores en MySQL, el servidor ofrece códigos relativos a esos errores que inicialmente no son inteligibles para el usuario. Estos mensajes de error informativos pueden ser interpretados a partir de sus códigos mediante la utilidad *perror* cuya sintaxis es la siguiente:

```
perror [OPCIONES][CODIGOERROR1 [CODIGOERROR2…]]
```

La utilidad *perror* explica el error correspondiente a cada código situado como argumento. Sus opciones más importantes son:

`-?, --help`	Proporciona ayuda sobre la utilidad.
`-I, --info`	Sinónimo de –help.
`-s, --silent`	Sólo imprime el mensaje de error.
`-v, --verbose`	Imprime el código de error y el mensaje.
`-V, --version`	Muestra la versión de la utilidad.

BLOQUE III: LA AUDITORÍA

La sabiduría reside en distinguir cuándo se puede cambiar la realidad, y cuándo hay que aceptar lo que no se puede modificar.
Séneca

4. AUDITORIA

4.1 INTRODUCCIÓN

A finales del siglo XX, los Sistemas Informáticos se han constituido en las herramientas más poderosas para materializar uno de los conceptos más vitales y necesarios para cualquier organización empresarial, los Sistemas de Información de la empresa.

La Informática hoy, está incluida en la gestión integral de la empresa, y por eso las normas y estándares propiamente informáticos deben estar, por lo tanto, sometidos a los generales de la misma. En consecuencia, las organizaciones informáticas forman parte de lo que se ha denominado el *management* o gestión de la empresa. Cabe aclarar que la Informática no gestiona propiamente la empresa, ayuda a la toma de decisiones, pero no decide por sí misma. Por ende, debido a su importancia en el funcionamiento de una empresa, existe la Auditoría Informática.

El término de Auditoría se ha empleado incorrectamente con frecuencia ya que se ha considerado como una evaluación cuyo único fin es detectar errores y señalar fallos. A causa de esto, se ha tomado la frase "Tiene Auditoría" como sinónimo de que, en dicha entidad, antes de realizarse la auditoría, ya se habían detectado fallos. El concepto de auditoría es mucho más que esto. Es un examen crítico que se realiza con el fin de evaluar la eficacia y eficiencia de una sección, un organismo, una entidad, etc. La palabra auditoría proviene del latín "*auditorius*", y de ésta proviene la palabra auditor, que se refiere a todo aquel que tiene la virtud de oír.

Se podría indicar que la auditoría es un examen crítico pero no mecánico, que no implica la preexistencia de fallos en la entidad auditada y que persigue el fin de evaluar y mejorar la eficacia y eficiencia de una sección o de un organismo.

Los principales objetivos que constituyen la Auditoría Informática son el control de la función informática, el análisis de la eficiencia de los Sistemas Informáticos que comporta, la verificación del cumplimiento de la Normativa general de la empresa en este ámbito, y la revisión de la eficaz gestión de los recursos informáticos materiales y humanos.

La finalidad del Auditor Informático debe ser el correcto uso de los recursos que la entidad o empresa utiliza para disponer de un eficiente y eficaz Sistema de Información. Claro está, que para la realización de una auditoría informática eficaz, se debe entender a la empresa en su más amplio sentido: desde la más pequeña de las empresas familiares hasta una Universidad, un Ministerio o un Hospital utilizan la informática para gestionar sus "negocios" de forma rápida y eficiente, con el fin de obtener un determinado beneficio.

Por eso, al igual que los demás órganos de la empresa, los Sistemas Informáticos deberían estar sometidos al control correspondiente. La importancia de llevar un control de esta herramienta se puede deducir de varios aspectos. Algunos de ellos son:

- Los ordenadores y los Centros de Proceso de Datos se convirtieron en blancos apetecibles no solo para el espionaje, sino para la delincuencia y el terrorismo. En este caso interviene la Auditoría Informática de la Seguridad.

- Los ordenadores creados para procesar y difundir resultados o información elaborada pueden producir resultados o información errónea si dichos datos son, a su vez, erróneos. Este concepto tan obvio es en ocasiones olvidado por las mismas empresas que terminan perdiendo de vista la naturaleza y calidad de los datos de entrada a sus Sistemas Informáticos, con la posibilidad de que se provoque un efecto dominó y afecte a Aplicaciones independientes. En este caso interviene la Auditoría Informática de Datos. Un Sistema Informático mal diseñado puede convertirse en una herramienta extremadamente peligrosa para la empresa: como las

máquinas obedecen a las órdenes recibidas y la modelización de la empresa está determinada por los ordenadores que materializan los Sistemas de Información, la gestión y la organización de la empresa no puede depender de un Software y Hardware mal diseñados.

Estos son solo algunos de los varios inconvenientes que puede presentar un Sistema Informático, por eso, la necesidad de la Auditoría de Sistemas.

La auditoría nace como un órgano de control de algunas instituciones estatales y privadas. **La función auditora debe se r absolutamente independiente** ; no tiene carácter ejecutivo, ni son vinculantes sus conclusiones. Queda a cargo de la empresa tomar las decisiones pertinentes.

La auditoría contiene elementos de análisis, de verificación y de exposición de debilidades y disfunciones. Pueden aparecer sugerencias y planes de acción para eliminar las disfunciones y debilidades antedichas. Estas sugerencias que quedan recogidas en el informe final de la auditoría reciben el nombre de recomendaciones.

Las funciones de análisis y revisión que el auditor informático realiza, puede chocar con la psicología del auditado, puesto que éste tiene la necesidad de realizar sus tareas con racionalidad y eficiencia. La desconfianza o reticencia del auditado es comprensible y, en ocasiones, fundada. El nivel técnico del auditor es a veces insuficiente, dada la gran complejidad de los Sistemas, unidos a los plazos (en ocasiones) demasiado breves de los que suelen disponer para realizar esta tarea tan importante.

Además del estudio de los Sistemas, el auditor somete al auditado a una serie de cuestionarios o *checklist*, son guardados celosamente por las empresas auditoras, ya que son activos importantes de su actividad.

Los cuestionarios tienen que realizarse por el auditor de manera precisa y exacta, ya que si son mal aplicados se pueden llegar a obtener resultados distintos a los esperados por la empresa auditora.

El cuestionario puede llegar a explicar cómo ocurren los hechos, pero no la razón por la que ocurren. El cuestionario debe estar subordinado a la norma y al método, puesto que sólo una metodología precisa puede desentrañar las causas por las cuales se realizan actividades teóricamente inadecuadas o se omiten otras correctas.

El auditor sólo puede emitir un juicio global o parcial basado en hechos, careciendo de poder para modificar la situación analizada por él mismo.

Miguel Ángel Ramos justifica la auditoría en [PIATTINI Y DEL PESO, 2001] mostrando el siguiente símil: “la existencia de normativa sin auditoría podría equivaler a la no-existencia de la Guardia Civil de Tráfico, lo que incrementaría los accidentes e iría convirtiendo la circulación en caótica y peligrosa”. También indica que una auditoría debe basarse en políticas y normas de la entidad auditada.

Hasta este momento, se ha tratado de explicar cómo almacenar la información y cómo asegurarla, pero queda controlar que el sistema funcione de una manera eficaz y eficiente, y vigilar que las medidas de seguridad sean las oportunas y estén disponibles ante cualquier eventualidad.

No se puede abordar la auditoría de un sistema informático concreto (en nuestro caso MySQL) sin auditar también otros elementos que rodean al sistema de información.

La supervisión de las auditorías es un aspecto muy importante por cuánto significa la comprobación y seguridad de que éstas se hayan realizado cumpliendo con los principios y normas establecidas para el ejercicio de la auditoría y permiten evaluar la correspondencia del dictamen emitido por el grupo de auditores con la situación real de la entidad auditada.

4.2 DEFINICIÓN

En primer es importante definir el término de auditoría. Según el Diccionario de la Lengua Española, la auditoría es:

> "**Revisión sistemática de una actividad o de una situación para evaluar el cumplimien to de las r eglas o criterios objetivos a que aquellas deben someterse**"

Por otro lado, la Asociación Americana de Contabilidad (*American Accounting Associaton*) lo define como:

> "Proceso sistemático de obtención y evaluación objetiva de evidencias acerca de las aseveraciones efectuadas por terceros referentes a hechos y eventos de naturaleza económica, para testimoniar el grado de correspondencia entre tales afirmaciones y un conjunto de criterios convencionales, comunicando los resultados obtenidos a los destinatarios y usuarios interesados".

Obviamente, esta definición se refiere a la Auditoría Financiera o Auditoría de Cuentas, pero contiene suficientes elementos útiles para comenzar a precisar las ideas sobre Auditoría en general. Se pueden remarcar varias ideas de la anterior definición:

1. Se señala que "es un proceso sistemático", luego conlleva una aproximación estructurada, disciplinada, lógica y profesional a la formación de la opinión del auditor y a la toma de decisiones.

2. Se introducen las tres fases en las que se desglosa todo método auditor, cuando afirma "...obtención y evaluación objetiva de evidencias... para testimoniar...". Tres operaciones (obtención de evidencias, evaluación de éstas, y emisión de la opinión) que, como se verá, se efectúan en todo tipo de auditoría, sea del tipo que sea.

3. Por último, se señala un "conjunto de criterios convencionales", contra los que el auditor contrasta la realidad percibida. Estos criterios

convencionales, que en la Auditoría Financiera son los Principios Contables Generalmente Aceptados, en la Auditoría de Sistemas de Información o Auditoría Informática son determinados por un Marco de Control y unos Objetivos de la Implementación de Controles en entornos de Tecnologías de la Información y la Comunicación (TIC).

Hoy la auditoría se entiende como un proceso de reducción del riesgo. La misión de la auditoría es entonces la de reducir el riesgo de información hasta un grado aceptable para los destinatarios de los estados financieros (auditoría financiera) o de la información que se trate (otros tipos de auditoría incluyendo la auditoría informática).

Se podría resumir, que la Auditoría Informática es el conjunto de "**una serie de exámenes periódicos o esporádicos de un sistema informático, cuya finalidad es analizar y evaluar la planificación, el control, la eficacia, la seguridad, la economía y la adecuación de la infraestructura informática de la empresa**".

4.2.1 El método de auditoría

El método auditor se basa en la ejecución secuencial de tres fases y un epílogo:

1. **Reconocimiento del problema**, como puede ser la existencia real de un determinado activo (por ejemplo hardware o software), la ocurrencia real de unos eventos (por ejemplo el seguimiento de unas prácticas de calidad y seguridad), o el registro apropiado de unos hechos (por ejemplo unos cambios de software de sistemas debidamente documentados), la cuantificación de unos hechos (por ejemplo número de transacciones procesadas por unidad de tiempo, o número de visitas diarias a un sitio de Web).

2. **Recolección de la evidencia** , construyendo una base sobre la que practicar las pruebas necesarias, mediante la obtención de información documental, inspección física, información testimonial, y razonamiento analítico.

3. **Evaluación de la evidencia** , mediante pruebas específicamente diseñadas. Estas pruebas deben ser pruebas de cumplimiento, para verificar el cumplimiento efectivo de los mecanismos de control que se dice que existen, y pruebas sustantivas.

4. **Formulación de un juicio profesional** sobre cómo la información evaluada concuerda con la realidad, tal y como el auditor percibe la realidad en ese momento.

4.2.2 Características de la auditoría informática

La información de la empresa y para la empresa, siempre importante, se ha convertido en un activo real de la misma, como sus *stocks* o materias primas (si las hay). Por tanto, han de realizarse inversiones informáticas, materia de la que se ocupa la Auditoría de Inversión Informática.

Del mismo modo, los Sistemas Informáticos han de protegerse de modo global y particular: a ello se debe la existencia de la Auditoría de Seguridad Informática en general, o a la auditoría de Seguridad de alguna de sus áreas, como pudieran ser Desarrollo o Técnica de Sistemas.

Cuando se producen cambios estructurales en la Informática, se reorganiza de alguna forma su función y se pasa a estar en el campo de la Auditoría de Organización Informática.

Estos tres tipos de auditorías engloban a las actividades auditoras que se realizan en una auditoría parcial. De otra manera: cuando se realiza una auditoría del área de Desarrollo de Proyectos de la Informática de una empresa, es porque en ese Desarrollo existen, además de ineficiencias, debilidades de organización, de inversiones, de seguridad, o alguna mezcla de ellas.

4.3 AUDITORIA DE LA SEGURIDAD INFORMATICA

Un ordenador es un instrumento que recoge y almacena gran cantidad de información, la cual puede ser confidencial para individuos, empresas o instituciones, y puede ser mal utilizada o divulgada a personas que hagan mal uso de ésta. En ocasiones pueden ocurrir robos, fraudes o sabotajes que provoquen la destrucción total o parcial de la actividad computacional. Esta información puede ser de suma importancia, y el no tenerla en el momento preciso puede provocar retrasos sumamente costosos.

En la actualidad y principalmente en los ordenadores personales y con la proliferación de Internet, se ha dado otro factor que hay que considerar: el llamado "virus" informático. Al auditar los sistemas se debe tener cuidado de que no se tengan copias "piratas" o bien que, al conectarnos en red con otros ordenadores, no exista la posibilidad de transmisión del virus.

La seguridad en la informática abarca los conceptos de seguridad física y seguridad lógica, entre otros.

La **seguridad física** se refiere a la protección del hardware y de los soportes de datos, así como a la de los edificios e instalaciones que los albergan. Contempla las situaciones de incendios, sabotajes, robos, catástrofes naturales, etc.

La **seguridad lógica** se refiere a la seguridad de uso del software, a la protección de los datos, procesos y programas, así como la del ordenador y autorizado acceso de los usuarios a la información. Un método eficaz para proteger sistemas de computación es el software de control de acceso.

Dicho simplemente, los paquetes de control de acceso protegen contra el acceso no autorizado, pues piden al usuario una contraseña antes de permitirle el acceso a información confidencial. Dichos paquetes han sido populares desde hace muchos años en el mundo de los grandes ordenadores, y los principales proveedores ponen a disposición de clientes algunos de estos paquetes.

La seguridad informática se puede dividir como Área General y como Área Específica (seguridad de Explotación, seguridad de las Aplicaciones, etc.). Así, se podrán efectuar auditorías de la Seguridad Global de una Instalación Informática (Seguridad General) y auditorías de la Seguridad de un área informática determinada (Seguridad Específica).

Con el incremento de agresiones a instalaciones informáticas en los últimos años, se han ido originando acciones para mejorar la Seguridad Informática a nivel físico. Los accesos y conexiones indebidos a través de las Redes de Comunicaciones, han acelerado el desarrollo de productos de Seguridad lógica y la utilización de sofisticados medios criptográficos.

El sistema integral de seguridad debe comprender:

- Elementos administrativos.
- Definición de una política de seguridad.
- Organización y división de responsabilidades.
- Seguridad física y contra catástrofes (incendios, terremotos, etc.).
- Prácticas de seguridad del personal.
- Elementos técnicos y procedimientos.
- Sistemas de seguridad (de equipos y de sistemas, incluyendo todos los elementos, tanto redes como terminales)
- Aplicación de los sistemas de seguridad, incluyendo datos y archivos.
- El papel de los auditores, tanto internos como externos.
- Planeación de programas de desastre y su prueba.

La decisión de realizar una Auditoría Informática de Seguridad Global en una empresa, se fundamenta en el estudio exhaustivo de los riesgos potenciales a los que está sometida. Se elaboran “matrices de riesgo”, en las que se consideran los factores de las “Amenazas” a las que está sometida una instalación y los “Impactos” que aquellas puedan causar cuando se presentan. Las matrices de riesgo se representan en cuadros de doble entrada <<Amenaza-Impacto>>, en donde se evalúan las probabilidades de ocurrencia de los elementos de la matriz.

4.4. METODOLOGÍA DE TRABAJO (AUDITORÍA INFORMÁTICA)

El método de trabajo del auditor pasa por las siguientes etapas:

- Alcance y objetivos de la auditoría informática.
- Estudio inicial del entorno auditable.
- Determinación de los recursos necesarios para realizar la auditoría.
- Elaboración del plan y de los programas de trabajo.
- Actividades propiamente dichas de la auditoría.
- Confección y redacción del informe final.
- Redacción de la carta de introducción o carta de presentación del informe final.

4.4.1. Definición de alcance y objetivos

El alcance de la auditoría expresa los límites de la misma. Debe existir un acuerdo muy preciso entre auditores y clientes sobre las funciones, las materias y las organizaciones a auditar.

Con la finalidad de acotar el trabajo, resulta muy ventajoso para ambas partes concretar las excepciones de alcance de la auditoría, es decir qué materias, funciones u organizaciones no van a ser auditadas.

Tanto los alcances como las excepciones deben figurar al comienzo del informe final.

Las personas que realizan la auditoría han de conocer con la mayor exactitud posible los objetivos a los que su tarea debe llegar. Deben comprender los deseos y pretensiones del cliente, de forma que las metas fijadas puedan ser cumplidas.

Una vez definidos los objetivos específicos, éstos se añadirán a los objetivos generales y comunes a toda Auditoría Informática: la operatividad de los sistemas y los controles generales de gestión informática.

4.4.2. Estudio inicial

Para realizar dicho estudio han de examinarse las funciones y actividades generales de la informática.

Para su realización el auditor debe conocer lo siguiente:

1. Organización: para el equipo auditor, el conocimiento de quién ordena, quién diseña y quién ejecuta es fundamental. Para realizar esto en auditor deberá tener en cuenta:

- Organigrama: el organigrama expresa la estructura oficial de la organización a auditar. Si se descubriera que existe un organigrama que se aplica diferente al oficial, se deberá señalar tal circunstancia

- Departamentos: se entiende como departamentos a los órganos que siguen inmediatamente a la Dirección. El equipo auditor describirá brevemente las funciones de cada uno de ellos.

- Número de puestos de trabajo: se comprobará que los nombres de los puestos de trabajo de la organización corresponden a las distintas funciones reales. Es frecuente que bajo nombres diferentes se realicen funciones idénticas, lo cual indica la existencia de funciones operativas redundantes. Esta situación pone de manifiesto deficiencias estructurales. El auditor dará a conocer esta circunstancia y expresará el número de puestos de trabajo que son verdaderamente diferentes.

- Relaciones Jerárquicas y funcionales entre órganos de la Organización: el auditor estudiará si se cumplen las relaciones funcionales y jerárquicas previstas por el organigrama, o por el contrario detectará si hay algún fallo en este sentido.

- Flujos de Información: el flujo de información entre las diferentes áreas y departamentos de una organización es necesario para una gestión eficiente, siempre y cuando dicho flujo no distorsione el propio organigrama. En

ocasiones, las organizaciones crean de manera espontánea canales alternativos de información, sin los cuales las funciones no podrían ejercerse con eficacia. Estos canales alternativos se producen porque hay pequeños o grandes fallos en la estructura y en el organigrama que los representa. Otras veces, la aparición de flujos de información no previstos obedece a afinidades personales o simple comodidad (es más fácil la comunicación con personas con las que tienes más afinidad, que con las que no se tiene ningún tipo de relación). Estos flujos de información son indeseables y producen graves perturbaciones en la organización.

- Número de personas por puesto de trabajo: es un parámetro que los auditores informáticos deben considerar. La inadecuación del personal determina que el número de personas que realizan las mismas funciones rara vez coincida con la estructura oficial de la organización.

2. Entorno Operacional: El equipo de auditoría informática debe poseer una adecuada referencia del entorno en el que va a realizar la auditoría. Este conocimiento previo se logra determinando, fundamentalmente, los siguientes puntos:

- Situación geográfica de los sistemas: se determinará la ubicación geográfica de los distintos Centros de Proceso de Datos en la empresa. A continuación, se verificará la existencia de responsables en cada uno de ellos, así como el uso de los mismos estándares de trabajo.

- Arquitectura y configuración de hardware y software: cuando existen varios equipos, es fundamental la configuración elegida para cada uno de ellos, ya que los mismos deben constituir un sistema compatible e intercomunicado. La configuración de los sistemas está muy ligada a las políticas de seguridad lógica de las compañías. Los auditores, en su estudio inicial, deben tener en su poder la distribución e interconexión de los equipos.

- Inventario de hardware y software: el auditor recopilará información escrita, en la cual deben figurar todos los elementos físicos y lógicos de la

instalación. En cuanto a hardware figurarán las CPUs, unidades de control locales y remotas, periféricos de todo tipo, etc. El inventario de software debe contener todos los productos lógicos del sistema, desde el software básico hasta los programas de utilidad adquiridos o desarrollados internamente.

- Comunicación y Redes de Comunicación: en el estudio inicial los auditores dispondrán del número, situación y características principales de las líneas, así como de los accesos a la red pública de comunicaciones. Igualmente, poseerán información de las Redes Locales de la Empresa.

3. Aplicaciones y Bases de datos: El estudio inicial que han de realizar los auditores se cierra y culmina con una idea general de los procesos informáticos realizados en la empresa auditada. Para ello deberán conocer lo siguiente:

- Volumen, antigüedad y complejidad de las aplicaciones.

- Metodología del diseño: se clasificará globalmente la existencia total o parcial de metodología en el desarrollo de las aplicaciones. Si se han utilizados varias a lo largo del tiempo se pondrá de manifiesto.

- Documentación: la existencia de una adecuada documentación de las aplicaciones proporciona beneficios tangibles e inmediatos muy importantes. La documentación de programas disminuye notablemente el mantenimiento posterior de los mismos.

- Cantidad y complejidad de bases de datos y ficheros: el auditor recopilará información de tamaño y características de las bases de datos, clasificándolas en relación y jerarquías. Hallará un promedio de número de accesos a ellas por hora o días. Esta operación se repetirá con los ficheros, así como la frecuencia de actualizaciones de los mismos. Estos datos proporcionan una visión aceptable de las características de la carga informática.

Con respecto a la metodología para la Auditoría de Bases de Datos se establecen las siguientes*:

1. Metodología tradicional: El auditor revisa el entorno con la ayuda de una lista de control que consta de una serie de cuestiones a verificar, como por ejemplo ¿existe una metodología de diseño de la base de datos?, cuya respuestas pueden ser: sí (S), no (N), no aplicable (N/A).El auditor debe registrar el resultado de su investigación. Este tipo de técnica suele ser aplicada a la auditoría de un producto de bases de datos, especificándose en la lista de control todos los aspectos a tener en cuenta.

2. Metodología de evaluación de riesgos: Se conoce también como "*risk oriented approach"* y es la que propone ISACA, y empieza fijando los objetivos de control que minimizan los riesgos potenciales a los que está sometido el entorno.

4.4.3. Determinación de recursos de la Auditoría Informática

Mediante los resultados del estudio inicial realizado se procede a determinar los recursos humanos y materiales que han de emplearse en la auditoría.

1. Recursos materiales: es muy importante su determinación, por cuanto la mayoría de ellos son proporcionados por el cliente. Las herramientas software propias del equipo van a utilizarse igualmente en el sistema auditado, por lo que han de convenirse en lo posible las fechas y horas de uso entre el auditor y cliente. Los recursos materiales del auditor son de dos tipos:

---

* *Extraído de: "Auditoría Informática: Un enfoque práctico" [PIATTINI Y DEL PESO]*

a) Recursos materiales Software:

- Programas propios de la auditoría: son muy potentes y flexibles. Habitualmente se añaden a las ejecuciones de los procesos del cliente para verificarlos.

- Monitores: se utilizan en función del grado de desarrollo observado en la actividad de Técnica de Sistemas del auditado, y de la cantidad y calidad de los datos ya existentes.

b) Recursos materiales Hardware: los recursos hardware que el auditor necesita son proporcionados por el cliente. Los procesos de control deben efectuarse necesariamente en los ordenadores del auditado, para lo cual habrá de convenir tiempo de máquina, espacio de disco, impresoras ocupadas, etc.

2. Recursos Humanos : la cantidad de recursos depende del volumen auditable, así como las características y perfiles del personal seleccionado dependen de la materia auditable. También hay que resaltar que la auditoría en general suele ser ejercida por profesionales universitarios y por otras personas de probada experiencia multidisciplinaria.

4.5. ACTIVIDADES DE LA AUDITORÍA INFORMATICA

La auditoría Informática general se realiza por áreas generales o por áreas específicas. Si se examina por grandes temas, resulta evidente la mayor calidad y el empleo de más tiempo total y mayores recursos.

Cuando la auditoría se realiza por áreas específicas, se abarcan de una vez todas las peculiaridades que afectan a la misma, de forma que el resultado se obtiene más rápidamente y con menor calidad.

- Técnicas de Trabajo:
    - Análisis de la información recabada del auditado.
    - Análisis de la información propia.
    - Cruce de las informaciones anteriores.
    - Entrevistas.
    - Simulación.
    - Muestreos.

- Herramientas:
    - Cuestionario general inicial.
    - Cuestionario *Checklist*.
    - Estándares.
    - Monitores.
    - Simuladores (generadores de datos).
    - Paquetes de auditoría (generadores de programas).
    - Matrices de riesgo.

Con respecto a la auditoría y control interno en un entorno de bases de datos:

- Software de Auditoría
- Sistema de monitorización y ajuste (*tunning*)
- Sistema operativo
- Monitor de transacciones
- Protocolos y sistemas distribuidos
- Paquetes de seguridad
- Diccionario de datos
- Herramientas CASE (*Computer Aided System/Software Engineering*)
- Lenguajes de 4ª Generación Independientes
- Herramientas de “minerías de datos”
- Aplicaciones

4.5.1. Cuestionarios

El trabajo de campo del auditor consiste en lograr una información necesaria para la emisión de un juicio global objetivo, siempre amparado en hechos que puedan demostrarse. A estos hechos se les denomina evidencias.

Para esto, suele ser habitual comenzar solicitando la complementación de cuestionarios que se envían a las personas concretas que el auditor crea adecuadas, sin que sea obligatorio que dicha personas sean las responsables oficiales de las diversas áreas a auditar.

Estos cuestionarios no pueden ni deben ser repetidos para instalaciones distintas, sino diferentes y muy específicos para cada situación.

Sobre esta base, se estudia y analiza la documentación recibida, de modo que tal análisis determine a su vez la información que deberá elaborar el propio auditor. El cruzamiento de ambos tipos de información es una de las bases fundamentales de la auditoría.

4.5.2 Entrevistas

El auditor debe comenzar un primer acercamiento con el auditado, y puede hacerlo de tres formas:

1. Mediante la petición de documentación concreta sobre alguna materia de su responsabilidad.

2. Mediante “entrevistas” en las que no se sigue un plan predeterminado ni un método estricto de sometimiento a un cuestionario.

3. Por medio de entrevistas en las que el auditor sigue un método preestablecido de antemano y busca unas finalidades concretas.

La entrevista es una de las actividades personales más importantes del auditor, puesto que en ésta se recoge información relevante.

4.6. INFORME FINAL

La función de la auditoría se materializa en un informe final escrito. Por lo tanto, la elaboración final es el exponente de su calidad. Resulta evidente la necesidad de redactar borradores e informes parciales previos al informe final, que son elementos de contraste de opinión entre auditor y auditado y que pueden descubrir fallos de apreciación en el auditor.

- Estructura del informe final: el informe comienza con la fecha de comienzo de la auditoría y la fecha de redacción del mismo. Se incluyen los nombres del equipo auditor y los nombres de todas las personas entrevistadas, con indicación de la jefatura, responsabilidad y puesto de trabajo que ostente.

- Definición de objetivos y alcance de la auditoría.

- Enumeración de temas considerados: antes de tratarlos con profundidad, se enumerarán lo más exhaustivamente posible todos los temas objeto de la auditoría.

- Cuerpo expositivo: para cada tema, se seguirá el siguiente orden:

    - *Situación actual*: cuando se trate de una revisión periódica, en la que se analiza no solamente una situación sino además su evolución en el tiempo, se expondrá la situación prevista y la situación real.

