



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

**“INFORME DE UN SISTEMA DE GESTIÓN  
DE CALIDAD PARA DISEÑO GRÁFICO.  
UTILIZANDO SOFTWARE LIBRE”**

**SEMINARIOS Y CURSOS DE  
ACTUALIZACIÓN Y CAPACITACIÓN  
PROFESIONAL**

**QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO EN COMPUTACIÓN**

**P R E S E N T A :**

**IVÁN ALEJANDRO NAVA CAMACHO**

**ASESOR: ING. SILVIA VEGA MUYTOY**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## ***DEDICATORIAS***

A mis padres Sergio e Irma por su inmenso amor, comprensión, apoyo y por creer en mi.

A mi hermano Sergio por su cariño incondicional.

A mis amigos por todas las penas y alegrías vividas juntos, y por su apoyo, gracias por su amistad.

A mi familia que siempre ha estado ahí cuando los necesito.

A todas las personas que han creído en mi...

## ***AGRADECIMIENTOS***

Agradezco a mis padres por estar presentes en cada uno de los momentos de mi vida, por compartir mis alegrías y tristezas, por no dejarme rendir en ningún momento y por que siempre me han brindado su amor.

A mi hermano, que siempre ha estado conmigo.

A mis amigos, por los momentos que hemos pasado juntos, las experiencias, éxitos y fracasos que compartimos.

A mi familia, por su apoyo incondicional, por creer en mi y apoyarme en todo momento, nunca me han dejado solo.

Y a todos aquellos que intervinieron para lograr este éxito, se los agradezco y los quiero.

<b>INTRODUCCIÓN</b>	<b>1</b>
<b>PRESENTACIÓN DEL DIPLOMADO DE DESARROLLO E IMPLEMENTACIÓN DE SISTEMAS CON SOFTWARE LIBRE EN LINUX.</b>	<b>1</b>
<b>OBJETIVO GENERAL.</b>	<b>1</b>
<b>OBJETIVOS ESPECÍFICOS.</b>	<b>1</b>
<b>I. INFORME GENERAL DEL DIPLOMADO EN “DESARROLLO E IMPLEMENTACIÓN DE SISTEMAS CON SOFTWARE LIBRE EN LINUX”.</b>	
<b>I.1 SISTEMA OPERATIVO LINUX</b>	<b>4</b>
I.1.1 CARACTERÍSTICAS	5
I.1.2 REQUERIMIENTOS DE LINUX	5
I.1.3 ESTRUCTURA DEL SISTEMA OPERATIVO	5
I.1.4 PROCESOS	6
I.1.5 COMANDO Y UTILERÍAS BÁSICAS	7
I.1.6 EDITOR DE TEXTO VI	10
<b>I.2 INSTALACIÓN Y ADMINISTRACIÓN DE LINUX</b>	<b>11</b>
I.2.1 PERFIL DEL ADMINISTRADOR	11
I.2.2 DISTRIBUCIONES	11
I.2.3 ETAPAS DE ALTA DEL SISTEMA OPERATIVO	16
I.2.4 NIVELES DE INICIO	16
I.2.5 PROGRAMAS DE ARRANQUE	16
I.2.6 CONFIGURACIÓN LILO	16
I.2.7 APAGADO DEL SISTEMA	17
I.2.8 ADMINISTRACIÓN DE USUARIOS Y GRUPOS	17
<b>I.3 EDITOR PARA LA CREACIÓN DE PÁGINAS WEB</b>	<b>19</b>
I.3.1 DOCUMENTO HTML	19
I.3.2 ETIQUETAS	20
I.3.3 LISTAS	21
I.3.4 TABLAS	21
I.3.5 FORMULARIOS	22
I.3.6 ENLACES	22
I.3.7 IMÁGENES	22
I.3.8 FRAME	22
<b>I.4 ADMINISTRACIÓN DE SERVIDORES WWW CON LINUX</b>	<b>23</b>
I.4.1 SERVIDOR HTTP APACHE	23
I.4.2 INSTALACIÓN	24
I.4.3 INICIANDO EL SERVIDOR APACHE	24
I.4.4 DIRECTIVAS	25
I.4.5 SERVIDORES VIRTUALES	25
I.4.6 CONTROL DE ACCESOS	26
I.4.7 PRUEBA AL SERVIDOR APACHE	26
<b>I.5 PROGRAMACIÓN EN PHP</b>	<b>27</b>
I.5.1 INSTALACIÓN DE PHP	27
I.5.2 COMENTARIOS	28
I.5.3 VARIABLES	29
I.5.4 CONSTANTES	29
I.5.5 OPERADORES	29
I.5.6 ESTRUCTURAS DE CONTROL	29
I.5.7 FUNCIONES	31
I.5.8 ARCHIVOS	31
I.5.9 FORMULARIO	31
I.5.10 CONTROL DE ACCESO	32

I.5.11 SESIONES	32
<b>I.6 INTERACCIÓN DE WWW CON BASE DE DATOS</b>	<b>33</b>
I.6.1 CARACTERÍSTICAS DE MYSQL	33
I.6.2 INSTALACIÓN Y CONFIGURACIÓN	33
I.6.3 INICIANDO MYSQL	35
I.6.4 ADMINISTRACIÓN	35
I.6.5 ESTRUCTURA DE MYSQL	36
I.6.6 COMANDOS DE SQL PARA MYSQL	36
I.6.7 INTERACCIÓN CON PHP	37
<b>I.7 INTRODUCCIÓN A LA SEGURIDAD EN CÓMPUTO</b>	<b>38</b>
I.7.1 DEFINICIONES	38
I.7.2 HERRAMIENTAS DE ESCANEADO O ATAQUES	40
I.7.3 MÉTODOS DE ATAQUES	41
I.7.4 MECANISMOS DE SEGURIDAD	42
I.7.5 CRIPTOGRAFÍA	43
I.7.6 ALGORITMO DIFFIE HELLMAN	44
I.7.7 FIREWALL	45
I.7.8 IPTABLE	45
I.7.9 RED VPN	46
<b>I.8 DESARROLLO DE APLICACIONES CON POSTGRESQL Y PHP</b>	<b>47</b>
I.8.1 PHP 5	47
I.8.2 PROGRAMACIÓN ORIENTADA A OBJETOS	48
I.8.3 POSTGRESQL	48
I.8.4 INTERACCIÓN CON PHP	51
I.8.5 SMARTY	52
I.8.6 ADO DB	54

## **II. PROYECTO DE UN SISTEMA DE GESTIÓN DE LA CALIDAD PARA DISEÑO GRÁFICO**

<b>II.1 DEFINICIÓN DEL SISTEMA</b>	<b>57</b>
II.1.1 JUSTIFICACIÓN	57
II.1.2 PROBLEMÁTICA	57
II.1.3 OBJETIVO DEL SISTEMA	57
II.1.4 ALCANCES	57
II.1.5 EXCLUSIONES	57
II.1.6 PLATAFORMA DE DESARROLLO	58
II.1.7 MENSAJES	58
<b>II.2 SISTEMA DE GESTIÓN DE CALIDAD PARA DISEÑO GRÁFICO</b>	<b>62</b>
II.2.1 MENÚ GENERAL DEL SISTEMA DE GESTIÓN DE CALIDAD PARA DISEÑO GRÁFICO	62
II.2.2 MÓDULOS DEL SISTEMA	66
<b>CONCLUSIONES</b>	<b>81</b>
<b>ANEXOS</b>	<b>82</b>
<b>REFERENCIAS</b>	<b>86</b>

## Introducción

Las comunicaciones y el manejo de la información han tomado gran importancia en cualquier empresa, desde compartir información, ideas, poder comunicarse de forma real con el cliente o los compañeros de trabajo, hasta mantener la información importante de la empresa lejos de ojos extraños o de competidores es vital, son cosas que preocupan.

Dentro de cualquier tipo de organización, el disponer de información completa, confiable en el momento oportuno, constituye un elemento esencial para garantizar la gestión eficaz de los recursos de la misma, así como, mejorar la calidad de los servicios que presta y adecuarse constantemente al entorno que lo rodea.

Por lo que se requiere, una administración adecuada de la información, que se planifiquen, desarrollen y mantengan sistemas de información eficientes, es decir, sistemas que produzcan en términos de calidad, cantidad y oportunidad la información que ayude o facilite el cumplimiento de los objetivos.

En la actualidad, las aplicaciones Web poseen características que las diferencian de los sistemas tradicionales. Estas características se deben al tamaño y complejidad de las aplicaciones, el carácter multidisciplinar del equipo de desarrollo, la tasa apresurada de entrega del proyecto, etc. Estas características hacen que los procesos, modelos y métricas existentes para evaluar la calidad tengan que ser adaptados para considerar los cambios impuestos por las nuevas tecnologías.

Combinado con la implementación de software libre, que garantiza un bajo costo, un alto rendimiento y una gran variedad de soporte, estas herramientas permiten a las empresas alcanzar un liderazgo nunca antes visto, al estar más cerca del cliente y poder abarcar más mercado al ofrecer sus servicios.

En el presente reporte se explicará de manera breve como se puede utilizar la mancuerna Software Libre – Aplicación en Web para poder desarrollar un sistema práctico que se pueda implementar a una empresa, tomando en cuenta la instalación del sistema operativo, la administración del servidor de Web, el desarrollo y administración de la base de datos e implementar las medidas de seguridad.

### ***Presentación del Diplomado de Desarrollo e Implementación de Sistemas con Software libre en LINUX.***

En la actualidad es necesario contar con sistemas que permitan la automatización de seguimiento y control de procesos que se desarrollan en diversas empresas y oficinas, para obtener un uso más eficiente de recursos y un control más preciso de los mismos, logrando, compartir información entre diversas áreas y oficinas, a través de la red local e incluso empleando Internet o la Intranet corporativa.

El desarrollo de estos sistemas puede ser sustentado empleando software libre como LINUX, PHP, APACHE-WWW-server y MySQL, que son herramientas que han demostrado tener un alto desempeño, gran escalabilidad y seguridad y, por el hecho de ser libres, permite reducir los costos que se generan por las licencias, logrando aprovechar al máximo los equipos de cómputo con que se cuenta.

Debido que creciente uso de las computadoras implica correr riesgos de seguridad en el acceso a la información, se ha vuelto necesario hacer conciencia de que la seguridad es una responsabilidad compartida; por lo que todos los usuarios de computadoras requieren un conocimiento esencial de los elementos de la seguridad que les ofrece Linux.

### ***Objetivo general.***

Conocer nuevas herramientas administrativas que le permitan desarrollar e implementar sistemas para el control de procesos e información, que funcionen de forma natural en red o por Internet, empleando herramientas de software libre que han demostrado tener una alta confiabilidad, alto desempeño y funcionalidad.

### ***Objetivos específicos.***

- Identificar los métodos de operación y administración del sistema operativo, utilizar los programas de manejo de usuarios, y llevar a cabo la instalación del dispositivo de respaldo, configuración de la red e instalación de software.
- Instalar el sistema operativo LINUX en PCs y configurar la tarjeta de red, video y particiones del disco duro; asimismo manejar el ambiente de usuario.

- Elaborar páginas WWW, apoyado con HTML.
- Identificar el procedimiento para instalar, configurar y administrar su propio servidor de WWW en un servidor de plataforma LINUX.
- Proporcionar al participante los conocimientos necesarios que le permitan crear aplicaciones dinámicas e interactivas para el Web utilizando el lenguaje PHP.
- Conocer y desarrollar una aplicación de bases de datos que funcione a través de WWW, empleando herramientas de software libre.
- Reconocer la importancia de la seguridad e identificar los elementos que le permitirán proteger el sistema y la información.

**I. INFORME GENERAL  
DEL DIPLOMADO EN  
“DESARROLLO E  
IMPLEMENTACIÓN DE  
SISTEMAS CON  
SOFTWARE LIBRE EN  
LINUX”**

## I.1 SISTEMA OPERATIVO LINUX

**OBJETIVO:** Identificar los métodos de operación y administración del sistema operativo, utilizar los programas de manejo de usuarios, y llevar a cabo la instalación del dispositivo de respaldo, configuración de la red e instalación de software.

LINUX es un sistema operativo, inspirado en UNIX como una implementación del sistema operativo de libre distribución para PC, servidores, y estaciones de trabajo. Fue desarrollado para la plataforma i386<sup>i</sup> por Linus Benedict Torvald, un estudiante finlandés que lo desarrollo a partir de una versión educativa de UNIX, llamada MINIX. LINUX es totalmente compatible con los estándares de UNIX y su publicación oficial fue el 5 de octubre de 1991 bajo la licencia GNU.

TABLA CRONOLÓGICA DE LOS EVENTOS MÁS IMPORTANTES DE LINUX		
Año	Evento	Descripción
1965	Origen	Bell Telephone Laboratories y General Electric Company intervienen en el proyecto MAC (del MIT) para desarrollar MULTICS.
1969-71	Infancia del UNIX	Se presenta la Versión 1 o Primera edición de UNIX, que nace de las cenizas de MULTICS.
1972-73	Nace el lenguaje de programación C	En la Versión 2 se da soporte al lenguaje de programación C y a los pipes. En la Versión 4 se reescribe UNIX en C.
1974-75	El momento	Las Versiones 5 y 6 de UNIX se distribuyen a las universidades. La Versión 6 circula en algunos ambientes comerciales y gubernamentales. AT&T impone ahora pagar una licencia, a pesar de que no puede promocionar UNIX debido a las duras regulaciones de EEUU al monopolio telefónico de AT&T.
1977	UNIX como producto	Interactive Systems es la primera compañía comercial que ofrece UNIX.
	Nace BSD	1BSD incluye un Shell Pascal, dispositivos y el editor ex.
1979	Versión 7	La Versión 7 de UNIX incluye el compilador completo K&R con uniones y definiciones de tipos, también añade el Bourne Shell.
	Trabajo en Red	BSD acrecentado por BBN incluye soporte para trabajar en red.
	Nace XENIX	Implementación para microcomputadoras ampliamente distribuido en hardware de bajo costo.
1980	Memoria Virtual	La capacidad de memoria virtual se añade en 4BSD.
	Nace ULTRIX	DEC realiza una versión de UNIX basado en BSD.
	Licencias en AT&T	La distribución de licencias abre el mercado.
1983	Nace System V	La versión más común de AT&T obtiene sus bases.
1984	Salida de SVR3	AT&T desata la versión más popular de System V hasta ahora.
1988	Motif vs Open Look	Sistemas por ventanas rivales son anunciados por OSF y UI.
	Siguiente paso	Un UNIX gráfico usa el Kernel Mach.
1990	OSF/1 vs. SVR4	Versiones rivales de UNIX son anunciadas por OSF y UI.
1992-95	Socialización	OSF/1 abandona la escena. SVR4 se convierte en el estándar. Sun vende más estaciones de trabajo para usuarios de Motif que para usuarios de Open Windows. Crece Windows/NT de Microsoft.

<sup>i</sup> i386 (Intel 80386) es un microprocesador CISC con arquitectura x86, con una arquitectura de 32 bits.

### I.1.1 Características

Las principales características de LINUX son:

- Multiproceso: Permite ejecutar más de una tarea, comando o aplicación simultáneamente.
- Multiplataforma: Funciona en la mayoría de las plataformas actuales.
- Multiusuario: Permite que múltiples usuarios estén conectados de manera simultánea y que puedan acceder a los recursos del sistema.

Otras características que tiene LINUX son:

- Protege la memoria para que un programa no pueda hacer caer al resto del sistema.
- Carga sólo las partes que se utilizan de un programa.
- Comparte la memoria entre las aplicaciones aumentando la velocidad y disminuyendo el uso de memoria.
- Utiliza un sistema de memoria virtual por páginas.
- Utiliza toda la memoria libre para caché<sup>ii</sup>.
- Permite utilizar bibliotecas enlazadas tanto estática como dinámicamente.
- Se distribuye con código fuente.
- Tiene un sistema de archivos avanzado pero puede usar los de los otros sistemas.
- Soporta redes tanto en TCP/IP como en otros protocolos.

### I.1.2 Requerimientos de LINUX

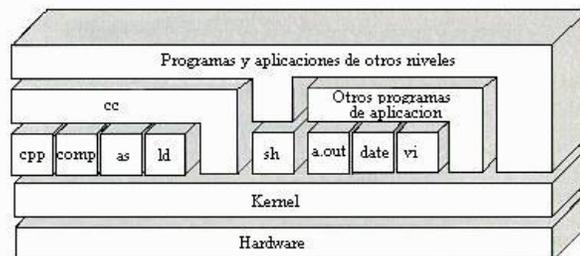
Debido a su eficiente aprovechamiento de recursos, LINUX tiene requisitos de hardware mínimos muy bajos: Una configuración mínima puede ser un procesador Intel tipo Pentium, superior y/o compatible, con 1MB de RAM, y una unidad de disquete (más teclado, tarjeta de vídeo, monitor, etc.). Esto es suficiente para arrancar y entrar al sistema.

Para un sistema más completo, se aconsejan 64 MB de memoria para el modo texto o 192 MB para modo gráfico. Si se van a tener muchos usuarios y/o muchos procesos a la vez, serían aconsejable de 256 MB a 512 MB que son suficientes para cargas pesadas a un máximo rendimiento. En lo que respecta a disco duro, depende de las aplicaciones que se instalen, se va desde los 10 MB básicos con los programas y archivos necesario hasta los 3 ó 4 GB de una distribución instalado con varias aplicaciones (incluye compiladores, paquetes de oficina, interfaz gráfica, etc.).

### I.1.3 Estructura del sistema operativo

La estructura de LINUX esta formada por cuatro elementos situados en bloques diferentes, cada uno de los cuales tiene encomendada una función y si se toma en cuenta al usuario serian 5 elementos:

- Hardware.
- Núcleo o Kernel.
- Shell.
- Aplicaciones.
- Usuario



I.1.1 Arquitectura del Sistema Unix

El usuario utiliza todas las aplicaciones basadas en LINUX, así como sus utilidades y ordenes propias. Su información estará organizada mediante un sistema de archivos; de todo el sistema sólo puede interactuar con el shell (encargado de interactuar con el usuario y el kernel o núcleo) que a su vez es el interprete de comandos, así como un pequeño lenguaje de programación con el que programar nuevas funciones o personalizar algunas existentes.

---

<sup>ii</sup> Caché: es un conjunto de datos duplicados de otros originales, con la propiedad de que los datos originales son costosos de acceder, normalmente en tiempo, respecto a la copia en el caché.

## I.1. Sistema Operativo LINUX

El núcleo es el único que interactúa con el hardware. Incluye entre sus funciones las operaciones más importantes de gestión del sistema operativo, como puede ser la gestión de memoria, mantenimiento del sistema de archivos, asignación de tiempo del CPU a cada una de las tareas, el control del acceso a una PC mediante claves, etc...

### **Estructura de los directorios**

La estructura básica de LINUX se comprende de los siguientes directorios:

Estructura de directorios	
Directorio	Contenido
/bin	Comandos ejecutables, compilados en forma dinámica
/boot	Núcleo, información de booteo
/dev	Dispositivos
/etc	Casi todos los archivos de configuración
/home	Directorios de cada usuario
/lib	Mayoría de las bibliotecas de funciones
/proa	Archivos especiales con información del estado del equipo
/root	Directorio del superusuario
/sbin	Binarios compilados en forma estática
/tmp	Archivos temporarios
/var	Archivos de monitoreo del equipo, spooler de impresoras, mail, etc.
/usr	Comandos, documentación, archivos del entorno gráfico, fuentes del núcleo, etc.

### **Sistemas de archivo**

Un sistema de archivos consta de tipos de datos abstractos, que son necesarios para el almacenamiento, organización jerárquica, manipulación, navegación, acceso y consulta de datos.

La mayoría de los sistemas operativos poseen su propio sistema de archivos. Los sistemas de archivos son representados ya sea textual o gráficamente utilizando gestores de archivos o shells. Los más comunes utilizan dispositivos de almacenamiento de datos que permiten el acceso a los datos como una cadena de bloques de un mismo tamaño, a veces llamados sectores, usualmente de 512 bytes de longitud. Los sistemas de archivo más utilizados por LINUX son: EXT2 (second extended filesystem), EXT3 (third extended filesystem) y ReiserFS.

### **I.1.4 Procesos**

Un sistema LINUX puede prestar muchos servicios simultáneamente, puede ser servidor de Web, al tiempo que es servidor de correo electrónico, puede atender varios usuarios y cada usuario puede estar realizando simultáneamente diversas acciones. A cada acción en un sistema LINUX se le llama proceso. Un proceso abstrae una acción que el sistema debe realizar, independiente del momento en que debe ejecutarse.

Cada proceso tiene asociado un número que lo identifica, un estado que indica como está operando, un grupo que lo asocia con otros procesos, una prioridad que determina su "importancia" con respecto a otros procesos y un dueño que puede controlarlo (normalmente el dueño es el usuario que inicia el proceso).

Cada programa que inicie desde la línea de comando se ejecutará en un nuevo proceso que por defecto estará en primer plano, es decir que bash suspenderá su ejecución y la reanudará cuando el programa que inició termine.

**ls -lR > arbol.txt &**

COMANDOS PARA EL MANEJO DE PROCESOS			
COMANDOS	DESCRIPCIÓN	SINTAXIS	Opciones
<b>JOBS</b>	lista por pantalla todos los trabajos que se han lanzado en background	jobs	

COMANDOS PARA EL MANEJO DE PROCESOS			
COMANDOS	DESCRIPCIÓN	SINTAXIS	Opciones
<b>PS</b>	Muestra los procesos activos	ps -[opciones]	-e: Procesos de todas las sesiones. -f : Proporciona los números del PID, del Ppid (padre), uso del procesador y tiempo de comienzo.
<b>FG</b>	Manda un proceso de primer plano a segundo plano	fg [% número proceso]	
<b>BG</b>	Regresa un proceso de segundo plano a primer plano	bg [% número proceso]	
<b>KILL</b>	Termina procesos	kill <número de señal> numero de proceso	
<b>TOP</b>	Presenta una lista de la actividad de los procesos en el sistema en tiempo real.	top	

### I.1.5 Comando y utilerías básicas

Un comando es una instrucción que el usuario le da al sistema, el cual generalmente está contenido en un archivo binario. Suele admitir modificadores, normalmente se indican tras un guión simple "-".

Una de las interfaces para el sistema operativo es la que maneja órdenes y en ella el usuario debe teclear el comando correspondiente para la ejecución de una operación en particular. Los comandos más utilizados son:

#### Identificación

Estos comandos se utilizan para obtener información del usuario con el que se esta conectado y de los usuarios que están conectados al servidor.

COMANDOS	DESCRIPCIÓN	SINTAXIS
<b>WHO</b>	Presenta quien esta conectado actualmente al servidor.	who
<b>WHOAMI</b>	Proporciona información del usuario con el que se esta conectado.	whoami

#### Administración de archivos y directorios.

Un sistema LINUX puede tener más de un millón de archivos que se identifican por nombres que pueden organizarse dentro de directorios. En el sistema de archivos de LINUX, los directorios se organizan en manera de árbol a partir de un directorio principal llamado directorio raíz; y que se denota por '/'. Cada directorio puede contener otros directorios o nombres de archivos.

**Permisos, dueño y grupo-** Dado que LINUX es multiusuario podría haber la posibilidad de que un usuario quisiera leer los archivos de otro. Para esto los archivos y directorios cuentan con 3 medidas de protección:

- 1) Permisos que se le asignan al archivo, estos pueden ser r (lectura), w (escritura) y x (ejecución). Estos permisos están divididos en 3 partes que se denominan UGO (Usuario-Grupo-Otros).
- 2) Un dueño que es el usuario que los creo y es el que tiene control sobre el.
- 3) Un grupo al que pertenece, que es al que pertenece el dueño.

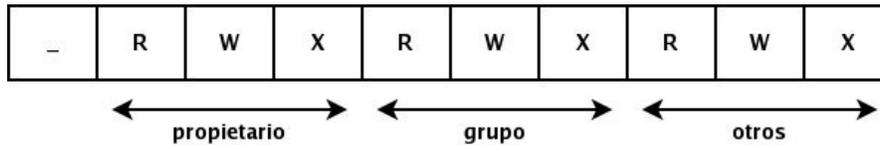
Los comandos de la siguiente tabla son los más usuales para la administración de archivos, entre las acciones que realizan están: listar, borrar, renombra, copiar o crear.

COMANDOS	DESCRIPCIÓN	SINTAXIS
<b>MKDIR</b>	Crea directorios	mkdir [-p] Nombre_Directorio
<b>CD</b>	Cambia de directorio de trabajo, utilizando rutas relativas o absolutas. <b>NOTA: SI SE TECLEA CD SIN OPCIONES REGRESA AL USUARIO A SU DIRECTORIO HOME.</b>	cd [Directorio_deseado]
<b>CP</b>	Copia un archivo origen en el directorio indicado.	cp archivo_origen archivo_copiado
<b>PWD</b>	Muestra la ruta completa del directorio de trabajo.	pwd
<b>MV</b>	Mueve o renombra archivos o directorios.	mv Archivo Directorio o mv Archivo Archivo_Nuevo_Nombre
<b>RMDIR</b>	Borra directorios vacíos, si el directorio tiene contenido marca un error.	rmdir directorio_vacio_borrar
<b>RM</b>	Borra archivos o directorios con contenido.	rm [-r] directorio_borrar
<b>CAT</b>	Muestra el contenido de un archivo de forma continua.	cat archivo
<b>MORE</b>	Muestra el contenido de un archivo, de forma periódica desplegando la información en forma de hojas.	more archivo
<b>TOUCH</b>	Crea archivos vacíos	touch nombre_archivo
<b>ECHO</b>	Expresa el contenido en el valor proporcional, usado para ver el contenido de las variables del sistema.	echo Variable

### Permisos

LINUX implementa una serie de permisos para mantener la privacidad de los usuarios, de manera que un usuario no pueda simplemente abrir un archivo que no le pertenece. Los permisos que maneja LINUX son: lectura (r), escritura (w) y ejecución (x).

Los permisos se manejan en tercias pudiendo asignar permisos al usuario, al grupo o a otros, lo que se denomina UGO. Cada parte de UGO tiene sus propios permisos.



I.1.2 Permisos

Los comandos que se utilizan para cambiar los permisos son:

COMANDOS	DESCRIPCIÓN	SINTAXIS
<b>CHMOD</b>	Cambia los permisos de cada archivo dado utilizando una representación simbólica o un número octal que representa el patrón de bits de los nuevos permisos.	<b>Representación simbólica</b> chmod <usuario> +/- <permiso> <archivo> Donde: <usuario> = u-user, g-group, o-others + Agregar; - Quitar <permiso> = r-read, w-write, x-exec  <b>Permisos en notación octal</b> chmod ### archivo ### = Ver tabla de permisos en numeración octal.
<b>CHOWN</b>	Cambia el usuario propietario de un archivos. Opcionalmente permite cambiar también el grupo.	chown [opciones] usuario[:grupo] archivo...

