



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

---

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

## INFORME DEL PROYECTO DE UN SISTEMA DE CONTROL DE INVENTARIOS CON SOFTWARE LIBRE

TRABAJO ESCRITO EN LA MODALIDAD DE  
SEMINARIOS Y CURSOS DE ACTUALIZACIÓN  
Y CAPACITACIÓN PROFESIONAL  
QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN COMPUTACIÓN  
P R E S E N T A:  
MIGUEL ÁNGEL MENDOZA HERNÁNDEZ

MÉXICO

2007



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.

## **Agradecimientos.**

A Dios por ser el aliento de vida que bebo a diario.

A mi Abuelo Fausto, que siempre ha estado a mi lado acompañándome en todo momento y siempre dispuesto a escucharme cuando más lo necesito. Donde quiera que estés gracias por tu amor y por mantener vivo tu reflejo en mí.

A mi Padre, mi mejor amigo, por su eterno apoyo y amor incondicional, sin sus sabios consejos no estaría en este lugar ahora.

A mi Madre, por ser mi ejemplo de lucha y superación constante, su apoyo y profundo amor incondicional, por haberme creado lo que soy y no dejarme ir por caminos equívocos, por ella estoy aquí.

A mis hermanos Elvia, Felipe, Claudia y Flor, por su amor y apoyo e inyectarme esas ganas de triunfar en todo momento, los amo.

A mis cuñados Luis y Leonardo y mi cuñada Mayra por su cariño y su apoyo incondicional.

A mis sobrinos, todos y cada uno son pieza fundamental en mi vida, los amo.

A mi suegro, por todo su cariño y comprensión y ese empuje constante que me hace ser mejor cada día, por ser quien es y mostrarme muchas veces la luz cuando veo oscuridad.

A mi suegra, por todo su cariño y apoyo incondicional y esa luz que me muestra en cada sonrisa y cada palmada en el hombro impulsándome a seguir adelante.

A mis cuñadas, por su apoyo, amistad y cariño y por inyectarme las ganas de superarme día con día.

A mis amigos César, Joslyn, Saúl, Javier, Ricardo, sin ustedes las escuela no hubiera sido igual, gracias por su apoyo y por ser un ejemplo constante de crecimiento.

A mis maestros, por compartir sus conocimientos y enseñarme a ser un hombre de bien.

A Rosy, mi adorada esposa, por todo su amor y apoyo, por ser un gran impulso para triunfar en la vida, por mostrarme el incansable deseo de superación día con día, por ser la compañera de mi vida. Te AMO.

A todos y cada uno de los amigos, familiares que han estado presentes en cada etapa de mi vida gracias.

Y finalmente gracias a la vida por permitirme estar aquí.

## ÍNDICE

ÍNDICE .....	1
Introducción.....	3
Comparativa de algunas distribuciones Linux.....	6
1. Informe sobre el Diplomado “Desarrollo de sistemas con Software Libre”.....	8
1.1 Sistema Operativo Linux.....	8
Objetivo General. ....	8
Objetivo específico.....	8
Informe Módulo I .....	8
1.2 Instalación y administración de Linux. ....	27
Objetivo General. ....	27
Objetivos específicos. ....	27
Informe del módulo II.....	27
1.3 Editores para la creación de páginas Web. ....	45
Objetivo General .....	45
Objetivo Especifico.....	45
Informe del módulo III.....	45
1.4 Administración de Servidores WEB con Linux.....	53
Objetivo General. ....	53
Objetivo específico.....	53
Informe del módulo IV .....	53
1.5 Programación con PHP .....	63
Objetivo General. ....	63
Objetivo específico.....	63
Informe del módulo V .....	63
1.6 Interacción de WWW con bases de datos MySQL. ....	72
Objetivo General. ....	72
Objetivo Especifico.....	72
Informe del Módulo VI. ....	72
1.7 Introducción a la Seguridad en cómputo.....	81
Objetivo general. ....	81
Objetivo específico. ....	81
Informe del módulo VII. ....	81
1.8 Desarrollo de aplicaciones con Postgresql y PHP.....	100
Objetivo General. ....	100
Objetivo específico.....	100
Informe del Módulo VIII .....	100
2. Desarrollo de un sistema de control de inventarios con software libre.....	120
2.1 Características de hardware y software utilizado en este proyecto. ....	120
2.2 La plataforma de software utilizado en este proyecto fue la siguiente:.....	121
Conclusiones.....	131
Bibliografía .....	132
Mesografía .....	133

## Introducción

## Introducción.

El presente trabajo refleja las características del software libre y su entorno de desarrollo, el impacto que ha tenido en la actualidad y sus posibles aplicaciones.

La década de los noventa sirvió como plataforma para el lanzamiento de Linux como sistema operativo basado en UNIX, y tomando como base la modificación directa del núcleo de Minix, un sistema operativo creado por Andrew S. Tannenbaum.

Contando con un equipo 80386 con MSDOS precargado, Linus Torvalds comenzó a diseñar el núcleo de lo que sería su versión de sistema operativo basado en UNIX, inspirado por las ideas plasmadas en el libro publicado por Maurice J. Bach en el año de 1986 titulado *Diseño del Sistema Operativo UNIX (Design of the Unix Operating System)*, modificando de manera gradual el núcleo de Minix crearía una adaptación del potente S.O. que fuera capaz de ejecutar el software creado por el *Proyecto GNU*, (Richard Stallman crea en 1983 la Fundación de Software Libre, '*Free Software Foundation*', y comenzó el proyecto *GNU* que significa GNU no es UNIX) pero sobre una arquitectura IBM/PC.

Es hasta el año de 1991 cuando Linus Torvalds anuncia la primera versión de su sistema operativo Linux capaz de ejecutar un interprete de comandos "shell" llamada BASH (Bourne Again Shell, en referencia a su creador Stephen Bourne) y el compilador de aplicaciones de software GNU GCC (Compilador de Aplicaciones GNU-GNU *Compiler Collection*).

El núcleo de Linux (*kernel* en inglés) está escrito casi en su totalidad en lenguaje C con algunas extensiones GNU C, con aportaciones de ideas y código al núcleo de miles de programadores en todo el mundo. Dicho propiamente el núcleo de Linux se refiere al software de sistema de bajo nivel que provee una capa de abstracción sobre el hardware, control de discos, sistema de archivos, multitarea, multiusuario, balance de carga, comunicación en red y medidas de seguridad.

El diseño estructural de Linux tiene su base en una filosofía de sistema y una estructura de sistema, cuando se habla de filosofía de sistema, se entiende que es la manera en que Linux maneja los archivos, en su diseño, Linux cuenta con determinadas características, entre ellas, que es un sistema operativo multitarea, esto quiere decir que puede estar ejecutando distintas tareas a la vez sin desfavorecer el rendimiento tal como cualquier sistema UNIX, es también un sistema multiusuario, ya que permite que múltiples usuarios desde distintos puntos o terminales remotas, trabajen al mismo tiempo sin que sus tareas y procesos ejecutados se vean afectados en lentitud o en ejecución, ya que todos los

procesos ejecutados por cada usuario les es asignado un identificador o número de proceso que es controlado por un administrador de tareas de cada usuario corriendo en modo oculto (background, como se le conoce en el idioma inglés), siendo estos totalmente independientes, lo que le hace un sistema operativo muy robusto, seguro y confiable.

La filosofía del sistema Linux, al igual que su estructura de archivos, y su diseño, esta basado totalmente en UNIX, se dice que un sistema operativo potente y complejo debe de ser simple, general y extensible, cuyo enfoque queda totalmente dirigido a la manera en que Linux maneja los archivos y la forma de manejar sus herramientas, basándose en el principio fundamental que busca el diseñar comandos que hagan solamente una cosa, pero que lo hagan de la manera más simple y correcta posible. Los archivos en Linux son vistos de manera simple y general dentro de un modelo de archivos único, es decir, ve y trata de la misma manera a los directorios, archivos comunes, a los dispositivos como por ejemplo un monitor, un teclado, el mouse, la tarjeta de video, terminales, etc.

Como se ha venido mencionando la estructura del sistema de archivos de Linux está estructurada por niveles o jerarquías, al igual que en UNIX, se puede ver como un modelo de capas en donde la capa más baja del modelo esta representada por el hardware con el que se cuenta, y el cual consiste en discos, CPU, memoria, y demás dispositivos, la capa inmediata superior que corre sobre el hardware se encuentra el kernel del sistema, que es como el corazón del mismo y cuya función es controlar totalmente todo el hardware, para presentarle al usuario información de forma coherente y de esta manera crear una interacción entre el usuario y la máquina y que le permitan la creación y manejo de procesos, archivos y demás recursos del sistema. La siguiente capa es el interprete de comandos mejor conocido como shell, el cual se encarga de ser el medio por el cual el usuario interacciona con el sistema, como lo mencionamos anteriormente un shell es un interprete de comandos que se encarga de analizar y ejecutar los comandos que son escritos por medio de una terminal y el cual puede ser ejecutado por cualquier usuario del sistema ya que no se necesita tener privilegios especiales puesto que para Linux es tan solo un programa más. La última capa es la del usuario y sus aplicaciones, en la cual se encuentran todos los programas utilizados por el usuario, aquí todo es más a nivel de software (programas y aplicaciones).

Desde la primera versión del núcleo de Linux liberada han sido varios los programadores que han contribuido a que el código crezca y el sistema operativo sea más sólido y estable.

Debido a que son miles los programadores que desarrollan código para el kernel de Linux, combinado con software desarrollado por otros tantos se determinó en llamar lo que hoy en día se conoce como versiones o *distribuciones*

del sistema GNU/Linux. Las distribuciones de Linux se caracterizan por sus desarrolladores encargados que normalmente mantienen sus propias versiones del núcleo y que en ocasiones incluyen, por ejemplo, controladores de hardware que no han sido incorporados en la versión oficial, o bien incluyen su propia *suite* de programas (software) para uso exclusivo de la distribución, en general, se destacan por las herramientas para su configuración y los sistemas de paquetes de software para instalar.

Para entender que es una distribución hay que comprender que es un sistema operativo. Un sistema operativo es simplemente un cúmulo de programas que permiten al usuario servirse de la computadora para hacer algo; eventualmente para cargar o instalar otros programas adicionales y usarlos para ciertas tareas, o si no, limitarse a usar lo que venga incluido en el propio sistema operativo.

Existen numerosas distribuciones de Linux, cada una de ellas puede incluir cualquier cantidad de software ya sea este libre o no, o como es el caso de algunas de ellas, que facilitan por completo el proceso de instalación y configuración del sistema sobre prácticamente cualquier equipo y que cuentan con una gran variedad de aplicaciones, entre ellas, interfaces o entornos gráficos, suites ofimáticas, servidores WEB, servidores de correo, servidores FTP, software de seguridad computacional entre otros.

La base de cada distribución tendrá incluido el núcleo de Linux (Kernel), con bibliotecas y herramientas del proyecto GNU, y también cuentan con otras herramientas de otros grupos o proyectos, tal es el caso de BSD.

Como es de esperarse debe existir un estándar entre las distintas distribuciones de Linux, y esta controlado por la *Fundación de Estándares Linux (Linux Standard Base)*, que es una organización enfocada a desarrollar una cooperación estrecha entre las distintas distribuciones. Entre otros estándares que deben manejarse en Linux se encuentra el estándar de manejo en el sistema de archivos conocido como *Estándar Jerárquico del Sistema de Archivos (Filesystem Hierarchy Standard)*, que es una herramienta muy importante de la organización para lograr una cierta normalización oficial.

A continuación se muestra una tabla comparativa, de las diferentes distribuciones Linux que se utilizan actualmente.



## Comparativa de algunas distribuciones Linux

### General

	Empresa	Fecha de la primera P.R.	Predecesor	Última versión estable	Precio	Licencia	Público	País
<a href="#">Debian GNU/Linux</a>	<a href="#">Debian Project</a>	<a href="#">Agosto 1993</a>	N/A	3.1r0a (sarge)	Gratis	cualquier <a href="#">DFSG</a>	Desktop, Workstation, Server	<a href="#">Mundial</a>
<a href="#">Fedora Core</a>	<a href="#">Fedora Project</a>	<a href="#">Noviembre 2003</a>	<a href="#">Fedora Linux</a> , <a href="#">Red Hat Linux</a>	4	Gratis	<a href="#">GPL</a>	Workstation, Server, Público	<a href="#">EEUU</a>
<a href="#">Gentoo</a>	<a href="#">Gentoo Foundation</a>	<a href="#">Marzo 2002</a>	Enoch	2005.1	Gratis	<a href="#">GPL</a>	Workstation, Server, Público	<a href="#">Mundial</a>
<a href="#">Mandriva Linux</a>	<a href="#">Mandriva</a>	<a href="#">Julio 1998</a>	<a href="#">Mandrakelinux</a> y <a href="#">Conectiva</a>	Limited Edition 2005	Gratis (Download edition)	<a href="#">GPL</a>	Desktop, Workstation, Server	<a href="#">Francia</a>
<a href="#">Slackware Linux</a>	<a href="#">Patrick Volkerding</a>	<a href="#">Julio 1993</a>	<a href="#">SLS</a>	10.2	Gratis	<a href="#">GPL</a>	Workstation, Server, Público	<a href="#">EEUU</a>
<a href="#">SUSE Linux</a>	<a href="#">Novell</a>	<a href="#">Marzo 1994</a>	<a href="#">Jurix</a>	Ninguna	Descarga gratuita disponible Ed. Profesional: 74,54	<a href="#">GPL</a>	Workstation, Server, Público	<a href="#">Alemania</a>

### Técnica

	<a href="#">Kernel</a>	<a href="#">Sistema de ficheros</a>	<a href="#">Sistemas de ficheros soportados</a>	Arquitectura	Herramienta de Actualización online	Administrador de Paquetes	API principal e Idioma para Aplicaciones Gráficas	API principal e Idioma para Aplicaciones CLI
<a href="#">Debian GNU/Linux</a>	<a href="#">Linux</a> 2.4.27/2.6.8	<a href="#">ext3</a>	<a href="#">ext2</a> , <a href="#">JFS</a> , <a href="#">XFS</a> , <a href="#">FAT</a> , <a href="#">NTFS</a> , <a href="#">ISO 9660</a> , <a href="#">UDF</a> , <a href="#">NFS</a> , <a href="#">ReiserFS</a>	<a href="#">x86</a> , <a href="#">IA64</a> , <a href="#">PPC</a> , <a href="#">SPARC</a> , <a href="#">SPARC64</a> , <a href="#">Alpha</a> , <a href="#">MIPS</a> , <a href="#">ARM</a> , <a href="#">PA-RISC</a> , Mac/VME <a href="#">68k</a> , <a href="#">S/390</a>	<a href="#">APT</a>	<a href="#">dpkg</a> , <a href="#">Synaptic</a> , and <a href="#">APT</a>	Various (no default)	pre- <a href="#">LSB</a> con C, otros (estandar <a href="#">POSIX</a> )
<a href="#">Fedora Core</a>	<a href="#">Linux</a> 2.6.11	<a href="#">ext3</a>	<a href="#">ext2</a> , <a href="#">ReiserFS</a> , <a href="#">FAT</a> , <a href="#">ISO 9660</a> , <a href="#">UDF</a> , <a href="#">NFS</a>	<a href="#">x86</a> , <a href="#">x86-64</a> , <a href="#">i386</a> , <a href="#">PowerPC</a>	<a href="#">up2date</a> , <a href="#">yum</a> , <a href="#">APT</a> (limited)	<a href="#">RPM</a> , <a href="#">yum</a>	Various (no default)	<a href="#">LSB</a> con C, otros (estandar <a href="#">POSIX</a> )
<a href="#">Mandriva Linux</a>	<a href="#">Linux</a> 2.6.11.6	<a href="#">ext3</a>	<a href="#">ext2</a> , <a href="#">JFS</a> , <a href="#">XFS</a> , <a href="#">FAT</a> , <a href="#">NTFS</a> , <a href="#">ISO 9660</a> , <a href="#">UDF</a> , <a href="#">NFS</a> , <a href="#">ReiserFS</a>	<a href="#">x86</a> (i586), <a href="#">x86-64</a> , <a href="#">PPC</a>	<a href="#">urpmi</a>	<a href="#">RpmDrake</a>	Various (no default)	<a href="#">LSB</a> con C, otros (estandar <a href="#">POSIX</a> )
<a href="#">Slackware Linux</a>	<a href="#">Linux</a> 2.4.29	<a href="#">ReiserFS</a> , <a href="#">ext3</a> / <a href="#">ext2</a>	<a href="#">JFS</a> , <a href="#">XFS</a> , <a href="#">FAT</a> , <a href="#">NTFS</a> , <a href="#">ISO 9660</a> , <a href="#">UDF</a> , <a href="#">NFS</a>	<a href="#">x86</a> , <a href="#">IA64</a> , <a href="#">S/390</a>	<a href="#">Swaret</a> , <a href="#">Slapt-get</a> , many other not official	<a href="#">installpkg</a> and <a href="#">upgradepkg</a>	Various (no default)	<a href="#">LSB</a> con C, otros (estandar <a href="#">POSIX</a> )
<a href="#">SUSE Linux</a>	<a href="#">Linux</a> 2.6.11.4	<a href="#">ReiserFS</a>	<a href="#">ext2</a> , <a href="#">ext3</a> , <a href="#">JFS</a> , <a href="#">XFS</a> , <a href="#">FAT</a> , <a href="#">NTFS</a> , <a href="#">ISO 9660</a> , <a href="#">UDF</a> , <a href="#">NFS</a> , <a href="#">Reiser4</a>	<a href="#">x86</a> , <a href="#">IA64</a> , <a href="#">x86-64</a> , <a href="#">PPC</a>	<a href="#">YaST2</a>	<a href="#">RPM</a> , YaST	Various (no default)	<a href="#">LSB</a> con C, otros (estandar <a href="#">POSIX</a> )