    - *Tendencias*: se tratará de hallar parámetros que permitan establecer tendencias futuras.

    - *Puntos débiles y amenazas*.

- *Recomendaciones y planes de acción*: constituyen junto con la exposición de puntos débiles, el verdadero objetivo de la auditoría informática.

- *Redacción posterior* de la carta de introducción o presentación.

- Modelo conceptual de la exposición del informe final:

  - El informe debe incluir solamente hechos importantes.

  - La inclusión de hechos poco relevantes o accesorios desvía la atención del lector.

  - El Informe debe consolidar los hechos que se describen en el mismo.

El término de "hechos consolidados" adquiere un especial significado de verificación objetiva, y deben estar documentalmente probados y soportados. La consolidación de los hechos debe satisfacer, al menos los siguientes criterios:

- El hecho debe poder ser sometido a cambios.

- Las ventajas del cambio deben superar los inconvenientes derivados de mantener la situación.

- No deben existir alternativas viables que superen al cambio propuesto.

- La recomendación del auditor sobre el hecho debe mantener o mejorar las normas y estándares existentes en la instalación.

La aparición de un hecho en un informe de auditoría implica necesariamente la existencia de una debilidad que ha de ser corregida. Podemos distinguir en el flujo del hecho o debilidad:

- Hecho encontrado:

- Ha de ser relevante para el auditor y para el cliente.
- Ha de ser exacto, y además convincente.
- No deben existir hechos repetidos.

- Consecuencias del hecho: las consecuencias deben redactarse de modo que sean directamente deducibles del hecho.

- Repercusión del hecho: se redactará las influencias directas que el hecho pueda tener sobre otros aspectos informáticos u otros ámbitos de la empresa.

- Conclusión del hecho: no deben redactarse conclusiones más que en los casos en que la exposición haya sido muy extensa o compleja.

- Recomendación del auditor informático:

  - Deberá entenderse por sí sola, con su simple lectura.

  - Deberá estar suficientemente soportada en el propio texto.

  - Deberá ser concreta y exacta en el tiempo, para que pueda ser verificada su implementación

  - La recomendación se redactará de forma que vaya dirigida expresamente a la persona o personas que puedan implementarla.

BLOQUE III: AUDITORÍA DE MySQL

La simplicidad es la forma de la verdadera grandeza.
Francisco De Santis

5. AUDITORÍA DE LA SEGURIDAD EN MySQL

5. 1. INTRODUCCIÓN

La auditoría y seguridad en Bases de Datos, es uno de los conocimientos imprescindibles de cualquier persona que quiera dedicarse de manera plena a la profesión de Auditor Informático.

La importancia de que un Auditor Informático tenga conocimientos de Auditoría y Seguridad en Bases de Datos radica en que actualmente cualquier aplicación, independientemente del lenguaje en el que está programada, consulta, modifica e introduce nuevos datos en una Base de Datos. Una base de datos se ha convenido en el punto en el que confluyen todas las aplicaciones. Es por tanto esencial, que un auditor sea capaz de auditar las metodologías utilizadas para el Diseño de la Base de Datos, los distintos entornos que se utilizan (Explotación, Desarrollo,…), así como la explotación que se hace de la Base de Datos.

Un Auditor Informático tiene que conocer aquellos procedimientos que debe utilizar para conocer los accesos no autorizados, los accesos de personas a información para la cual no tienen privilegios, así como el borrado o modificación de información privilegiada. En la actualidad la creación e imposición de procedimientos de seguridad ayudan a proteger lo que se está convirtiendo rápidamente en uno de los bienes más importantes y preciados de las empresas: los datos.

Como ya se ha comentado a lo largo de todo el proyecto, la información es un recurso crítico en todas las organizaciones que tienen que almacenarla y organizarla de la manera más completa, segura, fiable, y accesible posible. Uno de los métodos más extendidos en los últimos años, es el almacenamiento en bases de datos, cuya seguridad es, por tanto, de vital importancia. Esta importancia, no viene tanto desde un punto de vista competitivo, sino más bien desde el punto de vista de la supervivencia de la propia

organización, que puede verse seriamente afectada en caso de alterarse, destruirse o desvelarse parte o totalmente la información contenida en sus bases de datos.

La continuidad de la organización exige que los datos confidenciales estén únicamente disponibles a las personas autorizadas, siendo imposible para el resto, acceder a éstos. Rupturas o grietas en esta “privacidad de información” son por tanto extremadamente perjudiciales.

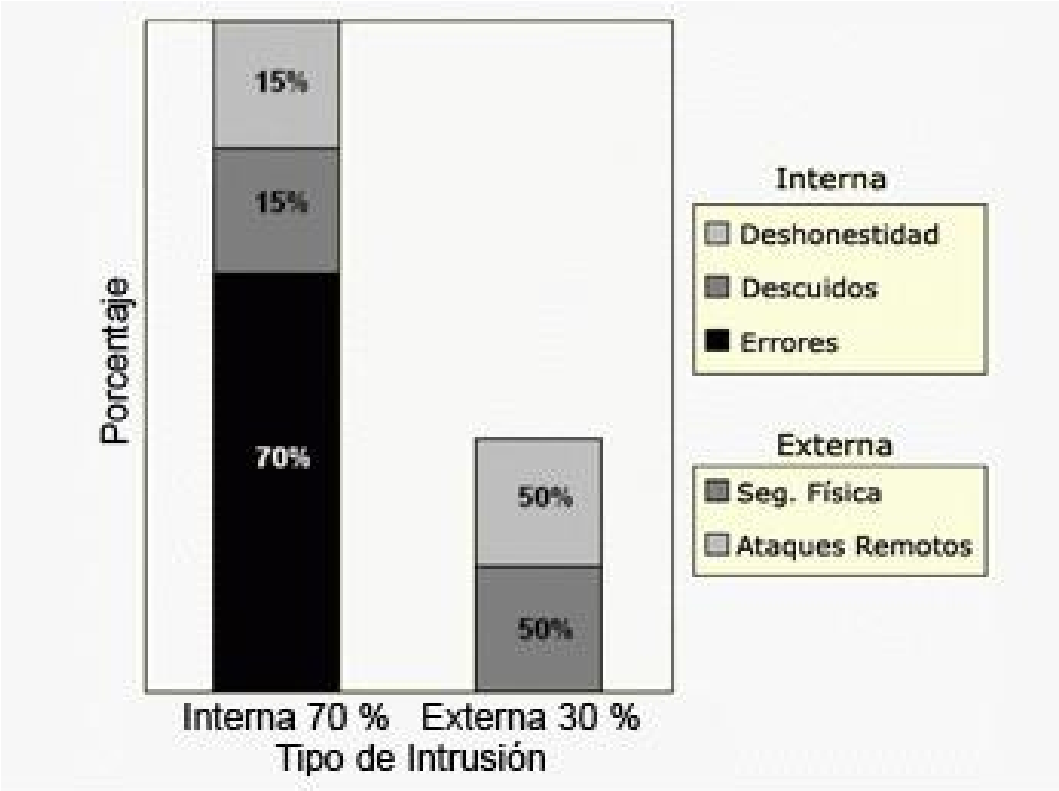
El trabajo del auditor será crucial, puesto que en sus manos estará la tarea de verificar y controlar que todo el sistema de bases de datos sea lo más seguro posible contra las posibles incidencias que puedan suponer un peligro para la organización.

A lo largo del proyecto, en apartados anteriores, se han comentado aquellos conceptos elementales necesarios que el auditor necesita conocer para elaborar de una forma correcta su trabajo y de esta forma proporcionar el control que el sistema de gestión de bases de datos MySQL necesita.

El auditor debería conocer profundamente el entorno y el sistema a auditar, puesto que de otra manera, podría dejar sin controlar algún área importante por un mal conocimiento del medio, como la gestión de privilegios, las contraseñas, etc.

Generalmente se tiene la idea preconcebida de que el personal de una organización es víctima de atacantes externos; sin embargo, de los robos, sabotajes o accidentes relacionados con los sistemas informáticos, el 70% (*) son causados por el propio personal de la organización propietaria de dichos sistemas ("*Inside Factor*").

Según este estudio casi las tres cuartas partes de los fallos o errores relacionados con los sistemas informáticos están causados por los propios trabajadores de la organización.



(*) Figura 17. *Según el estudio realizado por Cybsec: Tipos de intrusiones*

Realmente, este es un dato sumamente preocupante, puesto que una persona que está en contacto directo con la base de datos o con los sistemas conoce de primera mano el entorno y la base de datos, y por tanto, podrá conocer de una manera más exacta cuáles son los flancos por los que poder atacar el sistema, y cuáles son los puntos fuertes del mismo. Evidentemente, este hecho hace que la amenaza que supone esta persona sea significativamente mayor que cualquier otra, puesto que su ataque será seguramente más efectivo, directo y difícil de detectar, puesto que podría encubrirlo utilizando múltiples argucias.

El auditor deberá dar la importancia que merece este hecho y no descuidarlo, controlando aquellas áreas que tengan que ver con el personal de la organización, especialmente los controles de accesos, los privilegios, las contrataciones y despidos recientes, etc. Asimismo, tampoco tiene que olvidarse de los atacantes externos, que aunque – como se ha visto en el estudio realizado por la compañía de seguridad Cybsec – en menor grado, suponen un peligro real para la seguridad.

En base a ello, se pueden diferenciar dos grandes áreas: las amenazas internas (realizadas por el propio personal de la organización o *insiders)* y las amenazas externas (provenientes de personal externo a la organización denominados *outsiders).*

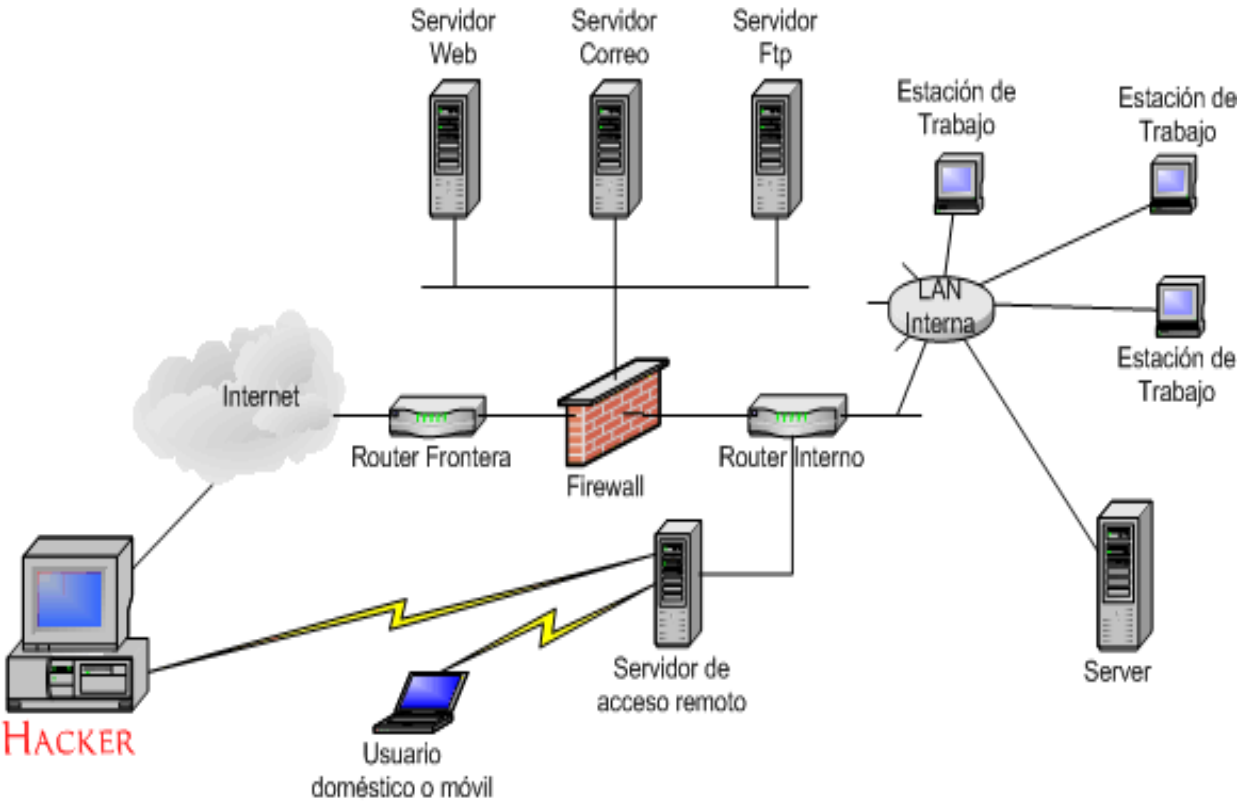


Figura 18. Representación gráfica de posibles ataques al sistema y sus barreras

5.2 AMENAZAS DE PERSONAS

5.2.1 Insiders o personal de la organización

Se podrían definir también como **amenazas internas** puesto que provienen del propio personal de la organización. Como se ha visto en la figura 6, de los tipos de intrusión se pueden distinguir tres grupos: Los errores producidos por empleados negligentes que no toman las precauciones que deberían, por descuidos de los mismos o por empleados deshonestos que de alguna manera han sido sobornados, o que quieren perjudicar a la organización por alguna razón.

El auditor deberá controlar que dichas personas no tengan la oportunidad de provocar ninguna situación de peligro que ponga a la organización para la que trabajan o han trabajado (no hay que olvidar nunca aquellos empleados que han sido despedidos o van a serlo en un breve período de tiempo; éstos pueden resultar muy peligrosos) en un verdadero aprieto.

Como ya se ha comentado a lo largo del presente proyecto en los apartados de normas ISO y COBIT el control del personal tiene que venir desde las primeras etapas en las que comienza la relación laboral entre la organización y el empleado. El currículo del aspirante a formar parte de la plantilla, debería haber sido verificado en todos los datos expuestos y en la medida de lo posible cotejar los datos proporcionados con las organizaciones para las que trabajó anteriormente. Aunque, cierto es, que este hecho no es indicativo de nada, puesto que no haya tenido ningún conflicto anteriormente, no asegura que no vaya a cometerlo en un futuro si se dan las circunstancias para ello, como su despido, supresión de privilegios, etc.

Hay una serie de principios, que el auditor debería controlar a fin de disminuir sensiblemente el daño que estas personas puedan infringir a la organización. Algunos de ellos se describen a continuación:

1. **Usuarios autenticados**. Estipula que las modificaciones sobre la base de datos tienen que se efectuadas por usuarios cuya identidad ha sido autenticada y que sea apropiada para la tarea que realiza.

2. **Sistema de menor privilegio**. Este fue uno de los primeros principios de seguridad que surgieron. En él se comenta la importancia de no dar más privilegios de los que el usuario requiere para desarrollar correctamente su función. Uno de los muchos términos que se designan a este principio es de *need-to-know*, *least temptation o need-to-do* (qué necesita saber, menor tentación o lo que necesita hacer).

3. **Separación de funciones** . Este principio es utilizado comúnmente para la prevención de fraudes y errores. Establece que ningún individuo aislado debería estar en condiciones de apropiarse

indebidamente de los activos por sí mismos. Esto significa que una cadena de eventos críticos deberían requerir la participación de varios individuos, de tal modo que una persona cuya misión es proteger la seguridad de la base de datos, no posea la capacidad de atentar contra ésta con total impunidad sin que nadie tuviese los conocimientos necesarios para percatarse de este hecho. También es conveniente la rotación temporal de funciones.

4. **Conocimiento parcia l.** Este principio es similar al anterior en el sentido que una persona con una misión crítica dentro del sistema de gestión de la base de datos no debería tener el dominio absoluto de dicha área, sino que la tarea debería estar compartida por lo menos con otra persona más, de manera que si por cualquier circunstancia una de estas personas comete un error la otra pueda subsanarlo. Asimismo, si uno de ellos fuese despedido o abandonase su puesto de trabajo, la otra persona podría manejar la situación haciéndose cargo temporalmente del sistema hasta que otro empleado reemplazase a la persona que se marchó.

5. **Eliminación de privilegios.** Cuando un usuario abandona el trabajo que desarrolla, debe eliminarse por completo el acceso al sistema y todos sus privilegios, de manera que éste no sea capaz de atentar contra el sistema ayudándose de sus conocimientos sobre el mismo y de las contraseñas de los recursos a los que podía acceder.

6. **Continuidad de operación.** Establece que las operaciones del sistema deben ser mantenidas por etapas en función de los posibles eventos que podrían provocar fallos en seguridad que están fuera de los márgenes de control de la organización, como son los desastres naturales.

Algunas políticas de seguridad para las amenazas internas que sería conveniente seguir son:

Directivas:

1. **Predecir Ataque/Riesgo:** Robo de información mediante el uso de ingeniería social.
2. **Amenaza:** Insider.
3. **Ataque:** Ingeniería social.
4. **Estrategia Proactiva:**
   a. Predecir posibles daños: pérdida de productividad y/o beneficios.
   b. Determinar y minimizar vulnerabilidades: concientización de los usuarios.
   c. Evaluar planes de contingencia.
5. **Estrategia Reactiva:**
   a. Evaluar daños: pérdida de beneficios e información.
   b. Determinar su origen: revelación de login y password por parte el usuario.
   c. Reparación de daños: implementar entrenamiento de los usuarios.
   d. Documentar y aprender.
   e. Implementar plan de contingencia.
6. **Examinar resultados y eficacia de la directiva:** Ajustar la directiva con los nuevos conceptos incorporados.

Figura 19*. Tipos de amenazas internas*

Directivas:

1. **Predecir Ataque/Riesgo:** Negación de servicios por abuso de recursos.
2. **Amenaza:** No existe. Empleado sin malas intenciones.
3. **Ataque:** No existe motivo ni herramienta, solo el desconocimiento por parte del usuario.
4. **Estrategia Proactiva:**
   a. Predecir posibles daños: pérdida de productividad por espacio de disco/memoria agotados.
   b. Determinar y minimizar vulnerabilidades: implementar cuotas de discos.
   c. Evaluar planes de contingencia: servidor backup.
   d. Capacitar el usuario.
5. **Estrategia Reactiva:**
   a. Evaluar daños: pérdida de producción.
   b. Determinar su origen y repararlos: hacer espacio en el disco.
   c. Documentar y aprender: implementar plan de contingencia.
6. **Examinar resultados y eficacia de la directiva:** Ajustar la directiva con los nuevos conceptos incorporados.

Figura 20. *Amenazas internas de personal por desconocimiento*

En base a lo que se ha venido comentando a lo largo de este apartado, las amenazas que provienen del interior de una organización son a las que menos atención se presta, y sin embargo, son amenazas de las más peligrosas. Veamos, pues, los diferentes tipos de amenazas internas a las que una organización está expuesta:

5.2.1.1 Personal de la organización: actual o antiguos

Son los empleados descontentos o con problemas en su trabajo o en su relación con la organización. Puede que una manera de evitar la situación de riesgo que esto supone, es que cuando la relación laboral de la organización con el empleado comienza a empeorar, se deberían tomar una serie de prevenciones antes que sea demasiado tarde. Habría que atajar el problema desde antes de que comience a producirse.

El auditor debería crear un grupo de riesgo. En dicho grupo se encontrarían aquellos usuarios con mayores problemas dentro de la organización, con comportamientos anormales en su conducta o aquellos que hubiesen expresado su descontento de manera explícita. Para realizar este control el auditor debería contar con la ayuda del jefe de personal, el supervisor de área, o con el departamento de recursos humanos que probablemente podrían ayudar al auditor poniendo a disposición de éste información valiosa referente a los empleados.

Este grupo de riesgo tendría que ser controlado a fondo, de forma que, el auditor tendría más probabilidades de descubrir de alguna manera puertas traseras o vacíos de seguridad abiertas por ellos deliberadamente y así poder atacar el sistema.

Esta amenaza podría ser realmente ofensiva si la persona que realiza el ataque es el administrador de la base de datos, un empleado que tuviese los privilegios de superusuario, o con privilegios excesivos. Antes de abandonar su puesto de trabajo el administrador o superusuario podría dejar una puerta abierta, o un acceso por el que podría fácilmente acceder al sistema y a la base de datos.

Una de las maneras que tienen los administradores de la base de datos de dejar un acceso abierto es creando una cuenta de superusuario. Esta cuenta tendría todos los privilegios que tiene el administrador de la base de datos, sin ninguna restricción. Esta persona podría acceder desde cualquier ordenador, únicamente teniendo instalado en el mismo el programa cliente *mysql* sin necesidad de que el ordenador fuese el servidor de la base de datos MySQL, únicamente teniendo acceso a Internet, a través de la red.

Hay que resaltar que en MySQL, cada cuenta está compuesta por un nombre de usuario, una contraseña y una ubicación (normalmente un nombre de máquina, una dirección IP o un comodín). Al tener una ubicación asociada al nombre de usuario se añade algo más de complejidad en relación con otros sistemas más sencillos.

En MySQL se utilizan una serie de tablas de permisos para mantener un registro de los usuarios y los diferentes privilegios que pueden tener. Esas tablas son tablas MySAM que se encuentran dentro de la base de datos. Tiene mucho sentido almacenar la información de la seguridad dentro de MySQL, ya que permite utilizar consultas SQL estándar para realizar los cambios en la seguridad. No existen archivos de configuración adicionales que la base de datos deba procesar. Sin embargo, como dato negativo, es que si un servidor no es configurado adecuadamente, cualquier usuario con nociones en lenguaje SQL puede hacer modificaciones a la seguridad.

Si el usuario con intención de atacar fuera el administrador de la base de datos, podría haber ejecutado la siguiente sentencia antes de marcharse:

```
GRANT ALL PRIVILEGES ON *.* TO  jessica@% IDENTIFIED BY
“mi_contraseña” WITH GRANT OPTION
```

La sentencia crearía una entrada para jessica@% (el símbolo “%” es un carácter comodín que especifica que el acceso puede ser desde cualquier dirección de *host* sin necesidad de ser el *host* local) en la tabla USER de la base de datos *mysql*, y que activaría todos los privilegios allí, porque es donde están almacenados los privilegios globales del superusuario.

¿De qué manera podría el auditor comprobar que este hecho no se ha producido?

Si el auditor ejecuta la sentencia:

```
SELECT * FROM information_schema.user_privileges WHERE
IS_GRANTABLE=’YES’
```

El resultado de esta consulta mostraría para un usuario determinado cuáles son las áreas o campos para los que tiene privilegios de administración. Si por ejemplo el resultado de la ejecución de la sentencia anterior fuese:

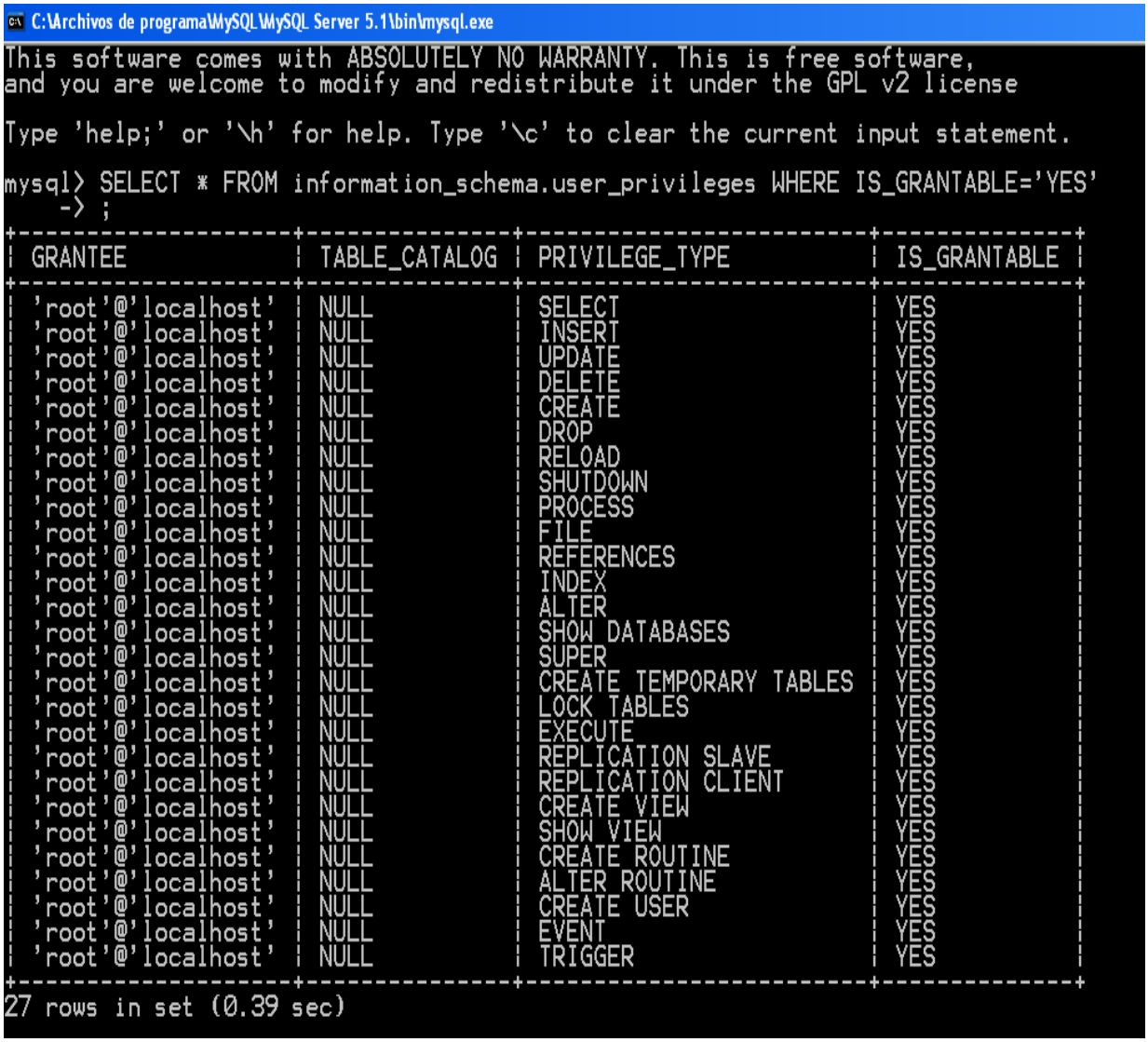


Figura 21. *El usuario root (administrador por defecto) tiene todos los permisos*

El auditor sabría que no hay ningún usuario más aparte del administrador que tenga ningún privilegio sobre selección, inserción, actualización, borrado… Este es un caso extremo, puesto que hay que pensar que en casi todas las bases de datos, habrá varias personas que tengan accesos sobre consultas, modificaciones, etc. La función del auditor será verificar y controlar que no haya ningún usuario creado que tenga privilegios sobre áreas que no debiera tener (como CREATE USER).

El administrador de la base de datos MySQL tiene acceso a bases de datos que para otros usuarios con menores privilegios no son visibles. La base de datos *mysql* sólo es visible para éste, y contiene información sensible que podría ser atacada por el administrador.

Para auditar de una manera lo más exhaustiva y completa posible se deberían contemplar las siguientes posibilidades:

- Comprobación de las últimas fechas de actualización o modificación de los archivos que pueden ser ejecutados por el sistema operativo.

- Comprobar que el usuario no tiene acceso al sistema eliminando su cuenta.

Estas dos primeras medidas se efectúan sobre el sistema operativo, y debiera servir como el primer punto que el auditor debe comprobar, aunque el sistema *mysql* permite la conexión desde fuera de la red local, se minimizarían los riesgos para acceder a la base de datos desde la propia organización cuando ya no debiera hacerlo.

- Revocar todos los privilegios que tuviera el administrador (con la sentencia `REVOKE`) y eliminar el usuario cuando se supiera a ciencia cierta la intención de éste de marcharse o al ser despedido y así evitar posibles complicaciones.