COMANDOS	DESCRIPCIÓN	SINTAXIS
<b>CHGRP.</b>	Cambia de grupo a los archivos dados. Con la opción -R cambia los subdirectorios de forma recursiva.	chgrp [-R] nuevo_grupo archivo

Número octal	Permiso
4000	Establece el número de identificación de usuario al ejecutarse
2000	Establece el número de identificación de grupo al ejecutarse
1000	Establece el bit adhesivo
0400	Lectura por parte del dueño
0200	Escritura por parte del dueño
0100	Ejecución por parte del dueño
0040	Lectura por parte del grupo
0020	Escritura por parte del grupo
0010	Ejecución por parte del grupo
0004	Lectura por parte de los otros
0002	Escritura por parte de los otros
0001	Ejecución por parte de los otros

*Tabla de permisos en numeración octal*

### Comandos del sistema

A continuación se presenta una tabla de los comandos más utilizados en LINUX para administra el sistema:

COMANDOS	DESCRIPCIÓN	SINTAXIS
<b>DATE</b>	Muestra la fecha del sistema.	date
<b>UNAME</b>	Muestra la información "personal" del sistema en que se esta conectado.	uname
<b>CAL</b>	Muestra un calendario. Si no se le proporcionan opciones muestra el calendario del mes actual.	cal [mes] [año]
<b>MAN</b>	Muestra las hojas de manual de los comandos.	man comando
<b>LS</b>	Lista el contenido de un directorio.	ls [opciones] directorio
<b>CLEAR</b>	Semejante al cls de MS-dos limpia la pantalla o la consola.	clear
<b>CHFN</b>	Cambia los valores del GECOS del etc/password	chfn Nuevo_valor
<b>CHSH</b>	Cambia el tipo de shell de la cuenta de un usuario.	chsh Nuevo_valor
<b>SU</b>	Permite conectarse al servidor con otro usuario sin terminar la sesión actual.	su - login (como root no pide password)
<b>TREE</b>	Lista en forma de árbol los directorios y subdirectorios	
<b>WC</b>	Cuenta, número de líneas, palabras bite de la entrada que se le proporcione.	wc [-l, -w, -c] archivo
<b>LAST</b>	Lista las últimas conexiones hechas al equipo	last
<b>GREP</b>	Permite realizar búsquedas dentro de los archivos utilizando expresiones regulares.	grep [-ri] expresión_regular
<b>FIND</b>	Sirve para buscar archivos o directorios en el sistema.	find [donde va a buscar] -name Nombre del archivo [-print]
<b>TEE</b>	Manda tanto a pantalla como a un archivo el resultado de la salida estándar	tee archivo
<b>MAIL</b>	Sirve para mandar un correo	Mail dirección
<b>WRITE</b>	Un mensajero en tiempo no-real	write login@servidor

### Redireccionamiento

Muchos comandos LINUX toman su entrada de algo conocido como entrada estándar y envían su salida a la salida estándar. El interprete de comandos configura el sistema de forma que la entrada estándar es el teclado y la salida la pantalla.

Cuando se esta leyendo de la entrada estándar, los comandos reconocen el fin de la entrada de datos cuando reciben el carácter EOT (end-of-text, fin de texto). Existen dos tipos de redireccionamiento: destructivo o no destructivo.

TIPO DE REDIRECCIONAMIENTO	DESCRIPCIÓN	SÍMBOLO	Ejemplo
SALIDA ESTÁNDAR	Permite enviar la salida de un comando o acción a un archivo en lugar de a la pantalla.	>	<i>echo "hola" &gt; saludos.txt</i>
	Si el archivo existe lo sobrescribe. Si el archivo existe agrega la salida estándar al final (Direccionamiento no destructivo).	>>	<i>echo "hola" &gt;&gt; saludos.txt</i>
ENTRADA ESTÁNDAR	Permite enviar la información contenida en un archivo como entrada a un comando.	<	<i>sort &lt; lista.txt</i>
PIPES (USO DE TUBERÍAS).	Permiten conectar una cadena de comandos en tubo, donde la salida estándar del primero es enviada directamente a la entrada estándar del segundo y así sucesivamente		<i>ls   sort -r</i>

### Ligas

Una liga no es un archivo, es un nombre adicional para otro archivo. Cada archivo tiene al menos una liga, usualmente el nombre bajo el cual fue originalmente creado. Cuando se hace una nueva liga hacia un archivo, un alias para este archivo es creado. Una liga es indistinguible del archivo al cual está ligado; LINUX mantiene el conteo de la cantidad de ligas que apuntan hacia un archivo en particular y no libera el espacio que ocupa el archivo hasta que la última liga es eliminada. Las ligas se crean utilizando el comando ln. Existen dos tipos de ligas: duras y suaves o simbólicas.

*ln [-s] archivo\_origen archivo\_liga*

La diferencia entre las ligas, es que las duras son una referencia directa, mientras las simbólicas son una referencia a través de un archivo, las simbólicas son el archivo en sí, por lo tanto, tienen un propio dueño y permisos.

TABLA COMPARATIVA		
	LIGAS SUAVES	LIGAS DURAS
<b>i-nodo</b>	Distintos	Iguales
<b>Contador de Ligas</b>	No cambia	Se incrementa a 1 por cada liga dura
<b>Disco</b>	Ocupa un espacio distinto al archivo al que hace referencia	Ocupa el mismo espacio físico de almacenamiento del archivo referenciado
<b>Tamaño en bytes</b>	El tamaño es igual a la longitud del nombre del archivo en bytes	Igual al tamaño del archivo referenciado.

### I.1.6 Editor de texto vi

Este es el editor más utilizado en UNIX y LINUX, dado que se puede acceder a él por medio de consola sin necesidad de interfaz gráfica. La sintaxis del comando vi es, al igual que la de cualquier comando UNIX:

*vi [opciones] archivo.*

En vi existen 3 modos de trabajo:

- **MODO TEXTO.** Se emplea para añadir o modificar el texto del documento.
- **MODO COMANDO.** Cada tecla tiene un significado, y se puede desplazar el cursor, copiar, pegar, mover, borrar, etc. Hay que tener mucho cuidado, pues un comando en minúsculas o en mayúsculas tienen efectos muy diferentes.
- **MODO EX O ULTIMA LÍNEA.** Inserción de comandos del editor de líneas ex. Se emplea para cambios masivos en el documento.

## I.2 INSTALACIÓN Y ADMINISTRACIÓN DE LINUX

**OBJETIVO:** Instalar el sistema operativo LINUX en una PC y configurar la tarjeta de red, video y particiones del disco duro; asimismo manejar el ambiente de usuario.

El administrador toma un papel importante dentro de la organización, al ser él, que se encarga de todo el manejo del servidor, en él recae la responsabilidad de la continuidad del servicio, además de ser encargado de proporcionarlo de alta calidad.

### I.2.1 Perfil del administrador

Un administrador de un servidor debe de cumplir ciertas características. El perfil del administrador se puede definir como: *"Responsable de configurar, mantener y actualizar el sistema que conforma una red. Debe cuidar del funcionamiento del hardware, software y de los periféricos"*.

RESPONSABILIDADES	CONOCIMIENTOS	ESTABLECER POLÍTICAS
<ul style="list-style-type: none"> <li>• Proporcionar un ambiente seguro y confiable.</li> <li>• Administrar los usuarios del sistema.</li> <li>• Configurar los dispositivos.</li> <li>• Hacer respaldos.</li> <li>• Capacitar y asesorar a los usuarios.</li> <li>• Asegurar el sistema.</li> <li>• Registrar los cambios del sistema llevando un registro en bitácoras.</li> <li>• Instalación y mantenimiento de los dispositivos.</li> <li>• Configurar la interfaz de red.</li> <li>• Administración de los recursos del sistema.</li> <li>• Monitoreo del sistema.</li> <li>• Detección de fallas.</li> <li>• Auditoria e implantación de la seguridad del sistema.</li> <li>• Conocer y estar actualizado el hardware y el software del equipo.</li> </ul>	<ul style="list-style-type: none"> <li>• Nociones técnicas de Programación.</li> <li>• Dominio de al menos un lenguaje de programación.</li> <li>• Funcionamiento del sistema operativo.</li> <li>• Técnica de Administración del sistema operativo.</li> <li>• Conocimientos básicos de hardware y mantenimiento de dispositivos.</li> <li>• Conocimiento y comprensión profunda de redireccionamiento, tuberías, procesamiento en segundo plano, etc.</li> <li>• Manejo del editor de texto vi.</li> <li>• Conocimiento de programación en shell.</li> </ul>	<ul style="list-style-type: none"> <li>• Creación de cuenta de usuario.</li> <li>• Establecer horarios de mantenimiento de hardware y software.</li> <li>• Borrar archivos temporales.</li> <li>• Cuotas de discos.</li> <li>• Seguridad en el sistema.</li> </ul>

### I.2.2 Distribuciones

Una distribución es un conjunto de aplicaciones reunidas por un grupo, empresa o persona para permitir instalar fácilmente un sistema operativo LINUX. Existen numerosas distribuciones, cada una de ellas puede incluir cualquier número de software adicional (libre o no), como algunos que facilitan la instalación del sistema y una enorme variedad de aplicaciones, entre ellos, entornos gráficos, suites ofimáticas, servidores Web, servidores de correo, servidores FTP, etcétera. La base de cada distribución incluye el kernel, con las bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos/grupos de software, como BSD.

LINUX Standard Base (Fundación de estándares LINUX) es una organización consagrada a desarrollar una cooperación estrecha entre diferentes distribuciones. El Filesystem Hierarchy Standard (Estándar jerárquico de sistema de archivos) es una importante herramienta de la organización para lograr una cierta normalización oficial.

Algunas distribuciones de LINUX son:

- Debian GNU/LINUX.
- Slackware (la primera distribución de LINUX).
- Fedora Core (x86/PPC, basada en Red Hat LINUX).

- Gentoo LINUX (x86/PPC).
- Knoppix (basada en Debian GNU/LINUX, de tipo CD Live).
- Mandriva LINUX x86/PPC (antes Mandrake LINUX, basada en Red Hat LINUX),
- Linspire (antes Lindows, basada en Debian GNU/LINUX).
- Red Hat Enterprise LINUX (antes Red Hat LINUX).
- SuSE LINUX (x86/PPC).
- YellowDog (exclusivo para PPC, basada en Fedora Core PPC).

### **Slackware LINUX**

La distribución que se eligió en este caso para instalar es: Slackware LINUX. Una distribución multitarea de 32-bits. Actualmente, se encuentra en la versión 10.1 y está basada en el kernel LINUX 2.4.29, pero puede utilizar el kernel versión 2.6.10 y la librería C del GNU versión 2.3.4.

La primera versión de Slackware salió a la luz en Abril de 1993, y desde sus inicios obedece a los estándares de LINUX publicados, como el LINUX File System Standard<sup>iii</sup>. La distribución de paquetes en Slackware se hace principalmente con archivos .tar comprimidos, si bien se puede utilizar también RPM. La interfase del programa de instalación es de modo texto.

Su instalación es una de las más fáciles de usar, proporcionando un modo gráfico basado en menús, tiene una extensa documentación, y un sistema de manejo de paquetes también basado en menús. El creador de la distribución es Patrick Volkerding. El sistema fue diseñado con dos objetivos: facilidad de uso y estabilidad. Incluye las más populares aplicaciones, conservando un sentido de tradición proporcionando simplicidad y facilidad de uso junto al poder y la flexibilidad.

Por razones de simplicidad, Slackware ha sido dividido en series de software. Una llamada "disk sets" por que fue diseñada para la instalación mediante Floppy, las series de software se usan actualmente para categorizar los paquetes incluidos en la instalación.

Slackware LINUX se puede conseguir en Internet, descargando las ISO<sup>iv</sup> de los CD de instalación (imágenes de CD's listas ya para grabarse<sup>v</sup>). Algunas páginas para descargar Slackware LINUX son:

- <http://www.slackware.com> (ingles)
- <http://www.eslackware.com> (español)
- <http://www.linuxiso.org> (ingles)

### **Instalación**

Lo primero a considerar para instalar LINUX es que el equipo cumpla las características de la instalación, lo segundo es haber escogido la distribución que se adecue más a las necesidades del usuario, como se mencionó anteriormente se instalara Slackware, una de las versiones de LINUX más veteranas, que cuenta con un gran soporte (foros de ayuda, paginas, etc) y se actualiza regularmente.

Los requisitos mínimos para poder instalar un servidor con un sistema operativo LINUX son:

- Procesador 486 superior o compatible
- Memoria RAM de 16 mb (32 mb recomendados)
- 100-500 mb de espacio en disco duro para la instalación mínima
- Floppy y/o CDRom
- Tarjeta de Red
- Software

---

<sup>iii</sup> Estándar de Jerarquía de Sistema de Archivos o FHS define los directorios principales y sus contenidos en LINUX y otros sistemas de la familia UNIX.

<sup>iv</sup> Archivo donde se almacena una copia o imagen exacta de un sistema de archivos, normalmente un disco compacto (como un CD o un DVD). Se rige por el estándar ISO 9660 de la Organización Internacional para la Estandarización.

<sup>v</sup> Significa copiar archivos o el contenido de una ISO a un medio de almacenamiento generalmente un CD o un DVD.

## I.2 INSTALACIÓN Y ADMINISTRACIÓN DE LINUX

---

Para empezar la instalación hay que asegurar que la PC inicie booteando<sup>vi</sup> desde el CD en lugar del disco duro, para esto hay que modificar la configuración del BIOS del equipo, hay que tomar en cuenta que en cada PC es diferente la forma de entrar al BIOS y que éste cambia dependiendo del fabricante del equipo y versiones de hardware y software. Terminando de configurar el booteo hay que introducir el CD 1 de la distribución (en algunas distribuciones se utiliza más de un disco, llegando incluso a utilizarse DVD's), Slackware utiliza dos discos de instalación.

En la pantalla siguiente se configura el teclado, donde se tiene que escoger el idioma que el usuario maneje con la finalidad de facilitar la instalación. Se tiene que escoger la opción es.map para configurar el teclado en idioma español.

Al terminar de configurar el teclado sólo hay que presionar enter para que el programa de instalación proporcione una consola<sup>vii</sup> con privilegios de root, esto es para verificar/crear las particiones y seguir la instalación.

### **Creando las particiones**

El sistema operativo LINUX necesita dos particiones en el disco duro para poder funcionar de manera correcta, una donde se crearan los sistemas de archivos de LINUX y es donde se instalará la denominada raíz ("/") y una para la memoria SWAP.

Cuando se este en la consola se va a utilizar el comando fdisk. Fdisk se utiliza para crear/borrar particiones y darles formato. El primer disco duro IDE (primary master) es /dev/hda, el esclavo primario(primary slave) es /dev/hdb, así sucesivamente.

Las opciones que más se utilizan son:

Opción	Descripción
d	Borrar Particiones
l	Muestra los tipos de particiones conocidos o reconocidos por fdisk
m	Muestra el menú de ayuda
n	nueva partición
p	Imprime la tabla de particiones del disco duro
q	Salir sin guardar
t	Se cambia el tipo de partición
w	Escribir o Guardar los cambios hechos en la tabla de pariciones

**NOTA: ES MUY IMPORTANTE QUE SE RESPALDE CUALQUIER INFORMACIÓN QUE SE DESEE CONSERVAR ANTES DE SEGUIR ADELANTE CON LA CREACIÓN DE SUS NUEVAS PARTICIONES , YA QUE EL CONTENIDO DE ESTAS SERÁ COMPLETAMENTE DESTRUIDO.**

Después de estudiar el menú, es hora de trabajar, para esto hay que ver las particiones existentes en el disco duro, a fin de no sobrescribir alguna donde se tenga otro sistema operativo o información importante, para esto se utiliza la opción **p: Command (m for help): p**

Esto desplegará toda la información sobre las particiones actuales. Generalmente se escoge una partición libre para la instalación, aunque se puede utilizar una ocupada formateándola, pero se perdería la información.

Hay que tener en cuenta si se va a instalar otro sistema operativo además de LINUX. Para borrar una partición hay que hacer lo siguiente:

#### **1) En el prompt hay que usar la opción d**

**Command (m for help): d**

**Partition number (1-4): #**

**# es el número de la partición que se quiere eliminar, estos se ven con la opción p.**

---

vi Arranque, bootaje o booteo es un proceso que inicia el sistema operativo cuando el usuario enciende un sistema informático. En inglés se conoce como booting.

vii Llamada también interprete de comandos o shell se encargada de ejecutar las órdenes básicas para el manejo del sistema. También se denomina shell.

## I.2 INSTALACIÓN Y ADMINISTRACIÓN DE LINUX

Como ya se menciono LINUX necesita dos particiones *SWAP* y / (*raíz*). Para crear las particiones se utiliza la opción *n*, el programa preguntara si la partición sera de tipo *e* ( extendida ) o *p* ( primaria ), LINUX se puede instalar en particiones primarias o lógicas, a diferencia de Windows. La decisión de instalar LINUX en una partición primaria o en una lógica es al gusto y necesidades del usuario, en este caso se usa *P* (primaria)para crear la partición la *SWAP*.

Para generar la partición de la memoria *SWAP* se tiene que tomar en cuenta las ciertas reglas:

- La memoria *SWAP* debe tener el doble de la memoria *RAM* del equipo.
- La memoria *SWAP* no debe de preferencia sobrepasar los 512 MB.

En este caso se va a utilizar una *SWAP* de 256 Mb, por lo que se tiene que teclear *+256M*, esto crea una partición de 256 MB.

Lo siguiente es crear la partición para *raíz* (/). Los pasos son los mismos que para crear la memoria *SWAP*, salvo algunas excepciones. Para LINUX se pueden crear múltiples particiones para los diferentes directorios del sistema: /home, /dev, /etc. Eso dependen de las necesidades del usuario y nivel de seguridad del servidor.

Por ultimo hay que darle formato a las particiones, los tipos de formato de partición se listan con la opción *L*.

### El Programa de Instalación

Una vez creadas las particiones, se puede empezar la instalación de Slackware. Para esto hay que ejecutar el programa de instalación, esto se realiza escribiendo **setup** en el prompt de la consola ( en donde se queda al terminar *fdisk*). Setup es un sistema manejado por menús (imagen I.2.1) que sirve para instalar los paquetes de Slackware y configurar el sistema.



I.2.1 Menú principal de la instalación

El proceso de instalación es así: se va paso a paso a través de cada una de las opción del programa de instalación. Las opciones del menú se seleccionan usando las teclas de navegación arriba y abajo, "Ok" y "Cancel" se seleccionan usando la flechas de izquierda y derecha. Las opciones que son elegibles o seleccionables (aquéllas indicadas con una [X]) se marcan usando la barra espaciadora.

**SOURCE** En la sección Source se selecciona de donde se va a instalará Slackware. Hay cuatro fuentes para escoger Disquetes, CD-ROM, NFS, o un directorio PRE-montado (imagen I.2.2).



I.2.2 Menú Source

## I.2 INSTALACIÓN Y ADMINISTRACIÓN DE LINUX

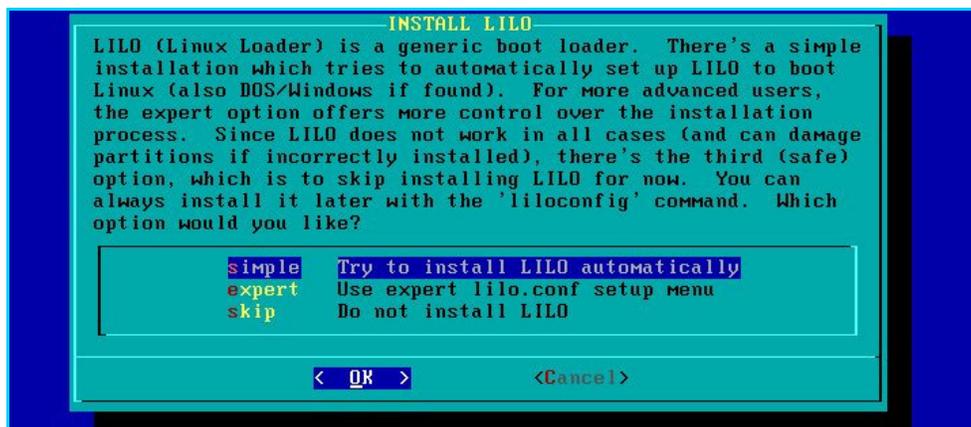
---

En este caso se selecciona la opción de CD-ROM. El programa ofrece la opción de buscar de forma automática la unidad de CD o en caso contrario muestra una lista para escoger el tipo de unidad de CD. Lo recomendable es dejar que el programa escoja la unidad de CD. Si todo a salido bien el programa pasará a la siguiente etapa de la instalación a través de su unidad de CD-ROM.

**INSTALL** Esta opción permite seleccionar seis tipos diferentes de instalación: "full", "newbie", "menu", "expert", "custom", "tagpath" y "help". La opción full, que es la que se va a utilizar, instalará cada paquete de todas las series de software que se escogieron en la sección "select". Éste es el método de la instalación más fácil y rápido, ya que no se necesita decidir los paquetes a instalar. Esta opción es recomendable usarla si se cuenta por lo menos con una partición de 3G de espacio.

**CONFIGURE** La sección configure permite hacer una configuración básica del sistema, ahora que los paquetes se han instalado. Lo que se ve aquí depende principalmente del software que se ha instalado. Sin embargo, siempre se verá lo siguiente:

**LILLO** (LINUX Loader o cargador de LINUX ). Este programa se utiliza sobre todo cuando se tiene más de un sistema operativo instalado, ya que su utilidad radica en mostrar un menú con los sistemas operativos booteables (imagen I.2.3).



I.2.3 Menú de instalación de LILLO

**X Window Manager** (El Gestor de Ventanas X) Esta sección permitirá configurar un gestor window predeterminado para X. Aquí se escoge la opción de KDE que es el entorno de escritorio que se va a utilizar.

La última cosa que hará el programa de instalación es pregunta si se desea poner una contraseña a root, esto es recomendable para mantener la integridad y seguridad del sistema y la información dentro de el.

**EXIT** Es la pantalla final de la instalación (imagen I.2.4). Para salir del programa de instalación se tiene que presionar OK, con esto mostrara el menú principal del programa de instalación donde se tiene que escoger la opción EXIT para regresar a la consola. Ya en el prom de la consola hay que presionar las teclas Ctrl+alt+del para reiniciar la máquina, también hay que sacar el CD de Slackware de la unidad de CD-rom para evitar entrar de nuevo al programa de instalación. Al prender la maquina debe presentarse el menú de LILLO y mostrar la opción de Slacware para iniciar el sistema operativo.



I.2.4 Fin de la instalación

### ***1.2.3 Etapas de alta del sistema operativo***

El sistema operativo LINUX tiene que cumplir una serie de pasos para poder arrancar y funcionar de manera correcta, estos pasos son:

1. Leer MBR (Master Boot Record)
  2. Leer y verificar el BIOS (Basic Input-Output System - sistema básico de entrada-salida. Programa incorporado en un chip de la placa base que se encarga de realizar las funciones básicas de manejo y configuración de la computadora).
- Carga del sistema operativo LINUX--**
3. Cargar el Kernel
  4. Carga de los dispositivos
  5. Lee el nivel de inicio configurado en el archivo inittab (/etc/inittab)
  6. Ejecuta los scripts del RunLevel
  7. Ejecuta el rc.local

**NOTA: TODOS LOS CAMBIOS QUE SE HAGAN A LA CONFIGURACIÓN DEL SISTEMA SE TIENEN QUE REALIZAR COMO ROOT PARA TENER LOS PERMISOS NECESARIOS.**

### ***1.2.4 Niveles de inicio***

El nivel de inicio o runlevel es el modo en que arranca LINUX. Por defecto el runlevel suele ser el 2 ó 3, es decir, arranque en modo texto o consola o en modo gráfico. Para cada distribución suele haber una lista de runlevels y sus significados.

#### ***Modo gráfico de Inicio***

Para iniciar con el modo gráfico existen dos formas: a) Modificar el runlevel a 4 o b) Utilizar el runlevel 3 y el archivo rc.local.

**NOTA: ANTES DE MODIFICAR ALGÚN ARCHIVO DE CONFIGURACIÓN ES SIEMPRE RECOMENDABLE CREAR UNA COPIA CON EL FIN DE TENER UN RESPALDO SI SE PRESENTA ALGÚN ERROR, ASÍ SIEMPRE SE PODRÍA REGRESAR A LA CONFIGURACIÓN ANTERIOR.**

De este modo ya se puede tener el nivel 3 con interfaz gráfica, y se mantiene las shell del sistema que por default da 6.

### ***1.2.5 Programas de arranque***

Los programas de arranque son los programas que se inician cuando se carga el sistema, por lo general se utilizan scripts para esta. Estos script se encuentran en la dirección /etc/rc.d/.

Este directorio contiene todos los scripts que inician y detienen los servicios asociados con todos los runlevel. El almacenar todos los comandos asociados con cada uno de los niveles de ejecución simplifica la asociación múltiple de un único servicio dado con varios niveles de ejecución.

La lista de los servicios asociados con cada nivel de ejecución de LINUX se almacena en un directorio específico para cada nivel. Por ejemplo, los servicios asociados al runlevel 3 se encuentran contenidos en el directorio /etc/rc.d/rc3.d.

Si se quiere que un programa inicie al arrancar el sistema se tiene que poner poner un script de ejecución, y se le tiene que cambiar los permisos para que el sistema lo identifique como ejecutable esto se hace con el comando chmod.

### ***1.2.6 Configuración LILO***

LILO (LINUX Loader) es un sistema de gestor de arranque. Se puede configurar de dos maneras:

- Manual – donde el administrador tiene que modificar los archivos de configuración.
- Por programa – por medio del programa liloconfig, que presenta una interfaz de menús.

Si se realiza la configuración a mano es recomendable siempre hacer una copia de respaldo de los archivos que se van a modificar. El archivo a modificar se encuentra en /etc/ y se llama lilo.conf.

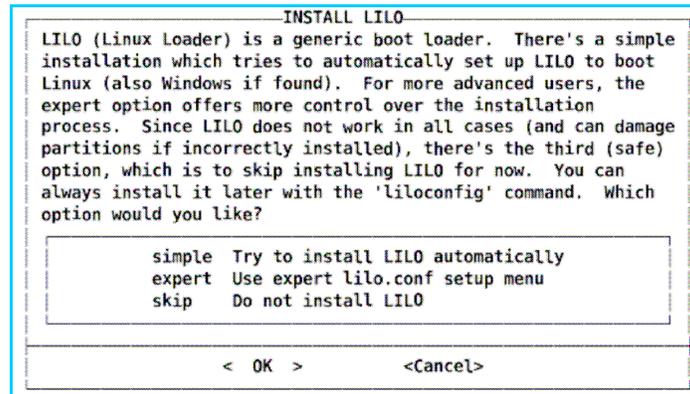
En este archivo se encuentra: la resolución de pantalla en la que se muestra el programa, las particiones configuradas como bootables y el tiempo de espera. LILO por default maneja opciones predefinidas como la resolución de pantalla.

### Configuración manual

Para esto hay que abrir el archivo lilo.conf con cualquier editor de texto, el archivo se encuentra en el directorio /etc. Las nuevas particiones booteable se agregan después de la línea # End LILO global section. Para declarar una nueva partición hay que llenar un pequeño formulario, que dependiendo del tipo de sistema cambian los datos.

### Configuración por Programa

Para el método de configuración por programa se realiza por medio del comando liloconfig.



I.2.5 Menú de instalación de LILO

Liloconfig es un comando propio de LILO que automatiza el proceso de configuración. El programa proporciona 3 opciones (imagen I.2.5):

- **Simple.** Está opción es para usuarios novatos, dado que no se hace mucha configuración manual sino que se deja al programa hacerla.
- **Expert.** Está opción es para usuarios avanzados dado que al configuración la realiza uno mismo.
- **Skip.** Salir del programa.

Para configurar LILO se utilizara la opción de Expert que es la que más libertades da, está opción no es recomendable para usuarios novatos.

### I.2.7 Apagado del sistema

Para apagar un sistema se puede utilizar diversos comandos:

COMANDOS	DESCRIPCIÓN
<b>SHUTDOWN</b>	Esté comando apagar el equipo
<b>INIT</b>	Esté comando trabaja con los runlevel de sistema, ejecutando el comando init # cambia el runlevel.
<b>HALT</b>	Se utiliza para apagar el equipo sin que este se vuelva a reiniciar.
<b>REBOOT</b>	Se utiliza para reinicializar el equipo.
<b>POWEROFF</b>	El comando poweroff se utiliza para apagar el equipo sin que este se vuelva a reiniciar

### I.2.8 Administración de usuarios y grupos

La administración de usuarios y grupos facilita la tarea a los administradores, dado que se tiene un control total los accesos al sistema, los programas en ejecución y se limitan los privilegios. Para la administración de usuario se necesita ser un usuario con privilegios de root, o que pueda modificar cualquier aspecto del sistema.

Cuando se agrega un usuario este debe pertenecer a un grupo, esto para un manejo más fácilmente y así ya no se configuran permisos por cada usuario sino por grupo. El tener grupos también facilita establecer jerarquías facilitando establecer permisos y privilegios, no es lo mismo un usuario root o administrador a un usuario mysql.

