



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE ESTUDIOS SUPERIORES

CAMPUS ARAGÓN

## *INGENIERÍA EN COMPUTACIÓN*

“Teoría, práctica y aplicación de diversos  
tópicos relacionados con el diseño de sistemas”

PROYECTO BAJO LA MODALIDAD DE  
“SEMINARIOS Y CURSOS DE ACTUALIZACIÓN  
Y CAPACITACIÓN PROFESIONAL”

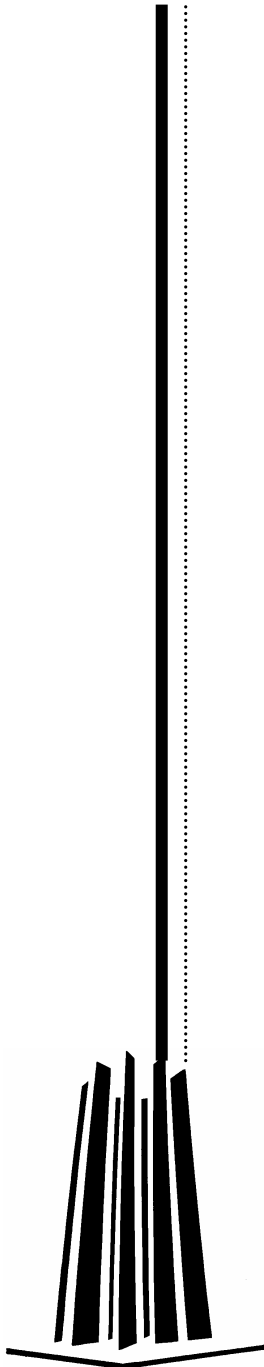
QUE PARA OBTENER EL TÍTULO DE

“INGENIERO EN COMPUTACIÓN”

PRESENTA:

**Rodríguez González María del Rosario**

Dirigido por la M. en E. Imelda de la Luz Flores Díaz





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.

Quiero agradecer a mis padres Alberto y Rosario por el apoyo y cariño que me brindaron para poder terminar mi carrera.

A mis hermanos Valeria y Alberto que me dieron ánimos para seguir siempre adelante y les agradezco sus consejos.

Un agradecimiento especial a Christian por estar siempre a mi lado, ayudarme y apoyarme a lo largo de la carrera. Así como por compartir los momentos difíciles y los momentos felices.

A mis amigos Vero y Emmanuel con quienes disfrute momentos muy padres en la escuela y fuera de ella.

## Índice

<b>Introducción.....</b>	<b>vi</b>
<b>Capítulo I: Linux 1.....</b>	<b>1</b>
1.1 Componentes.....	2
1.2 Sistema de archivos.....	3
1.3 Distribuciones.....	3
1.4 Instalación de Slackware.....	4
1.5 Discos.....	9
1.6 Administración.....	10
1.7 Inicio del sistema.....	10
1.8 Usuarios.....	11
1.9 Comandos.....	11
1.10 Redireccionamiento.....	14
1.11 Permisos.....	15
1.12 Procesos.....	17
1.13 Conclusión.....	17
<b>Capítulo II: HTML.....</b>	<b>18</b>
2.1 Lenguaje HTML.....	19
2.2 Etiquetas más importantes.....	20
2.3 Caracteres especiales.....	21
2.4 Otros elementos.....	22
2.5 Formularios.....	24
2.6 Frames.....	26
2.7 Comandos META.....	26
2.8 Conclusión.....	26
<b>Capítulo III: Apache.....</b>	<b>27</b>
3.1 Características de Apache.....	28
3.2 Descarga e instalación.....	28
3.3 Directivas.....	29
3.4 Virtual Host de Apache.....	32
3.5 Control de Acceso.....	33
3.6 Módulos.....	34
3.7 Conclusión.....	35
<b>Capítulo IV: PHP.....</b>	<b>36</b>
4.1 Descarga e instalación.....	36
4.2 Características del lenguaje PHP.....	38
4.3 Código PHP.....	38
4.4 Variables.....	39
4.5 Operadores.....	40
4.6 Estructuras de control.....	41
4.7 Funciones.....	43

4.8	Archivos.....	44
4.9	Conclusión.....	45
<b>Capítulo V: Diseño de bases de datos.....</b>		<b>46</b>
5.1	Tipos de bases de datos.....	47
5.2	Conceptos asociados a las bases de datos relacionales.....	48
5.3	Pasos implicados para la construcción de una base de datos.....	49
5.4	Normalización.....	50
5.5	Tipos de relaciones.....	51
5.6	Modelo entidad – relación.....	52
5.7	Sistemas Gestores de Bases de Datos.....	53
5.8	Conclusión.....	54
<b>Capítulo VI: MySQL y PostgreSQL.....</b>		<b>55</b>
6.1	Instalación.....	56
6.2	Sentencias.....	58
6.3	Manipulación de datos.....	60
6.4	Conexiones con PHP.....	62
6.5	Conclusión.....	64
<b>Capítulo VII: Seguridad.....</b>		<b>65</b>
7.1	Seguridad informática.....	66
7.2	Monitoreo.....	67
7.3	Criptología.....	68
7.4	Contraseñas.....	70
7.5	Huella digital.....	71
7.6	Esteganografía.....	71
7.7	Respaldos.....	72
7.8	Conclusión.....	73
<b>Capítulo VIII: Edición de publicaciones.....</b>		<b>74</b>
8.1	Formatos de imagen digital.....	74
8.2	Creación de un documento.....	75
8.3	Barra de herramientas.....	76
8.4	Paletas.....	79
8.5	Capas.....	80
8.6	Filtros.....	81
8.7	Guardar para Web.....	82
8.8	Conclusión.....	83

<b>Capítulo IX Proyecto aplicativo.....</b>	<b>84</b>
9.1 Planteamiento del problema.....	84
9.2 Requerimientos específicos.....	84
9.3 Administración de discos.....	85
9.4 Base de datos.....	88
9.5 Código.....	90
<b>Conclusión.....</b>	<b>98</b>
<b>Glosario.....</b>	<b>99</b>
<b>Bibliografía.....</b>	<b>102</b>
<b>Páginas de Internet Consultadas.....</b>	<b>103</b>

## Introducción

Hoy en día las empresas necesitan desarrollar sistemas informáticos especializados para las necesidades que se les presenten, ya que, estos sistemas llegan a ser herramientas muy eficientes y que permiten llevar una mejor administración. El software para el desarrollo de sistemas llega a ser de costos muy elevados por las licencias que requiere, por esto los sistemas de software libre han tomado mucha popularidad ya que las empresas necesitan economizar costos.

La comunicación con los usuarios es muy importante, y la difusión de información es un elemento clave que las empresas necesitan cubrir. El crecimiento que ha tenido Internet es una muestra de esto, ya que, existen miles de sitios que ofrecen los servicios y productos de las empresas.

Además de la información que se publica, existen tiendas en Internet donde se pueden realizar compras de diversos artículos, portales para realizar trámites burocráticos, inscripciones a eventos, etc. Por esto no solo las empresas se benefician de Internet, también existen sitios enfocados a diversas ramas como enseñanza vía Web, turismo, medicina, información de gobierno, la cultura y las artes, etc. en general los sitios de la Web tienen en común que necesitan ser diseñados pensando en los usuarios.

Existen algo conocido como **LAMP**, que son las iniciales de Linux, Apache, MySQL y PHP, que son herramientas para el diseño de sistemas en red, con una cosa en común: todas son software libre.

En este trabajo se mostrará como utilizar estas herramientas para la elaboración de un sistema, así mismo se mencionarán sus características principales y los conceptos relacionados a éstas.

En el primer capítulo se verá una descripción general de los sistemas Linux y se enfoca principalmente a la instalación del sistema Slackware y sus comandos principales.

En el segundo capítulo se muestran las etiquetas más importantes del lenguaje HTML y sus respectivos atributos, que nos permiten desarrollar páginas Web.

En el tercer capítulo se tratan las características más importantes de Apache, así como la forma de instalarlo y sus directivas más importantes.

Para crear páginas dinámicas en el capítulo IV se estudia el lenguaje PHP, su sintaxis, la forma en que se manejan sus variables y la manera más adecuada de instalarlo.

En el capítulo V se muestran los pasos a seguir para diseñar una base de datos y se da la definición de sus elementos más importantes.

Para poder hacer que las páginas Web dinámicas interactúen con una base de datos, en el capítulo VI se muestran las sentencias necesarias para la creación de dichas bases y las sentencias que el lenguaje PHP utiliza para acceder a ellas.

En el capítulo VII se muestran los programas y las técnicas que nos permiten dar mayor seguridad a un sistema, aunque solamente se muestran algunas herramientas ya que en cuestión de seguridad hay muchas opciones.

Para hacer más vistosa y agradable una página, en el capítulo VIII se muestran las herramientas que nos permiten editar imágenes y como poder almacenarlas para poder publicarlas en la Web.

Para finalizar en el capítulo 9 se presenta un proyecto en el cuál se utilizan en conjunto las herramientas mencionadas en este trabajo.



# Capítulo I

## Linux

**Objetivo:** Dar una introducción al sistema Linux Slackware, mostrando sus características y comandos más usados, así como sus ventajas y requerimientos. Así mismo aprender a instalar el sistema y administrarlo.

**Justificación:** Para poder manejar un sistema hay que comenzar por entender cuales son sus características y comandos principales, esto es necesario para poder tener las herramientas básicas y así utilizarlo adecuadamente.

**Introducción:** Linux inicia con el desarrollo de un pequeño programa llamado Minix, un sistema operativo escrito por el científico Andrew Tannebaum, este sistema operativo ganó popularidad siendo desarrollado para poder trabajar en diferentes plataformas de equipo de cómputo, lo que inspira el desarrollo de Linux.

### Contenido

En la década de los 60's se trató de crear un sistema independiente de su arquitectura hardware, el proyecto se nombró Multics. Cuando aparece el lenguaje C se lanza la segunda versión de este sistema bajo el nombre de UNIX (aberración de la palabra Multics), que permitía portabilidad entre diferentes arquitecturas de hardware pero que es un sistema “no libre”, ya que Novell compra los derechos de este producto.

En 1990 un estudiante Finlandés de nombre Linus Torvalds inició un proyecto personal basado en Minix (un sistema tipo UNIX), su intención era desarrollar un sistema compatible con plataforma PC el cual fuera más confiable que Minix, comenzó creando drivers para dispositivos de hardware en lenguaje C mejorando su plataforma, para 1991 libera la primera versión de Linux sin ser una distribución oficial, la 0.01.

En el mismo año pone a disposición del mundo a través de Internet la primera versión oficial de Linux, la 0.02. Linux es un sistema libre de licencia GNU y cumple con las normas oficiales de UNIX.

La licencia pública general (GNU) está destinada a garantizar la libertad de compartir y cambiar el software gratis. “Para asegurar que el software GNU permaneciera libre para que todos los usuarios pudieran ‘ejecutarlo, copiarlo, modificarlo y distribuirlo’, el proyecto debía ser liberado bajo una licencia diseñada para garantizar esos derechos al tiempo que evitase restricciones posteriores de los mismos. La idea se conoce en Inglés como copyleft y está contenida en la licencia pública general (GNU).”<sup>1</sup> El sistema GNU fue diseñado para ser totalmente compatible con UNIX.

Algunas de las características de Linux son:

---

<sup>1</sup> <http://es.wikipedia.org/wiki/Gnu>

- ☞ Es un sistema operativo de tiempo compartido
- ☞ Multiusuario
- ☞ Multitarea
- ☞ Multiproceso
- ☞ Multiplataforma
- ☞ Modular
- ☞ Portable entre sistemas abiertos
- ☞ Programable
- ☞ Es un sistema tipo cliente – servidor

Algunos conceptos son: “*Multitarea*: es la capacidad que este sistema operativo tiene, de ejecutar varios programas al mismo tiempo. Linux utiliza lo que se conoce por multitarea preventiva, que permite que todos los programas que se hayan abierto sean ejecutados [...] *Multiusuario*: es decir la posibilidad de que varias personas usen una misma máquina [...], con sus recursos y aplicaciones, sin interferir entre ellas, de esta forma un determinado documento no va ser manipulado y menos borrado, por otro usuario *Multiplataforma*: se puede ejecutar un Linux no sólo con los microprocesadores típicos del mundo PC (Intel y AMD), sino que también se piensa en otras plataformas”<sup>2</sup>

Las plataformas principales de Linux son:

I386 Para equipos tipo PC compatibles.

PPC Para equipos Maquintosh

Sparc Para estaciones de trabajo

Alpha Para arquitectura de trabajo con procesadores en paralelo de tipo alpha.

## Componentes

LINUX está compuesto básicamente de cuatro capas, la capa más interna está conformada por el hardware, que es el conjunto de piezas físicas del equipo de cómputo.

La segunda capa es el “Kernel”<sup>3</sup>, su función principal es interpretar las instrucciones proporcionadas por el usuario y convertirlas en lenguaje de máquina e indicarle al hardware lo que tiene que realizar con dicha información. Las versiones del kernel se numeran con 3 números, de la siguiente forma: XX.YY.ZZ

XX indica la serie principal del kernel, YY indica si la versión es de desarrollo o de producción. Un número impar, significa que es de desarrollo, uno par, que es de producción, ZZ indica nuevas revisiones dentro de una versión. Por ejemplo la versión: 2.4.26, indica que 2 es la versión del paquete, 4 significa que es estable y 26 es el numero de revisiones hechas.

La tercer capa está conformada por el grupo de los Shells o interpretes de comandos, los cuales funcionan como la interfaz entre el usuario y el kernel, proporcionan las herramientas para que el usuario se pueda comunicar con el núcleo del sistema de LINUX.

---

<sup>2</sup> García Jiménez Francisco, “Guía de campo de Linux” México, RA-MA, pp. 9-10

<sup>3</sup> Se suele traducir como núcleo, y en general se entiende como el centro o motor de cualquier Sistema Operativo

La cuarta y última capa es donde se encuentra el usuario junto con los programas y aplicaciones que se le han agregado al sistema como hojas de cálculo, lenguajes de programación, manejadores de bases de datos, procesadores de texto, etc.

## Sistema de archivos

LINUX está compuesto de un sistema de archivos jerárquico en el cual no existen unidades de disco, en su lugar cada unidad de almacenamiento así como cada dispositivo de hardware son reconocidos como un archivo o directorio dentro del sistema.

Existe sólo una raíz en el sistema la cual representa la parte más alta de la estructura de directorios y es conocida como root o raíz, es representada por el símbolo / a partir de este punto se desprenden diferentes ramas de directorios como las siguientes:

🌀 NetWare	🌀 initrd	🌀 sbin
🌀 auto	🌀 install@	🌀 tmp
🌀 bin	🌀 lib	🌀 usr
🌀 boot	🌀 lost+found	🌀 var
🌀 bru	🌀 mnt	🌀 /var/spool/mail
🌀 dev.	🌀 opt	🌀 /var/log
🌀 etc	🌀 proc	
🌀 home	🌀 root	

**Bin:** Contiene archivos binarios esenciales del sistema, que tienen que estar disponibles para todos los usuarios.

**Boot:** Contiene los archivos que Linux necesita para arrancar

**Dev:** Contiene archivos de dispositivos del sistema

**Home:** Dentro de este directorio se crean los directorios personales de los usuarios del sistema.

**Lib:** Contiene las librerías compartidas y necesarias para arrancar el sistema y ejecutar los diferentes programas.

**Root:** Es el directorio particular del superusuario o root. Es decir la persona que tiene plenos poderes sobre todo el sistema

## Distribuciones

Existen muchas distribuciones Linux que pueden ser adquiridas por medio de Internet, haciendo la compra en el sitio oficial que las distribuye o descargándolas gratuitamente de los sitios de FTP.

Todas las distribuciones utilizan la versión del kernel que se actualiza constantemente, la diferencia radica en que los diseñadores y programadores le dan su toque especial, principalmente en presentación de la interfaz gráfica, implementación de nuevas herramientas, programas, formatos gráficos, etc. Algunas de las distribuciones más comunes son: Red Hat, Debian, Mandrake, Suse, FreeBSD, etc.

Slackware de Patrick Volkerding es un avanzado sistema operativo Linux, diseñado con dos objetivos: facilidad para usar y estabilidad, además es un sistema enfocado a servidores. Los requerimientos de Slackware son:

Hardware	Requiere
Procesador	386
RAM	16 MB – 64M
Espacio en Disco	500MB – 3G
Floppy Drive	1.44 MB
Unidad CD-ROM	
Software	(pagina oficial)

La distribución de paquetes en Slackware se hace principalmente con archivos Tgz La interfaz del programa de instalación es por texto, y necesita un mayor conocimiento de Linux que la mayoría de las otras distribuciones.

## Instalación de Slackware

Para comenzar a instalar el sistema es necesario que el equipo pueda inicializarse desde el CD-ROM. Si no se cuenta con un CD-ROM se necesitarán crear discos de arranque.

En este caso para instalar el sistema se cuenta con 2 CD's. Para comenzar hay que colocar el primer disco en la bandeja del CD-ROM y con un enter comienza la instalación. Aparecerá una pantalla de bienvenida al programa de instalación de Linux, y en la parte inferior el indicador boot<sup>4</sup> (Fig.1):

```
ISOLINUX 2.10 2004-06-10 Copyright (C) 1994-2004 H. Peter Anvin
Welcome to Slackware version 10.0 (Linux kernel 2.4.26)!

If you need to pass extra parameters to the kernel, enter them at the prompt
below after the name of the kernel to boot (scsi.s etc). NOTE: In most cases
the kernel will detect your hardware, and parameters are not needed.

Here are some examples (and more can be found in the BOOTING file):
    hdx=cyls,heads,sects,wpcm,irq (needed in rare cases where probing fails)
or hdx=cdrom (force detection of an IDE/ATAPI CD-ROM drive)
where hdx can be any of hda through hdt.

In a pinch, you can boot your system from here with a command like:

For example, if the Linux system were on /dev/hda1.

boot: bare.i root=/dev/hda1 noinitrd ro

This prompt is just for entering extra parameters. If you don't need to enter
any parameters, hit ENTER to boot the default kernel "bare.i" or press [F2]
for a listing of more kernel choices.

boot: _
```

Fig.1 Pantalla de inicio de la instalación

<sup>4</sup>Las imágenes fueron obtenidas al instalar Linux Slackware en una maquina virtual de Windows

Cuando arranca la instalación, Linux detecta la mayor parte del hardware que está instalado en el equipo, lo primero que hay que hacer es seleccionar la configuración del teclado. Aparece un mensaje indicando que si se va a usar un teclado diferente al estadounidense se debe presionar el número 1 y un enter. Si se hace esto aparecerá una lista con las opciones de mapa de teclado.

Se elige la opción `es.map`, es el mapa del teclado para la configuración en español. El término de mapa se refiere a la ubicación de las teclas en un teclado (Fig.2).

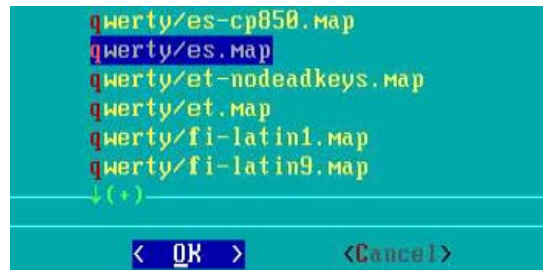


Fig.2 Selección del mapa del teclado

Se comprueba si el teclado funciona correctamente, es decir, se comprueba la localización de los caracteres en el teclado y se acepta la configuración con el número 1 y un enter. Hay que loguearse como root ante el sistema, esto entrega un shell y se puede seguir con la instalación.

Se deben crear las particiones que se necesitan. Como administrador se debe tener un esquema de disco que sea flexible para facilitar las tareas a realizar. Se recomienda hacer particiones pensando en la carga de archivos de sistema, de los usuarios, servicios, etc. Con esto se pueden facilitar las tareas de respaldo, actualización y administración.

Se recomienda que las particiones más grandes sean para los directorios `/home` y `/usr`. Para particionar el disco se usa el comando `fdisk`.

Hay que teclear `fdisk /dev/hda`. La última parte de esta sentencia tiene la siguiente nomenclatura:

`dev-` significa dispositivo

`hd-` Indica un disco duro IDE

`a` – Es la letra asignada al disco, si hay más de uno a los siguientes discos se les asignan las letras `b`, `c`, y así sucesivamente.

Se puede acceder a la ayuda de `fdisk` tecleando la letra `m`, aquí se despliega un menú con las opciones y letras que se deben teclear para crear, borrar y modificar las particiones. Para seguir con la instalación, se debe teclear la letra `n` para crear una nueva partición, se indica que es una partición primaria con la letra `p` y con `1` se indica que número de particiones.

Ahora se define el inicio y el final de la partición, por default ofrece el primer cilindro disponible del disco, para indicarle el tamaño de memoria en disco se usa la siguiente nomenclatura: signo “+” cantidad y unidad. Un ejemplo para indicar 512 megabytes es:

+512 m y enter. Para desplegar la lista de particiones se teclea la letra p y enter. La lista muestra las particiones y algunos datos acerca de estas como su tamaño.

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1		1	510	4096543+	2d	Unknown
/dev/hda2		511	538	224910	83	Linux
/dev/hda3		539	1025	3911827+	83	Linux

Fig.3 Lista de particiones realizadas en la instalación

El sistema necesita tener memoria swap, que utiliza cuando se agota la memoria RAM, esta memoria está en el disco duro. Para cambiar el tipo se teclea la letra t y se indica al sistema que se modificara la partición 1 (o la que se necesite). Pide el código en formato hexadecimal para indicar de que tipo será esta partición.

Con la letra l se listan los códigos que se pueden utilizar.

39	Plan 9	81	Linux / BSD Lin
3c	PartitionMagic	02	Linux swap
40	Unix 80286	83	Linux

Fig.4 Algunos de los códigos para asignar a las particiones

Una vez cambiado el tipo, que es 82 (Fig.4), se crea otra partición para el sistema. El número máximo de particiones primarias son 4. Se pueden crear varias particiones extendidas dependiendo de la capacidad del disco duro. Después de crear las particiones se guardan los cambios tecleando la letra w y se comienza con la instalación del sistema tecleando la palabra **setup** y dando un enter.

Un asistente de instalación en forma gráfica permitirá continuar con la instalación. Es necesario seleccionar el tipo de teclado una vez más, como anteriormente se hizo. Una vez definido el tipo de teclado el sistema detecta la partición swap, aquí indica que se debe darle formato y la habilitará como partición swap del sistema

Una vez que ha preparado el área de swap el proceso de instalación reconoce las particiones de Linux, la partición que se elija en este momento es donde se va a instalar el sistema. La partición seleccionada es para root. Ya que se elige la partición se le da un formato rápido (Fig.5)

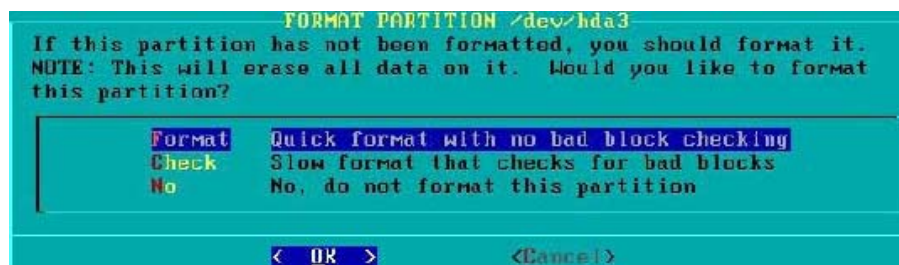


Fig.5 Ventana para formatear la partición seleccionada

Se solicita el tipo de sistema de archivos con el cual se va a formatear la partición. Se debe seleccionar que se formatee con ext3 y se le indica que el tamaño del inodo sea 4096.

Terminado el formato de todas las particiones se entrega un reporte con información de las particiones.

Lo siguiente es elegir el recurso desde el cual se va a realizar la instalación. En este caso se indica que se va a instalar desde los CD, el asistente mostrará un mensaje solicitando auto montaje o montaje del CD en forma manual.

Lo más conveniente es indicar que se aplique un montaje del CD en forma automática. Ya detectado el CD el asistente solicita que se indique la serie de paquetes que se van a instalar. Con la barra espaciadora se marca o desmarca las series de paquetes que se requieren.

Cuando se pregunta el modo de instalación se elige el modo de instalación Full que es automático. Terminada la instalación de paquetes se continúa la instalación del sistema indicándole desde que medio se instalará el kernel para el sistema operativo. Aquí indicamos que es desde el CD y seleccionamos el kernel dependiendo de la arquitectura y de las características de la computadora.

Un ejemplo es: `/cdrom/kernels/bare.i/bzImage`

El asistente da la posibilidad de crear un disco de arranque, este disco puede servir para recuperar el sistema si hay algún problema. Si se cuenta con un MODEM externo o en la ranura de expansión se puede configurar.

Lo siguiente es configurar el gestor de arranque del sistema. “LILO es un paquete cargador de arranque de propósito general que se utiliza en la mayoría de los sistemas Linux modernos. Con él se puede especificar cualquier número de sistemas operativos instalados que se podrá arrancar. El término LILO se deriva de las dos primeras letras de las palabras Linux Loader, cargador de Linux.”<sup>5</sup>

Con LILO se genera un menú de inicio para el sistema, la instalación del siguiente ejemplo está contemplada para una máquina que tiene dos sistemas operativos. Se inicia seleccionando un modo de instalación experto.



Fig.6 Ventana para instalar LILO

<sup>5</sup> Bandel David y Napier Robert, “Edición Especial Linux”, 6° edición., Madrid 2001, Prentice Hall pp. 305

El siguiente menú permite iniciar la configuración de LILO con las propiedades principales, para esto se inicia con la opción begin, el asistente nos permite elegir de un menú el tipo de resolución que se desee para el modo texto o consola, la mayoría de las tarjetas de video soportan una resolución de 1024x768x256.

Se debe indicar que Lilo se instale en el MBR del disco duro, el asistente también pregunta en que disco se instalará si se cuenta con más de uno. Se elije un tiempo de espera antes de iniciar con el primer sistema que se tiene por default.

Se tienen que agregar los sistemas operativos que serán visibles en el menú de booteo. Si se cuenta con un sistema Windows se comienza con éste.

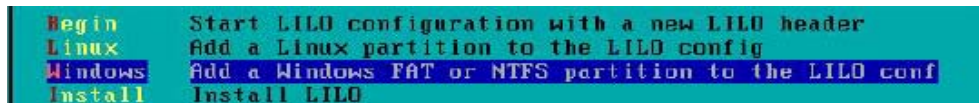


Fig.7 Ventana para agregar un sistema Windows al menú de booteo

El sistema muestra un listado de las particiones que son de tipo Windows solicitando la elección de la partición y un nombre para que aparezca en el menú de booteo. Después se agregan las particiones Linux de la misma manera.

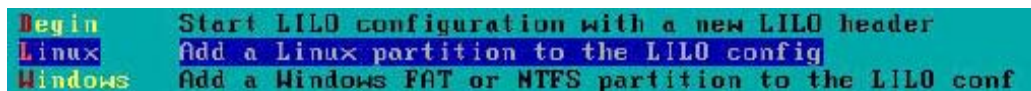


Fig.8 Ventana para agregar un sistema Linux al menú de booteo

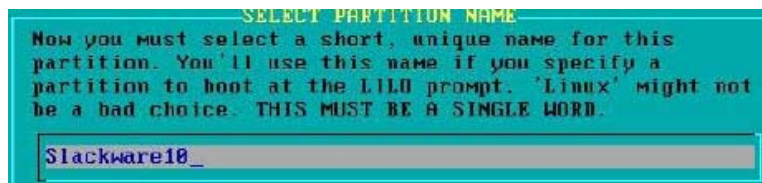


Fig.9 Ventana que permite asignar un alias al sistema operativo

De esta manera se han agregado los sistemas al gestor de arranque. Aparece una vez más el menú inicial y se le indica que instale el LiLo. Al terminar la instalación se indica en el listado que aparece, el tipo de Mouse que esta conectado al equipo, regularmente se selecciona el tipo ps2 y se habilita en modo consola.