## Extras

	Paquetes	¿Instalación Gráfica?	Explorador de ficheros	Navegador Web	Entorno Gráfico Principal	Window manager Principal	Tema de Escritorio	Paquete de Ofimática
<a href="#">Debian GNU/Linux</a>	18000	No	N/A	N/A	N/A	N/A	N/A	N/A
<a href="#">Fedora Core</a>	5000	Si	<a href="#">Nautilus</a>	<a href="#">Epiphany</a>	<a href="#">GNOME</a>	<a href="#">Metacity</a>	<a href="#">ClearLooks</a> theme for GNOME <a href="#">Bluecurve</a> theme for KDE	<a href="#">OpenOffice.org/KOffice/GNOME Office</a>
<a href="#">Mandriva Linux</a>	4000	Si	<a href="#">Konqueror</a>	<a href="#">Konqueror</a>	<a href="#">KDE</a>	<a href="#">kwin</a>	Mandrakegalaxy theme	<a href="#">OpenOffice.org</a>
<a href="#">Slackware Linux</a>	?	Si	N/A	N/A	N/A	N/A	N/A	N/A
<a href="#">SUSE Linux</a>	12500	Si	<a href="#">Nautilus</a>	<a href="#">Mozilla Firefox</a>	<a href="#">Gnome - KDE</a>	<a href="#">Metacity</a>	ClearLooks	<a href="#">OpenOffice.org</a>

Existen muchas distribuciones más, de las cuales las hay libres y comerciales, la distribución con la que se trabajo durante todo el diplomado fue Slackware en su versión 10.2, es la primera distribución de Linux, desarrollada por Patrick Volkerding en el año de 1993, contiene un programa de instalación fácil de usar, documentación extensa y un sistema de manejo de paquetes basado en menús. La instalación completa de esta distro incluye el Sistema de ventanas X (interfaz gráfica), entornos de escritorio GNOME, KDE, Xfce, entornos de desarrollo C/C++, Perl, Phyton, Java, también cuenta con utilidades de red, servidor de correo, servidor de noticias, servidor WEB, servidor FTP, navegadores WEB, suite ofimática OpenOffice, además de muchos otros programas, proporcionando así, simplicidad y facilidad en su uso, no dejando de lado la potencia y la flexibilidad de un sistema operativo Linux.

La distribución de paquetes para esta versión se realiza en archivos con extensión tar, pero también pueden utilizarse paquetes con extensión rpm. La instalación se lleva a cabo mediante una interfase de texto y por ende, se necesita un mayor conocimiento del sistema Linux que en la mayoría de las otras distribuciones, donde la interfaz de instalación es mucho más amigable, enfocada principalmente a una instalación por ambiente gráfico.

En el primer capitulo se describe el uso del sistema operativo Linux, su instalación y puesta en marcha con las distintas aplicaciones, la gestión masiva de información empleando bases de datos de manera local y, remota basada en Web. Aplicación de politicas de seguridad para conexiones remotas y manejo de datos, implementación de nuevas herramientas para facilitar el trabajo en el diseño de sistemas para su uso en Internet.

En el segundo capitulo se describe el desarrollo de un sistema para el control de inventario de equipo de cómputo para la planta productora de lácteos Mexilac Tizayuca, que forma parte de Grupo LALA, y que posteriormente fue implementado a nivel nacional controlado por cada ingeniero de zona.

## 1. Informe sobre el Diplomado “Desarrollo de sistemas con Software Libre”.

### 1.1 Sistema Operativo Linux

#### Objetivo General.

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

#### Objetivo específico.

El participante instalará el sistema operativo Linux en PCs y configurará la tarjeta de red, video y particiones del disco duro.

#### Informe Módulo I

En este primer módulo se trató la teoría relacionada con la funcionalidad del sistema, estudiando las principales características, así como de los requerimientos necesarios para trabajar con el mismo. Linux es un clon del sistema operativo UNIX, que corre sobre PC's basadas en procesadores Intel, AMD, además de que también opera sobre estaciones de trabajo con procesadores Sparc, Alpha y varios más. El 95% de los usuarios de Linux lo manejan exactamente igual que si estuvieran usando otro sistema Unix.

Linux no es una marca registrada y no tiene conexión alguna con la marca registrada Unix, que es una marca registrada de X/open.

Para comenzar a trabajar con Linux es necesario contar con un usuario y una contraseña que se identifican con el sistema, cada usuario tiene un registro de entrada, el cual requiere dos pasos: escribir un *login* (el nombre con el cual el sistema lo identifica), y escribir una *contraseña o password* (el cual es la llave secreta personal para poder registrarse a la cuenta), de esta manera se garantiza una entrada al sistema confiable debido a que nadie puede entrar registrándose al sistema bajo el mismo nombre de usuario si saber la contraseña.

Cada sistema Linux tiene un nombre (host name) asignado a este. Este nombre da a la máquina una identidad, define su carácter, clase y encanto. El nombre es usado para identificar al equipo en una red.

También se estudiaron los requerimientos para la instalación de Linux, en el cual se observó que para la instalación de la distribución Slackware se requería de contar con mucha más experiencia en el conocimiento de hardware, tales como la tarjeta madre (motherboard por su nombre en inglés), el CPU, conocer los distintos tipos de memoria con los que cuenta el sistema, con cuanta memoria se cuenta, las características del disco duro y conocimientos básicos de su manejo (particionamiento, formateo, preparación, tipo de sistema de archivos), monitor y tarjeta adaptadora de video (el conocer mejor este hardware permite una mejor configuración del entorno gráfico del sistema), tipo de ratón y teclado, controladores de lectores de discos flexibles y CDROM, interfaces o tarjetas de red, el conocer todo lo mencionado para identificar si el hardware con el que se contaba tenía soporte para la instalación del sistema en si. De igual manera se platico sobre el software libre con el que cuenta una instalación básica, personalizada y completa.

Durante este módulo se trabajó con una cuenta de usuario asignada en un servidor Linux del mismo centro a la cual nos conectábamos vía SSH (Secure Shell), un programa de libre distribución para clientes Windows que facilitan una conexión segura con un servidor Linux, posteriormente se acceso a un equipo con Linux instalado con el fin de conocer el entorno real del sistema.

Una vez conectados en el equipo se trabajó con una consola intérprete de comandos (prompt) llamada *shell* que es la interfaz de interacción entre el usuario y la máquina. El shell es utilizado para tres propósitos principales:

- Como intérprete de comandos. Para personalizar el ambiente de trabajo dentro de una sesión.
- Como lenguaje de programación. El trabajar en shell semeja un *prompt* de Ms-DOS pero con muchas más opciones.

Se comenzó a trabajar con comandos y utilerías básicas del sistema, se manejo el concepto básico de lo que es un comando y como es interpretado y ejecutado en Linux a través del intérprete de comandos, como se verá a continuación.

Al presentar ejemplos se utiliza la siguiente notación:

**\$ comando**

para indicar que el comando debe ser digitado en el "prompt" de BASH. Si se utiliza la notación:

**# comando**

Eso quiere decir que se debe digitar ese comando desde una cuenta privilegiada, es decir, root. Identificaremos por el símbolo que aparece antepuesto del comando #.

En general un comando puede recibir ninguna o más opciones:

**\$ comando [opciones]**

la notación [opciones] indica que no es necesario incluir esas opciones, es decir, que son exactamente eso: opcionales.

En cambio, si un parámetro es requerido, se indica de la forma siguiente:

**\$ comando {parámetro}**

Por ejemplo:

**\$ comando {nombre de archivo}**

Muchos comandos aceptan opciones que modifican el comportamiento del comando, es decir, son *switches* o *modificadores*. Estos switches se indican de la siguiente forma:

**\$ comando [-s]**

Eso quiere decir que el comando acepta el switch -s.

Los programas de GNU (y muchos otros también) aceptan también la notación:

**\$ comando [--switch]**

Los -- indican que a partir de ese punto el resto de los parámetros no son switches.

Esto es útil por ejemplo con los comandos que reciben nombres de archivos como parámetros, se utilizará el comando ls (list por sus siglas en inglés) para listar archivos:

```
$ ls
--un_archivo
otro_archivo
otro_mas
```

Utilizando un modificador

```
$ ls -l --un_archivo
ls: unrecognized option '--un_archivo'
Try `ls --help' for more information.
```

```
$ ls -l -- --un_archivo
-rw-r--r-- 1 usuario grupo 0 Jun  3 09:08 --un_archivo
```

A continuación se presenta una lista de comandos básicos en Linux, incluyendo los parámetros de uso más común. Si desea obtener más información acerca de un comando específico, refiérase a:

```
$ man {comando}
```

Es recomendable comenzar leyendo:

```
$ man man
```

*En la configuración por defecto puede abandonar el programa man digitando la letra q.*

Si desea *buscar* un comando que se refiere a una palabra clave particular, puede utilizar:

```
$ apropos mail
Mail (1)          - send and receive mail
exim (8)         - a Mail Transfer Agent
from (1)         - print names of those who have sent mail
procmail (1)     - autonomous mail processor
procmailrc (5)  - procmail rcfileç
```

Listar contenidos de un directorio: ls quiere decir "list".

```
Sinopsis: ls [opciones] [--] [archivos ...]
```

```
$ ls  
a.txt  
b.txt
```

Las opciones más utilizadas para **ls** son:

- a → Incluye archivos cuyo nombre comienza con '.' en la lista.
- l → Imprime información sobre el archivo, incluyendo dueño, grupo, tamaño y permisos
- h → Imprime la información del tamaño del archivo en un formato más fácilmente legible.

```
$ ls -alh  
total 2  
drwxr-xr-x 2 miguelmh users 96 May 25 23:13 .  
drwx--x--x 21 miguelmh users 1000 May 25 23:03 ..  
-rw-r--r-- 1 miguelmh users 0 May 25 23:13 .txt  
-rw-r--r-- 1 miguelmh users 0 May 25 23:13 a.txt  
-rw-r--r-- 1 miguelmh users 0 May 25 23:13 b.txt
```

La primera columna presenta los permisos del archivo; la segunda indica el número de enlaces o ligas que tiene el archivo; la tercera y cuarta indican el dueño y grupo al cual pertenece el archivo respectivamente; la quinta columna indica el tamaño del archivo en el disco; la información a continuación es la fecha (de modificación) del archivo seguida de su nombre.

Existe otro comando que nos permite ver espacio libre en disco y/o unidades que han sido 'montadas', se recurre al comando '**df**':

**Sinopsis: df [opciones] [archivos ...]**

```
$ df  
Filesystem 1K-blocks Used Available Use% Mounted on  
/dev/hdc1 586096 83564 502532 15% /  
/dev/hdc2 2929656 1307576 1622080 45% /usr  
/dev/hdc3 2929656 37376 2892280 2% /var  
/dev/hdc5 22362768 723228 21639540 4% /home
```

Esto muestra la información acerca del espacio disponible y utilizado de cada una de las particiones, así como el dispositivo montado y el punto de montaje dentro de la jerarquía de archivos del sistema. Entre las opciones más utilizadas de **df** están:

**-h** → Muestra la información de espacio utilizado y libre en unidades como KByte, MByte, GByte, etc.

**-m** → Muestra la misma información pero con bloques de 1 MB

por ejemplo:

```
$ df -hm /dev/hdc3
Filesystem 1M-blocks Used Available Use% Mounted on
/dev/hdc3   2861      37   2825     2%   /var
```

esta salida muestra la información acerca del dispositivo /dev/hdc3

Espacio utilizado por un archivo. El comando **du** reporta la cantidad de espacio de disco utilizado por archivos específicos. Sin argumento alguno, **du** reporta el espacio utilizado por el actual directorio.

### Sinopsis: **du [opciones] [archivos ...]**

Opciones más comunes:

**-h** → Despliega información en unidades de KByte, MByte, GByte, etc.

**-a** → Despliega la información de todos los archivos, no solamente los directorios.

**-s** → Suma el total para cada argumento.

Por ejemplo:

```
$ du -ah
1.0M ./a.txt
500K ./b.txt
1.5M .
```

Cantidad de memoria utilizada. El comando **free** despliega el total de memoria física y memoria swap libre y utilizada en el sistema, así como los buffers y caches consumidos por el kernel. Una invocación normal de **free**, no requiere ninguna opción.

```
$ free
      total        used          free   shared  buffers  cached
Mem:   256036    218412         37624      0    13344   151556
-/+ buffers/cache:    53512         202524
Swap:   493376           0         493376
```



Listado interactivo de procesos. **top** provee una mirada a la actividad del procesador en tiempo real. Despliega una lista de las tareas del CPU más intensas en el sistema. Las tareas pueden ser ordenadas por uso de CPU, memoria, y tiempo de ejecución (ver: man top). Para detener la ejecución de top, digite 'q'

Fecha del sistema. El comando **date** le permite al usuario ver la fecha y hora del sistema en un momento dado, para el administrador (root), date facilita también la configuración de la fecha y hora del sistema.

**Sinopsis: date [opcion] ... [+formato]**

Opciones más comunes:

**-s** → Permite al administrador fijar una hora y fecha específica de acuerdo a un formato de fecha establecido.

```
$ date  
Wed May 20 16:23:05 CST 2005
```

Usuarios actualmente utilizando el sistema: **who**.

El comando **who** muestra los usuarios conectados al sistema, la línea (puede ser terminal remota, o consola en el sistema) a la que se conectaron y la fecha y hora del inicio de sesión.

**Sinopsis: who [opcion] ... [archivo | argumentos ...]**

Opciones más comunes:

**-b** → Muestra la fecha del último inicio (booteo) de la máquina

**-a** → Muestra información detallada acerca de los usuarios conectados y otros.

**-l** → Resuelve las direcciones IP de terminales remotas a través de un DNS reverso.

```
$ who  
miguelmh tty1 March 11 15:34
```

### Cambio de directorio de trabajo: **cd**

El cambio de directorio se realiza con el comando '**cd**'.

<b>\$ cd</b>	#cambia al directorio padre.
<b>\$ cd</b>	#cambia al directorio de trabajo anterior
<b>\$ cd /ruta/a/directorio</b>	#cambia a la ruta especificada
<b>\$ cd</b>	#cambia al directorio home definido para el usuario

### Copia de archivos: **cp**

El comando '**cp**' ayuda para realizar la copia de archivos y/o directorios.

<b>Sinopsis: cp [opciones] archivo ... directorio cp [opciones] archivo path</b>
--

Opciones más comunes:

- f** → Remueve archivos existentes en el destino.
- r** → Copia recursivamente directorios y sus contenidos
- a** → Preserva, a medida de lo posible, la estructura y atributos de los archivos originales.
- d** → Copia links simbólicos en lugar de copiar los archivos a los que apunta. Este comportamiento lo ofrece la opción **-a** también.

Ejemplo:

```
$ cp -rf /home/miguelmh /tmp/backup
```

Copia recursivamente el directorio /home/miguelmh al directorio /tmp/backup.