- Auditar a intervalos regulares el sistema y especialmente el área que manejaba el empleado, para asegurarse que no ha realizado ninguna acción prohibida.

- Revisar los últimos movimientos realizados en el sistema. Utilizando el comando `SHOW STATUS` se podría observar cuáles han sido los últimos movimientos realizados, cuántas conexiones ha habido, intentos fallidos de acceso, usuarios conectados, etc.

- Para conocer el número de conexiones que han sido abortadas porque el cliente no cerró la conexión apropiadamente el auditor deberá teclear en la consola de MySQL `aborted_clients.`

- De la misma forma, que para conocer el número de intentos de conexión al servidor MySQL que han fallado se puede utilizar: `aborted_connects`.

- A través de la sentencia `skip-show-database` se evita que se utilice el comando `SHOW DATABASES` sin tener el privilegio `SHOW DATABASES`. De manera, que un usuario no pueda ver las bases de datos de otros usuarios

- Revisar la tabla de usuarios para comprobar que no existe ningún nuevo usuario creado con privilegios excesivos. De la tabla `INFORMATION_SCHEMA` se podría realizar la consulta de qué usuarios tienen el privilegio de crear nuevos usuarios.

```
SELECT create_user FROM user_privileges WHERE
is_grantable='yes';
```

- No se debería poder ejecutar el servidor MySQL con el usuario root en sistema UNIX, puesto que cualquier usuario con el privilegio FILE podría crear ficheros como root (-root/.bashrc). Aunque *mysqld* rechaza ejecutarse como root el usuario podría haber ejecutado explícitamente la sentencia –user=root. El auditor deberá tener en cuenta este hecho a la hora de realizar la auditoría.

- Si un usuario tuviera privilegios `PROCESS` o `SUPER`, la salida de `mysqladmin processlist` podría mostrar el texto de cualquier sentencia que se estuviese ejecutando, así que cualquier usuario al que se le permitiese ejecutar ese comando podría ser capaz de ver si otro usuario ejecuta la sentencia:

```
UPDATE user SET password=PASSWORD('no_segura');
```

  Con lo que podría haber un serio problema de seguridad relacionado con las contraseñas. El auditor deberá comprobar que no hay ningún privilegio de ese tipo en la tabla de privilegios de la manera que se ha comentado anteriormente.

- De forma predeterminada MySQL tiene dos usuarios definidos y una base de datos `TEST`. Los usuarios por defecto no tienen predefinida ninguna contraseña y

las tablas de la base de datos que comienzan por `TEST` tienen permisos de escritura para todo el mundo. El administrador debería comprobar que esta función ha sido deshabilitada, que tiene asignada una contraseña o que han sido eliminados para evitar contratiempos.

Para comprobar si se han eliminado estos usuarios predefinidos sin contraseña podría ejecutar la siguiente sentencia:

```
SELECT host, user, from mysql.user WHERE
host='localhost' AND user=''
```

- El usuario *root* (administrador) o un superusuario puede haber otorgado privilegios a un usuario (que no sea *root*) con acceso a la tabla *mysql.user* a la que como ya se comentó no deberían tener acceso otros usuarios, puesto que ésta contiene no sólo el *host* desde el que tiene permiso para conectarse un determinado usuario, sino también sus contraseñas de acceso. Las cuentas de usuario de MySQL se listan en la tabla *user* de la base de datos *mysql*. Cada cuenta MySQL tiene una contraseña asignada, aunque lo que se guarda en la columna *Password* de la tabla *user* no es una versión en texto plano de la contraseña, sino un valor *hash* computado a partir de la misma. Los valores *hash* de las contraseñas se obtienen a partir de la función `PASSWORD`() y tienen 41 bytes. Por ello, el usuario con acceso a esa información privilegiada podría utilizar algún algoritmo para interceptar la contraseña o comprobar que hubiese algún usuario sin contraseña asignada, con lo que esa cuenta sería fácilmente atacada.

- El auditor puede comprobar los procesos que se están ejecutando en un instante determinado para confirmar cuántos usuarios hay conectados y en qué base de datos se encuentra ejecutando la sentencia `SHOW PROCESSLIST`, que devuelve una información que puede llegar a ser muy útil al auditor.

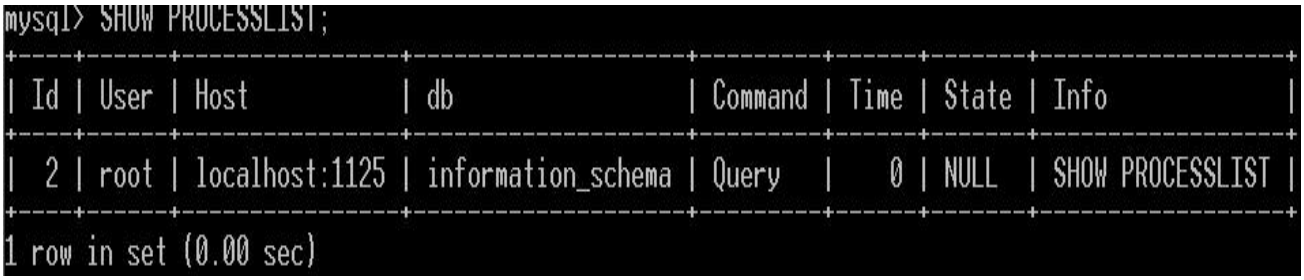


***Figura 22**. Resultado de la ejecución de la sentencia SHOW PROCESSLIST*

- Se podrían revisar los privilegios de tabla y de bases de datos ejecutando la sentencia en la base de datos *mysql*:

```
SELECT * FROM db;
```

  De esta forma el auditor podrá comprobar los privilegios que tiene un determinado usuario sobre tablas, columnas, globales, etc.

- Se debería comprobar que los usuarios que no deban tener acceso desde fuera de la organización no tengan el privilegio para hacerlo, es decir, que sólo puedan conectarse desde los equipos de la organización, evitando de esta manera, accesos no deseados desde otras localizaciones. Este control se realiza comprobando en la tabla *user* de *mysql* desde qué dirección de *host* un usuario puede conectarse.

```
SELECT user,host FROM mysql.user;
```

- El auditor debe tener en cuenta también, si el superusuario o usuario con privilegios ha podido realizar volcados de información de las bases de datos para comprometer a la organización, o ha manipulado los archivos logs que contienen información muy poderosa (puesto que registran todos los movimientos que se han realizado sobre la base de datos) o aún tiene privilegios sobre el servidor para poder cerrarlo desde fuera de la organización.

Aunque MySQL es un sistema de gestor de bases de datos que ha mejorado significativamente la seguridad, es bastante complicado asegurar completamente que está libre de amenazas y errores, puesto que aunque su sistema basado en otorgamiento

de privilegios es bastante útil, un error o fallo en la administración de los mismos podría ser fatal.

Los privilegios en My SQL : El control de acceso está compuesto por privilegios que controlan la forma de utilización y manipulación de los diferentes objetos: bases de datos, tablas, columnas e índices. Para cualquier combinación de objetos, los privilegios tienen un valor verdadero o falso que indica si se puede o no realizar una acción. Dichos privilegios por objeto aparecen después de las consultas SQL que se utilizan para lanzar las comprobaciones.

La lista completa de los privilegios por objeto son las siguientes (entre paréntesis aparecen sobre cada tipo de objeto aplicable para ese privilegio):

- **Select** (Bases de datos, tablas, columnas)
- **Insert** (Bases de datos, tablas, columnas)
- **Update** (Bases de datos, tablas, columnas)
- **Index** (Bases de datos, tablas)
- **Alter** (Bases de datos, tablas)
- **Create** (Bases de datos, tablas)
- **Grant** (Bases de datos, tablas)
- **Referentes** (Bases de datos, tablas, columnas)

No todos los privilegios se aplican a cada tipo de objeto en MySQL.

A parte los privilegios por objeto, existe otro grupo de privilegios (los privilegios globales) que tienen que ver con el funcionamiento en sí de la bases de datos y que se aplican a todo servidor. Estos privilegios son: **Reload, Shutdown, Process, File y Super**.

El auditor debería tener establecido cuáles son márgenes establecidos para definir la seguridad en el sistema, es decir, ¿cuándo el SGBD MySQL ha dejado de ser seguro?, ¿bajo qué parámetros?

Con respecto a los empleados negligentes hay que tener un control preventivo, especialmente en la gestión de privilegios y contraseñas. Al dar de alta un nuevo usuario, éste deberá configurar su nueva cuenta, de manera que la nueva cuenta tenga una contraseña de acceso para evitar que otra persona pueda acceder a ella y realizar acciones con ésta.

Si la organización tuviera una política de seguridad definida, todos los empleados deberían conocerla y cumplirla para evitar contingencias. Si se observa que un empleado no cumple con las normas de seguridad se debería informar y tomar las medidas preventivas que fueran necesarias. Una de estas medidas es la asignación de los mínimos privilegios posibles con los que puedan desarrollar de manera correcta su función.

Por último, está el personal deshonesto que ha sido sobornado, normalmente, por empresas del sector, que quieren perjudicar de alguna manera a la empresa para obtener una mejor posición con respecto a la otra. Como se ha visto en la figura 7 éste es el grupo de atacantes menos numeroso. Sin embargo, debido a que normalmente esto sucede en grandes empresas, sus posibilidades de hacer daño de alguna manera se incrementa exponencialmente. Su interés fundamental será atacar los puntos débiles de la organización, y la mejor manera de hacerlo, es desde su "centro neurálgico": a través de sus propios empleados, que conocen de manera profunda el sistema con el que diariamente trabajan. Este ataque puede venir de muchas maneras, robando código fuente, información que tengan almacenadas sus bases de datos (como clientes), introduciendo algún tipo de virus que hiciese que la información fuese destruida, etc.

El auditar para minimizar los efectos que un ataque de estas características pudiera tener en la empresa, deberá controlar todos los aspectos relacionados con el personal (sueldos, comportamientos extraños, aumento o disminución del rendimiento laboral, etc.). Asimismo, deberá tener controlada la información (entradas y salidas que se produzcan).

5.2.2 Outsiders o personas externas a la organización

Aunque está empíricamente demostrado que este tipo de ataques externos son menos numerosos, las empresas y organizaciones cuyos sistemas utilizan la red como

medio de comunicación o como herramienta en su trabajo utilizan muchos de sus esfuerzos para evitar la intrusión externa, dejando en un segundo plano –erróneamente- la seguridad interna de su personal.

Si bien es cierto que la seguridad en la red avanza rápidamente intentando descubrir nuevas técnicas con las que proteger sistemas y servidores, los usuarios de la misma avanzan paralelamente en sus conocimientos e investigaciones para intentar evitar y destruir esa ansiada seguridad que necesitan todas las organizaciones. De hecho, curiosamente, hay muchas compañías que evitan este tipo de intrusión ofreciendo puestos de gestión de seguridad a los mismos *hackers* que en un pasado pudieron atravesar su barrera de seguridad. El auditor deberá controlar de manera especial estos casos, puesto que si alguna vez actuaron en contra de la empresa, ¿quién asegura que esto no pueda volver a suceder?

En los últimos tiempos, en los que la tecnología de la información y de la informática ha avanzado de manera tan rápida, se ha utilizado el beneficio que se obtiene con la utilización de Internet para poder tener una mejor comunicación con otras empresas u organizaciones, posibles clientes, etc. para ser más competitivos en el mercado. El SGBD MySQL también se ha hecho eco de esta nueva tecnología y ha ofrecido a sus clientes la posibilidad de utilizar el servidor de forma externa, de manera que un empleado o usuario desde cualquier punto geográfico, únicamente con acceso a Internet y el cliente *mysql* pudiera acceder al servidor propio de MySQL sin necesidad de tener instalado el mismo en su propio equipo. Esto evidentemente, trae ventajas, pero también inconvenientes, especialmente relacionados con la seguridad y protección de la información. El auditor deberá ser consciente en todo momento de esta posibilidad y auditar también esta área para que el sistema sea lo más seguro posible frente a posibles ataques externos.

Algunas políticas de seguridad para evitar esto son:

Directivas:

7. **Predecir Ataque/Riesgo:** Ingreso al sistema por vulnerabilidades en los sistemas o política de claves ineficiente.
8. **Amenaza:** Outsider recopilando información significativa.
9. **Ataque:** Ingreso al sistema.
10. **Estrategia Proactiva:**
    a. Predecir posibles daños: Robo y venta de información. Daño a la imágen de la empresa.
    b. Determinar y minimizar vulnerabilidades: actualización de sistemas vulnerables. Concientización a los usuarios en el manejo de contraseñas fuertes.
    c. Evaluar planes de contingencia: implementación de servidor backup para casos de daño de la información. Recuperación de imagen. Evaluar formas de minimizar el daño por la información robada.
11. **Estrategia Reactiva:**
    f. Evaluar daños: información susceptible robada.
    g. Determinar su origen: ingreso al sistema.
    h. Reparación de daños.
    i. Documentar y aprender.
    j. Implementar plan de contingencia: servidor backup.
12. **Examinar resultados y eficacia de la directiva:** Ajustar la directiva con los nuevos conceptos incorporados.

***Figura 23**. Políticas de seguridad de Outsiders*

Este tipo de amenazas y ataques normalmente suelen provenir de personas con algún fin económico, intelectual (demostrar/se que son capaces de saltar las barreras de seguridad), retos, etc. Según la información contenida en algunas páginas de Internet[9] existen varios personajes dispuestos a asaltar las barreras de seguridad de las organizaciones. Algunos de ellos son:

- ***Hacker***: Es el más conocido, y aunque se confunde con personas malintencionadas, esto no es así. Según esta página un *hacker* es alguien que disfruta explorando los sistemas y programas y sabe cómo sacarles el máximo provecho, al contrario que la mayoría de los usuarios que prefieren conocer sólo lo imprescindible. Normalmente no intentan dañar a la empresa, únicamente demostrar sus habilidades, sin embargo, esto también puede ser perjudicial para la imagen frente a terceros de la organización.

9.Información extraída de *http://www.baquia.com/noticias.php?id=9109*

- ***Sneaker.*** Simular en ciertos aspectos: es aquel individuo contratado para romper los sistemas de seguridad por las empresas e instituciones con la intención de subsanar dichos errores.

  Estos últimos no suponen un problema para el auditor puesto que son contratados por la propia empresa, aunque tendrá que tener en cuenta cuáles han sido los puntos flojos hallados al finalizar el trabajo de este individuo para futuras auditorías de seguridad del sistema.

- ***Craker***: es un término acuñado por los hackers hacia 1985 para defenderse contra la mala utilización que hacían los periodistas de la palabra hacker y que se refiere al que rompe la seguridad de un sistema. Su intención al contrario que la del hacker sí es perjudicar a la empresa, normalmente enviados por un tercero y su fin último es puramente económico.

- ***Warez d00dz.*** Se dedican a obtener, desproteger y/o distribuir copias ilegales de software propietario (*warez*).

- ***Phreakers.*** Aquellos que 'rompen' y hacen un uso ilegal de las redes telefónicas.

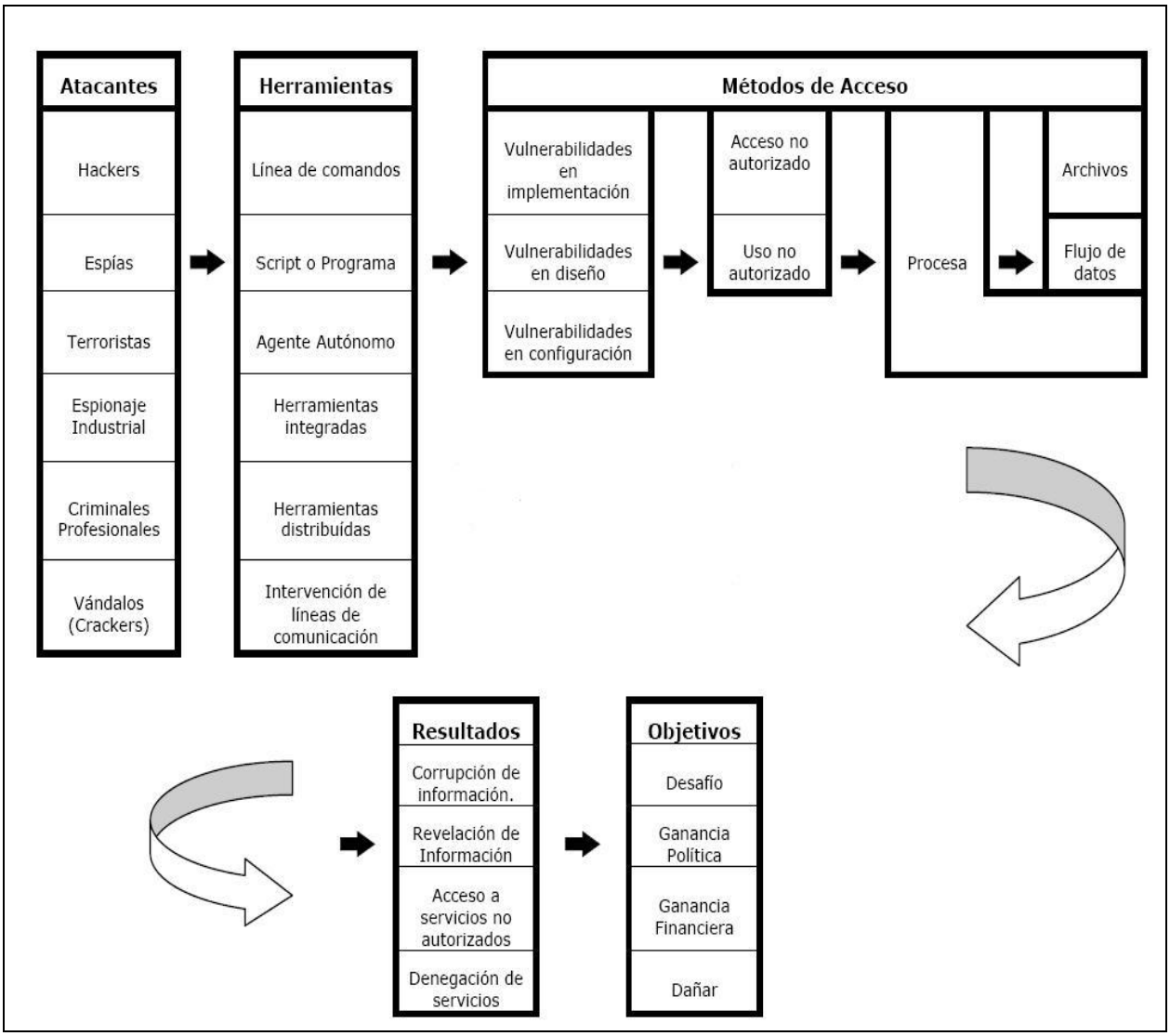


Figura 24. : *HOWARD, John D. Thesis: An Analysis of security on the Internet*

Para evitar en la medida de lo posible los ataques de todos estos personajes, el sistema deberá tener activadas por lo menos estas medidas de seguridad:

1. Un cortafuegos que aísle la red del exterior, también denominado *firewall*, que deberá estar actualizado y configurado de manera que aísle la base de datos mysql con el exterior, de esta manera se evita que intrusos puedan acceder a ella. La manera de verificar que esto es así, vendrá determinada por el firewall que la organización tenga contratado, así como el sistema operativo. (A partir de la versión Windows XP en adelante (Windows 7, Windows Vista, etc) existe un *firewall* por defecto fácilmente accesible y comprobable desde panel de control).

Estas dos medidas que se detallan a continuación deberían haberse auditado con anterioridad, tal y como se explica, en el apartado anterior.

2. Las distribuciones para Windows contienen tablas de permisos inicializadas que se instalan automáticamente, sin embargo, en un Unix, las tablas de permisos se llenan mediante el programa *mysql_install_db*. En un principio cualquier persona puede conectarse al servidor MySQL como root sin una contraseña y recibirá todos los privilegios, por ello, es sumamente importante establecer una contraseña para esta cuenta. Los usuarios anónimos tampoco tienen contraseña prefijada, así que habrá que eliminar las cuentas anónimas, o requerir que se establezca una contraseña para ellas:

```
UPDATE mysql.user SET password= PASSWORD
('nueva_contraseña) WHERE USER='';
```

Se debe comprobar también que no se almacene ninguna contraseña sin cifrar en la base de datos, puesto que si alguien tuviera acceso al ordenador, el intruso podría obtener una lista completa de claves y utilizarlas. Habría que utilizar alguna función de *hashing* de un sentido como MD5() o SHA1().

3. Con el comando `SHOW STATUS` observar las conexiones fallidas o erróneas, los intentos de acceso a la base de datos, etc. De esta manera el auditor puede hacerse una idea aproximada del nivel de peligrosidad en seguridad del sistema auditado.

Habrá que controlar especialmente los archivos *logs,* que contienen información privilegiada sobre lo que acontece en la base de datos. En MySQL Se guardan registros de las actualizaciones, borrados, modificaciones, etc. Una mala gestión en el acceso a esta carpeta podría suponer que un usuario malintencionado pudiese utilizar estos archivos o eliminarlos. Estos archivos contienen información valiosa para la tarea del auditor, por esta razón, éste deberá comprobar los permisos de carpeta en la que se encuentran dichos archivos, permitiendo sólo a los usuarios autorizados acceder a ella.

Los archivos de estado de MySQL son:

`HOSTNAME.pid`→ que contiene la ID de proceso del servidor

`HOSTNAME.err`→ que contiene los eventos de inicio y apagado así como las condiciones de error.

`HOSTNAME.log`→ eventos de conexión-desconexión y consulta de información.

`HOSTNAME.nnn`→ Texto de todas las consultas que modifican el contenido y la estructura de la tabla.

El auditor deberá conocer que tanto en Windows como en Unix, estos archivos se encuentran por defecto en la carpeta DATA. Deberá localizarlos y observar si ha habido algún movimiento extraño que quedase almacenado en el log.

Una de las ventajas que ofrece el sistema MySQL es que es de código abierto, esto quiere decir que cualquier persona podría acceder a los archivos que almacenan la configuración y los scripts del sistema y modificar a su antojo las opciones de configuración o introducir nuevos códigos para ajustar a su medida el sistema. Esto también puede ser peligroso, por ello el auditor deberá tenerlo en cuenta y auditar los archivos *my.ini* (configuración genérica del sistema) o en la carpeta de scripts comprobando que no se ha insertado ningún archivo SQL extraño.

Otro dato importante a conocer por el auditor es que si está auditando la versión de MySQL 5.0, se pueden limitar los siguientes recursos de servidor para cuentas individuales:

- El número de consultas que una cuenta puede realizar por hora
- El número de actualizaciones que una cuenta puede hacer por hora
- El número de veces que una cuenta puede conectar con el servidor por hora

Cualquier comando que un cliente puede realizar cuenta en el límite de consultas. Sólo los comandos que modifiquen la base de datos o las tablas cuentan en el límite de actualizaciones.

Desde MySQL 5.0.3, es posible limitar el número de conexiones simultáneas al servidor por cuenta. (Una cuenta en este contexto es un registro en la tabla user). Cada cuenta se identifica unívocamente por los valores de las columnas User y Host.

Como prerrequisito para usar esta característica, la tabla user en la base de datos MySQL debe contener las columnas relacionadas con el recurso. Los límites de recursos se guardan en las columnas *max_questions*, *max_updates*, *max_connections*, y *max_user_connections* .

MySQL ofrece cierta ayuda a la hora de impedir ataques a través de la red. Si se "advierte" que hay demasiadas conexiones en las que su resultado no es una sesión válida de MySQL, que provienen de una máquina concreta, entonces, se empieza a bloquear las conexiones que provengan de esa máquina determinada. La variable de servidor `max_connection_errors` determina cuántas conexiones "malas" debe permitir la base de datos antes de comenzar el bloqueo.

Cuando una máquina es bloqueada, MySQL guarda en el registro de roles un mensaje parecido a este:

```
Host 'host.malo.com' blocked because of many connection
errors.

Unblock with 'mysqladmin flush-hosts'
```

Como indica el mensaje, se puede utilizar el comando anterior para desbloquear dicha máquina, después de comprobar quién era la persona o entidad que tenía el problema y habiéndose solucionado ese. Ese comando, vacía las tablas de la memoria caché de las máquinas, resultando que todas ellas quedan desbloquedas, no habiendo ninguna manera de desbloquear una sola en concreto.

Si el auditor conoce los límites establecidos para un usuario podría verificar que se cumple con el comando `SELECT`, aplicando a la tabla o base de datos que esté auditando en ese momento. Si considera necesario cambiar el límite de recursos con un comando `GRANT` lo podría hacer utilizando la cláusula `WITH` que nombra cada recurso a ser limitado y un contador por hora indicando el valor límite. Por ejemplo, para crear una nueva cuenta que pueda acceder a la base de datos nueva_bd, pero sólo de forma limitada, se podría utilizar este comando (aunque debemos recordar que no es tarea del auditor modificar nada de la base de datos, se explica a modo de ejemplo):

```
mysql> GRANT ALL ON nueva_bd.* TO 'jessica'@'localhost'
    ->     IDENTIFIED BY 'contraseña'
```

```
->     WITH MAX_QUERIES_PER_HOUR 20
->          MAX_UPDATES_PER_HOUR 10
->          MAX_CONNECTIONS_PER_HOUR 5
->          MAX_USER_CONNECTIONS 2;
```

A través del comando `handler_read_rnd_next` y su resultado se comprueba el número de peticiones para leer el siguiente registro en el archivo de datos. El resultado de éste es alto si se están realizando muchos escaneos de tablas. Generalmente, esto sugiere que las tablas no están indexadas correctamente o que las consultas no están escritas para obtener ventaja de los índices que tienen, y el auditor tiene que tomar nota de este hecho.

5.3. PARAMETRIZACION Y RENDIMIENTO DEL SERVIDOR

A nadie le gusta perder datos. Si los discos dejan de funcionar, a menudo con pequeños avisos de alerta, es importante considerar configurar un RAID (*Redundant Array of Independent Disks* – como se le conoce en la actualidad, y antiguamente, *Redundant Array of Inexpensive Disks*-) en sus servidores de bases de datos para prevenir posibles fallos de disco que hagan perder tiempo y datos. Pero hay muchos tipos diferentes de RAID a considerar:

- Los niveles RAID estándar: 0, 1, 2, 3, 4, 5, etc.
- Los niveles RAID anidados: 0+1, 1+0, 30, etc.
- Los niveles RAID propietarios: Paridad doble, 1.5, 7, etc.

A modo de ejemplo se explicarán los más utilizados comúnmente:

RAID 0: De todos los tipos de RAID, RAID 0 es el que ofrece las mejores opciones de rendimiento. Lecturas y escrituras son las más rápidas que ninguna otra configuración. Este tipo de RAID debería utilizarse si no le preocupa la pérdida de datos (por ejemplo, si se está creando un clúster de esclavos MySQL, se obtienen todos los beneficios de rendimiento, y si se pierde algún servidor, siempre se puede clonar la información de otro de los esclavos).

RAID 1: O en espejo. No es tan rápido como RAID 0, pero proporciona redundancia; puede perder un disco y seguirá funcionando. El mayor rendimiento se aplica sólo a las lecturas. En el momento que estén los datos de cada disco duplicados en espejo, el sistema puede decidir leer los datos en paralelo desde los discos. El resultado, es que, en el caso más óptimo, puede leer la misma cantidad de datos en bruto al mismo tiempo. En el rendimiento de escritura, es tan bueno como si fuese un solo disco.

RAID 5: es un RAID 0 con bloques de paridad distribuido. Es el más utilizado. Cuando los fondos están ajustados y la redundancia es más importante que el rendimiento, es la mejor de las opciones.