El siguiente paso es configurar la red indicándole en primer lugar el nombre de la maquina o HOSTNAME, el dominio, el tipo de IP y su dirección para el servidor, la máscara de red, la puerta de enlace y por ultimo si se configurara un servidor DNS. Al terminar entrega un reporte de los datos que se dieron y pregunta si se guardan los cambios.

El asistente ofrece la posibilidad de indicar los servicios que se requiere estén en funcionamiento. No es recomendable configurar las fuentes, ésta es la siguiente opción que se muestra. Se define la zona horaria, en la lista se selecciona: *America/Mexico\_city*.



Se elige la interfaz gráfica con la que se desea trabajar. En este caso se elige la interfaz KDE. En la siguiente ventana se acepta remplazar el archivo /etc/fstam.

Lo siguiente es teclear una contraseña para el usuario root. Después de esto se ha terminado la instalación y se pide reiniciar el equipo, se retira el cd de la charola. Después de reiniciar el sistema aparece la pantalla del gestor de arranque LiLo (Fig.10), en la cual se puede elegir el sistema operativo con el cual se desea trabajar.



Fig.10 Pantalla de inicio con LILO

## Discos

El sistema dispone de un directorio /mnt el cual forma parte de la estructura estándar de Linux, dentro de este directorio existen algunos otros que se sugieren para montar las unidades de disco.

Cuando se instala Linux éste reconoce todas las unidades de disco y crea un directorio para cada una de ellas en el directorio /mnt que es el “Punto de montaje para aquellos dispositivos del sistema de archivos que sólo monta provisionalmente”<sup>6</sup> si se agregan más unidades de disco al CPU se pueden crear tantos directorios como sea necesario.

Para montar un nuevo disco se cuenta con el comando **mount**:

Sintaxis: mount (dispositivo) [tipo] (punto de montaje) Por ejemplo:

Mount /dev/sda1 /mnt/USB esta instrucción permite montar una memoria USB como un nuevo disco. Para montar un CD la instrucción es mount /dev/cdrom

Para desmontar un dispositivo se deben seguir ciertas reglas, primero el dispositivo no debe estar en uso, esto quiere decir que nadie debe estar realizando tareas como lectura o escritura, otra es que una unidad no se puede desmontar si alguien se encuentra colocado en la estructura de directorios de dicha unidad. El comando es:

Umount /mnt/cdrom o umount /dev/USB

---

<sup>6</sup> Dee-Ann Le Blanc, “La Biblia de administración de sistemas Linux”, México, Anaya, pp. 106

## Administración

Un administrador de sistemas cuida el funcionamiento del software, hardware y periféricos de forma que estén disponibles para ser utilizados por los usuarios. Proporciona un ambiente seguro, eficiente y confiable.

Las tareas administrativas más comunes son las siguientes:

- Administración de usuarios
- Configuración de dispositivos
- Programar respaldos periódicamente
- Capacitar usuarios
- Registrar los cambios del sistema
- Asesorar a los usuarios
- Mantenimiento de claves de usuario
- Instalación y mantenimiento de dispositivos
- Instalación y mantenimiento de software
- Configuración de las interfaces de red
- Administración de los recursos de la máquina (cpu, memoria, etc)
- Monitoreo del sistema
- Detección de fallas
- Implantación de la seguridad del sistema

Un administrador debe tener dominio de un lenguaje de programación, conocimientos del sistema operativo, conocimientos básicos de hardware y mantenimiento de dispositivos, programación shell y conocer las utilerías del sistema.

Como administrador se deben tener copias de seguridad, es decir todo lo que se almacene o se realice en el sistema debe estar respaldado: información de los usuarios, bases de datos, archivos originales del sistema, etc.

También es importante tener y establecer políticas de uso y administración como son: apertura de cuentas, horas de mantenimiento, responsabilidad de los respaldos, borrado de archivos temporales, cuotas de disco seguridad en el sistema mantener canales de comunicación con los usuarios

## Inicio del sistema

El administrador de sistemas debe conocer las diferentes formas para dar de alta o de baja el sistema. El nivel de inicio por default se encuentra definido en el archivo `/etc/inittab`, se necesita conocer los diferentes niveles que están predefinidos en el archivo de inicio:

# 0 = halt	# 3 = multiusuario	# 5 = desuso (levanta conf. 3)
# 1 = monousuario	# 4 = levata la interfaz gráfica	# 6 = reboot
# 2 = desuso		

En la siguiente línea del archivo se muestra la configuración y el nivel de inicio por default en Linux Slackware, que se encuentra definido en el nivel 3:

```
# Default runlevel. (Do not set to 0 or 6)  
id:3:initdefault:
```

Si se deseara que su Linux Slackware inicie por default en algún otro nivel solo hay que cambiar el número 3 por el identificador numérico del modo en el que desea que inicie el sistema, por ejemplo, si se deseara que la computadora inicie en modo gráfico se puede asignar el nivel 4 de inicio.

Desde la consola se puede conocer el nivel de inicio con el comando `/sbin/rulevel` . Se puede cambiar el nivel con el comando `init` y el número de identificador.

Para apagar el sistema se pueden usar varios comandos: `poweroff`, `halt`, `shutdown`, `reboot`, `init` y `telinit`. Cada uno varía en funciones y el que se recomienda más es `shutdown`, ya que da de baja el sistema de una forma segura. Este comando envía una señal de sincronización de discos, manda la señal de término a los programas y los cierra eficientemente, verifica si no hay programas en ejecución y apaga el sistema.

## Usuarios

Una de las principales tareas que efectúa un administrador es la administración de las cuentas de usuario, para esto se debe tener un plan administrativo de claves contemplando altas, bajas y cambios de las claves, también es importante definir los grupos a los que va a pertenecer uno o más usuarios.

También se debe tener un especial cuidado en la capacitación de los usuarios para elegir contraseñas fuertes, asignar permisos adecuados y administrar apropiadamente los recursos para no tener un abuso de ellos. Comandos para administrar los usuarios:

- ✓ `Adduser`.-Agrega usuarios modo comando
- ✓ `Userdel`.-Elimina usuario
- ✓ `Usermod`.-Modifica las propiedades del usuario
- ✓ `Groupadd`.-Agrega grupos
- ✓ `Groupdel`.-Elimina grupos
- ✓ `Groupmod`.-Modifica grupos
- ✓ `Groups`.-Lista los grupos a los que pertenece un usuario
- ✓ `Passwd`.-Modifica propiedades de usuarios/grupos
- ✓ `Kuser`.-Administra usuarios y grupos en modo grafico

## Comandos

La estructura de los comandos es muy rígida y generalmente siguen la misma sintaxis, es decir, siempre se escribe primero el comando, en seguida las opciones si se requieren y al final los argumentos necesarios:

**comando -opciones argumento1 argumento2 ...**

Estos comandos se deben escribir en la consola del sistema. Los comandos más importantes y sus características son los siguientes:

**Hostname:** Muestra el nombre del servidor.

**Uname:** Indica el tipo de LINUX que se esta utilizando. Tiene los siguientes argumentos:

- n Indica el nombre del servidor
- r Muestra la versión del kernel
- m Indica la arquitectura del procesador
- a Muestra todos los datos antes mencionados

**who:** Muestra un listado sencillo de quien se encuentra conectado en el servidor, a que hora se conectó y que esta haciendo.

**whoami:** Indica con que login se esta registrado dentro del servidor, este comando es útil ya que es posible que un usuario se convierta en otro diferente utilizando el comando **su**.

**passwd:** Este comando permite realizar el cambio de contraseña de una cuenta.

**mail:** Tecleando este comando sólo revisa la lista de correos, en caso de que no haya correos mandará un mensaje de que no hay correo en la cuenta.

**pwd:** Este comando imprime en pantalla el directorio actual de trabajo, es decir, donde uno se encuentra posicionado dentro del sistema de directorios indicando la ruta absoluta.

**mkdir (make directory):** Este comando permite crear directorios, es posible crear varios directorios en la misma línea de comando escribiendo uno a continuación de otro separados por un espacio. Ejemplo:

mkdir tareas o bien mkdir examen ejercicios música imágenes

- p Busca la rama de directorios y si no la encuentra la crea.

**rmdir (remove directory)** Este comando borra directorios, la única condición que debe cumplir es que se encuentre vacío el directorio a ser borrado. Ejemplo: rmdir examen

**rm (remove)** El comando rm borra archivos, puede ser borrado un solo archivo o varios haciendo uso de los metacaracteres \* ?. Ejemplo:

rm saludo                      rm lo\*                      rm -r directorio

La opción **-r** hace un borrado en forma recursiva, esta opción es de mucho cuidado ya que se borrará el directorio especificado y todo su contenido.

La opción **-f** fuerza a que los elementos sean borrados aunque uno no tenga permisos sobre ellos.

**ls (listado)** Muestra un listado del contenido del directorio actual de trabajo, este comando tiene varias opciones, entre las más importantes se encuentran las siguientes:

- l Muestra un listado con formato largo indicando todas las propiedades de los archivos y directorios del directorio actual de trabajo.
- a Muestra todo el contenido del directorio, incluyendo los archivos y directorios ocultos.
- r Muestra el listado en orden inverso alfabético.
- R Muestra un listado en forma recursiva, es decir lista el contenido de directorios y subdirectorios a partir de la ruta donde le indiquemos.
- t Muestra un listado tomando como opción la fecha de modificación del archivo o directorio.

**Comodines** Existen dos caracteres llamados comodines los cuales pueden sustituir un carácter o conjunto de caracteres durante la ejecución de comandos. Ejemplo:

`ls ??a` Muestra aquellos archivos y directorios los cuales tengan el nombre de tres letras y la tercera sea la letra a

`ls *u*t*` Muestra aquellos archivos y directorios los cuales tengan en el nombre una “u” y una “t” en cualquier posición pero en el orden que se le indica.

**Cd (Cambio de directorio)** Este comando permite moverse entre directorios, es necesario que se indique la ruta del directorio al que se va cambiar. Ejemplo:

```
cd dir1/dir2/dir3
```

Es posible utilizar rutas absolutas o rutas relativas según sea el caso. Las rutas relativas nunca inician con el símbolo de raíz / en lugar de ello inician la ruta a partir del directorio actual de trabajo.

**cp (Copiar)** El comando cp permite hacer copias de archivos uno por uno, múltiples archivos o incluso hacer copias de estructuras de directorios.

```
cp ruta_origen ruta_destino
```

Ejemplo:

Del directorio /etc copiar los archivos que inicien con las letras mo dentro del directorio configuracion que ya deberá estar creado. Comando: `cp /etc/mo* /configuración`

**mv (move)** El comando mv tiene dos funciones básicas, la primera es mover archivos o estructuras de directorios de un lugar a otro, la segunda es renombrar archivos o directorios.

**cat** permite ver el contenido de uno o mas archivos, su función original es concatenar archivos. Ejemplo:

```
cat archivo1 archivo2 ...
```

- n Utilizando esta opción el comando imprime el contenido del o los archivos numerando las líneas en forma ascendente.

**tar:** “La herramienta de empaquetado tar se usa en todas las plataformas UNIX. Su función principal es almacenar archivos. Se puede utilizar para almacenar un grupo de archivos o incluso varias jerarquías de directorios sin perder ni sus permisos ni la estructura de archivos.”<sup>7</sup>

Sintaxis:

tar [opciones] archivo.tar archivo

- c Crea un nuevo archivo tarreado.
- v Imprime en pantalla todo aquello que esta afectando.
- f Indica al sistema que va a afectar a un archivo
- x Extrae el contenido de un archivo .tar
- t Lista todos los archivos contenidos en un archivo .tar en formato largo mostrando todos sus detalles.

Ejemplo:

tar -xvf archivo.tar Aquí se esta recuperando la información almacenada dentro de un archivo empaquetado

## Redireccionamiento

En LINUX existe una salida estándar que es el monitor, de igual manera hay una entrada estándar que es el teclado, de tal modo que todo lo que se requiera hacer en forma convencional se le debe indicar al sistema operativo insertando las instrucciones desde el teclado, de la misma forma todo resultado de los comandos será impreso por default en la pantalla a no ser que se le indique otra cosa al sistema.

El carácter mayor que (>) permite redireccionar la salida de un programa hacia un archivo. Ejemplo:

```
ls > milista
```

Manda la salida del comando ls a un archivo llamado milista, el resultado del comando ls no se imprime en pantalla, sino en el archivo indicado. Si el archivo milista no existe será creado automáticamente, si el archivo existe será sobrescrito y su contenido se perderá.

El carácter menor que (<) permite redireccionar un archivo hacia un programa. Ejemplo:  
cat < milista

Este comando manda el archivo milista hacia el programa cat mostrando en pantalla el contenido del archivo milista.

```
mail usuario < milista
```

Manda el contenido del archivo milista por correo al usuario con login usuario.

---

<sup>7</sup> Dee-Ann Le Blanc, op. Cit, nota 6 pp. 152

Doble mayor que (>>) permite redireccionar la salida de un programa hacia un archivo, si el archivo no existe lo crea, pero si el archivo existe el resultado del comando lo añade al final del archivo existente sin eliminar el contenido del archivo. Ejemplo:

```
ls >> milista
```

Existe una salida estándar de verdadero y una de falso, a la salida de verdadero le corresponde el número 1 y a la salida de falso le corresponde el número 2, de tal forma que si en algún comando no se cumple una condición en lugar de que el error se imprima en pantalla puede ser redireccionado hacia un archivo, Ejemplo:

```
ls -R /etc 1>milistado 2>error
```

Aquí, la salida de verdad esta direccionada al archivo milistado mientras que la salida de error esta direccionada al archivo error.

**Grep** Este comando busca patrones en archivos. Por defecto devuelve todas las líneas que contienen un patrón determinado en uno o varios archivos. Utilizando las opciones se puede variar mucho este comportamiento.

Sintaxis: `grep [opciones] <patrón> [ficheros]`

Ejemplo:

Se necesita un filtrado del archivo `/etc/services`, del cual se requieren las líneas que contengan la palabra `ftp` pero sólo aquellas que hagan referencia al protocolo `tcp`, el resultado deberá guardarse en un archivo llamado `ftp` dentro del directorio `practical`.  
Comando:

```
grep ftp /etc/services | grep tcp > practical/ftp
```

## Permisos

“Los permisos controlan quien tiene acceso de lectura escritura y ejecución sobre los elementos de un sistema de archivos (archivos, directorios y mucho más). Este control se divide en diferentes segmentos de la población de usuarios para ofrecer un acceso más refinado a lo que puede conseguir cada uno.”<sup>8</sup>

Existen tres tipos de permisos que se aplican tanto a los archivos como a los directorios y son:

- **r** El permiso de lectura permite que el archivo pueda ser leído o copiado.
- **w** Permite escribir en el archivo, hacer modificaciones o borrarlo si así lo desea.
- **x** Los archivos ejecutables son aquellos que pueden realizar un proceso.

---

<sup>8</sup> Dee-Ann Le Blanc, op. Cit, nota 6 pp. 107

Estos permisos se encuentran asignados en tres bloques, los cuales representan al usuario o dueño del elemento listado, al grupo al que pertenece el usuario y a los otros usuarios del sistema. Ejemplo:

```

-rw-r--r-- 1 carlos becas 30208 Sep 29 23:54 apa1.jpg
-rw----- 1 carlos becas 1170 Sep 29 20:14 dead.letter
-rw-r--r-- 1 carlos becas 86796 Sep 29 23:58 Guerra1.jpg
drwxr-xr-x 2 carlos becas 512 Sep 30 01:16 hola
lrwxrwxrwx 1 root other 5 Sep 28 15:11 hola1 -> hola1
-rw-r--r-- 2 carlos becas 0 Sep 28 15:11 liga

```

**chmod** Permite cambiar los permisos a los archivos y directorios, la asignación de permisos puede hacerse de dos formas mediante el método simbólico o mediante el método octal.

Chmod utiliza las siguientes opciones:

- u Indica que se modificarán los permisos del usuario dueño del elemento.
- g Indica que se modificarán los permisos asignados al grupo.
- o Indica que se modificarán los permisos de los otros usuarios que no pertenecen al grupo.
- a Indica que afectará los permisos para todos los usuarios.

Para modificar los permisos se pone el indicador de los permisos a afectar ya sea para el usuario, el grupo o los otros, seguido del signo + si se quiere agregar el permiso o de - si se quiere quitar el permiso separando con una coma los bloques de permisos, al final se indica el nombre del elemento a afectar. Ejemplo:

```
chmod u+r,g-rx,o+r hola
```

Con el método octal se asignan directamente los permisos en base a una tabla de validación que va numerada de 0 a 7 dependiendo de los permisos que se deseen asignar, cada permiso adquiere un valor numérico:

X vale 1, W vale 2 y R vale 4

Para asignar los permisos se hace la suma de los valores, por ejemplo para asignar un permiso de lectura y ejecución se usa el número 5, sustituyendo el número correspondiente para cada bloque de permisos al usuario, al grupo y a los otros. Ejemplo: Si se requiere poner todos los permisos para el usuario, permisos de lectura y ejecución para el grupo y ninguno para los otros el comando quedaría de la siguiente forma: `chmod 750 hola`

## Procesos

Un proceso es un programa que se encuentra corriendo dentro del servidor, cada proceso tiene un tiempo de vida que va desde el momento en que se presiona la tecla de enter hasta que termina la ejecución del proceso.



Todos los procesos tienen un dueño, este dueño es el usuario quien invocó el proceso desde el intérprete de comandos, de tal forma que ningún otro usuario puede afectar las tareas que ese usuario está realizando, debido a que no son dueños del proceso que se está realizando.

**ps** Permite ver los procesos que se están ejecutando desde la sesión abierta, y que se encuentra en uso en este momento, muestra un listado de los procesos en formato corto.

**ps -u login** muestra un listado de todos los procesos que está ejecutando un usuario, aunque se encuentre trabajando en varias sesiones el sistema listará todos los procesos que le pertenecen a dicho usuario.

**ps -f** Entrega un listado de los procesos que se están ejecutando en la sesión, muestra los procesos en formato largo, donde se puede ver la dependencia de los procesos por el UID (identificador del Usuario dueño del proceso), PID (el número del identificador del Proceso en ejecución.) y el PPID (el identificador del Proceso Padre del Proceso en ejecución)

**ps -fea** Muestra todos los procesos ejecutándose en el servidor en formato largo.

**kill** Este comando permite entre otras opciones matar procesos, uno como usuario únicamente podrá matar sus propios procesos.

**Conclusión:** Si se conocen bien los comandos y utilidades del sistema se puede tener una buena administración y aprovechamiento del mismo. Slackware está enfocado a servidores por lo tanto se pueden hacer publicaciones en red, por esto, el encargado de éste debe tener una buena organización de los archivos y debe conocer bien el sistema. Es una ventaja que pueda ser distribuido de manera gratuita ya que esto implica que pueden bajar los costos en una empresa.

## Capítulo II

### HTML

**Objetivo:** Aprender la sintaxis básica del lenguaje HTML para la creación de páginas Web, así como el saber utilizar los editores que nos auxilian por medio de interfaces gráficas para que la programación sea más rápida y amigable.

**Justificación:** Para la creación de páginas Web es necesario conocer el lenguaje básico HTML. La creación de páginas Web es una necesidad de las empresas para poder brindar mejores servicios a través de la red para los usuarios.

**Introducción:** Las publicaciones en Internet o en una red interna de una empresa, se han incrementado debido a que los usuarios siempre están en busca de información. El diseño de un sitio y sus páginas es un tema muy importante ya que a la mayoría de los usuarios les atraen los sitios vistosos y que faciliten el acceso a la información.

### Contenido

Para el diseño de páginas es necesario establecer algunos de los parámetros más importantes a considerar como son los siguientes:

- a) Objetivo principal de la página
- b) Información necesaria para el desarrollo de la página
- c) Software empleado para el desarrollo de la página
- d) Manipulación y tratamiento de imágenes
- e) Equipo necesario para el montaje y/o desarrollo de la página
- f) Publicación y difusión de la página
- g) Actualización y mantenimiento de la página

El diseño de una página Web se ve apoyado en gran medida al emplear un editor de HTML, ya que construye por sí solo, las etiquetas de HTML, además de brindar una referencia rápida de éstas.

Algunos ejemplos de editores de HTML son:

- Front-page (Microsoft)
- Hot Dog
- Hot Metal
- Web Editor Pro
- Dreamweaver

La distribución Slackware ofrece el editor Quanta que ya viene instalado.

El navegador o browser: es el que permite desplegar las páginas en el monitor de la computadora interpretando el código HTML que compone la página; además de desplegar texto e imágenes y otras aplicaciones que se incorporan a la página.

También es necesario un Servidor de HTML o Web que permita aceptar y atender las peticiones que llegan a través de Internet para consultar la página. Una vez que la página (o páginas) están terminadas, el siguiente paso es montar o publicar la página en el Servidor de Web.

Para la difusión de la página, ésta se debe de registrar en los sistemas de búsqueda en Internet más importantes. Una vez terminada la página, es posible que se tengan que programar una serie de actividades encaminadas a estar actualizando y agregando información a la página según se necesite.

## Lenguaje HTML

“HTML es un lenguaje que se emplea para crear páginas Web. Un código escrito en este lenguaje es, básicamente, un texto que el navegador (Internet Explore, Netscape, Navigator, Opera o cualquier otro) mostrará en formato de página Web. Este texto puede generar color, tamaño y fuente de la letra, fondos, imágenes, hiperenlaces y entrada de datos, así como listas de selección, botones, etc., determinados y pueden ser configurados mediante los identificadores, también llamados tags.”<sup>9</sup>

Los identificadores o etiquetas de HTML son un conjunto de instrucciones o comandos que después son interpretados por los visualizadores de WEB, desplegando como resultado de esta interpretación las páginas WEB. Las Etiquetas de HTML se interpretan en el orden en que se ponen y van encerradas entre pico paréntesis (<>).

La estructura básica de una página debe llevar el siguiente esqueleto o estructura base:

```
<html>
  <head>
    <title>Espacio para título de la ventana</title>
  </head>
  <body>
    Espacio para escribir todo el contenido de la página.....
  </body>
</html>
```

Las páginas se guardan con el nombre que se elija seguido de **.html**, ya que sólo por la extensión el sistema operativo reconoce este archivo como un documento Web y no solo como un archivo de texto. Para ejecutar la página solo se debe hacer doble clic sobre el icono de ésta y automáticamente se abrirá el navegador y se cargará la página.

Si se accede a un servidor de páginas en la red la página se cargará automáticamente, generalmente la página principal tiene el nombre de Index.html.

---

<sup>9</sup> López Quijado José, “Domine HTML y DHTML” México, Alfaomega Ram-Ma, p.p. XI

## Etiquetas más importantes

**<html>.....</html>** Es la primera y la última etiqueta en un documento html y le indica al navegador que despliegue el documento como Hipertexto y no como texto sencillo.

**<head>.. </head>** La etiqueta encabezado identifica la sección inicial del documento html. Entre otras cosas escribe el título del documento de esa sección

**<title>...</title>** La etiqueta title contiene el título que aparece en la barra de título del navegador Web

**<body>...</body>** La etiqueta de cuerpo abarca la parte más grande de una página Web. Contiene todo lo que el usuario ve cuando entra a esa página. La mayoría de las etiquetas se colocan dentro de body. Esta etiqueta tiene los siguientes atributos:

- **text:** Define el color del texto de toda la página
- **link:** Color del texto y los marcos de los enlaces
- **alink:** Define el color de los enlaces que están siendo activados
- **vlink:** El color de los enlaces que han sido usados
- **bgcolor:** Contiene el color de fondo
- **background:** Se utiliza para poner el fondo a una página, puede ser una imagen

Los colores utilizan el siguiente orden en hexadecimal **RGB "RED GREEN BLUE"**  
Ejemplo:

```
<body " bgcolor="#00FF00" text="white" link="yellow" vlink="#0000ff">
```

**<h1>...texto...</h1>** (para  $n=1..6$ ) La etiqueta encabezado, crea varios tamaños de encabezados. La n se reemplaza por un número entre 1 y 6 cuanto menor sea el número mayor será el encabezado.

**<font>...</font>** Permite modificar los parámetros de la fuente de escritura que se utilice, cuenta con atributos como son:

**size="n"** Define un tamaño de fuente específico, **n** representa un número del 0 al 7

**face="nombre de fuente"** Permite definir un nombre de Fuente específico

**color="#00FF00"** Permite colocar un color a la fuente.

**<p>...</p>** La etiqueta de párrafo permite separar bloques de texto y hacerlos más legibles. Cada una de estas etiquetas se coloca al principio o al final de cada párrafo.

**<center> ..... </center>** Coloca alineado al centro de la página todos los elementos que se coloquen entre las etiquetas, incluido el texto. **<left>...</left>** y **<right>...</right>** Colocan los elementos alineados a la izquierda y a la derecha respectivamente.

**<br>** La etiqueta de interrupción origina un salto de línea, con excepción de que coloca una línea en blanco como separador obligadamente.

`<i>...</i>` Etiqueta de texto itálico o cursivas, `<b>...</b>` Etiqueta de texto bold o negritas, `<u>...</u>` Etiqueta de texto subrayado.

`<big>...</big>` y `<small>...</small>` agrandan o disminuyen el tamaño del texto. `<sub>...</sub>` y `<sup>...</sup>` Representa el texto sobre el renglón o bajo el renglón, como un superíndice o un subíndice.

`<hr=n>` La etiqueta de regla horizontal, dibuja una línea horizontal en medio de la página Web, cuyo grosor está representado por “n”.

## Caracteres especiales

Existen caracteres que no se pueden mostrar directamente en el navegador, como caracteres acentuados, existen una serie de secuencias de escape que “permiten que ciertos caracteres especiales sean leídos por todos los navegadores en sus distintas versiones. La escritura de cada caracter comienza con el signo ampersand (&) y debe terminar con punto y coma (;)”<sup>10</sup>. Algunos caracteres son:

Escribir	Para ver	Escribir	Para ver	Escribir	Para ver	Escribir	Para ver
<code>&amp;lt;</code>	<	<code>&amp;eacute;</code>	é	<code>&amp;Eacute;</code>	É	<code>&amp;Ntilde;</code>	Ñ
<code>&amp;gt;</code>	>	<code>&amp;iacute;</code>	í	<code>&amp;Iacute;</code>	Í	<code>&amp;uuml;</code>	ü
<code>&amp;amp;</code>	&	<code>&amp;oacute;</code>	ó	<code>&amp;Oacute;</code>	Ó	<code>&amp;Uuml;</code>	Ü
<code>&amp;quot;</code>	"	<code>&amp;uacute;</code>	ú	<code>&amp;Uacute;</code>	Ú	<code>&amp;#191;</code>	¿
<code>&amp;aacute;</code>	á	<code>&amp;Aacute;</code>	Á	<code>&amp;ntilde;</code>	ñ	<code>&amp;#161;</code>	¡

Ejemplo de código HTML:

```

<html>
  <head>
  <title>Ejemplo</title>
  </head>
  <body background="/root/Rosario/Mis imágenes/foca.jpg">
  <h1>Titulo</h1><br>
  <hr align="center" size="2">
  <font face="Arial" color="Blue" size="+2">Texto de color</font>
  <br><br>
  <p align="left">Texo a la izquierda</p><br>
  <p align="center">Texto al centro</p><br>
  <p align="right">Texto a la derecha</p><br>
  <B>Texto en negritas</B><br>
  <I>Texto en Italica</I><br>
  <U>Texto subrayado</U><br>
  &quot;Texto con caracteres especiales, &lt;como funci&oacute;n&gt;&quot;
  </body>
</html>

```

Fig.1 Ejemplo de código HTML en el editor Quanta

<sup>10</sup> <http://www.aprender.org.ar/aulas/avafp/recursos/html.htm>

Con el código de la Fig.1 se muestra la siguiente página:



Fig.2 Ejemplo de una pagina con etiquetas como: h1, font, center, etc.