### Renombrar y mover archivos: **mv**

El comando '**mv**' se utiliza para mover o renombrar un archivo o directorio. Si el último argumento es un directorio existente, entonces '**mv**' mueve los argumentos anteriores a este último directorio. Si solamente son dos argumentos y el último no existe como directorio, entonces cambia el nombre del primer argumento.

**Sinopsis: mv [opcion ...] fuente ... destino**

Opciones más comunes:

- f → Remueve un archivo o directorio destino existente sin informar al usuario.
- i → Hace una verificación en caso de que exista un destino.

Borrar archivos: **rm**

Para remover archivos y/o directorios del sistema se recurre al comando 'rm' por defecto no remueve directorios.

**Sinopsis: rm [opciones] archivos ...**

Opciones más comunes:

- r → Remueve recursivamente, permite remover directorios
- f → Ignora archivos no existentes sin preguntar al usuario
- i → Pregunta al usuario si desea remover dicho archivo

Ejemplo:

```
$ rm -i a.txt
rm: remove regular file `a.txt'? y
```

Crear directorios: **mkdir**

Para crear directorios puede utilizar el comando 'mkdir'.

**Sinopsis: mkdir [opciones] directorio ...**

Opciones más comunes:

- p → Crea el path específico aunque no exista
- m → Crea un directorio con los permisos específicos

Ejemplo:

```
$ mkdir -m 660 nuevo-dir
crea un directorio nuevo con permisos 660.

$ mkdir -p nuevo-dir/rama/de/directorios
```

Examinar archivos: **cat, more, less**

Hay diferentes lectores (visores) para archivos, entre ellos están **more** y **less** estos permiten navegar por todo el documento.

Utilizando **less**:

Movimiento

- j** → desplaza el cursor hacia abajo
- k** → desplaza el cursor hacia arriba
- l** → desplaza el cursor hacia la derecha
- h** → desplaza el cursor hacia la izquierda

Búsqueda

- /** → digite el caracter / y digite el texto a buscar, luego presione [ENTER]
- n** → dirige la búsqueda hacia abajo en el texto
- N** → dirige la búsqueda hacia arriba en el texto
- q** → termina la ejecución de less

Abriendo otros archivos

- e** → permite especificar otro archivo para leer

Los comandos **tail** y **head** permiten ver los encabezados (primeras 10 líneas) o finales (últimas 10 líneas) del archivo seleccionado. El mayor uso del comando tail es con la opción **-f**, esto permite 'monitorear' el archivo mientras cambia (usualmente un log).

## Sistema de archivos en Linux

Se estudio la estructura del sistema de archivos de Linux en comparativa con otros sistemas operativos y en caso explicito con UNIX, Linux esta estructurado de manera jerarquica arborescente, en donde el nodo principal es el directorio llamado raíz (representado por "/") y cada uno de los niveles del árbol representa directorios, y se presentan a continuación en la siguiente tabla:

<b>/</b>	Raíz del sistema
<b>/var</b>	Variables del sistema
<b>/var/spool/mail</b>	Archivos de correo electrónico
<b>/var/log</b>	Bitácoras del sistema
<b>/dev</b>	Dispositivos de hardware
<b>/etc</b>	Archivos de configuración del sistema
<b>/home</b>	Directorios de los usuarios
<b>/bin</b>	Comandos del sistema
<b>/sbin</b>	Comandos del sistema
<b>/boot</b>	Archivos de inicio del sistema
<b>/mnt</b>	Dispositivos montados

Tabla 1.1 Directorios comunes en Linux

Las características principales del sistema de archivos de Linux son las siguientes:

- Consistencia en el manejo de archivos
- Protección para los datos de los archivos
- Manejo de los dispositivos periféricos como archivos.

La estructura del sistema de archivos consiste en el bloque de boot, el super bloque, la lista de i-nodos y el bloque de datos.

El *bloque de boot* ocupa la parte del principio del sistema de archivo, por lo general el primer sector, y puede contener el código de boot o de arranque.

El *superbloque* describe el estado de un sistema de archivos. Contiene información acerca del tamaño de los sistemas de archivos, la lista de bloques libres disponibles, índice del siguiente bloque libre en la lista de los mismos, el tamaño, el total y la lista de los i-nodos, el índice del siguiente i-nodo libre, los campos de bloqueo de elementos de las listas de bloques libres y de i-nodos libres.

La *lista de i-nodos*, la cual se encuentra a continuación del superbloque. En esta lista, durante el proceso de arranque del sistema, el kernel lee la lista de inodos del disco y carga una copia en memoria, conocida como tabla de i-nodos.

La parte más importante de cualquier sistema de cómputo son los datos, es decir, la información que los programas almacenan y manipulan. Linux tiene un sistema de archivos cuyo trabajo es conservar toda la información que se almacene en la computadora, incluyendo programas, documentos, bases de datos, textos, etcétera.

En Linux el termino “archivo” se refiere a cualquier fuente de entrada o destino de salida, no solo a un deposito de datos. El espacio visible a los usuarios se basa en una estructura de árbol, con la raíz en lo alto. Los archivos y los directorios se colocan debajo de la raíz, como se mostró en la tabla 1.1.

En realidad, algunos de los directorios en el árbol de archivos están físicamente ubicados en diferentes particiones del disco, sobre diferentes discos y eventualmente en diferentes computadoras. Cuando una de estas particiones del disco esta ligado con el árbol de archivos en el directorio conocido como *punto de montaje* y a todos los directorios inferiores se referirán como el sistema de archivos.

También se habló sobre los tipos de archivos en Linux, y tiene siete tipos de archivos, se exponen 3 de ellos: *los ordinarios*, *los directorios* y *los especiales*. Los archivos ordinarios contienen datos y se almacenan en un disco o en un espacio

asignado por el usuario o por el administrador del sistema, pueden ser un programa, un archivo de texto, código fuente. Un directorio se almacena en disco o en un espacio asignado por el administrador del sistema pero este contiene información que se usa para organizar y permitir el acceso a otros archivos. Por último, los archivos especiales o bien *de dispositivo*, son la representación interna-lógica de un dispositivo físico.

En Linux se usa el término *montar* para determinar que se pondrá en funcionamiento un dispositivo de almacenamiento o lectura de datos como es una unidad de CDROM, una unidad de disco flexible o floppy, un dispositivo externo USB (memory stick, que son considerados por Linux como discos especiales tipo SCSI), particiones del disco duro o bien otros discos duros, estos están inactivos pero disponibles en el momento en que el usuario los pretenda utilizar.

El directorio por *asignación* (default) para montaje de dispositivos en Linux es /mnt, dentro de este se encuentran directorios como /mnt/floppy, /mnt/cdrom, /mnt/hdn donde n es la letra para identificar al disco duro en uso, y demás directorios para otros tipos de dispositivos de almacenamiento, en la práctica pudimos ver que se pueden crear directorios en raíz "/" para identificar y montar el dispositivo que se pretenda usar, más adelante se verán ejemplos explícitos de dicha acción.

Para poder montar el sistema de archivos en Linux, primero debemos tener una partición física en el disco, un espacio asignado previamente por el administrador del sistema, un CDROM, un floppy o bien puertos USB para conectar dispositivos extraíbles.

En Linux se usa el comando *mount* para montar el sistema de archivos, las formas más utilizadas en el uso de este comando son las siguientes:

```
#!/bin/mount [opciones] dispositivo punto-de-montaje  
#!/bin/mount punto-de-montaje  
#!/bin/mount -at nfs
```

Donde *opciones* son las banderas que soporta el comando mount, *dispositivo* es el nombre del archivo de dispositivos (device file) de modo bloque que se desea montar y *punto-de-montaje* es un subdirectorio que debe estar vacío, si este subdirectorio tiene alguna información, esta quedará oculta cuando se monte el sistema de archivos. El comando tiene varias banderas estas son:

- f → Termina todos los procesos excepto la actual llamada al sistema
- v → mount provee información adicional sobre lo que se trata de hacer.
- w → El sistema de archivos se montará con permisos de lectura y escritura.
- r → El sistema se montará con permisos de solo lectura.
- n → Se monta sin escribir las entradas en el archivo /etc/mtab.

**-t tipo** → Especifica el tipo de sistema de archivos a instalar. Los tipos válidos son: minix, ext, ext2, xiafs, msdos, hpfs, proc, nfs, umsdos, sysv e iso9660.

Hoy en día en muchas de las distribuciones de Linux ya no es necesario colocar esta bandera, ya que Linux reconoce automáticamente el sistema de archivos, basta con saber únicamente el tipo de dispositivo a montar.

**-a** → Intenta montar todo el sistema de archivos, descritos en `/etc/fstab`.

También se maneja la opción para hacer que los montajes se realicen automáticamente cuando el sistema entrará a modo multiusuario, solamente editando un archivo de configuración llamado `fstab` y que se encuentra dentro del directorio `/etc`.

El proceso contrario de montar es desmontar un sistema de archivos. Al igual que con el comando `mount` se tienen tres distintas formas muy usuales del comando `umount`, estas son:

```
#!/bin/umount dispositivo | punto-de-montaje  
#!/bin/umount -a  
#!/bin/umount -t fstipo
```

donde *dispositivo* es el nombre del dispositivo físico a desmontar y *punto-de-montaje* es el nombre del directorio donde fue montado. Solamente se necesita especificar una u otra opción. El comando tiene dos parámetros adicionales: `-a` desmonta todo el sistema de archivos, y `-t fs-tipo` actúa sólo en los sistemas de archivos especificado por `fs-tipo`.

De igual manera se estudió la posibilidad de añadir un nuevo disco duro a la computadora, cuando eso suceda, se tendrá que particionar el nuevo disco a fin de poder montar un sistema de archivos en él. La partición en Linux se realiza con el comando `fdisk`. Las formas usuales de manejo de este comando son:

```
#fdisk drive  
#fdisk /dev/hdb    donde hdb es el disco nuevo a instalar.
```

Una vez que se ha realizado una partición en el disco, se debe crear el sistema de archivos antes de proceder a ocupar el disco para comenzar a introducir datos. Esto se pudo realizar con el comando `mkfs`, aunque actualmente Linux cuenta con muchas herramientas de tipo gráfico que permiten al usuario o administrador del sistema realizar estas tareas sin complicaciones.

A continuación se verán ejemplos de lo antes descrito:

Para montar una unidad de CDROM desde línea de comandos, y que sea de solo lectura utilizamos el siguiente comando:

```
#!/bin/mount /dev/cdrom /mnt/cdrom
```

Para desmontar dicha unidad

```
#!/bin/umount /dev/cdrom o bien #/bin/umount /mnt/cdrom
```

Para montar una unidad extraíble tipo usb (memory stick). Creamos por comodidad un directorio a nivel / llamado /discusb y después montamos el dispositivo.

```
#mkdir /discusb  
#!/bin/mount /dev/sda1 /discusb
```

De esta manera queda montado el dispositivo y listo para usarse.

Para desmontarlo:

```
#!/bin/umount /dev/sda1 o bien #/bin/umount /discusb.
```

### **Sistema de arranque y paro, nociones de administración.**

Durante el primer módulo se manejaron también los conceptos de arranque y paro del sistema, el cual permitió conocer la metodología básica, acerca de la inicialización y detención del sistema Linux, así como conocer también los distintos niveles de ejecución con los que cuenta el sistema. En Linux existen varias formas de realizar el arranque.

Los métodos más comunes son a través del disco duro, o bien usando un floppy o un CDROM, en la práctica se maneja también la opción de los cds llamados LiveCD, que ejecutan el sistema sin que este tenga que ser instalado, trabajando totalmente sobre la memoria virtual de los equipos, en el módulo se estudió y analizó el caso de Knoppix.

Cuando se realiza el arranque con el disco duro, normalmente se está llamando a un administrador en especial llamada LILO (LInux LOader). Lilo es un administrador de arranque de propósito general. Al hacerlo de esta forma, el kernel de Linux es el encargado de verificar todo el sistema.

Existe un archivo y un programa que el kernel ejecuta al momento de inicializar los dispositivos. Uno es el archivo /sbin/init y el otro es el /etc/inittab. El



programa `/sbin/init` realiza nuevos procesos y restablece ciertos programas al momento de salir. Todo lo que `/sbin/init` realiza está controlado por el archivo `/etc/inittab`. En el archivo `/etc/inittab` se encuentran todos los parámetros de configuración de su sistema en particular que se tienen instalados.

Al igual que el arranque en el momento de querer detener el sistema se puede realizar de varias formas; la primera es apagar el sistema (aunque no es recomendable ya que puede causar daños a los archivos), la segunda y más recomendada es utilizar el comando que aquí veremos.

El comando que utilizaremos para la detención del sistema es: **`/sbin/shutdown`**, su sintaxis es la siguiente:

**`/sbin/shutdown [indicadores] tiempo [mensaje]`**

donde:

*mensaje* es un mensaje que se envía a todos los usuarios que están en sesión en ese momento *tiempo* es el momento en que va a suceder el paro, y se mide en segundos.

Para indicadores se tienen las siguientes opciones:

Bandera Descripción

- t *seg* Espera *seg* segundos para detener el proceso.
- k En realidad no detiene al sistema sólo envía el mensaje.
- r Vuelve a arrancar después del paro.
- h Una vez que se realizó el paro se detiene.
- c Cancela una detección que se encuentra en proceso.

Existen otros dos comandos en el sistema de Linux para realizar las tareas antes descritas, estos son: `halt` y `reboot`.

En el archivo `'/var/log/wtmp'` se guardan los cambios que se realizan con dichos comandos, y ambos, le dicen al kernel si parar o reiniciar el sistema.

Cuando las instrucciones `'halt'` o `'reboot'` son llamadas cuando el sistema no se encuentra en el nivel de corrida 0 ó 6, el comando `'shutdown'` se ejecuta en su lugar (con la bandera `-h` o `-r`).

La estructura de los comandos es la siguiente:

**`/sbin/halt [-n][-w][-d][-f]`  
`/sbin/reboot [-n][-w][-d][-f]`**

La descripción de las opciones se hace a continuación:

- n No existe sincronía antes del 'halt' o del 'reboot'
- w No hay actualización, pero se escribe un registro en el archivo '/var/log/wtmp'
- d No se escribe al archivo wtmp. Esta bandera implica -n
- f Forza el 'halt' o el 'reboot', no se hace una llamada a 'shutdown'

NOTA: Cuando se especifica la bandera -f se hace la llamada con la señal 9, mientras que en el otro caso se realiza la llamada con la señal 15 y en este caso también se llama a las funciones 'startup' y 'shutdown' respectivamente.

Se mostrará paso a paso un ejemplo de cómo arrancar Linux a través del disco duro:

1. Al encender el sistema se indique en el menú de arranque y selección de SO la opción de cargar Linux, mismo que se creó al realizar la instalación
2. En este momento se debe de seleccionar alguna de las etiquetas que tiene el menú de arranque de la computadora, seleccionando por supuesto Linux (los equipos utilizados para la capacitación tenían instalado Windows XP, Windows 2000 y Linux Slackware). Una vez seleccionado el sistema Linux presionamos la tecla enter, y el sistema arrancará inmediatamente.
3. En este momento se puede ir revisando lo que pasa en el sistema y que va apareciendo en el monitor.
4. Una vez que se han verificado todos los componentes del sistema, éste pasará del estado monousuario al estado multiusuario. En este estado el sistema esta listo para aceptar sesiones de trabajo. En la práctica entrábamos a sesión como super-usuario (root).
5. Ahora ya se puede utilizar el sistema para ejecutar los comandos que sean necesarios estudiar.
6. Ya dentro del sistema se procedió a trabajar con la parte del apagado del sistema, para esto se debe recordar que no es conveniente tan sólo apagar el sistema, ya que causaría problemas posteriores.
7. Por lo que para apagarlo, se abrirá una terminal o una consola y se tecleará el comando shutdown. El comando /sbin/shutdown permitirá detener el sistema de una forma conveniente.

Se tecllea el siguiente comando:

**# /sbin/shutdown -r now**

8. Una vez que se reinició el sistema se pudieron cambiar los parámetros del comando anterior y comprobar las diferentes formas de apagar el sistema.

Como se mencionó anteriormente existe un archivo de configuración en Linux que contiene información sobre los diferentes niveles en los que corre el sistema, y estos tienen una pequeña descripción que se verá a continuación:  
`/etc/inittab`

- 0) Paro total del sistema.
- 1) Modo monousuario ó administrativo. En este estado se trabaja para el chequeo de los dispositivos.
- 2) Mutiusuario sin *NFS*, donde no trabajan todos los programas en red.
- 3) Multiusuario completo.
- 4) No usado.
- 5) X11. El sistema X window se corre a este nivel.
- 6) Reboot. Donde primero se realiza el logout a los usuarios y después se da de baja el sistema.