5.3.1. Arquitecturas de replicación

Dado que el sistema de replicación de MySQL es relativamente simple comparado con otras bases de datos comerciales, se puede utilizar para construir arquitecturas más complejas que solucionen diferentes tipos de problemas.

Las reglas de la replicación son:

- Cada esclavo debe tener un único ID de servidor.
- Un esclavo debe tener solamente un maestro.
- Un maestro puede tener varios esclavos.
- Los esclavos también pueden ser maestros para otros esclavos

Estado del maestro: con el comando `SHOW MASTER STATUS`, el maestro informará sobre su estado de replicación.

Esta información incluye el nombre de archivo y posición del registro binario actual, en donde se escribirá la siguiente consulta.

Con respecto al estado del esclavo a través del comando `SHOW SLAVE STATUS`.

5.4 AMENAZAS AL SISTEMA

5.4.1 Introducción

Tan importante es la seguridad de la base de datos MySQL como el entorno en el que esté instalado. Es imprescindible auditar también el sistema operativo en el que se encuentre tanto el servidor de la base de datos como los clientes, así como controlar las contraseñas, usuarios y privilegios otorgados.

5.4.2. Control del sistema operativo

MySQL es un SGBD muy funcional, puesto que ofrece la posibilidad de instalarse en múltiples sistemas operativos. Dos de los más extendidos en la actualidad son el sistema operativo ofrecido por Microsoft: Windows, y UNIX.

De manera directa o indirecta el acceso a la base de datos está supeditado al sistema operativo, de manera que, cualquier usuario que desee realizar cualquier acción sobre la base de datos deberá en primer lugar acceder al sistema operativo en el que esté instalado. De ahí surge la necesidad de auditar y controlar no sólo la base de datos sino también el sistema operativo, puesto que si el sistema operativo no es seguro será muy complicado garantizar la seguridad de la base de datos.

Como ya se ha explicado en el apartado anterior, el SGBD MySQL utiliza una serie de archivos y carpetas que están almacenados en el sistema operativo y que requieren también un control estricto sobre accesos y permisos. Se comentaba que una posibilidad era establecer una serie de permisos de carpeta, posibilidad que UNIX ofrece

fácilmente utilizando el comando CHMOD que modifica las opciones de escritura, lectura y posibilidad de ejecución sobre un directorio, carpeta o fichero.

En Windows se ofrecen también múltiples posibilidades, como ocultar una carpeta a un usuario determinado si éste no tiene una contraseña de acceso. Otra posibilidad es establecer cuentas de usuario, con lo que no sólo se controlan los archivos y carpetas visibles y permitidas para un determinado usuario, sino que también se puede llevar un control más exhaustivo sobre las sesiones iniciadas por éste, y por qué no, desde qué equipos se conecta.

Otro punto a tener en cuenta en versiones antiguas de MySQL: si el sistema operativo en el que estuviera instalado el SGBD MySQL fuese UNIX el auditor podría determinar si el directorio de datos contiene o no archivos inseguros ejecutando el comando ls –l . Únicamente tendría que buscar los archivos o directorios que tuviesen activados los permisos “grupo” u “otros” (mysqlgrp), con lo que otra persona podría tomar esos datos, puesto que el directorio no tiene una buena gestión de las restricciones.

Los sistemas operativos utilizan capas de seguridad de la información con los que el auditor puede beneficiarse en un momento dado, controlando el sistema de privacidad de dicha información.

El auditor deberá comprobar que los puertos por los que puede haber conexiones externas (si estas son necesarias para la organización) están abiertos y seguros, y que existe un firewall actualizado que aísla al sistema operativo con el exterior. La lista de comprobaciones que debería llevar a cabo el auditor son:

- Intentar escanear los puertos desde Internet utilizando alguna herramienta como *nmap*. MySQL utiliza el puerto 3306 por defecto. Este puerto no debería ser accesible desde lugares no confiables. Otra manera simple de probar si el puerto MySQL está abierto o no, es intentar el siguiente comando desde alguna máquina remota, donde *server_host* es la máquina en la que el servidor MySQL se estuviese ejecutando, el auditor debería introducir el siguiente comando:

```
> telnet server_host 3306
```

Si consiguiese conectar y aparecen algunos caracteres extraños, el puerto está abierto, en caso contrario el puerto estaría bloqueado.

- Comprobar que los flujos de datos de MySQL están cifrados ejecutando el siguiente comando (para Linux):

```
tcpdump –l –i eth0 –w –src or dst port 3306 | strings
```

5.4.3. Control de usuarios y contraseñas

Como ya se comentó en el apartado anterior, el control de los usuarios deberá ser exhaustivo. En relación al sistema operativo, se deberá tener controlado el acceso que éstos tienen al sistema, administrando cuentas de usuario con privilegios restringidos que vendrán definidos por el sistema operativo que se esté auditando.

El auditor debe comprobar que existe una gestión eficaz de las contraseñas otorgadas a los usuarios:

- Verificar que éstas no sean fácilmente deducibles a partir de datos personales de los usuarios como fechas de nacimiento, nombres de personas (hijos, padres, hermanos...).

- Comprobar si las contraseñas tienen una fecha de expiración, en la que cada cierto tiempo pierden validez, y hay que renovar la contraseña.

- Comprobar si existe bloqueo de la cuenta si se cometen “x” errores en la introducción de la contraseña.

- Comprobar si existen sistemas de cifrado de contraseñas, utilizando algún algoritmo de cifrado para dificultar el secuestro de la misma.

- Comprobar que la contraseña no existe en el diccionario, ya que es muy insegura.

Se podría realizar el siguiente cuestionario para poder auditar el **control de acceso genérico**:

- ¿Se generan logs de auditoría del control de acceso?
- ¿Cuándo se almacenan, ante qué eventos?
    - Login exitoso **o** login fallido
    - Procedimientos de cambios de contraseñas satisfactorios
    - Procedimientos de cambios de contraseñas fallidos
    - Bloqueo de un usuario concreto
    - Utilización de herramientas del sistema
    - Modificación de ciertos datos (como datos de configuración, datos críticos, datos de otros usuarios)
- Acceso a Internet: información descargada, páginas visitadas, etc.
- Alertas de virus
- ¿Dónde se almacenan?
- ¿Quién tiene acceso a los logs?
- ¿Por cuanto tiempo permanecen guardados?
- Se borran cuando expira ese tiempo o se genera una estadística comprimida de los mismos y se guarda un análisis de ellos solamente?
-¿Qué datos se almacenan en los logs? ¿Se almacenan los siguientes datos?
    - Para todos los eventos:
    - Fecha y hora del evento
    - Tipo de evento (Ej. Login, modificación de datos, etc.)
    - Identificador (ID) de usuario
    - Origen del evento (desde qué terminal)
    - Acceso a Internet: cookies guardadas, archivos descargados, servicios utilizados

Realizar un cuestionario sobre el sistema de este tipo, puede resultar de gran ayuda al auditor.

De manera más específica en el entorno MySQL se podrían realizar las siguientes comprobaciones:

- ¿Se cifran las contraseñas?
- ¿Está restringido el acceso a la conexión interna?
- ¿Tiene el software de la base de datos los permisos de sistema operativo adecuado?
- ¿Tienen que aprobar los propietarios de los datos el acceso para escribir en sus tablas?
- ¿Hace uso de los roles el administrador para simplificar la administración del acceso de los usuarios?
- ¿Se han parcheado los “bugs” o vulnerabilidades de MySQL?
- ¿Se realizan copias de seguridad del sistema periódicamente?

5.5 COMUNICACIONES

5.5.1. Introducción

En los dos apartados anteriores se han estudiado los elementos relativos a un SGBD MySQL que todo buen auditor debería revisar para asegurar el control y seguridad del mismo.

Actualmente es difícil encontrar un sistema que no esté conectado a la red puesto que la distribución de la capacidad de proceso entre varios servidores y la posibilidad de compartir la información mejora notablemente los recursos accesibles y disponibles de la organización.

El sistema MySQL ofrece la posibilidad de utilizar su servidor de manera que, los usuarios que estén conectados a éste no tengan el requisito indispensable de tener instalado dicho servidor en su propio equipo, sino que a través de programas cliente (como *mysql*), tener acceso a la base de datos.

Por ello, es necesario comprobar que la red es segura, y que la comunicación puede realizarse con total confianza, y el auditor deberá controlar que se cumplan algunos requisitos necesarios para que esto suceda.

5.5.2. Conexiones seguras SSL en MySQL

A partir de MySQL 4.1 se incluye soporte para conexiones seguras (cifradas) entre los clientes MySQL y el servidor, utilizando el protocolo SSL (*Secure Sockets Layer*).

El auditor debe conocer que, por defecto, MySQL utiliza conexiones sin cifrar entre el cliente y el servidor. Esto significa que cualquiera con acceso a la red podría ver el tráfico y mirar los datos que están siendo enviados o recibidos. Incluso podría cambiar los datos mientras están aún en tránsito entre el cliente y el servidor.
Para mejorar la seguridad un poco, se podría comprimir el tráfico entre el cliente y el servidor utilizando la opción `--compress` cuando ejecute programas cliente.

Cuando se necesita mover información sobre una red de una manera segura, una conexión sin cifrar es inadmisible. El **cifrado** es la manera de hacer que cualquier dato sea ilegible. De hecho, hoy en día la práctica requiere muchos elementos adicionales de seguridad en los algoritmos de cifrado. Deben resistir muchos tipos de ataques conocidos.

El protocolo SSL utiliza diferentes algoritmos de cifrado para asegurarse de que los datos recibidos a través de una red pública son seguros. Tiene mecanismos para detectar cambios de datos, pérdidas, o reenvíos. SSL también incorpora algoritmos que proveen de verificación de identidad, utilizando el estándar X509.

El estándar X509 hace posible identificar a alguien en Internet. Es utilizado comúnmente en aplicaciones de comercio electrónico. En resumen, debe haber alguna compañía, llamada ``Autoridad Certificada" (CA) que asigna certificados electrónicos a cualquiera que los necesita. Los certificados se basan en algoritmos de cifrado asimétricos que tienen dos claves de cifrado (una pública, y otra secreta). El propietario

de un certificado puede enseñárselo a otra entidad como prueba de su identidad. Un certificado consiste en la clave pública de su propietario. Cualquier dato cifrado con esta clave pública puede ser solo accedido utilizando la clave secreta correspondiente, que está en posesión del propietario del certificado.

En MySQL hay diferentes maneras de limitar los tipos de conexión para una cuenta:

• Si una cuenta no tiene requerimientos de SSL o X509, las conexiones sin cifrar se permiten siempre que el nombre de usuario y la clave sean válidas. De cualquier manera, se pueden también utilizar conexiones cifradas, si el cliente tiene los certificados y archivos de claves apropiados. El auditor deberá estudiar si la cuenta tiene o no los requerimientos necesarios.

• La opción `REQUIRE SSL` limita al servidor para que acepte únicamente conexiones cifradas SSL para la cuenta.

```
mysql> GRANT ALL PRIVILEGES ON test.* TO
'root'@'localhost' IDENTIFIED BY 'clave' REQUIRE
SSL;
```

• `REQUIRE X509` significa que el cliente debe tener un certificado pero que el certificado exacto, entidad certificadora y sujeto no importan. El único requerimiento es que debería ser posible verificar su firma con uno de los certificados CA.

```
mysql> GRANT ALL PRIVILEGES ON test.* TO
'root'@'localhost' IDENTIFIED BY 'clave' REQUIRE
X509;
```

• `REQUIRE ISSUER` 'issuer' coloca una restricción en la conexión mediante la cual el cliente debe presentar un certificado X509 válido, emitido por la CA 'issuer'. Si el cliente presenta un certificado que es válido pero tiene un emisor diferente, el servidor rechaza la conexión.

La utilización de certificados X509 siempre implica cifrado, así que la opción SSL no es necesaria.

```
mysql> GRANT ALL PRIVILEGES ON test.* TO
'root'@'localhost'
```

• `REQUIRE SUBJECT` 'subject' establece la restricción a los intentos de conexión de que el cliente debe presentar un certificado X509 válido con sujeto 'subject'. Si el cliente presenta un certificado que, aunque válido, tiene un sujeto diferente, el servidor rechaza la conexión.

El auditor deberá comprobar que se utilizan conexiones seguras entre el cliente y el servidor.

Hay que tener en cuenta que las versiones compiladas de MySQL que viene con la mayor parte de las distribuciones de Linux (y aquellas que están disponibles en el sitio Web de MySQL.com) no tienen activado SSL de forma predeterminada, por lo que para comprobar en qué estado se encuentra el servidor habría que mirar el valor de la variable `have_openssl.`

Para comprobar si un servidor *mysqld* tiene soporte OpenSSL el auditor debería examinar el resultado de teclear la siguiente sentencia:

```
SHOW VARIABLES LIKE ‘have_openssl’
```

Si el resultado es NO, entonces el administrador no cuenta con los nieveles de seguridad a nivel de bases de datos adecuados.

También se podría comprobar que en el script configure están las opciones: `--with –vio` y `– with –opensll`

5.5.3. Conexiones de programas cliente al servidor

El acceso a las bases de datos de MySQL por los programas cliente que conectan con el servidor en la red está controlado por los contenidos de las tablas de concesión. Estas tablas están localizadas en la base de datos mysql y están inicializadas durante el proceso de instalación de MySQL.

Hay dos etapas en el control de acceso del cliente cuando usa MySQL. La primera etapa es cuando intenta conectar con el servidor. El servidor mira la tabla “user” almacenada en la base de datos mysql para ver si puede encontrar una entrada que coincida con su nombre, el host desde el que se está conectando y la contraseña que ha proporcionado. Si no hay coincidencias, no podría conectarse. Si la hay establece la conexión y continúa a la segunda etapa. En ésta, por cada consulta que emite, el servidor verifica las tablas para ver si tiene o no suficientes privilegios para realizar la consulta. Dicha etapa continúa hasta que termine su sesión con el servidor.

El auditor llegada esta fase de la auditoría debería haber realizado las consultas sobre los privilegios sobre tablas, columnas y bases de datos de los usuarios que tienen la posibilidad de conectarse desde fuera del host local. Como ya se ha comentado anteriormente la consulta se realiza utilizando el comando SELECT sobre la base de datos *mysql*.

5.6. COPIAS DE SEGURIDAD O BACKUP

A pesar de las medidas de prevención que se puedan tomar para evitar errores, fallos y ataques en el SGBD y en el sistema operativo, siempre quedará una posibilidad por muy pequeña que sea, que el sistema falle por cualquiera de las razones comentadas en los apartados anteriores y no es posible asegurar completamente que estará exento de cualquier amenaza y/o fallo.

Por esta razón es también muy importante mantener un buen plan de gestión de contingencias posibles para poder recuperar lo antes posible el control en una situación caótica. Una manera muy útil de almacenar información que será muy útil en caso de

que el sistema falle, es realizar copias de seguridad de las bases de datos, tablas y aquella información crítica.

El auditor deberá realizar un cuestionario parecido al que propone María Dolores Cerini en su Tesis sobre seguridad de la información (2003):

- ¿Con qué frecuencia hacen los backups?
- ¿Qué datos se almacenan? (datos y programas de aplicación y sistemas, equipamiento, requerimientos de comunicaciones, documentación)
    - Software de base y su configuración:
        - ¿Se hacen discos de inicio de Windows?
        - ¿Hay imágenes *Ghost* de las máquinas?
        - ¿Se hacen backups de la configuración de red?
    - Software aplicativo.
    - Parámetros de sistema.
    - Logs e informes de auditorías.
    - Datos
    - ¿Qué más?
    - Backups del Hardware.
        - Modalidad externa: ¿contratan un tercero que proporcione los insumos necesarios en caso de emergencia?
        - Modalidad interna: si tienen más de un local, en ambos locales deben tener señalados los equipos, que por sus características técnicas y capacidades, son susceptibles de ser usados como equipos de emergencia del otro local. ¿Se realizan actividades en la empresa?
        - Radio: ¿si se cae un nodo de la radio, qué pasa? ¿Hay algún servicio técnico o de respaldo para ésto?

- ¿Hay backup especiales (con datos distintos o particulares)? ¿Cada qué periodo de tiempo se hacen? ¿Qué datos guardan?
- ¿Qué tipo de backup hacen? (backups normales, backups incrementales, backups diferenciales) ¿En qué áreas o datos usan incrementales, en cuáles normales, etc.?
- ¿En qué medio se almacena? ¿Con qué dispositivo se hace?
- ¿Cómo es la rotación de los medios de backup? ¿En una semana, un mes?

Figura 25*. Cuestionario de auditoría de backups*

- ¿Con qué aplicación se hacen? ¿Con algún tipo especial de aplicación de manejo de backup? ¿Es una del sistema operativo, del administrador de archivos u otra? ¿Utilizan archivos de tipo específicos o archivos .zip, por ejemplo?
- ¿Hay herramientas de back up automáticas, o sea que a través de una agenda hacen las copias?
- ¿Quién es el encargado o el responsable? ¿Los hace el administrador de sistemas?
- ¿Tienen formalizados los procedimientos de back up? ¿Existe un procedimiento escrito? ¿Si falta el responsable de backup quién los hace?
- ¿Existen procedimientos escritos para recuperar archivos copiados, o un Plan de backup?
- ¿Hacen pruebas periódicas de recuperación de backups?
- ¿Quién puede levantar los archivos de los usuarios, los backups de Mis prioridades? ¿Según qué se determinó la prioridad de las máquinas: según un análisis de impacto, según la confidencialidad de la información?
- ¿Los backups se almacenan dentro y fuera del edificio? ¿Estos lugares son seguros?
- ¿Cómo se rotulan e identifican?
- ¿Hay documentación escrita sobre los backups hechos, sus modificaciones, fechas, etc.?
- ¿Se necesita algún dispositivo (llaves, tarjeta) para entrar al almacén de cintas?
- ¿Se crean disco de inicio de Windows?
- ¿Hay información afuera de la red interna de la empresa que sea valiosa? ¿El web host tiene datos importantes de usuarios? ¿Se hacen backups de estos datos? ¿Dentro de la empresa o por el web host?
- ¿Hay backups de las páginas web y de sus actualizaciones?
- ¿Existen procedimientos automáticos para que, en caso que un usuario cometa un error en la base de datos, ésta pueda volverse a su estado anterior? ¿Cómo se hace?

Figura 26. *(Cont) Cuestionario de auditoría de backups*

5.7. RECUPERACION ANTE DESASTRES

La recuperación ante desastres tiene que tener especial relevancia a la hora de comprobar que una base de datos está correctamente configurada y que es robusta. En primer lugar hay que definir qué se entiende por desastre.

Un desastre es un evento por el cual partes significativas de los datos se corrompen o se pierden. Algunos ejemplos de éstos son:

- Eliminación accidental de datos.
- Fallo en hardware
- Fallo en software
- Robo del servidor
- Destrucción física del servidor, etc.

En cualquier momento puede producirse cualquiera de estos hechos. La posibilidad de que ocurra uno de ellos es baja, pero no es imposible.

5.8. SEGURIDAD FÍSICA

Es muy importante ser consciente de que por más que nuestra empresa sea la más segura desde el punto de vista de ataques externos y ataques internos, la seguridad de la misma será nula si no se ha previsto como combatir un incendio.

La seguridad física es uno de los aspectos más olvidados a la hora del diseño de un sistema informático. Si bien algunos de los aspectos tratados a continuación se prevén, otros, como la detección de un atacante interno a la empresa que intenta acceder físicamente a una sala de operaciones de la misma, no. Esto puede derivar en que para un atacante sea más fácil lograr tomar y copiar algún dato que se encuentre en algún soporte como CD o DVD, que intentar acceder vía lógica a la misma.

Así, la **Seguridad Física** consiste en la "aplicación de barreras físicas y procedimientos de control, como medidas de prevención y contramedidas ante amenazas a los recursos e información confidencial". Se refiere a los controles y mecanismos de seguridad dentro y alrededor del centro de trabajo de la organización así como los medios de acceso remoto al y desde el mismo; implementados para proteger el hardware y medios de almacenamiento de datos.

Las principales amenazas que se prevén en la seguridad física son:

1. Desastres naturales, incendios accidentales tormentas e inundaciones.
2. Amenazas ocasionadas por el hombre.
3. Disturbios, sabotajes internos y externos deliberados.

No hace falta recurrir a acciones complicadas para obtener la máxima seguridad en un sistema informático, además de que la solución sería extremadamente cara. A veces basta recurrir al sentido común para darse cuenta que cerrar una puerta con llave o cortar la electricidad en ciertas áreas siguen siendo técnicas válidas en cualquier entorno.

El auditor deberá comprobar que:

- Existe una adecuada protección física y mantenimiento permanente de los equipos e instalaciones que conforman los activos de la empresa.

- El acceso físico a las **áreas críticas** a toda persona no autorizada está restringido.

- Se deberá asegurar que todos los **individuos** que entren a áreas restringidas se identifiquen y sean autenticados y autorizados para entrar.

- El **área** donde se encuentran los servidores, y demás equipamiento crítico solo debe tener permitido el acceso a los administradores.

- El personal de los centros de procesamiento así como el personal contratado sólo podrá permanecer en las instalaciones de las empresas durante el **horario autorizado.** Se deberá establecer un procedimiento de autorización para el personal que deba permanecer fuera de su horario habitual de trabajo.

- Se debe realizar un adecuado mantenimiento y **prueba de los procedimientos** para la restricción de acceso físico, así como de los

dispositivos de seguridad para la prevención, detección y extinción del fuego.

En el control de **acceso a equipos**:

- Los **lectores de CD** deberán deshabilitarse en aquellas máquinas en que no se necesiten.

- Los equipos de la empresa deberán tener un *password* de administrador que deberá gestionar el administrador del sistema.

- Los servidores deberán tener una **llave de bloqueo** de hardware.

- Cualquier **dispositivo externo** que no se encuentre en uso, deberá permanecer guardado bajo llave.

- El administrador deberá realizar exámenes **periódicos** para comprobar:
    - la correcta instalación de los dispositivos de los equipos,
    - su buen funcionamiento.

- Los **servidores deberán apagarse** automáticamente una vez que han cerrado todas las sucursales de la empresa.

En el control de **dispositivos de soporte:**

- Deberán existir los siguientes **dispositivos de soporte** en la empresa: aire acondicionado y calefacción. Extintores que deberán estar instalados en lugares estratégicos de la empresa para extinguir incendios en equipos eléctricos.

5.9. PROBLEMAS COMUNES Y LIMITACIONES

El auditor debe prever que hay situaciones en las que MySQL puede comportarse de una forma inesperada. Entre todas ellas las más significativas son:

1. No es posible revocar ciertos privilegios: La base de datos proporciona un método para otorgar privilegios –como se ha visto anteriormente a través de las tablas USER, HOST, etc-, pero no dispone de un sistema paralelo para denegar privilegios, por lo que no se puede denegar un privilegio más específico que el que haya otorgado el usuario dado.

   Para solucionar el problema se tienen que eliminar todos los privilegios de usuario y otorgar uno a uno aquellos que se desea que tenga ese usuario

2. Los criterios de bases de datos y *hosts* no pueden excluir otros criterios: Si se desea restringir el acceso únicamente a una máquina no hay forma de hacerlo. La única manera es bloquear la máquina a nivel de red o añadir un registro con una contraseña incorrecta en la tabla USER.

   Esto puede ser debido a que MySQL fue diseñado para hacer sencillo el mecanismo de otorgar privilegios, pero no para denegarlos.

3. Los privilegios no desaparecen cuando sí lo hacen los objetos: No se produce una limpieza de los permisos otorgados a aquellos que se eliminan.

4. Seguridad en los sistemas operativos: Como ya se ha explicado en apartados anteriores, a pesar de que las tablas de permisos estén bien diseñadas y seguras poco pueden hacer si un pirata informático obtuviese acceso de administrador al servidor, ya que éste lo tendría sencillo si deseara copiar todos los archivos de datos en otra máquina utilizando MySQL.

5. Directrices a seguir: Son unas pautas básicas, como por ejemplo, no ejectuar MySQL utilizando una cuenta privilegiada (ya que el usuario “root” de UNIX y el usuario “administrador” de Windows poseen el control absoluto del sistema), por lo que si descubre un fallo de seguridad y se está ejecutando como usuario privilegiado un posible ataque sería mucho más sencillo. Habría que tener también el sistema operativo actualizado, así como restringir el inicio de sesión en el ordenador con la base de datos, hacer auditorías del servidor, etc.

6. Conexiones sólo desde máquinas locales: Si se tiene la posibilidad sería conveniente limitar las conexiones externas, de manera que se reduciría drásticamente el número de posibilidades que tiene una persona ajena a la entidad de obtener los datos del servidor MySQL.

7. Cortafuegos: Es importante permitir conexiones desde máquinas con autorización. La manera más eficaz es utilizar un *firewall* denegando todas las conexiones de forma predeterminada y añadir las reglas (si se puede) que permitan el acceso a algunos servicios que necesiten ser accesibles desde otras máquinas.

8. Sin ruta predeterminada: Si se considerara la posibilidad de no tener configurada una ruta predeterminada para los servidores MySQL protegidos por un *firewall*. De esa forma, aunque la configuración del cortafuegos estuviera en riesgo, y alguien intentara conectarse al servidor desde fuera, los paquetes nunca volverían a él.

   Por ejemplo, si el servidor MySQL estuviera en la máquina 192.168.0.13 y que la red local utiliza una máscara 255.255.255.0 Con esa configuración, cualquier paquete de datos que proviniese de 192.168.0.0/24 sería considerado como local, ya que podría alcanzarse directamente a través de la interfaz de red asociada. El tráfico que provenga de cualquier otra dirección sería redirigido a una pasarela para que alcanzase su destino final, y dado que no existe una ruta predeterminada, no habría ninguna forma de que esos paquetes alcanzasen la pasarela y pudiesen llegar a su destino.

Si se tuviera la necesidad de permitir el acceso a unas cuantas máquinas externas (aparte del servidor con el cortafuegos), se podrían añadir rutas estáticas para ellas. De esa forma, se aseguraría que el servidor responde a cuantas menos máquinas externas posibles.

9. Cifrado de conexiones: Ya se ha visto en apartados anteriores la necesidad de cifrar las conexiones, al hacer eso, se hace mucho más complicado que cualquier persona tenga más dificultades a la hora de intentar interceptar las conexiones y espiar los datos que se transmiten.
   Como beneficio añadido, algunos algoritmos de cifrado producen como resultado la compresión de los datos, de manera, que los datos no sólo son más seguros, sino que además se ahorra en ancho de banda.

10. Redes privadas virtuales: Una empresa que tenga varias oficinas ubicadas en diferentes localizaciones podría configurar una red privada virtual (VPN) entre ellas utilizando diferentes tecnologías. Una solución utilizada comúnmente en estos casos es que los routers externos de cada oficina cifraran el tráfico destinado a las otras oficinas.

11. TCP *Wrappers*: Hay que saber también que MySQL puede ser compilado para tener compatibilidad con TCP *Wrappers* en sistemas UNIX. Si no es posible utilizar un completo cortafuegos, los TCP *Wrappers* proporcionan un nivel básico de defensa. Se podría obtener un control adicional sobre con qué máquinas se pueden comunicar los servidores MySQL, sin necesidad de cambiar las tablas de permisos.

    Para poder utilizar los TCP Wrappers, se necesita compilar el código de MySQL y utilizar la opción `–with-libwrap` para configurarlo, de forma que sea capaz de encontrar los archivos de cabecera adecuados en el sistema operativo:

```
$ ./configure –with-libwrap=/usr/local/tcp_wrappers
```

Suponiendo que se tenga una entrada en el archivo `/etc/hosts.deny` que rechaza todas las conexiones de forma predeterminada:

```
# deny all connections
ALL: ALL
```

Habría que introducir también una entrada apropiada en `/etc/services` para MySQL.

```
Mysql 3306/tcp   #MySQL Server
```

5.10. Configuración de MySQL 5.1 en Windows

En primer lugar explicar que existen actualmente (y más concretamente en la página web de MySQL) multitud de versiones de la base de datos que pueden ser configuradas en varios sistemas operativos. Por lo que debería tenerse claro cual es la distribución de MySQL que se quiere configurar, sobre qué sistema operativo, y si el ordenador donde se instalará cumple los requisitos necesarios.