La definición de grupos y usuarios es la siguiente:

**Usuarios.** Personas con permiso de entrar al sistema y utilizar sus recursos, cada usuario tiene un home o directorio propio y pertenece a uno o más grupos.

**Grupo.** Es un conjunto de usuarios, el grupo contiene los privilegios y permisos que se heredaran a los usuarios que permanezcan a ese grupo.

Los comandos utilizados para la administración de grupos y usuarios son:

### **Administración de Usuarios**

- ◆ **USERADD.** Este comando es más complejo que adduser dado que todas las opciones se tienen que dar por medio de la línea de comando.
- ◆ **USERDEL.** Se utiliza para borrar usuarios.
- ◆ **ADDUSER.** Permite agregar un usuario mediante preguntas. El comando va preguntando cada opción del usuario.

NOTA: PARA DEJAR LAS OPCIONES POR DEFAULT O QUE PROPORCIONA EL SISTEMA SÓLO HAY QUE PRESIONAR ENTER EN LA PREGUNTA.

### **Administración de grupos**

- ◆ **GROUPADD.** Este comando se utiliza para agregar grupos al sistema.
- ◆ **GROUPDEL.** Borra un grupo específico, si un usuario de ese grupo está activo manda error.

### **Otros comandos de administración**

COMANDOS	DESCRIPCIÓN
<b>PASSWD</b>	Permite a un usuario cambiar su contraseña.
<b>LOGIN</b>	Permite que un usuario que ya está dentro del sistema se cambie a otra cuenta.
<b>SU</b>	Permite que un usuario que ya está dentro del sistema se cambie a otra cuenta.
<b>SG</b>	Permite que un usuario que ya está dentro del sistema se cambie del grupo que pertenece a otro.

### **Archivos de administración de cuentas de usuarios**

El control de las cuentas de usuarios y de los grupos se lleva a cabo utilizando tres archivos principales: **passwd**, **shadow** y **group**.

1) **Passwd.** Contiene la información de los usuarios existentes en el sistema.

2) **Shadow.** El sistema deposita la clave encriptada del usuario en `/etc/shadow` al que sólo tiene acceso el root. En el lugar de la clave encriptada, en el archivo `/etc/passwd` se coloca una "x" :

3) **Group.** Este archivo contiene la información de los grupos existentes en el sistema y los usuarios asignados a cada grupo.

# I.3 EDITOR PARA LA CREACIÓN DE PÁGINAS WEB

**OBJETIVO:** Elaborar páginas WWW, apoyados con HTML

HTML (HyperText Markup Language) es un lenguaje muy sencillo que permite escribir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, permitiendo agregar elementos multimedia (gráficos, sonido...).

Se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, listas, definiciones, citas, etc) así como los diferentes efectos que se quieren dar (especificar la posición en el documento donde se debe poner cursiva, negrita, o un elemento multimedia determinado) y dejar la presentación final de dicho hipertexto se realice por un programa especializado, por ejemplo un navegador<sup>viii</sup>.

Los archivos de HTML sólo son archivos de texto plano<sup>ix</sup>, lo que los identifica son las extensiones que pueden tener, estas pueden ser .html o .htm, que son las que le dicen al servidor que son páginas Web y que deben ser interpretadas por el navegador.

### ***1.3.1 Documento HTML***

#### ***Etiquetas***

HTML utiliza etiquetas o tags para dar la estructura a una página. Las etiquetas están formadas por determinados códigos encerrados entre los signos < y >, y con la barra / cuando se trata de la segunda etiqueta de un efecto (la de cierre). Las etiquetas le indican al navegador como interpretar la información, ya sea que lo despliegue como un texto normal, liga o imagen, además de que proporciona el formato ya sea cursiva, resaltado, etc.

Los efectos se especifican rodeando el texto entre dos etiquetas, que definen el efecto o unidad lógica que se desea.

Ciertas directivas sólo se ponen una vez en el lugar del texto donde requiere que aparezca el efecto concreto. A veces es necesario ofrecer datos adicionales en una directiva.

#### ***Estructura de un documento HTML***

Un documento HTML comienza con la etiqueta <html>, y termina con </html>. Dentro del documento (entre las etiquetas de principio y fin de HTML), hay dos zonas diferenciadas:

- 1) **Encabezado.** Delimitado por <head> y </head>, que sirve para definir diversos valores que se utilizarán en todo el documento.
- 2) **Cuerpo.** Delimitado por <body> y </body>, donde reside la información del documento.

La etiqueta más utilizada en el encabezado es <title>, que permite especificar el título de un documento HTML. Este título no forma parte del documento en sí, sino que suele servir como título de la ventana del programa que despliega el documento.

El cuerpo de un documento HTML contiene el texto que, con la presentación y los efectos que se decidan. Dichos efectos se especifican exclusivamente a través de etiquetas. Esto quiere decir que los espacios, tabulaciones y retornos de carro que se introduzcan en el archivo fuente no tienen ningún efecto a la hora de la presentación final del documento.

---

<sup>viii</sup> Programa que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet

<sup>ix</sup> Archivo que están compuestos únicamente por texto sin formato, sólo caracteres.

## I.3 EDITOR PARA LA CREACIÓN DE PÁGINAS WEB

La estructura básica de un documento HTML queda de la forma siguiente:

Estructura básica de un documento HTML	
<HTML>	
<head>	
<title>Título</title>	
</head>	
<body>	Texto del documento, menciones a gráficos, enlaces, etc.
</body>	
</HTML>	

### I.3.2 Etiquetas

Las etiquetas, son las que dan la estructura a una página, a continuación se muestra una lista de las etiquetas HTML más utilizadas:

#### **Etiquetas que delimitan el documento HTML**

Estas etiquetas indican donde inicia y termina un documento HTML:

ETIQUETA	DESCRIPCIÓN	SINTAXIS
<b>HTML</b>	Define el inicio del documento HTML, le indica al navegador que todo lo que sigue debe tratarlo como una serie de códigos HTML.	<HTML>Documento</HTML>
<b>BODY</b>	Se emplea para definir el inicio del cuerpo del texto de la página.	<BODY [opciones]></BODY>
<b>HEAD</b>	Contiene información sobre el documento, como el título, palabras clave que pueden ser de utilidad para motores de búsqueda, y otros datos que no se consideran parte del contenido del documento.	<HEAD>información o etiquetas</HEAD>
<b>TITLE</b>	Da el título a una página. El título suele ser visualizado por los navegadores en la cabecera de página.	<TITLE>Titulo de la página</TITLE>

#### **Modificadores de texto**

Estas etiquetas se utilizan para dar formato al texto y dar una mejor presentación:

ETIQUETA	DESCRIPCIÓN	SINTAXIS
<b>H</b>	Da formato a un título, se maneja por medio de números siendo 1 el mayor o principal. Mientras más pequeño el número menos importancia (y tamaño de letra) tendrá el título.	<H# [opciones]>Titulo</H#>
<b>P</b>	Forma y separa párrafos.	<P [opciones]></P>
<b>FONT</b>	Da formato al texto dentro de la página.	<FONT [opciones]>Texto</FONT>
<b>DIV</b>	Crea párrafos y sirve para dar formato al texto.	<DIV [opciones]></DIV>
<b>PRE</b>	Texto preformateado, respeta los espacios y retornos de carro que tenga el texto fuente.	<PRE>Texto</PRE>
<b>ADDRESS</b>	Indica información relacionada con cualquier tipo de dirección (e-mail, teléfono, etc.).	<ADDRESS>Texto</ADDRESS>
<b>HR</b>	Dibuja una línea que separa el texto.	<HR [opciones]>
<b>B</b>	Formatea el texto en negritas.	<b>Texto</b>
<b>I</b>	Formatea el texto en cursiva.	<i>Texto</i>
<b>U</b>	Se emplea para subrayar un texto.	<u>Texto</u>
<b>TT</b>	Formatea el texto para parecer letra de máquina de escribir.	<tt>Texto</tt>

### I.3.3 Listas

Se pueden emplear tres tipos distintos de listas en HTML, cada una con una presentación diferente: no numeradas, numeradas y listas de definiciones. Las listas se pueden anidar, es decir, en el lugar donde debería ir uno de los términos de la lista se pone una nueva lista, que por supuesto no tiene por que ser del mismo tipo.

En las listas siempre se tiene una etiqueta de inicio y fin, dependiendo de la etiqueta que se utilice sera el tipo de lista que se desplegara:

- `<ul></ul>` para listas no numeradas
- `<ol></ol>` para listas numeradas
- `<dt></dt>` para listas de definición

EJEMPLO DE LISTAS		
LISTA NO NUMERADA	LISTA NUMERADA	LISTA DE DEFINICIÓN
<pre>&lt;ul&gt; &lt;li&gt; Texto 1 &lt;li&gt; Texto 2 &lt;li&gt; Texto 3 &lt;/ul&gt;</pre>	<pre>&lt;ol&gt; &lt;li&gt; Texto 1 &lt;li&gt; Texto 2 &lt;li&gt; Texto 3 &lt;/ol&gt;</pre>	<pre>&lt;dl&gt; &lt;dt&gt;Perro (&lt;i&gt;n. masc.&lt;/i&gt;) &lt;dd&gt;Animal de cuatro patas que ladra. &lt;dt&gt;Gato (&lt;i&gt;n. masc.&lt;/i&gt;) &lt;dd&gt;Animal de cuatro patas que maúlla y se lleva muy mal con el perro. &lt;dt&gt;Pescadilla (&lt;i&gt;n. fem.&lt;/i&gt;) &lt;dd&gt;Animal que vive en el mar y está recubierto de escamas. &lt;/dl&gt;</pre>

### I.3.4 Tablas

Las tablas se utilizan para dar una mejor presentación a la página. Se especificarán siempre por filas; es decir, primero se escribirá la fila 1, después la fila 2, etc. La etiqueta que se utiliza para delimitar una tabla es `<TABLE>...</TABLE>`.

Cada fila se especifica con etiqueta `<TR>...</TR>` y, dentro de ella, cada celda se especifica con la etiqueta `<TD>...</TD>`. Se puede sustituir `<TD>` por `<TH>` cuando se quiera mostrar una celda con un texto sobresaltado.

La presencia de bordes en la tabla se especifica con el atributo `border` en la etiqueta `<TABLE>`. Con ello, se logrará un borde de dimensión la unidad; si se desea hacer el borde más ancho se debe dar un valor numérico al atributo: `border=anchura`

CÓDIGO	RESULTADO														
<pre>&lt;TABLE border&gt; &lt;CAPTION&gt;Ejemplo &lt;/CAPTION&gt; &lt;TR&gt;&lt;TH rowspan=2&gt;&lt;/TH&gt;&lt;TH colspan=2&gt;Valores&lt;/TH&gt;&lt;TH rowspan=2&gt;Precio&lt;/TH&gt;&lt;/TR&gt; &lt;TR&gt;&lt;TH&gt;Peso&lt;/TH&gt;&lt;TH&gt;Potencia&lt;/TH&gt;&lt;/TR&gt; &lt;TR ALIGN=CENTER&gt;&lt;TH&gt;Modelo 1&lt;/TH&gt;&lt;TD&gt;890&lt;/TD&gt;&lt;TD&gt;70&lt;/TD&gt;&lt;TD&gt;995&lt;/TD&gt;&lt;/TR&gt; &lt;TR ALIGN=CENTER&gt;&lt;TH&gt;Modelo 2&lt;/TH&gt;&lt;TD&gt;1200&lt;/TD&gt;&lt;TD&gt;110&lt;/TD&gt;&lt;TD&gt;2245&lt;/TD&gt; &lt;/TABLE&gt;</pre>	<table border="1"> <thead> <tr> <th rowspan="2"></th> <th colspan="2">Valores</th> <th rowspan="2">Precio</th> </tr> <tr> <th>Peso</th> <th>Potencia</th> </tr> </thead> <tbody> <tr> <td>Modelo 1</td> <td>890</td> <td>70</td> <td>995</td> </tr> <tr> <td>Modelo 2</td> <td>1200</td> <td>110</td> <td>2245</td> </tr> </tbody> </table>		Valores		Precio	Peso	Potencia	Modelo 1	890	70	995	Modelo 2	1200	110	2245
	Valores		Precio												
	Peso	Potencia													
Modelo 1	890	70	995												
Modelo 2	1200	110	2245												

### I.3.5 Formularios

Los formularios es como el usuario interactúa con las páginas, llenando una serie de campos, enviando esos datos a un programa de tratamiento, denominado genéricamente CGI

HTML utiliza la etiqueta <FORM> ... </FORM> para especificar un formulario. Dentro de ella se especifican todos los campos que intervienen en el formulario:

ETIQUETA	DESCRIPCIÓN	SINTAXIS
<b>FORM</b>	Especifica el tratamiento que se realizará con los campos introducidos por el usuario.	<form [opciones]></form>
<b>INPUT</b>	Permite especificar algunas clases de campos de entrada, dependiendo de los atributos asociados.	<input [opciones]>
<b>SELECT</b>	Define listas de opciones dentro de un formulario.	<select [opciones]> <option></option value="">Texto </select>
<b>TEXTAREA</b>	Permite crear una zona de texto especificando el número de filas (atributo ROWS) y de columnas (atributo COLS) de la ventana. Si se desea especificar un texto, se hará entre la directiva de apertura y la de cierre.	

### I.3.6 Enlaces

También llamados ligas, son redireccionamientos a una página Web ya sea dentro del mismo sitio o un sitio externo. Se define por la etiqueta <a>, con las que se va a poder formar hipervínculos<sup>x</sup>.

**Sintaxis:** <a href="dirección" >Texto que indica la liga</a>

Esta etiqueta llevara a la parte de la página donde se encuentre el ancla, indicada de la siguiente manera:

**<a name = "dirección">**

Para que esto suceda la dirección debe estar escrita de forma idéntica. Se puede agregar tantas anclas como se desee.

### I.3.7 Imágenes

La etiqueta que se utiliza para insertar imágenes en texto de las paginas es <img>.

**Sintaxis:** 

### I.3.8 Frame

Se emplean para poder visualizar varios documento en el mismo navegador. Con los frames se puede dividir la pantalla de visualización en varias zonas donde cada una puede visualizar un documento diferente.

Se utilizan dos etiquetas para ello: FRAMESET y FRAME, y un atributo en las etiquetas A y FORM, TARGET. En un frame HTML se define el aspecto de las zonas de visualización y los documentos que se mostrarán en cada una de ellas.

#### FRAMESET.

Es la etiqueta que divide la zona en subzonas, bien verticalmente, bien horizontalmente.

**Sintaxis:** <FRAMESET COLS="\*" ROWS="\*">

#### FRAME.

Se utiliza para personalizar cada una de las zonas definidas mediante FRAMESET.

**Sintaxis:** <FRAME NAME="indice" SRC="indice.html">

#### TARGET

Indicar a un enlace en qué zona se debe visualizar. Para ello, el atributo tomara como valor el NAME de un marco del panel u otro valor preestablecido.

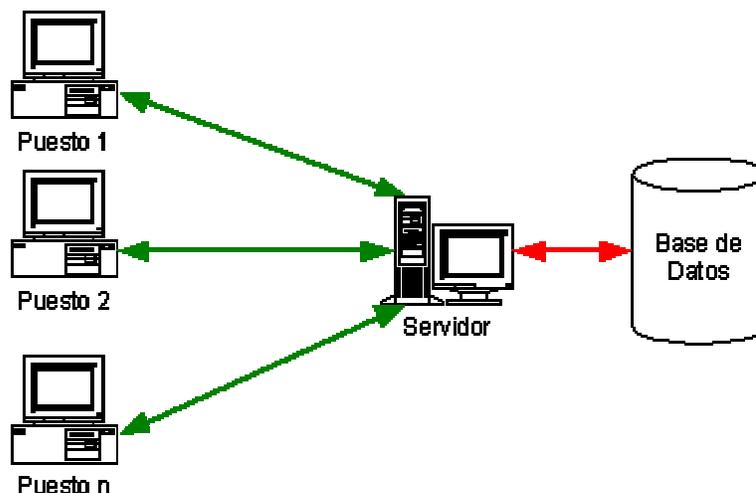
<sup>x</sup> Un simplemente enlace (también llamado vínculo o hipervínculo), es una referencia en un documento de hipertexto a otro documento o recurso.

## I.4 ADMINISTRACIÓN DE SERVIDORES WWW CON LINUX

**OBJETIVO:** Identificar el procedimiento para instalar, configurar y administrar su propio servidor de WWW en un servidor de plataforma LINUX.

Un servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor.

El servidor Web trabaja con la arquitectura cliente-servidor (imagen I.4.1), que se refiere a dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, empleado en un servidor en que un usuario hace una solicitud al servidor y éste devuelve una página en HTML.



I.4.1 Arquitectura Cliente-Servidor

Antes de instalar el servidor se tiene que tomar algunas consideraciones como:

- Función del servidor
- Experiencia de los administradores
- Plataformas disponibles
- Número de conexiones concurrentes
- Número de transacciones por segundo
- Costo computacional por transacción
- Proyección del crecimiento esperado
- Soporte para la tecnología utilizada para el desarrollo
- Análisis del retorno de inversión

Este estudio debe realizarse antes de tomar cualquier decisión, ya que los resultados podrían afectar el servicio proporcionado.

### I.4.1 Servidor HTTP Apache

El servidor HTTP Apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual. Inició su desarrollo en 1995, basado en el código de NCSA HTTPd 1.3<sup>xi</sup>, pero más tarde fue reescrito por completo. Su nombre se debe a que

<sup>xi</sup> Servidor Web desarrollado originalmente en el National Center for Supercomputing Applications por Robert McCool y una lista de colaboradores. Suspendido en 1998, el código sobrevivió gracias al Proyecto Apache, pero el código de NCSA se ha ido reescribiendo progresivamente en versiones de Apache.

## I.4 ADMINISTRACIÓN DE SERVIDORES WWW CON LINUX

---

originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor parcheado), de donde toma su nombre: a patchy server → Apache Server.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

Las características principales de Apache son:

- Robusto y soporta un gran número de transacciones
- Multisistema
- Tiene un nivel de seguridad alto, maneja proxy y soporta scripting de diferentes lenguajes
- Se puede obtener el código fuente, trabaja con SSL y es libre y gratuito.

### I.4.2 Instalación

La versión que se utiliza es la 2.0, la más actual hasta ese momento. La instalación se realiza de la manera siguiente:

- 1) Para iniciar la instalación de Apache hay que tener el código fuente (source), este se puede descargar de la página principal de Apache ([www.Apache.org](http://www.Apache.org)).
- 2) El archivo del código fuente es un *.tar.gz*, un archivo comprimido nativo de LINUX, para descomprimir el código fuente se utiliza el comando *tar*. ***tar -zxvf httpd-2.0.47.tar.gz***
- 3) Entrar en la carpeta que se creó al descomprimir el *.tar.gz*. ***cd httpd-2.0.47***
- 4) Se ejecuta el programa *configure* que verifica que existan las librerías y requerimientos necesarios para la compilación de Apache, y se especifica la ruta donde se va a instalar con la opción *prefix*.

***./configure --prefix=/usr/local/Apache2/***

- 5) A continuación se ejecutan los comandos *make* y *make install*. El comando *make* compila o genera los binarios de la aplicación y *make install* copia los binarios en las rutas requeridas para su funcionamiento.  
***make***  
***make install***

### I.4.3 Iniciando el servidor Apache

Para iniciar el servidor Apache se utiliza el comando *apachectl*, que se encuentra dentro del directorio *bin* de la ruta donde se instaló (*prefix*).

**NOTA: SE DENOMINA PREFIX A LA DIRECCIÓN DONDE SE HA INSTALADO APACHE, DEFINIDO EN EL PASO 4 DE LA INSTALACIÓN, EN ESTE CASO /USR/LOCAL/APACHE2/**

Para iniciar el servidor se teclea en la línea de comandos: ***./apachectl start***.

Para comprobar que está corriendo hay que teclear la siguiente línea: *ps -fea | grep -a httpd*, que comprueba que el proceso *httpd*, que es el Apache, está en marcha, si aparece es que se inició el servidor Apache.

Otra forma de verificar el funcionamiento de Apache es abrir un navegador y teclear *127.0.0.1* o *localhost*, si aparece la pantalla de bienvenida de Apache es que está funcionando.

#### **APACHECTL.**

Inicia/Detiene/Reinicia el servidor Apache.

#### **Sintaxis.**

*Apachectl* [opciones]

#### **Opciones.**

**Start** – inicia el servidor Apache

**Stop** – detiene el servidor Apache

**Restart** – reinicia el servidor Apache, se emplea cuando se hacen modificaciones a la configuración, y no se desea detener el servicio.

### **I.4.4 Directivas**

Las directivas son reglas que tiene que seguir Apache para funcionar, en ellas se puede definir desde el puerto de conexión hasta crear host virtuales. Apache es administrado por más de 200 directivas, en LINUX el administrador controla que directivas estarán disponibles de acuerdo a los módulos con los que se ha compilado Apache.

Las directivas de Apache se pueden dividir en tres grupos:

- **Global Environment** – Administra las directivas generales de operación para Apache.
- **Main Server** – administra las directivas del servidor principal o estándar de Apache.
- **Virtual Servers** – administra las directivas donde los mismos procesos de Apache soportan diversas Ip's o nombres de dominio.

Todas las directivas se encuentran en el archivo httpd.conf encargado de la configuración del servidor Apache. Se encuentra en el directorio prefix/conf/.

**NOTA: CUANDO SE VAYA A MODIFICAR ALGO DE ALGÚN ARCHIVO DE CONFIGURACIÓN SE HAGA UN RESPALDO DE LOS ARCHIVOS.**

### **Configuración**

Este grupo de directivas permite configurar la forma de respuesta del servidor, cambiando desde el puerto, si permite servidores virtuales o permite que archivos PHP sean interpretados.

### **Bitácoras**

El servidor utiliza archivos en texto plano para registrar los eventos, Apache maneja dos archivos principales:

- ◆ Access\_log. El servidor registrará los accesos a la página web. Se guardará el acceso, el agente y el referer.
- ◆ Error\_log. El servidor registrará los mensajes de error.

### **Directory**

Permite aplicar otras directivas de forma específica a todos los recursos dentro de la ruta definida por "dir".

### **CGI**

Los CGI son scripts de diferentes lenguajes que se utilizan sobre todo para manejo de datos. Se definen "Common Gateway Interface", y se utilizan para generar páginas dinámicas.

### **I.4.5 Servidores virtuales**

Los servidores virtuales permiten que un mismo servidor Apache pueda responder a diferentes solicitudes, con lo cual es posible mantener múltiples sitios Web con diferentes nombres y/o direcciones IPs.

En la configuración se definen como:

```
<VirtualHostservername>  
    ...Opciones...  
</VirtualHost>
```

Un servidor virtual trabajaría como si fueran otro servidor, por lo tanto hay que definirle sus propias opciones como son:

- **ServerAdmin** root@web.ejemplo.com
- **ServerName** www.pagina.com
- **DocumentRoot** /home/pagina/htdocs/

Cuando se defina un servidor virtual la primera línea debe ser:

```
NameVirtualHost direccionip:puerto
```

Esta línea define la dirección IP y puerto se va a acceder.

Ejemplo de servidor virtual
NameVirtualHost 192.168.40.5 <VirtualHost 192.168.40.5> ServerAdmin root@web.ejemplo.com ServerName www.micasa.com DocumentRoot /home/micasa/htdocs/ </VirtualHost>

### ***1.4.6 Control de accesos***

Apache trae su propio control de accesos a recursos protegidos, mediante ACL (Access Control List), con el cual es posible determinar las direcciones IPs y los usuarios que tendrán acceso.

Esta configuración se realiza por medio de directivas que van dentro de Directory:

- **Order.** Define el orden en que allow o deny serán implementadas
- **Allow.** Define los hosts que tendrán acceso al recurso.
- **Deny.** Define los hosts que no tendrán acceso al recurso.

**Sintaxis.**

***Allow from all|host|env=env-variable [host|env=env-variable]***

***Deny from all|host|env=env-variable [host|env=env-variable]***

También se pueden autenticar usuarios con contraseñas por medio de las directivas.

### ***1.4.7 Prueba al servidor Apache***

Para verificar como funcionan algunas de las directivas se realiza un benchmarking<sup>xii</sup> (prueba de rendimiento):

Para realizar el benchmarking se utilizaran dos ventanas de consolas y dos programas. En cada consola se va a realizar una operación.

#### **RESULTADO**

Este benchmarking debe proporcionar como resultado el número de conexiones que el servidor puede soportar, además de comprobar que la configuración de las configuración es correcta.

---

<sup>xii</sup> Un proceso sistemático y continuo para evaluar los productos, servicios y procesos de trabajo.

# I.5 PROGRAMACIÓN EN PHP

**OBJETIVO:** Proporcionar los conocimientos necesarios que permitan crear aplicaciones dinámicas e interactivas para el Web utilizando el lenguaje PHP.

PHP es un lenguaje de programación de scripts, sus siglas significan Hypertext Preprocessor. Su principal uso son los CGIs, pequeños scripts para realizar acciones en el servidor Web como manejo de datos, archivos, sesiones, obteniendo datos de bases de datos o de páginas Web, destaca su capacidad de ser embebido en el código HTML.

PHP actualmente forma parte de páginas Web dinámicas, y es de facilidad para el usuario ya que se ejecuta del lado del servidor. Cuando el cliente hace una petición al servidor para que le envíe una determinada página Web, además se le envían ciertas variables que permiten dar un formato a la página resultante. La página resultante es enviada al cliente y le llega a éste transformada, generalmente, en código HTML (aunque también es capaz de generar ficheros PDF, Flash, JPG,...).

PHP es un lenguaje manejado por intérprete esto quiere decir que va a ir ejecutando el código hasta que termine o encuentre un error. Para poder utilizar PHP en LINUX hay que instalar un servidor, en este caso es Apache, la instalación de Apache ya se vio en un módulo anterior, por lo que no se explicará aquí.

### I.5.1 Instalación de PHP

Lo primero que se tiene que tomar en cuenta es como se quiere instalar PHP, dado que hay dos formas:

- **CGI** - Disminuye el rendimiento ya que es el sistema operativo quien se encarga de gestionar todos los procesos derivados de la ejecución del script de PHP.
- **Módulo** - Es más rápido y eficiente ya que las ejecuciones del programa las realiza el servidor Web.

En este caso se va a instalar como módulo, por sus ventajas de rapidez y eficiencia, además de que se instala como CGI, se tendría que recompilar Apache.

La instalación de PHP como módulo es la siguiente:

- 1) Verificar que se tiene instalado Apache.
- 2) Descargar el código fuente (archivo tar.gz) de la página de PHP ([www.php.net](http://www.php.net)), es recomendable descargar la última versión de PHP, aquí se va a utilizar la versión 4.
- 3) Descomprimir el archivo tar.gz. ***tar zxvf PHP-4.4.0.tar.gz***
- 4) Entrar en el directorio creado al descomprimir el código fuente. ***cd php-4.4.0***
- 5) Se ejecuta el programa configure que verifica que existan las librerías y requerimientos necesarios para la compilación de PHP, y se especifica la ruta donde se va a instalar con la opción `prefix`, en este caso `/usr/local/php4`, la opción `--with-apxs2` indica donde está el comando `apxs`, que pertenece a Apache, esta opción sirve para indicar que PHP se va a instalar como módulo de Apache, y la opción `--with-mysql` indica que hay interacción entre PHP y mysql.

```
/configure --prefix=/usr/local/php4 --with-apxs2=/usr/local/apache2/bin/apxs --with-mysql
```

- 6) A continuación se ejecutan los comandos `make` y `make install`. El comando `make` compila o genera los binarios de la aplicación y `make install` copia los binarios en las rutas requeridas para su funcionamiento.
- 7) Antes de continuar, hay que verificar que en la carpeta `module` de Apache (`/usr/local/apache2/module`) exista el archivo `mod_so.c`, que es el que da soporte para los módulos de Apache, si este archivo no está es que ha ocurrido un error y se tiene que volver a instalar PHP.