## Otros elementos

Las tablas siempre han presentado una manera muy cómoda y visual de presentar la información. Las Tablas permiten incorporar a las páginas de WEB las estructuras compuestas por renglones y columnas que tradicionalmente se manejan en los procesadores de texto. Para formar una tabla se escribe una estructura base como la siguiente:

```
<table>
  <tr>
    <td> Contenido </td> primer renglón, primer columna
    <td> Contenido </td> primer renglón, segunda columna
  </tr>
  <tr>
    <td> Contenido </td> segundo renglón, primer columna
    <td> Contenido </td> segundo renglón, segunda columna
  </tr>
</table>
```

Las tablas presentan algunos atributos como:

- **border=n** y **bordercolor="#00FF00"** **n** = representa al grosor del borde de la tabla. También se podrá especificar el color del borde
- **cellspacing=n** Las celdas podrán separarse por el **n** valor
- **cellpadding=n** El contenido podrá estar separado del borde por el valor indicado en **n**
- **height=n** o porcentaje y **width=n** o porcentaje, se puede establecer un valor o un porcentaje para definir el tamaño de la tabla

- **bgcolor="#00FF00" background="imagen.gif"** Se puede establecer un color de fondo de toda la tabla o colocar una imagen de fondo

“Se pueden colocar imágenes como fondo de la página pero también como contenido de la misma, es decir una imagen es un objeto más de los que componen la página, como un párrafo de texto. Para esto se emplea el tag **<img>**.”<sup>11</sup>

**** Esta etiqueta permite incorporar imágenes. Las imágenes deben estar en formato GIF o JPG ya que estos son los únicos formatos que reconoce el visualizador. Es posible cambiar el tamaño de la imagen empleando los modificadores de esta etiqueta WIDTH y HEIGHT que se definen en píxeles o por porcentaje como se ve en el siguiente ejemplo:

```
 ó  

```

Las hiperligas permiten definir vínculos para hacer referencias o “brincar” hacia otras páginas y servicios de Internet. La estructura de una liga de hipertexto es como sigue:

```
<a href="dirección del recurso"> Texto o imagen del vínculo</a>
```

Este tipo de ligas puede acceder a recursos como otra página WEB:

```
<a href="http://www.google.com"> Ir a Google </A>
```

Los vínculos no solo llevan a otros documentos de la Web, si no que también pueden utilizarse para moverse dentro un mismo documento o hacer un enlace a un correo electrónico:

```
<a href = "mailto:dirección e-mail"> texto indicativo del enlace </a>
```

Otro elemento son los listados. “Las listas en HTML permiten elaborar rápidamente relaciones de elementos que son mostrados en diferentes formatos en las páginas de WEB dependiendo del tipo de lista que se emplea. HTML soporta 3 tipos de listas: ordenadas, no ordenadas y listas de definiciones, cerradas con los pares **<OL> </OL>**, **<UL> </UL>**, y **<DL> </DL>** respectivamente. Las listas ordenadas se usan para algoritmos mientras que las listas no ordenadas se usan para listar artículos en los que no importa el orden. Las listas por definición en cambio proveen un formato parecido al de un diccionario o glosario.”<sup>12</sup>

Con la etiqueta **<LH>** se define el título de la lista, este es opcional y aparecerá en la parte superior de la lista. Los elementos individuales dentro de una lista ordenada son identificados con una etiqueta **<LI>** (El elemento **</LI>** no existe).

Para indicar desde donde hay que empezar a numerar una lista con números o letras se usa **<OL START=#** donde # es el número o posición de alfabeto desde donde debe arrancar la numeración. Letras capitales y números romanos con **<OL type= A, a, I, i, 1>**.



---

<sup>11</sup> López Quijado José, op. Cit, nota 1 pp. 79

<sup>12</sup> <http://www.ing.ula.ve/~herman/comht.html>

En una lista no ordenada se puede usar <UL type= disc, circle, ó square>

## Tablas

TITULO1	TITULO2	TITULO3
elemento1	elemento2	elemento3
		

[Ir a Google](#)

### LISTA

- ◊ elemento1
- ◊ elemento2
- ◊ elemento3

Fig.3 Ejemplo de tabla, vínculo y lista:

## Formularios

“Los documentos HTML normales sirven solamente para el transporte de información en una dirección, del servidor al browser. Hay una necesidad muy frecuente por parte de los ofertantes de información de poder recibir respuestas por parte de los usuarios, por esto fue necesario crear la posibilidad de transportar información en sentido inverso, es decir, del browser al servidor. La solución consiste en los llamados ‘forms’ o formularios.”<sup>13</sup>

La mayor parte de las aplicaciones necesitan en algún momento información introducida por el usuario, y en el entorno Web, ésta consiste en entradas que proceden de formularios. HTML proporciona un conjunto básico de elementos o controles de entrada adaptados a una amplia gama de requisitos de entrada de datos. Este conjunto incluye:

**Elementos de entrada de texto:** Casillas rectangulares para entradas de una sola línea o varias líneas.

**Menús de selección:** Listas de opciones mostradas en un cuadro de lista desplegable. Tienen una forma externa mostrada en pantalla y un valor de código interno asociado al elemento o elementos seleccionados.

**Botones:** Controles rectangulares que simulan un botón en un panel de control. Se utilizan con frecuencia principalmente para iniciar un comando, enviar un formulario o restablecer los campos de entrada a su estado inicial.

<sup>13</sup> Dr Shwarte Jachim, “El gran libro de HTML como publicar en Internet”, México, Marcombo, p.p. 92



**Casillas de verificación:** Pequeños rectángulos que se pueden activar o desactivar. Las casillas de verificación se pueden utilizar para especificar opciones con valores de "sí" o "no".

**Botones de opción:** Similares a las casillas de verificación, los botones de opción indican valores de "si" o "no". Sin embargo, ocurren en grupos mutuamente excluyentes, de forma que al seleccionar uno se deseleccionan los demás.

**Elementos ocultos:** Elementos no visuales que se utilizan para crear parámetros con valores constantes.

Los formularios emplean una serie de instrucciones que definen acciones o elementos dentro del formulario tales como la siguiente etiqueta de HTML:

```
<form action="compra.html" method="get">
<select name="productos">
<option value="pan">Pantalones</option>
<option value="cam">Camisas</option>
</select>
Cantidad de unidades<input type="text" name="cantidad" size="5" maxlength="2">
<input type="submit" value="Realizar compra">
```

La etiqueta <form> define el inicio de la forma y lleva los modificadores method, que puede ser POST o GET según el método de envío al servidor. En el método GET toda la información se convierten en elementos del URL, que se emplea para solicitar la respuesta esperada. En el caso del método POST se emplea cuando hay más datos a transmitir de los que pueden transmitirse dentro de la URL.

**action** indica el nombre del programa a ejecutar para procesar la información enviada por el formulario. Envía la respuesta inmediatamente a la dirección especificada en formato de texto.

<input type="..." name="..."> la etiqueta <input> indica un elemento del formulario, donde type define la clase de elementos tales como:

- ✓ **text:** para campos de captura de texto junto con size=# donde # define el número de caracteres
- ✓ **radio:** para casillas de opciones únicas
- ✓ **check:** para casillas de verdadero o falso
- ✓ **submit:** para botones, con value="texto" se define el texto que aparece en el botón
- ✓ **reset:** que genera un botón que borra los valores introducidos y regresa a los valores por default y que junto con value="texto" indica el nombre del botón.
- ✓ **name:** indica el nombre del elemento

Es posible crear listas de opciones que se muestran como menús desplegables, para esto se utiliza la etiqueta <select>, se define el nombre de la variable con name y las opciones se escriben con la etiqueta <option>. Ejemplo de Formulario (Fig.4):



Fig.4 Formulario

## Frames

Los Frames son divisiones en forma de ventanas que se pueden definir desde HTML con la etiqueta <Frameset>. Para incluir frames en una página se sustituye el <Body> y se trabaja con <Frameset>, de tal forma que se pueden crear frames Horizontales o Verticales dependiendo si se usa <frameset cols=""> o <frameset rows="">.

En cada Frame se puede incorporar una página de WWW de tal forma que se emplea la etiqueta <frame src="" Name=""> para incorporar el contenido de cada FRAME.

## Comandos META

META permite definir propiedades internas del documento. “En el registro automático de páginas, muchos motores de búsqueda sólo leen los 200 primeros caracteres de un archivo y transmiten esos caracteres a su base de datos. Por esto es conveniente incluir en esos 200 caracteres del documento HTML un número suficiente de palabras claves importantes. Esto es casi imposible en las páginas en que se utilizan frames, por esto los servidores de búsqueda tienen problemas para incorporar páginas diseñadas con frames.”<sup>14</sup>

Con ayuda de las informaciones meta se pueden formular algunos criterios propios para la inclusión de páginas en los motores de búsqueda. El comando Meta se utiliza para transmitir al motor de búsqueda indicaciones sobre cómo incorporar la página a su índice. Por ejemplo:

```
<meta name="description" content="Descripción"> ó
<meta name="keywords" content="keyword1, keyword2">
```

En la palabra clave description se ofrece una breve descripción del contenido de la página. En keywords las palabras claves que deben utilizarse para encontrar la página.

**Conclusión:** Para la creación de páginas Web se debe tener en cuenta que existen muchas opciones del lenguaje HTML, esto es muy útil si se requiere que las páginas sean vistosas y atractivas para el usuario. Se debe planear como será la estructura de un sitio y el contenido de las páginas para saber que elementos del lenguaje hay que utilizar.

<sup>14</sup> Pott Oliver, “Gran libro de HTML y XML Referencia y Práctica” México, Marcombo- Boixareu Editores, p.p577

## Capítulo III

### APACHE

**Objetivo:** Conocer cuales son las funciones más importantes del servidor Apache y aprender a instalarlo así como configurarlo. También conocer de que forma es posible administrar las páginas Web en un servidor.

**Justificación:** Después de aprender como se elaboran las páginas de Internet es necesario conocer como son colocadas en un servidor para que sean visitadas en la red por los usuarios, además se debe conocer como configurar los mensajes que se presentaran al usuario si hay un error o alguna situación inesperada.

**Introducción:** “El servidor Apache es un servidor HTTP<sup>15</sup> de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows y otras, que implementa el protocolo HTTP y la noción de sitio virtual. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características: mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.”<sup>16</sup>

### Contenido

Cuando una persona quiere acceder a los recursos de un servidor por medio del protocolo http ocurren los siguientes pasos:

1. El cliente HTTP abre una conexión.
2. El Servidor manda un “acknowledge” notificando que se ha abierto una sesión.
3. El cliente envía su “request message” solicitando un recurso.
4. El servidor responde con “response message” que contiene el recurso solicitado y cierra la conexión.

La figura 1 muestra la arquitectura cliente servidor que se maneja:

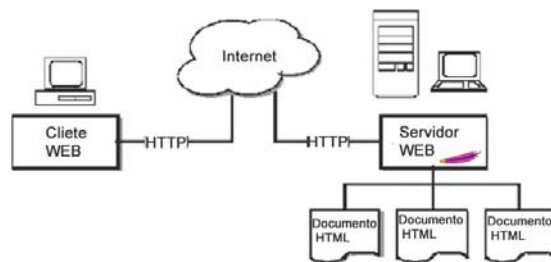


Fig1. Arquitectura cliente-servidor

<sup>15</sup> HTTP protocolo de transferencia de hipertexto es el lenguaje que utilizan el cliente y el servidor para comunicarse entre si.

<sup>16</sup> [http://es.wikipedia.org/wiki/Servidor\\_web\\_Apache](http://es.wikipedia.org/wiki/Servidor_web_Apache)

“El servidor Web Apache se originó a partir del servidor Web del Nacional Center for Supercomputing Applications (NCSA), **httpd**. Tradicionalmente a la hora establecer la evolución de los programas fuente abierta, alguna gente decidió que quería más funcionalidades añadidas al httpd original. Estas funcionalidades empezaron con un patch aquí y otro allí, cada uno aplicado al daemon original para un empleo personal del individuo. Con el tiempo, estos patches se reunieron, al final se trataba de un servidor Web ‘patchy’, de ahí su nombre ‘Apache’ que nació a finales de 1995.”<sup>17</sup>

## Características de Apache

Apache es el un servidor que se utiliza mucho por las siguientes características:

- Es robusto, y soporta un gran número de transacciones
- Configurable para diferentes entornos de trabajo
- Con un alto nivel de seguridad
- Disponible para una gran variedad de plataformas
- Soporte para servicio de proxy
- Soporte para granjas de servidores
- Soporte para Scripting languages integrados como modulos (por ejemplo PHP, mod\_perl)
- Incluye el código fuente del servidor
- Soporte para accesos restringidos
- Soporte para SSL
- Es gratuito

## Descarga e instalación

Apache se puede obtener del sitio Web: <http://www.apache.org/download.cgi>. Se descarga la versión que se va utilizar, en este caso httpd-2.0.54.tar. Para instalarlo primero es necesario desinstalar el Apache que ya viene en Slackware por default, se accede a la consola de trabajo y se teclean las siguientes instrucciones:

- ❖ Pkgtool y enter
- ❖ Se selecciona la opción remove
- ❖ Se busca Apache y lo se selecciona, para finalizar se da un enter

Para instalar la versión que se descargó primero hay que descomprimir el archivo, acceder a la carpeta descomprimida e indicar en que dirección se debe instalar, esto debe de ser desde la consola, las instrucciones son las siguientes:

- ❖ tar zxvf httpd-2.0.54.tar
- ❖ cd httpd-2.0.54

---

<sup>17</sup> Dee-Ann Le Blanc, Op. Cit, nota 6 Capitulo 1 pp. 503

- ❖ ./configure --prefix=/usr/local/apache
- ❖ make
- ❖ make install

Después de instalarlo se debe levantar el servidor desde la consola de trabajo con las siguientes instrucciones:

- ❖ cd/usr/local/apache/bin
- ❖ ./apachectl start y enter
- ❖ para dar de baja el servidor: ./apachectl stop y enter
- ❖ para resetear el servidor: ./apachectl restart y enter

Todas estas instrucciones deben ser escritas dentro del directorio **bin** de apache. Cuando se levanta apache es necesario saber si esta bien instalado, para esto se abre un navegador y se escribe la dirección: <http://localhost> debe aparecer una página que indica que Apache fue instalado correctamente.

Para saber si el servidor esta funcionando se utiliza la siguiente instrucción en consola:

```
ps -fea | grep httpd
```

El archivo PidFile contiene un valor numérico que identifica al proceso httpd “padre”. Si este archivo se pierde es común tener problemas cuando arranca o se detiene apache

Para “matar” a un proceso de apache se ejecuta la siguiente instrucción:

Kill -9 proceso, ejemplo: Kill -9 12986.

Para matar todos los procesos: pkill httpd

## Directivas

Si se necesita configurar Apache sólo se debe de modificar el fichero httpd.conf y después reiniciar o bien apagar y arrancar el proceso del comando httpd. Antes de modificar el fichero httpd.conf se debe de copiar el fichero original con otro nombre para tener una copia de seguridad.

Apache es administrado por más de 200 directivas las cuales permiten que determinada funcionalidad pueda ser incluida. Una directiva es un parámetro de configuración que determina el comportamiento del servidor web, estas directivas se encuentran incluidas en el archivo httpd.conf, que se encuentra en el directorio /usr/local/apache/conf

En apache la configuración se realiza mediante directivas organizadas en tres grupos:

- ☞ **Global Environment:** administra las directivas generales de operación para apache.
- ☞ **Main Server:** administra las directivas del servidor principal o estandar de apache.
- ☞ **Virtual Servers:** administra las directivas donde los mismos procesos de apache soportan diversas Ip’s o nombres de dominio.

Usando un editor como vi o pico se pueden modificar los parámetros del archivo `httpd.conf`.

**ServerName:** “Esta directiva especifica el nombre del servidor, que se usa cuando el servidor está construyendo direcciones de redireccionamiento. Si no se especifica esto, el servidor tratará de saberlo por sí mismo mediante búsquedas DNS. Sin embargo, es posible que así no se consiga el nombre deseado.”<sup>18</sup>

**User:** Esta directiva define el user ID mediante el cual apache operará. Valor por Default: `User #-1`

**Group:** Esta directiva define el group ID mediante el cual apache operará. Valor por Default: `User #-1`

**Listen:** Esta directiva define cual es el puerto en que operará el servidor de Web.

**ServerAdmin:** Esta directiva define el correo electrónico del administrador del servidor Web e indica la dirección a incluirse en los mensajes de error que serán enviados al cliente. Es necesario tener habilitada la directiva **ServerSignature Email**.

**DocumentRoot:** Esta directiva define la ruta absoluta donde se almacenarán los archivos HTML que se desean publicar.

Valor asignado: `/usr/local/apache/htdocs`

Valor preinstalado: `/var/www/html`

En el directorio `htdocs` es donde se debe poner la página principal con el nombre `Index`. Si se requiere que la página este dentro de otra carpeta se debe indicar la ruta: `/usr/local/apache/htdocs/principales`. No se indica el documento `Index`. Para ver esta página en el navegador se escribe: `http://localhost/principales`

**ServerRoot:** Esta directiva define la ruta absoluta donde los directorios `conf` y `logs` podrán ser encontrados.

Valor por Default: `/usr/local/apache`

Apache cuenta con dos archivos donde residen las bitácoras:

↳ `acces.log`: Registra todos los accesos al sitio

↳ `error.log`: Registra los errores que genere un acceso al sitio o que los procesos de Apache reporten

**MaxClients:** Establece un límite al total de los procesos del servidor (es decir, clientes conectados simultáneamente) que se ejecutan a la vez. Se debe mantener esta directiva a un valor alto porque no se permitirán nuevas conexiones una vez que se alcance el número máximo de clientes simultáneamente conectados. Por default son 20.

---

<sup>18</sup>Bowen Rich & Coar Ken, “Servidor Apache al Descubierto”, Madrid 2000, Prentice Hall P.p.115

**MinSpareServers y MaxSpareServers:** El servidor Web Apache se adapta dinámicamente a la carga percibida manteniendo un número apropiado de servidores libres basado en el tráfico. El servidor comprueba el número de servidores que esperan peticiones y elimina algunos si el número es más alto que MaxSpareServers o crea algunos si el número de servidores es menor que MinSpareServers.

“El valor predeterminado de MinSpareServers es **5** y el de MaxSpareServers es **20**. Estos valores predeterminados son suficientes en la mayoría de los casos. El número de **MinSpareServers** no debería de ser elevado ya que creará una gran carga incluso cuando el tráfico fuese bajo.

**StartServers:** Establece cuantos procesos serán creados al arrancar. Ya que el servidor Web crea y elimina dinámicamente servidores según el tráfico. Por default son 5<sup>19</sup>

**ErrorDocument:** En caso de que un error o problema ocurra con Apache cuando se solicite un recurso, este puede ser configurado para que haga una de las siguientes acciones:

1. Enviar un mensaje de error (default).
2. Enviar un mensaje personalizado.
3. Redirigir a un URL local para manejar el problema o error.
4. Redirigir a un URL externo quien manejará el problema o error.

Por ejemplo:

ErrorDocument 403 “Lo sentimos en este momento no es posible atenderle”

Para hacer una página de error en HTML se edita una pagina HTML con un mensaje de error, se guarda en el directorio htdocs y en el httpd.conf, en la directiva de error se escribe la ruta.

Ejemplo: ErrorDocument 404 /documentoerror.html

Solo puede ser habilitada una de estas dos opciones, ya sea el mensaje escrito o la página HTML.

**PidFile:** La directiva PidFile define el lugar donde se almacenará el archivo PidFile (ya mencionado antes) Default: logs/httpd.pid

**ErrorLog <filename>:** Esta directiva define el nombre del archivo en el cual el servidor registrará los errores que se presenten.

**CustomLog <filename>:** Esta directiva define el nombre del archivo en el cual el servidor registrará los accesos que se presenten.

---

<sup>19</sup> <http://www.europe.redhat.com/documentation/rh17.1/rhl-rg-es-7.1/ch-configuration.php3>

**HostNameLookup:** HostNameLookup [on|off ] Esta directiva habilita la resolución de nombres en el DNS cuando se establece una conexión. Por razones de rendimiento se sugiere no habilitarla. Para la resolución de nombres en las bitácoras existe el programa “logresolve”, que puede realizar la resolución de nombre fuera de línea.

**<Directory> y <DirectoryMatch>:** La directiva Directory permite aplicar otras directivas de forma específica a todos los recursos dentro de la ruta definida. Ejemplo: teniendo la carpeta público y privado en htdocs se pueden mostrar unas páginas y otras no:

```
<Directory /usr/local/apache/htdocs/privado>
  options -Indexes                               Es privado
</Directory>
```

```
<Directory /usr/local/apache/htdocs/publico>
  options +Indexes                               Es público
</Directory>
```

La directiva **options** controla que características del servidor están disponibles para un directorio en particular.

## Virtual Host de Apache

“Un modo de configurar una máquina con múltiples sitios Web es utilizar una dirección IP diferente para cada sitio. Esto es lo que se denomina hospedaje virtual basado en IP, o basado en direcciones. Para configurar el servidor Web para escuchar múltiples direcciones y responder de un modo diferente a cada uno se utiliza la directriz <VirtualHost>.”<sup>20</sup>

**VirtualHost:** 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.

```
<VirtualHost servername>
    .
    .
    .
</VirtualHost>
```

Un ejemplo es el siguiente:

```
<VirtualHost 132.248.75.IP:8080>
  ServerAdmin root@atena.mascarones.unam.mx
  DocumentRoot /home/virtualservers/8080/htdocs
  ServerName 132.248.75.IP:8080
</VirtualHost>
```

---

<sup>20</sup> Bandel David y Napier Robert, op. Cit, nota 5 Capitulo 1 pp. 816



Se necesita de un servidor DNS para cambiar el nombre de la dirección. En el directorio /etc/resolv.conf se edita el archivo resolv.conf y se indica la dirección del servidor DNS. Los listen deben estar habilitados.

```
<VirtualHost 192.168.40.5>
  ServerAdmin webmaster@pegaso.mascarones.unam.mx
  ServerName intranet179.empresa.com.mx
  DocumentRoot /home/virtualservers/intranet179/htdocs
</VirtualHost>
```

## Control de Acceso

Apache ofrece la funcionalidad al estilo “ACL”, con lo cual es factible determinar las direcciones IPs y los usuarios que tendrán acceso sobre recursos específicos del servidor Web.

La configuración de acceso vía nombre de dominio o vía IP puede realizarse con ayuda de las siguientes directivas:

**Order:** Define el orden en que las directivas allow o deny serán implementadas

**Allow:** Define los hosts que tendrán acceso al recurso.

**Deny:** Define los hosts que no tendrán acceso al recurso.

La definición de los nombres se puede realizar mediante alguna de las siguientes formas:

- ↗ El nombre parcial de un dominio parcial: .atacantes.com.ar
- ↗ Una dirección IP: 200.38.166.1
- ↗ La pareja Red y máscara: 10.2.0.0/255.255.255.0
- ↗ Un dirección de red definida por Classless Inter-Domain Routing (CIDR)  
10.2.2.110/24

**order allow, deny:** Permite explícitamente a los clientes definidos en allow, niega a todos los demás. Los clientes que se encuentren en ambas directivas (allow y deny) serán denegados. *“Todo lo que no está explícitamente permitido está prohibido”*

**order deny,allow:** Niega explícitamente a los clientes definidos en deny, permite a todos los demás. Los clientes que se encuentren en ambas directivas (allow y deny) serán permitidos. *“Todo lo que no está explícitamente prohibido está permitido”*

Ejemplo:

```
<Directory /usr/local/apache/htdocs/intranet >
  order allow,deny
  Allow from 192.168.0.0/255.255.255.0
  Deny from all
</Directory>
```

También es posible asignar un login y password por usuario, quienes tendrán acceso sobre recursos específicos del servidor Web.

**AuthUserFile <archivo de autenticación>:** Indica cual es el archivo que contiene la lista de usuarios y passwords para realizar la autenticación.

Ejemplo: `usr/local/apache/conf/.htpasswd`

**AuthGroupFile <archivo de autenticación>:** Define cual es el archivo que contiene la lista de grupos y los usuarios que la conforman para realizar la autenticación.

Ejemplo `/dev/nul`

**AuthName “banner-string”:** Define el nombre de autorización para el recurso protegido, éste aparecerá como un mensaje en la caja de diálogo que solicita el password para acceder al recurso protegido.

**require valid-user:** Esta directiva indica que se requiere de un usuario y una clave de acceso para un recurso determinado

Un Ejemplo en general de autenticación es:

```
<Directory /usr/apache/htdocs/intranet >
  AuthUserFile /usr/apache/conf/claves/.htpasswd
  AuthGroupFile /dev/null
  AuthName “Bienvenidos a la Intranet”
  AuthType Basic
  require valid-user
</Directory>
```

Dentro de un VirtualHost se debe crear un directorio Intranet y las claves.

## Módulos

“Sin los módulos, Apache no sería tan capaz o popular como lo es hoy en día.[...] La modularidad de los programas es un concepto clave. Sus ventajas son una mejor capacidad de ampliación (ampliar el programa añadiendo opciones), una mejor descomposición (desarrollando y probando opciones independientemente la una de la otra) y la flexibilidad de la configuración (activar o desactivar opciones en función de las necesidades).”<sup>21</sup>

**Módulos estáticos:** Se agregan al momento de compilar apache.

**Módulos dinámicos:** Se agregan durante el ambiente productivo, permiten adicionar funcionalidad sin necesidad de recompilar apache. PHP se puede configurar como modulo DSO en Apache.

---

<sup>21</sup> Bowen Rich & Coar Ken, , op. Cit, nota 18 pp. 358

Un sitio Web dinámico puede ser creado con lenguajes como: PHP, JSP, ASP, CGIs, etc. La tecnología CGI “Common Gateway Interface” define un modelo de programación que puede ser implementado por múltiples lenguajes.

Los CGI's son programas que siguen un estándar definido, corren en el servidor, reciben parámetros desde el cliente y su salida es enviada al navegador. Fueron la primera alternativa para generar dinamismo a un sitio Web.

**ScriptAlias <alias><path-filesystem>:** Convierte las solicitudes vía URL al path absoluto donde residen los programas cgi. Para echar a andar los CGIs:

```
ScriptAlias /cgi-bin “/usr/local/apache/cgi-bin”
```

**AddHandler cgi-scripts .cgi .pl:** AddHandler permite que determinada extensión sea relacionada a un evento en particular, en el caso de los CGIs lo que se indica es que las extensiones .cgi y .pl quedan identificadas como un script. Apache las interpretará en lugar de solo enviar el contenido del archivo al cliente.

**Conclusión:** El servidor Apache es una buena opción para ser utilizado como servidor de páginas HTML, ya que se puede descargar de manera gratuita y ser configurado para las necesidades del sitio Web o de la empresa que lo maneje. Aunque, es necesario conocer bien el funcionamiento de las directivas y módulos para poder aprovechar al máximo los recursos del servidor.

## Capítulo IV

### PHP

**Objetivo:** Conocer el lenguaje de programación PHP y aprender a utilizarlo para la creación de páginas dinámicas, así mismo poder modificar el código PHP que se presente y manipularlo como mejor se requiera

**Justificación:** Las páginas estáticas que se pueden crear con el lenguaje HTML no siempre cumplen con las necesidades del usuario, es por esto que el lenguaje PHP nos permite crear páginas dinámicas que realizan tareas más complejas.