Linux tiene un nivel de corrida que inicia el sistema por default la línea que se encuentra en el archivo `/etc/inittab`, es la siguiente:

**id:3:initdefault:**

donde:

id -> Identificador de la función a ejecutar.

3 -> Número de nivel en donde se ejecuta.

initdefault -> Lo que representa (en esta cadena tienen más significado, ver el páginas del manual para ver más detalles).

Cada campo está separado por ":". Se llevó a la práctica el cambiar de distintos niveles alterando este archivo de configuración o bien en modo línea de comando utilizando los comandos antes descritos.

Se realizó el cambio para que el sistema arrancará automáticamente en modo multiusuario con modo gráfico, es decir, con sistema gráfico de ventanas X-window (X11), solo se editó el archivo `inittab`, sustituyendo el valor de la línea original por el nuevo valor como se muestra a continuación:

**#pico /etc/inittab se abre el archivo con un editor de texto plano**

línea original **id:3:initdefault:** se sustituye el valor 3 de la línea original por el valor 5

línea modificada **id:5:initdefault:**

Se guardaron los cambios y se reinició el sistema para verificar dicha configuración.

Se realizaron otras pruebas una vez iniciado el sistema en modo multiusuario.

Se probó la entrada al sistema en modo mantenimiento (Nivel 1) entonces se tecleó:

```
#!/sbin/init 1  
bash# (así aparece el prompt indicando que el sistema esta en nivel monousuario)
```

Para regresar al nivel original, solo se tecleaba el siguiente comando:  
bash#!/sbin/init 3 de esta manera se regresa al nivel original, o bien también tecleando el comando exit.

Se notó que, se puede cambiar arbitrariamente a los diferentes niveles que tiene Linux.

Cuando se hacen estos cambios de niveles, se dan de baja unos servicios y se levantan otros. Pero, ¿Donde están esos archivos?

Estos archivos son llamados *scripts* de configuración y para los sistemas Linux funcionan para dar de alta y baja servicios. Estos *scripts* son similares en función al archivo *AUTOEXEC.BAT* de *DOS*, pero estos son muchos más flexibles y potentes.

Slackware, como otras distribuciones, un caso es Debian, guardan tradición con el Unix System V en cuanto a los directorios */etc/rc.d*.

Existe un directorio que contiene casi todos los scripts de una configuración básica del sistema, llamado *init.d*, y que se verán a continuación:

El *script rc* (run control) es responsable de poner en función variables básicas como el *Hostname*. Aquí también esta para iniciar servicios, cuando hacemos cambios de nivel.

El *script rc.local*, determina lo que aparece en el archivo */etc/issue* entre otras cosas tales como variables ó donde se pueden agregar variables personalizadas ó funciones.

El *script rc.sysinit* corre únicamente cuando el sistema esta iniciando; inicia los *scripts* general de red y activa la partición swap (intercambio).

Los directorios *rc0.d* al *rc6.d* contienen ligas que especifican *scripts* que se ejecutan cuando hay cambios de cada nivel.

Ahora se dará una breve descripción de los servicios que se encuentran dentro del directorio *init.d*

<b>atd.</b>	Inicia el demonio <i>at</i> , el cual ejecuta en un tiempo, comandos ó <i>scripts</i> .
<b>crond.</b>	Inicia el demonio <i>crontab</i> , el cual ejecuta tareas periódicamente.
<b>functions.</b>	Contiene funciones usadas por otros <i>scripts</i> .
<b>gpm.</b>	Inicia el programa <i>gpm</i> , habilitando el ratón desde una pantalla de texto.
<b>halt.</b>	Función para detener la computadora de reboot ó shutdown.
<b>inet.</b>	Inicia la red TCP/IP. Configura la interfaz ethernet, pone el ruteo funcionando y otros servicios.
<b>kerneld.</b>	Carga automáticamente los módulos del kernel cuando se necesiten.
<b>keytable.</b>	Mapea el teclado.
<b>killall.</b>	Para los demonios innecesarios.
<b>lpd.</b>	Inicia y detiene el demonio de impresión <i>lpd</i> .
<b>network.</b>	Para e inicia la red.
<b>nfsfs.</b>	Monta y desmonta los sistemas de archivos remotos (NFS).
<b>pcmcia.</b>	Inicia los adaptadores PCMCIA de las laptops.
<b>random.</b>	Inicia el número generado aleatoriamente.
<b>routed.</b>	Inicia el demonio <i>routed</i> , esto usa el protocolo RIP, automáticamente actualiza la tabla de ruteo.
<b>rusersd.</b>	El demonio <i>rusers</i> , ayuda a localizar usuarios en maquinas remotas.
<b>rwhod.</b>	El demonio <i>rwho</i> lista a los usuarios loggeados en una maquina remota.
<b>sendmail.</b>	Inicia y para el demonio <i>sendmail</i> , el cual transfiere mensajes e
<b>smb.</b>	Inicia y para los servicios <i>samba</i> .
<b>syslog.</b>	Inicia el sistema de <i>logging</i> . Esto es muy buena para la seguridad y otras funciones administrativas, ya que todos estos recursos tienen auditoría.

En el siguiente punto se describirán los procesos de instalación y administración del sistema operativo Linux utilizando herramientas administrativas de software libre.

## 1.2 Instalación y administración de Linux.

### Objetivo General.

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

### Objetivos específicos.

El participante instalará el sistema operativo Linux en PCs y configurará la tarjeta de red, video y particiones del disco duro; asimismo manejará el ambiente de usuario.

### Informe del módulo II.

Todo sistema Linux tiene un administrador que se encarga de todas aquellas tareas administrativas comunes, como tal es el responsable directo de configurar, mantener y actualizar el sistema o conjunto de sistemas que formen parte de una red, cuidando siempre el correcto funcionamiento del software, hardware y periféricos, garantizando que estén disponibles para ser utilizados por los usuarios.

Así pues la importancia de una buena administración proporcionará un ambiente seguro, eficiente y confiable para los usuarios que hagan uso del sistema.

Entre las tareas comunes de un buen administrador que se estudiaron durante este módulo fueron las siguientes:

- ✚ Configuración de dispositivos.
- ✚ Creación de respaldos.
- ✚ Capacitación de usuarios.
- ✚ Asegurar el sistema.
- ✚ Registro de los cambios del sistema.
- ✚ Administración de usuarios.
- ✚ Asesoramiento a los usuarios.
- ✚ Mantenimiento de claves de usuarios.
- ✚ Instalación y mantenimiento de dispositivos.

- ...Impresoras.
- Discos.
- Unidades de respaldo.
- ⊕ Instalación y actualización de software (comercial y dominio público).
- ⊕ Configuración de las interfaces de red.
- ⊕ Administración de los recursos (cpu, memoria y disco).
- ⊕ Atención a usuarios.
- ⊕ Monitoreo del sistema.
- ⊕ Detección de fallas.
- ⊕ Auditoria e implantación de la seguridad del sistema.

Los conocimientos deseados con los que debe contar todo administrador de sistemas son los siguientes:

- ⊕ Técnicas de programación.
- ⊕ Dominio de al menos un lenguaje de programación.
- ⊕ Funcionamiento del sistema operativo.
- ⊕ Técnicas de administración del sistema operativo.
- ⊕ Conocimientos básicos de hardware y mantenimiento de dispositivos.
- ⊕ Comprensión profunda sobre redirección, tuberías, procesamiento en segundo plano, etc.
- ⊕ Manejo de vi, pues es el común denominador entre los sistemas UNIX.
- ⊕ Programación shell.
- ⊕ Planear las actividades.
- ⊕ Absolutamente todas las actividades de administración se planean.
- ⊕ Guardar copias de seguridad.
  - ⊕ Jamás modificar archivo alguno sin respaldar previamente.

Durante el curso se aprendió a planear las actividades, ya que además de ser un requisito en los conocimientos que debe tener un administrador, todas las actividades de administración se planean sin excepción, en la práctica se realizó un cuadro de actividades administrativas que permitieron enfocar el potencial de Linux si este se maneja de forma apropiada.

El establecer políticas de uso del equipo, la apertura de cuentas por solicitud, las horas de mantenimiento al servidor, la responsabilidad de los respaldos por cada usuario, la eliminación de archivos temporales, cuotas de espacio en el disco para cada usuario, y la propia seguridad del sistema fueron algunos de los puntos más importantes que se manejaron en dicha tabla.

Se hacía hincapié de realizar respaldos de archivos de configuración antes de ser modificados y mantener estos mismos en un lugar seguro fuera del alcance de cualquier otro usuario que no fuera el administrador.

Al comenzar a estudiar las características del sistema operativo y su instalación en los equipos en los cuales se estaba trabajando se llegó a una conclusión muy importante, que implicaba el conocer más a detalle el hardware de la máquina en la que se planeaba instalar Linux, así como el conocer la documentación existente tanto de la distribución, como los manuales del sistema, con el fin de que al momento de realizar la instalación no se tuvieran complicaciones con cada uno de los pasos que exige el proceso de instalación, tales como particionamiento del disco duro, configuración básica de los dispositivos como el teclado, el ratón, el monitor, configuración de la tarjeta o interfaz de red, tarjeta de video, etc.

La versión o distribución instalada fue Slackware 10.0 y es una de las distribuciones que exigen al usuario un conocimiento profundo del hardware con el que cuenta, tales como, el modelo del equipo, sus características, tipo de procesador, tipo de tarjeta de red (chip integrado o tarjeta PCI), tarjeta de video (chip integrado, tarjeta PCI o AGP), capacidad y tipo de disco duro, unidad lectora de CD, tipo de ratón y teclado con el que se cuenta, características del monitor en cuanto a su frecuencia y rango de operación, así como sus capacidades de resolución en pantalla

En el proceso de análisis del equipo donde se instaló Slackware, se observó que esta distribución no requiere de sistema extremadamente potente para ejecutarse, ya que este puede ejecutarse en equipos 386 o superiores, los requerimientos mínimos para instalar y ejecutar esta versión son los siguientes:

- ✚ Procesador 386.
- ✚ 16MB en RAM.
- ✚ 50 megabytes de espacio libre en disco duro.
- ✚ Unidad de 3,5 ”.
- ✚ Hardware adicional tarjeta de vídeo para ejecutar X Windows y una tarjeta de red por si se desea tener acceso a algún tipo de red.

Slackware es una de las distribuciones de Linux más antiguas, es la primera distribución en forma que se logró del sistema, esta contiene un conjunto de paquetes que se reparten en series y comienzan desde la letra A a la Z, esto formo un estándar, era mas rápido conseguir las actualizaciones de Slackware de esta manera, ya que no era necesario descargar todo el software completo de una distribución con paquetes que no se iban a emplear, pero era mucho mas rápido y fácil descargar series de paquetes individuales.

Hoy en día las distribuciones de Linux son distribuidas en CD-ROM e incluyen todos los paquetes completos. Se pueden omitir algunos paquetes en la instalación por ejemplo, si no se desea trabajar con X Windows se puede omitir la serie de paquetes X, es aquí donde se puede realizar la selección de paquetes individuales que se quieran instalar en el equipo.



Las series están distribuidas de la siguiente manera:

- ⊕ A - es la base contiene suficiente software para levantar y ejecutar Slackware, contiene algunos editores de texto y programas de comunicaciones.
- ⊕ AP - varias aplicaciones que no requieren del sistema X Windows.
- ⊕ D - Herramientas de desarrollo de programas, Compiladores, depuradores, intérpretes, y los manuales man.
- ⊕ DES - Incluye la función de crypt() de libc de GNU.
- ⊕ E - Gnu Emacs, es tan grande que requiere su propia serie.
- ⊕ F - Contiene FAQs, HOWTOs, y otro tipo de documentación.
- ⊕ GTK - Contiene el ambiente de escritorio de GNOME, biblioteca de widget de GTK, y el GIMP.
- ⊕ K - El código de fuente para el núcleo de Linux.
- ⊕ KDE - Contiene el ambiente de trabajo de escritorio KDE.
- ⊕ N - Contiene programas para configuración de una red. Demonios, programas de correo, telnet, programas de lectura de noticias, etcétera.
- ⊕ T - Contiene el sistema de formato de documentos teTeX.
- ⊕ TCL - Contiene las herramientas del lenguaje de comandos, el Tk, el TclX, y el TkDesk.
- ⊕ U - Contiene paquetes de programas diseñados específicamente para trabajar solamente en sistemas de UltraSPARC.
- ⊕ X - Contiene la base del sistema X Window.
- ⊕ XAP - Contiene aplicaciones X que no son parte de un ambiente de escritorio importante. Por ejemplo Ghostscript y Netscape.
- ⊕ XD - Contiene Bibliotecas, kit de la conexión del servidor, y ayuda de PEX. Para el desarrollo de X11.
- ⊕ XV - Contiene Bibliotecas de XView, manejadores de ventanas aplicaciones de otras de XView.
- ⊕ Y - Contiene juegos (una colección de juegos de BSD).

Para comenzar con la instalación de Linux se accedió al BIOS (Basic Input Output System), para verificar los datos del equipo, sus características más comunes (cantidad de memoria RAM, tipo y tamaño de disco duro, y demás datos), así como para verificar que el modo de arranque (booteo) comience por la unidad de CDROM, ya que es de esta forma que se llevó a cabo la instalación. BIOS es un programa que se ejecuta oprimiendo una tecla especial antes de que el sistema operativo sea cargado y después del conteo de la memoria RAM. La forma de entrar a la BIOS es muy sencilla, solo hay que seguir los siguientes pasos:

- ✦ Reiniciar el equipo o si se encuentra apagado habrá que encenderlo.
- ✦ Se debe esperar a que el sistema termine de contar la cantidad de memoria RAM que esta instalada en el equipo, durante este proceso se muestra un mensaje con una leyenda que indica que se oprima la tecla F1, Supr, Del, F12, etc. para entrar al setup (BIOS), esta tecla dependerá del equipo con el que se cuente. Después aparecerá una pantalla azul (la mayoría de las veces) con un menú (por el cual se podrá mover a través de estas opciones por medio de las teclas de movimiento, una vez que se elige una opción se oprime la tecla Enter para entrar o Esc para salir de la opción).
- ✦ Se localiza la opción donde se pueda modificar la secuencia de arranque que por defecto suele ser C:, A: o A:,C: (la primera letra indica con que unidad arrancara el sistema si es desde un disco duro [C:] o una unidad de disco flexible [A:]).
- ✦ En ocasiones aparece una tercera opción que es la de arrancar desde un CD-ROM (CD-ROM,A:,C:) si se cuenta con esta opción se tendrá que habilitar.
- ✦ Se guardan los cambios realizados (F10 o Salvar los cambios y salir) antes de salir de la BIOS e inmediatamente después el sistema se reiniciará.
- ✦ Dependiendo de las características del equipo, se puede instalar Linux Slackware de diferentes formas:
  - ◆ CD-ROM  
Si el equipo cuenta con la opción de arranque por medio de un CD-ROM, esta sería la forma más fácil de instalar Linux Slackware.
  - ◆ Discos de arranque  
En caso de no contar con la opción de arranque desde un CD-ROM se necesitará crear los discos de arranque boot y root.
  - ◆ Disco de red. Esta opción se emplea cuando se necesita hacer la instalación de varios equipos, se emplea la red para este propósito y se puede hacer mediante NFS, FTP y HTTP.
  - ◆ Disco PCMCIA. Este método se emplea cuando se desea instalar Linux en una computadora portátil, este tipo de equipos cuentan con dispositivos PCMCIA.

Hechos los cambios en el programa de configuración (setup), se reinició el equipo y se insertó el disco 1 de instalación el cual es de auto arranque. Una vez elegido el método de instalación se procedió a instalar Linux. Aparece una pantalla de bienvenida al programa de instalación de Linux, en la parte inferior de esta pantalla aparece el indicador boot (figura 1)

Esta pantalla contiene información sobre algunas opciones de arranque, se eligió la opción más sencilla y solo se presiona la tecla enter de esta forma iniciara la instalación.

```
ISOLINUX 1.62 2001-04-24 Copyright (C) 1994-2001 H. Peter Anvin
Welcome to Slackware version 8.0.0 (Linux kernel 2.2.19 or 2.4.5)!

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,wpcom,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 hdh.

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: linux root=/dev/hda1 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 "scsi.s" or press [F1]
for a listing of more kernel choices.

boot:
```

Figura 1. Pantalla de Bienvenida del programa de instalación

En caso de que no se haya elegido cualquiera de estas opciones en un determinado tiempo (1 minuto), el programa de instalación se iniciara.