## I.5 PROGRAMACIÓN EN PHP

- 8) Después de la instalación hay que copiar el archivo de configuración de PHP, *php.ini-dist*, que se encuentra en el directorio creado al descomprimir el *.tar.gz* (en la misma donde se realiza el *.configure*) al directorio *prefix/lib/* (prefix se definió en el paso 5, es el directorio donde se instaló PHP).

```
cp php.ini-dist/usr/local/php4/lib/php.ini
```

- 9) En el archivo de configuración de Apache (en este caso */usr/local/apache2/conf/*) hay que agregar las siguientes líneas para que reconozca las extensiones como archivos PHP y los envíe al intérprete:

```
AddTypeapplication/x-httpd-php.php.phtml
AddTypeapplication/x-httpd-php-source.phps
```

- 10) Hay que verificar que en ese mismo archivo exista la siguiente línea:

```
LoadModulephp4_module modules/libphp4.so
```

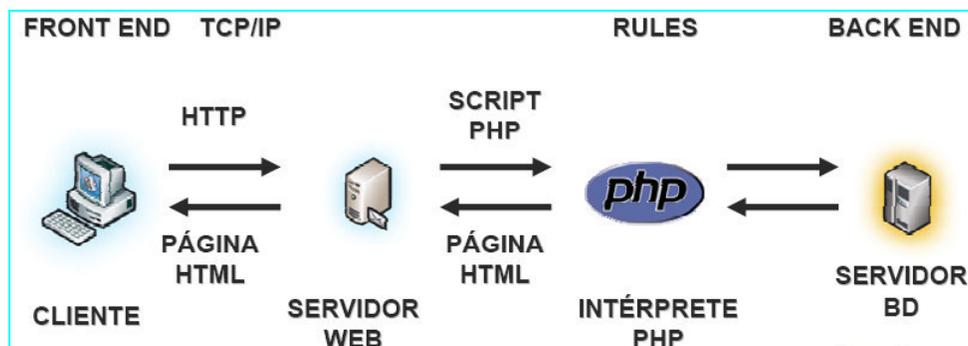
Esto indica que PHP se ha instalado como módulo.

- 11) Para verificar que PHP se está ejecutando con el servidor hay que hacer una prueba, esta se realiza por medio de un archivo php, en este caso el archivo se llamará *phpinfo.php*, para esto hay que seguir los siguientes pasos:

Algunas de las razones para instalar PHP como CGI son:

- Generar páginas de forma dinámica.
- Interacción con Bases de datos.
- Lectura y escritura de archivos.
- Foros de discusión.
- Procesamiento de formularios.
- Comercio electrónico.
- Motores de búsqueda.

La forma en que trabaja PHP es ayudando al servidor, por lo que se adecua a la arquitectura cliente/servidor (imagen I.5.1).



I.5.1 Arquitectura Cliente/Servidor

Esto es tomando los datos de las páginas HTML o PHP, puede generar páginas dinámicas, manejar formularios o ejecutar diversos scripts.

### I.5.2 Comentarios

PHP maneja 3 formas de introducir comentarios en el código:

- *//* - Permite comentarios de una sola línea
- */\* \*/* - Permite comentarios de múltiples líneas
- *#* - Permite comentarios de una sola línea, esto es por medio del estilo de LINUX

### 1.5.3 Variables

En PHP no hay una declaración de variables propiamente dichas, a diferencia de lenguajes como C o Visual Basic, por lo que se le denomina como débilmente tipado. Esto quiere decir que una misma variable puede almacenar datos de diferente naturaleza.

Las variables se expresan con un signo de '\$' al inicio, por ejemplo \$pesos. El nombre de una variable puede contener: letras, números y \_.

PHP soporta 5 tipos de datos básicos:

TIPO	DESCRIPCIÓN
<b>NUMÉRICOS</b>	Los <b>enteros</b> son números que varían entre -2 billones y 2 billones, se pueden representar en formato decimal, octal o hexadecimal. Los <b>flotantes</b> son números con punto decimal.
<b>CADENAS</b>	Una cadena está formada por cero o más caracteres encerrados entre "" ó ' '.
<b>ARREGLOS</b>	PHP maneja los arreglos utilizando un índice numérico, por ejemplo \$INTA[1]=1, especifica el elemento 1 del arreglo INTA. También maneja arreglos asociativos, esto quiere decir que en lugar de usar índices numéricos utiliza palabras como índices.
<b>OBJETOS</b>	Los objetos son estructuras que definen características propias (denominadas propiedades) y sus funcionalidades (denominadas métodos).

### 1.5.4 Constantes

Una constante es una variable que no puede ser modificada durante la ejecución del programa, su principal uso es definir valores que son muy utilizados y que necesiten actualizaciones frecuentes, así sólo se modifica una línea y no todo el programa y se tiene la seguridad de que el valor no puede ser modificado externamente. La sintaxis para definición una constante es:

**define("constante",valor);**

### 1.5.5 Operadores

PHP maneja 7 tipos de operadores, estos son:

OPERADOR	SIMBOLOS	DESCRIPCIÓN
<b>Aritméticos</b>	+, -, *, /, %.	Para realizar operaciones con números.
<b>De asignación</b>	=, +=, -=, *=, /=, %=, .=.	Para darle valores a las variables.
<b>De cadenas</b>	. (concatenación), .= (concatenación y asignación).	Para manipular cadenas.
<b>De incremento</b>	++, --.	Para incrementar el valor de variables numéricas.
<b>De comparación</b>	==, !=, ===(Idénticos y del mismo tipo), !== (diferentes y de diferente tipo), <, >, <=, >=.	Para comparar expresiones.
<b>A nivel de bit</b>	& (AND),   (OR), ^ (XOR), ~ (CA1), << (DESP. IZQ.), >> (DESP. DER.),	Realizar operaciones a nivel de bits.
<b>Lógicos</b>	&& (AND),    (OR), !, XOR	Se utilizan con expresiones que devuelven valores lógicos.

### 1.5.6 Estructuras de control

Las estructuras de control se utilizan para modificar el flujo de ejecución, esto es por medio de estructuras condicionales que permiten la ejecución de diferentes códigos, al escoger un camino o estructuras cíclicas que ejecuta un determinado número de veces un código.

### Estructuras Condicionales

Estas estructuras son de decisión, permiten escoger un camino u otro.

ESTRUCTURA	DESCRIPCIÓN	SINTAXIS
<b>IF</b>	Si la expresión es cierta realiza una acción.	<pre>if (expresión) {     Sentencias }</pre>
<b>IF-ELSE</b>	Si la expresión es cierta, realiza una acción si es falsa realiza otra acción.	<pre>if (expresión) {     sentencia } else {     sentencia }</pre>
<b>IF-ELSEIF</b>	Es una variación de los if anidados.	<pre>if (expresión){     Sentencia } elseif(expresion){     Sentencia }</pre>
<b>IF COMPACTO (OPERACIÓN TERNARIA)</b>	Esta clase de if se recomienda para condiciones sencillas dado que su sintaxis es complicada.	<pre>&lt;expresión1&gt; ? &lt;expresión2&gt; : expresi3&gt;</pre>
<b>SWITCH</b>	Esta estructura es selectiva, dependiendo del valor que se le proporcione puede escoger un caso dentro de sus opciones, en caso de proporcionarle una opción fuera del rango tiene un apartado denominado default para poner un mensaje de alerta o realizar alguna acción.	<pre>switch(\$variable){     case valor1:         sentencias;         break;      case valorN:         sentecias;         break;      default:         sentencias; }</pre>

### Estructuras Cíclicas

Estas estructuras se utilizan para repetir un cierto número de veces alguna acción, mientras se cumpla una condición.

ESTRUCTURA	DESCRIPCIÓN	SINTAXIS
<b>FOR</b>	Permite realizar un conjunto de instrucciones un determinado número de veces.	<pre>for(inicialización; condición;incremento){     sentencias; }</pre>
<b>FOREACH</b>	Se utiliza para recorrer las estructuras de tipo arreglo, obteniendo en cada iteración uno de sus elementos componentes. Puede manejar los dos tipos de arreglos de PHP, con índice numérico o asociativo.	<b>Forma Numérica</b> <pre>foreach(nombre_arregloas \$valor){     sentencias; }</pre>
		<b>Forma de arreglos asociativos</b> <pre>foreach(nombre_arregloas \$clave=&gt; \$valor){     sentencia s; }</pre>
<b>WHILE</b>	Permite repetirá alguna acción hasta que se cumpla la expresión.	<pre>While(expresión){     sentencias; }</pre>
<b>DO WHILE</b>	Igual que while, pero esta asegura que se va a efectuar al menos una ves las sentencias.	<pre>Do{     Sentencias; }while(expresión)</pre>

### I.5.7 Funciones

#### Definición

Las funciones son secciones separadas del cuerpo principal del programa, que tiene un propósito específico.

Sintaxis
<pre>Function nombre_funcion(parámetros){     sentencias;         returnvalor; }</pre>

A las funciones se les puede proporcionar valores por medio de 3 formas:

TIPO DE FUNCIÓN	SINTAXIS
<b>Por valor</b> – Cuando se mandan a llamar en el cuerpo principal se les envía el valor de la variable, no la variable en si, lo que significa que genera una copia de la variable original, esto no modifica el valor de la variable original.	<pre>Function iExponente(\$a){     return\$a*\$a; }</pre>
<b>Por referencia</b> – Se hace referencia a la variable, en lugar de generar una copia, si se modifica el valor se modifica también el valor de la variable original. Se especifica que es por referencia cuando se pone un símbolo “&” al frente de la función en la declaración.	<pre>functioniExponente(&amp;\$a){     return\$a*\$a; }</pre>
<b>Por defecto</b> – Se definen parámetros opcionales, este caso se utiliza cuando no es preciso el número de parámetros que se van a enviar pero si se requieren los valores de todas las variables, entonces se les asigna un valor por defecto.	<pre>functioniExponente(\$a,\$exponente=2){     return\$a*\$a; }</pre>

#### Funciones predefinidas

PHP maneja algunas funciones para el control de variables, estas se utilizan sobre todo para seguridad de datos, ya que validan desde si fueron iniciadas antes de ser utilizadas hasta para verificar el origen de sus valores hasta saber su tipo.

### I.5.8 Archivos

#### Incluir archivos

Una característica de PHP es permite incluir archivos por medio de la función **INCLUDE**, esto sirve por ejemplo cuando se tiene un encabezado y un pie de página que se repite, con esto solo se crea un archivo para cada uno, por ejemplo encabezado.ini y pie.ini, y se van agregando a los archivos de código.

#### Manejo de archivos

Con PHP se puede llevar un registro de las operaciones realizadas por medio de archivos. El manejo de archivos es muy similar al de C, se realizan las operaciones básicas, abrir, cerrar, leer y escribir.

#### NAVEGANDO EN EL ARCHIVO

Las siguientes funciones se utilizan para poder moverse en el archivo, ya sea al principio a una posición deseada o al final:

- **rewind(apuntador)** - Sitúa el apuntador de lectura/escritura al principio del archivo.
- **fseek(apuntador,desp[,desde\_pos])** - Desplaza al apuntador desp posiciones a partir de su posición actual, el tercer parámetro puede tomar los valores SEEK\_SET, SEEK\_CURy SEEK\_END.
- **ftell(apuntador)** - Devuelve la posición actual del apuntador.

### I.5.9 Formulario

PHP es capas de recoger la información generada a través de formularios HTML, por medio de los métodos POST o GET, que son la manera en que HTML manda la información. En la mayoría de los casos los servidores permiten utilizar

los datos como si fueran variables globales, esto representa un gran problema de seguridad dado que no se puede validar los datos de forma correcta y un usuario o una persona externa podría aprovechar este medio para dañar el sistema.

Para deshabilitar esta forma de pasos de valores, se modifica el archivo *php.ini*, en la directiva `register_globals` hay que ponerla en off y reiniciar Apache.

### **I.5.10 Control de Acceso**

PHP un control de acceso muy básico pero efectivo, esto es por medio de la validación de dos variables de servidor:

- `$_SERVER[PHP_AUTH_USER]` - Nombre de usuario introducido
- `$_SERVER[PHP_AUTH_PW]` - Contraseña introducida

**NOTA: ESTAS SOLO FUNCIONAN SI SE INSTALO PHP COMO MODULO DE APACHE.**

### **I.5.11 Sesiones**

PHP tiene dos formas de manejar sesiones: **COOKIES** o **SESSION**.

#### **Cookies**

Las cookies se manejan a través del arreglo asociativo `$_COOKIE[]` estos son archivos de texto que almacenan información con la estructura clave=valor su tamaño es pequeño, no mayor a 4Kb y se utilizan para almacenar poca información por periodos prolongados de tiempo.

Algunas de las ventajas para los administradores es que las cookies son almacenadas en los equipos cliente.

Para crear una cookie se utiliza la función **SETCOOKIE**:

#### **Sintaxis.**

`setcookie(stringnombre [,stringvalor] [,intcaducidad] [,stringruta] [,stringdominio] [,intseguro])`

Para eliminar una cookie se necesita el comando `setcookie("nombre_cookie")`.

#### **Session**

Se manejan a través del arreglo asociativo `$_SESSION[]`, una sesión es el tiempo que transcurre desde que el usuario se conecta a un servidor hasta que sale de la aplicación, cierra el navegador o permanece un lapso de tiempo sin interactuar con la aplicación. La variables de sesión se almacena en el servidor, cada una está vinculada a una única sesión/usuario a través de un identificador de sesión único.

Para iniciar sesión se utiliza la función **SESSION\_START()** y para eliminarla se utiliza **UNSET(\$\_SESSION['NOMBRE\_VARIABLE'])**.

Un manejo de cookies con sesión podría facilitar mucho la utilización de recursos y la seguridad.

# I.6 INTERACCIÓN DE WWW CON BASE DE DATOS

**OBJETIVO:** Conocer y desarrollar una aplicación de bases de datos que funcione a través del WWW, empleando herramientas de software libre.

MySQL es un sistema gestor de bases de datos basado en la mentalidad de código abierto. Es un sistema de gestión de bases de datos relacional, licenciado bajo la licencia GPL. La facilidad de uso, la velocidad y la flexibilidad en diferentes sistemas operativos hacen popular a MySQL.

Proporciona grandes ventajas en las tareas de administración, la protección de los datos, la gestión de usuarios, la optimización del rendimiento, la conectividad y las posibilidades que la programación de aplicaciones de acceso a datos.

Su diseño multihilo<sup>xiii</sup> le permite soportar una gran carga de forma muy eficiente. Algo que ha ayudado a su popularidad es que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

El software de bases de datos MySQL consiste de un sistema cliente/servidor que se compone de un servidor SQL multihilo, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación (APIs). Se puede obtener también como una biblioteca multihilo que se puede enlazar dentro de otras aplicaciones para obtener un producto más pequeño, más rápido, y más fácil de manejar.

MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. Surgió como un intento de conectar el gestor mSQL a las tablas propias de MySQL AB, usando sus propias rutinas a bajo nivel. Tras unas primeras pruebas, la empresa noto que mSQL no era lo bastante flexible para lo que necesitaban, por lo que tuvieron que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, con una interfaz totalmente compatible a mSQL.

No se tiene certeza de donde proviene el nombre de MySQL, por un lado dicen que sus librerías han llevado el prefijo 'my' durante los diez últimos años. Por otro lado, la hija de uno de los desarrolladores se llama My.

### ***1.6.1 Características de MySQL***

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- Bajo consumo para equipos con escasos recursos.
- Utilidades de administración fáciles de configurar e instalar.
- Probabilidad reducida de corromper los datos
- Bajo o nulo costo.

Un dato curioso de MySQL es que la empresa MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece.

### ***1.6.2 Instalación y configuración***

MySQL se puede instalar tanto en Windows como en LINUX. Esta creado de forma que pueda trabajar de forma óptima con Apache como servidor Web y con PHP como lenguaje de programación para páginas dinámicas, aunque también puede trabajar con otras aplicaciones.

---

<sup>xiii</sup> Capacidad del sistema de apoyar la interacción del usuario para más de una tarea.

## I.6 INTERACCIÓN DE WWW CON BASE DE DATOS

---

Para obtener la versión libre y más actual de MySQL hay que descargar el código fuente de la página de MySQL, [www.mysql.com](http://www.mysql.com). Al descargar MySQL se tiene un archivo con el nombre de MySQL y la versión con extensión *.tar.gz*, parecido al siguiente: ***mysql-4.1.11.tar.gz***

Los números 4.1.11 corresponden a la versión de MySQL.

### PASOS PARA LA INSTALACIÓN

1.- Descomprimir el archivo que se descargo. Para hacer esto se emplea el comando *tar* de la siguiente forma

```
tar -xvzf mysql-4.1.11.tar.gz
```

Al descomprimir los archivos se crea el un directorio que lleva el nombre del MySQL y la versión es decir se crea el directorio *mysql-4.1.11*

2.- Hay que cambiarse al directorio creado (*mysql-4.1.11*). ***cd mysql-4.1.11***

**NOTA: ANTES DE REALIZAR CUALQUIER ACCIÓN ES PREFERIBLE REVISAR EL MANUAL DE INSTALACIÓN LLAMADO INSTALL-SOURCE**

3.- Antes de continuar con la instalación hay que crear el usuario de la aplicación, denominado *mysql* además de un grupo con el mismo nombre, este usuario y grupo se emplean en el motor de base de datos para levantar el servidor de MySQL y realizar las tareas internas del servidor:

```
groupadd mysql  
useradd -g mysql mysql
```

4.- Se ejecuta el programa *configure*. Este programa verifica que existan las librerías y requerimientos necesarios para la compilación del MySQL, así como la especificación de parámetros adicionales de configuración, por ejemplo el parámetro *--prefix=* especifica el directorio en donde se instalaran los binarios y las bases de datos de MySQL.

```
./configure --prefix=/usr/local/mysql
```

El comando se tiene que escribir con un *./* para poder decirle al sistema que es un programa ejecutable.

5.- A continuación se utilizan los comandos *make* y *make install*. El comando *make* compilan o generan los binarios del MySQL, *make install* copia los binarios y aplicaciones en las rutas requeridas para su funcionamiento.

**Nota: Para la instalación de una aplicación, siempre se utiliza primero el comando *make* y después el comando *make install*.**

6.- Lo siguiente es copiar el archivo *my-medium.cnf*, que se encuentra en la carpeta *mysql-4.1.11* al directorio */etc* con el nombre *my.cnf*. ***cp support-files/my-medium.cnf /etc/my.cnf***

7.- Par continuar hay que cambiarse de directorio a donde se instalo MySQL. Ya estando hay se utiliza el comando *mysql\_install\_db* para generar las tablas de configuración.

```
cd /usr/local/mysql  
bin/mysql_install_db --user=mysql
```

8.- Para terminar la instalación hay que cambiar el dueño de algunos archivos y directorios, para esto se utiliza el comandos *chown* para cambiar el duelo y el comando *chgrp* para cambiar el grupo.

### I.6.3 Iniciando MySQL

Una vez que se completo la instalación es posible iniciar el servidor de MySQL, utilizando los siguientes pasos: Utilizando el comando `mysqld_safe`, este comando se encarga de levanta el socket que se encarga de las conexiones a las bases de datos. `bin/mysqld_safe --user=mysql &`

El parámetro `--user=mysql` indica que el servicio lo arranca con el usuario `mysql` y el `&` manda el proceso a segundo plano para que se mantenga funcionando aunque el usuario cierre su terminal.

**NOTA: UNO DE LOS ERRORES MÁS FRECUENTES AL ARRANCAR EL SERVIDOR MYSQL ES EN EL SOCKET, ESTE ARCHIVO ESTABLECE LAS CONEXIONES Y TRANSACCIONES DE MYSQL. ESTE ERROR PUEDE OCURRIR POR LOS PERMISOS DE CARPETAS, NO TENER UN ARCHIVO MY.CNF O ESTA MAL EJECUTADO O INSTALADO EL COMANDO MYSQLD\_SAFE.**

### I.6.4 Administración

#### Como administrar MySQL

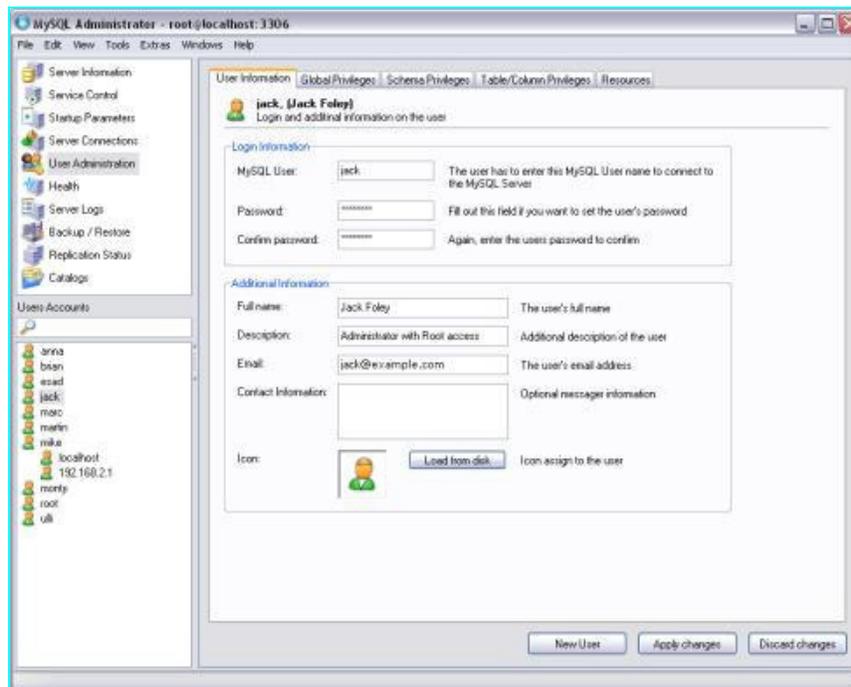
Se puede administrar MySQL por medio de la línea de comandos o utilizando una interfaz gráfica.

#### LÍNEA de COMANDOS

La línea de comandos es fácil de usar pero no es intuitiva, todo se maneja por medio de comandos en SQL estándar. Para acceder a ella se utiliza el comando `mysql`.

#### INTERFAZ GRÁFICA

Esta forma es más fácil y recomendada para novatos, ya que no se manejan comandos en SQL sino sólo se utilizan botones y se rellenan campos. Algunas herramientas de este tipo son: MySQL CC, mysqladmin y WebAdmin.



I.6.1 Interfaz gráfica de MySQL Administrator

Para manejar de forma correcta las bases de datos y tener acceso a ellas, el usuario debe tener privilegios o permisos sobre las bases. El administrador debe establecer niveles de usuarios y definirles permisos, ya que no es lo mismo un usuario de un servicio Web a un usuario de administración. Para realizar las conexiones también es necesario registrar las conexiones ya sean remota o local. Cada base de datos debe tener definidos sus permisos.

### I.6.5 Estructura de MySQL

Todo el sistema de permisos de acceso al servidor, a las bases de datos y tablas, se encuentra almacenado en una base de datos llamada `mysql`, la cual se componen de cinco tablas: `host`, `user`, `db`, `tables_priv`, `columns_priv`. Es ahí donde se guardaran todos los permisos y restricciones a los datos de las bases de datos.

MySQL crea por defecto al usuario `root` con todos los permisos posibles habilitados, éste es preferible utilizarlo como administrador o en su defecto se puede crear otro. Se vaya a utilizar o no la cuenta `root`, siempre hay que cambiarle la contraseña.

Para cambiar la contraseña de cualquier usuario se usa el comando `mysqladmin` con las opciones `-u` para especifica el usuario y `password` en el que se indica el nuevo password. **`mysqladmin -u root password nueva_contraseña`**

La estructura de MySQL es la siguiente:

Cuando se crea una base de datos MySQL crea un directorio con el nombre que se le ha asignado a la base. Dentro de este directorio, por cada tabla que se defina MySQL va ha crear tres archivos: `mitabla.ISD`, `mitabla.ISM`, `mitabla.frm`

El archivo con extensión `ISD`, es el contiene los datos de la tabla

El archivo con extensión `ISM` contiene información acerca de las claves y otro datos que MySQL utiliza para buscar datos en el fichero `ISD`.

El archivo con extensión `frm` contiene la estructura de la propia tabla.

#### TABLAS.

- ◆ **USER.** Contiene información sobre los usuarios.
- ◆ **HOST.** Contiene informa sobre que máquinas podrán acceder al sistema.
- ◆ **DB.** Proveen de un control individual de las bases de datos, tablas y columnas.

### I.6.6 Comandos de SQL para MySQL

Para ejecutar los comandos de SQL se emplea la línea de comandos de MySQL empleando el comando `mysql` para acceder a ella.

En esta interfaz en posible emplear comandos para ver el estado del servidor, las bases de datos existentes, elegir la base de datos con la que se va a trabajar, ver las tablas existentes dentro de la base de datos elegida o ver la estructura de la tabla.

#### Base de datos

El comando de SQL para crear bases de datos es **CREATE** con la opción `database`: **`create database nombrebased ;`**

Sólo es necesario especificar el nombre de la base de datos en lugar del parámetro `nombrebased`.

Para seleccionar esta base de datos basta con emplear el comando **USE** proporcionándole el nombre de la base que se quiere usar: **`use nombrebased ;`**

Para borrar una base de datos se utiliza el comando **DROP** con la opción `DATABASE` y el nombre de la base de datos que se quiere eliminar: **`DROP DATABASE nombrebase ;`**

#### Tablas

Para crear una tabla se utiliza el mismo comando **CREATE** sólo especificando que es una tabla, con la opción `table`, se le proporciona el nombre de la tabla y los campos y su tipo:

**`create table nombretabla (campo1 tipo, campo2 tipo, campoN tipoN, primary key (campo1), unique id (campo1));`**

## I.6 INTERACCIÓN DE WWW CON BASE DE DATOS

Este comando crea un tabla de nombre nombretabla con los campos campo1, campo2, campoN en donde se van dando los nombres que identifiquen el dato que será almacenado en ese campo. Los parámetros primary key y unique id especifican la llave primaria de la tabla y que esta no permita valores repetidos respectivamente. Para crear tablas referenciadas o con relación se utilizan los atributos:

**Primary Key** – Se utiliza para establecer una clave representativa de los registros de la tabla, entre sus características se encuentra:

- Que sea un numérico incremental automático que no tenga relevancia a nivel de información.
- Que este formada por campos que respondan a información y identifique al registro como registro único.

**InnoDB** - Tabla tipo transaccional. Puede trabajar con transacciones en MySQL, también tiene soporte para la definición de claves foráneas, por lo que se permite definir reglas o restricciones que garanticen la integridad referencial de los registros.

**Foreign Key** – Se utiliza para especificar claves foráneas. Para esto se necesita las siguientes condiciones:

- Crear ambas tablas del tipo InnoDB.
- Crear un índice en el campo que ha sido declarado clave foránea.

Las columnas correspondientes en la clave foránea y en la clave referenciada deben tener tipos de datos similares para que puedan ser comparadas sin la necesidad de hacer una conversión de tipos. El tamaño y el signo de los tipos enteros debe ser el mismo. En las columnas de tipo carácter, el tamaño no tiene que ser el mismo necesariamente.