Según el manual de ayuda que proporciona dicha página, los pasos a seguir antes y durante la instalación son los siguientes:

1 .Debe determinarse si la pla taforma donde se desea hacer la instalación está soportada. Nótese que no todos los sistemas soportados son igualmente adecuados para ejecutar MySQL. En algunas plataformas el funcionamiento será mucho más robusto y eficiente que en otras.

2. **Debe elegirse la distribución que se instalará.** Hay varias versiones de MySQL disponibles, y la mayoría lo están en varios formatos de distribución. Se puede elegir entre distribuciones prearmadas que contienen programas binarios (precompilados) o bien código fuente. En caso de duda, debe elegirse una distribución binaria. También se provee acceso público al código fuente para quienes deseen ver los desarrollos más recientes y colaborar en el testeo de código nuevo.

3. **Descargar la distribución que se desea instalar.**

4. **Instalar la distribución.** Para instalar MySQL desde una distribución binaria, a partir de una distribución de código fuente o desde el directorio de desarrollo actual.

5. **Realizar cualquier ajuste que sea necesario con posterioridad a la instalación.**

El ejemplo elegido es la distribución de MySQL 5.1. en Windows XP

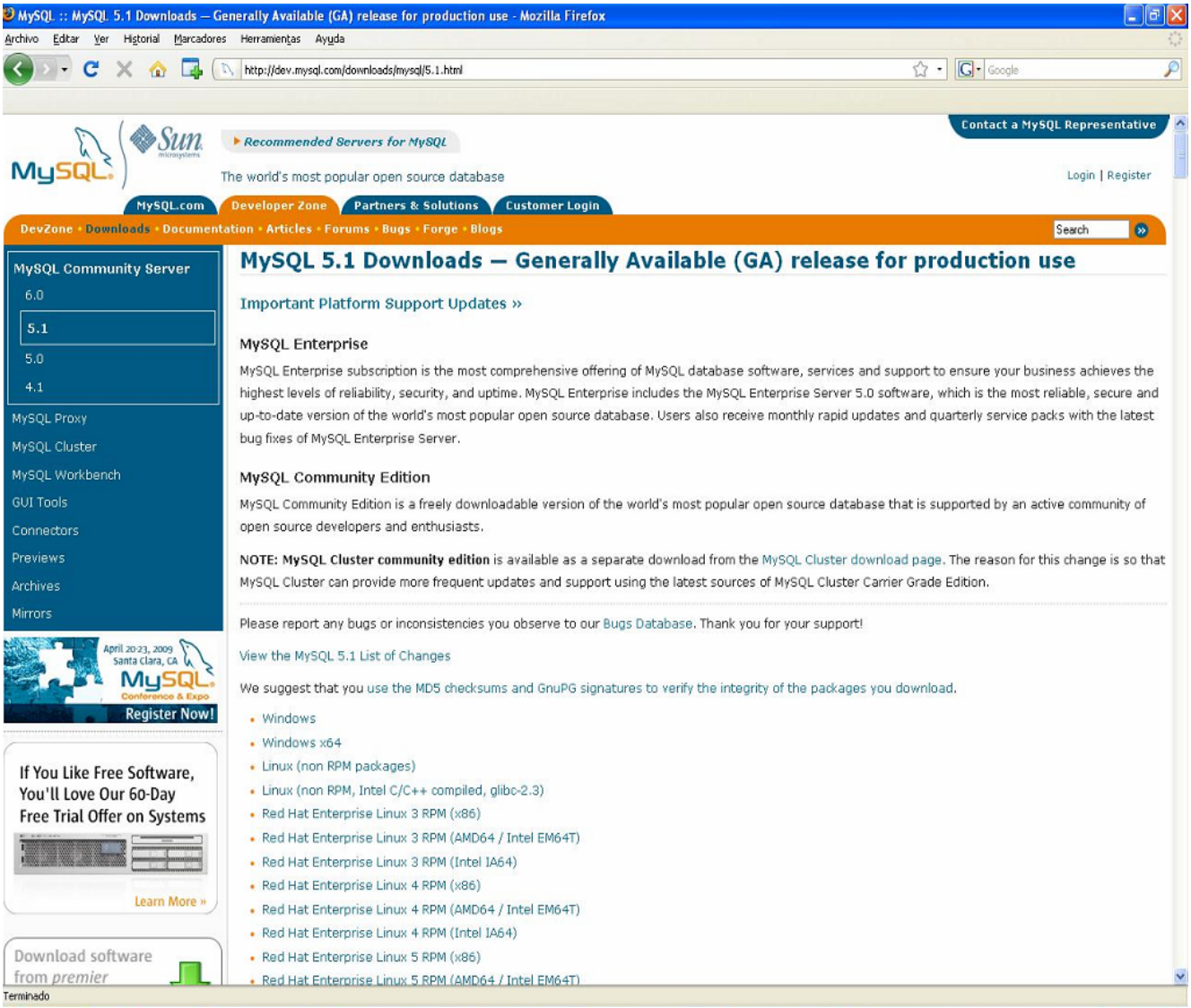


Figura 27. *Página de descargas del programa*

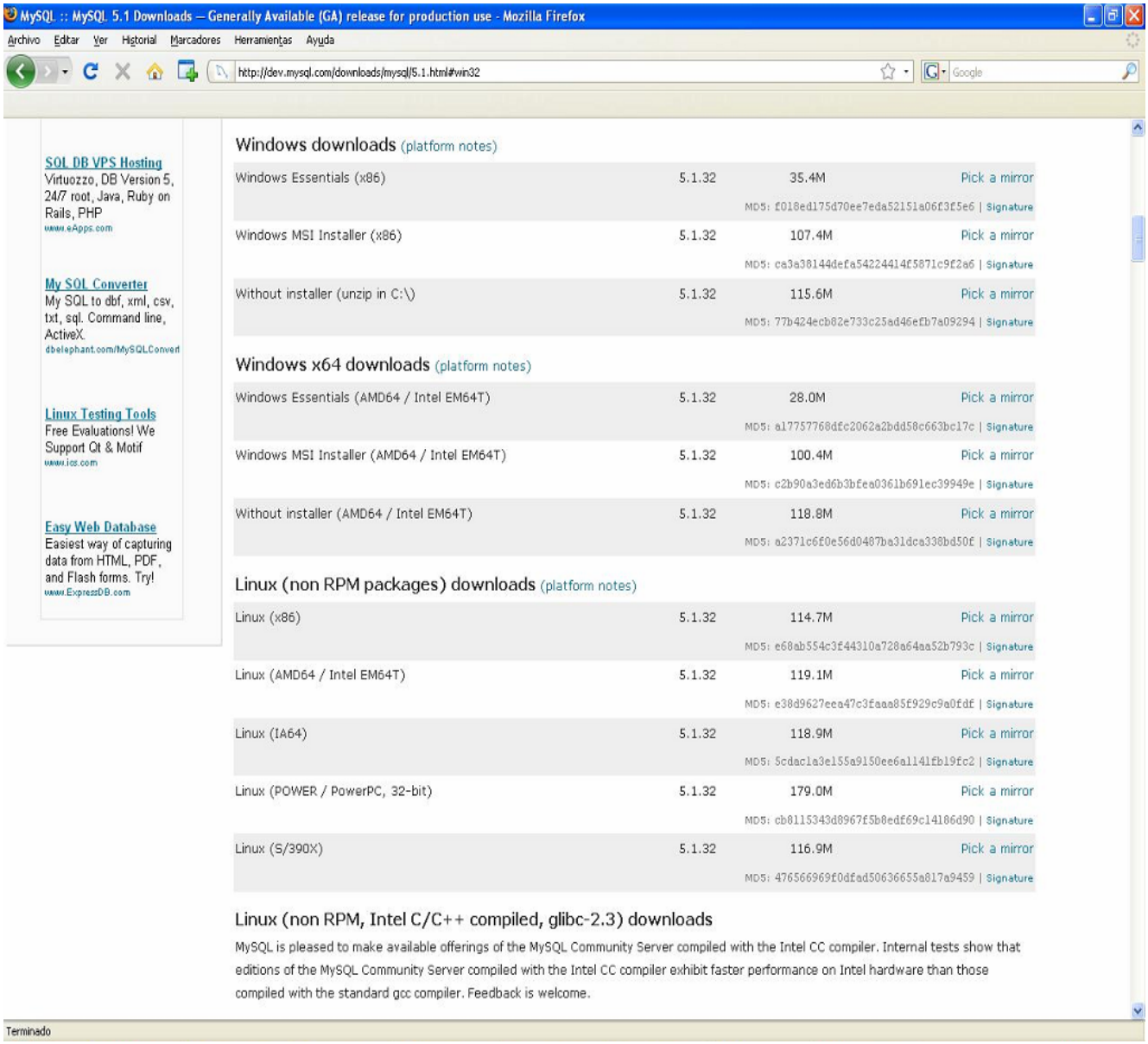


Figura 28. *Descarga de la página Web del programa: la versión que se desee, según el sistema operativo.*

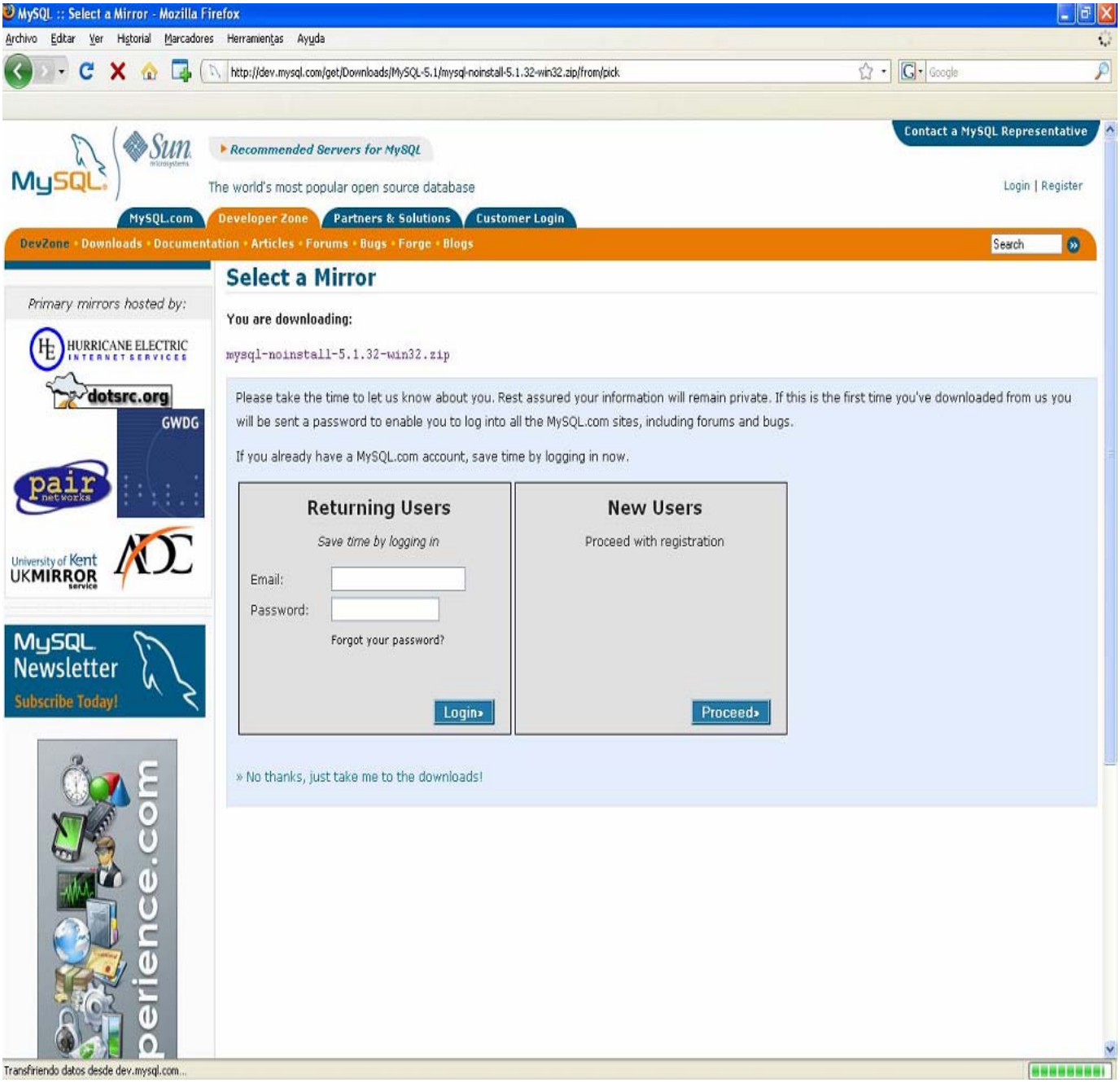


Figura 29. *Introducción de usuario y contraseña (si ya está dado de alta) o nuevos usuarios*

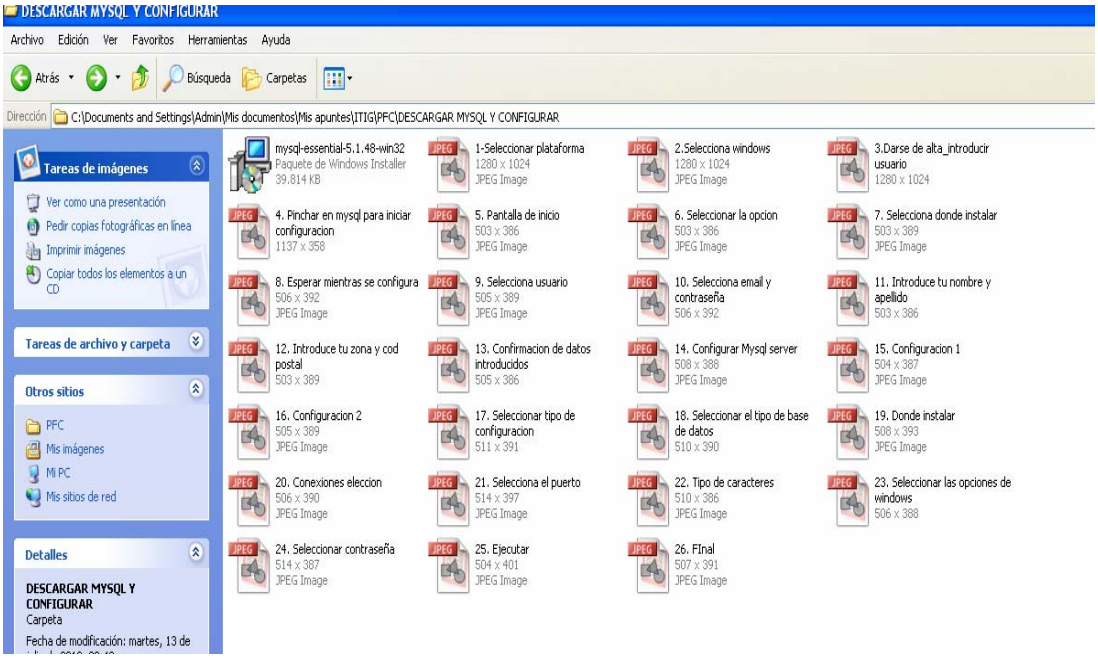


Figura 30. *Selección de la carpeta en la que se desea almacenar el ejecutable*

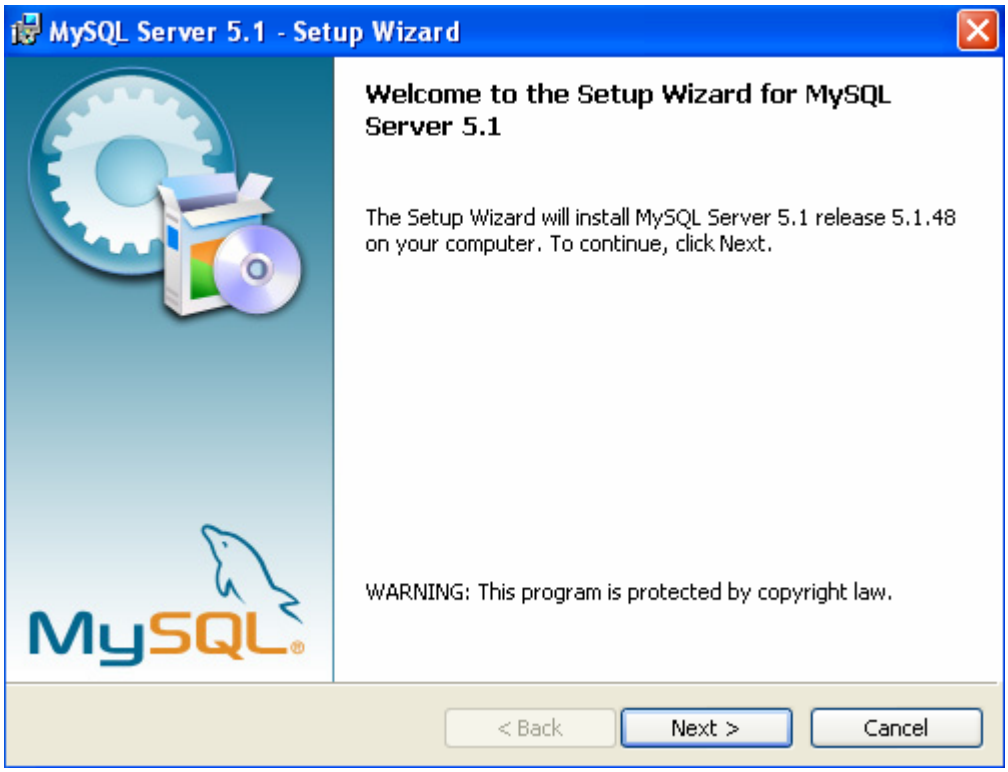


Figura 31. *Pantalla de inicio de la instalación*

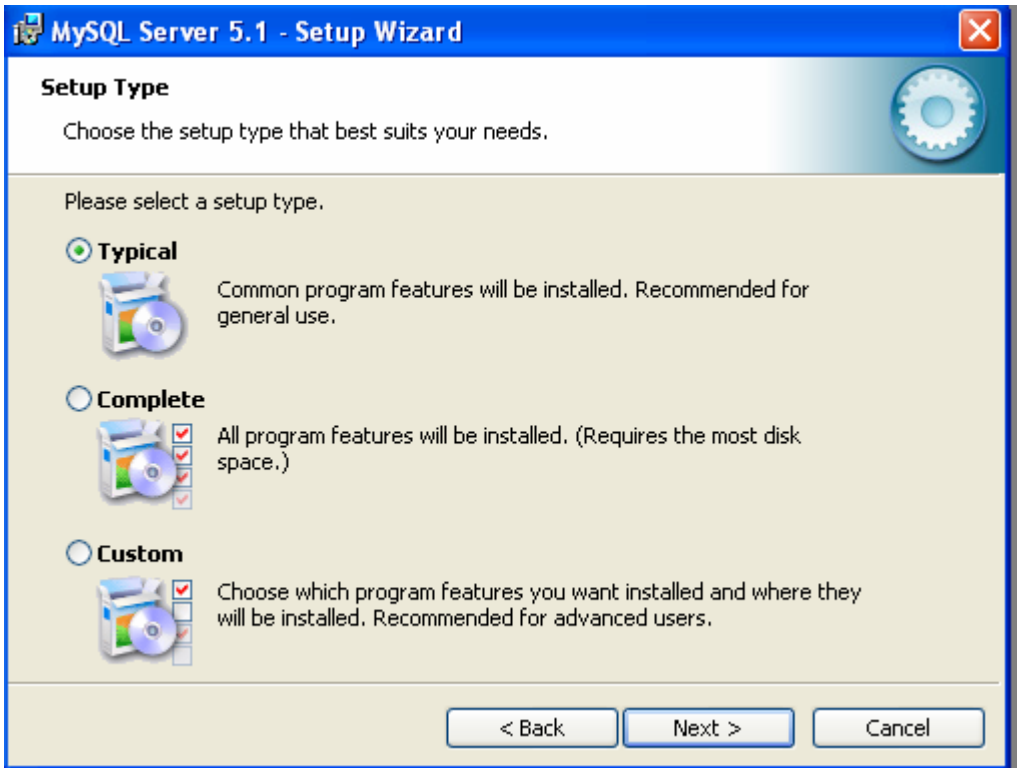


Figura 32. *Selección de la modalidad de instalación deseada*

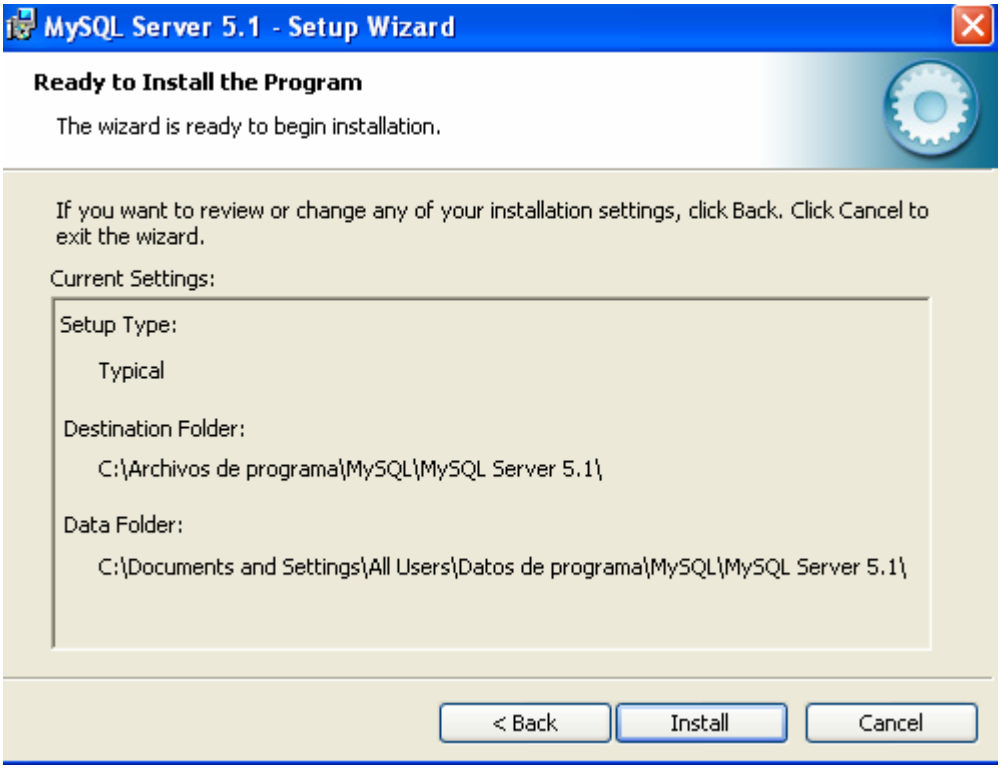


Figura 33. *Elección de la ruta destino en la que se instalará la aplicación*

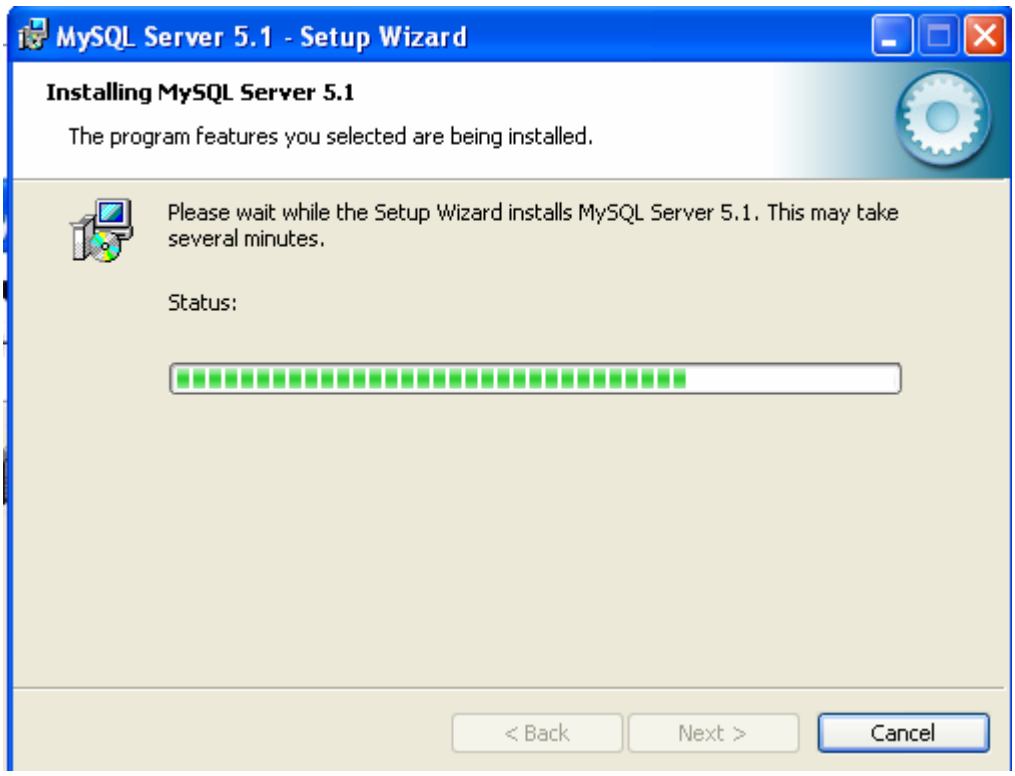


Figura 34. *Comienzo de la instalación*

Figura 35. *Creación de una nueva cuenta de MySQL.com*

Figura 36. *Cumplimentación de los campos, dirección de correo electrónico y contraseña*

Figura 37. *Cumplimentación de los campos obligatorios: nombre y apellidos*

Figura 38. *Cumplimentación de los campos obligatorios: código postal, país, estado*

Figura 39. *Pantalla de confirmación de datos introducidos*

Figura 40. *Pantalla de comienzo de configuración del servidor*

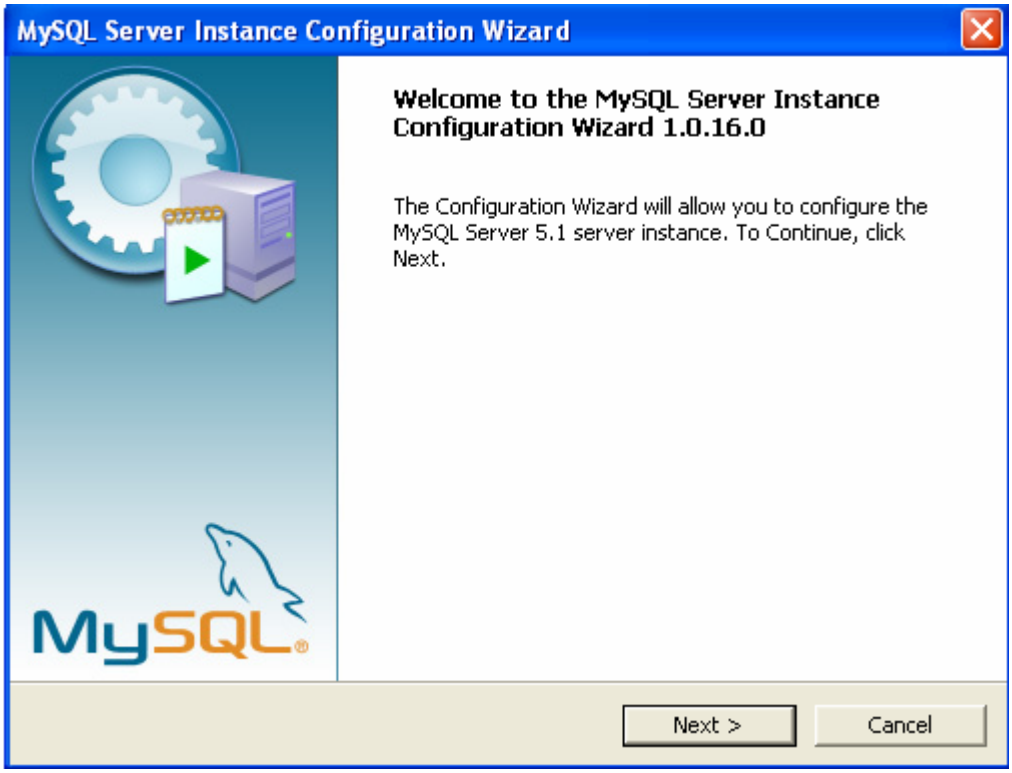


Figura 41. *Segunda pantalla de comienzo de instalación. Pulsar "next" para continuar*