**Introducción:** PHP es un lenguaje de programación usado generalmente para la creación de contenido de sitios Web, se trata de un lenguaje interpretado<sup>22</sup> usado para elaborar aplicaciones para servidores o creación de contenido dinámico para páginas en Internet.

#### Contenido:

PHP fue originalmente diseñado en Perl por Rasmus Lerdorf en el año de 1994 para mostrar su currículum vitae y guardar ciertos datos, en un principio fue conocido como "Personal Home Page Tools".

En 1997 se desarrolla la versión 2.0 denominada PHP/FI(Forms Interpreter) y un año después los programadores Zeev Suraski y Andi Gutmans renuevan el lenguaje creando PHP 3.0 y a partir de aquí se utiliza el acrónimo recursivo de "PHP: Hypertext Preprocessor". En 1999 se libera la versión 4.0 y en 2004 se crea el motor Zend 2.0 y se libera PHP 5.0 que introduce un modelo de orientación a objetos muy similar al de Java.

Cuando un cliente hace una petición al servidor para que le envíe una página Web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado y genera el contenido dinámico que regresa al servidor y este lo envía al cliente.

Si ya se cuenta con un servidor Apache instalado en el sistema Linux Slackware solo se debe descargar el archivo php-xxxx.tar.gz donde x es la versión, que se requiere de PHP e instalarlo.

#### Descarga e instalación:

PHP se puede descargar de las siguientes direcciones de manera gratuita:

<http://www.apache.org/>  
<http://www.php.net/>

---

<sup>22</sup> Los lenguajes interpretados van siendo codificados por la computadora mientras se están ejecutando

“El intérprete PHP puede ser ejecutado por el servidor Web de dos modos distintos: como módulo del propio servidor (interfaz SAPI, Server Application Programming Interface) o como programa externo a éste (modo CGI, Common Gateway Interface).

**Como 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, como son: cargar el programa en memoria, anotarlo en la lista de tareas, lanzar su ejecución esperar que termine y descargarlo de memoria y lista de tareas. Tantas veces un cliente pida la ejecución de un programa CGI, tantas veces se repetirán estas acciones en la máquina servidora.

**Como Módulo:** La ejecución se hace en el espacio de la memoria del propio servidor. Es más rápido y eficiente ya que las ejecuciones del programa, las realiza el servidor Web (No habrá creación de procesos nuevos ni ocupación extra de la memoria física con copias de éstos).”<sup>23</sup>

Para instalar la versión que se descargó como módulo del servidor primero hay que detener el servidor Apache, descomprimir el archivo, acceder a la carpeta que se crea al descomprimir el archivo, e indicar en que dirección se debe instalar PHP, esto debe de ser desde la consola de comandos, las instrucciones son las siguientes:

```
☞ tar -zxvf php-4.2.3.tar.  
☞ cd php-4.2.3/  
☞ ./configure --prefix=/usr/local/php4 --with-apxs2=/usr/local/apache2/bin/apxs --  
with-mysql  
☞ make  
☞ make install  
☞ cp php.ini-dist /usr/local/php4/lib/php.ini
```

Después de esto se deben agregar las siguientes líneas al archivo httpd.conf en la sección del Script Alias:

```
☞ AddType application/x-httpd-php .php .phtml  
☞ AddType application/x-httpd-php-source .phps
```

Hay que verificar que exista la línea: LoadModule php4\_module modules/libphp4.so.  
Este archivo se encuentra en /usr/local/apache/conf

Para comprobar que la instalación finalizó de manera correcta se crea un archivo en la carpeta DocumentRoot de Apache con nombre info.php que tenga el siguiente contenido:

```
<? Phpinfo();?>
```

---

<sup>23</sup> Guíérrez Abraham & Bravo Ginés, “PHP5 a través de ejemplos”, México 2005, Alfaomega RA-MA, p.p.1

Se reinicia el servidor apache y se teclea en el navegador: <http://localhost/info.php>. Si se presenta una pagina que muestra en varias tablas las características del PHP que se instalo, quiere decir que el PHP se instalo de manera correcta.

## Características del lenguaje PHP

Las características más importantes del lenguaje PHP son:

- “Es un lenguaje multiplataforma
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones)
- Posee una amplia documentación en su página oficial
- Es libre por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos
- Permite crear formularios para la Web”<sup>24</sup>

PHP puede ser mezclado con lenguaje HTML. Permite la conexión a diferentes tipos de servidores de Bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, Microsoft SQLServer, Firebird y SQLite, con esto se pueden crear aplicaciones robustas.

También tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX, Linux, Windows y Mac OS X, y puede interactuar con los servidores Web más populares como ISAP, IIS y Apache

## Código PHP

“Un punto importante es que la sintaxis de PHP deriva o es similar a la del lenguaje C, que es muy popular: el que tenga conocimientos en este lenguaje se acercará con mas facilidad a PHP. La meta de este lenguaje es permitir escribir a los creadores de sitios Web páginas dinámicas de una manera rápida y fácil.”<sup>25</sup>

El lenguaje PHP tiene delimitadores que indican donde inicia el código y también indican donde termina. Estos delimitadores son:

```
<?php ?>    <?    ?>    <%    %>
```

Cuando se incluye código PHP en código HTML los delimitadores también permiten visualizar mejor el código.

---

<sup>24</sup> <http://es.wikipedia.org/wiki/Php>

<sup>25</sup> Minera Francisco José, PHP y MySQL integración total, Argentina 2005, MP Ediciones, p.p. 28

Cuando se crea una página con lenguaje PHP se debe guardar con extensión .PHP, aunque contenga código HTML. Como ya se mencionó PHP es un lenguaje de programación y permite crear aplicaciones complejas, cuenta con estructuras de control, funciones y manejo de variables.

## Variables

Una variable es un contenedor de información, en el que es posible almacenar números enteros, números decimales, caracteres, etc., el contenido de las variables se puede leer y se puede cambiar durante la ejecución de una página PHP.

En PHP todas las variables comienzan con el símbolo del dólar \$ y no es necesario definir una variable antes de usarla. Tampoco tienen tipos, es decir que una misma variable puede contener un número y luego puede contener caracteres. Para declarar una variable la sintaxis es:

\$nombre\_variable

PHP puede diferenciar entre minúsculas y mayúsculas. Soporta tres tipos de datos simples:

- ❖ **Integer:** Números enteros que varían entre -2 billones y 2 billones, se puede representar en formato decimal, octal o hexadecimal
- ❖ **Float:** Números con decimales
- ❖ **String:** Cadenas de caracteres. Una cadena está formada por cero o más caracteres encerrados entre "" o bien ''

Algunos caracteres especiales para las cadenas:

- |                      |                     |
|----------------------|---------------------|
| ❖ \n Cambio de línea | ❖ \' Comilla simple |
| ❖ \t Tabulación      | ❖ \" Comilla doble  |
| ❖ \\ Barra inversa   | ❖ \\$ caracter \$   |

También soporta dos tipos de datos compuestos:

- ❖ **Array:** Arreglos, son estructuras que permiten el almacenamiento de un conjunto de datos bajo el mismo nombre. En PHP los elementos que componen el arreglo pueden ser de distintos tipos.
- ❖ **Objet:** Es una estructura que define características propias (denominadas propiedades) y sus funcionalidades (denominadas métodos).

Cuando se utilizan variables de distinto tipo, el intérprete de PHP tiende a homogeneizar sus diferentes tipos en función de la operación que se pretende realizar. Es posible convertir las variables a un tipo específico, si se desea, para esto se escribe entre paréntesis el tipo deseado antes de la variable. Ejemplo:

(float)\$cadena

Una constante es una variable que mantiene el mismo valor durante toda la ejecución del programa la sintaxis de definición de constantes es:

define("constante" , valor)

## Operadores

Un operador es un símbolo matemático que denota un conjunto de operaciones que han de realizarse. Estos son los operadores que se pueden aplicar a las variables y constantes numéricas.

<b>Operadores aritméticos</b> <sup>26</sup>	
+	Suma dos valores
-	Resta dos valores (o pasa a negativo un valor)
*	Multiplica dos valores
/	Divide dos valores
%	Resto de dividir dos valores
++	Incremento en una unidad
--	Decremento en una unidad

<b>Operadores de asignación</b>	
=	Asigna a la parte derecha el valor izquierdo
+=	Realiza la suma de la derecha con la izquierda y la asigna a la derecha
-=	Realiza la resta de la derecha con la izquierda y la asigna a la derecha
* =	Realiza la multiplicación de la derecha con la izquierda y la asigna a la derecha
/=	Realiza la división de la derecha con la izquierda y la asigna a la derecha
%=	Se obtiene el resto y se asigna
.=	Concatena el valor de la izquierda con la derecha y lo asigna a la derecha

<b>Operadores lógicos</b>	
!	Operador NO o negacion.Si era true pasa a false y viceversa
and	Operador Y, si ambos son verdaderos vale verdadero
or	Operador O, vale verdadero si alguno de los dos es verdadero
xor	Verdadero si alguno de los dos es true pero nunca ambos
&&	True si ambos lo son
	True si alguno lo es

<sup>26</sup> <http://www.webtaller.com/manual-php/operadores.php>



<b>Operadores condicionales</b>	
==	Comprueba si dos números son iguales
!=	Comprueba si dos números son distintos
>	Mayor que, devuelve true en caso afirmativo
<	Menor que, devuelve true en caso afirmativo
>=	Mayor o igual
<=	Menor o igual

<b>Operadores de cadenas</b>	
.	Concatenación de dos cadenas
.=	Concatenación y asignación

## Estructuras de control

“Las estructuras de control o sentencias de control permiten modificar el flujo de ejecución básico del programa, es decir, gracias a ellas la ejecución no tiene que ser totalmente secuencial, es posible controlar el flujo lógico de cualquier programa.”<sup>27</sup>

Se puede dividir el flujo del programa (estructuras condicionales) o bien que determinado código se ejecute un determinado número de veces (estructuras cíclicas).

Las estructuras condicionales permiten elegir diferentes caminos de ejecución cuando se cumple una condición. Las estructuras condicionales en PHP son:

**If**, su sintaxis es:

```
if(expresión)
{ sentencias; }
```

O bien **if...else**, su sintaxis es:

```
if(expresión)
{ sentencias; }
else{ sentencias; }
```

La expresión o condición debe ser lógica, es decir que devuelva verdadero o falso

Y **switch**, que se utiliza para comparar un dato con un conjunto de posibles valores su sintaxis es:

<sup>27</sup> Guiérrez Abraham & Bravo Ginés, Op. Cit nota 23, p.p.65

```

Switch ($variable){
    Case valor 1:
        sentencias ;
        break;
    Case valor 2:
        sentencias ;
        break;
    .....

    Case valor N:
        sentencias ;
        break;
    Default:
        sentencias;
    }

```

Las estructuras cíclicas se utilizan para ejecutar una o mas instrucciones un determinado número de veces, generalmente se utilizan para contar o para recorrer los elementos de un arreglo.

En PHP existen 4 tipos:

1. **for**: Permite realizar un conjunto de instrucciones un determinado número de veces. Su sintaxis es:

```

for(inicialización; condición; incremento){
    sentencias;
}

```

2. **foreach**: Se utiliza para recorrer las estructuras de tipo arreglo, obteniendo en cada iteración uno de sus elementos componentes. Su sintaxis:

```

foreach(nombre_arreglo as $valor){
    sentencias;
}

```

3. **while**: Se ejecuta un número indeterminado de veces, siempre y cuando el resultado de comprobar la condición sea verdadero. Su sintaxis es:

```

while(condición){
    sentencias;
}

```

4. **do while**, esta sentencia realiza lo mismo que while, la diferencia es que por lo menos se ejecuta una vez, ya que la condición se evalúa al final del ciclo. su sintaxis es:

```

do{

```

```
    sentencias;  
} while(condición);
```

**break:** se utiliza para forzar la terminación de un ciclo o en el caso del switch para que no se sigan evaluando los case.

**Continue:** se utiliza dentro de los ciclos cuando se requiere que no se efectúen una serie de instrucciones del ciclo y se pretende pasar a la siguiente iteración.

## Funciones

“Una función es el nombre que se le da a un bloque de código que realiza una tarea específica posiblemente sustituyendo un conjunto de valores que se le den, o parámetros, y posiblemente regresando un simple valor. Las funciones en un programa de PHP pueden ya estar hechas o definidas por el usuario”<sup>28</sup>

Se utiliza para dividir el código en partes menores. A una función se le asigna un nombre y su sintaxis es la siguiente:

```
function nombre_función (parámetros){  
    sentencias;  
    return valor;  
}
```

El uso de funciones da la capacidad de agrupar varias instrucciones bajo un solo nombre y poder llamarlas varias veces desde diferentes sitios, sin la necesidad de escribirlas de nuevo. La lista de sentencias puede incluir código HTML. Se puede declarar una función PHP que no contenga código PHP.

Opcionalmente es posible pasarle parámetros a las funciones que se trataran como variables locales y así mismo es posible devolver un resultado con la instrucción *return* valor; Esto produce la terminación de la función retornando un valor.

Las funciones para variables permiten realizar algunas acciones como son:

- **gettype:** Devuelve el tipo de dato del parámetro.
- **settype:** Establece el tipo de dato a guardar en una variable, realiza conversiones de tipo de datos.
- **isset:** Devuelve true si una variable ha sido inicializada con un valor, de lo contrario devuelve false.
- **empty:** Devuelve true si la variable no ha sido inicializada, si tiene un valor 0 ó si es una cadena vacía, de lo contrario devuelve false.
- **is\_int:** True si la variable es entera.

En el lenguaje PHP el tratamiento de cadenas es muy importante, existen varias funciones para el manejo de cadenas, algunas de ellas son:

---

<sup>28</sup> Lerdorf Ramus and McGredy Ric, “Programming PHP”, USA 2002, O’REILLY, p.p. 61

- **strlen**(cadena). Devuelve el número de caracteres de una cadena.
- **split**(separador, cadena). Divide una cadena en varias usando un caracter separador.
- **sprintf**(cadena de formato, var1, var2...). Formatea una cadena de texto al igual que printf pero el resultado es devuelto como una cadena.
- **substr**(cadena, inicio, longitud). Devuelve una subcadena de otra, empezando por inicio y de longitud *longitud*.
- **chop**(cadena). Elimina los saltos de línea y los espacios finales de una cadena.
- **strpos**(cadena1, cadena2). Busca la *cadena2* dentro de *cadena1* indicándonos la posición en la que se encuentra.
- **str\_replace**(cadena1, cadena2, texto). Reemplaza la *cadena1* por la *cadena2* en el *texto*.
- **trim**(cadena). Devuelve la cadena sin los espacios en blanco que se encontraban al inicio y al final de la cadena
- **strtoupper**(cadena). Convierte una cadena a mayúsculas
- **strtolower**(cadena). Convierte una cadena a minúsculas
- **explode**(patron, cadena). Devuelve un array, resultado de dividir *cadena* en subcadenas debido al criterio de separación dado por *patron*
- **implode**(nexo, cadena). Devuelve una cadena, resultado de unir todos los elementos de un array separados por un *nexo*.

## Archivos

El almacenamiento permanente de información es necesario en muchas ocasiones. Por esto es necesario realizar operaciones con archivos. PHP provee funciones que permiten el manejo de archivos.

Las operaciones que se pueden realizar con los archivos son:

- 1) Abrir, sintaxis: **fopen**(nombre\_archivo, modo\_apertura).

Modo de apertura	Acción
r	Lectura. El apuntador se coloca al inicio del archivo.
r+	Lectura y escritura. El apuntador se coloca al inicio del archivo.
w	Escritura. Si no existe el archivo se crea, si ya existe se borra su contenido.
w+	Lectura y escritura. Si no existe el archivo se crea, si ya existe se borra su contenido.
a	Escritura. Si no existe el archivo se crea, si ya existe se coloca al final del archivo para añadir datos.
a+	Lectura y escritura. Si no existe el archivo se crea, si ya existe se coloca al final del archivo para añadir datos.

- 2) Cerrar, sintaxis: **fclose**(apuntador) Es recomendable cerrar un archivo cuando ya no se va usar, si no se cierra PHP lo hará al terminar de ejecutar el script.

3) Leer, sintaxis:

- a) **fgetc**(apuntador), Devuelve un caracter del archivo referenciado por *apuntador*, si se ha llegado al final del archivo devuelve false.
- b) **fgets**(apuntador,[total\_caracteres\_a\_leer]) . Devuelve una cadena de total de caracteres a *leer-1* o de menor longitud si se ha encontrado un cambio de línea que se incluiría en la cadena a devolver
- c) **fread**(apuntador,[total\_caracteres\_a\_leer]). Igual a fgets() sólo que no deja de leer cuando encuentra un cambio de línea y devuelve total de caracteres a leer.
- d) **feof**(apuntador). Devuelve true si se ha llegado al final del archivo.
- e) **file**(nombre\_archivo). Lee todo el contenido de un archivo y lo devuelve en forma de array: una línea en cada posición del arreglo.
- f) **readfile**(nombre\_archivo). Lee el contenido de un archivo y lo muestra por la salida estándar.

4) Escribir, sintaxis:

- a) **fwrite**(apuntador,cadena)
- b) **fputs**(apuntador,cadena). Ambas funciones escriben la cadena pasada como parámetro, devuelven el total de caracteres escritos o false si se produjo algún error.
- c) **rewind**(apuntador). Sitúa el apuntador de lectura/escritura al principio del archivo.
- d) **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, que le indican que se desplaza n *desp* a partir del principio, posición actual o final del archivo respectivamente (en este caso *desp* debe ser negativo).
- e) **ftell**(apuntador). Devuelve la posición actual del apuntador.

Algunas funciones para archivos son:

- **file\_exists**(archivo). Regresa true si el archivo existe
- **filesize**(archivo). Devuelve el tamaño en bytes del archivo.
- **copy**(archivo\_origen, archivo\_destino). Devuelve true si se logró copiar el archivo.

Una mejor opción para almacenar información es con las bases de datos. Como ya se mencionó PHP permite hacer conexiones de diversos servidores a bases de datos.

**Conclusión:** El lenguaje PHP es muy útil para la creación de Sitios Web dinámicos, ya que al ser un lenguaje de programación nos permite manipular en gran medida los elementos de las páginas y su característica de poder mezclar código HTML con elementos PHP lo hace una herramienta muy eficiente.

## Capítulo V

### Diseño de bases de datos

**Objetivo:** Aprender las técnicas necesarias para poder diseñar bases de datos basadas en el modelo relacional, así como identificar sus aplicaciones y conocer los conceptos mas usados de éstas.

**Justificación:** La mayoría de los sistemas informáticos utilizan una base de datos para poder manejar y almacenar la información que se produce en la empresa o centro de trabajo, por esto es necesario conocer como se diseña una BD y cuales son sus características.

**Introducción:** Todos generamos y utilizamos información, desde información escolar hasta información de las grandes empresas, esto nos permite tomar decisiones importantes. Si la información es mucha se crea un histórico. Hoy en día la información se almacena en medios electrónicos, sobretodo en archivos, y cuando los niveles de información son muy grandes se utilizan las bases de datos.

### Contenido

La información es un conjunto de datos significativos estructurados lógicamente, un dato es un conjunto de caracteres que por si solos no proporciona ninguna información en informática, por ejemplo “cuadro”. Y un carácter es cualquier símbolo numérico, alfabético, alfanumérico y caracteres especiales incluido el espacio en blanco, como son: 2, %, T, a, ,etc.

Una base de datos es un conjunto de tablas dedicadas a guardar información relacionada entre sí, con referencias entre ellas de manera que se complementen con el principio de no duplicidad de datos.

Una tabla es un conjunto de campos y registros, a una tabla también se le llama entidad y puede representar cualquier cosa. Un campo es una característica o atributos que representan o describen a una tabla también se le llama columna y los registros son filas que representan un conjunto de campos y que proporcionan información. Ejemplo:

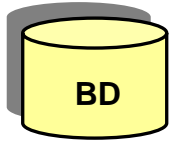
TABLA

NOMBRE	APELLIDO	DIRECCION	DELEG	CP
Mónica	López	Arte 159	Tlalpan	50100
Jorge	Ruiz	Tlacopac 18	San Ángel	18000
Rosa	Sandoval	Loreto 26	Tizapán	21400
Cristian	Martínez	Colima 33	Roma	04200

CAMPO

Fig.1 Ejemplo de tabla

El símbolo de una BD es el contenedor (Fig.2), ya que una BD es el contenedor de información de un tema en específico en donde se organiza la información mediante tablas relacionadas entre sí con la finalidad de que la información sea: concisa, integra y se mantenga una alta seguridad.



Es recomendable tener varias BD en un sistema informático grande, no es práctico almacenar toda la información en una sola base, por ejemplo en una empresa es conveniente tener una BD para nómina, otra para proyectos, otra para clientes, etc.

Fig.2

## Tipos de bases de datos

**“Bases de datos jerárquicas:** Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol visto al revés, en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.”<sup>29</sup>

**Bases de datos de red:** Los datos se representan mediante colecciones de registros y las relaciones entre los datos se representan mediante enlaces, los cuales pueden verse como punteros. Los registros en la base de datos se organizan como colecciones de grafos arbitrarios.

**Bases de datos relacionales:** Representan los datos y las relaciones entre los datos mediante una colección de tablas, cada una de las cuales tienen un número de columnas con nombres únicos.

La relación es el elemento básico del modelo relacional y se puede representar como una tabla (aunque tiene una serie de elementos característicos que la distinguen de la tabla), ya que no se admiten filas duplicadas. En el cruce de una fila y de una columna sólo puede haber un valor.

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar.

La información puede ser recuperada o almacenada mediante "consultas" que ofrecen flexibilidad y poder para administrar la información. El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL (*Structured Query Language* o *Lenguaje Estructurado de Consultas*) un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

---

<sup>29</sup> [http://es.wikipedia.org/wiki/Bases\\_de\\_datos](http://es.wikipedia.org/wiki/Bases_de_datos)

**Bases de datos orientadas a objetos:** Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, incorpora todos los conceptos importantes del paradigma de objetos:

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

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos. La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz.

Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

## Conceptos asociados a las bases de datos relacionales

**Relación:** Es una vinculación entre ideas, una forma de asociar entidades para lograr el objetivo en común que tienen. Una relación es una asociación entre identidades.

“**Atributo:** Un atributo define y diferencia una entidad de otra. Los atributos pueden verse como características de las entidades.”<sup>30</sup>

**Redundancia:** Se refiere a reducir en lo posible la repetición de una misma información (característica o atributo) a través de la organización, es decir, que la información solo exista una vez sobre cualquier medio

**Integridad:** Es la capacidad de ejecutar aisladamente las transacciones, es decir sin que otra transacción se ejecute concurrentemente.

**Seguridad:** Es la protección contra los accesos mal intencionados a la base de datos. Y para proteger a la base de datos se adoptan medidas de seguridad de varios niveles: físico, humano, sistema operativo, red y el propio sistema de BD.

“**Independencia Física:** El modo en que se almacenan los datos no debe influir en su manipulación lógica y, por tanto, los usuarios que acceden a esos datos no ha de modificar sus programas por cambios en el almacenamiento físico.

---

<sup>30</sup> Minera Francisco José, Op. Cit., nota 25 Capitulo 4, p.p. 43



**Independencia Lógica:** Añadir, eliminar o modificar cualquier elemento de la base de datos no debe repercutir en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismos.”<sup>31</sup>

**Integridad referencial:** Proporciona un medio para asegurar que los cambios efectuados en la BD por usuarios autorizados no resulten una pérdida de consistencia de los datos, de esta manera las reglas de integridad protegen la base de datos contra daños accidentales. Por lo general se requiere que un valor registrado en una tabla para un conjunto de atributos dados, también aparezca para cierto conjunto de atributos para otra tabla.

**Índices:** Es una estructura de datos en donde se almacena información extra acerca de una columna. Se usan para encontrar filas con valores específicos en alguna de sus columnas. Un tipo de índice son las llaves.

**Claves o llaves:** Toda tabla debe estar asociada con una clave que permita su identificación. A veces la tabla se identifica mediante un único atributo o en ocasiones mediante más. Es el atributo que permite significar de manera única a una entidad (campo a partir del cual se pueden inferir otros campos de una tabla). Existen tres tipos de llaves:

1. Llaves candidatas: Campo único y obligatorio con posibilidades de ser llave la tabla
2. Llave Primaria (Pk – Primary Key): Es el campo con valor único y obligatorio que identificará a cada registro como único y debe ser de preferencia numérico y el valor más pequeño.
3. Llave Foránea (Fk): Es la llave primaria de una tabla que será consultada por otra. Lo que permitirá la interrelación de las tablas (compartir datos y minimizar redundancia)

## **Pasos implicados para la construcción de una base de datos**

**Análisis de las tareas:** Para la creación de una BD se necesita analizar el proceso automatizado, hay que conocer la actividad de principio a fin, en este primer paso se debe hacer un listado de todas las tareas que se requieren.

**Determinación de la finalidad de la base de datos:** Expresar que es lo que se pretende que realice la base de datos, por ejemplo: Llevar el control de vehículos de un lote de autos, de acuerdo a sus características.

**Esquematación del flujo de las tareas:** Ordenar temas, identificar tareas relevantes e identificar secuencias de tareas.

---

<sup>31</sup> Castaño Adoración de Miguel y Piattini Mario, “Fundamentos y modelos de bases de datos”, 2<sup>da</sup> Edición, México 1999, Alfaomega-Ra-Ma, p.p 124

**Determinación de las tablas de la base de datos:** En este paso podemos sugerir que tablas estarán presentes en la base de datos. Se recomienda que las tablas sean representadas como rectángulos con esquinas redondeadas, que el título de la tabla este en mayúsculas y sus campos en minúsculas.

**Determinar que campos se requieren en las tablas:** Por ejemplo la tabla AUTOS debe tener los campos de: placas, submarca, modelo, transmisión, etc. Aquí se definen los atributos de estos campos, si debe ser un campo único, obligatorio u opcional.

También se debe determinar el tipo de dato y longitud de los campos, estos pueden ser texto, numérico, boleano, fecha/hora o definido por el usuario.

## Normalización

Después de determinar las tablas y sus valores se deben normalizar las tablas, la normalización “es el proceso que consiste en evitar redundancias, es decir, que la información esté duplicada ya que, si hubiera que cambiar un dato que estuviera repetido, habría que cambiarlo varias veces.”<sup>32</sup> Se deben cumplir tres formas normales:

Primera forma normal 1FN

- ❖ Cada atributo debe tener uno y solo un valor
- ❖ Los atributos deben de ser descompuestos a su mínima expresión (valor atómico)
- ❖ Las columnas repetidas deben eliminarse y colocarse en tablas separadas si es necesario

Segunda forma normal 2FN

- ❖ Cumplir con la 1FN
- ❖ Cada atributo no identificador debe depender completamente del identificador de la entidad es decir, que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas

Tercera forma normal 3FN

- ❖ Cumplir con la 1FN y 2FN
- ❖ Ningún atributo no identificador debe depender de otro atributo no identificador

Después de normalizar las tablas se deben determinar los campos llave, si hay campos únicos y obligatorios se considera que son llaves candidatas a ser la llave principal. La llave primaria debe ser un número o dato alfanumérico y debe ser corto. Por ejemplo: placas y No. de serie el motor pueden ser únicas y obligatorias, pero se sugiere que placas sea la llave primaria.