### I.6.7 Interacción con PHP

La interacción con PHP para generar páginas dinámicas se realiza por medio de instrucciones a través del código. A continuación se muestran algunas instrucciones utilizadas:

COMANDO	DESCRIPCION	SINTAXIS
<b>MYSQL_CONNECT</b>	Establece una conexión a un servidor de MySQL. Todos los argumentos son optativos, y si no se especifican, los valores los da por defecto.	mysql_connect(string [hostname] , string [username] , string [password] );
<b>MYSQL_PCONNECT</b>	Establece una conexión a un servidor de MySQL, donde la conexión al servidor SQL no será cerrada cuando finalice la ejecución del script.	int mysql_pconnect(string [hostname] , string [username] , string [password] );
<b>MYSQL_CLOSE</b>	Cierra la conexión a la base de datos MySQL asociada al identificador de conexión especificado. No cierra una conexión persistente.	int mysql_close(int [link_identifier] );
<b>MYSQL_QUERY</b>	Ejecuta una consulta a la base de datos activa.	int mysql_query(string query, int [link_identifier] );
<b>MYSQL_RESULT</b>	El contenido de una celda del resultado.	int mysql_result(int result, int row, mixed field);
<b>MYSQL_DB_QUERY</b>	Ejecuta una consulta en una base de datos.	int mysql_db_query(string database, string query, int link_identifier);
<b>MYSQL_SELECT_DB</b>	Establece la base de datos activa en el servidor.	int mysql_select_db(string database_name, int [link_identifier] );
<b>MYSQL_AFFECTED_ROWS</b>	Devuelve el número de filas afectado en el último UPDATE o DELETE.	
<b>MYSQL_NUM_ROWS</b>	Devuelve el número de filas del resultado de una consulta.	int mysql_num_rows(string result);
<b>MYSQL_FETCH_ARRAY</b>	Almacenar los datos a través de índices numéricos de un arreglo.	array mysql_fetch_array(int result);

# I.7 INTRODUCCIÓN A LA SEGURIDAD EN CÓMPUTO

**OBJETIVO:** Reconocer la importancia de la seguridad e identificar los elementos que permitirán proteger el sistema y la información.

Garantizar que los recursos informáticos estén disponibles para cumplir sus propósitos, es decir, que no estén dañados o alterados por circunstancias o factores externos, es una definición útil para conocer lo que implica el concepto de seguridad informática.

En términos generales, la seguridad puede entenderse como aquellas reglas técnicas y/o actividades destinadas a prevenir, proteger y resguardar lo que es considerado como susceptible de robo, pérdida o daño, ya sea de manera personal, grupal o empresarial.

En este sentido, es la información el elemento principal a proteger, resguardar y recuperar dentro de las redes empresariales. La seguridad surge por tres necesidades: Confidencialidad, Integridad y Disponibilidad.

La **confidencialidad** previene la divulgación, ya sea intencional o por negligencia, del contenido de un mensaje, archivo, correo, etcétera.

La **integridad**, asegura que nadie pueda modificar datos confidenciales, específicamente por personas no autorizadas. También evita que, quienes tienen privilegios de realizar las modificaciones, lo hagan sin autorización. Y, finalmente, lo más importante es que los datos sean consistentes, es decir, que la información refleje la verdad.

La **disponibilidad** asegura que la información esté a la mano cuando sea necesario, en tiempo y espacio, para las personas que la requieren. En otras palabras, la disponibilidad garantiza que los sistemas funcionen cuando se necesitan.

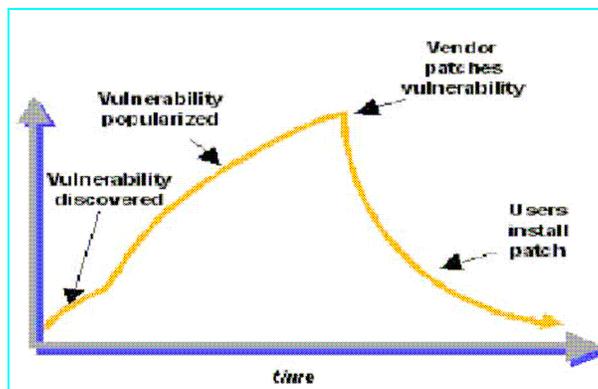
### I.7.1 Definiciones

#### **Vulnerabilidad**

La vulnerabilidad se define como la ausencia de una contramedida, o debilidad de la misma, de un sistema informático que permite que sus propiedades de sistema seguro sean violadas. La debilidad puede originarse en el diseño, la implementación o en los procedimientos para operar y administrar el sistema.

Una vulnerabilidad cumple con un ciclo de vida, como se ve en la imagen I.7.1:

- 1.- Se descubre la vulnerabilidad
- 2.- La vulnerabilidad se hace popular.
- 3.- la empresa encargada del sistema genera y proporciona un parche que corrige la vulnerabilidad.
- 4.- El usuario instala el parche y en este momento se termina el tiempo de vida de la vulnerabilidad.



I.7.1 Ciclo de vida de una vulnerabilidad.

### **Amenazas**

Una amenaza se puede definir como la circunstancia o evento que puede causar daño violando la confidencialidad, integridad o disponibilidad. La información se ve sometida a distintas amenazas que pueden clasificarse en intencionadas, no intencionadas y naturales.

Las **amenazas intencionadas** las ejercen usuarios no autorizados que acceden de forma indebida a los datos o información sensible. Los usuarios no autorizados pueden ser externos o pertenecientes a la propia organización y se pueden clasificar como curiosos o maliciosos.

### **Exploit**

Un exploit es un método concreto de usar un error de algún programa, también denominado bug, para entrar en un sistema informático. Generalmente, un exploit suele ser un programa que se aprovecha de algún error del sistema operativo, por ejemplo, para obtener los privilegios del administrador y así tener un control total sobre el sistema. Aunque un exploit no tiene que ser un programa, por definición es simplemente una forma de sacar provecho de un error en un programa (explotar un error).

Las **amenazas no intencionadas** provienen típicamente de empleados con poca formación o negligentes que no han seguido los pasos para proteger sus contraseñas, asegurar adecuadamente sus equipos, no tener instalado o actualizar con poca frecuencia un antivirus<sup>xiv</sup>. Las amenazas no intencionadas también implican a veces a los programadores o personal de proceso de datos cuando no se siguen las normas y procedimientos de seguridad establecidos.

Las **amenazas naturales** incluyen fallos de equipos y calamidades tales como incendios, inundaciones y terremotos que pueden causar la pérdida de equipos y datos. Las amenazas naturales suelen afectar a la disponibilidad de los recursos y de la información.

### **Políticas de seguridad**

El término política de seguridad se suele definir como el conjunto de requisitos definidos por los responsables directos o indirectos de un sistema que indica en términos generales qué está y qué no está permitido en el área de seguridad durante la operación general de dicho sistema

Una política de seguridad puede ser **prohibitiva**, si todo lo que no está expresamente permitido está denegado, o **permisiva**, si todo lo que no está expresamente prohibido está permitido.

Cualquier política ha de contemplar seis elementos claves en la seguridad de un sistema informático:

- **Disponibilidad:** Es necesario garantizar que los recursos del sistema se encontrarán disponibles cuando se necesitan, especialmente la información crítica.
- **Utilidad:** Los recursos del sistema y la información manejada en el mismo ha de ser útil para alguna función.
- **Integridad:** La información del sistema ha de estar disponible tal y como se almacenó por un agente autorizado.
- **Autenticidad:** El sistema ha de ser capaz de verificar la identidad de sus usuarios, y los usuarios la del sistema.
- **Confidencialidad:** La información sólo ha de estar disponible para agentes autorizados, especialmente su propietario.
- **Poseción:** Los propietarios de un sistema han de ser capaces de controlarlo en todo momento; perder este control en favor de un usuario malicioso compromete la seguridad del sistema hacia el resto de usuarios.

Para cubrir de forma adecuada los seis elementos, una política se suele dividir en puntos más concretos a veces llamados normativas.

Algunas instituciones se especializan en crear y regular políticas y estándares de seguridad, además de certificar a las instituciones o personas que las siguen e implementan:

### **Certificado a las instituciones**

- ISO 17799

---

<sup>xiv</sup> Programas cuya función es detectar y eliminar virus informáticos y otros programas maliciosos

### **Certificado a los individuos**

- CISSP
- GIAC
- SANS
- NGCERT

### **Ataques**

Los ataques se definen como una acción o acciones que tienen por objetivo el que cualquier parte de un sistema de información automatizado, deje de funcionar de acuerdo con su propósito definido. Los ataques se dividen principalmente en dos tipos *pasivos* y *activos*.

**Ataques pasivos:** El atacante no altera la comunicación, sino que únicamente la escucha o monitoriza, para obtener información que está siendo transmitida. Sus objetivos son la interceptación de datos y el análisis de tráfico, una técnica más sutil para obtener información de la comunicación.

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.

**Ataques activos:** Estos ataques implican algún tipo de modificación del flujo de datos transmitido o la creación de un falso flujo de datos, pudiendo subdividirse en cuatro categorías; **Suplantación de identidad, Reactuación, Modificación de mensajes, Degradación fraudulenta del servicio.**

### **Análisis de riesgo**

Es el proceso de identificación y evaluación del riesgo a sufrir un ataque y perder datos, tiempo y horas de trabajo, comparándolo con el costo que significaría la prevención de este suceso. El análisis no sólo establece un nivel adecuado de seguridad, sino que permite conocer mejor el sistema que se va a proteger.

## **I.7.2 Herramientas de escaneo o ataques**

### **TCPDUMP.**

Permite monitorear tráfico de red en tiempo real. Están disponibles una variedad de formatos de salida y puedes filtrar la salida para buscar sólo un tipo particular de tráfico. Permite examinar todas las conversaciones entre el cliente y el servidor, incluyendo mensajes de broadcast<sup>xv</sup>, SMB y NMB. Mientras que sus capacidades de detección de errores están principalmente a nivel de capa de red (OSI<sup>xvi</sup>).

### **ETHERREAL.**

Es un analizador de protocolos, utilizado para solucionar problemas de red, análisis, desarrollo de software y protocolos.

Ethereal está desarrollado bajo la licencia open source<sup>xvii</sup>, y se ejecuta sobre la mayoría de sistemas operativos UNIX y compatibles con UNIX, incluyendo LINUX, Solaris, FreeBSD, NetBSD, OpenBSD, y Mac OS X, así como en Microsoft Windows.

### **HPING.**

Una herramienta muy potente, basada en línea de comando y orientada a TCP/IP, ICMP y UDP, una especie de laboratorio para redes, capaz de ensamblar y enviar paquetes al igual que si se tratará del comando ping, a diferencia de este último que ofrece mayor control y compatibilidad con muchos protocolos.

Con hping se puede realizar pruebas a firewall<sup>xviii</sup>, escaneo de puertos de forma avanzada, traceroute<sup>xix</sup> y monitorización TCP/IP. Soportado por GNU/LINUX, FreeBSD, NetBSD, OpenBSD, Solaris y Mac OS X.

---

<sup>xv</sup> Mensaje a todos los dispositivos de una red

<sup>xvi</sup> Open Systems Interconnection, propuesto por la Organización Internacional para la Estandarización (ISO), se utiliza para estandarizar la interconexión de sistemas abiertos.

<sup>xvii</sup> Término por el que se conoce al software distribuido y desarrollado en una determinada forma.

### **TELNET.**

Telnet es el nombre de un protocolo (y del programa que implementa el cliente) que sirve para acceder mediante una red a otra máquina, para manejarla como si se estuviera sentado delante de ella. Para que la conexión funcione, como en todos los servicios de Internet, la máquina a la que se accedía debe tener un programa especial que reciba y gestione las conexiones. El puerto que se utiliza generalmente es el 23. Sólo sirve para acceder en modo terminal

### **NMAP.**

NMap" ó "Network Mapper" es una herramienta GPL de exploración de red y escaner de seguridad. Ha sido diseñado para permitir a administradores de sistemas el escaneo de grandes redes para determinar que servidores se encuentran activos y que servicios ofrecen. NMap es compatible con un gran número de técnicas de escaneo.

### **NESSUS.**

Potente escáner de redes de Software Libre que consta de dos partes: cliente y servidor; éstas pueden estar instaladas en las misma máquina por simplicidad. Nessus puede grabar informes donde hay enlaces que explican que tipo de vulnerabilidad encontrada, cómo "explotarla" y cómo "evitarla".

### **METASPLOIT.**

Es una herramienta diseñada para facilitar la "gestión de exploits" y tomar sin permiso el control de una amplia variedad de plataformas. La herramienta, desarrollada bajo licencia Open Source, está escrita en Perl para entornos UNIX, también incluye un shell Cygwin<sup>xx</sup> que permite correrla bajo Windows. En cuanto a su uso la herramienta se destaca por su sencillez.

### **SNORT.**

Sniffer de paquetes y detector de intrusos. Es un software muy flexible que ofrece capacidades de almacenamiento de sus bitácoras tanto en archivos de texto como en bases de datos abiertas como lo es MySQL.

## ***1.7.3 Métodos de ataques***

### ***Métodos de ataques pasivos***

#### **SNIFFING.**

El término "sniffing" se refiere al espionaje del tráfico de una red por parte de una máquina que captura la información, filtrándola para conseguir todos los datos relativos a las comunicaciones que desea monitorizar.

El objetivo de esta práctica es la comprensión del funcionamiento de los protocolos de red TCP HTTP al observar los mensajes intercambiados entre dos entidades en base a unos protocolos de comunicación determinados. Para ello se correrán diversas aplicaciones en la PC y se observa como los protocolos de red intercambian mensajes con entidades ejecutadas en máquinas remotas. Esta técnica funciona debido a que Ethernet utiliza broadcast para el envío de sus tramas.

#### **ESCAÑO DE PUERTOS.**

Es el proceso de enviar paquetes de datos a través de la red a números de puertos seleccionados (HTTP: 80, Telnet: 23, etc) de un equipo de red con el propósito de identificar la disponibilidad de servicios de red en ese sistema. El escaneo de un puerto es un método para obtener información y cuando se efectúa por individuos desconocidos se considera el preludio de un ataque.

Analizando qué puertos están abiertos en un sistema, el atacante puede buscar agujeros en cada uno de los servicios ofrecidos: cada puerto abierto en una máquina es una potencial puerta de entrada a la misma.

### ***Metodos de ataques activos***

#### **ARP- POISONING**

---

<sup>xviii</sup> Elemento de hardware o software utilizado en las redes para prevenir algunos tipos de comunicaciones prohibidas por las políticas de red

<sup>xix</sup> Permite ver cómo y por dónde viaja la información a través de la Red

<sup>xx</sup> Permite tener un entorno LINUX bajo un sistema Operativo Windows.

## I.7 INTRODUCCIÓN A LA SEGURIDAD EN CÓMPUTO

---

ARP spoofing es un método en la que se explota la interacción de la IP y el protocolo de Ethernet. Esto permite al atacante ver todo lo que el equipo manda y recibe. El objetivo del ataque es envenenar la tabla de ARP's de un sistema. Sólo puede llevarse a cabo cuando el atacante está "conectado a la misma LAN" que las víctimas.

Una vez inundada la tabla ARP, es posible escuchar todas las comunicaciones establecidas por los equipos que se encuentran en la red segmentada.

### **BUFFER OVERFLOW**

Un desbordamiento de búfer (buffer overflow) es un error de software que se produce cuando se copia una cantidad más grande de datos sobre un área más pequeña sin interrumpir la operación, sobrescribiendo otras zonas de memoria. Esto es posible dado que en las arquitecturas comunes de PC's, la memoria no tiene separación entre la dedicada a datos y a programa.

Si el programa que tiene el error en cuestión tiene privilegios especiales se convierte además en un fallo de seguridad. El código copiado especialmente preparado para obtener los privilegios del programa atacado se llama shellcode.

### **SHELL CODE**

Una Shell code es una serie de ordenes en ensamblador que hace algo a lo cual se saca provecho; ejecutar /bin/sh para obtener una shell, por ejemplo copiar una shell y setearla con suid root, etc.

Se usa para conseguir ejecutar un código después de haber sobrescrito la dirección de retorno de un programa/función mediante un overflow, o mediante cualquier otro método válido. Es decir, el valor de la dirección de retorno que se sobrescribirá será la del shellcode del programa.

### **NEGACIÓN DE SERVICIO (DOS)**

Las negaciones de servicio (conocidas como DoS, Denial of Service) son ataques dirigidos contra un recurso informático (generalmente una máquina o una red, con el objetivo de degradar total o parcialmente los servicios prestados por ese recurso a sus usuarios legítimos; constituyen en muchos casos uno de los ataques más sencillos y contundentes contra todo tipo de servicios, y en entornos donde la disponibilidad es valorada por encima de otros parámetros de la seguridad.

### **SMURFING**

El smurf, llamado así debido al nombre de uno de los programas que se utilizan para realizar dichos ataques. El ataque consiste en que la máquina que origina el ataque envía un gran número de pings hacia direcciones de broadcast, teniendo cada uno de ellos como dirección de origen, la IP del servidor a ser atacado. Si el ruteador que entrega el paquete a la IP destino envía el paquete de broadcast a nivel 2 (del modelo de referencia OSI), la mayoría de los hosts dentro de la red responderán al ping, multiplicando así el tráfico original en tantas veces como el número de máquinas que respondan al echo de ICMP. En redes multi-acceso en donde se esté utilizando TCP/IP probablemente habrá una gran cantidad de equipos que respondan a cada ping que reciban.

## ***1.7.4 Mecanismos de seguridad***

Herramienta o medios básicos para garantizar la protección de los sistemas de una red. Los mecanismos de seguridad se dividen en tres grandes grupos:

**Mecanismos de prevención:** son aquellos que aumentan la seguridad de un sistema durante el funcionamiento normal de éste, previniendo la ocurrencia de violaciones a la seguridad; por ejemplo, el uso de cifrado en la transmisión de datos se puede considerar un mecanismo de este tipo, ya que evita que un posible atacante escuche las conexiones hacia o desde un sistema UNIX en la red.

**Mecanismos de detección:** se conoce a aquellos que se utilizan para detectar violaciones de la seguridad o intentos de violación; ejemplos de estos mecanismos son los programas de auditoria como Tripwire.

**Mecanismos de recuperación:** son aquellos que se aplican cuando una violación del sistema se ha detectado, es decir, retornar éste a su funcionamiento correcto; ejemplos de estos mecanismos son la utilización de copias de seguridad o el hardware adicional. Dentro de este último grupo de mecanismos de seguridad se encuentra un subgrupo denominado **mecanismos de análisis forense**, cuyo objetivo no es simplemente retornar al sistema a su modo de trabajo normal,

sino averiguar el alcance de la violación, las actividades de un intruso en el sistema, y la puerta utilizada para entrar; de esta forma se previenen ataques posteriores y se detectan ataques a otros sistemas.

Entre los mecanismos de recuperación más importantes se encuentra:

- **Respaldos:** Consiste en copiar los archivos e información importante en un medio externo u otro servidor con el fin de tenerlos disponibles para prevenir accidentes.
- **Planes de contingencia:** Consiste en un análisis pormenorizado de las áreas que componen una organización para establecer una política de recuperación ante un desastre.
- **Bitácoras:** Se refiere al procedimiento a través del cual un sistema operativo registra eventos conforme van ocurriendo y los preserva para un uso posterior.
- **DRP (Disaster Recovery Planning):** Recuperar la operación de los servicios computacionales y de telecomunicaciones después de un desastre
- **BCP (Business Continuity Planning):** Capacidad para mantener la continuidad de las operaciones.

### ***1.7.5 Criptografía***

#### ***Historia de la criptografía***

Las primeras civilizaciones desarrollaron técnicas para enviar mensajes durante las campañas militares de forma que si el mensajero era interceptado la información que portaba no corriera el peligro de caer en manos del enemigo. Posiblemente, el primer criptosistema que se conoce fuera documentado por el historiador griego Polybios: un sistema de sustitución basado en la posición de las letras en una tabla. También los romanos utilizaron sistemas de sustitución, siendo el método actualmente conocido como César, porque supuestamente Julio César lo utilizó en sus campañas.

Durante los siglos XVII, XVIII y XIX, el interés de los monarcas por la criptografía fue notable. Las huestes de Felipe II utilizaron durante mucho tiempo una cifra con un alfabeto de más de 500 símbolos que los matemáticos del rey consideraban inexpugnable. Cuando el matemático francés Francois Viete consiguió criptoanalizar aquel sistema para el rey de francia, a la sazón Enrique IV, el conocimiento mostrado por el rey francés impulsó una queja de la corte española ante del papa Pío V acusando a Enrique IV de utilizar magia negra para vencer a sus ejércitos.

Es en el siglo XX cuando la historia de la criptografía vuelve a presentar importantes avances, durante las dos contiendas bélicas que marcaron al siglo: la Gran Guerra y la Segunda Guerra Mundial. La máquina alemana Enigma automatizaba considerablemente los cálculos que eran necesarios realizar para las operaciones de cifrado y descifrado de mensajes. Para vencer al ingenio alemán, fue necesario el concurso de los mejores matemáticos de la época y un gran esfuerzo computacional.

A mediados de los años 70 el Departamento de Normas y Estándares norteamericano publica el primer diseño lógico de un cifrador que estaría llamado a ser el principal sistema criptográfico de finales de siglo: el Estándar de Cifrado de Datos o DES. En esas mismas fechas ya se empezaba a gestar lo que sería la, hasta ahora, última revolución de la criptografía teórica y práctica: los sistemas asimétricos. Estos sistemas supusieron un salto cualitativo importante ya que permitieron introducir la criptografía en otros campos que hoy día son esenciales como el de la firma digital.

#### ***Definición***

Del griego kryptos (ocultar) y grafos (escribir), literalmente escritura oculta, la criptografía es el arte o ciencia de cifrar y descifrar información utilizando técnicas matemáticas que hagan posible el intercambio de mensajes de manera que sólo puedan ser leídos por las personas a quienes van dirigidos.

La finalidad de la criptografía es, garantizar el secreto en la comunicación entre dos entidades (personas, organizaciones, etc), y asegurar que la información que se envía es auténtica en un doble sentido: que el remitente sea realmente quien dice ser; y por último impedir que el contenido del mensaje enviado (habitualmente denominado criptograma) sea modificado en su tránsito.

El cifrado es el proceso de convertir el texto plano (información original) en un símbolos ilegibles, denominado texto cifrado o criptograma. Las dos técnicas más básicas de cifrado en la criptografía clásica son la sustitución (que supone el

cambio de significado de los elementos básicos del mensaje -las letras, los dígitos o los símbolos-) y la transposición (que supone una reordenación de las mismas).

El descifrado es el proceso inverso que recupera el texto plano a partir del criptograma y la clave.

### **Tipos de cifrado**

**CIFRADO SIMÉTRICO.** Cuando se emplea la misma clave en las operaciones de cifrado y descifrado, se dice que el sistema criptográfico es simétrico o de clave secreta. Ésta es la opción utilizada para cifrar el cuerpo de los mensajes en el correo electrónico o los datos intercambiados en las comunicaciones digitales.

**CIFRADO ASIMÉTRICO.** Cuando se utiliza una pareja de claves para separar los procesos de cifrado y descifrado, se dice que el sistema criptográfico es asimétrico o de clave pública. Una clave, la privada, se mantiene secreta, mientras que la segunda clave, la pública, es conocida por todos. De forma general, las claves públicas se utilizan para cifrar y las privadas, para descifrar.

### **Estenografía**

La estenografía es la ciencia que estudia los procedimientos encaminados a ocultar la existencia de un mensaje en lugar de ocultar su contenido. El objetivo de la estenografía es ocultar ese mensaje dentro de otro sin información importante, de forma que el atacante ni siquiera se entere de la existencia de dicha información oculta. No se trata de sustituir al cifrado convencional sino de complementarlo: ocultar un mensaje reduce las posibilidades de que sea descubierto; no obstante, si lo es, el que ese mensaje haya sido cifrado introduce un nivel adicional de seguridad.

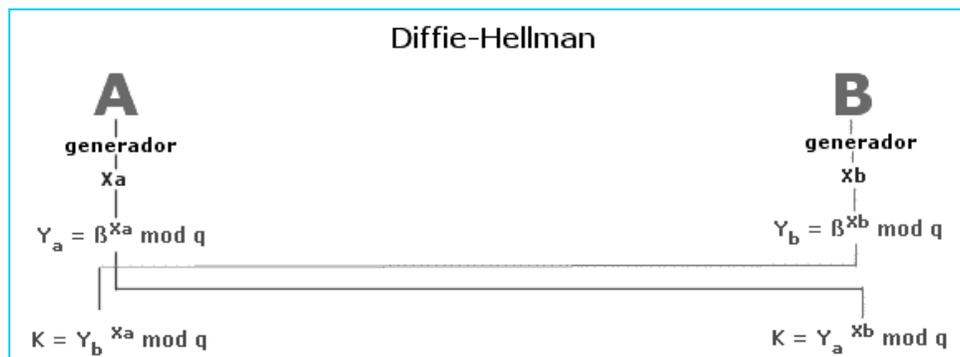
### **1.7.6 Algoritmo Diffie Hellman**

Este algoritmo de encriptación de Whitfield Diffie y Martin Hellman fue el punto de partida para los sistemas asimétricos, basados en dos claves diferentes, la pública y la privada.

Matemáticamente se basa en las potencias de los números y en la función mod (módulo discreto). Uniendo estos dos conceptos se define la potencia discreta de un número como  $Y = X^a \text{ mod } q$ . Si bien el cálculo de potencias discretas es fácil, la obtención de su función inversa, el logaritmo discreto, no tiene una solución analítica para números grandes.

Para implementar el sistema se realizan los siguientes pasos:

1. Se busca un número primo muy grande, denominado  $q$ .
2. Se obtiene el número  $\beta$ , raíz primitiva de  $q$ , es decir, que cumple que  $\beta \text{ mod } q, \beta^2 \text{ mod } q, \dots, \beta^{q-1} \text{ mod } q$  son números diferentes.
3.  $\beta$  y  $q$  son las claves públicas.



I.7.2 Diagrama del algoritmo Diffie Hellman

Para generar una clave simétrica compartida entre dos usuarios, A y B, ambos parten de un generador de números pseudoaleatorios, que suministra un número de este tipo diferente a cada uno,  $X_a$  y  $X_b$ . Estos son las claves privadas de A y B. Con estos números y las claves públicas  $\beta$  y  $q$  que ambos conocen, cada uno genera un número intermedio,  $Y_a$  e  $Y_b$ , mediante las fórmulas:

## I.7 INTRODUCCIÓN A LA SEGURIDAD EN CÓMPUTO

$$Y_a = \beta X_a \text{ mod } q$$

$$Y_b = \beta X_b \text{ mod } q$$

Estos números son intercambiados entre ambos, y luego cada uno opera con el que recibe del otro, obteniendo en el proceso el mismo número ambos:

$$K = Y_b X_a \text{ mod } q$$

$$K = Y_a X_b \text{ mod } q$$

Este número K es la clave simétrica que a partir de ese momento ambos comparten, y que pueden usar para establecer una comunicación cifrada mediante cualquiera de los sistemas simétricos.

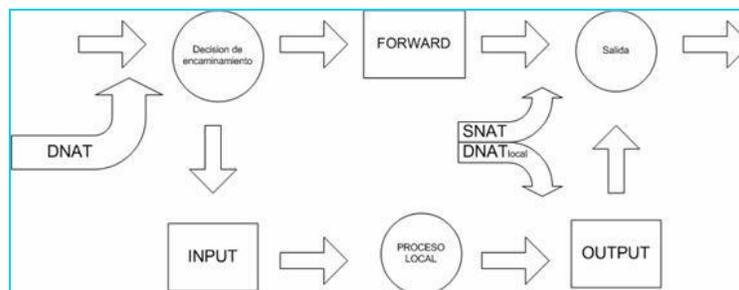
### I.7.7 Firewall

Es un dispositivo de control, ya sea hardware o software, de acceso que determina qué conexiones son permitidas "ingresar a" y "salir de" una red en base a un conjunto de reglas predefinidas. Trabaja como un filtro que controla todas las comunicaciones que pasan de una red a la otra y en función de lo que sea permite o deniega su paso. Para esto el firewall examina el tipo de servicio al que corresponde, como pueden ser un servidor Web o correo. Dependiendo del servicio el firewall decide si lo permite o no. Además, examina si la comunicación es entrante o saliente y dependiendo de su dirección puede permitirla o no.

### I.7.8 IPTABLE

Es un sistema de firewall vinculado al kernel de LINUX, es parte del sistema operativo, no es un programa sino un conjunto de reglas. Ejecutando el comando IPTABLE, se añaden, borran, o crean reglas. IPTABLE no es sino un simple script de shell en el que se van ejecutando las reglas de firewall.