Cuando la instalación se realiza por CD-ROM automáticamente se cargan las imágenes de los discos boot y root. Después de arrancar la instalación Linux detecta la mayor parte del hardware que esta instalado en el equipo.

El primer paso de la instalación es elegir la configuración del teclado. Se debe responder la pregunta y elegir la opción es.map del menú, que es el mapa del teclado para la configuración en español. El termino mapa del teclado se refiere a la ubicación de las teclas en un teclado y dependiendo del idioma estas tendrán una ubicación diferente.

Enseguida aparecerá una pantalla donde se realiza una prueba al teclado escribiendo algo solo para probar la nueva configuración del teclado, se escribe 1 si esta de acuerdo con la configuración y 2 si se desea seleccionar alguna otra imagen.

```
Uniform CD-ROM driver Revision: 3.11
Floppy drive(s): fd0 is 1.44M
FDC 0 is a post-1991 82077
md driver 0.36.6 MAX_MD_DEU=4, MAX_REAL=8
linear personality registered
raid0 personality registered
scsi: <fdomain> Detection failed (no card)
NCR53c406a: no available ports found
scsi : 0 hosts.
scsi : detected total.
Partition check:
 hda: unknown partition table
RAMDISK: Compressed image found at block 0
VFS: Mounted root (minix filesystem).
Freeing unused kernel memory: 120k freed
init started: BusyBox v0.51 (2001.05.19-22:24+0000) multi-call binary
none on /proc type proc (rw)

<OPTION TO LOAD SUPPORT FOR NON-US KEYBOARD>

If you are not using a US keyboard, you may now load a different
keyboard map. To select a different keyboard map, please enter 1
now. To continue using the US map, just hit enter.

Enter 1 to select a keyboard map:
```

Figura 2. Pantalla de Selección de Teclado

El siguiente paso es ingresar al programa de instalación (setup) tecleando root cuando aparezca el siguiente mensaje.

### Slackware login: root

En seguida se obtiene un prompt donde se puede ejecutar una serie de comandos básicos para la instalación de Linux. La pantalla se muestra a continuación:

```
Welcome to the Slackware Linux bootable installation CD! (version 8.0.0)
##### IMPORTANT! READ THE INFORMATION BELOW CAREFULLY. #####
- You will need one or more partitions of type 'Linux native' prepared. It is
  also recommended that you create a swap partition (type 'Linux swap') prior
  to installation. For more information, run 'setup' and read the help file.
- If you're having problems that you think might be related to low memory (this
  is possible on machines with 16 or less megabytes of system memory), you can
  try activating a swap partition before you run setup. After making a swap
  partition (type 82) with cfdisk or fdisk, activate it like this:
  mkswap /dev/<partition> ; swapon /dev/<partition>
- Once you have prepared the disk partitions for Linux, type 'setup' to begin
  the installation process.
- If you do not have a color monitor, type: TERM=vt100
  before you start 'setup'.
You may now login as 'root'.
slackware login:
```

Figura 3. Pantalla de entrada al Sistema de configuración

Antes de instalar los paquetes se deben crear las particiones necesarias para la instalación. Se ejecuta el comando fdisk, para poder trabajar con las particiones. Existen diferentes programas con los cuales se pueden crear, borrar, modificar particiones en un disco duro. Algunos de estos programas son gratis y se pueden conseguir en Internet, otros son desarrolladas por empresas comerciales, pero casi todos los sistemas operativos traen un programa consigo lo bastante potente para trabajar con particiones en disco duro.

En Ms-Dos/Windows se llama FDISK, este programa solo puede trabajar con particiones de Ms-Dos/Windows. En Linux también se llama FDISK, pero es un programa más robusto, capaz de trabajar y crear particiones tanto para Linux como otros sistemas operativos.

La forma de arrancar fdisk es la siguiente:

### **fdisk /dev/hda**

Linux tiene su propia forma de nombrar a los diferentes dispositivos que se encuentran instalados en la computadora, como son: discos duros, discos flexibles, unidades de CD-ROM, etcétera. Los almacena dentro de un directorio especial llamado /dev. Es aquí donde se encuentran todos los dispositivos nombres de dispositivos que Linux puede manejar. Los discos duros en especial son nombrados de acuerdo a su interfaz de conexión y al número de particiones.

Al contrario de lo que sucede en Ms-Dos o Windows, en donde los dispositivos tales como discos flexibles, unidades de CD-ROM, discos duros y sus particiones se nombran con una letra del abecedario C:, D:, E:, ....., etcétera, sin importar el tipo de interfaz de conexión, esta es tal vez la forma en como los usuarios reconocen su disco duro.

Enseguida se muestra una tabla comparativa entre Windows y Linux, y la forma en como reconoce cada sistema los discos duros y sus particiones:

Linux

- + /dev/hda - disco duro ide primario (maestro).
- + /dev/sda - disco duro scsi primario (maestro).

Windows

- + C: - Disco duro ide primario (maestro).

Linux

- + /dev/hdb - disco duro ide secundario (esclavo).
- + /dev/sdb - disco duro scsi secundario (esclavo).

Windows

- + D: - Disco duro ide secundario (esclavo).

#### Linux

- + /dev/hda1 - disco duro ide primera partición (maestro).
- + /dev/sda1 - disco duro scsi primera particion (maestro).

#### Windows

- + C:,D:,E:,F:,etcétera. - Disco duro ide primera partición (maestro).

En Linux es habitual tener más de una partición, el área de swap y el sistema de archivos linux, esto es la mayoría de los casos. Muchas personas instalan Windows y Linux en el mismo disco duro, en particiones independientes lo que permite flexibilidad y compatibilidad unidireccional.

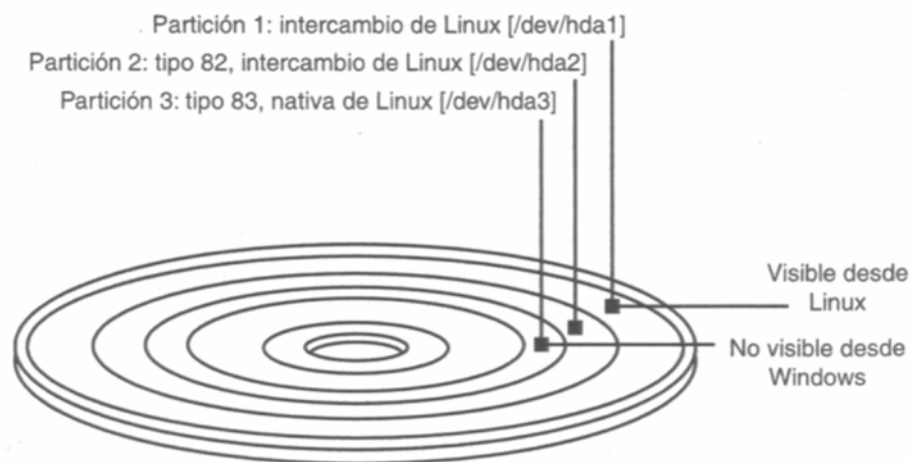


Figura 4. Tipos de Particiones del Disco Duro

Muchos usuarios de Linux se sienten satisfechos si pueden finalizar la instalación sin problemas, por lo que son propensos a evitar esquemas de particiones mas complicados.

Existen pocas rutinas de instalación que resalte la relación entre particiones y seguridad. No indican que dichas configuraciones conllevan ciertos riesgos. No se debe colocar el sistema de archivos raíz y de usuario en la misma partición Linux, pues esto aumenta la posibilidad de que las personas que deseen realizar ataques puedan explotar los programas SUID para acceder a áreas restringidas, además de que tener en una misma partición todo el Linux dificulta la tarea de los administradores de sistemas y puede dificultar la capacidad para actualizar o hacer copias de seguridad.

Incluso algunos archivos dañados pueden provocar problemas, lo que significa que la jerarquía de un directorio dañado puede afectar a los restantes,

incluso puede obligar a reinstalar Linux. Para evitar estos problemas, se debe crear una partición independiente para cada uno de los sistemas principales de archivos. Dicha configuración mejora la seguridad y permite gestionar las copias de seguridad y su posterior recuperación.

La existencia de varias particiones ofrece varias ventajas por ejemplo:

- Una sencilla gestión de las copias de seguridad y actualizaciones.
- Arranque más rápido (en algunos casos).
- La capacidad para controlar como se monta cada sistema de archivos.
- Protección contra programas SUID renegados.
- Otra ventaja más es que evita la denegación de servicio accidental y protege de desbordamientos al sistema de archivos raíz.

Como se ha indicado, los usuarios nuevos tienden a huir de la creación de varias particiones, debido a que se debe tomar algunas decisiones importantes, como por ejemplo el tamaño que debe tener cada una de las particiones y donde serán montadas cada una de ellas. El termino montar hace referencia a la forma en que linux permite utilizar los distintos sistemas de archivos. Cuando linux monta un sistema de archivos local o externo, conecta el sistema a un dispositivo o directorio local, lo que permite un punto de acceso o punto de montaje.

A excepción de las particiones de raíz y swap, el tamaño y el punto de montaje de algunas particiones dependen de distintos factores. Uno de esos factores es la función que va a tener Linux. Por ejemplo, si va a haber muchos usuarios, o si va a proporcionar algún tipo de servicio como correo electrónico, paginas Web, etcétera.

Algo que puede ayudar es conocer las funciones de los distintos sistemas de archivos. Dichas funciones son:

- /: contiene relativamente pocos archivos (sobre todo scripts de inicio).
- /usr: contiene la mayoría del software.
- /home: contiene los directorios de los usuarios./opt: contiene el software complementario de terceros (Netscape, StarOffice, etcétera.).
- /var: contiene registros administrativos, correo y noticias.

Los parámetros de las particiones anteriores solo se han utilizado para facilitar las explicaciones. Es posible solo trabajar con tres particiones, sobre todo si solo van a acceder al sistema linux unos pocos usuarios.

Estos son algunos consejos importantes:

Los sistemas de archivos importantes que deben estar en particiones independientes son la raíz (/), /var y /tmp desde el punto de vista de la seguridad,

o la raíz (/), var y /usr desde el punto de vista administrativo. Si se asignan particiones a sistemas operativos que no sean Linux, se debe analizar detenidamente como se desea que Linux las monte. Cuando se definen puntos de montaje de las particiones. A veces las políticas restrictivas pueden causar problemas administrativos. En la mayoría de los casos las particiones de mayor tamaño albergarán los directorios /home y /usr.

A continuación se verá un ejemplo.

Filesystem	1M-blocks	Used	Available	Use%	Mounted on
/dev/hda1	3396	2010	1211	63%	/
/dev/hda5	1937	885	952	49%	/home
/dev/hda6	972	216	706	24%	/var/spool/mail
/dev/hda7	486	9	453	2%	/home/users
/dev/hda8	296	9	272	3%	/var/spool/mail/users

Como se ha indicado para evitar problemas de seguridad y facilitar el trabajo del administrador, se crearán particiones independientes para los sistemas de archivos principales del sistema. A continuación se muestra el esquema de particionamiento propuesto para este curso.

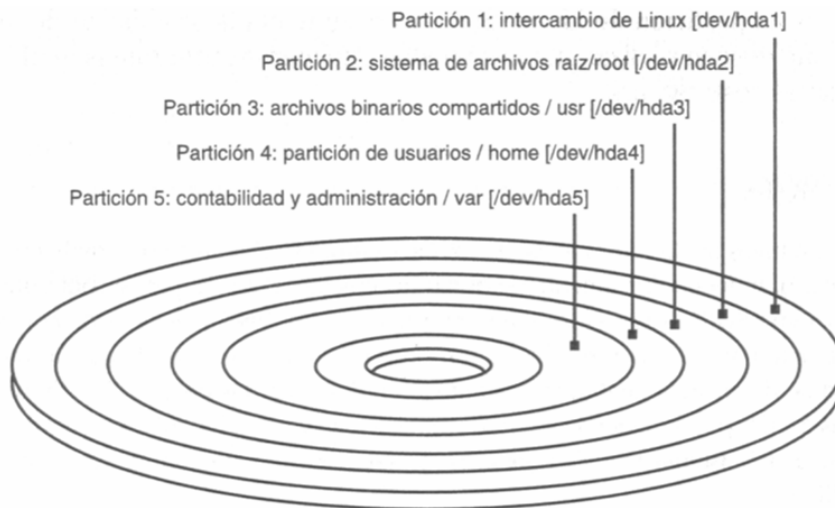


Figura 5: Particionamiento del Disco Duro

Se utilizó el comando fdisk para preparar el disco duro, para lanzar fdisk se teclea lo siguiente:

**fdisk /dev/hda**



Para saber cuales son las opciones que tiene fdisk para crear particiones se debe teclear la letra m y así aparecerán todas las opciones. Las opciones más importantes de fdisk son:

- p - despliega la tabla de particiones actual.
- m - despliega en la pantalla la ayuda.
- d - elimina una partición.
- n - crear una nueva partición.
- t - cambia el sistema de identificación de la partición.
- q - sale de fdisk sin guardar los cambios.
- w - escribe los cambios en el disco y sale de fdisk.

Antes de comenzar a trabajar con fdisk necesita saber cuantas particiones existen en el disco duro y cuanto espacio están ocupando, se teclea p para ver la tabla de particiones.

**fdisk** pregunta si se quiere crear una partición primaria o extendida, se debe tener como precaución que todas las particiones en donde se va a instalar algún tipo de sistema operativo siempre deben ser primarias, así es que se teclea p para una partición primaria.

**fdisk** preguntará por el numero de partición (1- 4) depende de cuantas particiones se tenga, enseguida pedirá el primer cilindro para la nueva partición, solo se presiona enter y preguntara por el ultimo cilindro, el cual se puede especificar en megas de la siguiente manera: +tamañoM.

Donde tamaño es la cantidad en megas de la partición por ejemplo:

- +2000M - partición de 2Gb.
- +3500M - partición de 3.5Gb.

De acuerdo con el tamaño de la partición se debe indicar de este modo. Si ya se cuenta con la primera partición nativa de Linux donde estará instalado el sistema en si, se deberá crear la partición swap. La forma de crear esta partición es la misma. Cuando se hayan creado las dos particiones aparecerán como Linux nativa, y se debe de cambiar el tipo de la segunda partición que es la partición swap.

Para realizar dicho cambio se oprime la letra t para cambiar el identificador, se indica el numero de la partición (1-4) que se va a modificar y se selecciona el tipo de identificador de una lista oprimiendo la tecla l, se elige el identificador, que es el 82 se teclea y presiona enter. Ahora ya se tienen las dos particiones, se guardan los cambios, con la tecla w y se continua con la instalación.