---

<sup>32</sup> Guiérrez Abraham & Bravo Ginés, Op. Cit. Nota 23 Capitulo 4, p.p. 338

Las llaves foráneas son las que nos indican que un campo tiene relación con otra tabla, por ejemplo: la tabla AUTOS tiene el campo submarca que puede ser un identificador numérico, existirá por esto una tabla de nombre SUBMARCA que dependiendo del identificador mostrará una submarca. Esto es:

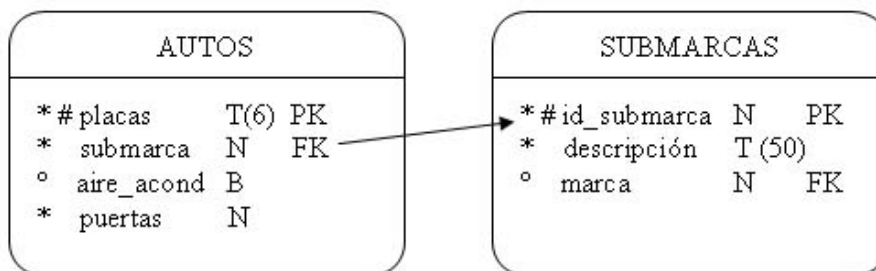


Fig.3 Ejemplo de tablas con llaves

Notación:

# campo único	N numérico	PK llave primaria
* campo obligatorio	T texto	FK llave foránea
° campo opcional	B boleano	

Las llaves foráneas deben ser el mismo tipo de dato en las dos tablas.

Para poder asignar las llaves a las tablas es necesario comprender el concepto de relación, que es una regla de correspondencia entre dos tablas con el objeto de compartir información entre dichas tablas. También se puede decir que es el producto cartesiano de dos o varios dominios.

## Tipos de relaciones

Una relación es una regla de correspondencia entre dos tablas con el objeto de compartir información entre dichas tablas. También se puede decir que es el producto cartesiano de dos o varios dominios.

- ✓ **Relación de uno a uno:** Es la relación de uno a uno entre dos entidades cuando a cada elemento de la entidad A se le asigna un único elemento de la entidad B y para cada elemento de la entidad B existe un único elemento en la entidad A. Es posible hablar de asociación 1 : 1 obligatoria. Ejemplo: una tabla de presidente y una tabla de un país, a cada presidente solo le corresponde un país.
- ✓ **Relación de uno a muchos:** Se dice que existe una relación de una a vario entre una entidad A y una entidad B cuando una clave de la entidad A posee varios elementos relacionados en la entidad B. Ejemplo: una tabla autor y una tabla libros, un autor puede tener varios libros relacionados a el y un libro solo tiene relacionado un autor.
- ✓ **“Relación de muchos a muchos:** Una relación es muchos a muchos entre las entidades A y B cuando una clave de la entidad A posee varios elementos

relacionados en la entidad B y a su vez una clave de la entidad B posee varios elementos en la entidad A.”<sup>33</sup> Por ejemplo: En la universidad varios alumnos pertenecen a distintos grupos y cada grupo tiene muchos alumnos.

La relacion M : M no existe en la práctica dentro de los manejadores de bases de datos por lo que se debe utilizar una tabla intermedia (tabla detalle) que será la tabla que mantendrá los campos llaves de cada una de estas dos tablas que tienen relación y mediante ella se establecerá la relación de 1 : M y M : 1.

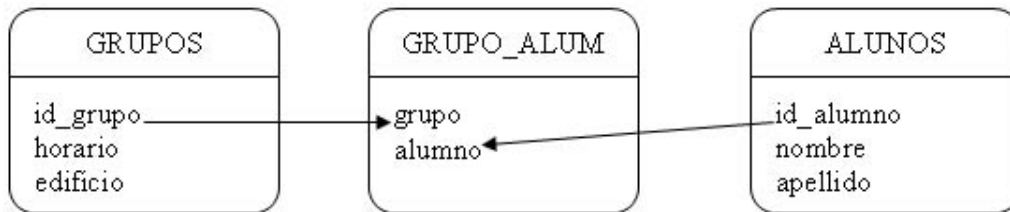


Fig.4 Ejemplo de tabla detalle

## Modelo entidad - relación

Es una herramienta gráfica del diseño de DB, que ayuda a cualquier persona a conocer el funcionamiento general de la misma. Éste se debe leer iniciando en la parte superior izquierda dirigiéndose a la derecha y después hacia abajo.

“El modelo E/R tiene como restricción inherente que sólo permite establecer relaciones entre entidades. También obliga el modelo a que todas las entidades tengan un identificador, lo que así mismo podría considerarse una restricción inherente”<sup>34</sup>

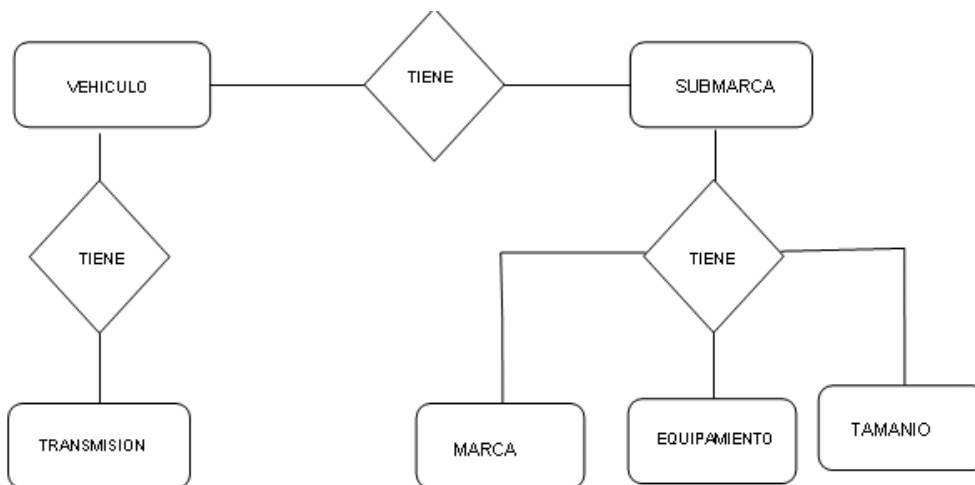


Fig.5 Modelo entidad-relación

<sup>33</sup> Minera Francisco José, Op. Cit., nota 25 Capitulo 4, p.p. 52-53

<sup>34</sup> Castaño Adoración de Miguel, Piattini Mario y Martínez Esperanza Marcos, “Diseño de bases de datos relacionales”, México 2000, Alfaomega-Ra-Ma, p.p 56

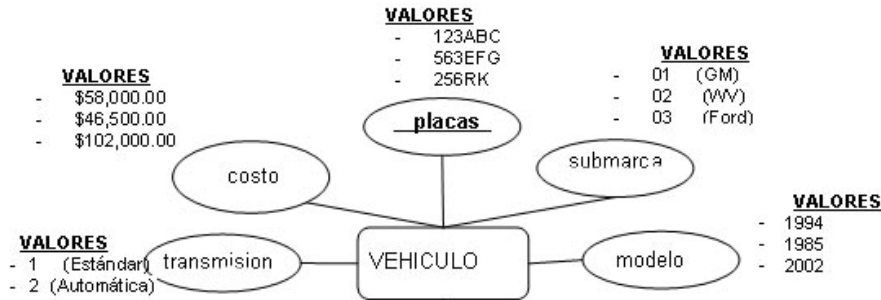


Fig.6 Posibles valores de los atributos

Simbología (Notación)	Significado
Rectángulo	Tabla (Entidad)
Rombo	Relación (Generalmente es verbo-Acción)
Ovalo	Atributos
Atributo subrayado	Clave o llave de la Tabla

Diagrama Entidad – Relación: Este también es un gráfico que se diseña en el sistema manejador de bases de datos (como Access) o bien en modeladores de bases de datos (como Fabulous Force Database Designer 4), ya que es aquí cuando se establecen las relaciones de las tablas, es decir se fijan las llaves foráneas y primarias. Esto después de tener las tablas normalizadas con valores.

## Sistemas Gestores de Bases de Datos

“Los Sistemas Gestores de Bases o sistemas manejadores de DB son un tipo de software muy específico, dedicado a servir de interfaz entre la Base de datos y el usuario. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.”<sup>35</sup>

Las funciones principales de un sistema gestor de DB son:

- ✓ Crear y organizar las bases de datos
- ✓ Establecer y mantener las trayectorias de acceso a la base de datos, para que los datos puedan ser accedidos rápidamente
- ✓ Manejar los datos de acuerdo a las peticiones del usuario
- ✓ Registrar el uso de las bases de datos
- ✓ Respaldo y recuperación de datos si ocurre una falla en el sistema
- ✓ Controlar la interacción de los usuarios concurrentes

“El lenguaje SQL estándar cuenta con elementos destinados a la definición, modificación, control, protección de los datos y acceso a bases de datos distribuidas en una red, haciendo factible que varios usuarios puedan consultar la misma base de datos de forma simultánea.”<sup>36</sup>

<sup>35</sup> [http://es.wikipedia.org/wiki/Sistema\\_Gestor\\_de\\_Bases\\_de\\_Datos](http://es.wikipedia.org/wiki/Sistema_Gestor_de_Bases_de_Datos)

<sup>36</sup> Microsoft® Encarta® 2007. © 1993-2006 Microsoft Corporation. Reservados todos los derechos

El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por un motor de base de datos. El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado.

Existen dos tipos de comandos SQL:

Los DDL que permiten crear y definir nuevas bases de datos, campos e índices.

Los DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

Comandos DDL:

**CREATE** Utilizado para crear nuevas tablas, campos e índices

**DROP** Empleado para eliminar tablas e índices

**ALTER** Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

Comandos DML:

**SELECT** Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado

**INSERT** Utilizado para cargar lotes de datos en la base de datos en una única operación.

**UPDATE** Utilizado para modificar los valores de los campos y registros especificados

**DELETE** Utilizado para eliminar registros de una tabla de una base de datos

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

**FROM** Utilizada para especificar la tabla de la cual se van a seleccionar los registros

**WHERE** Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar

**GROUP BY** Utilizada para separar los registros seleccionados en grupos específicos

**HAVING** Utilizada para expresar la condición que debe satisfacer cada grupo

**ORDER BY** Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

Dos sistemas gestores de bases de datos muy conocidos son MySQL y PostgreSQL que serán tratados en el siguiente capítulo.

**Conclusión:** Para poder diseñar una base de datos es necesario conocer las necesidades del sistema y seguir los pasos que se indicaron para que la base sea óptima. Aunque hay muchos conceptos que se deben tomar en cuenta para el diseño la mayoría son fáciles de llevar a la práctica. Lo que si es necesario conocer, si se van a manipular bases en un gestor, es el lenguaje SQL, ya que una parte es diseñar la base y otra programarla.

## Capítulo VI

# MySQL y PostgreSQL

**Objetivo:** Conocer y manejar las sentencias necesarias para la creación y manejo de bases de datos con los gestores MySQL y PostgreSQL. Así mismo conocer como acceder a las Bases de datos con ayuda del lenguaje PHP.

**Justificación:** Es necesario aprender como interactúan las bases de datos en red ya que la mayoría de los sitios Web requieren que los usuarios ingresen información para almacenarla o los mismos usuarios requieren información de una base de datos.

**Introducción:** MySQL y PostgreSQL son dos gestores de bases de datos que utilizan el lenguaje SQL y son muy parecidos en sus sentencias para la administración de BD. Ambos pueden ser obtenidos de forma gratuita por el tipo de licencia que manejan y son ampliamente utilizados por diversas empresas.

### Contenido

“MySQL es un sistema de gestión de bases de datos relacional, bajo licencia GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creado por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, 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, y la posibilidad de integrar este gestor en un software propietario.”<sup>37</sup>

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a 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. Sus características son:

- ☞ Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- ☞ Soporta gran cantidad de tipos de datos para las columnas.
- ☞ 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.
- ☞ Rapidez

---

<sup>37</sup> [http://www.netpecos.org/docs/mysql\\_postgres/index.html](http://www.netpecos.org/docs/mysql_postgres/index.html)

“PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. PostgreSQL fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales.

PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.”<sup>38</sup> Sus características principales son:

- ☞ Implementación del estándar SQL92/SQL99.
- ☞ Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. También permite la creación de tipos propios.
- ☞ Incorpora una estructura de datos array.
- ☞ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- ☞ Permite la declaración de funciones propias, así como la definición de disparadores.
- ☞ Soporta el uso de índices, reglas y vistas.
- ☞ Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- ☞ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

La arquitectura cliente-servidor de las bases de datos es:



Fig.1 Arquitectura cliente-servidor

## Instalación

Para instalar MySQL se puede acceder a la dirección: <http://www.mysql.com> y descargar la versión mas actual. Se debe tener instalado el servidor Apache, también PHP con el modulo para Mysql:

<sup>38</sup> [http://www.netpecos.org/docs/mysql\\_postgres/index.html](http://www.netpecos.org/docs/mysql_postgres/index.html)



```

❧ cd php-4.2.3/
❧ ./configure --prefix=/usr/local/php4
  --with-apxs2=/usr/local/apache2/bin/apxs --with-mysql
❧ make
❧ make install

```

Se realiza una prueba para ejecutar código PHP, si esto resulta lo siguiente es instalar MySQL:

Primero se descomprime el archivo que se haya descargado, se accede a la carpeta que se crea y se indica donde se debe instalar. Se utilizan los siguientes comandos:

```

❧ tar -zxvf mysql-4.1.7.tar.gz
❧ cd mysql-4.1.7
❧ ./configure --prefix=/usr/local/mysql4
❧ make
❧ makeinstall
❧ cp support-files/my-medium.cnf /etc/my.cnf

```

Después de instalarse se debe crear un grupo y un usuario llamados mysql, y se ingresa al directorio donde esta el archivo de instalación:

```

Groupadd mysql
Useradd -g mysql mysql
cd/usr/local/mysql4

```

Se deben crear las tablas necesarias para que MySQL funcione correctamente y se cambian algunos permisos:

```

/usr/local/mysql4/bin/mysql_install_db--user=mysql
chown-R root /usr/local/mysql4
chown-R mysqlvar /usr/local/mysql4
chgrp-R mysql

```

Para iniciar el servidor:

```

/usr/local/mysql4/bin/mysqld_safe --user=mysql &
Para acceder al interprete de comandos en la dirección /usr/local/mysql4/bin/ se ejecuta
./mysql

```

y para darlo de baja:

```

/usr/local/mysql4/bin/mysqladminshutdown

```

Para instalar PostgreSQL se debe tener instalado PHP con el módulo necesario:

```

❧ cd php-4.2.3/
❧ ./configure --prefix=/usr/local/php4
  --with-apxs2=/usr/local/apache2/bin/apxs --with-mysql --with-pgsql
❧ make
❧ make install

```

Se realiza una prueba para ejecutar código PHP, si esto resulta lo siguiente es instalar PostgreSQL que puede ser descargado de forma gratuita de la siguiente dirección: <http://www.postgresql.org/> y los pasos para instalarlo son los siguientes:

```

tar -zxvf postgresql-7.4.6.tar
cd postgresql-7.4.6
./configure --prefix=/usr/local/pgsql
gmake
gmake install
adduser postgres

```

Se debe crear la base de datos de instalación:

```

mkdir /usr/local/pgsql/data
chown postgres /usr/local/pgsql/data
su - postgres
/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data

```

Para arrancar y detener el servidor:

```

/usr/local/pgsql/bin/postmaster -D /usr/local/pgsql/data &
/usr/local/pgsql/bin/pgctl -D /usr/local/pgsql/data stop

```

## Sentencias

“El intérprete de comandos de MySQL es comúnmente usado para crear bases de datos y tablas en aplicaciones de bases de datos Web y probar consultas. Las siguientes sentencias pueden ser introducidas directamente en el intérprete de comandos y ejecutarse. Una vez que el servidor MySQL este corriendo el intérprete de comandos puede ser usado.”<sup>39</sup>En el intérprete de comando de PostgreSQL también se ejecutan estas sentencias:

**CREATE DATABASE Nombre\_DB**, permite crear una base de datos.

Para acceder a la base de datos:

```

MySQL: USE Nombre_DB
PostgreSQL: psql Nombre_DB

```

---

<sup>39</sup> Williams E. Hugo and Lane David, “Web Database Applications with PHP and MySQL”, USA 2004, O’REILLY, p.p. 140

**DROP DATABASE Nombre\_DB** permite eliminar una base de datos

**SHOW DATABASES** permite visualizar las bases de datos que se han creado.

Todas las instrucciones que se den en el interprete de comandos deben terminar con ; para que se ejecuten. Después de seleccionar una base de datos es posible crear las tablas que esta necesita.

Para crear una tabla se utiliza la siguiente sintaxis:

```
CREATE TABLE Nombre_Tabla ( campo tipo [NOT NULL | NULL] [DEFAULT valor] [AUTO_INCREMENT], PRIMARY KEY (campo), INDEX (campo), UNIQUE (campo), FOREIGN KEY (campo), REFERENCES tabla_padre (campo) [[ON DELETE CASCADE | ON UPDATE CASCADE] | SET NULL | NO ACTION | RESTRICT] ) TYPE= TIPO_DE_TABLA;
```

“Los identificadores o campos de las columnas pueden empezar por cualquier letra o número, pero no pueden estar constituidos sólo por números; además, pueden tener una longitud máxima de 54 caracteres.”<sup>40</sup> Algunos de los distintos tipos de datos que se manejan son:

- CHAR(L): Se usa para almacenar cadenas de L caracteres, puede tener un tamaño máximo de 255 caracteres. Ejemplo: nombre CHAR (50)
- VARCHAR (L): Es igual que CHAR, excepto en que internamente sólo se almacena el tamaño real del dato. Ejemplo: nombre VARCHAR (50)
- INT [unsigned]: Especifica un número entero cuyos valores van desde -2,147,483,6448 hasta 2,147,483,6448. Si se especifica unsigned, los valores van desde 0 hasta 4,294,967,295. Ejemplo: edad INT
- FLOAT [(E,D)]: Especifica valores en coma flotante de precisión simple. Ya que hay una parte entera y una real se puede especificar el total de dígitos para ambas cantidades. Ejemplo: media\_aritmet FLOAT, temperatura FLOAT (2,1)
- DATE: Almacena valores de tipo fecha. Los valores que admite son cadenas de caracteres con distintos formatos: YYYY-MM-DD o YYMMDD. Los valores que puede tomar van desde 1000-01-01 hasta 9999-12-31
- TIME: Almacena valores de tipo hora. Admite los formatos HH:MM:SS o HHMMSS o HHMM
- TIMESTAMP: Almacena valores de tipo instante con el formato YYYYMMDDhhmmss

Al especificar un tipo de dato también es posible indicar algunas características como son:

- ❖ PRIMARY KEY: Dado que este campo es la clave primaria de la tabla el gestor de base de datos indexará automáticamente la tabla en función de esta columna. Se usa para diferenciar una fila de otra.

---

<sup>40</sup> Guiérrez Abraham & Bravo Ginés, Op. Cit. Capitulo 4, nota 23, p.p. 341

- ❖ **NULL:** Son campos cuyos valores serán NULL o aquéllos especificados por DEFAULT
- ❖ **NOT NULL:** Un campo que tenga esta característica no podrá nunca asignársele un valor NULL (nulo)
- ❖ **AUTOINCREMENT:** Solo se aplica a campos de tipo entero ya que su efecto es, al insertar un registro en la tabla y dar el valor NULL para dicho campo asignara al correspondiente campo del registro nuevo el valor más grande que haya en esa columna mas uno.
- ❖ **DEFAULT valor\_por\_defecto:** Al insertar una fila se asignará (a la columna que tenga esta característica ) valor\_por\_defecto cuando se especifique NULL como valor a asignar
- ❖ **UNIQUE:** No permite que hay otro valor duplicado en la columna

Un ejemplo de tabla es:

```
CREATE TABLE PRODUCTOS (id_producto integer PRIMARY KEY, nombre_producto text, precio_producto float NOT NULL, clave_producto UNIQUE) TYPE=MYISAM;
```

Esta sentencia indica que la tabla a crear tiene 4 campos: id\_producto, nombre\_producto, precio\_producto y clave\_producto. Con sus respectivos tipos y características: un id único e irrepetible al ser llave primaria, un nombre que es de tipo texto, un precio que puede tener decimales y no permite valores nulos y una clave que es única.

**DROP TABLE Nombre\_Tabla;** elimina una tabla

**DESCRIBE NombreTabla;** muestra la estructura de la tabla

**SHOW CREATE TABLE Nombre\_tabla;** muestra como fue creada la tabla

“**ALTER TABLE** permite modificar la estructura de una tabla existente. Por ejemplo, se pueden añadir o eliminar columnas, crear y destruir índices, cambiar el tipo de una columna existente o renombrar columnas o la propia tabla. También es posible modificar el comentario y el tipo de la tabla.”<sup>41</sup> No es muy recomendable alterar una tabla ya que se podrían tener problemas si se hace esto.

## Manipulación de datos

Teniendo creada la base de datos con sus tablas e índices, el siguiente paso es insertar datos en las tablas. La sentencia para inserción de datos es la siguiente:

```
INSERT INTO tabla [campo1, campo2,...campoN] VALUES (valor1, valor2,...valorn)
```

---

<sup>41</sup> [http://www.conclase.net/c/mysql/index.php?sen=ALTER\\_TABLE](http://www.conclase.net/c/mysql/index.php?sen=ALTER_TABLE)

“El número de valores insertados es el mismo número de atributos en la tabla (y un error es generado si el número de valores no coincide con el número de atributos). Si no se requiere proporcionar un dato para un atributo se puede incluir NULL en lugar del valor.”<sup>42</sup>

Si no se hace uso de la parte condicional en la que se indican los campos a insertar, la lista de valores deberá contemplar a todos los elementos de la fila en el mismo orden en que se declararon cuando se creó la tabla.

Existe una segunda forma de la sentencia INSERT en la que los nombres de las columnas se especifican expresamente junto a su valor.

**INSERT INTO tabla SET columna1=valor1, columna2=valor2, ...**

Si el valor es de tipo carácter se debe englobar entre comillas. Un ejemplo de INSERT es:

```
INSERT INTO productos VALUES (1, 'refresco', 6.50, 15973);  
INSERT INTO productos (id_producto, precio) VALUES (1, 6.50);
```

Para hacer un listado de los registros y realizar búsquedas de datos que cumplan con una condición dentro de una tabla en la base de datos se utiliza la siguiente sentencia:

**SELECT lista\_de\_campos FROM tabla WHERE condición [GROUP BY (campos) [HAVING condición]] [ORDER BY (campos) [ASC|DESC]]**

Aquí lista\_de\_campos indica las columnas que se desean obtener en la consulta, puede ser el nombre de un campo o una lista de ellos separados con comas. Si se desean obtener todas las columnas se utiliza el carácter \*.

En WHERE condición será una expresión de tipo nombre\_de\_campo='un\_valor', se utiliza cuando se requiere una fila cuyo campo de búsqueda tenga una coincidencia exacta con el valor indicado. Las condiciones de búsqueda se pueden hacer más complejas usando los operadores lógicos AND, OR y NOT. Las cláusulas GROUP BY y ORDER BY permiten agrupar u ordenar el resultado. Ejemplo:

```
SELECT id_producto, nombre_producto FROM PRODUCTOS;  
SELECT * FROM PRODUCTOS WHERE id_producto > 12;
```

Cuando ya se tienen varios registros almacenados y se requiere modificar alguno se utiliza la siguiente sentencia:

**UPDATE tabla SET campo1=valor1...[WHERE condición]**

La cláusula SET indica las columnas a modificar y los valores que deben tomar. La cláusula WHERE, si se da, especifica qué filas deben ser actualizadas. Si no se especifica, serán actualizadas todas ellas.

---

<sup>42</sup> Williams E. Hugo and Lane David, Op. Cit. Nota 39, p.p. 148

```
UPDATE PRODUCTOS SET precio_producto=7.50 WHERE precio_producto= 6.50;  
UPDATE PRODUCTOS SET precio_producto= precio_producto*1.15;
```

Por ultimo para borrar un registro se utiliza:

### **DELETE FROM tabla WHERE condición**

Si se usa una sentencia DELETE sin cláusula WHERE, todas las filas serán borradas

```
DELETE FROM PRODUCTOS;  
DELETE FROM PRODUCTOS WHERE precio_producto = 6.50;
```

## **Conexiones con PHP**

Para interactuar con una base de datos se debe establecer una conexión entre PHP y el servidor de base de datos. Para esto PHP ofrece las siguientes funciones:

`mysql_connect`: Intenta conectar a una base de datos y requiere los siguientes parámetros en orden:

```
mysql_connect (“servidor”,”nombre de usuario”,”password”);
```

Todos los argumentos son opcionales, y si no hay, se asumen los valores por defecto ('localhost', usuario propietario del proceso del servidor, password vacío).

Para PostgreSQL

```
pg_pconnect ("host", "port", "options", "tty", "dbname");
```

“Cada uno de los argumentos debe ser una cadena entrecomillada, incluyendo el número de puerto. Los parámetros options y tty son opcionales y pueden ser omitidos.”<sup>43</sup>

Ambas funciones devuelven un identificador de enlace positivo si tiene éxito, o falso si error.

**mysql\_pconnect** y **pg\_pconnect** Abren una conexión persistente, es decir que no se cierra cuando termina la ejecución del archivo PHP, utilizan los mismos parámetros que las anteriores.

**mysql\_close** y **pg\_close** “permiten cerrar una conexión que haya sido abierta previamente. Aunque no cerrara la conexión si ésta fue abierta utilizando la función **pconnect**”<sup>44</sup>

---

<sup>43</sup> <http://www.php.net/manual/es/function.pg-connect.php>

<sup>44</sup> Minera Francisco José, Op.Cit. capitulo 4, nota 25, p.p. 146

Ya que normalmente en un servidor hay almacenadas más de una base de datos se debe indicar cual de ellas se va utilizar para hacer las consultas. PHP proporciona para MySQL una función que es:

**mysql\_select\_db (“nombre\_bd”, [identificador]);**

El identificador es de conexión, es opcional y si se omite la función asume la última conexión abierta con el servidor. No hay instrucción en PostgreSQL ya que cuando se usa `pg_connect` se indica la base de datos.

Para ejecutar los comandos DML<sup>45</sup> como son SELECT, INSERT, UPDATE y DELETE se utiliza la siguiente sentencia:

**mysql\_query (“instrucción”, [identificador]);** o en PostgreSQL:  
**pg\_query (conexión, “instrucción”);**

La instrucción es la cadena que contiene la consulta SQL a realizar.

Para saber el número de registros que han sido afectados con INSERT, UPDATE o DELETE se utilizan las sentencias:

**mysql\_affected\_rows (identificador);**  
**pg\_affected\_rows (identificador);**

El identificador es de conexión.

Al realizar una consulta a una base de datos, se obtiene un conjunto de registros. PHP permite almacenarlos en forma de objetos o también de arrays con el objetivo de acceder a ellos utilizando índices. Una función utilizada es:

**mysql\_fetch\_array (id\_consulta, [tipo\_indice]);**

Recupera una fila de una consulta como un array, ésta consulta es echa al servidor (con `mysql_query`) y se denomina `id_consulta`.

La función `tipo_indice` trata a cada fila como si fuera una matriz. Permite definir el tipo de índice. Las opciones son: `MYSQL_ASSOC`, `MYSQL_NUM` y `MYSQL_BOTH`. Este argumento es opcional y si no se incluye se asume que es `MYSQL_ASSOC`

La función en PostgreSQL es: **pg\_fetch\_array** donde `tipo_indice` puede tomar cualquiera de los siguientes valores: `PGSQL_ASSOC`, `PGSQL_NUM`, y `PGSQL_BOTH`.

Si ocurre un error al realizar una sentencia SQL y se requiere el recuperar el último mensaje de error se utilizan las sentencias:

---

<sup>45</sup>Comandos que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos. Tratados en el capítulo 5

**mysql\_error (identificador);**  
**pg\_result\_error (identificador);**

El identificador es de conexión.

Estas son algunas de las funciones que se pueden utilizar en PHP para hacer conexiones a las bases de datos. Existen más funciones que pueden ser consultadas en:

<http://www.php.net/manual/es>

Aquí se encuentran las funciones tanto para PostgreSQL como para MySQL.