Las reglas de firewall están a nivel de kernel, y a éste lo que le llega es un paquete y tiene que decidir que hacer con él. El kernel lo que hace es, dependiendo si el paquete es para la propia maquina o para otra maquina, consultar las reglas de firewall y decidir que hacer con el paquete según las reglas. Este es el camino que seguiría un paquete en el kernel:

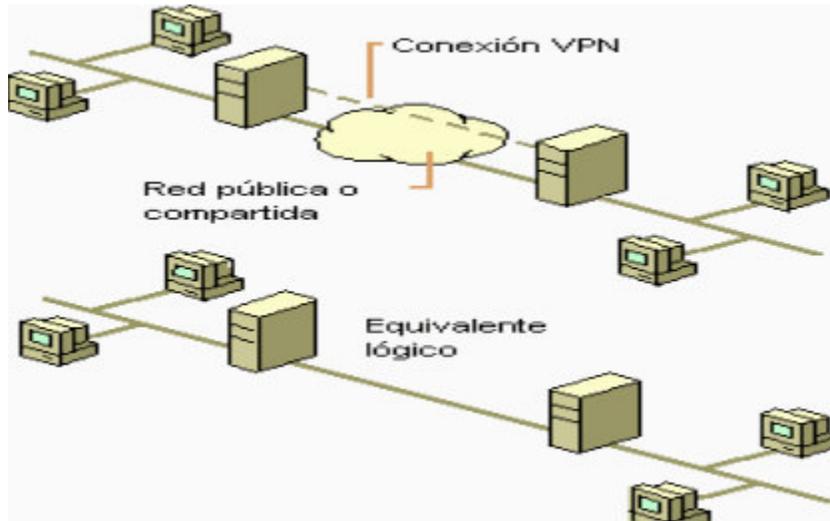


I.7.3 IPTABLE

Para los paquetes (o datagramas, según el protocolo) que van a la propia maquina se aplican las reglas INPUT y OUTPUT, y para filtrar paquetes que van a otras redes o maquinas se aplican simplemente reglas FORWARD.

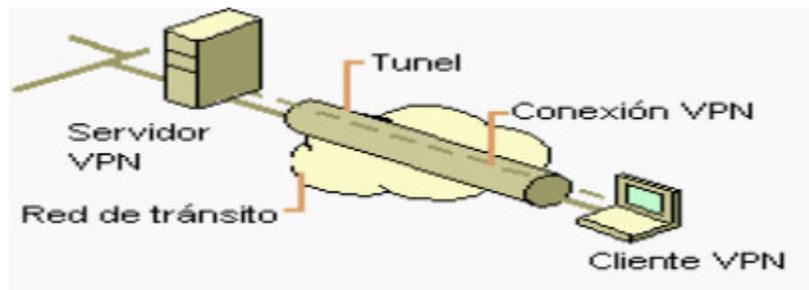
### I.7.9 Red VPN

Es una red privada que se extiende, mediante un proceso de encapsulación y en su caso de encriptación, de los paquetes de datos a distintos puntos remotos mediante el uso de unas infraestructuras públicas de transporte. Los paquetes de datos de la red privada viajan por medio de un "túnel" definido en la red pública.



I.7.4 VPN

Las redes privadas virtuales crean un túnel o conducto de un sitio a otro para transferir datos a esto se le conoce como encapsulación además los paquetes van encriptados de forma que los datos son ilegibles para los extraños.



I.7.5 Diagrama de una VPN

El servidor busca mediante un ruteador la dirección IP del cliente VPN y en la red de tránsito se envían los datos sin problemas. La VPN debe ser capaz de verificar la identidad de los usuarios y restringir el acceso a la VPN a aquellos usuarios que no estén autorizados.

# I.8 DESARROLLO DE APLICACIONES CON POSTGRESQL Y PHP

**OBJETIVO:** Crear una aplicación dinámica e interactiva de base de datos para Internet con técnicas avanzadas del lenguaje PHP y la base de datos PostgreSQL.

### **I.8.1 PHP 5**

PHP 5 es la respuesta a los nuevos lenguajes de programación que están tomando auge en Internet, como ejemplo esta Java de SUN o la tecnologías .NET de Microsoft que están 100% enfocados a objetos.

El nuevo PHP5 se apoya en la llamada Zend Engine 2, la nueva versión del motor Zend -desarrollado por Zeev Suraski y Andi Gutmans, padres de PHP. Entre las nuevas características de PHP5 se encuentran:

#### **Objetos y Referencias**

En PHP4 las variables que nombran los objetos guardan el objeto completo en sí (sus datos), tal y como hace una variable simple o de array. Aunque este comportamiento puede parecer deseable a simple vista, acarrea una serie de problemas. Por ejemplo, sino se quería realizar una copia del objeto sino una referencia del mismo se debía usar el operador de referencia '&' para indicarlo, esto ocurre igual con el paso de parámetros a una función, lo que provoca de forma continua la proliferación de errores por "descuidos" y olvidos.

En PHP5 todas las variables que nombran objetos son en realidad referencias. No hay que usar el operador '&' ni en las asignaciones, ni en el paso de parámetros que son objetos. Este modelo esta basado en las diseño de variables de Java.

#### **Clases**

En PHP5 se incluyen modificadores de control de acceso para implementar la encapsulación. Se introduce tres palabras clave (public, private y protected) que sustituyen a var en la definición de variables miembro -atributos- de la clase, y que preceden a la definición de funciones miembro -métodos-.

Los tres modificadores tienen el significado de:

- **public:** la variable o función es accesible desde cualquier ámbito, la misma clase, otra clase o el ámbito global.
- **private:** la variable o función sólo es accesible desde dentro de la clase a la que pertenece.
- **protected:** la variable o función sólo es accesible desde dentro de la clase a la que pertenece o de cualquiera de sus clases derivadas.

En PHP4 todas las variables y las funciones son públicas. Por compatibilidad, PHP5 considera públicas todas las funciones a las que explícitamente no se les proporcione un modificador de acceso de los tres anteriores, también admite por compatibilidad el uso de var como sinónimo de public.

#### **Sobrecarga de métodos (Overloading)**

La versión de PHP 5 no soporta sobrecarga de métodos, pero permite emular la sobrecarga. El método `__call()` es invocado para todo método que no exista en la clase actual. Ello permite utilizarlo como una función *catch-all* desde la que controlar todas las funciones que se quieran sobrecargar, agrupando todo el código de gestión de la "sobrecarga" en un único punto.

También en el caso de `__call()` sólo se llama cuando un método no ha sido definido. Si se ha definido, se llama al método normalmente sin invocar a `__call()` (y si es privado, el control de acceso producirá el error que debería dar).

#### **Constructores y destructores**

En PHP4, el constructor de un objeto se denotaba con el mismo nombre que la clase y estos constructores se heredan, creando una serie de inconvenientes.

En PHP5 se ha escogido que el constructor tenga un nombre homogéneo en todas las clases: `__construct()`. Por compatibilidad hacia atrás, si dicho método `__construct()` no existe, se seguirá la misma regla que en PHP4, se buscará

un método con el mismo nombre que la clase; y si tampoco se encuentra, se proporcionará un constructor por defecto que simplemente reservará el espacio en memoria para el objeto creado con *new*.

Un detalle a notar es que, al no existir la sobrecarga de métodos, tampoco se puede sobrecargar el constructor, por lo que el constructor es único. PHP5 también soporta el concepto de destructor, implementado mediante el identificador `__destruct()`. Este destructor funciona así: cuando un objeto ha quedado inaccesible, al llegar la cuenta de referencias a cero, el destructor se invoca antes de liberar la memoria definitivamente.

### **Manejo de Errores**

PHP 5 tiene un modelo de excepciones similar al de otros lenguajes de programación, por ejemplo java. Una excepción puede ser lanzada, intentada o capturada en PHP. Un bloque de intento (try) debe incluir por lo menos un bloque de captura (catch). Las excepciones pueden ser lanzadas dentro de bloques de captura.

Cuando es lanzada una excepción, la siguiente línea de código no será ejecutada y PHP intentará encontrar el primer bloque de captura de excepciones. Si una excepción no es capturada se despliega un error fatal de PHP con un mensaje de que la excepción no fue capturada, a menos que exista un manejador de errores definido como `set_exception_handler()`.

### **I.8.2 Programación orientada a objetos**

La programación orientada a objetos es una metodología de programación avanzada y bastante extendida, en la que los sistemas se modelan creando clases, que son un conjunto de datos y funcionalidades. Las clases son definiciones, a partir de las que se crean objetos. Los objetos son ejemplares de una clase determinada y como tal, disponen de los datos y funcionalidades definidos en la clase.

La programación orientada a objetos tiene un mecanismo llamado herencia por el que se pueden definir clases a partir de otras clases. Las clases realizadas a partir de otra clase o mejor dicho, que extienden a otra clase, se llaman clases extendidas o clases derivadas.

Las clases extendidas heredan todos los atributos y métodos de la clase base. Además, pueden tener tantos atributos y métodos nuevos como se desee.

### **I.8.3 PostgreSQL**

PostgreSQL es un servidor de base de datos relacional, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.

Con cerca de una década de desarrollo tras él, PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multiversión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl, tcl y python).

Algunas de sus principales características son:

- Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas
- Disparadores (triggers)
- Vistas
- Integridad transaccional
- Acceso concurrente multiversión
- Capacidad de albergar programas en el servidor en varios lenguajes
- Herencia de tablas
- Tipos de datos y operaciones geométricas

Estas características colocan a PostgreSQL en la categoría de las bases de datos identificadas como objeto-relacionales. PostgreSQL tiene algunas características que son propias del mundo de las bases de datos orientadas a objetos.

### **Historia**

La implementación del DBMS Postgres comenzó en 1986. Los conceptos iniciales para el sistema fueron presentados en The Design of Postgres y la definición del modelo de datos inicial apareció en The Postgres Data Model.

Illustra Information Technologies tomó el código y lo comercializó. Postgres llegó a ser el principal gestor de datos para el proyecto científico de computación Sequoia 2000 a finales de 1992. En un esfuerzo por reducir esta carga, el proyecto terminó oficialmente con la Versión 4.2.

En 1994, Andrew Yu y Jolly Chen añadieron un intérprete de lenguaje SQL a Postgres, llamando a esta versión Postgres95. El código de Postgres95 fue adaptado a ANSI C y su tamaño reducido en un 25%. Muchos cambios internos mejoraron el rendimiento y la facilidad de mantenimiento.

Al llegar el año de 1996 se pensó que el nombre de Postgres95 ya no era el indicado por lo que se cambió a PostgreSQL, para reflejar la relación entre el Postgres original y las versiones más recientes con capacidades SQL.

### **Instalación de PostgrSQL**

Para instalar *PostgreSQL* primero se tiene que conseguir el código fuente, para esto se puede descargar de <http://www.postgresql.org/> y es preferible descargar la última versión estable.

#### **PASOS PARA LA INSTALACIÓN**

1.- Descomprimir el archivo *.tar.gz*. Para hacer esto se emplea el comando *tar* de la siguiente forma

```
tar zxvf postgresql-8.0.3.tar.gz
```

**NOTA: LOS NÚMEROS 8.0.3 CORRESPONDEN A LA VERSIÓN DE POSTGRESQL.**

Al ejecutar este comando se descomprime el código fuente dentro de un directorio que lleva el nombre de *postgresql-8.0.3*.

2.- Hay que cambiarse al directorio creado (*postgresql-8.0.3*). ***cd postgresql-8.0.3***

3.- Se ejecuta el programa *configure*. Este programa verifica que existan las librerías y requerimientos necesarios para la compilación del PostgreSQL, así como la especificación de parámetros adicionales de la configuración, por ejemplo el parámetro *--prefix=* especifica el directorio en donde se instalarán los binarios.

```
./configure --prefix=/usr/local/pgsql
```

El comando se tiene que escribir con un *./* para poder decirle al sistema que es un programa ejecutable.

4.- A continuación se utilizan los comandos *make* y *make install*. El comando *make* compila o genera los binarios de la aplicación y *make install* copia los binarios en las rutas requeridas para su funcionamiento.

**NOTA: PARA LA INSTALACIÓN DE UNA APLICACIÓN, SIEMPRE SE UTILIZA PRIMERO EL COMANDO MAKE Y DESPUÉS EL COMANDO MAKE INSTALL.**

5.- Para continuar con la instalación hay que crear el usuario de la aplicación, denominado *postgres* al que se le da como *home* el directorio donde se instaló PostgreSQL: ***useradd postgres -d /usr/local/pgsql/***

6.- En el directorio donde se instaló PostgreSQL hay que crear la carpeta *data* y hay que asignarle como dueño el usuario *postgres*, en esta carpeta se instalan las bases de datos del sistema.

```
mkdir /usr/local/pgsql/data  
chown postgres /usr/local/pgsql/data
```

### **Inicio del servicio de PostgreSQL**

Una vez que se completo la instalación es posible iniciar el servidor de PostgreSQL con los siguientes pasos:

1.- Primero se tiene que logear con el usuario postgres: **su – postgres**

2.- Para instalar las tablas requeridas por el sistema se utiliza el comando *initdb* al cual se le proporciona la ubicación del directorio en donde se van a instalar: **/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data**

3.-Para iniciar el servidor de PostgreSQL se utiliza el comando *postmaster*, al que se le proporciona la ubicación del directorio donde están las bases de datos.

```
/usr/local/pgsql/bin/postmaster -D /usr/local/pgsql/data >logfile 2>&1 &
```

4.- Utilizando el comando *createdb* se crea una base de datos para probar el servidor, para realizar la conexión se utiliza el comand *psql* con el nombre de la base de datos, el usuario es opcional sino se le proporciona uno toma el login del usuario que esta ejecutando el comando:

```
/usr/local/pgsql/bin/createdb test  
/usr/local/pgsql/bin/psql test
```

Si la conexión se realiza con éxito en la pantalla se presentara algo parecido a:

Welcome to the POSTGRESQL interactive sql monitor:

Please read the file COPYRIGHT for copyright terms of POSTGRESQL

type \? for help on slash commands

type \q to quit

type \g or terminate with semicolon to execute query

You are currently connected to the database: template1

test=>

En PostgreSQL la opción de solicitar el password para realizar conexiones a la base de datos esta deshabilitada. Para habilitarla hay que modificar el archivo *pg\_hba.conf* que controla las conexiones por host.

El archivo *pg\_hba.conf* se encuentra en el directorio donde están las bases de datos (en data) y para habilitar la petición de password hay que modificar todo lo que indique trust por password.

### **Comandos de SQL para PostgreSQL**

Para ejecutar los comandos de SQL se emplea la interfaz de comandos de PostgreSQL empleando el comando *psql* para acceder a ella.

En esta interfaz en posible emplear comandos propios del servidor, estos comandos se reconocen por que inician con '\'.

Instrucción	Descripción
\?	Presenta la ayuda de la línea de comandos de PostgreSQL
\h [instruccion]	Presenta la ayuda de las instrucciones de SQL
\i nombre_del_archivo	Permite ejecutar archivos que contengan instrucciones SQL
\q	Salir del servidor de PostgreSQL
\d [nombretabla]	Presenta las tablas de una base de datos, si se le proporciona el nombre de una tabla despliega la estructura de la misma.
\c database	Se conecta a otra base de datos

**createuser -P Ivan** - Para crear un usuario, la opción **-P** significa que el password es obligatorio.

**create database nombrebase ;** - Para crear bases de datos.

**createdb nombrebase usuario;** - Sólo es necesario especificar el nombre de la base de datos en lugar del parámetro nombrebase y la base de datos estará creada.

**drop database nombrebase ;** - Para borrar una base de datos.

**dropdb nombrebase usuario;** - Solo un usuario con privilegios sobre una base puede borrarla.

**create table nombretabla (campo1 tipo, campoN tipoN, primary key (campo1), unique id (campo1));** - Este comando crea una tabla de nombre nombretabla con los campos campo1, campoN en donde se van dando los nombres que identifiquen el dato que será almacenado en ese campo. Los parámetros primary key y unique id especifican la llave primaria de la tabla y que no se permitan valores repetidos respectivamente.

**drop table nombretabla;** - Para borrar una tabla.

**Insertar datos (INSERT)** - Para agregar datos a la base de datos.

### Constraints

Los constraints son restricciones en alguna columna en la definición de una tabla. Hay varios tipos de restricciones disponibles tanto para las tablas como para las columnas, y se implementan de varias maneras. Algunas restricciones son implementadas como triggers, algunas como índices, y otras son atributos de la definición de una columna.

A continuación se presenta una lista de las posibles restricciones y cómo se implementan. Las letras entre paréntesis indican el tipo de restricción: (p)rimary key, (u)nique, (f)function, (c)heck.

Cláusula de Constraint	Nombre de Constraint	Usos	Nombre de Índice/Trigger	Función de Trigger
PRIMARY KEY	table_pkey (p)	índice	table_pkey	
UNIQUE	table_column_key (u)	índice	table_column_key	
FOREIGN KEY	\$n (f)	trigger	RI_ConstraintTrigger_n	RI_FKey_check_ins
tabla referida	\$n (f)	trigger	RI_ConstraintTrigger_n	RI_FKey_noaction_del
tabla referida	\$n (f)	trigger	RI_ConstraintTrigger_n	RI_FKey_noaction_upd
CHECK	table_column (c)	check	ninguno	

**Primary Key.** Se utiliza para establecer una clave representativa de los registros de la tabla.

**Foreign Key.** Se utiliza para especificar claves foráneas.

Este constrain incluye además dos cláusulas ON DELETE y ON UPDATE, que indican qué acción hay que ejecutar en el caso en que un registro en la tabla referenciada sea eliminado o actualizado. De hecho, en dichos casos en la columna referenciada (que es la que se está definiendo) podría haber valores inconsistentes.

### 1.8.4 Interacción con PHP

La interacción con PHP para generar páginas dinámicas se realiza por medio de instrucciones a través del código PHP. A continuación se muestran algunas instrucciones utilizadas y su descripción:

**PG\_CONNECT.** Establece una conexión a una base de datos de PostgreSQL. Cada uno de los argumentos debe ser una cadena entrecomillada, incluyendo de ser necesario el número de puerto.

La conexión al servidor se cerrará en cuanto la ejecución del script acabe, a menos que se cierre antes con la función `pg_close`.

#### Sintaxis.

`pg_connect ( string host, string port, string options, string dbname )`

**PG\_PCONNECT.** Crea una conexión persistente con un base de datos de PostgreSQL. Primero, la función intenta encontrar una conexión persistente que ya esté abierta con la misma máquina, usuario y password. Si es encontrada,

devuelve el identificador de la misma, en lugar de crear una nueva conexión. Segundo, la conexión al servidor SQL no será cerrada cuando finalice la ejecución del script, sino que permanece abierta para un uso posterior.

### Sintaxis.

```
int pg_pconnect(string [hostname] , string [username] , string [password] );
```

**PG\_CLOSE.** Cierra la conexión a la base de datos de PostgreSQL asociada al identificador de conexión especificado. Si no se especifica un identificador de conexión, se asume la última conexión abierta.

### Sintaxis.

```
int pg_close(int conexión );
```

**PG\_QUERY.** Ejecuta una consulta en la base de datos. Para poder utilizar *pg\_query* se tiene que haber establecido una conexión a la base de datos usando *pg\_connect* o *pg\_pconnect*.

### Sintaxis.

```
resource pg_query ( string query )
```

## I.8.5 SMARTY

Smarty es un motor de plantillas para PHP. Sirve para separar la parte de programación de la parte de diseño del sistema o sitio Web, donde el diseñador edita la plantilla, y usa una combinación de etiquetas HTML y etiquetas de plantilla para formatear la presentación de estos elementos (HTML, tablas, color de fondo, tamaño de letras, hojas de estilo, etc...) y el programador genera el código para recuperar datos o realizar procesos.

Algunas de las características de Smarty:

- Rápido.
- Eficiente
- No analiza gramaticalmente desde arriba el template, sólo compila una vez.
- El lenguaje de la platilla es altamente extensible.
- Sintaxis de etiquetas delimitadoras para configuración de la plantilla, así lo puede usar {}, {{}}, <!--{}-->, etc.
- Permite un anidamiento ilimitado de sections, ifs, etc.
- Es posible incrustar directamente código PHP en los archivos de plantilla.
- Funciones habituales de manipulación de cache.
- Arquitectura de Plugin.

### Instalación

La instalación de Smarty es simple, primero hay que obtener las librerías, de preferencia la última versión, y es recomendable descargarlas de su página Web (<http://smarty.php.net/>). Las librerías se encuentran dentro de un archivo *.tar.gz*, por lo que hay que descomprimirlas.

```
tar zxvf Smarty-2.6.9.tar.gz
```

Este comando descomprime las librerías en una carpeta con el nombre de Smarty-2.6.9.

**NOTA: PARA PODER UTILIZAR LAS CLASES DE SMARTY SE TIENE QUE TENER ACCESOS A LAS LIBRERÍAS, ESTO SE PUEDE HACER DE DOS FORMAS, 1) DECLARANDO LAS LIBRERÍAS DENTRO DEL PATH DE PHP, MODIFICANDO EL PHP.INI PARA QUE LOS SCRIPT ACCEDAN A ELLAS DE FORMA AUTOMÁTICA O 2) INCLUYENDO LAS LIBRERÍAS EN CADA SCRIPT EN LAS QUE SE NECESITE, EN ESE CASO SE INDISPENSABLE PROPORCIONAR LA RUTA ABSOLUTA DE DONDE SE ENCUENTRAN LAS CLASES DE SMARTY.**

### Usando Smarty

Para utilizar Smarty primero hay que instanciar un objeto y se tienen que definir 4 variables principales:

- **\$template\_dir** – Indica la ruta del directorio donde se encuentran los templates.
- **\$compile\_dir** – Indica la ruta del directorio donde los templates compilados están localizados.
- **\$config\_dir** – Indica la ruta del directorio usado para almacenar archivos de configuración de los templates.
- **\$cache\_dir** – Indica la ruta del directorio donde los caches del template son almacenados.

Esta declaración se hace inmediatamente después de que se instancia el objeto.

### **Comandos**

En Smarty se utilizan principalmente dos funciones:

**ASSIGN.** Se utiliza para crear variables y darles valores.

#### **Sintaxis.**

```
$objeto_Smarty->assign('variable', valor);
```

El tipo de variable que se puede crear son iguales a los que soporta PHP.

**DISPLAY.** Esta función se utiliza para definir el template a utilizar y para desplegar o imprimir el template ya con las variables definidas.

#### **Sintaxis.**

```
$smarty->display('index.tpl');
```

### **Definición de un template**

Los templates son archivos HTML que utiliza Smarty para genera las páginas que se despliegan en el navegador. Estos archivos también llamados plantillas facilitan el proceso de edición, eliminando el código PHP incrustado, sin eliminar la funcionalidad.

La mayor ventaja de utilizar templates es que se mantiene separado la programación PHP del diseño del sitio y tanto el programador como el diseñador utilizan archivos separados.

La mayor desventaja es que hace más complicada la programación y consume tiempo de procesamiento de CPU. Los templates trabajan de la siguiente manera: el archivo PHP invoca a la librería que genera de forma automática el HTML, reemplazando las etiquetas que fueron definidas en el arreglo de variables.

Smarty también proporciona sus propias sentencias de control, en ellas se puede introducir funciones de PHP, aunque no es muy recomendable.

### **Sentencias de control**

Smarty implementa sus propias sentencias de control, además de incorpora las más utilizadas.

#### **IF.**

La sentencia if es un condicional, lo que significa que hace una cosa u otra, determinada por la expresión que se le proporcione. La sentencia if se puede anidar, pero mientras más profundo sea el anidamiento más complicada es la programación.

#### **SECTION.**

Section es usado para realizar un ciclo de un arreglo de datos. Todas las etiquetas section deben tener su par /section. Los parámetros requeridos son name y loop. Sections pueden ser anidado.

Las variables del loop (normalmente un arreglo) determina el número de veces que se va a repetir el ciclo, y name es el nombre de la variable que va a estar tomando el valor del elemento del arreglo con cada repetición.

Si el arreglo es asociativo y se quiera acceder al valor de los elementos se tiene que poner un punto al final de la variable name y poner el nombre del elemento que se quiere acceder, este mismo comportamiento lo maneja en todo el código de Smarty.

#### **FOREACH.**

Foreach es usado para desplegar cada elemento de un arreglo asociativo simple o de índices numéricos. La sintaxis para foreach es mucho más simple que section, pero tiene la desventaja de que sólo puede usar un arreglo.

El ciclo foreach se cierra con la etiqueta /foreach. Los parámetros requeridos son from e item. Los ciclos foreach pueden ser anidados.

### **I.8.6 ADOdb**

ADOdb es un conjunto de librerías de bases de datos que permite conectar PHP con varias bases de datos de una manera portable.

ADOdb surgió como una necesidad para los programadores de PHP, dado que las funciones de acceso a base de datos no están estandarizadas y esto dificultaba la migración del sistema a otro manejador de base de datos. Esto requiere una librería que esconda las diferencias entre cada API de base de datos (encapsular las diferencias) para que se pueda cambiar fácilmente de base de datos.

ADOdb es compatible con los siguientes manejadores de bases de datos: MySQL, Oracle, Microsoft SQL Server, Sybase, Sybase SQL Anywhere, Informix, PostgreSQL, FrontBase, SQLite, Interbase (versiones de Firebird y Borland), Foxpro, Access, ADO, DB2, SAP DB y ODBC.

#### **Instalación de ADOdb**

Primero hay que obtener las librerías, en el sitio principal de ADOdb (<http://adodb.sourceforge.net/#download>) se encuentra la última versión.

Las versiones de PHP que soportan ADOdb son 4.0.4 en adelante. La instalación de las librerías es bastante simple, sólo hay que extraer los archivos de *.tar.gz* en un directorio accesible a servidor Web o también se puede agregar a la librería de PHP, modificando el archivo *php.ini* para tener acceso a ellas sin tener que referenciar la carpeta donde se encuentra. ***tar -zxvf httpd-2.0.47.tgz***

Este comando descomprime las librerías de ADOdb generando una carpeta con el nombre de adodb.

Por comodidad y para facilitar la programación es recomendable tener un directorio que este accesible a los scripts, por lo que es recomendable poner un directorio adodb en la carpeta del servidor (en apache es *htdocs*) de forma que no se tengan que incluir ligas demasiado largas y que puedan dificultar la programación o la migración del código.

```
cp -r adodb/ /usr/local/apache/htdocs/adodb/
```

El comando cp con la opción *-r* copia el directorio adodb dentro del directorio htdocs del servidor de apache para los scripts tengan el acceso a las librerías.

#### **Usando ADOdb**

ADOdb se utiliza como cualquier clase de PHP, se tiene que importar la clase y se tienen que instanciar un objeto.

```
include('adodb/adodb.inc.php');  
$db = ADONewConnection($driver);
```

#### **Funciones**

ADOdb tiene muchas funciones, a continuación se declaran algunas.

**ADONNECTION.** Objeto que realiza la conexión a la base de datos, ejecuta enunciados SQL y tiene un conjunto de funciones de utilería para estandarizar el formato de enunciados SQL para cosas como la concatenación y el formateo de fechas.

**ADONNEWCONNECTION().** Declarara o instancia un objeto de conexión. Esta función trabaja como constructor.

**CONNECT().** Realiza una conexión no persistente al servidor.

**EXECUTE()**. Ejecuta una instrucción SQL y regresa un objeto de la clase ADORecordSet si es correcto.

**GETASSOC(\$FORCE\_ARRAY)**. Genera un arreglo asociativo del recordset.

**GETARRAY(\$NUMBER\_OF\_ROWS)**. Genera un arreglo bidimensional de registros desde la posición actual del cursor, indexado desde 0 a (\$number\_of\_rows - 1). Si no se define \$number\_of\_rows, se indexa hasta el final de recordset (EOF<sup>xxi</sup>).

**RS2HTML()**. Esta es una función imprime un ADORecordSet, \$adorecordset como una tabla HTML. El parámetro \$tableheader\_attributes permite controlar los atributos de la tabla (cellpadding, cellspacing y border). Finalmente se pueden personaliza el nombre de las columnas de la base de datos modificando el arreglo \$col\_titles.