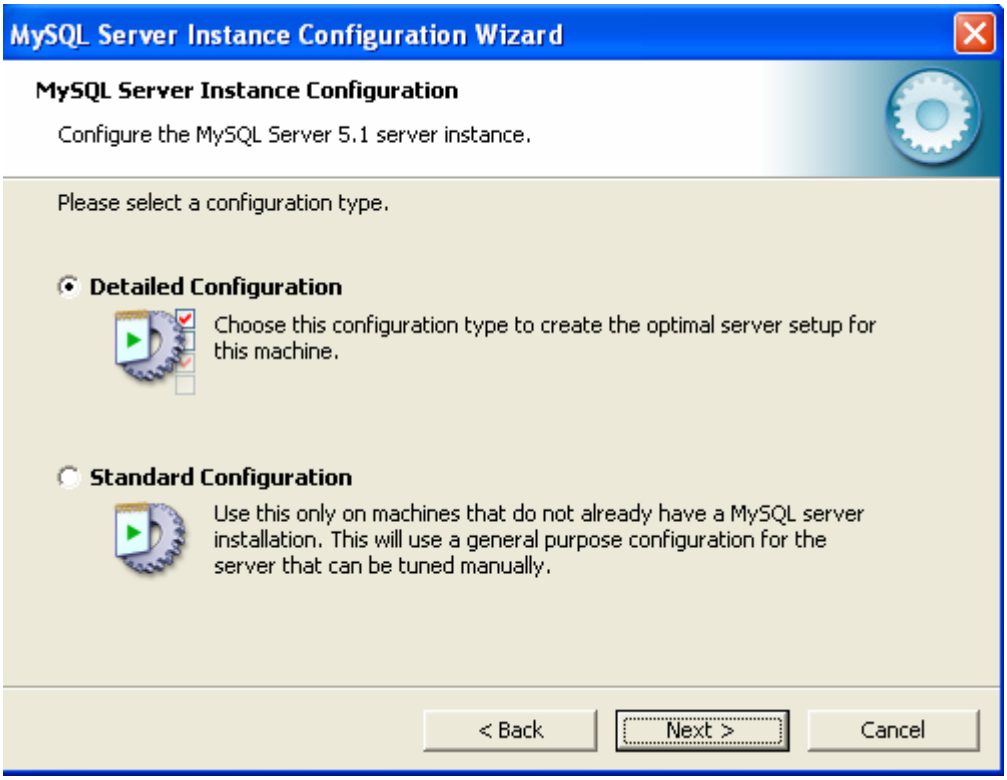


Figura 42. *Configuración detallada, o configuración estándar*

Figura 43. *Selección del tipo de servidor, influye la memoria de la que se dispone y el uso que se le de*

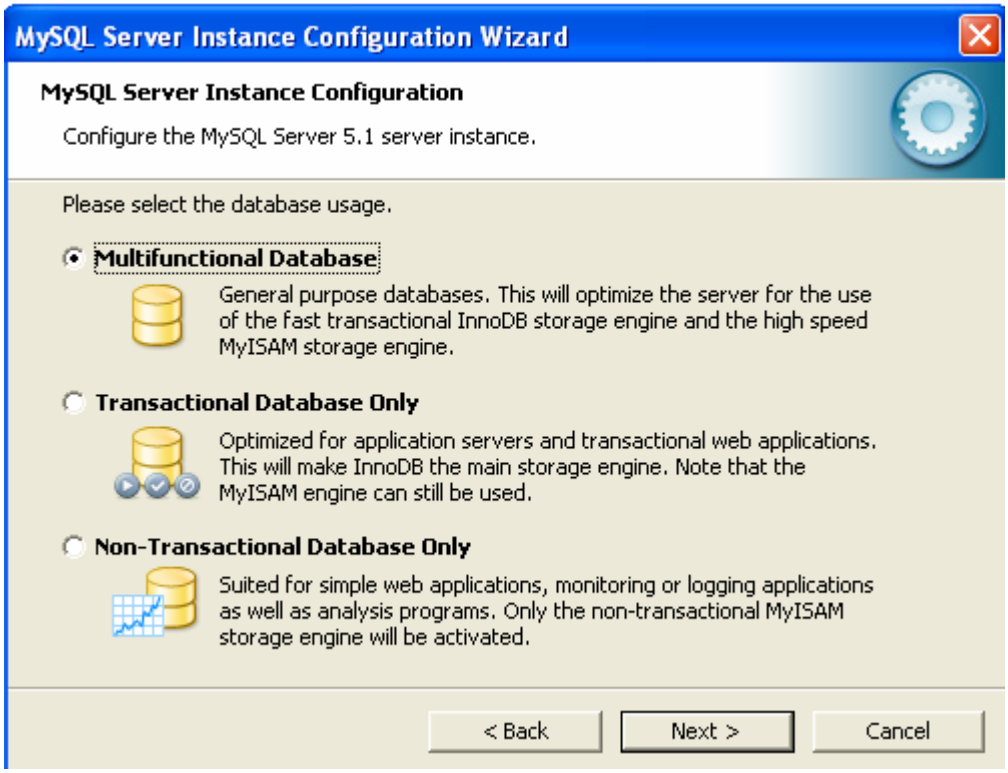


Figura 44. *Selección del tipo de base de datos en función del uso*

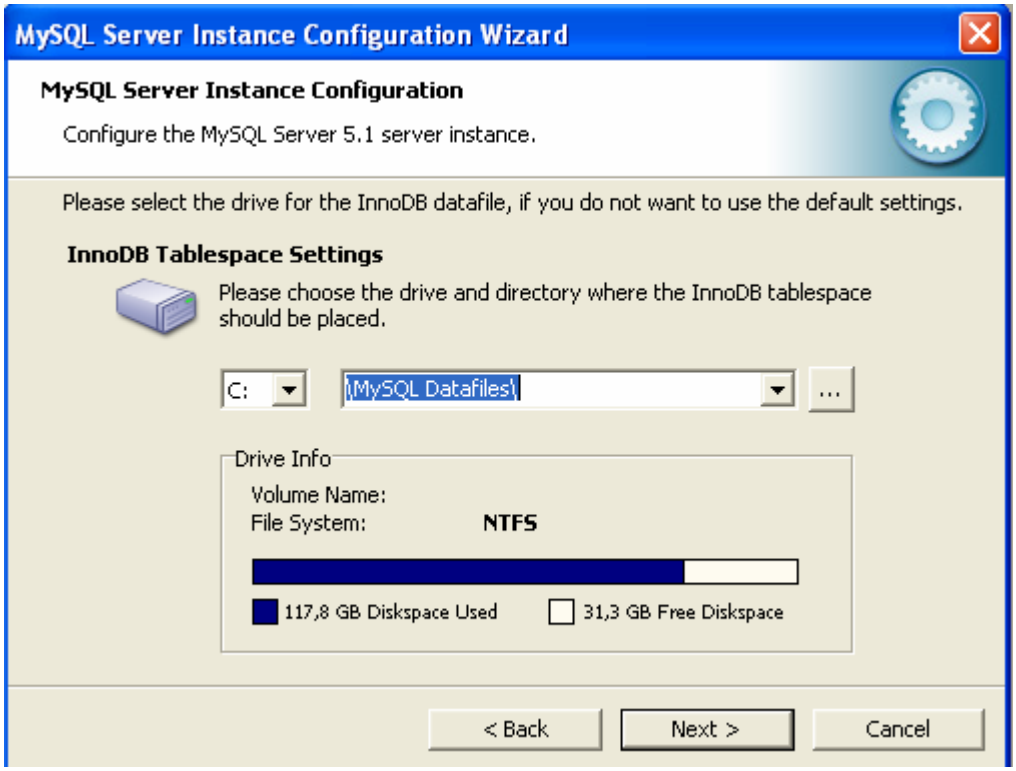


Figura 45. *Selección de la unidad de disco y el directorio donde se creará la unidad lógica de almacenamiento "InnoDB"*

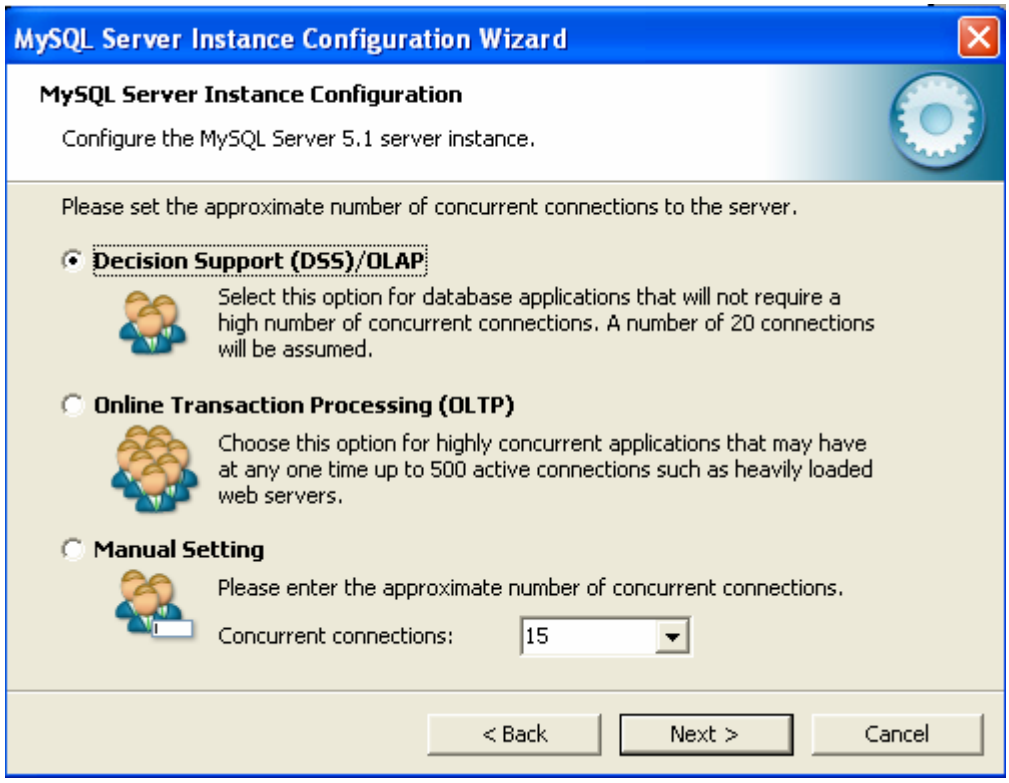


Figura 46. *Selección dependiendo del tipo de conexiones concurrentes al servidor*

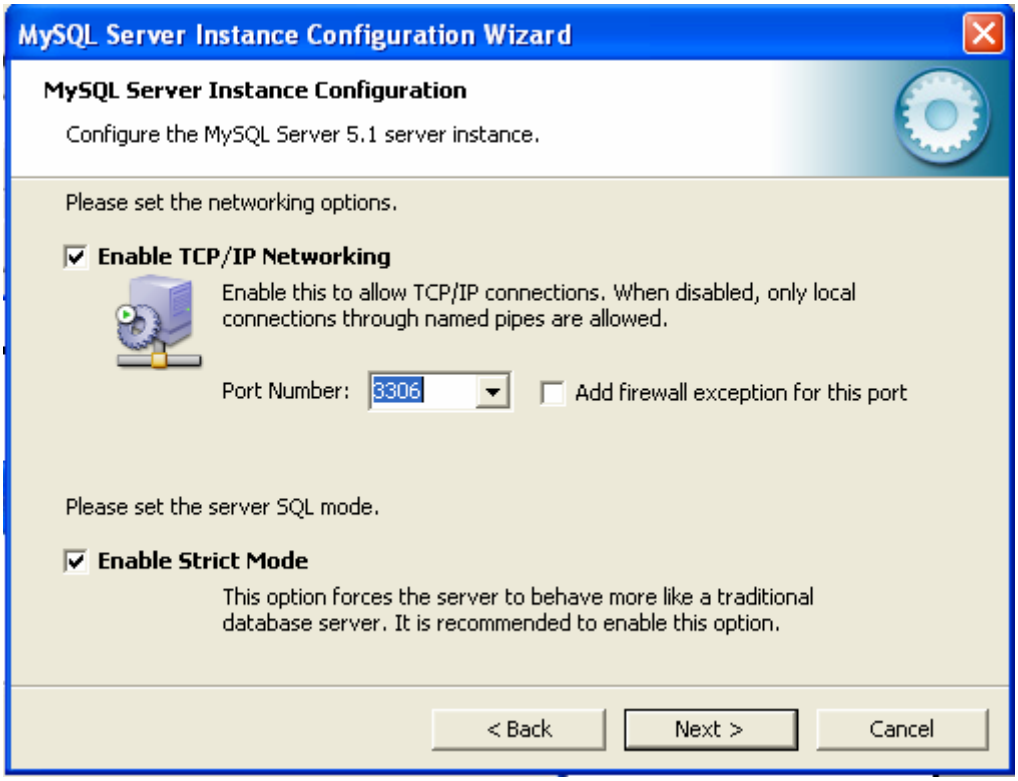


Figura 47. *Puerto TCP/IP de conexiones. Seleccionado el puerto por defecto.*

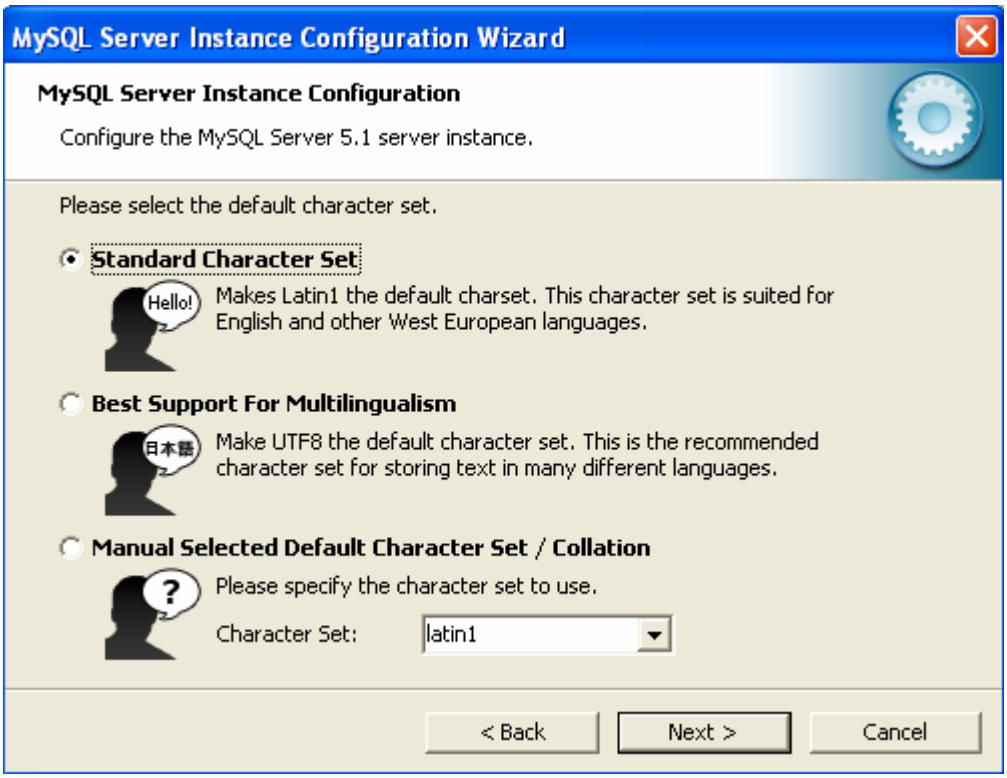


Figura 48. *Selección del conjunto de caracteres. Por defecto seleccionado el estándar*

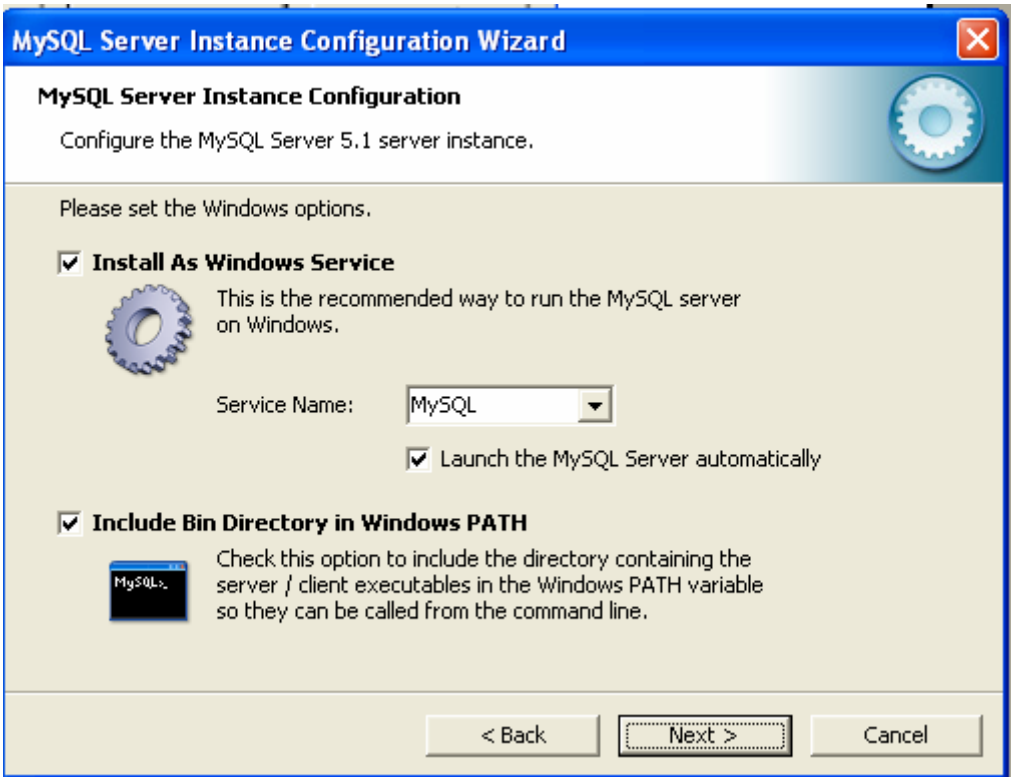


Figura 49. *Instalación de servicios necesarios para la correcta ejecución del programa en Windows*

Figura 50. *Introducción de la contraseña de seguridad del administrador de la BD*

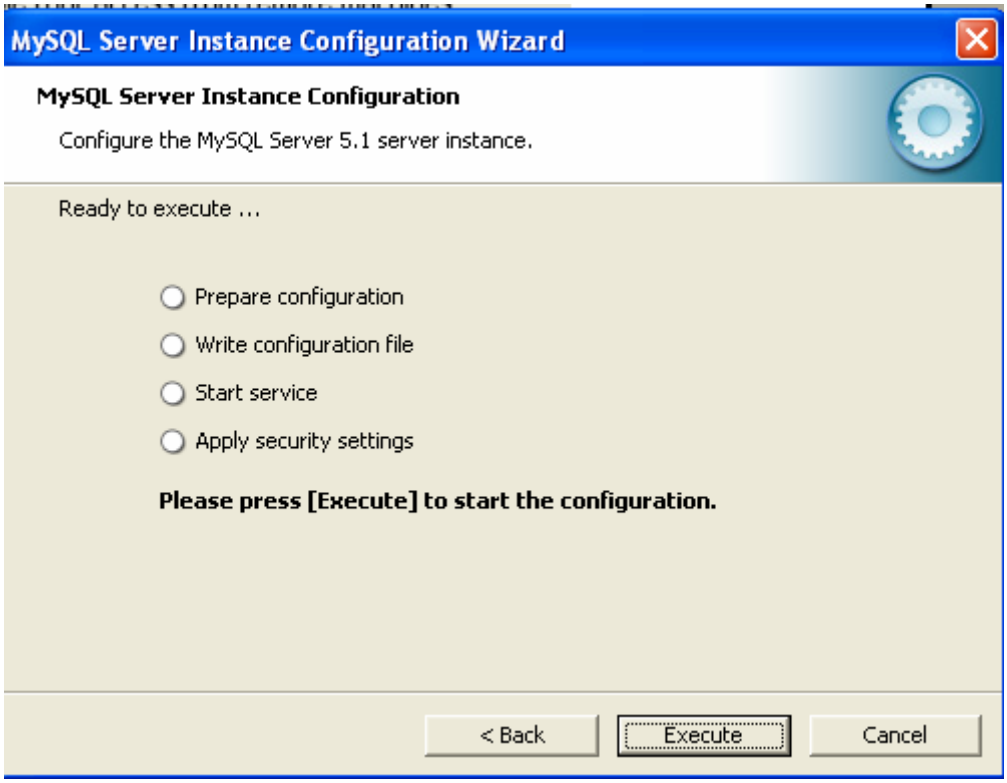


Figura 51. *Pantalla de finalización de configuración*

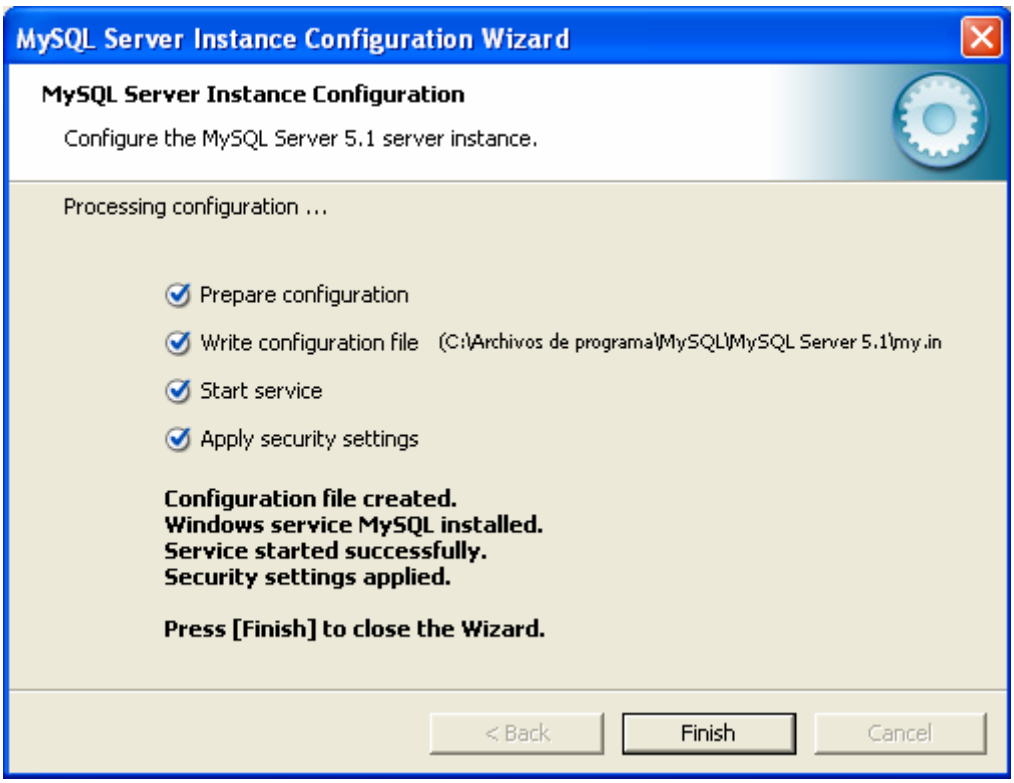


Figura 52. *Finalmente, la configuración ha sido terminada con éxito y el programa puede comenzar a ejecutarse*

Para comprobar que la instalación de MySQL se ha hecho correctamente bastará con abrir una consola y teclear “mysql -u root -p”, introduciendo la contraseña establecida anteriormente. El sistema mostrará una pantalla parecida a esta:

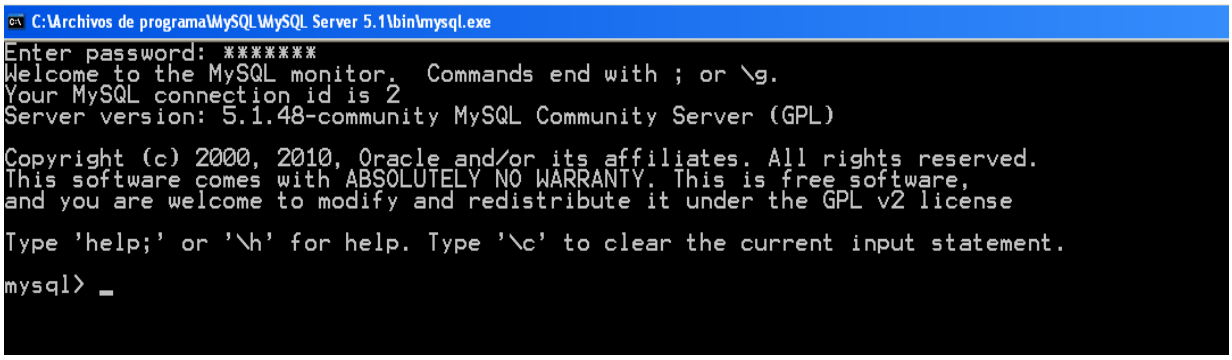


Figura 53. *Comprobación funcionamiento MySQL*

Hay que añadir para finalizar este apartado, que a finales de diciembre de 2010, Oracle liberó la nueva versión estable de MySQL: la versión MySQL 5.5.8.

Entre algunas de las características que resaltan en esta nueva versión (bastante extensa) destacan las siguientes:

- Escalabilidad mejorada en microprocesadores multinúcleo, aprovechando así los ciclos de procesamiento, eliminando en la medida de lo posible los cuellos de botella.

- Uso más efectivo de las capacidades del subsistema de Entrada / Salida (I/O) de InnoDB.

- Operación mejorada del servidor de MySQL en Solaris.

- Mejor acceso a la información, gracias a las nuevas características de diagnóstico y monitoreo.

- El nuevo motor de almacenamiento por defecto deja de ser MyISAM, que pasa a ser InnoDB.

- Soporte para las sentencias SQL estándares: SIGNAL y RESIGNAL.

- Mejoras en la funcionalidad XML, incluyendo una nueva sentencia: LOAD XML.

- En esta versión es posible la eliminación de todos los registros de una o más particiones de una tabla usando la sentencia ALTER TABLE.

- Soporte para autenticación al servidor MySQL por medio de extensiones y por usuarios proxy. No será necesaria la utilización de usuario/contraseña para acceder a una base de datos.

- Nueva función TO_SECONDS(), que convierte una expresión date o *datetime* a la cantidad de segundos transcurrida desde el año 0.

- MySQL 5.5 ahora es compilado usando CMake en vez del legendario GNU puestos informáticas, detalle importante a tomar en cuenta si se desea instalar MySQL 5.5 desde su código fuente.