Un ejemplo de cómo se usan estas funciones se encuentra en el capítulo 9.

**Conclusión:** Aunque estos dos gestores de bases de datos son muy parecidos cada uno cuenta con sus características propias, MySQL es más rápido pero no soporta tanta carga como PostgreSQL, quien también maneja conceptos de programación orientada a objetos en tablas, pero en MySQL los comandos para gestionar la base de datos son más intuitivos. Al final cada uno tiene sus propias características y la ventaja de ambos es que son software libre que se adapta muy bien con Linux, Apache y PHP.



## Capítulo VII

### Seguridad

**Objetivo:** Conocer cuales son las técnicas y programas que nos permitirán dar mayor seguridad a un sistema. También conocer como monitorear el equipo para detectar anomalías.

**Justificación:** Es necesario tener los conocimientos básicos de seguridad ya que al tener un servidor de páginas y base de datos, es posible que exista gente mal intencionada que pretenda acceder a la información y modificarla.

**Introducción:** La seguridad es un mecanismo que da la garantía de un buen funcionamiento y que evita, en caso de fallo de otro mecanismo, que se produzca un accidente o daño.

### Contenido

Las primeras medidas de seguridad que se necesitan tener en cuenta son las de seguridad física del sistema. Hay que tomar en cuenta quienes tienen acceso físico a las máquinas y si realmente necesitan acceder a éstas.

El nivel de seguridad física que se necesita en el sistema depende de la situación concreta. Un usuario doméstico no necesita tanta protección física. En una oficina, en un banco o empresa es diferente. Es por esto que se deben tener políticas de seguridad que establezcan, reglas formales a seguir, que deben ser respetadas por las personas que tienen acceso al sistema informático.

Pueden ser reglas desde tener bajo llave el cuarto donde se encuentra el sistema informático y cambiar cada cierto tiempo la clave de acceso, hasta contratar gente de seguridad y sistemas de identificación de ADN. Esto depende de la evaluación y análisis que se haga en cada caso.



Fig.1 Pasos a seguir para aplicar seguridad

Se deben realizar análisis de riesgos, es decir identificar y evaluar el riesgo a sufrir un ataque y perder datos, tiempo y horas de trabajo, comparándolo con el costo que significaría prevenir un ataque. Esto lleva a establecer un nivel de seguridad.

La Ingeniería Social engloba a todas aquellas conductas útiles para conseguir información de las personas del entorno de un sistema informático. Son engaños que utilizan las personas, algunos son externos al propio sistema informático, por ejemplo: entrar en el edificio como periodistas, aprovecharse de la vanidad de la gente, fingir ser parte de la comunidad de usuarios, etc. todo esto para conseguir información importante.

Otros engaños son internos como aprovechar la confianza del usuario al abrir un correo que dice ser importante, y que contiene un programa capaz de introducirse en el sistema para después enviar información de este a otra persona.

## **Seguridad informática**

“La Seguridad informática consiste en técnicas desarrolladas para proteger los equipos informáticos individuales y conectados en una red frente a daños accidentales o intencionados. Estos daños incluyen el mal funcionamiento del hardware, la pérdida física de datos y el acceso a bases de datos por personas no autorizadas.”<sup>46</sup>

Actualmente los daños mas comunes que se producen al navegar en la red son ocasionados por los virus informáticos. Un virus es un programa, generalmente destructivo, que se introducen en el ordenador al acceder a una red informática o al leer un disco, y pueden provocar pérdida de la información almacenada en el disco duro o algún daño al sistema operativo. Para evitar este problema lo más recomendable es contar con un antivirus instalado en el sistema y actualizarlo constantemente.

Otro problema de seguridad es el acceso no autorizado a datos. En un sistema protegido es necesario que el usuario se identifique mediante una clave de acceso antes de realizar cualquier operación. Una clave o contraseña es una secuencia confidencial de caracteres. Una clave adecuada no debe ser una palabra conocida, debe contener letras, números y caracteres especiales, en general deben ser fáciles de recordar pero difíciles de adivinar.

Cada vez que uno se aleja de la máquina es aconsejable dejar bloqueada el sistema para que nadie acceda al trabajo que se estaba realizando.

Los sistemas Linux son sistemas multiusuario real, puede haber varios usuarios distintos trabajando a la vez cada uno desde su terminal. El sistema tiene la obligación de proteger a unos usuarios frente a otros y protegerse a sí mismo.

El usuario root del sistema es un usuario que cuenta con todos los privilegios sobre el sistema, es por esto que muchos ataques se centran en conseguir los privilegios de root. El administrador principal es quien puede crear grupos y usuarios además de asignarles los privilegios que él considere necesarios para desempeñar su trabajo.

---

<sup>46</sup> Seguridad informática," *Enciclopedia Microsoft® Encarta® 2000*. © 1993-1999 Microsoft Corporation. Reservados todos los derechos.

Esto es también una desventaja ya que el administrador principal puede borrar cualquier fichero e incluso destruir el propio sistema, mientras que un usuario normal sólo puede perjudicarse a sí mismo. Una norma que siempre se debería tener presente es usar la cuenta de root sólo para realizar tareas concretas, breves y el resto hacerlo como usuario normal.

La vulnerabilidad describe el nivel de protección del equipo frente a otra red, y la posibilidad potencial para alguien que pueda obtener acceso no autorizado.

Nessus es un escáner de vulnerabilidades. “Es un programa que en operación comienza escaneando los puertos con nmap (es un comando de Linux) o con su propio escaneador para buscar puertos abiertos y después intentar varios exploits<sup>47</sup> para atacarlo. Las pruebas de vulnerabilidad son escritas en NASL (Nessus Attack Scripting Language, Lenguaje de Scripting de Ataque Nessus por sus siglas en inglés), un lenguaje scripting optimizado para interacciones personalizadas en redes.

Opcionalmente, los resultados del escaneo pueden ser exportados en reportes en varios formatos, como texto plano, XML, HTML, etc. Los resultados también pueden ser guardados en una base de conocimiento para referencia en futuros escaneos de vulnerabilidades. Algunas de las pruebas de vulnerabilidades de Nessus pueden causar que los servicios o sistemas operativos se corrompan y caigan.”<sup>48</sup>

## Monitoreo

El sistema permite monitorear las actividades que se realizan en un determinado momento. Algunos de los comandos mas utilizados son los siguientes:

- **lsoft -i tcp:** Muestra todos los programas que abren una aplicación en red
- **lsoft -p No.\_de\_proceso:** Indica que usuario lo esta ejecutando
- **lsoft -u usuario:** Muestra las librerías y las acciones que root ejecuta en ese momento
- **lsoft -c comando:** Muestra los comandos que se están usando
- **lsmmod:** Muestra los comandos cargados en el kernel
- **netstat -ant:** Nos dice que computadora esta conectada al sistema
- **netstat --inet --listenin:** Muestra los servicios que están escuchando en red

“Osiris es un programa gratuito de la red y que monitorea la integridad del host y supervisa periódicamente unos o más hosts por si hay cambios. Mantiene registros detallados de cambios al sistema de ficheros, las listas del usuario y del grupo, los módulos residentes en el núcleo, y más.

Los hosts se exploran periódicamente y, si se desea, los registros se pueden mantener guardados para propósitos legales. Osiris mantiene al administrador informado de posibles

---

<sup>47</sup> Programa malicioso, o parte del programa, que trata de forzar alguna deficiencia o vulnerabilidad de otro programa

<sup>48</sup> <http://es.wikipedia.org/wiki/Nessus>

ataques y/o de programas troyanos. El propósito del programa es aislar los cambios que indican un robo o un sistema en riesgo.”<sup>49</sup>

## Criptología

Es la ciencia que estudia los aspectos y contenidos de información en condiciones de secreto. Se divide en criptografía y criptoanálisis. La Criptografía es el conjunto de técnicas o procedimientos que alteran los símbolos de información sin alterar el contenido, convirtiendo la información modificada en un conjunto de símbolos sin sentido para las partes que no disponen de las técnicas que se utilizaron para alterar los símbolos.

El criptoanálisis son las tecnologías y técnicas que permiten recuperar la información que ha sido previamente tratada por un procedimiento criptográfico, sin conocer la técnica utilizada para la criptografía.

El protocolo SSH es un programa para acceder a otra computadora a través de una red para ejecutar comandos a una máquina remota y mover archivos de una computadora a otra. Provee una fuente de autenticación y comunicaciones seguras por medio de encriptación sobre canales inseguros.

La criptografía asimétrica o sistema de cifrado de clave pública es un método criptográfico que usa un par de claves o llaves para el envío de mensajes. Con las claves públicas no es necesario que el remitente y el destinatario se pongan de acuerdo en la clave a emplear. Todo lo que se requiere es que, antes de iniciar la comunicación secreta, el remitente consiga una copia de la clave pública del destinatario

“**Pretty Good Privacy** o **PGP** (privacidad bastante buena) es un programa desarrollado por Phil Zimmerman y su finalidad es proteger la información distribuida a través de Internet mediante el uso de criptografía de clave o llave publica, así como facilitar la autenticación de documentos gracias a firmas digitales.”<sup>50</sup>

Con PGP se deben generar un par de llaves una pública y una privada, las dos claves pertenecen a la persona que envía el mensaje. La clave pública se puede entregar a cualquier persona. La clave privada es del propietario y debe guardarla de modo que nadie tenga acceso a ella. La pública se utiliza para cifrar un archivo, la privada para descifrarlo. El remitente usa la clave pública del destinatario para cifrar el mensaje, y una vez cifrado, sólo la clave privada del destinatario podrá descifrar este mensaje.

Es muy posible que PGP ya venga instalado por defecto en el sistema. A diferencia de protocolos de seguridad como SSL<sup>51</sup> que sólo protege los datos en tránsito (es decir, mientras se transmiten a través de la red), PGP también puede utilizarse para proteger datos almacenados en discos, copias de seguridad, etcétera.

---

<sup>49</sup> <http://osdir.com/Downloads+index-req-viewdownloadetails-lid-829-ttitle-osiris.phtml>

<sup>50</sup> <http://es.wikipedia.org/wiki/PGP>

<sup>51</sup> Protocolo diseñado por Netscape para dotar de seguridad a las sesiones de navegación a través de Internet codificando la información

Para crear la llave se escribe el siguiente comando en la consola del sistema Linux:

```
gpg --gen-key
```

Se introducen algunos datos como el tipo de llave a crear, la fecha de expiración, la frase contraseña para acceder a los datos y otros datos.

Para encriptar un archivo la sentencia es:

```
gpg -e archivo_a_encriptar y enter
```

Se escribe la contraseña de llave pública y doble enter.

Para recuperar el archivo:

```
gpg -d archivo.gpg > archivo_recuperado.c y enter
```

y se escribe la contraseña de la llave privada.

Para exportar la llave pública:

```
gpg --export --armor > llave_publica.asc y enter
```

Para enviar:

scp llave\_publica.asc dirección. Por ejemplo:

```
scp rosi.asc root@132.248.75.175:/root
```

Para importar una llave pública:

```
gpg --import llave_publica.asc ejemplo:
```

```
Gpg --import christiann.asc
```

Para encriptar un archivo con otra llave pública:

```
gpg --se -r christin archivo
```

Cuando se tienen varias llaves públicas, que enviaron otros usuarios, se puede tener un “llavero” con el cual es posible enviar información a los usuarios con seguridad.

## **Contraseñas**

En Linux las contraseñas se guardan en el fichero `/etc/passwd` y este fichero es accedido por algunos programas que necesitan información del usuario. No se pueden guardar las

contraseñas sin antes cifrarlas con algún método complicado para que no se pueda acceder a ellas fácilmente. Las contraseñas en un sistema se cifran con un método de cifrado indescifrable.

“Aunque suena un poco absurdo hay que recordar que las contraseñas se guardan para comprobar si la contraseña que el usuario ha proporcionado es correcta. No se pueden descifrar las contraseñas guardadas para comprobar si es correcto lo que el usuario ha introducido por teclado, el truco está en cifrar lo que el usuario ha introducido y comparar cifrado con cifrado.

Al utilizar estos algoritmos de cifrado indescifrables o de un solo sentido ("only-one-way"), no se puede descifrar la contraseña, pero se puede aplicar ese mismo algoritmo a la clave que el usuario proporcionó y comparar el resultado con la contraseña cifrada. Con estos algoritmos, "sólo se puede ir hacia la derecha", es decir, sólo se cifra, ésta es la mejor manera de guardar contraseñas.”<sup>52</sup>

Otro método para ingresar a un sistema es aplicar un ataque de diccionario. No es posible "ir hacia la izquierda", es decir, descifrar la contraseña, pero se puede ir tantas veces como se desee "hacia la derecha" (cifrar palabras), hasta dar con el resultado. Así, lo más habitual es tomar todas las palabras de un diccionario, y aplicarles un algoritmo de cifrado, hasta dar con la palabra correcta.

Estas listas de palabras suelen hacerse tomando todas las palabras del inglés, todas las del castellano, todas las del italiano, nombres de ciudades, de equipos deportivos, de famosos, etc. y al final se obtiene un listado gigantesco de posibles contraseñas con las que probar este ataque de diccionario. La ingeniería social también puede ayudar a esto ya que se puede averiguar cual es la mayor afición del usuario, que podría ser una película, deporte, libro, etc., e incluir las palabras relacionadas a esto en el diccionario.

Esto no se hace de forma manual, existen muchos programas para crackear contraseñas, aunque el más famoso de todos es "John the Ripper", que es un programa pensado para el mundo UNIX en general, así que tiene más sentido utilizarlo desde Linux.

No todas las contraseñas son palabras del diccionario, algunas incluso no son ni pronunciables y contienen números y letras mezclados. En estos casos se usa un enfoque incremental para crackear las contraseñas, probando absolutamente todas las combinaciones: primero todas las de una letra, luego las de dos letras, luego las de tres, etc. A esto se le llama ataques incrementales por fuerza bruta.

De esta manera se asegura que al final la contraseña será adivinada ya que se prueban absolutamente todas las posibilidades, pero puede darse el caso de que la contraseña sea lo suficientemente grande y compleja como para que el tiempo que se necesite para descifrarla sea de varios años. Normalmente en todo ese tiempo un usuario habrá cambiado varias veces de contraseña, así que el trabajo no valdría la pena.

---

<sup>52</sup> [http://metabolik.hacklabs.org/alephandria/txt/txipi\\_seguridad.html](http://metabolik.hacklabs.org/alephandria/txt/txipi_seguridad.html)

## Huella digital

La salida producida por una función hash<sup>53</sup> aplicada a un documento es conocida con el nombre de huella digital. Cualquier cambio en el documento produce una huella diferente. También es conocida como compendio de mensaje

MD5 es una función hash que toma como entrada un mensaje de longitud arbitraria y regresa como salida una huella digital de 128 bits. Es imposible tener dos mensajes con la misma huella digital.

En algunos sitios de Internet, al descargar un archivo se muestra en la página la huella digital creada por la función MD5, al tener el archivo en el equipo es posible aplicar esta función al archivo para verificar que este fue descargado correctamente. El comando en Linux es:

md5sum archivo y enter Ejemplo: *md5sum httpd-2.0.54.tar.gz*

El comando entrega una secuencia de caracteres que es la huella digital, la cual se puede comparar con la que se muestra en la página de donde se descargo.

## Esteganografía

“La esteganografía es un caso particular dentro de la criptografía. La palabra proviene del griego y significa "escritura encubierta". Mediante la esteganografía se consigue ocultar un mensaje dentro de otro. Hoy en día se utiliza mucho la esteganografía para ocultar documentos electrónicos dentro de imágenes.”<sup>54</sup>

Cuando se encripta un mensaje se pueden dar sospechas de que algún tipo de información importante está contenida dentro del mensaje, pero la esteganografía nunca dará sospechas ya que se pueden usar imágenes inofensivas para enviar un documento.

Un ejemplo:

Imagen Original:



Imagen con información:



Texto oculto:



Fig.2 Ejemplo de esteganografía

<sup>53</sup> Una función hash es una función pública que mapea un archivo de cualquier tamaño en un valor de tamaño fijo, el cual sirve de autenticador

<sup>54</sup> [http://metabolik.hacklabs.org/alephandria/txt/txipi\\_seguridad.html](http://metabolik.hacklabs.org/alephandria/txt/txipi_seguridad.html)

“OutGuess es una herramienta esteganográfica universal que permite la inserción de información oculta en los bits redundantes de fuentes de datos.”<sup>55</sup> Dentro de outguess existe una herramienta de detección de esteganografía que busca patrones esteganográficos dentro de imágenes y demás ficheros. Normalmente detecta versiones anteriores de outguess y otros programas similares.

Es posible bajar el programa de Outguess de la red e instalarlo en el equipo:

Primero se descomprime el archivo y se instala, para esto se siguen los siguientes pasos:

- `tar -zxvf outguess-0.2.tar.gz`
- `cd outguess`
- `./configure`
- `make`
- `make install`

Para ejecutarlo y colocar texto sobre imagen:

`Outguess -k "llave" -d archivo_oculto image_salida.jpg` (solo se pueden utilizar imagenes jpg). Ejemplo:

`Outguess -k "curso_1" -d poema.txt logo.jpg`

Para recuperar el texto:

`Outguess -k "llave" -r imagen_con_texto texto_de_salida`

`Outguess -k "curso_1" -r logo.jpg salida.c`

Es posible ocultar una imagen sobre texto y la diferencia de tamaño entre la imagen original y la que oculta información es pequeña, pero no es posible comprimir la imagen para ocultarla en un texto ya que el tamaño del archivo de texto aumentaría significativamente.

## Respaldos

Un respaldo es una copia de los datos almacenados en un sistema, escrita en un disco externo u otro medio de almacenamiento duradero. Es recomendable tener respaldos de la información y hacerlos con frecuencia ya que si ocurre un problema y se pierde información las consecuencias van desde perder un documento hasta una base de datos. El administrador de un sitio puede tener la responsabilidad de respaldar docenas o incluso cientos de máquinas.

También es recomendable llevar una bitácora, que es el registro de los eventos que van ocurriendo en el sistema, por ejemplo se almacenan los errores que ocurrieron, quien y

---

<sup>55</sup> <http://www.outguess.org/>



cuando accedió, actualizaciones, etc., y se puede almacenar en uno o en diferentes archivos.

Se deben tener planes de contingencia que establezcan una política de recuperación de datos ante un desastre. Si no se prevén estas situaciones es posible sufrir una pérdida irremediable y más costosa que la implementación de un plan.

Por último lo más importante en la seguridad es prevenir un ataque o daño con políticas de seguridad, autenticación de usuarios, control de acceso, privacidad de los datos (cifrado). También es necesario tener la característica de no repudiación en un sistema informático, es decir poder comprobar las acciones realizadas por usuarios y equipos. Es necesario garantizar que alguien que haya realizado una acción no pueda negar que la hizo.

Es necesario detectar si se puede dar una situación de riesgo esto es monitorear el sistema y buscar vulnerabilidades. Y si se llega a presentar un ataque o daño hacer una recuperación de los datos y restaurarlos con los respaldos.

**Conclusiones:** Es importante conocer los aspectos básicos de la seguridad en cómputo. Aunque los temas tratados fueron tratados de manera básica dan una idea de las herramientas que se pueden utilizar. También es necesario analizar que herramientas conviene utilizar dependiendo de las necesidades que se tengan.

## Capítulo VIII

### Edición de publicaciones

**Objetivo:** Conocer las herramientas más importantes del programa Photoshop, para la edición y creación de imágenes. También aprender como almacenar las imágenes para su uso en medios electrónicos o paginas Web.

**Justificación:** Las Paginas Web hoy en día necesitan una interfaz gráfica que sea amigable y cuente con imágenes y diseños atractivos para los usuarios, también se requiere que las imágenes se puedan descargar de manera rápida.

**Introducción:** Los programas de retoque fotográfico son programas informáticos con los que se pueden modificar imágenes fotográficas digitalizadas. Un retoque fotográfico puede ser la supresión o la adición de elementos, el empleo de efectos como son filtros y herramientas como pinceles brochas y aerógrafos. También permite restaurar fotografías, colorearlas y hacer montajes.

### Contenido

Photoshop es una aplicación informática de edición y retoque de imágenes digitales elaborada por la compañía de software Adobe, se usa en multitud de disciplinas del campo del diseño y fotografía, como diseño Web, composición de imágenes digitales, y básicamente en cualquier actividad que requiera el tratamiento de imágenes digitales.

Photoshop soporta muchos tipos de archivos de imágenes, como BMP, JPG, PNG, GIF y otros, también tiene ciertas extensiones propias como son:

PSD (Photoshop Document): Es un formato que guarda una imagen como un grupo de capas, métodos de fusión, colores, textos, máscaras, canales de color, canales alfa, trazados, formas, configuración de tonos, entre otras.

PSB: Es una nueva versión del formato PSD, diseñado especialmente para archivos mayores a 2 GB.

### Formatos de imagen digital

Los mapas de bits, que son el sistema estándar utilizado por Windows, utilizan un excesivo tamaño a la hora de alojarse en un disco, esto reduce la posibilidad de almacenar varias imágenes en un sistema de almacenamiento portátil, como un disquete. Las imágenes que se encuentran en las páginas Web no pueden ser de gran tamaño, debido a que a la hora de descargar la página, tardaría mucho al mostrar las imágenes. Una solución a este problema es la compresión de datos.

“Mediante la compresión se consigue que las imágenes ocupen menos espacio en el disco basándose en datos contiguos repetidos. Hay sistemas de compresión de archivos que mediante funciones matemáticas, analizan la información contenida en el archivo buscando

datos repetidos, consiguiendo una compresión mayor y por lo tanto un ahorro óptimo de espacio.”<sup>56</sup>

JPG y GIF son los formatos más comunes, siendo el primero de mayor calidad, pero ocupa más espacio en disco, mientras que el formato GIF es de menor calidad, sólo permite 256 colores como máximo, pero muchísimo más ligero, además de permitir transparencias y animaciones.

## Creación de un documento

Para crear una nueva imagen en photoshop versión 7.0 se da clic en el menú desplegable Archivo y Nuevo, aparecerá un cuadro que solicita algunos datos como son:

- ✓ Nombre: Es el nombre que se le asigna al documento.
- ✓ Anchura y Altura: Aquí se define el tamaño del “lienzo” y se puede especificar la unidad de medida, pueden ser centímetros, pixeles, pulgadas, etc. También se puede elegir un tamaño estándar en el cuadro predefinir y elegir uno de los formatos disponibles.
- ✓ Resolución: Indica el nivel de detalle con el que se representará la imagen. La unidad de medida más utilizada es pixeles/pulgada. La resolución recomendada para Internet es de 72 hasta 150, para medios electrónicos de 300 a 600 y de 600 a 2400 para impresiones. Mientras más resolución tenga un documento de mayor tamaño será
- ✓ Modo de color: Puede elegirse el modo RGB, CMYK, escala de grises y otros. Con estos sistemas se pueden representar casi todos los colores del espectro visual aunque con pequeñas diferencias.
  - ✎ El sistema RGB utiliza los colores Rojo Verde y Azul (colores primarios para la luz) para conseguir casi cualquier tonalidad. Son los llamados colores aditivos ya que al mezclarse suman componentes aclarándose, al mezclarse todos se obtiene el color blanco. Este sistema es utilizado para representar imágenes en pantalla. Se utiliza para medios electrónicos y en Internet.
  - ✎ El sistema CMYK utiliza los colores Cian, Magenta, Amarillo y Negro, estos son llamados colores sustractivos ya que la mezcla de los tres primeros, en condiciones óptimas, genera el color negro, pero el resultado generalmente es un marrón oscuro, por esto se agrega el color negro. Es el sistema utilizado para hacer impresiones.
- ✓ Contenido: Permite establecer el color de fondo general que tendrá la imagen, puede ser un fondo blanco, el que se tenga seleccionado en ese momento o ningún color, esta opción es muy útil cuando es necesario generar mezclas de imágenes.

---

<sup>56</sup> Pascual Francisco, Photoshop 7, Madrid 2003, Alfaomega Ra-Ma, p.p. 34

También es posible extraer una imagen del disco duro para realizar algún retoque fotográfico. Para esto es posible seleccionar la opción Abrir del menú Archivo y buscar la imagen que se requiere modificar.

Se puede indicar el formato con el que esta almacenada la imagen

## Barra de herramientas

Para la manipulación de imágenes Photoshop cuenta con una barra de herramientas(Fig.1):

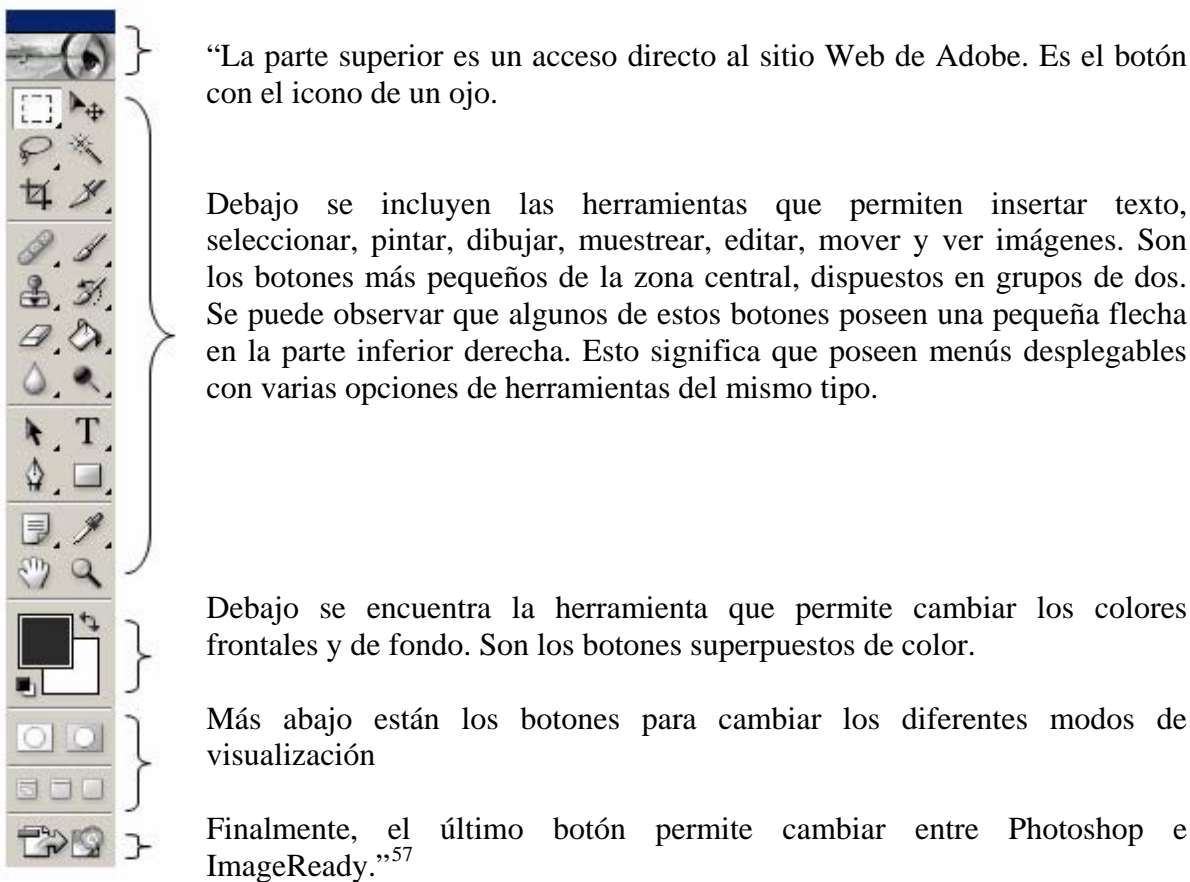
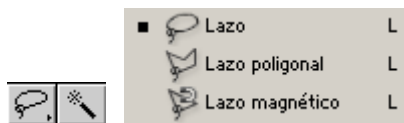


Fig.1

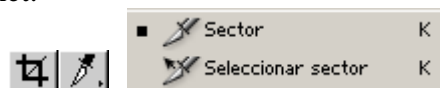
Las herramientas que forman parte de la zona central de la barra y que permiten tratar digitalmente las imágenes son:

Las herramientas Lazo realizan selecciones a mano alzada, lazo poligonal: imágenes con bordes rectos y lazo magnético selecciona objetos de una imagen ajustándose automáticamente a su contorno. La herramienta Varita mágica selecciona áreas de un color similar.