**Sintaxis.**

```
function rs2html($adorecordset,[$tableheader_attributes], [$col_titles])
```

**NOTA: RS2HTML NECESITA INCLUIR EL ARCHIVO TOHTML.INC.PHP.**

**Manejo de errores.**

ADODB proporciona un manejo de excepciones que se adapta al que PHP 5 tiene, modificando un poco la estructura de *try* y *catch* ya que no se necesita generar un objeto de excepción sino que se lanzan automáticamente, parecido a lo que ocurre en java.

Para generar las excepciones hay que incluir la librería `adodb-exceptions.inc.php` para que se pueden atrapar las excepciones conforme ocurran los errores.

Las siguientes funciones se pueden anexar al manejo de errores de PHP 5, esta funciones regresan mensajes de error que haya ocurrido o se haya generado por parte de ADODB.

**ERRORMSG()**.

Regresa el último mensaje de error. El mensaje de error se cambia después de cada llamado a la función `Execute()`.

**ERRORNO()**.

Regresa el último número de error. Este número de error se actualiza después de cada invocación de `Execute()`.

---

<sup>xxi</sup> End Of File o final de un archivo

**II. PROYECTO DE UN  
SISTEMA DE GESTIÓN  
DE LA CALIDAD PARA  
DISEÑO GRÁFICO**

### II.1 Definición del sistema

El presente informe muestra los aspectos del Sistema de Solicitud de Material de Diseño Gráfico versión 1.0, el cuál permite profundizar en los detalles de las tareas de altas, revisión, control de cambios y término de una solicitud.

Este informe consta de varios apartados, presenta en primera instancia, los antecedentes que dieron origen a esta aplicación, posteriormente se presenta el objetivo del Sistema Solicitud de Material de Diseño Gráfico.

Posteriormente, se describen las convenciones y términos utilizados en el reporte, una descripción de los elementos comunes que el usuario encontrará en el módulo, los requerimientos técnicos para poder tener acceso a éste, los cuales como observará son mínimos, siendo esta una ventaja de las aplicaciones desarrolladas en plataformas WEB.

Continuando con lo anterior se presentan los objetivos que persigue cada módulo, una breve descripción de los elementos que lo componen y finalmente se presenta una descripción detallada de la funcionalidad de cada una de las pantallas que conforman cada submódulo.

#### II.1.1 Justificación

Para el sistema se utilizo software libre por las siguientes ventajas:

- El costo de utilizar software libre es mínimo.
- Cada programa elegido para interactuar con el sistema ofrece una gran confiabilidad y seguridad.
- Los programas elegidos estan relacionados y trabajan de forma óptima entre si.

Los programas elegidos para realizar el sistem son los siguientes:

- 1.- Sistema Operativo: Se eligio Linux por su bajo costo de operación y adquisición. Ofrece una gran seguridad y esta abalado por empresas de alto renombre. Como soporte cuenta con un gran apoyo de foros y expertos en Internet.
- 2.- Servidor Web: Se eligio Apache, que es el servidor más utilizado en estos momentos, cuenta con un gran soporte y proporciona una alta seguridad y rendimiento.
- 3.- Lenguaje de Programación: Se eligio PHP por su facilidad de programación, proporciona seguridad y rendimiento además de que existe mucha documentación de ayuda.
- 4.- Base de datos: Para la base de datos se utiliza MySQL, un administrador de base de datos libre basado en SQL, su utilización es relativamente simple e interactúa con PHP de forma óptima.

#### II.1.2 Problemática

- ◆ La necesidad de regular el uso gráfico en los medios de comunicación.
- ◆ La necesidad de agilizar y facilitar el trámite para generar material gráfico.
- ◆ La necesidad de estandarizar un formato para las solicitudes de material gráfico.
- ◆ La posibilidad de llevar un control del trabajo del material gráfico generado.

#### II.1.3 Objetivo del sistema

- ◆ El objetivo principal de este sistema es mantener un control sobre las solicitudes de material gráfico, administrando desde Web el registro de las solicitudes y en la Base de Datos de Solicitudes.

#### II.1.4 Alcances

- ◆ Validación en formato y valor de los registros recibidos.
- ◆ Generación automática de Número de Solicitud.
- ◆ Generación de Correo electrónico de confirmación de alta de solicitud.

#### II.1.5 Exclusiones

- ◆ No envía al solicitante un aviso del término de su solicitud.
- ◆ No envía al solicitante la liga del cuestionario de mejora.

### II.1.6 Plataforma de desarrollo

#### Desarrollo solicitud

La solicitud en si esta integrada por 2 leguajes de programación en Web y un manejador de bases de datos:

- ◆ PHP: (Hypertext Preprocessor) es un lenguaje de programación de alto nivel, es un "preprocesador del hipertexto" que se basa en una sintaxis parecida al Lenguaje C, al Java y al Perl. El lenguaje se ha desarrollado como proyecto open-source. PHP es un lenguaje de scripting server-side.
- ◆ MySQL: El sistema gestor de bases de datos de código abierto.
- ◆ JavaScript: Es un lenguaje interpretado que permite incluir macros en páginas Web. Estas macros se ejecutan en la computadora del usuario de las páginas, y no en él.

#### Requisitos

##### SERVIDOR WEB APACHE.

Apache versión 1.3.27 o superior.

El servidio Apache debe cumplir con las siguientes características:

- Debe tener al lenguaje PHP como módulo.
- Debe estar compilado con la opción de conexión a MySQL.

##### LENGUAJE DE PROGRAMACIÓN PHP.

PHP versión 4.3.10 o superior.

En la configuración debe cumplir con las siguientes características:

- Permitir que archivos con extensión php y html sean compilados.
- La opción register\_globals este en OFF por cuestiones de seguridad.
- Permitir la etiqutea '<? ?>' como de apertura y cierre.

##### MANEJADOR DE BASE DE DATOS MYSQL

Mysql versión 4.0.13 o superior.

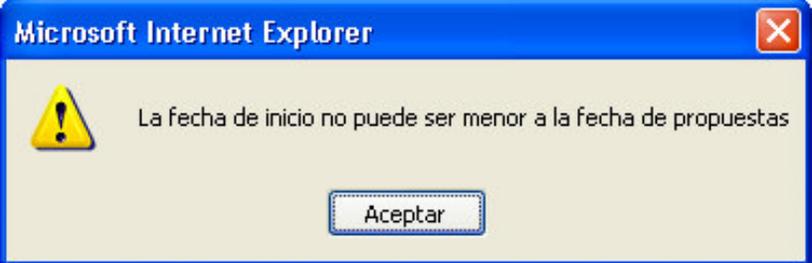
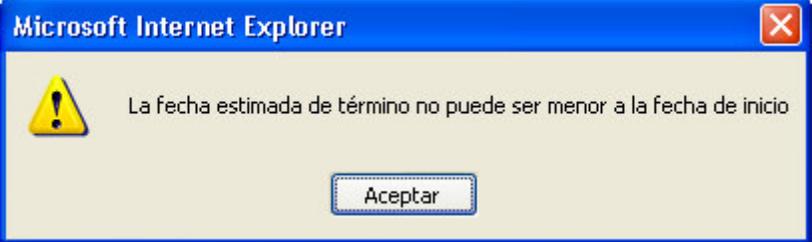
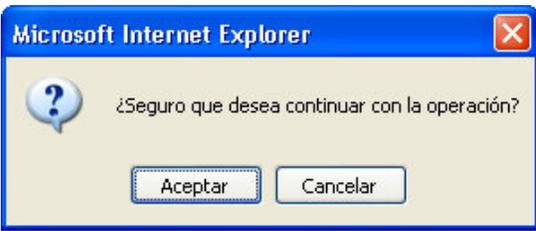
La configuración debe cumplir las siguientes características:

- Debe tener especificado un usuario para la conexión del sitio web.
- El usuario debe contar con password.

### II.1.7 Mensajes

MENSAJE	
<b>Mensajes de campos vacíos.</b> Cuando un campo que es obligatorio no contiene datos aparecerá un mensaje de error indicando que algún campo requerido en la pantalla esta vacío.	
	
	

MENSAJE	
<b>Mensajes de campos vacíos.</b> Cuando un campo que es obligatorio no contiene datos aparecerá un mensaje de error indicando que algún campo requerido en la pantalla esta vacío.	
	
	
	
	
	

MENSAJE	
<p><b>Mensajes de campos vacíos.</b> Cuando un campo que es obligatorio no contiene datos aparecerá un mensaje de error indicando que algún campo requerido en la pantalla esta vacío.</p>	
	
	
	
<p><b>Mensajes de Validación.</b> Al tratar de introducir un valor en el campo "Password" este debe tener 6 caracteres de longitud mínimo.</p>	
<p><b>Mensajes de Confirmación.</b> Cuando se desea continuar con la alta de una solicitud, aparece un cuadro solicitando la confirmación de la acción.</p>	
	
<p><b>Mensaje de Alta exitosa.</b> Este mensaje indica que la solicitud se registro satisfactoriamente</p>	<p><b>SU SOLICITUD HA SIDO COMPLETADA</b></p> <p>SU NÚMERO DE SOLICITUD ES:</p> <p>Recuerde o anote su número de solicitud para revisar sus avances.</p>

## II. SISTEMA DE GESTIÓN DE LA CALIDAD PARA DISEÑO GRÁFICO

<b>MENSAJE</b>	
<b>Mensaje de Error.</b> Este mensaje indica que ha ocurrido un error y que no se puede dar de alta la solicitud.	
<b>HA OCURRIDO UN ERROR</b> SU SOLICITUD NO HA SIDO REGISTRADA	<b>HA OCURRIDO UN ERROR</b> NO SE HA PODIDO ACTUALIZAR EL COMENTARIO
SU SOLICITUD NO HA SIDO COMPLETADA	<b>HA OCURRIDO UN ERROR</b> NO SE HA COMPLETADA LA OPERACIÓN
<b>Mensaje de Información.</b> Este mensaje indica que la solicitud solicitada ha finalizado.	
<b>LA SOLICITUD HA FINALIZADO</b>	<b>NO EXISTE ESA SOLICITUD</b>
PASSWORD O LOGIN INCORRECTO	<b>ACTUALIZADO</b>
LA SOLICITUD HA SIDO ACTUALIZADA	LA SOLICITUD HA SIDO TERMINADA
<b>EL CAMBIO HA SIDO GUARDADO</b>	SIN CAMBIOS

## II.2 SISTEMA DE GESTIÓN DE CALIDAD PARA DISEÑO GRÁFICO

### II.2.1 Menú general del sistema de gestión de calidad para diseño gráfico

En la parte central se encuentra el menú que conduce a los módulos que integran el SISTEMA DE GESTIÓN DE CALIDAD PARA DISEÑO GRÁFICO.

#### Pantalla Menú



### SOLICITUDES

ESCOGE QUE DESEAS HACER ▾

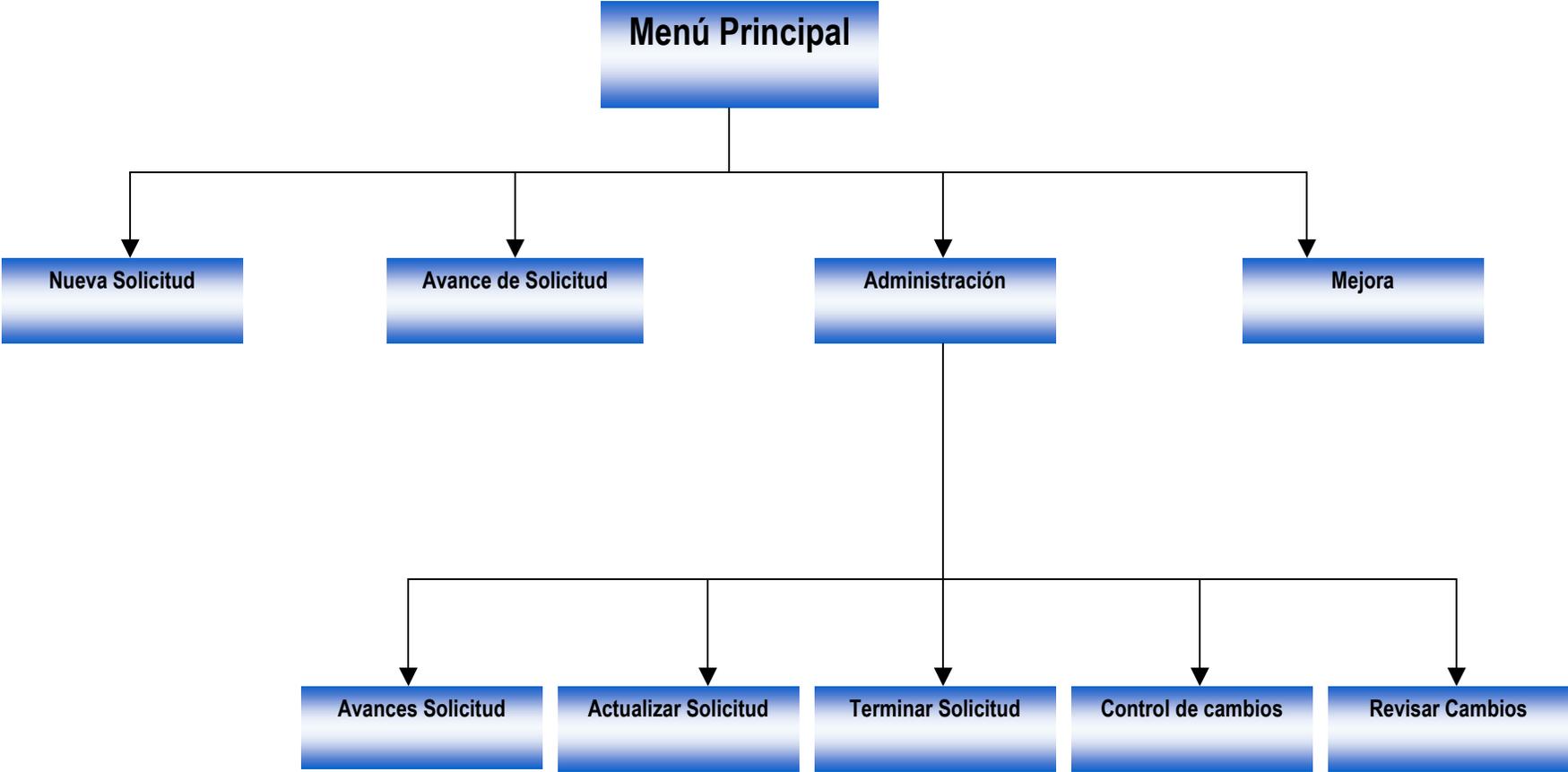
Ver

Regresar

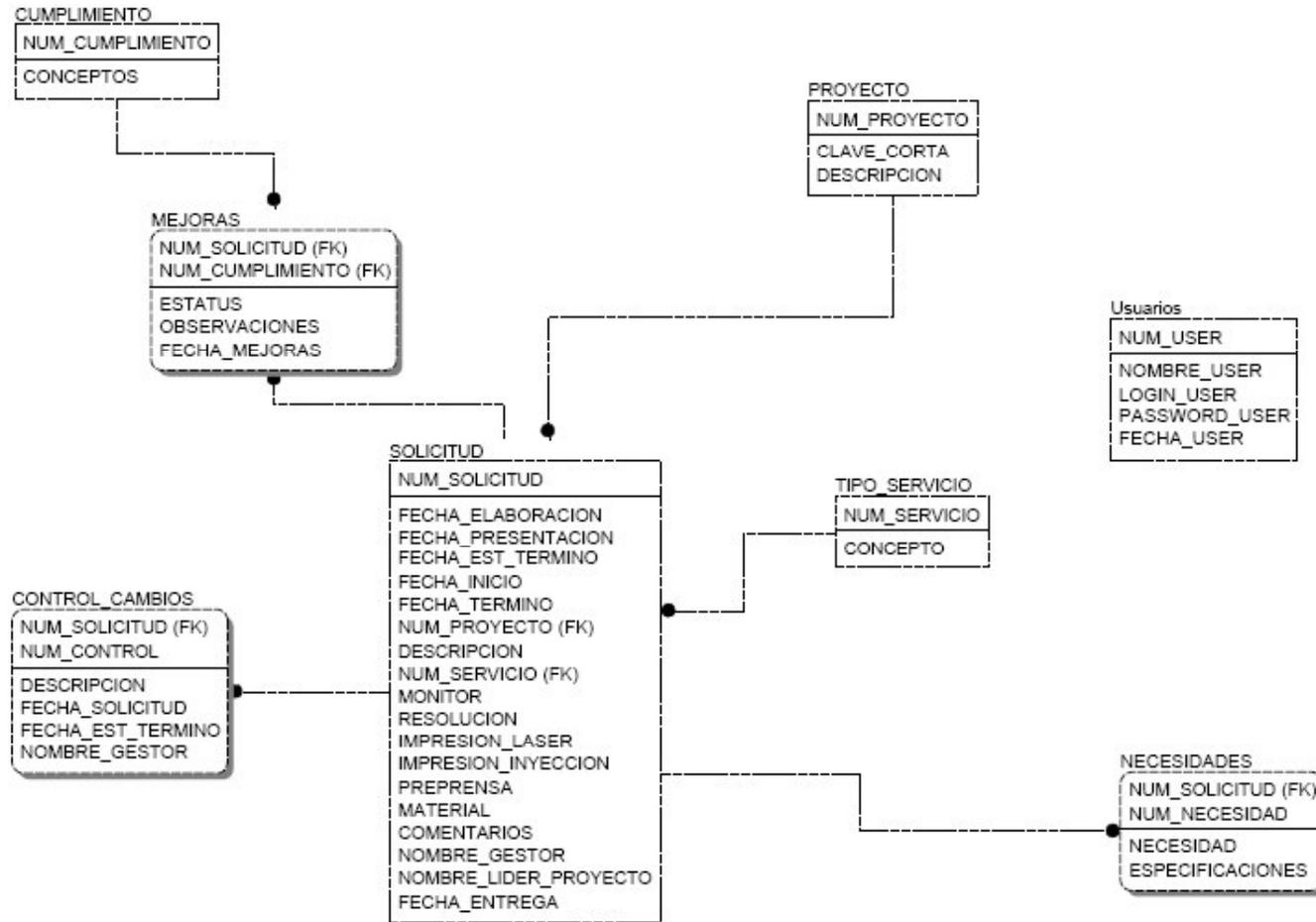
En este menú usted puede elegir entre las siguientes opciones:

- ◆ Nueva Solicitud.
- ◆ Avances de Solicitud. Menú que permite realizar:
  - Consulta. Permite consultar el nivel de progreso que lleva una solicitud.
  - Administración. Menú que permite realizar cualquiera de las siguientes operaciones:
    - Consulta. Permite consultar el nivel de progreso que lleva una solicitud.
    - Cambio. Permite realizar cambios a los datos de las solicitudes dadas de alta.
    - Terminar una Solicitud. Permite al administrador finalizar una solicitud.
    - Control de Cambios. Permite al administrador registrar cambios que el usuario solicita para su material gráfico.
    - Revisión de Cambios. Permite al administrador llevar un registro de los cambios solicitados por el usuario y con la posibilidad de introducir un comentario general.
    - Mejora. Permite al usuario expresar su opinión sobre el desarrollo y atención, así como manifestar posibles mejoras en el servicio.

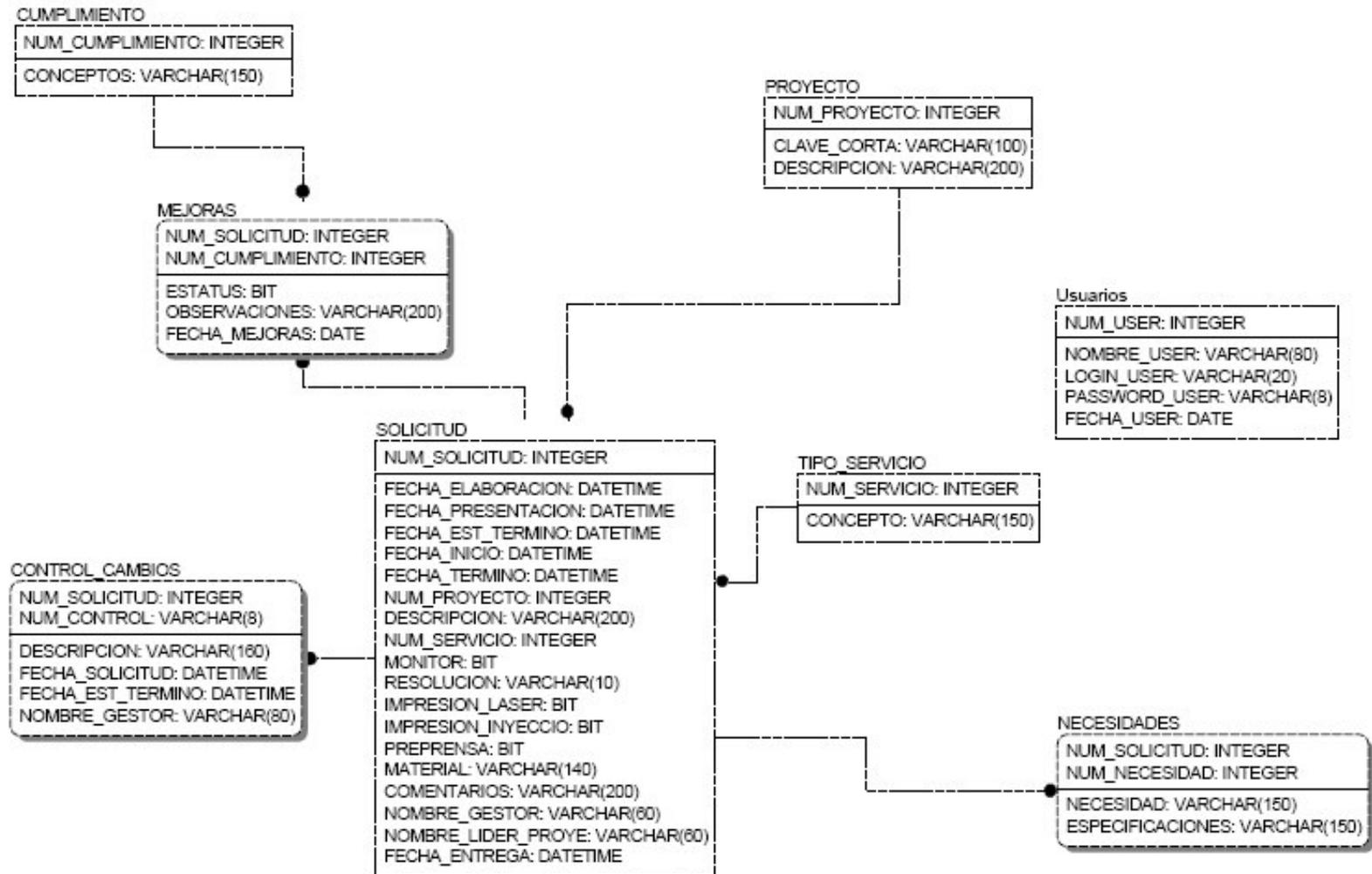
# MAPA DE MÓDULOS DEL SISTEMA



# MODELO LÓGICO DE LA BASE DE DATOS



# MODELO FISICO DE LA BASE DE DATOS



### **II.2.2 Módulos del sistema**

#### **Módulo nueva solicitud**

El objetivo del módulo consiste en registrar una solicitud que requiere de la generación de material gráfico.

Para ingresar al módulo es necesario realizar los siguientes pasos:

1. Especifique en su navegador, la dirección correspondiente al Sistema de gestión de calidad para diseño gráfico.
2. Seleccione el módulo que desea consultar y presione el botón **[Ver]** para ingresar a él:

#### **Pantalla Menú Principal**



## SOLICITUDES

ESCOGE QUE DESEAS HACER ▼
<b>ESCOGE QUE DESEAS HACER</b>
Nueva Solicitud
Avances de Solicitud
Administración

Regresar

3. Al seleccionar el Módulo se le presenta la siguiente pantalla:

## II. SISTEMA DE GESTIÓN DE LA CALIDAD PARA DISEÑO GRÁFICO

---

### Pantalla Formato



FECHA:

PROYECTO:  ▼

DESCRIPCIÓN:

TIPO DE  
SERVICIO

diseño

rediseño

elementos

consulta

SALIDA

monitor

impresión láser

impresión inyección

prerensa

RESOLUCIÓN ▼

#### NECESIDADES

1.	<input type="text"/>
2.	<input type="text"/>
3.	<input type="text"/>
4.	<input type="text"/>
5.	<input type="text"/>
6.	<input type="text"/>
7.	<input type="text"/>
8.	<input type="text"/>

#### MATERIALES PROPORCIONADOS

NOMBRE DEL LÍDER DEL PROYECTO

NOMBRE DEL GESTOR

CORREO ELECTRÓNICO

## II. SISTEMA DE GESTIÓN DE LA CALIDAD PARA DISEÑO GRÁFICO

---

4. Ingrese los datos solicitados en el formulario y presione el botón **[Enviar Solicitud]** para dar de alta.

### 4.1 Datos.

CAMPO	DESCRIPCIÓN
Proyecto	Este es un menú desplegable, contiene el catálogo de los proyectos que actualmente pueden solicitar un trabajo de diseño gráfico.
Descripción	En este campo el usuario escribirá una breve descripción del trabajo que desea.
Tipo de Servicio	El usuario utilizando estos cuadro de selección única para especificar que tipo de servicio desea. <ul style="list-style-type: none"><li>◆ Diseño: Propuestas de disposición de elementos gráficos a partir de las necesidades de un proyecto.</li><li>◆ Rediseño: Propuestas de disposición de elementos gráficos a partir de un diseño previo del que se conservarán algunas características.</li><li>◆ Elementos: Creación de elementos gráficos para integración en proyectos en desarrollo.</li><li>◆ Consulta: Asesoría relacionada con el manejo de imágenes, software o equipo de producción gráfica.</li></ul>
Salida	Utilizando estos cuadros de verificación, el usuario va a especificar que tipo de salida desea: <ul style="list-style-type: none"><li>◆ Monitor: Los gráficos a elaborar tienen como fin principal ser presentados en WEB, proyector o sistemas en red.</li><li>◆ Resolución: Se escoge la resolución del monitor en la que se va a mostrar el material gráfico.</li><li>◆ Láser: Los gráficos a elaborar tienen como fin principal crear documentos análogos destinados a imprimirse en láser.</li><li>◆ Inyección: Los gráficos a elaborar tienen como fin principal crear documentos análogos que deben ser impresos con las ventajas de los dispositivos de inyección de tinta.</li><li>◆ Preprensa: Los materiales gráficos a elaborar tienen un destino de proceso externo destinados a prensa.</li></ul>
Necesidades	En esta parte de la página el usuario podrá definir características específicas del material gráfico que solicita
Materiales Proporcionados	En esta sección el usuario deberá especificar el material que le proporcionará al Diseñador para realizar el material gráfico.
Datos Solicitante	Para llevar un control de las solicitudes se le pide que llene algunos campos con información personal: <ul style="list-style-type: none"><li>◆ Nombre del Líder de Proyecto.</li><li>◆ Nombre del Gestor: Nombre del solicitante, puede o no ser el Líder de proyecto.</li><li>◆ Correo Electrónico</li></ul>

5. Número de Solicitud: Esta pantalla es importante porque en ella aparece el número de solicitud que le corresponde, guarde este número ya que se le solicitara cada que requiera revisar el avance de su solicitud o realizar algún cambio en ella. Presione el botón **[Aceptar]** para volver al menú principal.

**Pantalla de confirmación**



FECHA:

**SU SOLICITUD HA SIDO COMPLETADA**

SU NÚMERO DE SOLICITUD ES: 3

Recuerde o anote su número de solicitud para revisar sus avances.

Aceptar

**Módulo avance de solicitud**

El objetivo del módulo consiste en visualizar los avances de una solicitud ya registrada.