Para arrancar el programa de instalación se teclea la palabra setup y aparecerá una pantalla como la siguiente:

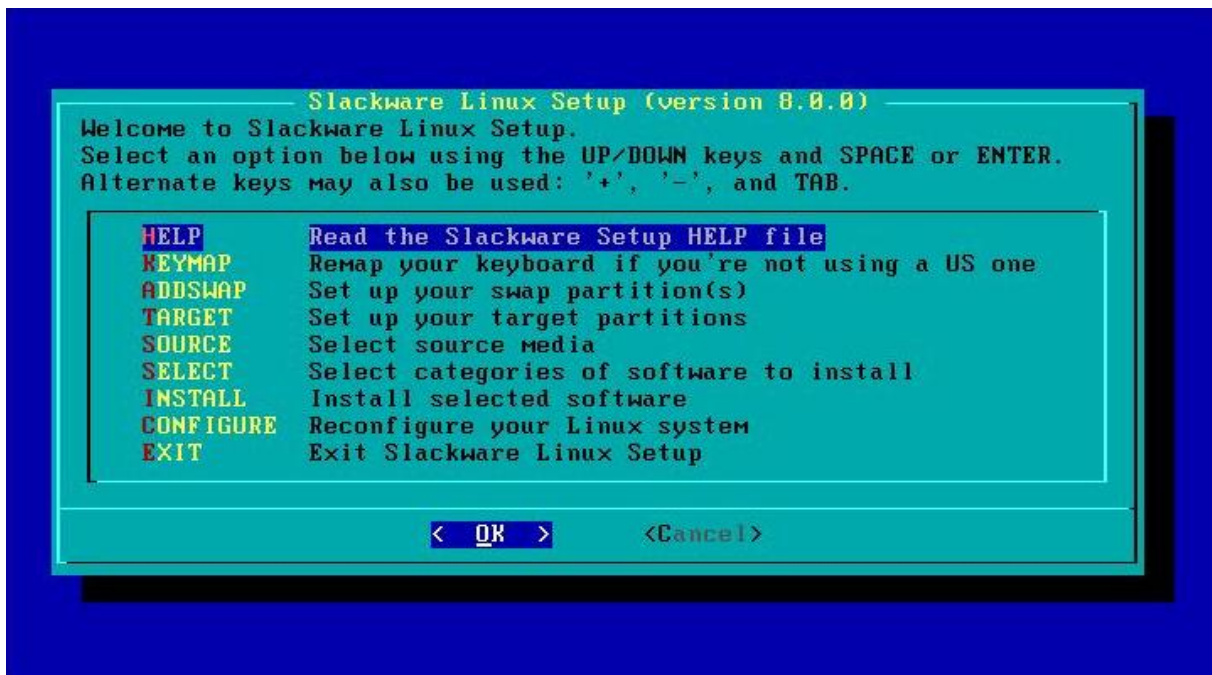


Figura 6. Menú de Configuración

Se puede mover a través del menú con las teclas de navegación y con la tecla Tab. Se selecciona la tercera opción add swap, el programa detectara la partición swap y preguntara si se quiere activar y dar formato, se elige la opción OK y presiona la tecla enter.

El programa de instalación tiene una interfaz muy intuitiva que facilita completamente el proceso y que brinda las facilidades necesarias para proceso. Cuando el programa de instalación detecta particiones que contienen otro sistema operativo, solicita un punto de montaje para que sea accesible desde Linux, solo se debe indicar la ruta en donde estará montada la partición y el nombre del directorio.

Ejemplificando: /windows o /winXP o /win2000.

Hay un paso en el proceso de instalación que muestra la opción instalar lilo este se alojara en el Sector Maestro de Arranque (por sus siglas en inglés MBR Master Boot Record) del disco duro.

Al terminar la instalación se debe reiniciar el equipo, se selecciona EXIT del menú y se oprimen las teclas Control+Alt+Supr.

Cuando el sistema arranque nuevamente se selecciona Linux del menú de arranque, y comenzará a cargar el sistema, una vez cargado, se presentará una pantalla de bienvenida donde se tiene que acceder con el usuario root y su contraseña previamente establecida para conseguir una terminal y poder comenzar a trabajar con Linux.

## **Administración de usuarios**

Una vez que ya se tuvo instalado Linux, se probó que funcionará correctamente y se revisaron algunos parámetros básicos de configuración, como la configuración del equipo en red se comenzó a trabajar con las tareas básicas de todo administrador, una de ellas es la *Administración de usuarios*, cuyo objetivo logrado en el curso fue aprender las diversas herramientas y técnicas que pueden ser empleadas para la administración de cuentas de usuarios.

Como administradores de sistemas Linux se aprendió a registrar usuarios, ajustar sus privilegios, creación y asignación de directorios personales, asignación de usuarios a grupos, asignación de cuota por cuenta de usuario y la eliminación de usuarios en caso necesario. Todo el poder administrativo que se tiene en Linux es otorgado a root también conocido como super usuario. Este controla a todos los usuarios individuales a los grupos y a todos los archivos.

En Linux como en otros sistemas operativos se tienen cuentas de usuario para poder hacer uso del equipo, una cuenta de usuario en Linux consta de dos elementos básicos: un nombre de usuario (login) y una contraseña (password) válidos para el sistema, ya dentro del sistema se tendrá un directorio de trabajo inicial siempre asignado por el administrador que le proporcionará al usuario un acceso al shell de Linux, de esta manera, cuando un usuario intenta iniciar una sesión de trabajo en un equipo con Linux instalado, ya sea esta conexión local (conexión al sistema de manera física, la conexión se hace desde consola) o remota (conexión al sistema a través de un software y de manera distante, conectándose a través de una red, la conexión se realiza desde una emulación de terminal o una conexión remota), Linux comprueba si se cumplen los requisitos mínimos de conexión, es decir, verifica la existencia de la cuenta y su contraseña respectivamente, haciendo totalmente seguro al sistema.

Dicha comprobación la hace revisando el contenido del archivo passwd que es un archivo de configuración del sistema y se encuentra ubicado dentro del directorio /etc/passwd.

Cada línea contenida en este archivo almacena el registro de una cuenta de usuario válido en el sistema, donde cada registro está compuesto de siete campos, y estos campos están delimitados por columnas.

Las tareas que se llevan a cabo para generar una cuenta de usuario son:

- Se edita el archivo `/etc/passwd`
- Se crea un directorio de casa para la cuenta (HOME DIRECTORY).
- Se edita el archivo `/etc/group`
- Se copian los archivos de configuración.
- Se asigna una contraseña (password).
- Se asignan recursos.

El archivo `/etc/passwd` es el archivo de registro de los usuarios válidos en el sistema, la información contenida es la siguiente:

- Nombre de la cuenta de usuario
- Password cifrado. Si la versión de Linux usa el archivo `/etc/shadow` tendrá una x en este campo.
- Identificador de usuario (userID o UID – User Identification)
- Identificador de grupo de default (groupID o GID – Group Identification)
- Información particular de la cuenta (GECOS - General Electric Comprehensive Operating System – Real Name)
- Directorio de casa (HOME DIRECTORY)
- Shell de inicio.(shell predeterminado)

Dependiendo de la versión de Linux se tiene que editar el archivo `/etc/shadow`, hoy en día todos los sistemas modernos cuentan con este archivo para almacenamiento de contraseñas cifradas. Ya que muchas de las distribuciones de Linux almacenan las contraseñas de los usuarios en `/etc/passwd`, lo que no resulta seguro, ya que `/etc/passwd` es y debe ser legible. De ahí que cualquier usuario pueda ver el contenido de `/etc/passwd`. Las contraseñas Linux están encriptadas, de esta manera aunque `/etc/passwd` es público la contraseña es ilegible para cualquier usuario curioso. `/etc/shadow` es un archivo especial que almacena no solo las contraseñas de los usuarios sino también de indicadores de reglas especiales. Desde varios puntos de vista, `/etc/shadow` se asemeja a `/etc/passwd`. El archivo consta de un registro por línea y cada registro se divide en nueve campos separados por dos puntos (:), los cuales serán nombrados a continuación:

- El nombre de usuario.
- La contraseña de usuario.
- El número de días desde el 1 de enero de 1970, fecha en que se cambió la contraseña por última vez.
- El número de días que quedan antes de que se permita al usuario cambiar su contraseña.
- El número de días que quedan antes de que el usuario tenga que cambiar su contraseña.

- El número de días de anticipación con que se avisa al usuario de que pronto tendrá que cambiar su contraseña.
- El número de días que quedan para que el usuario cambie su contraseña antes de que su cuenta sea cancelada.
- El número de días desde el 1 de enero de 1970 que la cuenta ha sido cancelada.
- El último campo esta reservado.

Como se comentó anteriormente una cuenta en Linux consta de un nombre de usuario o login y una contraseña o password, para cambiar o asignar el password a determinada cuenta se emplea el comando passwd.

El comando passwd cambia la clave de acceso o contraseña, al ejecutar este comando (sin parámetros ni opciones), el sistema pregunta por el password anterior, esta pregunta no se hace si el usuario no posee password alguno, luego el sistema solicita el nuevo password de acceso dos veces (para evitar errores de tecleo). Si las dos claves tecleadas son iguales, entonces el sistema cambia la clave de acceso:

```
$passwd
passwd: Changing password for miguelmh
Enter login password:
New Password:
Re-enter new password:
passwd (SYSTEM): passwd successfully changed for miguelmh
$
```

En la práctica se llevo a cabo la elaboración de password o claves de acceso que cumplieran con ciertos requisitos:

- Por lo menos 6 caracteres. Solo los primeros 8 caracteres son significativos.
- Por lo menos dos caracteres alfanuméricos y al menos un carácter numérico o especial.
- Las claves de acceso deben contener letras mayúsculas y minúsculas.
- La clave de acceso debe diferir del nombre del usuario (login name).

El password es cifrado y puesto en el archivo /etc/shadow, que puede ser leído solamente por el super usuario o root. Las claves de acceso en algunos sistemas tienen un tiempo de vida limitado, después del cual el sistema mismo solicita el cambio de la contraseña, esto con el fin de mantener un nivel de seguridad optimo dentro del sistema. Ya que a menudo por pereza de los usuarios se crean claves de acceso demasiado comunes y muy débiles, como para sufrir un ataque malintencionado y que su contraseña sea descubierta, utilizando parámetros y palabras demasiado sencillas y fáciles de descifrar, como, su fecha

de nacimiento, nombres de sus amigos o familiares, nombres de sus artistas favoritos, nombres de sus mascotas, alguna palabra del diccionario, secuencias numéricas o del teclado, nombres escritos al revés.

Se realizaron algunas prácticas de asignar contraseñas débiles para después descifrarlas por medio de un programa llamado Jack The Ripper o Jhonny The Ripper, así como el manejo correcto de permisos en los archivos y directorios y la búsqueda de archivos vulnerables, en Linux dichos archivos tienen permisos especiales y los hay de dos tipos:

- ✦ SGID (define el ID de grupo, 2000 octal).
- ✦ SUID (define el ID de usuario, 4000 octal).

Los programas con permisos de SGID y SUID son especiales, ya que los permisos de su propietario se respetan aun cuando los ejecuten otros usuarios.

Esto es si se define el valor root SUID en un programa, éste siempre se ejecutará como root, aunque lo utilice un usuario normal.

Los archivos SUID se pueden buscar con el siguiente comando:

**find / -perm +4000**

Lo cual fue muy ilustrativo y se consiguió comprender el objetivo de este punto.

En Linux existen varias formas de añadir usuarios al sistema, pueden utilizarse herramientas gráficas, herramientas de línea de comandos, editando manualmente el archivo `/etc/passwd`, y en caso de eliminar usuarios se pueden emplear los mismos métodos.

En la práctica, esta tarea se llevó a cabo mediante el comando `useradd` o `adduser`, que de manera semiautomática realiza paso a paso la creación de cuentas de usuario. Aunque también se manejaron algunas utilidades gráficas como `kuser` para agregar y administrar cuentas de usuario y los grupos existentes en el sistema, así como la creación de cuentas de usuario para uso especial como para la administración de algún proceso o aplicación o bien cuentas de usuario que tenían la función de la ejecución de comandos “de una sola vez”. Que posteriormente servirían de base para comprender los requerimientos de algunas aplicaciones que se manejaron durante el diplomado (Apache, MySQL, PostgreSQL, etc.), y que requerían de un usuario en específico con características especiales para la ejecución de sus procesos.

Se analizó la interfaz gráfica de Linux y se realizaron practicas con el fin de comparar otros entornos de trabajo gráfico como el caso de Windows, UNIX o Apple, llegando a la conclusión de que la nueva interfaz gráfica de Linux tiene

muchas bondades en cuanto a accesibilidad con el usuario, además de que es 100% configurable consumiendo un mínimo de recursos a diferencia de otros sistemas como Windows. El escritorio con el cual se trabajó fue KDE (K Desktop Environment) es un entorno de escritorio gráfico e infraestructura de desarrollo para sistemas Unix y en particular Linux. La 'K' originariamente representaba la palabra "Kool", pero su significado fue abandonado más tarde. Actualmente significa simplemente 'K', la letra inmediatamente anterior a la 'L' (inicial de Linux) en el alfabeto, el cual al iniciarlo por primera vez se ejecutaba un asistente de configuración del entorno de trabajo personalizado para cada usuario. Se probaron y configuraron los otros entornos gráficos con lo que cuenta Linux como es el caso de GNOME o Xfish, Blackbox solo para verificar diferencias y similitudes entre todos ellos.

Se observó que KDE cumple con la finalidad de llenar la necesidad de un entorno gráfico en estaciones de trabajo, es muy parecido a los entornos gráficos Mac OS y Windows, y cubre la necesidad de facilitar el manejo de la computadora a los usuarios. Para esto se tiene una mejor comunicación entre aplicaciones, reutilización de componentes, usos generalizados del ratón. De esta manera KDE ofrece mucho más que los tradicionales ambientes gráficos para UNIX y Linux.

Entre las múltiples actividades de un usuario de sistemas Linux esta la de crear documentos para el WEB, Linux tiene una amplia gama de aplicaciones para la creación de los mismos, que es el punto de partida que se verá a continuación.

### **1.3 Editores para la creación de páginas Web.**

#### **Objetivo General**

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

#### **Objetivo Especifico**

Crear documentos HTML, conocer y desarrollar DTD (Definición de Tipo de Datos) y documentos XML válidos, para el desarrollo de aplicaciones Web.

#### **Informe del módulo III**

Durante este módulo se estudió la manera de mostrar información en Web mediante páginas y haciendo uso del lenguaje HTML (Lenguaje Etiquetado de Hipertexto – Hypertext Markup Language), partiendo de una base de conocimiento del lenguaje, sintaxis, etiquetas básicas, esqueleto del documento, características de las etiquetas, editores (en el curso se estudio el editor Quanta).

HTML o Hypertext Markup Language. La traducción que se le da al español es "Lenguaje Etiquetado de Hipertexto". Probablemente no es la traducción exacta pero es muy significativa. A continuación se detalla cada uno de estos conceptos.

#### **Hipertexto (HyperText)**

Navegar en Internet es como navegar en el océano, si no se lleva consigo una buena brújula o si no se sabe el lugar al que se quiere ir, lo más seguro es que se perderá en este océano digital. Al seguir un vínculo, se hace para buscar cierta información, ver una imagen o reproducir un sonido o video. Esta capacidad de ir uniendo páginas con otras páginas es lo que le da a Internet su dinámismo. Como se sabe, la información es la esencia de Internet, o mejor dicho, la transferencia de esa información. En si, el concepto del Hipertexto fue uno de los más importantes detonantes de lo que ahora se conoce como World Wide Web que podría ser definido llanamente como un conjunto casi infinito de vínculos.



Una definición más general que se podría dar de Hipertexto es la siguiente:

*"Es la organización de las unidades de información interconectadas entre si"*

Supóngase por ejemplo una página o sitio, con varios vínculos incrustados en ella. La lectura de dicha página se puede hacer de manera lineal, olvidando que los vínculos existen, o podemos llevar una lectura no lineal siguiendo cada uno de los vínculos. Así la definición podría ser de la siguiente forma:

*"Es la organización no lineal de las unidades de información interconectadas entre si"*

Si se quiere ser purista, se podría decir que el hipertexto solo se refiere a texto y al hablar de vínculos a objetos multimedia se podría hablar de hipermedia.

Una nota final, el termino hipertexto fue usado por primera vez por Ted Nelson, inventor de Xanadu, un antecesor del Web.

## **Etiquetado (Markup)**

Imagínese esta escena: está sentado frente a su computadora escribiendo un documento en su procesador de texto favorito. Conforme va escribiendo va decidiendo el tamaño de las letras de los títulos y del texto normal, también define colores, tipos de letras o crea tablas o incluso inserta un dibujo ó gráfica.

Ahora imagine esta escena: Va en el transporte colectivo tratando de escribir ese mismo documento y su procesador de texto es papel y lápiz. Así que, si quiere indicar la forma en que se verá el documento, tendrá que anotar las características de ese título o de esa imagen que debe ir aquí y centrada. La mejor forma sería inventar una forma de escribir esas características como un ejemplo a continuación:

<b>&lt;P&gt;</b>	para indicar un párrafo nuevo
<b>&lt;B&gt;</b>	para indicar negritas
<b>&lt;H1&gt;</b>	para indicar un titulo
<b>&lt;IMG&gt;</b>	para indicar que aquí va una imagen

Estas, en si, son etiquetas utilizadas en HTML para indicar en donde van las cosas y cuales son sus características.

Así que las etiquetas son solo cadenas de caracteres especiales que se insertan en el documento para indicarle al navegador como se vera o imprimirá una página en particular.

## Lenguaje (Language)

En si HTML no es un lenguaje de programación; es solo un Lenguaje Etiquetado de Hipertexto.