5.11. LA HERRAMIENTA MYSQL WORKBENCH

A la hora de realizar la auditoría de una empresa que utilice un/os servidor/es MySQL, el auditor puede encontrarse con dificultades si no conoce a la perfección este SGBD. Por ello, éste debería utilizar todas aquellas herramientas que estén a su alcance para intentar facilitar su trabajo y hacerlo más sencillo, a pesar de no tener excesivos conocimientos en esa área concreta.

Por ello, el auditor en aras de la realización de su auditoría puede utilizar como herramienta MySQL Workbench, que tiene como características una interfaz agradable, intuitiva y fácilmente manejable. En primer lugar porque no es necesario conocer de antemano ciertos comandos que en principio se necesitan para su introducción en la consola. Por otro lado, la experiencia dice que, un programa basado en ventanas tipo Windows, en el que toda la información relevante está encapsulada y que muestra la información resumida y en lenguaje natural, será más manejable para el no iniciado, que no necesita conocimientos previos para "intuir" cual será el resultado de ejecutar una determinada función. Se podría calificar la herramienta como una "herramienta gráfica".

De hecho, las dos herramientas de trabajo MySQL Administrator y Query Browser que eran utilizadas en versiones anteriores a la 5.1 fueron sustuidas por esta herramienta.

Con respecto a su funcionalidad se puede dividir en tres áreas principales: SQL para el Desarrollo (*SQL Development*), Modelado de datos (*Data Modeling*)y Administración del Servidor (*Server Administration*):

1. SQL para el desarrollo: Permite crear y administrar las conexiones a los servidores de base de datos, así como configurar los parámetros de conexión, MySQL Workbench proporciona la capacidad para ejecutar consultas SQL en base a un editor de SQL (que viene a sustituir a *Query Browser*). Editar tablas, *scripts* SQL, etc.

2. Modelado de datos: Permite la creación de modelos gráficos de esquema de base de datos, ingeniería directa e inversa, sincronizar y comparar esquemas, informes.

3. Administración del Servidor – Permite crear y administrar instancias de servidor, administrar la seguridad, gestionar las importaciones y exportaciones. Esta funcionalidad sustituye a la anterior herramienta *MySQL Administrator*.

MySQL Workbench está disponible en la página web de MySQL, en el apartado de descargas, en dos ediciones. La *Community Edition* y la edición estándar. La *Community Edition* está disponible de forma gratuita, mientras que la edición estándar ofrece funciones adicionales de empresa a coste bajo (según la página Web).

A modo de ejemplo se procederá a explicar como se hace la instalación he la herramienta en un sistema operativo Windows que tiene instalado MySQl Server 5.5, y que desea utilizar MySQL Workbench 5.2.31.

En primer lugar, se ha de tener en cuenta los requerimientos de software para llevar a cabo la instalación (sistema operativo, memoria, etc.) así como de hardware.

La descarga se realiza desde la sección de descargas de la página Web oficial de MySQL. Se selecciona el sistema operativo en el que se va a ejecutar la herramienta y la versión deseada. A continuación y una vez que está descargada la aplicación es comienza la instalación y configuración de la misma.

Se procederá pues a mostrar los mensajes que aparecerían al realizar la instalación:

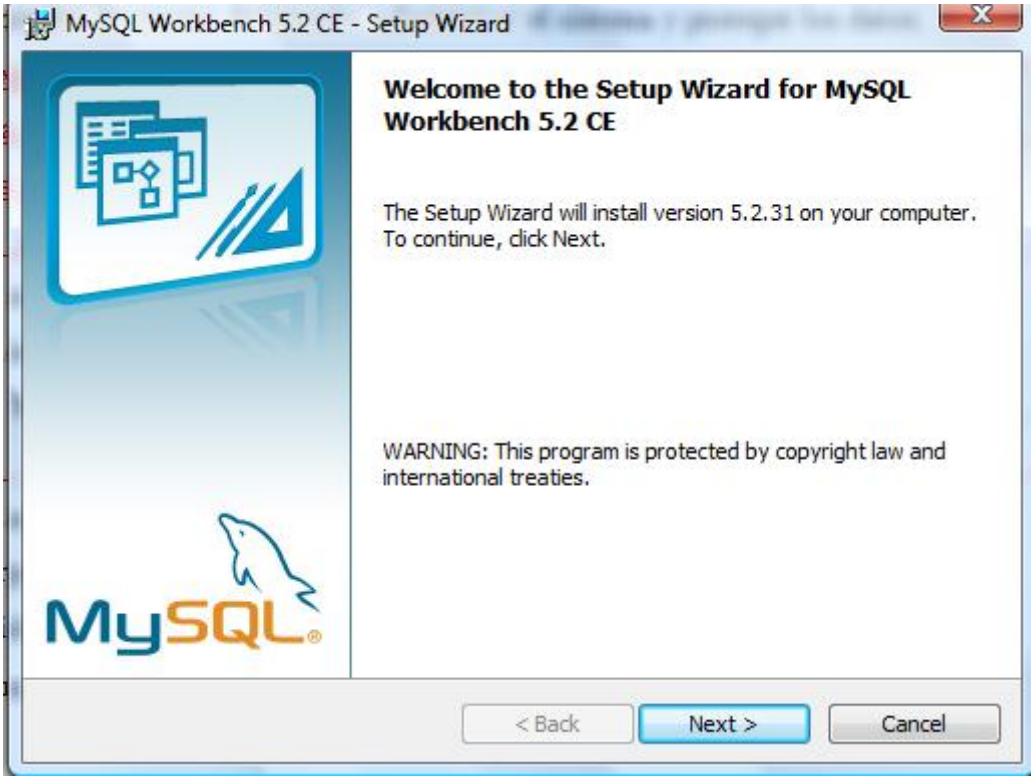


Figura 54. *Pantalla de bienvenida en la configuración de la aplicación*

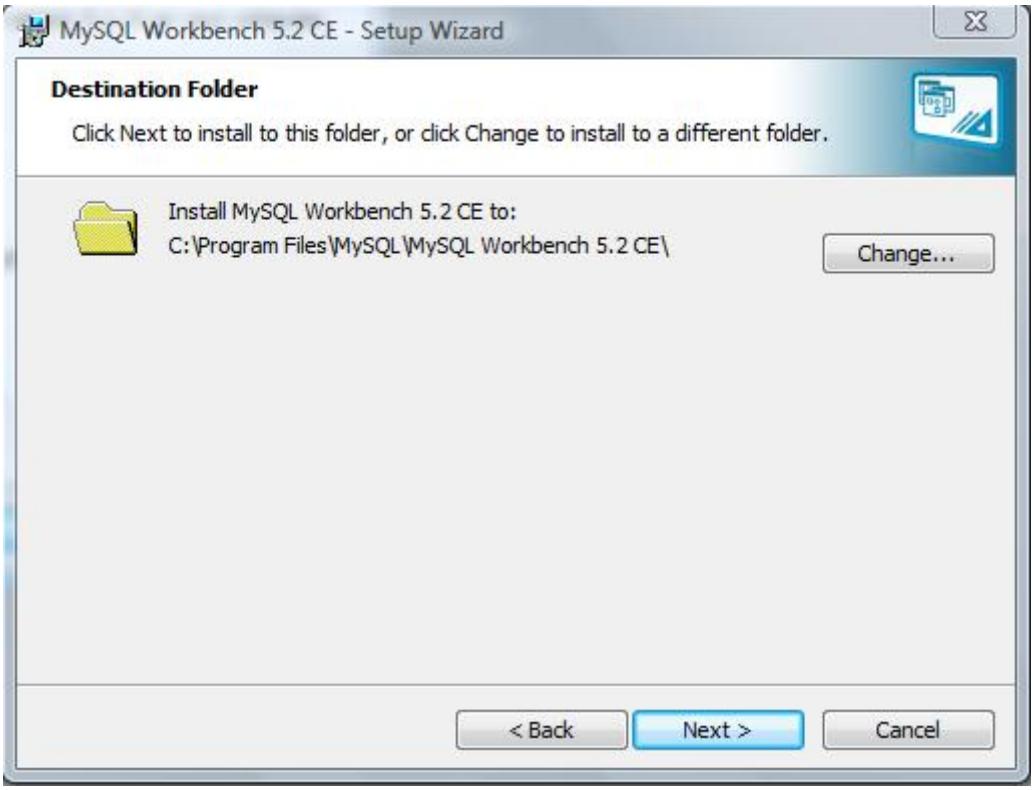


Figura 55. *Se indica en la carpeta y directorio en el que se almacenará la herramienta*

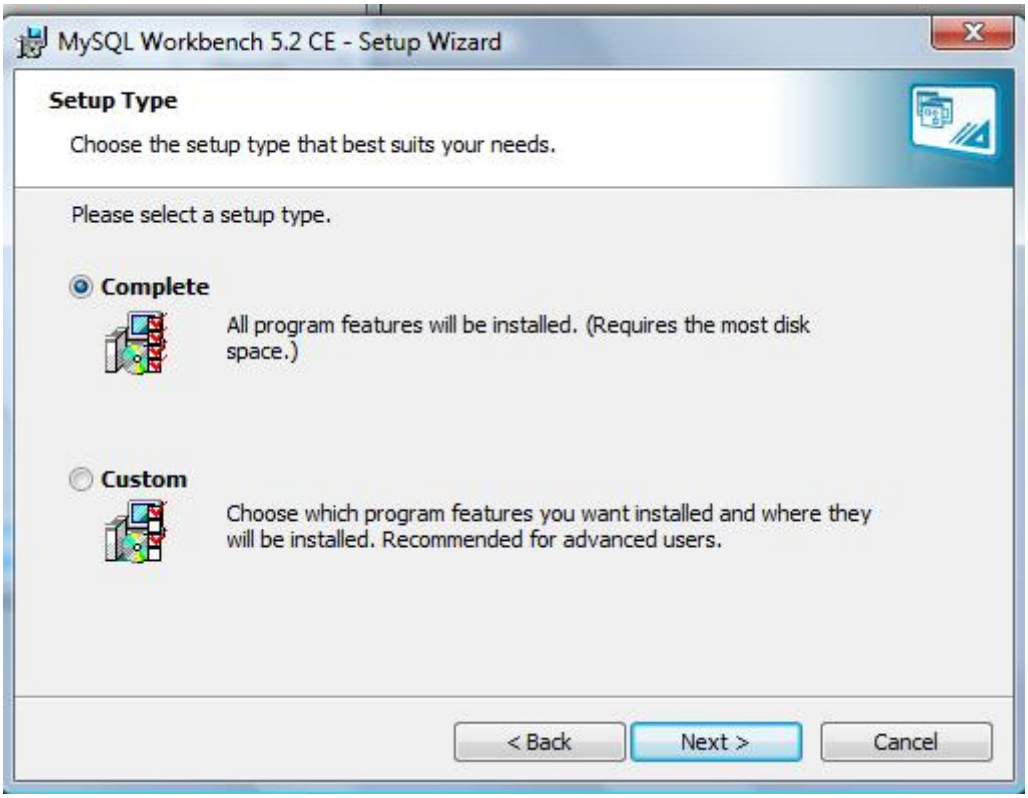


Figura 56. *Se selecciona el tipo de instalación, por defecto aparece la completa*

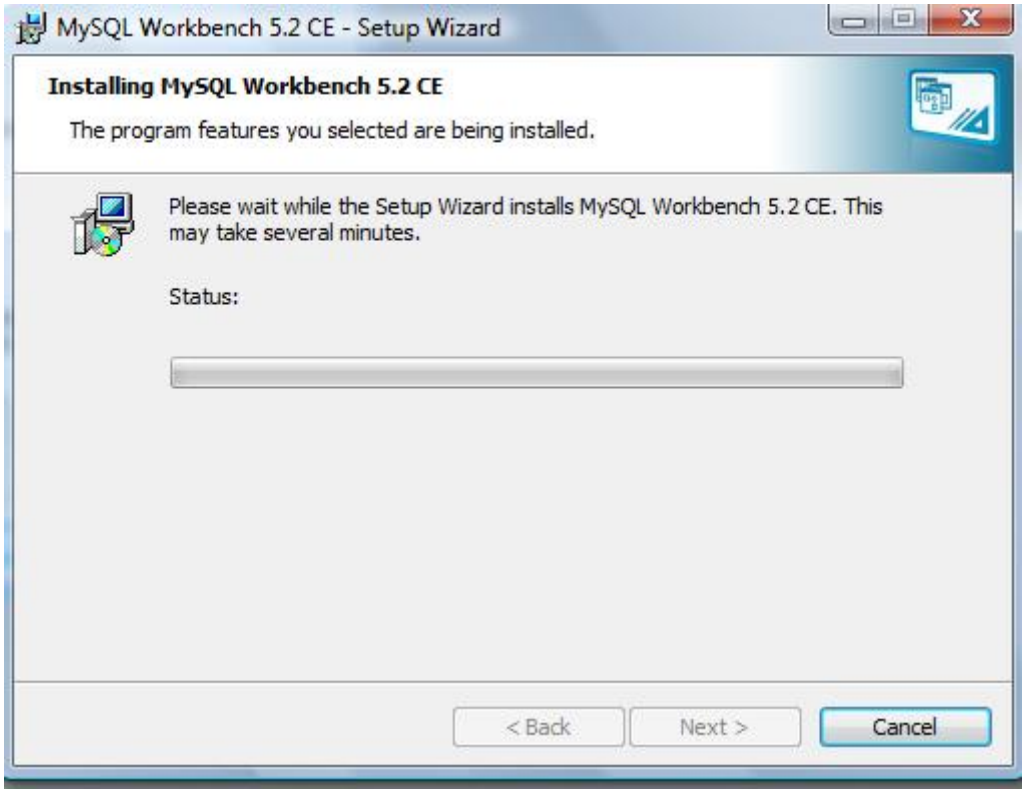


Figura 57. *Una vez seleccionado el tipo, hay que esperar unos minutos de carga*

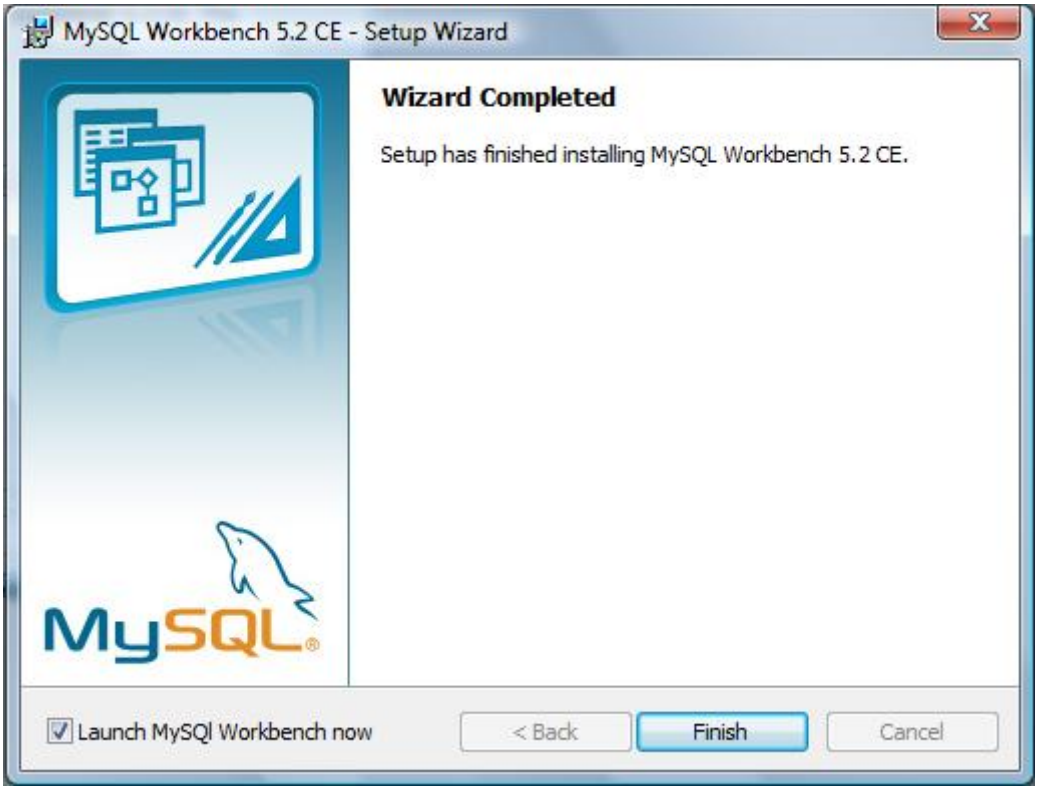


Figura 58. *Mensaje que indica que la instalación se ha completado con éxito*

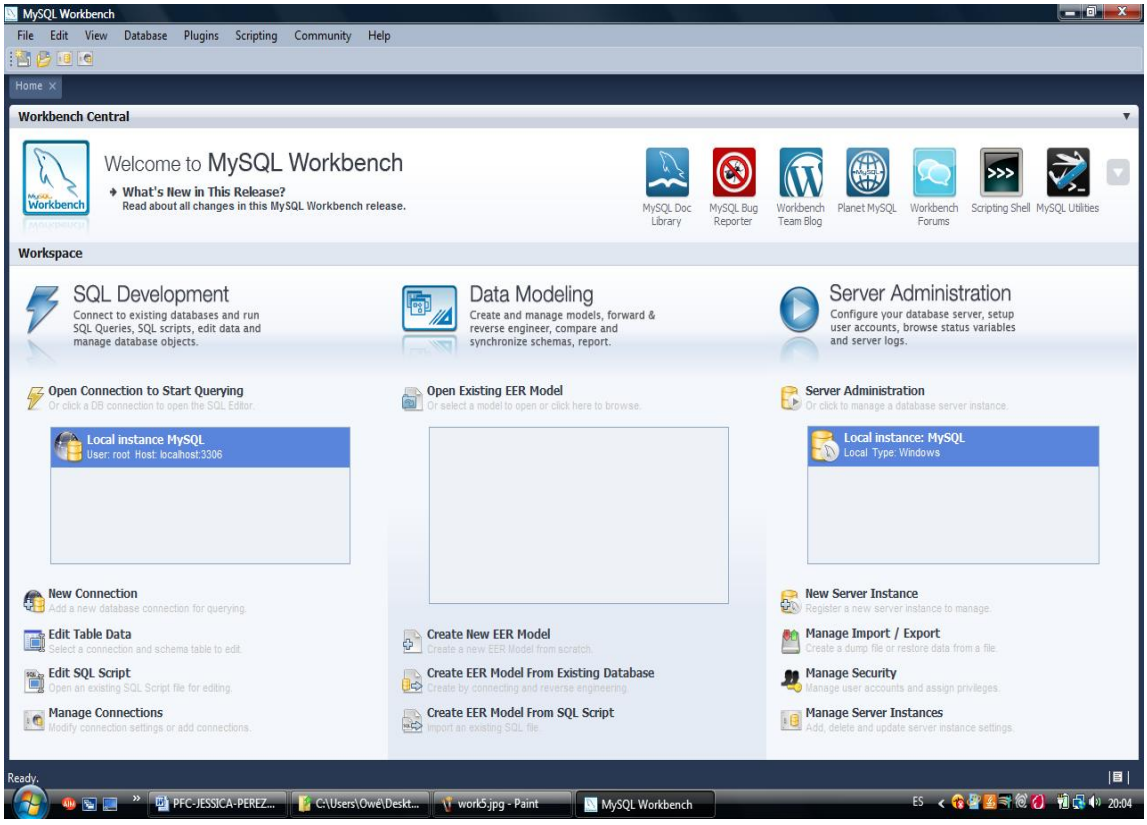


Figura 59. *Pantalla de comienzo de la herramienta*

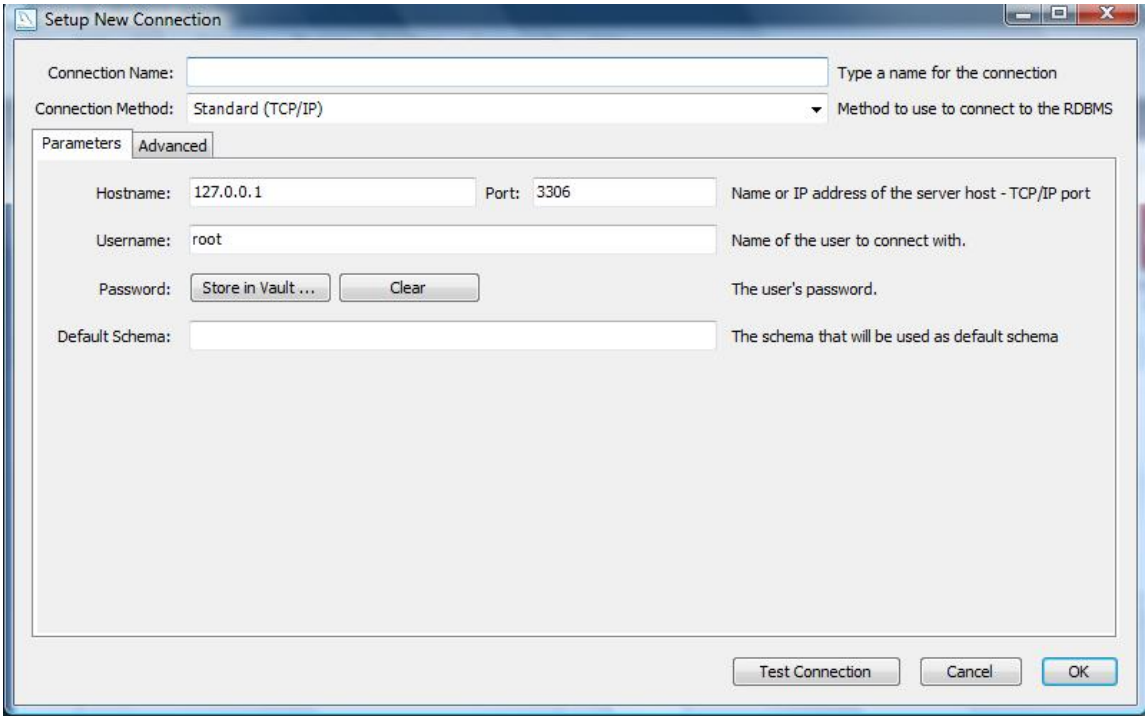


Figura 60. *Pantalla de configuración de nueva conexión*

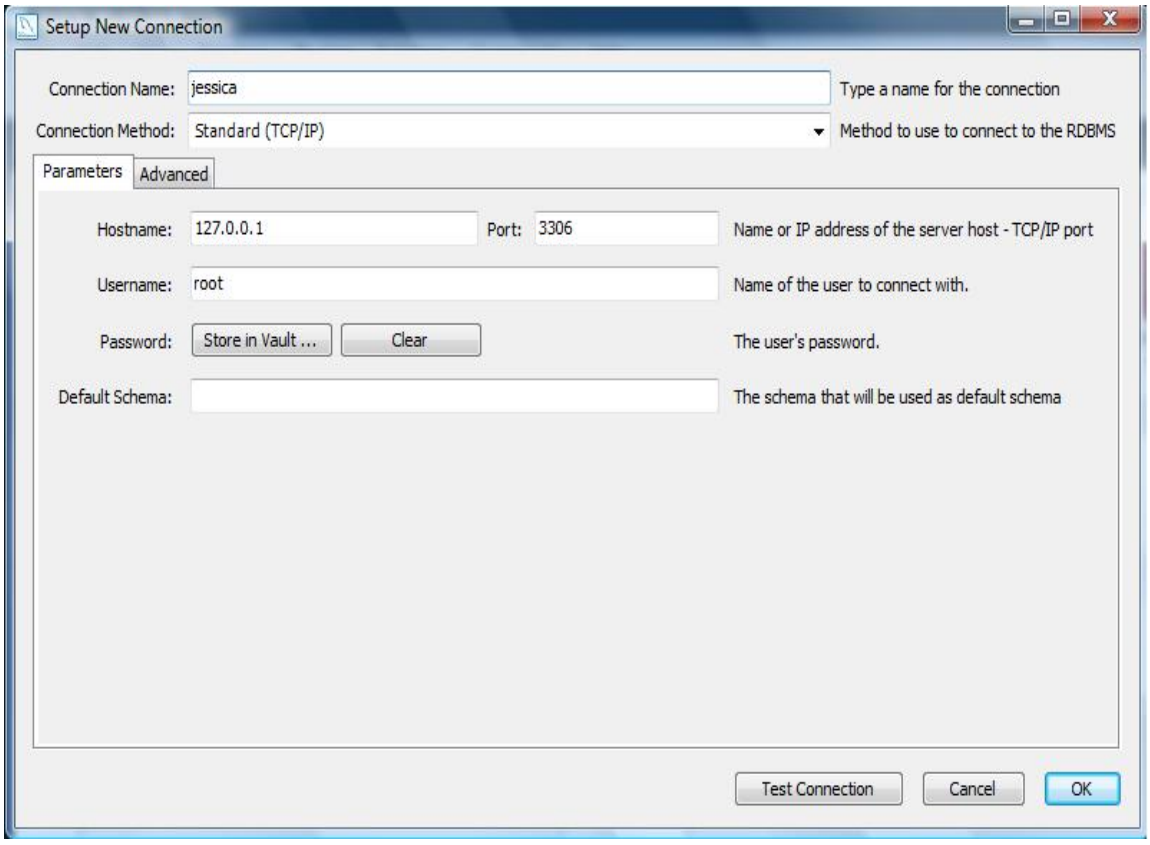


Figura 61. *Pantalla de introducción del nombre de la conexión*

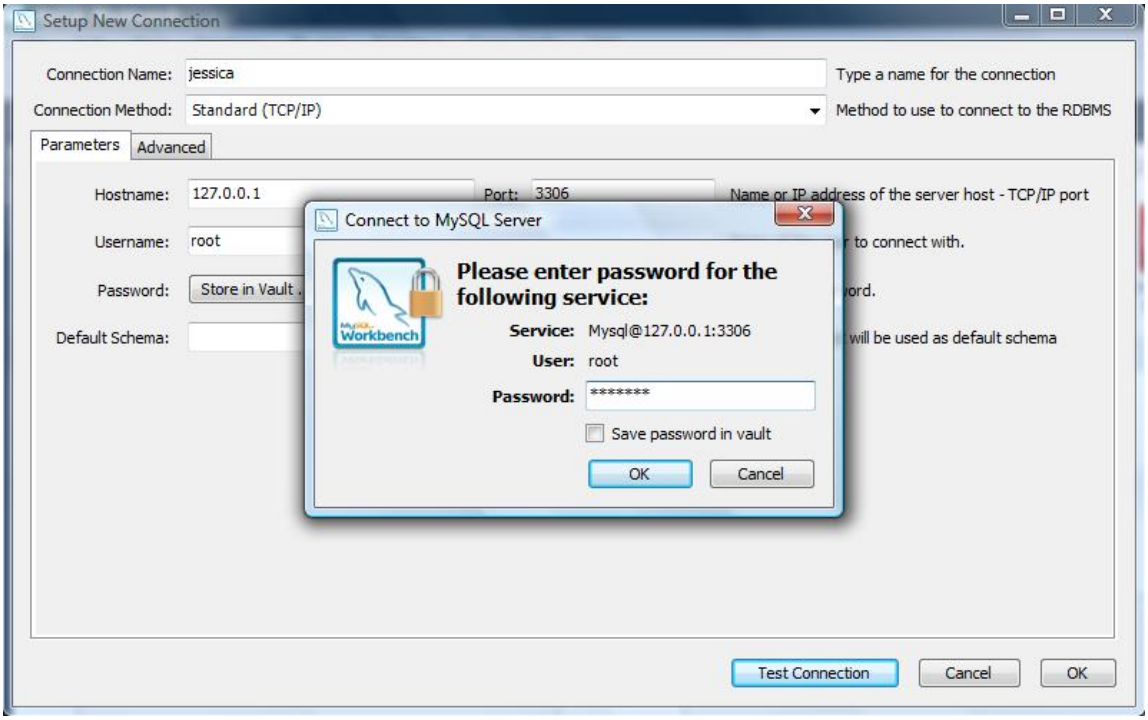


Figura 62. *Pantalla de introducción de la contraseña*

Entre las funcionalidades que más destacan en la ayuda en la auditoría de la seguridad se encuentran en el área de administración del servidor, puesto que aquí es donde es engloban la mayoría de las funciones que se deben controlar.