Para ingresar al módulo es necesario realizar los siguientes pasos:

1. Especifique en su navegador, la dirección correspondiente al Sistema de gestión de calidad para diseño gráfico.
2. Seleccione el módulo que desea consultar y presione el botón **[Ver]** para ingresar a él:

**Pantalla Menú Principal**



**SOLICITUDES**

ESCOGE QUE DESEAS HACER ▼

ESCOGE QUE DESEAS HACER

- Nueva Solicitud
- Avances de Solicitud
- Administración

Regresar

## II. SISTEMA DE GESTIÓN DE LA CALIDAD PARA DISEÑO GRÁFICO

---

3. Al seleccionar el Módulo se le presenta la siguiente pantalla:

### Pantalla Número de Solicitud



# SOLICITUDES

FECHA: / /

SOLICITUD:

4. Ingrese el número de solicitud que desea ver y presione el botón **[Ver]** para hacer la consulta. Para regresar al menú anterior presione el botón **[Regresar]**.

5. Al confirmar el número de la solicitud, aparece un formulario de sólo lectura en el que se muestran los datos de la solicitud, y los cambios que le ha hecho el Diseñador. Presione el botón **[Regresar]** para regresar a la pantalla anterior.

### Pantalla Formulario Sólo Lectura



Nº de SOLICITUD: 2

FECHA: / /

PRESENTACIÓN DE PROPUESTAS: / /

FECHA ESTIMADA DE TERMINO: / /

FECHA DE INICIO:	18/11/2004		
------------------	------------	--	--

PROYECTO: Credencialización SEGOB

DESCRIPCIÓN:

Crear un nuevo Logo.

### **Módulo administración**

El objetivo del módulo consiste en administrar las solicitudes y realizar las operaciones de actualización, finalización y control de cambios.

Para ingresar al módulo es necesario realizar los siguientes pasos:

1. Especifique en su navegador, la dirección correspondiente al Sistema de gestión de calidad para diseño gráfico.
2. En el Sistema de gestión de calidad para diseño gráfico y seleccione el módulo de Administración y presione el botón **[Ver]**.

### Pantalla Menú Principal



## SOLICITUDES

ESCOGE QUE DESEAS HACER ▼

ESCOGE QUE DESEAS HACER

- Nueva Solicitud
- Avances de Solicitud
- Administración

Regresar

3. Al Seleccionar el Módulo se le presenta la siguiente pantalla, en donde tiene que proporcionar un login y un password válidos y presione el botón **[Aceptar]** para ingresar al módulo. Para regresar al menú anterior presione el botón **[Regresar]**.

### Pantalla Seguridad



## PROYECTO

FECHA: / /

LOGIN:

PASSWORD:

Recuerda, el login y el password son sensibles a mayúsculas y minúsculas.

Aceptar

Regresar

4. Al confirmar el login y password, aparece la siguiente pantalla un menú con las siguientes opciones:

- Avances Solicitud
- Actualizar Solicitud
- Terminar una solicitud
- Control de Cambios
- Revisar Cambios

**Pantalla Menú**



**SOLICITUDES**

ESCOGE QUE DESEAS HACER ▾
<b>ESCOGE QUE DESEAS HACER</b>
Avances Solicitud
Actualizar Solicitud
Terminar una solicitud
Control de Cambios
Revisar Cambios

***Submódulo avance solicitud***

Para ingresar al módulo es necesario realizar los siguientes pasos:

1. Al seleccionar el Módulo respectivo se le presenta la siguiente pantalla:

**Pantalla Menú Solicitud**



**SOLICITUDES**

FECHA:

SOLICITUD: 

ELIJA LA SOLICITUD QUE DESEA ▾
<b>ELIJA LA SOLICITUD QUE DESEA</b>
2
3
4

Regresar

2. Seleccione el número de solicitud que desea consultar y presione el botón **[Ver]** para ingresar a él o presione el botón **[Regresar]** para volver a la pantalla anterior:

**Pantalla Número de Solicitud**



# SOLICITUDES

FECHA:

SOLICITUD:

- ELIJA LA SOLICITUD QUE DESEA
- 2
- 3
- 4

Regresar

3. Al confirmar el número de la solicitud, aparece un formulario de sólo lectura en el que se muestran los datos de la solicitud, y los cambios que le ha hecho el Diseñador. Presione el botón **[Regresar]** para regresar a al pantalla anterior.

**Pantalla Formulario Sólo Lectura**

NECESIDADES No utilizar colores muy vivos	ESPECIFICACIONES Colores Opacos
MATERIALES PROPORCIONADOS	
Logo anterior	
Líder NOMBRE DEL LÍDER DEL PROYECTO	Gestor NOMBRE DEL GESTOR
COMENTARIOS Sin Comentarios	
Regresar	

**Submódulo actualizar solicitud**

Para ingresar al módulo es necesario realizar los siguientes pasos:

1. Al seleccionar el Módulo respectivo se le presenta la siguiente pantalla:

**Pantalla Número de Solicitud**



# SOLICITUDES

FECHA:

SOLICITUD:

- ELIJA LA SOLICITUD QUE DESEA
- 2
- 3
- 4

2. Seleccione el número de solicitud que desea actualizar y presione el botón **[Ver]** para ingresar a él o presione el botón **[Regresar]** para volver a la pantalla anterior:

3. Al seleccionar el Módulo se le presenta la siguiente pantalla:

## II. SISTEMA DE GESTIÓN DE LA CALIDAD PARA DISEÑO GRÁFICO

### Pantalla Formulario



Nº c  
Solicitud

FECHA: /:

PRESENTACIÓN DE PROPUESTAS: [v] [v] [v]

FECHA ESTIMADA DE TERMINO: [v] [v] [v]

FECHA DE INICIO: [v] [v] [v]

PROYECTO: [v]

DESCRIPCIÓN:

TIPO DE SERVICIO    diseño     rediseño     elementos     consulta

SALIDA    monitor     impresión láser     impresión inyección     prerensa

Resolución: [1152x864 v]

NECESIDADES

no

ESPECIFICACIONES


MATERIALES PROPORCIONADOS

[v]  
NOMBRE DEL LÍDER DEL PROYECTO  
[v]  
CORREO ELECTRÓNICO

[v]  
NOMBRE DEL GESTOR

COMENTARIOS

[Actualizar Solicitud]  
[Cancelar]

## II. SISTEMA DE GESTIÓN DE LA CALIDAD PARA DISEÑO GRÁFICO

4. Ingrese los datos solicitados en el formulario y presione el botón **[Actualizar Solicitud]** para actualizar la solicitud.

CAMPO	DESCRIPCIÓN
Presentación De Propuestas	Esta fecha la establece el diseñador del material gráfico para presentar sus ideas o conceptos para el trabajo.
Fecha Estimada De Término	Esta fecha la establece el diseñador a consideración de la complejidad del trabajo, el número de solicitudes en espera y de su destreza.
Fecha De Inicio	Esta fecha la establece el diseñador, indica cuando inicio el desarrollo del material gráfico.
Proyecto	Este es un menú desplegable, contiene el catálogo de los proyectos que actualmente pueden solicitar un trabajo de diseño gráfico.
Descripción	En este campo el usuario escribirá una breve descripción del trabajo que desea.
Tipo de Servicio	El usuario utilizando estos cuadro de selección única para especificar que tipo de servicio desea. <ul style="list-style-type: none"> <li>◆ Diseño: Propuestas de disposición de elementos gráficos a partir de las necesidades de un proyecto.</li> <li>◆ Rediseño: Propuestas de disposición de elementos gráficos a partir de un diseño previo del que se conservarán algunas características.</li> <li>◆ Elementos: Creación de elementos gráficos para integración en proyectos en desarrollo.</li> <li>◆ Consulta: Asesoría relacionada con el manejo de imágenes, software o equipo de producción gráfica.</li> </ul>
Salida	Utilizando estos cuadros de verificación, el usuario va a especificar que tipo de salida desea: <ul style="list-style-type: none"> <li>◆ Monitor: Los gráficos a elaborar tienen como fin principal ser presentados en WEB, proyector o sistemas en red. <ul style="list-style-type: none"> <li>○ Resolución: Se escoge la resolución en la que se va a mostrar el material gráfico.</li> </ul> </li> <li>◆ Láser: Los gráficos a elaborar tienen como fin principal crear documentos análogos destinados a imprimirse en láser.</li> <li>◆ Inyección: Los gráficos a elaborar tienen como fin principal crear documentos análogos que deben ser impresos con las ventajas de los dispositivos de inyección de tinta.</li> <li>◆ Prerensa: Los materiales gráficos a elaborar tienen un destino de proceso externo destinados a prensa.</li> </ul>
Necesidades	En esta parte de la página el usuario podrá definir características específicas del material gráfico que solicita.
Especificaciones	En esta parte el Diseñador pondrá observaciones acerca de las necesidades de los usuarios.
Materiales Proporcionados	En esta sección el usuario deberá especificar el material que le proporcionará al Diseñador para realizar el material gráfico.
Datos Solicitantes	Para llevar un control de las solicitudes se le pide que llene algunos campos con información personal: <ul style="list-style-type: none"> <li>◆ Nombre del Líder de Proyecto.</li> <li>◆ Nombre del Gestor: Nombre del solicitante, puede o no ser el Líder de proyecto.</li> <li>◆ Correo Electrónico.</li> </ul>
Comentario	En este campo el diseñador podrá introducir un comentario acerca de la solicitud.

5. Al confirmar la actualización se le presenta la siguiente pantalla, presione el botón **[Aceptar]** para regresar al menú:

### Pantalla Termino de Solicitud



FECHA:

LA SOLICITUD HA SIDO ACTUALIZADA

Aceptar

### Submódulo terminar solicitud

Para ingresar al módulo es necesario realizar los siguientes pasos:

1. Al seleccionar el Módulo respectivo se le presenta la siguiente pantalla:

### Pantalla Número de Solicitud



# SOLICITUDES

FECHA:

SOLICITUD:

ELIJA LA SOLICITUD QUE DESEA
2
3
4

Regresar

2. Seleccione el número de solicitud que desea consultar y presione el botón **[Ver]** para ingresar a él o presione el botón **[Regresar]** para volver a la pantalla anterior:

3. Al confirmar el número de la solicitud, aparece un formulario de sólo lectura en el que se muestran los datos de la solicitud. Para terminar la solicitud presione el botón **[Terminar]**, para regresar a la pantalla anterior presione el botón **[Regresar]**.

### Pantalla Formulario Solo Lectura

MATERIALES  
PROPORCIONADOS

Logo anterior

Líder  
NOMBRE Y FIRMA DEL LÍDER DEL PROYECTO

Gestor  
NOMBRE Y FIRMA DEL GESTOR

COMENTARIOS  
Sin Comentarios

Terminar Solicitud

Regresar

4. Al confirmar, se le presenta la pantalla de confirmación, presione el botón **[Aceptar]** para regresar al menú.

### Submódulo control de cambios

Para ingresar al módulo es necesario realizar los siguientes pasos:

1. Al seleccionar el Módulo respectivo se le presenta la siguiente pantalla:

### Pantalla Número de Solicitud



SOLICITUD: ELIJA LA SOLICITUD QUE DESEA ▾

ELIJA LA SOLICITUD QUE DESEA

2

3

4

Regresar

2. Seleccione el número de solicitud que desea agregar un cambio y presione el botón **[Ver]** para ingresar a él o presione el botón **[Regresar]** para volver a la pantalla anterior:

3. Al confirmar el número de la solicitud, aparece un formulario en el que se tiene que capturar los cambios requeridos. Para registrar el cambio presione el botón **[Guardar Cambio]** o presione el botón **[Regresar]** para volver a la pantalla anterior.

- Fecha Estimada de Terminación: Esta fecha la establece el realizador a consideración de la complejidad del trabajo, el número de solicitudes en espera y de su destreza.
- Descripción: En este campo se especificará la naturaleza del cambio, así como las características y observaciones que el usuario tenga del trabajo que desea.
- Nombre del gestor: Nombre del solicitante del cambio.

### Pantalla Formulario



**CONTROL DE CAMBIOS**

Cambio Nº = 2

FECHA DE SOLICITUD: 23/NOVIEMBRE/2004      FECHA EST. DE TERMINO: 26 Noviembre 2004

--

NOMBRE DEL GESTOR

7. Al confirmar, se presenta la pantalla de confirmación, presione el botón **[Aceptar]** para regresar al menú.

### **Submódulo revisar cambios**

Para ingresar al módulo es necesario realizar los siguientes pasos:

1. Al seleccionar el Módulo respectivo se le presenta la siguiente pantalla:

### Pantalla Número de Solicitud



## SOLICITUDES

FECHA:

SOLICITUD:

2
3
4

2. Seleccione el número de solicitud que desea consultar los cambios registrados y presione el botón **[Ver]** para ingresar a él o presione el botón **[Regresar]** para volver a la pantalla anterior:

3. Al confirmar el número de la solicitud, aparece una tabla mostrando los registros de los cambios, además muestra un campo de comentario general. Para registrar el comentario general el botón **[Actualizar]**, para regresar a la pantalla anterior presione el botón **[Regresar]**.

**Pantalla Tabla**



Nº de Solicitud:  
3

**CONTROL DE CAMBIOS**

Número del Cambio	Descripción	Fecha Inicio	Fecha Estimada de Término	Nombre del Gestor
1	Colores Serios.	29/11/2004	02/12/2004	Saidd Cosio

COMENTARIOS:

Actualizar

Regresar

4. Al confirmar, se le presenta la pantalla de confirmación, presione el botón **[Aceptar]** para regresar al menú.

**Módulo mejora**

El objetivo del módulo permite al usuario expresar su opinión sobre el desarrollo y atención, así como manifestar posibles mejoras en el servicio.

Para ingresar al módulo debe contar la dirección correspondiente, ésta es proporcionada por el diseñador:

1. Especifique en su navegador, la dirección correspondiente al módulo de mejoras.
2. Enseguida se presenta la siguiente pantalla:

**Pantalla Formulario**



Nº de Solicitud:  
2

**QUEREMOS BRINDARTE UN MEJOR SERVICIO**

CONCEPTO	CUMPLIÓ	NO CUMPLIÓ	OBSERVACIONES
ENTREGA EN TIEMPO ACORDADO	<input checked="" type="radio"/>	<input type="radio"/>	
EL PRODUCTO CUBRE LAS NECESIDADES	<input checked="" type="radio"/>	<input type="radio"/>	
IMPACTO FAVORABLEMENTE EN EL PROYECTO	<input checked="" type="radio"/>	<input type="radio"/>	
ATENCIÓN DEL COORDINADOR SATISFACTORIA	<input checked="" type="radio"/>	<input type="radio"/>	
CANALES DE COMUNICACIÓN ADECUADOS	<input checked="" type="radio"/>	<input type="radio"/>	
COMENTARIOS ADICIONALES			

Enviar Mejoras

3. Ingrese los datos solicitados en el formulario y presione el botón **[Enviar Mejoras]** para registrar sus opiniones.

## CONCLUSIONES

Ante el azote de las grandes corporaciones que cobran exorbitantes precios por utilizar sus productos y que estos no se pueden personalizar para las necesidades que uno tiene, el software libre esta tomando gran importancia. Entre las principales ventajas de utilizar software libre se encuentran su alto desempeño, su gran adaptabilidad a diferentes necesidades y tipos de organizaciones, su gran respaldo tanto de desarrollo como de soporte y sobre todo a su bajo costo, que permite que empresas pequeñas que no tenían la capacidad de pagar licencias puedan tener una ventaja competitiva.

Reduciendo costos, proporcionando lenguajes de programación portable, permitiendo que una aplicación sea modificada para cubrir las necesidades del cliente, el programador ya no esta limitado a desarrollar en plataformas cerradas, limitando el campo de acción y restringiendo a la aplicación a utilizar software propietario.

El software libre ha demostrado que puede competir con el software comercial de par a par, y eso ha sido un gran aliciente para que Internet alcance la popularidad que actualmente cuenta. En la programación en Web se tienen que tomar muchas consideraciones, desde la funcionalidad y el diseño hasta la seguridad, conocimientos con los que yo contaba pero de forma limitada, el diplomado abrió mi espectro de posibilidades.

En lo referente a mí, el software libre me ha brindado una gran herramienta para el trabajo que realizo, el desarrollo de aplicaciones Web, más específicamente tiendas electrónicas o e-commerce. El comercio electrónico ha tenido un gran auge en los últimos años y el software libre ha dado oportunidad de crecimiento que antes era imposible.

El conocimiento que tenía de LINUX, aunque suficiente para desempeñar mi trabajo, era limitado. El diplomado ha tenido un gran impacto en mi desempeño por que ya puedo realizar tareas más avanzadas, desde administración del servidor hasta medidas de seguridad.

Los módulos de administración y seguridad son los que más he utilizado, por que me brindaron herramientas para tener un buen manejo del servidor donde están alojadas las aplicaciones Web, y el de seguridad por que me mostró las diferentes formas de ataques que se pueden realizar para bloquear una aplicación.

El diplomado me proporcionó las herramientas para mejorar mi capacidad de respuesta a diferentes situaciones, con el módulo de Apache entendí la importancia de una buena configuración, y me proporcionó el conocimiento para reiniciar el servicio ante un ataque, en el módulo de PostgreSQL me permitió conocer un manejador de base de datos del que tenía gran interés, además de proporcionarme una aplicación que permite cambiar de manejador de base de datos sin modificar código fuente. Cada módulo del diplomado me proporcionó el conocimiento para mejorar mi desempeño y ampliar mi campo de trabajo y mi experiencia laboral.

## ANEXOS

### ANEXO I. Comandos de vi

#### Modo texto

Se emplea para añadir o modificar el texto del documento. Los comandos que utiliza son:

#### Acceso a modo texto

Comando	Descripción	Comando	Descripción
i	Insertar antes del cursor.	I	Insertar al principio de la línea.
a	Añadir después del cursor.	A	Añadir al final de la línea.
o	Abrir línea debajo de la actual.	O	Abrir línea encima de la actual.
R	Sobrescribir (cambiar) texto.	r	Sobrescribir el carácter sobre el que está el cursor.
c	Reemplaza caracteres.	cw	Reemplaza palabras.
cw	Reemplaza palabras.	C o c\$	Reemplaza hasta el fin de línea.
c0	Reemplaza desde el comienzo de línea.		

Los comandos de reemplazo admiten multiplicadores que consisten en un número antecediendo al comando. Al dar un comando de reemplazo el editor coloca un símbolo \$ en donde termina el pedido de reemplazo. El usuario sobrescribe normalmente hasta donde necesite, y sale con la tecla ESC. Estos comandos admiten multiplicadores, por ejemplo 3cw abre un área de reemplazo para 3 palabras.

#### Otros

Comando	Descripción
BS (Back Space)	Borrar carácter hacia la izquierda.
ESC	Salir de Modo Texto y pasar a modo comando.

#### Modo comando

#### Movimiento del cursor y Control de pantalla

Comando	Descripción	Comando	Descripción
flechas	Mover en distintas direcciones.	G	Fin del archivo.
h ó BS (Back Space)	Una posición hacia la izquierda.	Ctrl-G	Mostrar número de línea actual.
l ó SP (Espacio)	Una posición hacia la derecha.	w	Comienzo de la palabra siguiente.
k ó -	Una línea hacia arriba.	e	Fin de la palabra siguiente.
j ó +	Una línea hacia abajo.	E	Fin de la palabra siguiente antes de espacio.
\$	Fin de línea.	b	Principio de la palabra anterior.
0 (Cero)	Principio de línea.	^	Primera palabra de la línea.
1G	Comienzo del archivo.	%	Hasta el paréntesis que aparece.
Ctrl-u	Media pantalla atrás.	H	Parte superior de la pantalla.
L	Parte inferior de la pantalla.	M	Al medio de la pantalla.
23	Cursor a la columna 23.	Ctrl-f	Una pantalla adelante.
Ctrl-b	Una pantalla atrás.	Ctrl-l	Refrescar la pantalla.
Ctrl-d	Media pantalla adelante.		

## Búsqueda

Comando	Descripción	Comando	Descripción
/str	Buscar hacia adelante cadena de caracteres 'str'.	?str	Buscar hacia atrás cadena de caracteres 'str'.
n	Buscar siguiente (si se usó /) o anterior (si se usó ?).	N	Buscar anterior (si se usó /) o siguiente (si se usó ?).
f c	Buscar el siguiente carácter 'c' en la línea.	F c	Buscar el anterior carácter 'c' en la línea.
t c	Ir al carácter anterior al siguiente 'c'.	T c	Ir al carácter posterior al precedente 'c'.
;	Repetir el último comando f, F, t, o T.	,	último comando f, F, t, o T en sentido inverso.

La cadena a buscar en / o ? puede ser una expresión regular.

La acción de f, F, t y T alcanza sólo a la línea actual; si el carácter buscado no está en esa línea el cursor no se mueve.

## Borrar, Copiar y Pegar

Comando	Descripción	Comando	Descripción
x	Borrar carácter bajo el cursor.	dd	Borrar línea, queda guardada.
D	Borrar desde cursor a fin de línea.	dw	Borrar desde cursor a fin de palabra.
d\$	Borrar desde cursor a fin de línea.	d0	Borrar desde cursor a principio de línea.
d)	Borra hasta el final del párrafo.	Y o yY	Copiar línea.
P	Pegar antes del cursor.	p	Pegar después del cursor.
yw	Copiar palabra.	y\$	Copiar de cursor a fin de línea.
"aY o "aY	Copiar línea en buffer llamado 'a'.	'a' "ayw	Copiar palabra en buffer llamado 'a'.
"ap	Pegar desde buffer 'a', a la derecha del cursor.	"aP	Pegar desde buffer 'a', a la izquierda del cursor.
"bdd	Borrar línea y guardar en buffer 'b'.	"bdw	Borrar palabra y guardar en buffer 'b'.
J	Unir la línea actual y la siguiente (suprimir el retorno de carro).		

## Otros

Comando	Descripción
ZZ	Grabar cambios si los hubo y salir.
u	Deshacer última acción.
U	Deshacer todos los cambios en una línea desde que se posicionó el cursor en ella.
.	Repetir el último comando.

**Modo ex o ultima línea****Generales**

Comando	Descripción	Comando	Descripción
:q	Salir si no hubo cambios.	:q!	Salir sin guardar cambios.
:w	Guardar cambios.	:w arch1	Guardar cambios en archivo arch1.
:wq	Guardar cambios y salir.	:r arch2	Insertar un archivo.
:e arch2	Editar un nuevo archivo.	:e! arch2	Idem sin salvar anterior.
:r! comando	Insertar salida de comando.	:shell	Salir al shell (vuelve con exit).
:=	Muestra el número de línea en que se halla en cursor.		

**Mover**

Comando	Descripción
:1	Mueve a línea 1.
:15	Mueve a línea 15.
:\$	Mueve a última línea.

**Opciones, abreviaturas y macros**

Comando	Descripción	Comando	Descripción
:set	Cambio de opciones.	:set nu	Mostrar números de línea.
:set nonu	No mostrar números de línea .	:set showmode	Mostrar modo actual de vi.
:set noshowmode	No mostrar modo actual de vi.	:ab [abr] [cadena]	Cada vez que se teclee la cadena abr, ésta se expandirá escribiendo cadena.
:ab	Muestra las abreviaturas creadas.	:una [abr]	Elimina la abreviatura abr.
:map [c] [accion]	Genera la macro c que ejecutará la acción indicada.	:map	Muestra las macros definidas.

**Otros**

Comando	Descripción
:[desde],[hasta] s/[buscar]/[reemplazar]/	Reemplaza todas la primera ocurrencia de la cadena [buscar], en cada línea, por la cadena [reemplazar] entre las líneas [desde] y [hasta] del archivo.
:[desde],[hasta] s/[buscar]/[reemplazar]/g	Reemplaza todas las ocurrencias de la cadena [buscar] por la cadena [reemplazar] entre las líneas [desde] y [hasta] del archivo.
:[desde],[hasta] d	Borra las líneas comprendidas entre las líneas [desde] y [hasta].

**ANEXO II. Tipos de datos de PostgreSQL**

Tipo	Descripción	Tipo	Descripción
Abstime	fecha y hora absoluta de rango limitado (Unix system time)	oid	tipo de identificación de objetos
Aclitem	lista de control de acceso	oid8	arreglo de 8 oids, utilizado en tablas del sistema
Bool	booleano 'true'/'false'	path	trayectoria geométrica abierta o cerrada en el plano '(pt1, ...)'
Box	rectángulo geométrico '(izquierda abajo, derecha arriba)'	point	punto geométrico '(x, y)'
Bpchar	caracteres rellenos con espacios, longitud especificada al momento de creación	polygon	Trayectoria geométrica cerrada en el plano '(pt1, ...)'
Bytea	arreglo de bytes de longitud variable	regproc	procedimiento registrado
Char	un sólo carácter	reltime	intervalo de tiempo de rango limitado y relativo (Unix delta time)
cid	command identifier type, identificador de secuencia en transacciones	smgr	manejador de almacenamiento (storage manager)
Cidr	dirección de red	serial	identificador numerico único
circle	círculo geométrico '(centro, radio)'	text	cadena de caracteres nativa de longitud variable
date	fecha ANSI SQL 'aaaa-mm-dd'	tid	tipo de identificador de tupla, localización física de tupla
datetime	fecha y hora 'aaaa-mm-dd hh:mm:ss'	time	hora ANSI SQL 'hh:mm:ss'
filename	nombre de archivo usado en tablas del sistema	timespan	intervalo de tiempo '@ <number> <units>'
float4	número real de precisión simple de 4 bytes	timestamp	fecha y hora en formato ISO de rango limitado
float8	número real de precisión doble de 8 bytes	Tinterval	intervalo de tiempo '(abstime, abstime)'
inet	dirección de red	Unknown	tipo desconocido
int2	número entero de dos bytes, de -32k a 32k	Varchar()	cadena de caracteres sin espacios al final, longitud especificada al momento de creación
int28	8 numeros enteros de 2 bytes, usado internamente	Xid	identificador de transacción
int4	número entero de 4 bytes, -2B to 2B		
int8	número entero de 8 bytes, 90#9018 dígitos		
line	línea geométrica '(pt1, pt2)'		
lseg	segmento de línea geométrica '(pt1, pt2)'		
macaddr	dirección MAC		
money	unidad monetaria '\$d,ddd.cc'		
name	tipo de 31 caracteres para guardar identificadores del sistema		
numeric	número de precisión múltiple		

## REFERENCIAS

### *Bibliografía*

- ◆ Ratschiller, Tobias. *Creación de aplicaciones web con PHP 4*, Prentice may, España, 2000.
- ◆ Blanco J., Vicente. *Linux, instalación, administración y uso del sistema*, Omega, Colombia, 1992.
- ◆ Welling, Luke. *PHP and MySQL web Development*, Sams Publishing, USA, 2001.
- ◆ Boach, Rich. *Servidor Apache al descubierto*, Prentice may, España, 2000

### *Referencias Electrónicas*

- ◆ Marjedan, Alejo. *La seguridad de la Información* [En línea], Marzo 2002. Disponible en Web: [http://alerta-antivirus.red.es/seguridad/ver\\_pag.html?tema=S&articulo=4&pagina=1](http://alerta-antivirus.red.es/seguridad/ver_pag.html?tema=S&articulo=4&pagina=1)
- ◆ Revista RED. *Seguridad Informática, ¿qué?, ¿por qué y para qué?* [En línea], Noviembre 2002. Disponible en Web: <http://ciberhabitat.gob.mx/museo/cerquita/redes/seguridad/intro.htm>
- ◆ Villalón Huerta, Antonio. *Seguridad En Unix Y Redes* [En línea], Julio 2002. Disponible en Web: <http://es.tldp.org/Manuales-LuCAS/doc-unixsec/unixsec-html/>
- ◆ Moreno, Moreno. *Criptografía* [En línea], Disponible en Web: [http://www.htmlweb.net/seguridad/cripto/cripto\\_9.html](http://www.htmlweb.net/seguridad/cripto/cripto_9.html)
- ◆ Gallús, Carlos. *Artículos sobre PHP* [En línea], Agosto 2002, Disponible en Web: <http://www.webestilo.com/php/articulos.phtml>
- ◆ Achour, Mehdi. *Manual de PHP* [En línea], Marzo 2005, Disponible en Web: <http://mx2.php.net/manual/es/index.php>
- ◆ Laboratorio Nacional de Informática Avanzada, A.C. (LANIA). *Iniciación al Lenguaje SQL* [En línea]. Disponible en Web: <http://www.lania.mx/biblioteca/seminarios/basedatos/sql.html>