Sin embargo se pueden encontrar algunas analogías como por ejemplo decir que las etiquetas son palabras reservadas, etc. Es mejor tomar a HTML como un formato que como un lenguaje de programación.

Por ultimo, cabe mencionar que HTML no es el único lenguaje, o formato, que maneja etiquetas, por ejemplo, también existe el XML.

Para trabajar con HTML se necesita disponer de algunas herramientas, en si lo único que se necesita es un simple editor de textos como Pico o Vi, o en el caso de estudio que fue Quanta, que es un editor de documentos HTML que resalta el código del texto simple y de esta manera facilita el trabajo, además de que es un editor del tipo WYSIWYG (What You See Is What You Get – Lo Que Ve Es Lo Que Obtiene), es decir un editor que muestra en una ventana del mismo como va quedando el documento en cuanto a forma y armado del mismo, la única característica a tomar en cuenta sobre el editor con el que se pretende trabajar es que respete el formato ASCII.

Existen editores que crean documentos Web de manera sencilla y muy vistosos, pero el objetivo de este módulo es aprender el código fuente y comprender todas las características adicionales con las que cuenta el lenguaje desde etiqueta y no desde software.

Además del editor también se necesito un navegador Web, Firefox, Mozilla, Netscape, Konqueror, para poder visualizar los documentos creados.

Las principales ventajas del servicio WWW son tres.

- Primera, que puede combinar texto, gráficos y multimedia.
- Segunda, que los vínculos permiten cargar páginas de cualquier otro servidor conectado a Internet, es decir, da igual que esté localizado en México o en Australia.
- Tercera, que la creación de páginas WWW es bastante sencilla mediante el *lenguaje HTML*

El gran éxito del Web no se debe solamente al empleo de *hipertexto*.

Es normal encontrar que los documentos WWW están compuestos por texto, gráficos.

Los vínculos con otros documentos pueden ser palabras subrayadas o resaltadas, pero también una imagen o incluso partes de ella puede ser un vínculo.

Aunado a esto los navegadores actuales permiten integrar en un mismo documento además de texto y gráficos: sonidos, animaciones e incluso vídeo.

Cuando se crea un nuevo documento HTML, este debe de ser guardado con la extensión .html o .htm que es la manera en que son reconocidos por el navegador estos documentos.

La sintaxis de HTML es sencilla, la forma de escribir una etiqueta básica es **<ETIQUETA>**; además la gran mayoría de estas se deben "cerrar" de esta manera: **</ETIQUETA>**. Hay dos tipos de etiquetas, abiertas y cerradas, las etiquetas cerradas indican al navegador un límite de afectación a todo lo contenido en ellas, las etiquetas abiertas, son por sintaxis abiertas y no necesitan de cerrarse para que estas sean comprendidas por el navegador.

La estructura básica de la mayoría de las etiquetas es la siguiente:

**<etiqueta atributo="valor">elemento afectado </etiqueta>**

El atributo es la parte adicional de las etiquetas que contienen opciones de estas. Una etiqueta puede no tener atributos o tener "n" atributos, siendo estos separados por un espacio en blanco.

El valor es asignado siempre a un atributo.

Por ultimo, las etiquetas pueden estar escritas en minúsculas o mayúsculas esto es, la etiqueta **<ETIQUETA>** es igual a la etiqueta **<etiqueta>**.

La estructura básica de un documento HTML consta de dos partes:

- Encabezado (HEAD): Que es la parte del documento que contiene información referente a la página; como el título, etiquetas de documentación (Meta), Scripts de JavaScript, Definiciones de estilo (CSS), entre otros elementos.
- Cuerpo (BODY): Es la parte que contiene toda aquella información que es desplegada como contenido principal del documento por el navegador: es decir, es la parte en la que va la información que se quiere dar a conocer.

El nombre de las páginas WEB deben ser en minúsculas y sin caracteres especiales incluyendo espacios en blanco, ya que estas se almacenan en

servidores WEB, para poder hacerlas accesibles en Internet, y recordar que la mayoría de estos servidores están basados en sistemas operativos UNIX o Linux, en los que, no se pueden usar caracteres especiales y son sensibles a mayúsculas y minúsculas.

Todo el contenido de un documento HTML debe encontrarse englobado en el interior de las etiquetas <HTML>...</HTML>. Esto indica al navegador que todo lo que se encuentra entre estas dos etiquetas, es código HTML y que debe ser interpretado como tal.

<b>&lt;html&gt;</b>	Inicio del documento.
<b>&lt;head&gt;</b>	Inicio de la cabecera.
<b>&lt;title&gt;</b>	Inicio del texto de la barra de título
<b>&lt;/title&gt;</b>	Final del texto de la barra de título
<b>&lt;/head&gt;</b>	Final de la cabecera
<b>&lt;body &gt;</b>	Inicio del cuerpo
<b>&lt;/body&gt;</b>	Final del cuerpo
<b>&lt;/html&gt;</b>	Final del documento.

Este es el esqueleto principal de todo documento HTML, cabe hacer notar que todas las etiquetas principales del documento son etiquetas de tipo cerrado, de esta manera son identificadas por el navegador.

Durante el curso se fue diseñando un proyecto que contenía todo lo visto en clase, conforme se iban estudiando nuevas etiquetas se avanzaba en el proyecto. El proyecto consistió en lo siguiente:

Elaborar una página Web, el tema que se trato fue individual, fue necesario que dicho tema fuera desglosado en la página, por ello éste debía de ser lo suficientemente amplio para permitir su desarrollo en varias partes, igual que cualquier trabajo de investigación. El trabajo debía cumplir los siguientes parámetros:

### **Utilizar Frames (Obligatorio)**

- Emplear varios tipos de letra, cambiar su tamaño y color.
- Cambiar el color de fondo de la página o poner imagen de fondo, o incluso si es frames se podían hacer las dos cosas.
- Utilizar tablas.
- Insertar ligas a otras páginas (y que alguna imagen fuera liga)
- Insertar imágenes (simples y animadas)
- Insertar algún tipo de lista o una tabla
- Hacer un ejemplo de envío de datos vía formulario con la herramienta mailto.

El proyecto que se presentó fue el siguiente, se mostraran las ventanas representativas de dicho proyecto.



Figura 6. Ventana principal de inicio.

La imagen anterior muestra lo solicitado en el proyecto, uso de frames o marcos, empleo de distintos tipos de letra, imágenes de fondo, uso de hojas de estilo en cascada CSS, vínculos con texto e imágenes funcionando como vínculos, uso de tablas para manejo de información, menús dinámicos colocados en tablas.



Figura 7a. Enlaces activos hacia las páginas

# Informe del Proyecto de un Sistema de Control de Inventarios con Software Libre



Figura 7b. Enlaces activos hacía las páginas

Se creó también un formulario que realizaba la acción de enviar información a un determinado correo utilizando la acción especial de HTML mailto.

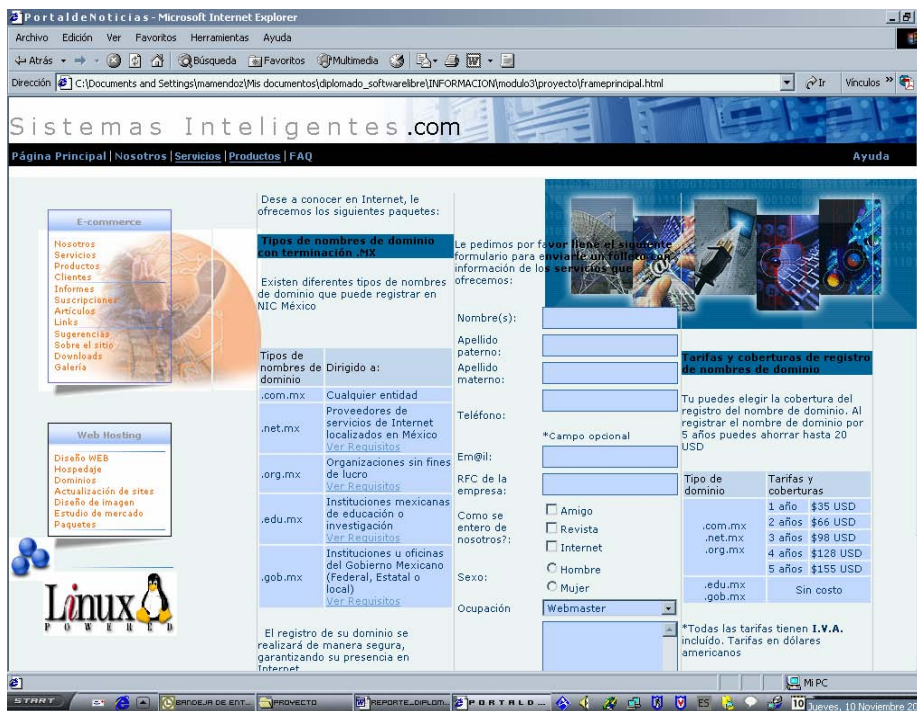


Figura 8. Formulario

En general se cumplió con el objetivo del módulo que sirvió para plantar las bases necesarias para comprensión de los módulos subsecuentes.

Se concluye también que el conocer HTML desde su código, da cimiento al desarrollo de sistemas actuales, que en gran medida basan su estructura en documentos que contienen código HTML. El lenguaje HTML es la base de muchos de los documentos que se encuentran disponibles en la red, en conjunto con lenguajes de programación como PHP se vuelven indispensables para el diseñador y programador de sitios Web dinámicos.

En el siguiente punto se describirá el uso y administración de servidores WEB en Linux y el papel que tienen las páginas WEB, la creación de documentos más complejos, así como su correcta administración e implementación.

## 1.4 Administración de Servidores WEB con Linux.

### Objetivo General.

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

### Objetivo específico.

El participante identificará el procedimiento para instalar, configurar y administrar su propio servidor de WWW en un servidor de plataforma Linux.

### Informe del módulo IV

Hoy en día, el panorama informativo y de conocimientos ha evolucionado gracias a las nuevas tecnologías. Internet se ha convertido en una biblioteca mundial de información, todo esto gracias a la infraestructura con la que se cuenta, la presentación de dicha información esta basada en estándares mundiales donde el denominador común son los servicios ofrecidos por equipos denominados *servidores* que tienen instalado un software capaz de interpretar la información presentada en los documentos denominados *página web*.

Dichos documentos son creados a partir de la base de lenguaje HTML, estudiado en el modulo III de este diplomado. El módulo trata principalmente de enfocar el proceso de presentación de información entre una máquina que se denominará *cliente* y una máquina que se denominará *servidor*, a dicho modelo se le denomina *modelo cliente-servidor*, el cual será analizado a continuación:

El modelo cliente servidor, habla sobre la relación existente entre dos equipos remotos para la transferencia y presentación de información de tipo documento HTML o bien página WEB, donde el *cliente* es un equipo remoto localizado en cualquier parte del mundo que realiza peticiones de información o requiere de algún servicio, a un *servidor*, que es un equipo remoto localizado también en cualquier parte del mundo y que contiene un software que es capaz de interpretar, procesar y presentar la información requerida o bien permitir acceso al servicio requerido, mediante una red, ya sea local o mundial. Estos servicios pueden ser: peticiones de datos de una base de datos, de información contenida



en archivos o los archivos en sí mismos o peticiones de imprimir datos en una impresora asociada.

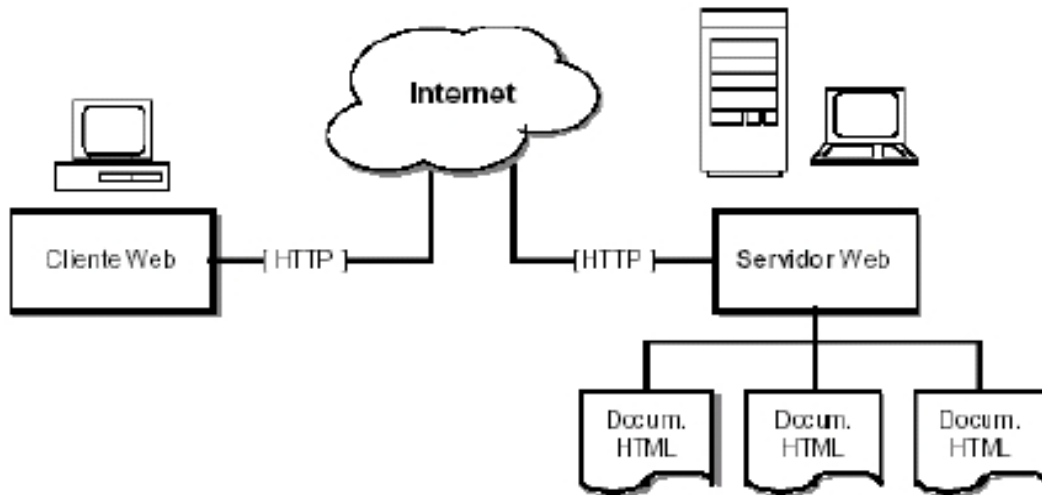


Figura 9. Modelo Cliente - Servidor

En este caso la relación existente entre un cliente y un servidor es exclusivamente para la presentación de información contenida en el servidor para ser mostrada en el cliente, solicitada por peticiones a la base de datos o simplemente por información contenida en archivos HTML (páginas web).

El servidor de páginas Web o servidor WWW ofrece servicios dentro de la red mundial de información (World Wide Web) y no es otra cosa que un programa encargado de ofrecer comunicación del servidor con los clientes a través de protocolos de red, el principal protocolo de comunicación es HTTP (Protocolo de Transferencia de Hipertexto – Hypertext Transfer Protocol), este protocolo se encuentra en la capa de aplicación del modelo OSI, basa su operación en la arquitectura cliente-servidor descrita anteriormente.

HTTP está encargado de publicar “recursos” electrónicos consultando los recursos que el servidor ofrece, su funcionamiento es simple pero poderoso y será detallado a continuación:

1. El cliente HTTP abre una conexión.
2. El servidor manda una señal de aceptación conocida como “acknowledge” notificando que se ha abierto una sesión.
3. El cliente envía un mensaje de petición llamado “request message” solicitando un recurso.
4. El servidor responde con un mensaje de respuesta “response message” que contiene el recurso solicitado y cierra la conexión.

La demostración de este proceso es simple:

Si se abre una sesión de algún navegador Web (Netscape, Firefox, Mozilla, Konqueror, Internet Explorer) y en la barra de direcciones se teclea la dirección o URL (Localizador Uniforme de Recursos – Uniform Resource Locator) de algún sitio Web, por ejemplo <http://www.google.com>, el cliente realizará una solicitud de presentación de información hacia el servidor y este a su vez mostrará una página web respondiendo a la petición del cliente, realizando dicho proceso de manera transparente para el usuario, pues el usuario no visualiza dicho proceso, el solo ve la información o la página web solicitada en su pantalla.

Existen muchos servidores de páginas web en el mercado, los hay gratuitos, propietarios, versiones de prueba (shareware) etc., el utilizado durante el diplomado fue *Apache*, que es un servidor libre tipo *open source*, y trabaja sobre todas las plataformas conocidas (UNIX, Linux, Windows), la versión de servidor con la cual se trabajo fue la 2.0, y el cual tiene una serie de características que lo hacen único y se mostrará a continuación:

- Robusto, soporte de un gran numero 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 codigo fuente del servidor
- Soporte para accesos restringidos
- Soporte para SSL
- Y además es gratuito.

La instalación de apache sobre Linux se llevo a cabo de la siguiente manera:

1. Se obtuvo una copia del software en el sitio Web del proyecto apache
2. <http://www.apache.org>
3. Una vez obtenida una copia del archivo se realiza lo siguiente:
4. `$ tar zxvf httpd-2.0.54.tar.gz`
5. `$ cd httpd-2.0.54`
6. `$ ./configure --prefix=/usr/local/apache2`
7. `$ make`
8. `$ make install`

Se continúa con el proceso de instalación y una vez terminado se procede a configurar Apache mediante el archivo de configuración principal llamado

*httpd.conf*, el cual se encuentra ubicado dentro del directorio de sistema: `/usr/local/apache2/conf/`.

El archivo de configuración de Apache está diseñado para ser administrado por más de 200 directivas, las cuales permiten que determinada funcionalidad pueda ser incluida, como administrador estas directivas son controladas para su disponibilidad de acuerdo a los módulos con lo que se compila Apache.