En la versión de MySQL Workbench 5.2.31. se observa en la pantalla de comienzo (figura 59) las tres áreas que se han definido anteriormente. El área de administración del servidor (*Server Administration)* está dividida en diferentes funcionalidades: nueva instancia de servidor, gestión de la importación/exportación (crea archivos de volcado o restaura datos de un archivo), gestión de la seguridad (seguridad en cuentas y asignación de privilegios), y gestión de las instancias del servidor (añadir, eliminar, modificar la configuración de las instancias del servidor).

La manera en la que está configurado el servidor, los usuarios, gestión de privilegios, etc. puede servir de guía y ayuda al auditor, especialmente, si esta información se muestra de manera tan intuitiva y visual.

Conclusión

A lo largo del presente proyecto se ha intentado resaltar la importancia que tiene la información para cualquier organización; por ello, la auditoría sobre el control y seguridad de los datos que almacenan los sistemas, pasa a ser fundamental para cualquier empresa.

Uno de las “soportes” de almacenamiento de información más extendido en la actualidad, es la base de datos. Por ello será tarea del auditor, controlar y evaluar la seguridad de todo aquello pueda afectar directa o indirectamente a la seguridad de los datos almacenados.

Como se ha explicado a lo largo de todo el proyecto, es sumamente importante no descuidar ningún aspecto: tan importante es la seguridad lógica, como la seguridad física, y por ello, el auditor debe también controlar estas áreas.

Como consecuencia directa de este control, probablemente se observaría una mejora de la organización, de lo que se beneficiará todo su entorno, tanto personal como clientes de la misma.

La dificultad que radica en realizar un proyecto sobre un área que está en continuo avance y transformación (en este caso concreto bases de datos, y particularmente MySQL), es que desde el mismo momento en que se plantea su estudio, de alguna manera ya está obsoleto.

Como particularidad hay que añadir que en los últimos tiempos MySQL ha sufrido numerosos cambios (puede ser también fruto de su adquisición por parte de Oracle), avances y mejoras en el desarrollo de su aplicación. Por ello, hay que entender este PFC como una pequeña guía o introducción a las áreas generales que habría que tener en cuenta a la hora de hacer una auditoría en una empresa u organización que utilizara una base de datos, en el caso concreto que nos ocupa MySQL, pero haciendo

hincapié, en que el auditor o auditores han de tener en consideración que las bases de datos van modificando sus especificaciones en función de las necesidades de los usuarios y por ello, deben estudiar con detalle qué versión del producto es la que se está utilizando, y cuales son sus particularidades propias para luego extrapolarlo al desarrollo de su auditoría. Obviamente, no será lo mismo analizar una aplicación que utilice MySQL versión 4.0 que una que utilice la versión 5.5.

Por ello, como línea futura de desarrollo de este proyecto podría ser el estudio detallado de la seguridad que se utiliza en alguna versión concreta de MySQL, así como la elaboración de un prototipo de ayuda a la auditoría en la que puedan apoyarse los auditores. Asimismo, existen en el mercado numerosas aplicaciones de ayuda a la auditoría, que lo hacen no sólo de esta base de datos concreta, sino de otras muchas (como Oracle, Microsoft SQL Server, SyBase SQL Server, etc, en sus numerosas versiones) por lo que su utilización por parte del auditor puede ser bastante útil.

6.1 Presupuesto

6.1.1. Estimación de la realización del Proyecto

Para la estimación presupuestaria de lo que ha supuesto la elaboración del presente trabajo únicamente se tendrán en cuenta dos factores primordiales: el tiempo y el coste material.

Con respecto al tiempo, es muy difícil cuantificar el tiempo invertido en su elaboración, ya que por diversas causas se ha ido retrasando, por lo que la manera de recoger ese tiempo es casi imposible. Las sucesivas modificaciones, ya sea por defecto en la forma o contenido del trabajo, en las periódicas actualizaciones y cambios sufridas a lo largo de este tiempo del Sistema Gestor de Bases de Datos MySQL, hace que su control sea aún más complicado.

A modo informativo, únicamente se dirá que la elaboración del trabajo comenzó cuando el desarrollo de MySQL estaba aún bajo la empresa MySQL LAB (antes de enero de 2008).

Los costes materiales asumidos son los que supone la impresión sucesiva y eventual del trabajo en alguna copistería, así como el encuadernado del mismo para una mejor visualización. El gasto estimado ha sido de unos 50 euros por impresión, encuadernación y material de oficina.

Tanto el ordenador, como la licencia del software del sistema operativo, así como el sistema antivirus y la conexión a Internet ya estaban asumidos antes de la elaboración del proyecto, por tanto, esos gastos se consideran despreciables y sin relevancia en el avalúo.

El software de MySQL Server es gratuito al descargarse la versión disponible en la página Web (http://www.mysql.com/), por lo que su instalación tampoco ha supuesto ningún gasto.

La consulta de las páginas Web accedidas así como las revistas electrónicas se ha realizado en páginas de consultas o de descarga libre, así como en servicios de consulta para alumnos de la UC3M. Así como material que otras universidades o empresas han difundido de manera gratuita.

Los libros han sido consultados en la biblioteca de la universidad.

Si seguimos el modelo de presupuesto que está disponible para ser descargado en la página Web de la Universidad Carlos III de Madrid (http://www.uc3m.es), habría que completar un modelo análogo al siguiente:

UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor: Jessica Pérez Sandoval

2.- Departamento: Ingeniería Informática

3.- Descripción del Proyecto: Las bases de datos, su seguridad y auditoría. El caso de MySQL
- Titulo
- Duración (meses) **9**
Tasa de costes Indirectos: **20%**

4.- Presupuesto total del Proyecto (valores en Euros):
Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a titulo informativo)	Categoría	Dedicación (hombres mes) [a]	Coste hombre mes	Coste (Euro)	Firma de conformidad
					0,00	
		Ingeniero Senior		4.289,54	0,00	
Jessica Pérez Sandoval		Ingeniero Técnico	1	2.694,39	2.694,39	
					0,00	
					0,00	
		Hombres mes 1		**Total**	2.694,39	

[a] 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable [d]
Portátil	799,00	100	9	60	119,85
		100		60	0,00
		100		60	0,00
		100		60	0,00
		100		60	0,00
					0,00
				Total	119,85

[d] Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \text{xCxD}$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación (60 meses)
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100%)

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
	Total	0,00

OTROS COSTES DIRECTOS DEL PROYECTO[e]

Descripción	Empresa	Costes imputable
Fotocopias y material oficina		50,00
	Total	50,00

[e] Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,…

6.- Resumen de costes

Presupuesto Costes Totales	**Presupuesto Costes Totales**
Personal	2.694
Amortización	120
Subcontratación de tareas	0
Costes de funcionamiento	50
Costes Indirectos	573
Total	3.437

6.1.2. Estimación de la Auditoría

Por otro lado, si se quisiera realizar la estimación presupuestaria de lo que supone el llevar a cabo la auditoría de la seguridad de una organización o empresa que tenga implantado como SGBD MySQL hay que tener en cuenta varios factores primordiales:

1. El tamaño de la organización a auditar. MySQL puede implantarse en empresas de gran tamaño debido a que es fácilmente portable, y que por ende, puede haber multitud de puestos "cliente" o varios servidores. Por tanto, es esencial el conocimiento de este dato. ¿Cuántos servidores hay? ¿Cuántos puestos interconectados existen?

2. La localización de la empresa. No es lo mismo la auditoría de una empresa que únicamente tiene una sede en una determinada localización, a otra empresa que tiene otras en diferentes localizaciones y diferentes ambientes. La seguridad física dependerá también de este factor.

3. Número de empleados. Como se ha explicado en el primer punto, la auditoría no será la misma –ni en esfuerzo ni en tiempo necesario- en una empresa con 10 empleados que en una empresa con 500 empleados.

4. Si es auditoría interna o externa. Si se requieren soluciones.

Para el desarrollo de este presupuesto, se estima una empresa mediana de entre 20 a 30 empleados, una única localización, un solo servidor y 15 puestos clientes repartidos en dos salas de "procesos".

Se estima que para una correcta auditoría correcta se necesita una media de entre 4-7 días de auditoría. Consta de dos apartados: gastos de ejecución material y mano de obra.

6.1.2.1. Ejecución material

En este apartado se incluyen los gastos en herramienta utilizados.

Concepto	Precio (Euros)
Ordenador Windows XP	450
Impresora Láser	120
Licencia Office 2003	75
Material de oficina	150
Encuadernación informes	70
TOTAL	865

Figura 63. *Gastos ejecución material*

6.1.2.2. Mano de obra

En este apartado se incluyen los gastos que suponen los recursos humanos que intervienen en la gestión y desarrollo de la auditoría y en la elaboración del informe.

Concepto	Salario semanal (bruto)	Semanas	Total (euros)
Ing. Informático (auditoría)	500	1	500
Ing. Téc Informático (desarrollo)	425	1	425
Mecanógrafo	275	0 (2 días de 5 laborables)	110
TOTAL			1035

Figura 64. *Gastos de mano de obra*

A los gastos anteriormente declarados hay que añadir el mantenimiento de la aplicación a la ayuda de la auditoría que presumiblemente tendrá/n el/los auditor/es que llevan a cabo la auditoría de determinada empresa, así como otros gastos no definidos o gastos indirectos (transporte, dietas, otros conceptos) aparte de los recursos citados anteriormente. Por ello, el coste total de mano de obra de 1035 euros, se pondera sobre 1,5 para dar cobertura estos imprevistos, alcanzando una cuantía de 1552,50 euros.

El gasto total estimado resulta de: 2417,50 euros.

Este presupuesto es totalmente subjetivo y sólo puede ser visto como una mera aproximación a un caso supuesto con unas características muy determinadas. Dicho presupuesto variará en función de todos aquellos factores y variables relevantes que se han de tener en cuenta para su cálculo: empleados, servidores, puestos informáticos, entorno, etc.

Anexo 1: Batería de preguntas propuestas

La labor del auditor es muy compleja, puesto que bajo su responsabilidad está la de la auditoría de sistemas, así como el manejo de información muy importante para la organización para la que trabaje, y un error o fallo, aun ya sea de forma no intencionada, en ese manejo de información podría provocar graves pérdidas a la empresa.

Entre las funciones que puede desarrollar un auditor destacan:

- Informar acerca de la integridad de la información y de los datos.
- Análisis de eficacia, efectividad y continuidad de los sistemas.
- Análisis de los elementos lógicos.
- Análisis de los equipos físicos y situación de los edificios de la organización.
- Seguridad de los activos que forman parte de la organización.
- Revisión de los controles internos.
- Comprobar que se siguen las políticas de seguridad.

Para ayudarse en su labor, el auditor deberá utilizar una serie de recursos que facilitarán en gran medida su trabajo. Entre éstos recursos se encuentran las listas de comprobación.

A partir de la realización de cuestionarios y sus respuestas, el auditor, puede conocer más profundamente el área que esté auditando en un determinado momento. A modo de ejemplo se propone una serie de batería de preguntas o listas de comprobación muy breve que el auditor de cualquier base de datos así como de cualquier sistema debería realizar:

ASPECTOS LÓGICOS DE LA BASE DE DATOS:

	S	N	N/A
¿La base de datos permite la independencia de los datos y del tratamiento de los mismos?			
¿La redundancia de la base de datos está controlada?			
¿Los datos están disponibles para los usuarios autorizados?			
¿La base de datos posee valor informativo?			
¿La base de datos está convenientemente documentada?			
¿Los datos son validados antes de ser introducidos en la base de datos?			
¿Existen controles periódicos a realizar para verificar el cumplimiento del documento?			
¿En la práctica las personas que tienen atribuciones y privilegios dentro del sistema para conceder derechos de acceso son las autorizadas e incluidas en el Documento de Seguridad?			

DESARROLLO DE LA BASE DE DATOS:

	S	N	N/A
¿Se ha documentado apropiadamente todas las etapas del desarrollo de la base de datos de forma no ambigua, inconsistente o incompleta?			
¿Se ha utilizado para el desarrollo algún ciclo de vida?			
¿Se han utilizado estándares o métodos que guíen las distintas etapas del desarrollo?			
¿Se ha realizado una batería de pruebas completas antes de la implantación de la base de datos?			
¿Se han documentado los problemas encontrados tras la implantación de la base de datos?			
¿El sistema de ayuda a los usuarios es completo y entendible por ellos?			

ADMINISTRADOR DE LA BASE DE DATOS:

	S	N	N/A
¿El administrador realiza una gestión adecuada del diccionario de datos?			
¿El administrador interactúa y mantiene comunicación fluida con usuarios, directivos, analistas, programadores, suministradores y personal de administración?			
¿El administrador tiene formación adecuada para el desarrollo de sus funciones?			
¿Tiene experiencia?			
¿Ha realizado una descripción conceptual y lógica de la base de datos?			
¿Se ha involucrado en la formación de los usuarios?			
¿El administrador tiene los conocimientos necesarios sobre el sistema gestor de base de datos MySQL para realizar una buena descripción física de la base de datos?			
¿Ha establecido estándares para asegurar que la base de datos mantiene la seguridad e integridad de los datos?			
¿Se realizan reuniones periódicas con la alta dirección y se divulgan los resultados?			
¿Existen políticas y procedimientos de seguridad para la base de datos MySQL?			
¿Se realizan auditorías periódicas?			
¿El personal que utiliza la base de datos ha sido formado?			

SEGURIDAD Y PROTECCIÓN DE LOS DATOS

	S	N	N/A
¿Cuando se recoge información personal, el usuario es informado de la existencia de un fichero o almacén que contiene esa información y los derechos que tiene sobre ellos?			
¿La persona que tiene almacenados sus datos en un fichero o tabla de la base de datos puede consultarlos y acceder a ellos?			
¿Se puede garantizar la seguridad de la información personal?			

MEMORIA:

	S	N	N/A
¿Hay suficiente espacio físico para almacenar Completamente y correctamente los datos?			
¿El tamaño de la base de datos es la correcta para almacenar los datos e índices necesarios?			
¿En la estimación se ha tenido en cuenta el espacio necesario para la caché de datos?			

CALIDAD:

	S	N	N/A
¿La base de datos tiene la calidad necesaria?			
¿La dirección de la entidad acepta el compromiso de calidad?			
¿La organización tiene política de calidad? Si se tiene, ¿se cumple?			
¿La calidad involucra a todos los miembros de la organización?			
¿La calidad se comunica en la empresa?			
¿Se ha realizado alguna inspección, control de calidad, etc.?			
¿Si se ha realizado una auditoría de calidad previa ¿se han realizado las mejoras recomendadas?			
¿Se utiliza en la organización alguna normativa?			
¿Está actualizada?			

BASE DE DATOS:	S	N	N/A
¿La base de datos permite la independencia de los datos y del tratamiento de éstos?			
¿La redundancia de la base de datos está controlada?			
¿Los datos están disponibles para los usuarios autorizados?			
¿La base de datos posee valor informativo?			
¿La base de datos está suficientemente documentada?			
¿Los datos son validados antes de ser introducidos en la base de datos?			
¿Se utilizan técnicas de compactación de datos para disminuir el almacenamiento de datos?			
¿Existe procedimiento de incidencia para pérdida de datos en la base de datos?			
¿Los usuarios son autenticados correctamente a la hora de establecer conexión con la base de datos?			
¿Existe gestión de privilegios mediante GRANT y REVOKE para todos los usuarios de la base de datos?			
¿Quién tiene acceso a la gestión de esos privilegios?			
¿Esta gestión de privilegios ha sido asignada según los principios de “necesidad de uso” y “caso por caso” atendiendo a cada caso en particular o cada empleado y/o usuario concreto?			

¿Los usuarios han firmado un compromiso para mantener en secreto sus contraseñas personales?			
¿Se revisan a intervalos de tiempo regulares los derechos de acceso de los usuarios?			
¿Se revocan los privilegios asignados de manera temporal a aquellos usuarios que ya no lo necesiten?			
¿El acceso a la base de datos se realiza mediante una conexión segura?			
¿Se muestra un mensaje que advierte de la restricción a la base de datos sólo a usuarios autorizados para ello?			
¿Se valida la información de conexión únicamente tras rellenar todos los datos de entrada (sin indicarse qué partes son las que se han introducido correctamente) si se produce error?			
¿Se limita o controla de alguna manera el número de intentos fallidos de conexión?			
¿Existe un tiempo forzoso de espera antes de permitir un nuevo intento de conexión?			
¿Existe desconexión automática de la base de datos tras un tiempo prudencial de inactividad?			

DESARROLLO DE BASES DE DATOS:	S	N	N/A
¿Se ha documentado de manera adecuada todas las etapas de desarrollo de la base de datos de forma no ambigua, inconsistente o incompleta?			
¿Se ha utilizado para el desarrollo algún ciclo de vida?			
¿Se han utilizado métodos que guían las distintas etapas de desarrollo o estándares?			
¿Se satisfacen las necesidades de los usuarios por la base de datos?			
¿El usuario final fue involucrado en el proceso de desarrollo?			
¿Se han realizado suficientes pruebas con datos antes de la instalación de la base de datos para evitar problemas posteriores?			
¿En la implantación se encontraron problemas?			
¿Está bien documentada la base de datos? ¿Existe manuales de utilización accesibles a los usuarios de la base de datos?			
¿Los manuales o guías de ayuda a los usuarios son completos y entendibles?			
¿Se han estudiado y eliminado las incompatibilidades de la base de datos en el entorno en el que se ha implantado?			
¿Existe un mecanismo de comunicación simple y rápido entre el Administrador y el personal de mantenimiento de la base de datos?			

ADMINISTRACIÓN:	S	N	N/A
¿El administrador o administradores tienen una formación adecuada para el desarrollo de sus funciones?			
¿El administrador tiene experiencia previa?			
¿El administrador ha creado y documentado una estructura de la base de datos que cubra las necesidades reales de los usuarios?			
¿Se ha realizado una descripción conceptual y lógica de la base de datos expresiva, simple, mínima y formal?			
¿El administrador tiene los conocimientos necesarios sobre el sistema gestor de datos MySQL para realizar una buena descripción física de la base de datos?			
¿El administrador ha realizado las vistas necesarias que permitan a los usuarios operar con la base de datos?			
¿El administrador ha establecido estándares para asegurar que la base de datos mantiene la seguridad e integridad de los datos?			
¿Realiza la gestión de disponibilidad, integridad y confidencialidad?			
¿Se ha involucrado en formación de usuarios?			
¿El administrador interactúa y mantiene comunicación con usuarios, directivos, analistas, programadores, operadores, suministradores, y el personal administrativo?			
¿Se hacen reuniones periódicas con la alta dirección y se divulgan resultado?			
¿Existe política y procedimiento de seguridad para la base de datos MySQL?			
¿El personal que interactúa con la base de datos es entrenado mediante cursos para la correcta utilización de la base de datos?			
¿Existe un departamento de auditoría interna?			

LOPD Y SEGURIDAD:

	S	N	N/A
¿Los datos de la base de datos y su tratamiento son adecuados, pertinentes y no excesivos?			
¿Cuando se recoge información personal el afectado es informado de la existencia de un fichero que contiene esos datos y sus derechos sobre los mismos?			
¿El consentimiento es almacenado para que pueda ser comprobado posteriormente?			
¿Se puede garantizar la seguridad de los datos personales almacenados en la base de datos en función del nivel de seguridad?			
¿Los datos sólo se comunican si es necesario para fines empresariales?			
¿La persona que cede sus datos puede verlos, rectificarlos en caso de ser erróneos o haber cambiado. Puede eliminarlos si así lo desea?			
¿Se ha notificado a la Agencia de Protección de Datos la existencia de estos ficheros?			
¿Se realizan auditorías periódicas para verificar que se cumple con la LOPD y el Reglamento de Medidas de Seguridad?			

POLÍTICA DE SEGURIDAD:

	S	N	N/A
¿Existe un documento de política de seguridad de la información accesible para los empleados y puedan realizar consultas sobre ella?			
¿Esta política se revisa periódicamente para asegurarse que no ha variado nada y que se recoge todo lo necesario?			
¿Se han tenido en cuenta los posibles cambios tecnológicos o nuevos riesgos que han de estar controlados?			
¿Los empleados conocen las consecuencias de las violaciones de la política de seguridad?			
¿En la política se haya recogido la prevención y detección de virus, así como otro software malicioso?			
¿Los usuarios y empleados realmente respetan las políticas de seguridad que se recogen en los manuales?			

SEGURIDAD DE ACCESO

	S	N	N/A
¿Existe seguridad de acceso al recinto, ya sea mediante guardas de seguridad, tarjetas de control, seguridad biométrica, etc.?			
¿El acceso físico a despachos, equipos, etc., se encuentra realmente bien controlado?			
¿Hay alarma contra intrusos interconectada con la policía?			
¿Existe recinto cerrado (habitación, despacho, local) donde se encuentran situados todos los servidores en el que se pueda controlar su acceso (cerradura, guarda de seguridad, etc.).?			
¿Se han adoptado medidas de seguridad en el departamento de sistemas de información?			
¿Existe una persona responsable de la seguridad?			
¿La vigilancia se contrata? ¿Si lo hace es directamente? ¿Por medio de empresas que venden ese servicio?			
¿Se controla el trabajo fuera de horario?			

PROTECCIÓN DE SOFTWARE MALICIOSO

	S	N	N/A
¿Se ha controlado la descarga de software no autorizado?			
¿Existe antivirus instalado, en uso y actualizado en todos los puestos donde esté instalada la base de datos (servidor, clientes)?			
¿Se realizan revisiones periódicas del software instalado en los sistemas?			
¿Se investiga formalmente la presencia de ficheros no aprobados o de la modificación de ficheros no autorizados?			
¿Se realiza un "escaneo" previo de todo fichero adjunto en un correo electrónico a través de un programa antivirus antes de descargarlo?			
¿Se ha formado a los empleados en la correcta utilización de los programas antivirus y *firewall*?			
¿Existen boletines de alerta periódicos e informativos sobre nuevas técnicas en relación al software malicioso?			
¿Se tiene una lista de direcciones de Internet confiables? ¿Existen páginas de Internet en las que no se puede acceder por ser poco confiables?			
¿Se controla el acceso a los servicios en redes internas y externas?			

Asimismo se debería auditar la seguridad física, los objetos de la base de datos, la calidad de la base de datos, el control de accesos físicos y lógicos, la autenticación de usuarios, el sistema de detección intrusos, la seguridad en las comunicaciones (configuración de la red), el sistema de recuperación de errores, el sistema de prevención, etc.

Bibliografía

- BARRIOS, Rita: "*Auditoría y control de entornos Informix*".2007. PFC UC3M.
- BURBANO PROAÑO, Diego Javier. "*Análisis comparativo de bases de datos de código abierto y código cerrado*". Documento electrónico. 2006.
- CONTROL OBJECTIVES FOR INFORMATION TECHNOLOGY (COBIT). "*Audit Guidelines*" 3ª. Edición. 2000.
- CONTROL OBJECTIVES FOR INFORMATION TECHNOLOGY (COBIT). "*Control Objectives*" 3ª. Edición. 2000.
- CONTROL OBJECTIVES FOR INFORMATION TECHNOLOGY (COBIT). 4ª Edición.
- COY, Juan María. *Auditoría y Seguridad en Bases de Datos. Enfoque práctico con Oracle 8i.* Apuntes electrónicos.
- CUADRA, Dolores (VV.AA). *Diseño de Bases de Datos.* Documento Electrónico. ISBN: 84-369-3473-3
- DE VEGA, María José: **"***Función de la auditoría interna de Sistemas de Información" 1998. PFC UC3M.*
- DUBOIS, Paul. *Edición especial de MySQL*. Prentice Hall. 2001. ISBN: 84-205-2956-7.
- ECHEÑIQUE, José Antonio. *Auditoría en Informática*. 2ª Edición. Mc Graw Hill. 2001. ISBN: 97-010-3356-6.
- GARCIA, Daniel. "*Auditoría y control en entornos Oracle 9i*".2003. PFC UC3M, Leganés.
- GONZALEZ, Ana Belén. "*Auditoría de bases de datos*".1997. PFC UC3M, Leganés.
- HEREDERO, Manuel. *Legislación Informática*. Tecnos. 1994. ISBN: 84-309-2395-0.
- INTERNATIONAL STANDARDIZATION ORGANIZATION (ISO). "*Estándar de Seguridad ISO 27002*" Edición 2002.

- INTERNATIONAL STANDARDIZATION ORGANIZATION (ISO/IEC). "*ISO 27002*" Edición 2005.

- LOPEZ, Carlos: "*Programa para Gestión de Rutas de Operarios bajo Java y MySQL*". 2008. PFC UC3M.

- MEDINA-HEIGL, Román: "*Análisis de seguridad, optimización y mejora de un portal web basado en PHP y MySQL". 2002. PFC Universidad de Sevilla.*

- MELONI, Julie C. *Programación PHP, MySQL, APACHE.* Anaya Multimedia.

- MySQL AB. *Manual de referencia MySQL 5.0*. MySQL AB. 2006.

- PEREZ, César. *MySQL para Windows y Linux***.** 2ª Edición. Ra-Ma. 2007. ISBN: 84-789-7790-2.

- PRA, Pablo Ignacio. *Plan de seguridad informática.* Tesis Universidad de Córdoba. 2002.

- RAMIO, Jorge. *Curso de seguridad informática.* Apuntes electrónicos. 2002

- RIVAS, José Luis. *Protección de la Información***.** Virtua Libro (Libro electrónico). 2003.

- VV.AA. DOHERTY, Jim y ANDERSON, Neil. *Manual imprescindible de Redes Locales***.** Anaya. 2006. ISBN: 84-415-1980-3.

- VV.AA. PIATTINI, Mario Gerardo y DEL PESO , Emilio. *Auditoría Informática: un enfoque práctico.* Alfa-Omega - Ra-ma. 1998. ISBN: 84-789-7444-X

- ZAWODNY, Jeremy D. *MySQL avanzado : [optimización, copias de seguridad, replicación y equilibrado de carga]***.**Anaya multimedia.

Direcciones de Internet

http://www.mysql.com
http://www.mysql-hispano.org
http://mysql.rediris.es
http://www.isaca.org
http://www.agpd.es
http://www.eweek.com
http://www.webtaller.com/maletin/articulos/seguridad-mysql.php
http://www.securiteam.com
http://www.iec.csic.es/criptonomicon
http://www.recoverylabs.com
http://www.blasten.com/modules.php?cat=mysql
http://www.salnet.com.ar/inv_mysql/mysql.htm
http://www.cybsec.es
http://www.kryptopolis.com
http://www.aeat.es/normlegi/otros/lortad2000.htm
http://blog.segu-info.com/
http://www.wikipedia.es
http://en.wikipedia.org/
http://www.google.es
http://www.uc3m.es
http://www.itgi.org/
http://www.aenor.es/
http://gespadas.com/mysql-5-5
http://dmi.uib.es/~bbuades/auditoría/index.htm
http://www.calitae.com/manuales/tutorial-mysql.pdf
http://jtagua.wordpress.com/2010/06/20/mysql-workbench-community-edition/
http://dev.mysql.com/doc/workbench/en/index.html
http://www.criptored.upm.es/paginas/docencia.htm
http://www.hispasec.com
http://www.gpdnet.net+