<sup>57</sup> [http://todo-photoshop.com/tutorialphotoshop/basicos/cuadro\\_herramientas\\_photoshop.html](http://todo-photoshop.com/tutorialphotoshop/basicos/cuadro_herramientas_photoshop.html)

“La herramienta Recortar permite recorta una porción de la imagen. La herramienta Sector crea sectores para asociarles hipervínculos. Es una función práctica si se va a crear imágenes destinadas a Internet.”<sup>58</sup>



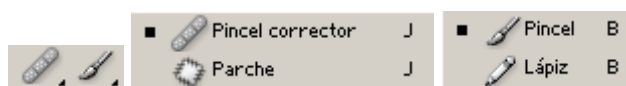
Cuando se selecciona una herramienta aparece en la parte superior de la pantalla una barra de opciones de la herramienta, en el caso de la herramienta recortar es la siguiente:



Fig.2 Barra de opciones herramienta recortar

La herramienta pincel corrector se utiliza para corregir imperfecciones y daños de una imagen mediante duplicados de otras zonas de la imagen. La herramienta parche es otra herramienta de corrección, selecciona una zona no dañada de la imagen y arrastra esa área hasta la zona dañada.

La herramienta pincel permite dibujar como se haría con un pincel, genera un trazo según sea arrastrado por la imagen. La herramienta lápiz reutiliza para trazar líneas a mano alzada como se haría con un lápiz sobre el papel.



El tampón de clonar permite generar duplicados de partes de la imagen, con ella es posible desaparecer objetos o clonarlos. La herramienta tampón de motivo se maneja de manera similar al tampón de clonado pero en lugar de clonar zonas de la imagen dibuja en la imagen con el motivo que se elija, este aparece en la barra de opciones de herramienta.

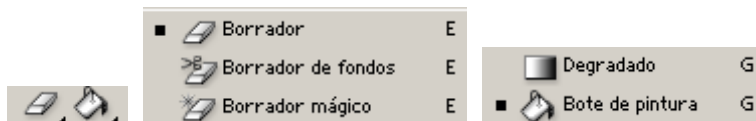
La herramienta pincel de historia permite borrar las pinceladas trazadas en la imagen con el pincel o el lápiz dejando intacta la imagen original. El pincel histórico se aplica como el pincel, pero su trazo consiste en desfigurar ligeramente la imagen por donde se pasa el pincel, depende del tipo y tamaño del pincel que se elija.



La herramienta Borrador se utiliza como una goma al pasarlo por una superficie que contenga puntos de color los elimina. La herramienta Borrador de fondos al arrastrar, borra áreas y las transforma en transparentes. La herramienta Borrador al hacer un clic en un punto de la imagen borra todos los píxeles del mismo color que se encuentren junto aquél en el que se hizo clic.

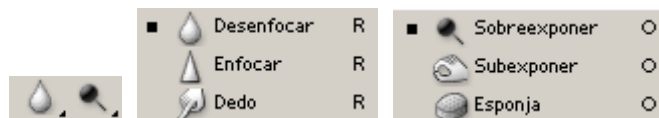
<sup>58</sup> Pascual Francisco, Op. Cit Nota 56, p.p. 93

Las herramientas Degradado permite rellenar una superficie con una escala de colores, permite crear fusiones lineales, radiales, angulares, reflejadas y de diamantes entre colores. La herramienta Bote de pintura rellena de manera similar las áreas coloreadas con el color frontal



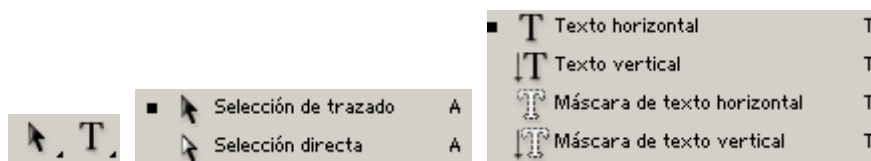
La herramienta desenfocar como su nombre indica, se usa para desenfocar partes de la imagen. La herramienta enfocar genera el efecto contrario a la herramienta anterior, es decir permite enfocar la imagen. La herramienta dedo permite generar un efecto similar e deslizar un dedo sobre pintura fresca.

La herramienta sobreexponer permite aclarar zonas de la imagen basándose en el mismo método que se utiliza en fotografía: sobreexponiendo partes de la imagen a la luz. La herramienta subexponer oscurece en lugar de aclarar. La herramienta esponja permite aumentar o disminuir la saturación del color en una imagen.



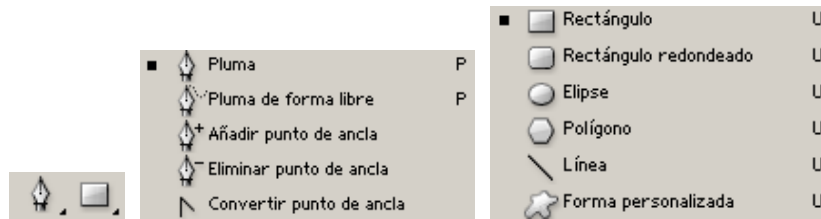
La herramienta selección de trazado se emplea para seleccionar trazados dibujados con la herramienta pluma. La herramienta selección directa también se utiliza en trazados hechos con pluma, permite mover los puntos de anclaje o dirección

La herramienta texto permite escribir texto sobre la imagen, con la barra de opciones de esta herramienta es posible seleccionar la fuente, el tamaño y su estilo. También se cuenta con la herramienta que permite colocar el texto en forma vertical.



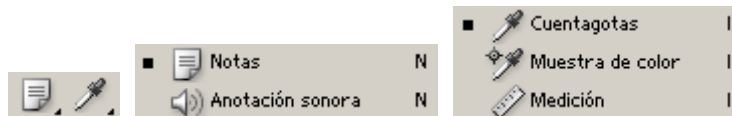
La herramienta pluma se emplea para trazar líneas rectas, líneas curvas o figuras que contengan mezclas de ambas. Cuando se dibuja una línea aparecen dos elementos fundamentales de esta, el punto de anclaje que de mantiene fijo, varios puntos de dirección que permiten acentuar o suavizar las líneas curvas. Las herramientas de pluma que se despliegan permiten añadir puntos, eliminarlos o convertirlos en puntos de esquina o curva.

Las herramientas de dibujo permiten colocar las figuras que indican en una zona de la imagen. La herramienta forma personalizada permite dibujar figuras de varios tipos desde la lista desplegable forma de la barra de opciones.



La herramienta notas permite colocar anotaciones en la imagen que pueden servir como recordatorios. La herramienta de anotación sonora tiene la misma función, la diferencia es que son anotaciones de audio.

La herramienta cuenta gotas permite seleccionar el color que hay en un punto de la imagen, y utilizarlo. La herramienta muestra de color es similar al cuenta gotas pero con esta herramienta se obtienen los valores del color, recopila información de hasta cuatro puntos distintos de la imagen. La herramienta medición permite comprobar la distancia exacta entre dos puntos de la imagen.



Por último la herramienta mano permite moverse por la imagen que es más grande que la ventana. Y la herramienta zoom se utiliza para aumentar o disminuir la vista de la imagen.



En el panel de herramientas se encuentran también los selectores de color, en Photoshop se trabaja siempre con dos colores activos, que pueden ser totalmente configurados por el usuario. El color situado al frente será el color Frontal, la mayoría de las herramientas lo tomarán como color de trabajo. El que se encuentra detrás es el color de Fondo, es posible establecer este color como el color del fondo del documento, de esta forma se pueden utilizar herramientas que pintan con el color de fondo y así imitar de alguna forma el borrado de píxeles.



Para seleccionar un color se hace clic sobre uno de los recuadros y se abrirá el cuadro de diálogo Selector de color. Una vez aquí es posible elegir un color de entre toda la paleta que Photoshop ofrece.

## Paletas

Las paletas de Photoshop tienen como objetivo facilitar el trabajo al usuario ofreciendo funciones que se utilizan con mucha frecuencia. En el menú ventana es posible seleccionar las paletas que se necesitan tener activas.



La paleta Info muestra información acerca de la posición de puntero del ratón, los tonos RGB y CMYK del píxel en el que se encuentra en ese momento y las coordenadas en las que empieza y termina el marco de la parte seleccionada actual.

La paleta Navegador es un sistema de ampliación y reducción de la vista de la imagen.



Con la paleta color es posible formar un color desplazando los controles a izquierda y derecha y es necesario encontrar la proporción exacta de rojo, azul y verde para formar el color que se necesita.

La paleta Muestras ofrece diferentes colores guardados en el sistema

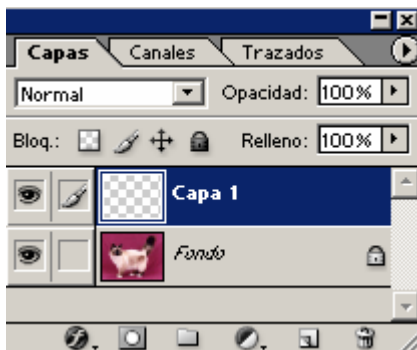
La paleta Estilos contiene diferentes estilos de relleno o texturas se pueden utilizar en las composiciones que se hagan.



Historia, muestra por orden cronológico las acciones realizadas sobre una imagen, donde las posiciones al pie de la lista son más recientes que las situadas al principio

Acciones ofrece la posibilidad de lanzar un conjunto de acciones mecanizadas predefinidas con anterioridad. En principio, el conjunto de acciones disponibles son las predeterminadas por Photoshop, aunque existe la

posibilidad de crear acciones propias



En la pestaña Capas es posible manejar las capas de la imagen (Este tema se vera mas adelante).

La paleta Canales consiste unicamente en una lista de los canales que comprende el sistema de color que se tenga activo para la imagen. Si la imagen esta formada mediante el sistema RGB, se podrán observar los canales del rojo, verde y azul.

La paleta de Trazados y la herramienta pluma permiten dibujar líneas de estilo vectorial. La paleta de trazados contiene algunos botones con funciones que permitirán


modificar los trazos generados por la herramienta.

## Capas

“Una capa es como una lámina de acetato transparente y el programa permite trabajar con cuantas capas sea necesario. Es como un conjunto de láminas de acetato; la que se encuentre en la parte superior tapaná a las otras, pero sólo en las zonas en las que tenga pintura, el resto permanecerá transparente.”<sup>59</sup>

<sup>59</sup> [http://www.aulacliic.es/photoshop/t\\_4\\_2.htm](http://www.aulacliic.es/photoshop/t_4_2.htm)



Si se desea crear una nueva capa en el menú Capa -> Nueva -> capa, o bien en la paleta de capas se da un clic en el icono . Esto añade una capa transparente a la paleta Capas, el programa nombra a las capas por defecto, es posible cambiar el nombre para tener un mejor control de ellas, dando un clic derecho sobre la capa y elegir propiedades de capa, en el cuadro que aparece se cambia el nombre y es posible darle un color.

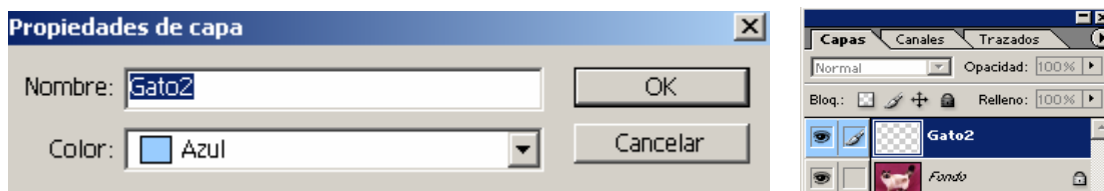


Fig.3 Ejemplo de capas

Gracias a las capas se podrá trabajar con imágenes estructuradas en capas independientes y aplicar una amplia gama de técnicas. Es decir que las operaciones que se realicen en una capa solo afectaran a la imagen que este en esa capa. Para seleccionar una capa solo se da un clic sobre ella en la paleta capas.

Las capas superiores tapan a las inferiores siempre y cuando la superior tenga pintura. En el caso contrario, si la capa superior tiene zonas transparentes, éstas dejarán ver las capas inferiores.

Desde la ventana de Capas puedes modificar la opacidad de la capa activa. El valor 0% significaría que la capa sería totalmente transparente y dejaría ver completamente el contenido de las capas inferiores.



En este caso a la capa de los ojos se le dio una opacidad de 50% y esta bajo la capa montañas, y la flor esta sobre la capa montañas, esto gracias a que las capas se pueden desplazar dando un clic sobre la capa y arrastrándola entre las capas hasta que ocupe el lugar que sea mas conveniente.

Fig.4 Imagen con varias capas

Existe la opción de “acoplar imagen” en la paleta Capas, esto permite que todas las capas se unan en una sola imagen.

## Filtros

“Se denomina genéricamente filtros a un amplio grupo de herramientas cuya función es la de transformar las imágenes siguiendo diferentes criterios. La potencia de estas herramientas y los resultados que se consiguen al aplicarlas acumulativamente sobre una imagen o parte de ella exceden cualquier previsión.”<sup>60</sup>

<sup>60</sup> Córdoba Enrique, González Carmen y Cordoba Carmen, Photoshop CS Curso Completo, España 2005, Alfaomega- RA-MA, p.p. 125

En el menú filtro hay una gran variedad de filtros que permiten modificar la con diferentes efectos. Se puede aplicar un filtro a la imagen que se abrió y con la cual se va a trabajar o se puede aplicar un filtro a cada capa que se tenga.

Al seleccionar un filtro aparece una pantalla que da opciones de cómo modificar los parámetros del filtro y la previsualización de la imagen, por ejemplo:

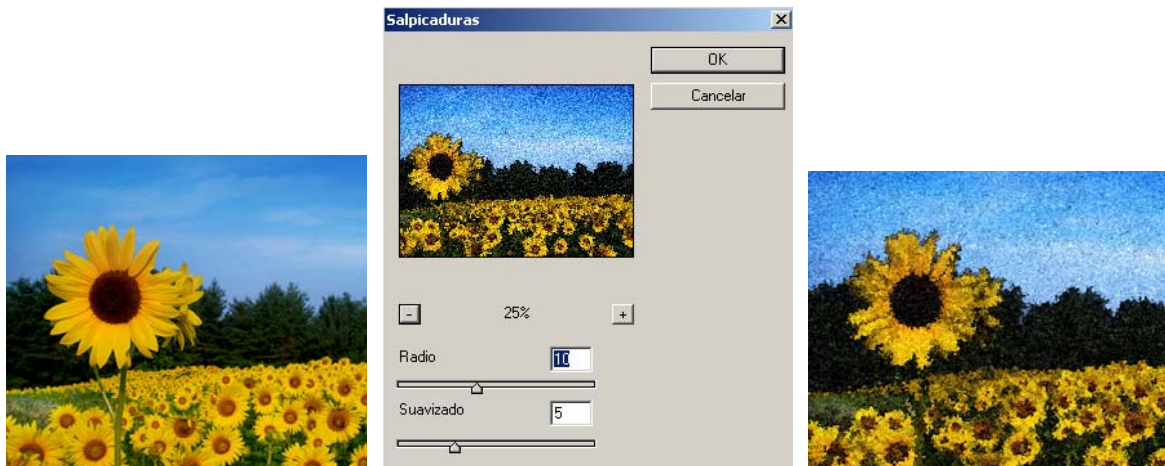


Fig. 5 Imagen original, cuadro de filtro salpicaduras e imagen final tratada con filtros

En este caso en la imagen original después de aplicar el efecto fresco, texturizar y salpicaduras parece haber sido pintada a mano. Se debe tener cuidado en que filtro se usara ya que un filtro mal usado puede arruinar la imagen.

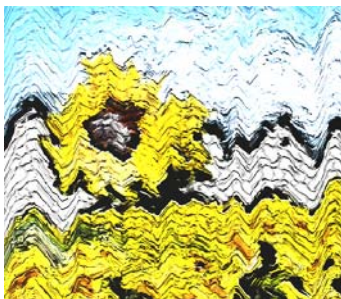


Fig.6 Imagen tratada con filtros

## Guardar para Web

En Photoshop, se dispone de una opción con la que se puede grabar en el disco duro las imágenes del mejor modo posible cuando su finalidad es colocarlas en una página Web. La imagen ocupara poco espacio en el disco al estar comprimida en formato GIF o JPEG, pero tendrá buena calidad.

Para almacenar la imagen de este modo, se recurre a la opción Guardar para Web del menú archivo. Una vez activada esta opción aparece un cuadro en pantalla, este contiene un grupo

de opciones con el nombre de ajustes, con estas opciones se optimiza la imagen para el Web.

Este cuadro también tiene la opción de ver la imagen en el explorador para saber como se vera la imagen en un caso real.

Otra herramienta que permite manejar opciones Web es Seleccionar Sector que permite seleccionar un sector al cual se le podrá asignar un hipervínculo.

Con la ayuda del botón opciones de sector de la barra de opciones se obtiene un cuadro de dialogo en que es posible establecer todos los datos relativos a dicho sector, como por ejemplo la pagina a la que se debe acceder cuando el usuario haga clic sobre ese sector. Si la pagina se abrirá en la misma ventana o abrirá una ventana nueva, etc.

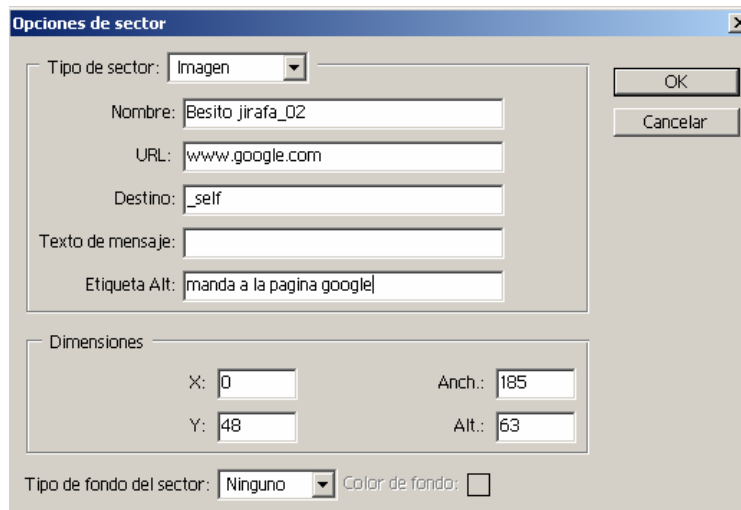


Fig.7 Cuadro para establecer datos de un sector

**Conclusiones:** Photoshop cuenta con muchas herramientas que nos permiten tratar una o varias imágenes para editarla como más nos convenga. Es posible reconstruir una imagen deteriorada, darle efectos más vistosos a las imágenes con los filtros, e inclusive utilizar una imagen para una pagina Web, teniendo los requerimientos de buena calidad y menor tamaño. Photoshop es una herramienta muy útil ya que, hoy en día la mayoría de las aplicaciones Web manejan imágenes.

## Capítulo IX

### Proyecto aplicativo

**Objetivo:** Aplicar los conceptos que se han tratado en los temas anteriores para desarrollar un portal de ventas en Internet según lo que se pide.

### Portal de venta y administración de discos

#### Planteamiento del problema

Una tienda de nombre Pixup que se dedica a la venta de discos, requiere tener una base de datos y un portal en Internet para que los usuarios puedan realizar compras vía Web. Para poder realizar la compra los usuarios deberán estar registrados en la base de datos de la tienda.

Se requiere que los empleados y administradores de la tienda puedan acceder al portal para hacer modificaciones a los catálogos con los que se cuenta. Es decir se podrán insertar, borrar o modificar los registros de los discos.

Para esto se contará con una tabla en la base de datos que contenga las claves de los administradores. El administrador general es el único que puede agregar o borrar usuarios de la tabla de administradores.

Todos los empleados pueden consultar la lista de ventas que se lleve, la cual deberá estar ordenada por fechas.

#### Requerimientos específicos:

Se deberán de generar los siguientes scripts, y los que sean necesarios, para interaccionar a través de PHP con MYSQL:

genero.php  
autor.php  
disquera.php  
administradisco.php

Estos scripts deben de listar el contenido de su correspondiente tabla además de poder insertar un nuevo registro (SELECT e INSERT), como por ejemplo para autor.php:

#### CATALOGO DE ARTISTAS

Artista	Dirección	Disquera
Shakira	Square Garden 25	Sony Music
Los Tigres del Norte	Juan Garza 25	Fonovisa

## INSERTAR UN NUEVO ARTISTA

Nombre:

Dirección:

Disquera:

## Administración de discos

Para el scrip de discos se debe permitir insertar un nuevo disco, actualizar, eliminar y consultar los discos existentes a través de un menú donde se selecciona la opción de lo que se quiere realizar, de la siguiente manera:

Para la administración de discos se requieren las siguientes opciones:

Acción	Scrip
<a href="#">Insertar un disco</a>	insertadisco.php
<a href="#">Eliminar un disco</a>	eliminadisco.php
<a href="#">Actualizar un disco</a>	actualizadisco.php
<a href="#">Listado de discos</a>	consultadisco.php

Todas las páginas anteriores deben de estar restringidas para usuarios autorizados (ya que se utilizan para cuestiones de administración de la tienda), la página que permite el acceso a todas estas páginas (donde se debe generar la sesión) se debe llamar administra.php que lo que debe de hacer es validar al usuario contra la tabla ADMIN(de la base de datos que contiene los usuarios autorizados), los usuarios autorizados que deben de estar en la tabla son:

<b>Login:</b> pajaro	<b>Login:</b> castor	<b>Login:</b> master
<b>Password:</b> loco	<b>Password:</b> australiano	<b>Password:</b> master

Si se logra el acceso para el usuario pajaro y castor se debe de mostrar el siguiente menú:

### Administración General

[Administrar Catálogo Género](#)  
[Administrar Catálogo Autor](#)  
[Administrar Catálogo Disquera](#)  
[Administrar discos](#)  
[Ventas realizadas](#)

Si accesa el usuario master debe aparecer el siguiente menú, ya que es el único que puede administrar usuarios

## Administración General

Administrar Usuarios  
Administrar Catálogo Género  
Administrar Catálogo Autor  
Administrar Catálogo Disquera  
Administrar discos  
Ventas realizadas

La página de administrar usuarios se debe llamar administrausuario.php con el siguiente menú:

### Administración de usuarios

Acción	Scrip
<a href="#">Insertar un usuario</a>	insertausuario.php
<a href="#">Eliminar usuario</a>	eliminausuario.php
<a href="#">Actualizar un usuario</a>	actualizausuario.php
<a href="#">Listado de usuarios</a>	consultausuario.php

Con la diferencia que en eliminausuario.php se debe poder eliminar más de un usuario en una sola operación.

La página ventas realizadas es un listado que muestra todas las ventas registradas, la página se debe llamar ventas.php y debe mostrar algo como lo siguiente:

### Listado de compras

Título	Artista	Unidades Vendidas	Importe de la venta	Fecha de venta	Cliente
Éxitos	Shakira	2	\$360	24/10/2005	Rosario Rodríguez
Las inmortales	Los tigres del norte	1	\$660	24/10/2005	Christian Padilla

La página abierta al público se llamará tienda.php que debe listar todos los discos dados de alta en la tienda y permitir comprar un título (el número de unidades a comprar debe ser menor a 10, siempre y cuando haya en existencia) por venta, si se desea otro título, se debe realizar otra compra, la pagina deberá ser similar a la siguiente:

### TIENDA DE DISCOS PIXUP

Título	Artista	Genero	Costo	Descripción	
Éxitos	Shakira	Pop	\$180	Incluye todos los éxitos que la hicieron popular	<a href="#">Comprar</a>
Las inmortales	Los Tigres del Norte	Norteña	\$220	Solo para aficionados a la música norteña	<a href="#">Comprar</a>

El enlace donde se compra el disco se llama compra.php y debe tener un esquema como el siguiente:

### Realizar Compra

¿Si ya ha realizado compras con nosotros teclee su Id aquí?

Obtener Cliente

---

Si no cuenta con su Id llene el siguiente formulario:

RFC:

Nombre:

Apellido Paterno:

Apellido Materno:

Dirección:

Dirección de Entrega:

El cliente debe contar con un id que tiene que teclear para realizar una compra, de lo contrario debe de llenar todo el formulario para realizar la compra.

Su id para futuras compras es de 45

Cuando un usuario ya esta registrado se muestra lo siguiente:

### Articulo a comprar

Título	Artista	Género	Costo	Descripción	Cantidad de discos que desea adquirir:	
Éxitos	Shakira	Pop	\$180	Incluye todos los éxitos que la hicieron popular	<input type="text"/>	comprar

Si no hay ese titulo en existencia se debe mandar un mensaje de aviso y un enlace para regresar a la tienda, si hay en existencia el disco se continúa.

Una vez que se presiona el botón comprar se manda a la página adquirir.php que debe desplegar algo como lo siguiente:

## TIENDA DE DISCOS PIXUP

Gracias por su compra, su producto llegara en 10 días hábiles

### SU COMPRA

Título	Artista	Costo total de la compra
Éxitos	Shakira	\$360

[Adquirir otro título](#)

Se debe validar que hay el número de discos en existencia y si es así actualizar el campo existencia en la tabla DISCO. Validar aunque llene el formulario que no existe ya un registro de ese cliente (hacer validación sólo sobre el RFC, si ya se encuentra en la base de datos no se debe de volver a agregar y se debe registrar el id que ya estaba asignado).

Si el cliente no se encontraba en la base de datos se debe de agregar a la tabla CLIENTE, cuando se realiza la compra se debe asignar la correspondiente compra en la tabla VENTA, con esto concluye la venta y se le da la posibilidad de realizar una nueva compra.

### Base de datos

Para la creación de la base de datos se debe utilizar el DER (Diagrama Entidad Relación) de la Fig.1, y el editor de MySQL con los comandos vistos en el capítulo 6. Ejemplo:

Para la creación de la base de datos: `mysql> create database Pixup;`

Para la tabla de género:

```
mysql> create table Genero(IdGenero Int Not NULL auto_increment, primary key (IdGenero) ,  
Nombre varchar (25) not null, unique (Nombre))Type=innodb;
```

Para la tabla de disco se utiliza una llave foránea que hace referencia a autor y a genero:

```
mysql> create table Disco( IdDisco int not null auto_increment primary key, IdAutor int not null,  
IdGenero int not null, Titulo varchar (20) not null, imagen varchar (20), Costo DECIMAL (10,2)  
not null, Existencia int not null, Descripcion text, Foreign key (IdAutor) references  
Autor(IdAutor)ON DELETE CASCADE ON UPDATE CASCADE, foreign key (IdGenero)  
references Genero(IdGenero)ON DELETE CASCADE ON UPDATE CASCADE) type=innodb;
```

Para la table ventas que tiene relacionado el cliente y el disco:

```
mysql> create table Venta( IdVenta int not null auto_increment primary key, IdCliente int not null,  
IdDisco int not null, UnidadesVendidas int not null, ImporteVenta DECIMAL (10,2) not null,  
FechaVenta date not null, foreign key (IdCliente) references Cliente(IdCliente) ON DELETE NO  
ACTION ON UPDATE NO ACTION, foreign key (IdDisco) references Disco(IdDisco) ON  
DELETE NO ACTION ON UPDATE NO ACTION )type=innodb;
```



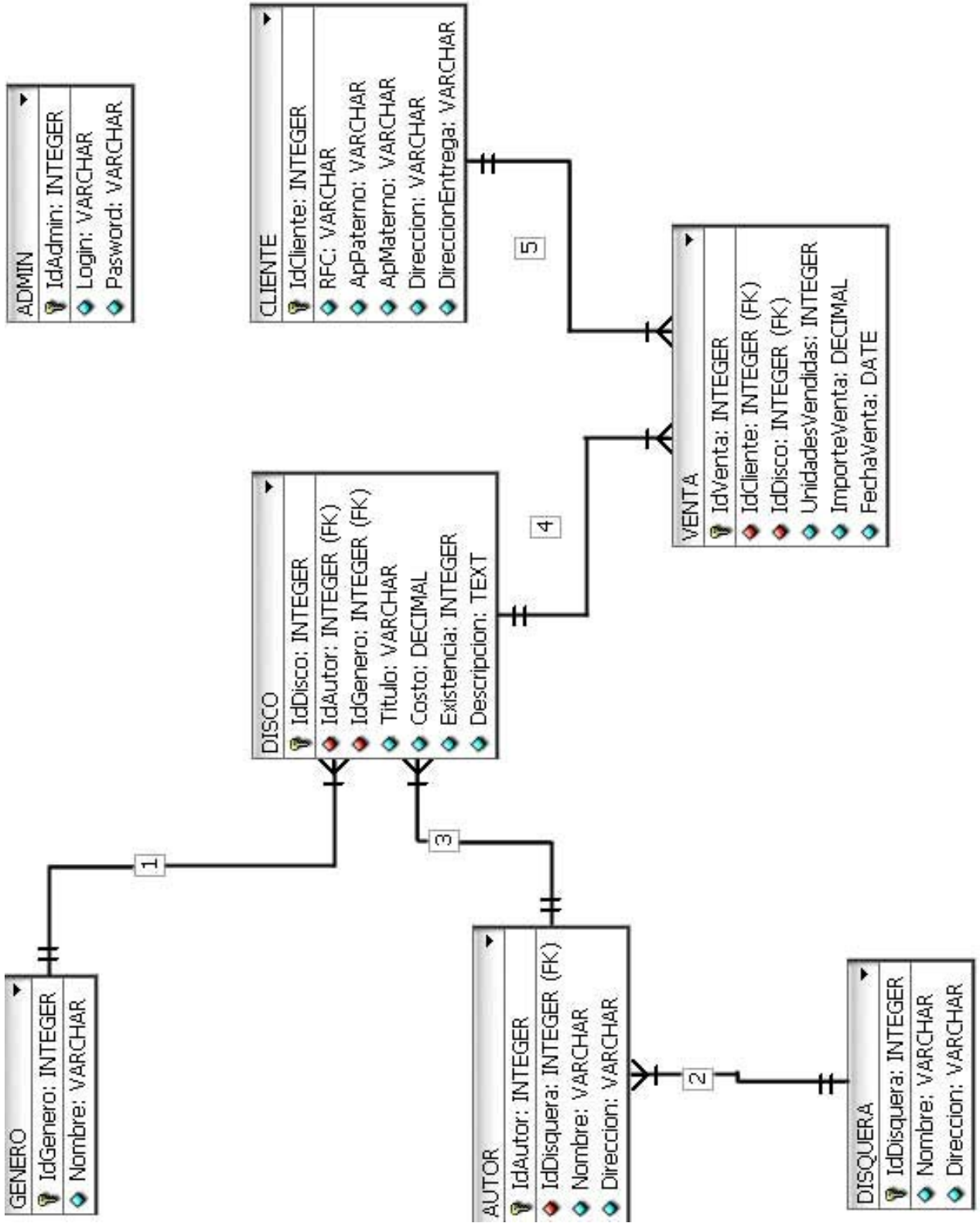


Fig.1 DER Pixup

## Código

Para hacer una conexión a la base de datos creada se utiliza el siguiente código en una página de nombre conexión.php:

```
<?                               Esta función permite hacer la conexión a la base de datos
function conectarse(){
    define("HOST","localhost");
    define("USUARIO","root");
    define("PASSWORD","");
    $conexion = mysql_pconnect(HOST,USUARIO,PASSWORD) or die("Error al conectarse " .
mysql_error());
return $conexion;
}
?>
```

Para verificar que la persona que introduce el login y password estan dentro de la base de datos se utiliza el siguiente código y la pagina que se presentase muestra en la Fig.2:

```
<?
session_start();                 Permite iniciar una sesión
if($_POST){                       Si hay un envío de datos POST
    require_once "lib/conexion.php";   se manda llamar la conexión
    $nick=$_POST['nick'];
    $password=$_POST['password'];
    function generasesion($nick,$password)   se declara una función para generar una sesión
    {
        $conexion=conectarse();
        Este es el comando que permite seleccionar la base de datos:
        $bd="Pixup";

        mysql_select_db($bd,$conexion) or die ("Error al seleccionar la base de datos " .
mysql_error($conexion));
        $sqlconsulta="SELECT * FROM Admin"; Se asigna una consulta a una variable
        Esta instrucción ejecuta la consulta y guarda el resultado en una variable
        $resultado = mysql_query($sqlconsulta,$conexion) or die("Error en la consulta " .
mysql_error($conexion));
```

El resultado que se obtiene se almacena en un arreglo y se comparan las dos cadenas ignorando si son mayúsculas o minúsculas. Si coinciden se regresa verdadero

```
while( ($fila=mysql_fetch_array($resultado,MYSQL_ASSOC)) ) {
    if((strcasecmp($nick,$fila['Login'])==0)&&strcmp($password,$fila['Password'])==0)
    {
        return true;
    }
    }   cierra while
return false;
} Cierra la función para generar la sesión
```

Si el login y el password no son validos no se genera la sesión

```
if(!generasesion($nick,$password))
{
?>
<html> Si en la base no se encuentra el nombre o password se muestra una pagina informativa
<body background="denegado.jpg">

<h1 align="center"><FONT color="White" size="+4">Error</FONT></h1><hr>
<?include "cabecera.html"; ?> se incluye una cabecera con código HTML

<p align="center"><FONT size=+3 face="Sans Serif" color="White">Lo sentimos <br>
Nombre o contraseña Incorrecta</FONT></p>
<br><br>
<?include "pie.php"; ?>
</body>
</html>

<?
}
else{
$_SESSION['nick']=$nick;
```

Si la persona que se registro es master se muestra una página con las siguientes ligas

```
if ($nick== "master"){
?>
<html>
<head>
<title>Administración</title>
</head>
<body background="fondo2.jpg">
<h1 align="center">Administración General</h1><hr>
<?include "cabecera.html"; ?> <br><br>
<center>
<a href="administrausuario.php">AdministrarUsuarios</a> <br>
<a href="genero.php">Administrar Catalogo Género</a> <br>
<a href="autor.php">Administrar Catalogo Autor</a><br>
<a href="disquera.php">Administrar Catalogo Disquera</a> <br>
<a href="administradisco.php">Administrar Discos</a> <br>
<a href="ventas.php">Ventas realizadas</a> <br>
<br><br><?include "pie.php"; ?>
</form>
</center>
</body>
</html>
<?
}
```

El resultado de este código se muestra en la Fig.3



Fig2 Pagina de inicio



Fig.3 Pagina administrador principal

Un ejemplo de cómo se introducen valores a la base de datos es la inserción de un disco:

```
<?
if($_GET){
    Si la pagina recibe los datos de un disco se asignan a variables
    $Nombre = $_GET['Nombre'];
    $Autor = $_GET['Autor'];
    $Genero = $_GET['Genero'];
    $Costo = $_GET['Costo'];
    $Existencia = $_GET['Existencia'];
    $Descripcion = $_GET['Descripcion'];
```

Con esta sentencia se le asigna a una variable a la sentencia a realizar

```
$sqlInserta="INSERT INTO Disco values ('','$Autor','$Genero','$Nombre','','$Costo',
'$Existencia','$Descripcion')";
```

Aquí se ejecuta la sentencia que en este caso inserta los datos a la tabla  
`$resultado=mysql_query($sqlInserta,$conexion) or die ("Error en la consulta "  
.mysql_error($conexion));`

Estas instrucciones informan si se agregaron los datos a la base

```
print "<br>Registro Insertado";  
print "<br>Disco agregado : <b>$Nombre</b><br><br>";
```

El resultado del código de inserción de discos es la Fig.4

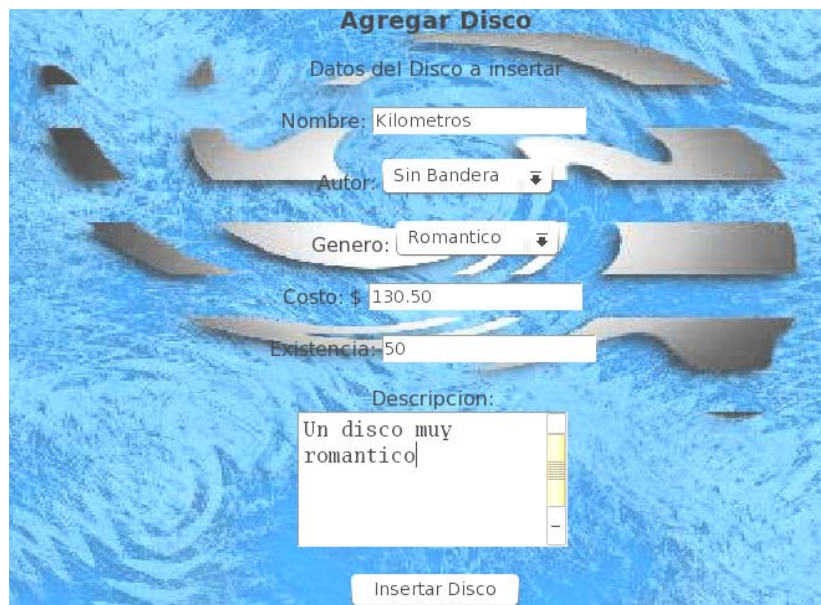


Fig. 4 Inserción de un disco



Fig.5 Pagina que muestra el disco insertado

El código con el que se hace la compra de un disco es:

```
<?
$IdDisco=$_POST['IdDisco'];
$Nombre=$_POST['Nombre'];
$Autor=$_POST['Autor'];
$Costo=$_POST['Costo'];           En esta parte se recuperan los valores de la venta
$Cantidad=$_POST['Cantidad'];
$IdCliente=$_POST['IdCliente'];
$Existencia=$_POST['Existencia'];
$Total=$Cantidad*$Costo;        Se hace el calculo de la cantidad a pagar

if ($Existencia<$Cantidad)  Se comprueba que la cantidad requerida este en existencia
{
echo "<h4>No hay la cantidad suficiente<br> de discos para este pedido.<br>La máxima cantidad
a <br>comprar es de: $Existencia </h4>";
    ?>
    <br><a href="tienda.php"><b>Regresar a tienda</b></a>
    <?
    }
else{

echo "<table border='1' align='center' bgcolor='White'>";
echo " <tr><td><b>Titulo</b></td><td><b>Artista</b></td><td><b>Costo total de la compra
</b></td></tr>";
echo "<tr>";
echo "<td > " . $Nombre . "</td>";           En una tabla se muestran los valores
echo "<td > " . $Autor . "</td>";           del disco que se eligió
echo "<td > " . "\$". $Total . "</td>";
echo "</tr>";
    echo "</table>";
    ?>
    <br><a href="tienda.php">Regresar a tienda</a>
    <?
    $resto=$Existencia-$Cantidad;
    $fecha=date("Y-m-d");
```

Aquí va el código que actualiza las tablas de la base de datos:

```
require_once "lib/conexion.php";
$conexion=conectarse();
$bd="Pixup";
    Se hace la conexión a la base de datos
mysql_select_db($bd,$conexion) or die ("Error al seleccionar la base de datos" .
mysql_error());
    Se actualizan la existencia del disco
$sqlactualiza="UPDATE Disco SET Existencia='$resto' WHERE IdDisco=$IdDisco";
```

```
$resultado=mysql_query($sqlactualiza,$conexion) or die ("Error al actualizar el disco".mysql_error());
```

```
require_once "lib/conexion.php";
```

```
$conexion=conectarse();
```

```
$bd="Pixup";
```

```
mysql_select_db($bd,$conexion) or die ("Error al seleccionar la base de datos" .mysql_error());
```

Se agrega la venta realizada a la tabla de ventas

```
$sqlactualiza="INSERT INTO Venta Values ('','$IdCliente', '$IdDisco', '$Cantidad', '$Total', '$fecha' )";
```

```
$resultado=mysql_query($sqlactualiza,$conexion) or die ("Error al actualizar el disco".mysql_error());
```

```
}  
>>
```



Fig.6 Ventana para realizar una compra



Fig.7 Ventana que muestra la compra realizada

En la página de ventas solo se hace una consulta de los datos de tabla ventas Fig.8.

Se hace una consulta a la base de datos a varias tablas, como cliente, disco, autor y venta. La información se ordenara por fecha

```
$sqlConsulta="SELECT Cliente.Nombre as 'Cliente', Cliente.ApPaterno as 'ApP',
Cliente.ApMaterno as 'ApM', Disco.Titulo as 'Titulo', Autor.Nombre as 'Artista',
Venta.UnidadesVendidas as 'Unidades', Venta.FechaVenta as 'Fecha', Venta.ImporteVenta as
'Importe' from Cliente, Venta, Autor, Disco where Venta.IdCliente=Cliente.IdCliente AND
Venta.IdDisco=Disco.IdDisco AND Disco.IdAutor=Autor.IdAutor ORDER BY
Venta.FechaVenta";
```

Se ejecuta la consulta

```
$resultado=mysql_query($sqlConsulta,$conexion) or die ("Error en la consulta " .
mysql_error($conexion));
```

se pinta una tabla

```
echo "<table border='1' align='center' bgcolor='White'>";
echo "<tr><td><b>Autor</b></td>";
echo "<td><b>Artista</b></td>";
echo "<td><b>Unidades<br> Vendidas</b></td>";
echo "<td><b>Importe de<br> la Venta</b>";
echo "</td><td><b>Fecha de<br> Venta</b></td>";
echo "<td><b>Cliente</b></td></tr>";
```

Los elementos de la consulta se almacenan en un arreglo

```
while($fila=mysql_fetch_array($resultado,MYSQL_ASSOC))
{
Mientras haya información en el resultado de la consulta se muestra en la tabla
echo "<tr>";
echo "<td > " . $fila[Titulo] . "</td>";
echo "<td > " . $fila[Artista] . "</td>";
echo "<td align='center' > " . $fila[Unidades] . "</td>";
echo "<td align='center'> " . "\$". $fila[Importe] . "</td>";
echo "<td > " . $fila[Fecha] . "</td>";
echo "<td > " . $fila[Cliente]. " " . $fila[ApP]. " " . $fila[ApM] . "</td>";
echo "</tr>";
}
echo "</table>";
```



**Administración General Ventas**

**Tienda de discos**



**Consulta de Ventas**

Autor	Artista	Unidades Vendidas	Importe de la Venta	Fecha de Venta	Cliente
No te apartes de mi	Yahir	1	\$76.50	2005-10-08	Rosario Rodríguez Gonzalez
No te apartes de mi	Yahir	4	\$306.00	2005-10-08	Valeria Rodríguez Gonzalez
Una lagrima	Sin Bandera	6	\$693.00	2005-10-08	Valeria Rodríguez Gonzalez
Miedo	Pepe Aguilar	2	\$192.00	2005-10-08	Christian Padilla Cabrera
Una lagrima	Sin Bandera	1	\$115.50	2006-11-27	Rosario Rodríguez Gonzalez
Una lagrima	Sin Bandera	1	\$115.50	2006-11-27	Rosario Rodríguez Gonzalez
Kilometros	Sin Bandera	6	\$783.00	2006-11-28	

Fecha: 28 / 11 / 2006

Pixup creada por  
Rodríguez González María del Rosario

Fig.8 Ventana de ventas

Para finalizar se debe mencionar que en este capítulo se mostró solamente parte del código básico del proyecto tanto para la creación de la base de datos como para las páginas.

Este capítulo muestra como se pueden aplicar las herramientas vistas a lo largo de éste trabajo para desarrollar un sistema, que en este caso es una tienda virtual. Es posible crear sistemas más complejos pero esto depende de las necesidades del usuario.

## Conclusión

Considero que las herramientas que se trataron en este trabajo son ampliamente recomendables para el desarrollo de sistemas en red, ya que cuentan con muchas opciones para adaptarlas a las necesidades que se tengan.

Aunque la mayoría las herramientas vistas son de licencia gratuita, también pueden ser utilizadas en sistemas que cuentan con una licencia diferente, como el caso del sistema operativo Windows.

Cada una de las herramientas puede ser utilizada de forma individual, ya sea para crear una base de datos, para diseñar una página o para crear una imagen, pero creo que es una buena opción usarlas en conjunto.

Como se observó en el capítulo 9 se utilizan HTML, PHP y MySQL para desarrollar una tienda virtual, que tiene una base de datos, la cual esta almacenada en un equipo con sistema Slackware, éste a su vez cuenta con un servidor Apache que nos permite acceder a las paginas Web de este proyecto.

El software libre comienza a tener una amplia demanda para diseñar y administrar sistemas, por esto es recomendable conocer no solo las herramientas vistas en este trabajo sino las que también permiten realizar otras tareas.

Además de aprendan a utilizar las herramientas vistas a lo largo de este trabajo, es importante estar al tanto de sus actualizaciones (en su página oficial) y sobre todo tenerlas en constante práctica.

## Glosario

**Apache:** Es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows y otras, que implementa el protocolo HTTP y la noción de sitio virtual.

**Archivo:** Sinónimo de fichero y esencia del almacenamiento informático. Delimitado por una cabecera y una marca de final, lugar donde puede estar ubicada cualquier tipo de información, ya sea texto, programas, imágenes, sonidos, etc.

**Base de datos:** Es un conjunto de tablas dedicadas a guardar información relacionada entre sí, con referencias entre ellas de manera que se complementen con el principio de no duplicidad de datos.

**Comando:** Es una orden que se da, generalmente al sistema operativo de la computadora, para que realice la tarea que esa orden conlleva.

**Consola:** Ver Shell.

**Driver:** Aunque puede referirse a dispositivos físicos, se utiliza normalmente para indicar los controladores de periféricos, es decir, el software que permite al Sistema Operativo reconocer y utilizar lo que tiene conectado el ordenador.

**Hardware** Se denomina así al conjunto de componentes físicos dentro de la informática (un teclado, una placa, por ej.).

**Hilo:** Un hilo de ejecución, en sistemas operativos, es similar a un proceso en que ambos representan una secuencia simple de instrucciones ejecutada en paralelo con otras secuencias.

**Host:** Ver Servidor.

**HTML:** Es un lenguaje que se emplea para crear páginas Web. Un código escrito en este lenguaje es, básicamente, un texto que el navegador mostrará en formato de página Web.

**HTTP:** Protocolo de transferencia de hipertexto es el lenguaje que utilizan el cliente y el servidor para comunicarse entre si.

**Inodo:** En informática, un inodo, nodo-i, nodo índice o i-node en inglés es una estructura de datos propia de los sistemas de archivos tradicionalmente empleados en los sistemas operativos tipo UNIX como es el caso de Linux. Un inodo contiene las características (permisos, fechas, ubicación, pero NO el nombre) de un archivo regular, directorio, o cualquier otro objeto que pueda contener el sistema de ficheros.

**Kernel** Se suele traducir como núcleo, y en general se entiende como el centro o motor de cualquier Sistema Operativo.

**Lenguaje de Programación:** Son secuencias de instrucciones, más o menos parecidas al lenguaje humano, que permiten, una vez traducidas o compiladas, ejecutar las órdenes incluidas en el programa.

**Librerías** Conjunto de programas con un formato determinado para ser utilizados por los desarrolladores de aplicaciones, con el objetivo principal de evitar la repetición de procesos.

**Licencia GNU:** La licencia pública general (GNU) está destinada a garantizar la libertad de compartir y cambiar el software gratis.

**LILO:** ("Linux LOader") es un gestor de arranque de Linux que permite instalar este sistema operativo junto con otras plataformas en el mismo ordenador.

**MBR:** Master Boot Record o registro de arranque maestro. Es el primer sector del disco al cual se accede buscando la carga del Sistema Operativo como ocurre con Windows, o un sistema de arranque, como es el caso de Linux.

**Memoria:** Tanto en ordenadores como en periféricos, lugar donde se almacenan datos o programas mientras se están utilizando.

**Multitarea:** Realización simultánea de varios procesos en un ordenador, teniendo en cuenta que éste y su Sistema Operativo lo permitan.

**Multiusuario:** Utilización simultánea de un ordenador por varios usuarios, o dicho de otra manera, ordenador y Sistema Operativo con capacidad para permitir el trabajar desde distintas consolas con la misma CPU.

**MySQL:** Es un sistema de gestión de bases de datos relacional, bajo licencia GNU.

**Página:** Documento situado en una red informática, al que se accede mediante enlaces de hipertexto.

**PHP:** Es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. PHP es un acrónimo recurrente que significa "PHP Hypertext Pre-processor" y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web.

**PostgreSQL:** Es un motor de base de datos, es servidor de base de datos relacional libre, liberado bajo la licencia BSD.

**Programa:** Instrucciones que varían según el lenguaje que se utiliza, pero cuyo fin es el de controlar las acciones que tiene que llevar a cabo el ordenador y sus periféricos.

**Protocolo:** Es un término de comunicaciones y su función es fijar unas reglas de funcionamiento, a todos los niveles, a las que han de atenerse los distintos sistemas informáticos para poder comprenderse.

**Script** Son programas que se ejecutan en un servidor Web dedicados a procesar las peticiones que le llegan de los navegadores.

**Seguridad:** Es un mecanismo que da la garantía de un buen funcionamiento y que evita, en caso de fallo de otro mecanismo, que se produzca un accidente o daño.

**Sentencia:** El conjunto de órdenes que tienen un sentido, de tal forma que el ordenador sepa ejecutar una acción. Un ejemplo, en pseudocódigo: Si I es mayor que X haz Función Z.

**Servidor** Se denomina así al ordenador que se encarga de suministrar lo necesario a una red, dependiendo de cual sea la finalidad de ésta.

**Shell:** Es el interprete de comandos de los sistemas de la familia UNIX. Básicamente funcionan dando una línea de comandos al usuario y ejecutando los comandos que se le piden.

**Sistema Operativo** Es un conjunto de programas que sirve de enlace entre el ordenador y el programador usuario. Son los responsables de gestionar los recursos del ordenador, discos duros, memorias, control de periféricos como pantallas, teclados, etc. De alguna forma unifican y estandarizan el funcionamiento de los ordenadores.

**Sitio (Site):** Se refiere a un lugar o dirección Web cualquiera con una URL.

**Software:** Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

**Software libre:** Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El software libre suele estar disponible gratuitamente en Internet, o a precio del coste de la distribución a través de otros medios.

**SQL:** Structured Query Language. Lenguaje o sentencias para el manejo y consulta de bases de datos.

**Virus:** Es un programa, generalmente destructivo, que se introducen en el ordenador al acceder a una red informática o al leer un disco, y pueden provocar pérdida de la información almacenada en el disco duro o algún daño al sistema operativo.

**Web:** Puede referirse a "una Web" como una página, sitio o conjunto de sitios que proveen información por los medios descritos, o a "la Web", que es la enorme e interconectada red disponible prácticamente en todos los sitios de Internet.

## Bibliografía:

- Bandel David y Napier Robert, “Edición Especial Linux”, 6° edición., Madrid 2001, Prentice Hall
- Bowen Rich & Coar Ken, “Servidor Apache al Descubierto”, Madrid 2000, Prentice Hall
- Castaño Adoración de Miguel y Piattini Mario, “Fundamentos y modelos de bases de datos”, 2<sup>da</sup> Edición, México 1999, Alfaomega-Ra-Ma
- Castaño Adoración de Miguel, Piattini Mario y Martínez Esperanza Marcos, “Diseño de bases de datos relacionales”, México 2000, Alfaomega-Ra-Ma
- Córdoba Enrique, González Carmen y Cordoba Carmen, Photoshop “CS Curso Completo”, España 2005, Alfaomega- RA-MA
- Dee-Ann Le Blanc, “La Biblia de administración de sistemas Linux”, México, Anaya
- Dr Shwarte Jachim, “El gran libro de HTML como publicar en Internet”, México, Marcombo
- García Jiménez Francisco, “Guía de campo de Linux” México, RA-MA
- Guiérrez Abraham & Bravo Ginés, “PHP5 a través de ejemplos”, México 2005, Alfaomega RA-MA
- Lerdorf Ramus and McGredy Ric, “Programming PHP”, USA 2002, O'REILLY
- López Quijado José, “Domine HTML y DHTML” México, Alfaomega Ram-Ma
- Microsoft® Encarta® 2007. © 1993-2006 Microsoft Corporation. Reservados todos los derechos
- Minera Francisco José, PHP y MySQL integración total, Argentina 2005, MP Ediciones
- Pascual Francisco, “Photoshop 7”, Madrid 2003, Alfaomega Ra-Ma
- Pott Oliver, “Gran libro de HTML y XML Referencia y Práctica” México, Marcombo- Boixareu Editores
- Williams E. Hugo and Lane David, “Web Database Applications with PHP and MySQL”, USA 2004, O'REILLY

## Páginas de Internet Consultadas

- [http://es.wikipedia.org/wiki/Bases\\_de\\_datos](http://es.wikipedia.org/wiki/Bases_de_datos)
- <http://es.wikipedia.org/wiki/Gnu>
- <http://es.wikipedia.org/wiki/Nessus>
- <http://es.wikipedia.org/wiki/PGP>
- <http://es.wikipedia.org/wiki/Php>
- [http://es.wikipedia.org/wiki/Servidor\\_web\\_Apache](http://es.wikipedia.org/wiki/Servidor_web_Apache)
- [http://es.wikipedia.org/wiki/Sistema\\_Gestor\\_de\\_Bases\\_de\\_Datos](http://es.wikipedia.org/wiki/Sistema_Gestor_de_Bases_de_Datos)
- [http://metabolik.hacklabs.org/alephandria/txt/txipi\\_seguridad.html](http://metabolik.hacklabs.org/alephandria/txt/txipi_seguridad.html)
- [http://metabolik.hacklabs.org/alephandria/txt/txipi\\_seguridad.html](http://metabolik.hacklabs.org/alephandria/txt/txipi_seguridad.html)
- <http://osdir.com/Downloads+index-req-viewdownloaddetails-lid-829-ttitle-osiris.phtml>
- [http://todo-photoshop.com/tutorialphotoshop/basicos/cuadro\\_herramientas\\_photoshop.html](http://todo-photoshop.com/tutorialphotoshop/basicos/cuadro_herramientas_photoshop.html)
- <http://www.aprender.org.ar/aulas/avafp/recursos/html.htm>
- [http://www.aulacli.es/photoshop/t\\_4\\_2.htm](http://www.aulacli.es/photoshop/t_4_2.htm)
- [http://www.conclase.net/c/mysql/index.php?sen=ALTER\\_TABLE](http://www.conclase.net/c/mysql/index.php?sen=ALTER_TABLE)
- <http://www.europe.redhat.com/documentation/rhl7.1/rhl-rg-es-7.1/ch-configuration.php3>
- <http://www.ing.ula.ve/~herman/comht.html>
- [http://www.netpecos.org/docs/mysql\\_postgres/index.html](http://www.netpecos.org/docs/mysql_postgres/index.html)
- <http://www.outguess.org/>
- <http://www.php.net/manual/es/function.pg-connect.php>
- <http://www.webtaller.com/manual-php/operadores.php>