La configuración de Apache se realiza mediante directivas organizadas en tres grandes grupos:

- Directivas de ambiente global *Global Environment*
- Directivas principales de servidor *Main Server*
- Directivas para servidores virtuales *Virtual Servers*

La sección *Global Environment* administra las directivas generales de operación para apache.

La sección *Main Server* administra las directivas del servidor principal o estándar de apache.

La sección *Virtual Host*, administra las directivas donde los mismos procesos de apache soportan diversas IP o nombres de dominio.

En la práctica se configuró Apache solo en algunas directivas para su funcionamiento mínimo, tales como el nombre del servidor, el puerto donde se ejecuta Apache, numero de peticiones máximas permitidas, usuario con derechos de ejecución para Apache, alias o dirección de internet IP para el servidor, como se muestra a continuación:

```
### Section 1: Global Environment
#
# The directives in this section affect the overall operation of Apache,
# such as the number of concurrent requests it can handle or where it
# can find its configuration files.
#
ServerRoot "/usr/local/apache2"
.
.
Timeout 300

KeepAlive On

MaxKeepAliveRequests 100

KeepAliveTimeout 15
.
.
```

```
Listen 132.248.75.164:8080
    Listen 8081
Listen 8082
```

```
### Section 2: 'Main' server configuration
```

```
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition. These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#
```

```
<IfModule !mpm_winnt.c>
<IfModule !mpm_netware.c>
```

```
User webfinal
Group #-1
</IfModule>
</IfModule>
```

```
ServerAdmin miguelmh@aulat-164.mascarones.unam.mx
```

```
#ServerName www.example.com:80
```

```
UseCanonicalName Off
```

```
DocumentRoot "/usr/local/apache2/htdocs"
```

```
.
```

```
### Section 3: Virtual Hosts
```

```
#
# VirtualHost: If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them. Most configurations
# use only name-based virtual hosts so the server doesn't need to worry about
# IP addresses. This is indicated by the asterisks in the directives below.
#
# Please see the documentation at
# <URL:http://httpd.apache.org/docs-2.0/vhosts/>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-S' to verify your virtual host
# configuration.
```

```
#NameVirtualHost *:80
```

```
<virtualhost 132.248.75.164:8081>
    ServerAdmin webmaster@sistemasinteligentes.com.mx
    DocumentRoot /home/virtual/ventas/htdocs
    ServerName 132.248.75.164:8081
    ErrorLog logs/132.248.75.164:8081-error_log
    CustomLog logs/132.248.75.164:8081-access_log common
```

```
Options Indexes
<Directory "/home/virtual/ventas/htdocs/test">
  Options -Indexes
</Directory>
</virtualhost>

<virtualhost 132.248.75.164:8082>
  ServerAdmin webmaster@sistemasinteligentes.com.mx
  DocumentRoot /home/virtual/direccion/htdocs
  ServerName 132.248.75.164:8082
  ErrorLog logs/132.248.75.164:8082-error_log
  CustomLog logs/132.248.75.164:8082-access_log common
<Directory "/home/virtual/direccion/htdocs/test">
  Options Indexes
</Directory>
<Directory "/home/virtual/direccion/htdocs/ssl-bin">
  Options ExecCGI Indexes Includes
  AddHandler cgi-script .cgi
  AddType text/html .shtml
  AddOutputFilter INCLUDES .shtml
</Directory>
</virtualhost>
```

Una vez configurado el servidor Apache con los parámetros antes vistos, se procedió a estudiar los archivos donde residen las bitácoras del servidor, el archivo *access.log* que registra todos los accesos al sitio, el archivo *error.log* que registra los errores que genere un acceso al sitio o bien que los procesos de Apache reporten.

Después se procedió a arrancar el servidor Apache utilizando varios métodos, el primero de ellos es ejecutar desde línea de comandos el proceso como se muestra a continuación:

**`$/usr/local/apache2/bin/apachectl start`** la ejecución de este comando le indica al sistema que arranque Apache con su proceso conocido (httpd).

Si todo está configurado correctamente al ejecutar dicho comando no debe emitir ninguna advertencia y se puede comprobar que el proceso se encuentra en ejecución utilizando el siguiente comando:

**`$ps -fea | grep httpd`**

Si lo que se necesita es detener el proceso se ejecuta el siguiente comando:

**`$/usr/local/apache2/bin/apachectl stop`**

La ejecución de este comando le indica al sistema que detenga Apache con su proceso conocido (httpd).

Una vez instalado, configurado se verifica que Apache este funcionando, la comprobación es simple, solo basta con abrir una ventana del navegador Web y colocar en la url:

**http://localhost** Esta dirección indica que se abra la página por definición (default) en el navegador para poder ser visualizada, la página contiene información básica sobre la correcta configuración de Apache y muestra también algunos parámetros que toma del archivo de configuración como página de inicio.



Figura 10. Página predeterminada del servidor Apache

De esta manera queda configurado Apache, y esta listo para servir páginas Web.

Se realizaron pruebas creando algunas páginas web y publicándolas en el sitio para su visualización. El directorio que contiene las paginas Web es *htdocs* y se encuentra ubicado dentro de */usr/local/apache2/htdocs*, dentro de este directorio se crearon otros subdirectorios donde se colocaron nuevas páginas para comprobar el funcionamiento de Apache como servidor de páginas Web y sitios Web Virtuales.

Apache no solo funciona como servidor de páginas web, ya que puede ser utilizado en conjunto con otras herramientas y de esta manera crear sitios dinámicos, es decir, la presentación de páginas estáticas (páginas creadas con código HTML puro) quedo atrás, hoy en día todos los sitios disponibles en Internet poseen al menos una aplicación creada en Java, alguna script programada en algún lenguaje de programación dinámico como JavaScript, Perl, Python, C, o PHP para realizar alguna función específica con alguna iteración directa con alguna base de datos realizada en Oracle, PostgreSQL, MySQL. La integración de estas herramientas ha dado lugar a sitios web totalmente interactivos dando ventaja a la misma actualización de contenidos en línea, la presentación de la información por resultados de búsquedas desde aplicaciones conocidos como motores de búsqueda, la personalización de portales y sitios web mediante módulos.

Este tipo de integración de herramientas genera una plataforma ideal para desarrollar entornos dinámicos en Internet, el uso de Linux, Apache, MySQL y PHP proporcionan a los usuarios una herramienta de trabajo poderosa convirtiendo a Internet no solo en una fuente inagotable de información sino en una fuente de trabajo a nivel mundial.

Las aplicaciones creadas con estas herramientas no necesitan instalarse sobre los clientes para poder ser visualizadas, son aplicaciones multiplataforma que pueden utilizarse en clientes con UNIX, Linux, Windows o MacOS sin diferencia alguna, convirtiendo al servidor de páginas web en un servidor de aplicaciones.

En la práctica se llevo a cabo la configuración de Apache para la ejecución de contenido dinámico mediante el uso de SSI (Inclusiones del Lado del Servidor - Server Side Includes), que funcionan como banderas de presentación de información dinámica.

La modificación hecha al archivo de configuración httpd.conf fue la siguiente:

```
<Directory "/usr/local/apache2/htdocs/partessi">  
  Options ExecCGI Indexes Includes  
  AddHandler cgi-script .cgi  
  AddType text/html .shtml  
  AddOutputFilter INCLUDES .shtml  
</Directory>
```

Los pasos a seguir para que funcionaran los SSI fueron los siguientes:

- Se instaló apache en su última versión.
- Se ubicó el directorio donde se almacenan los documentos web y dentro se creó un subdirectorio donde se crearon los archivos con extensión shtml, y que incluyen las variables de valores SSI.
- Se agregaron las directivas necesarias para la activación de includes y soporte de archivos shtml en el segmento correspondiente al directorio antes descrito en el archivo de configuración de Apache httpd.
- Por último se reinició el servidor Apache para que surtieran efecto los cambios realizados, se probó que se visualizarán los documentos en un cliente WEB.

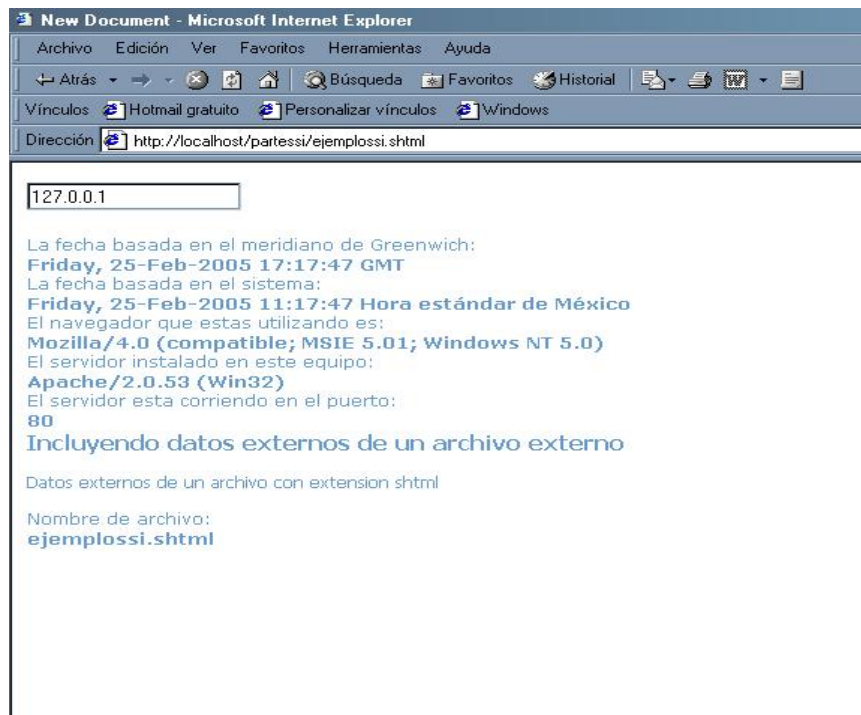


Figura 11. Ejemplificando documentos SSI

El código de los archivos con extensión .shtml es el siguiente

```
<HTML>
<HEAD>
<TITLE> Ejemplo de SSI </TITLE>
</HEAD>
<BODY>
<form method="post" action="" name="forma1">
```



```
<input type="text" name="ipaddr" value="<!--#echo var='REMOTE_ADDR'-->" readonly>
</form>
<FONT SIZE="2" COLOR="#6699CC" face="verdana">La fecha basada en el meridiano de Greenwich:</FONT>
<br>
<FONT SIZE="3" COLOR="#6699cc" face="verdana"><b><!--#echo var="DATE_GMT"--></b></FONT> <br>
<FONT SIZE="2" COLOR="#6699cc" face="verdana">La fecha basada en el sistema:</FONT> <br>
<FONT SIZE="3" COLOR="#6699cc" face="verdana"><b><!--#echo var="DATE_LOCAL"--></b></FONT><br>
<FONT SIZE="2" COLOR="#6699cc" face="verdana">El navegador que estas utilizando es:</FONT> <br>
<FONT SIZE="3" COLOR="#6699cc" face="verdana"><b><!--#echo var="HTTP_USER_AGENT"--
></b></FONT><br>
<FONT SIZE="2" COLOR="#6699cc" face="verdana">El servidor instalado en este equipo:</FONT> <br>
<FONT SIZE="3" COLOR="#6699cc" face="verdana"><b><!--#echo var="SERVER_SOFTWARE"--
></b></FONT><br>
<FONT SIZE="2" COLOR="#6699cc" face="verdana">El servidor esta corriendo en el puerto:</FONT> <br>
<FONT SIZE="3" COLOR="#6699cc" face="verdana"><b><!--#echo var="SERVER_PORT"--></b></FONT><br>

<B><FONT SIZE="4" COLOR="#6699cc" face="tahoma">Incluyendo datos externos de un archivo
externo</FONT></B>
<BR><BR>
<!--#include virtual="inserta.shtml" -->
</body>
</HTML>
```

El código anterior es del archivo llamado ejemplossi.shtml, a continuación se muestra el código de la página web que esta insertada y se llama inserta.shtml

```
<HTML>
<HEAD>
<TITLE> Archivo de insercion </TITLE>
</HEAD>
<BODY>
<FONT SIZE="2" COLOR="#6699cc" face="tahoma">Datos externos de un archivo con extension
shtml</FONT><BR><BR>
<FONT SIZE="2" COLOR="#6699cc" face="verdana">Nombre de archivo:</FONT> <br>
<FONT SIZE="3" COLOR="#6699cc" face="verdana"><b><!--#echo var="DOCUMENT_NAME"--
></b></FONT><br>
</BODY>
</HTML>
```

Se concluye que Apache es un servidor de páginas WEB seguro, potente, confiable y muy sencillo de configurar y administrar, en este módulo se dio el paso para comenzar a aprender a desarrollar sistemas y sitios web dinámicos que es finalmente el objetivo primordial del diplomado, y que se tratará en el siguiente punto.

## 1.5 Programación con PHP

### Objetivo General.

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

### Objetivo específico.

Proporcionar al participante los conocimientos necesarios que le permitan crear aplicaciones dinámicas e interactivas para el Web utilizando el lenguaje PHP.

### Informe del módulo V

PHP es un lenguaje de programación orientado totalmente a la Web, la programación se realiza mediante guiones o scripts y estos son ejecutados en el servidor Web (Server Side), es manejado como código embebido en páginas HTML como etiquetas, la instalación de este con Apache es mediante un módulo como interprete, entre sus funciones y aplicaciones comunes, funciona como autenticador HTTP mediante cabeceras HTTP, ó métodos mas complejos con cookies, identificadores en urls o directorios.

Para la creación de imágenes GIF, usando PHP con la librería GD están disponibles funciones para crear líneas, círculos, texto, etc en formato de imagen con extensión .gif.

Para el manejo de bases de datos, con soporte nativo para Oracle, Adabas D, Sybase, FilePro, mSQL, Velocis, MySQL, Informix, Solid, dBase, ODBC, Unix dbm, PostgreSQL.

Tiene soporte PDF, IMAP, conexiones de red, XML, archivos comprimidos con extensión .gz, etc., estas diferentes habilidades extra, son dadas por módulos/librerías.

La instalación en Linux es muy sencilla y se puede realizar de dos formas:

– “Sencillo y Practico”

Por paquetes de la distribución que se tenga instalada.

– “En cualquier UNIX Linux”

Se obtiene la última versión del paquete del sitio <http://www.php.net>

Compilación

Se descomprime el archivo para obtener un directorio con los archivos de instalación

```
$ tar zxvf php-5.0.4.tar.gz
```

Se cambia al directorio creado de la descompresión

```
$ cd php-5.0.4
```

Se ejecuta comando de instalación para que PHP sea configurado con Apache y con algún manejador de basas de datos como PostgreSQL o MySQL

```
./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-pgsql --with-mysql
```

Se crean los archivos binarios de preinstalación con opción de configuración previa

```
$ make
```

Se ejecuta la instalación con las dependencias creadas y se copian los archivos y librerías a las rutas especificadas

```
# make install
```

Una vez finalizada la instalación, se copia el archivo de configuración de PHP a un directorio del sistema. El archivo de configuración (llamado **php.ini**) es leído cuando arranca PHP. Para las versiones de PHP como módulo de servidor esto sólo ocurre una vez al arrancar el servidor web. Para la versión **CGI** y **CLI**, esto ocurre en cada llamada.

Las directivas php.ini gestionadas por extensiones están documentadas en cada una de las páginas de las extensiones respectivamente. La lista de directivas de núcleo se encuentra disponible en el apéndice. La mayoría de las directivas PHP están listadas en ini\_set() con los respectivos permisos y enlaces a la documentación.

### # cp php.ini-dist /usr/local/lib/php.ini

Para que PHP funcione como módulo de Apache y se pueda trabajar con scripts con extensión .php se tiene que editar el archivo httpd.conf y agregar las siguientes líneas:

```
LoadModule php5_module modules/libphp5.so  
AddType application/x-httpd-php .php .phtml  
AddType application/x-httpd-php-source .phps
```

Para validar que Apache y PHP han sido completamente instalados y configurados se realiza lo siguiente:

#### 1. Arrancar el Servidor de Web

```
/usr/local/apache2/bin/apachectl start
```

Si Apache se encuentra corriendo se debe detener el proceso y reiniciar el servidor para que los cambios realizados en el archivo httpd.conf surtan efecto

#### 2. Validar que el módulo de PHP mod\_PHP ha quedado integrado a Apache

3. Generar y probar el script info.php, el cual debe de ir dentro del directorio htdocs:

```
<?php  
phpinfo();  
<?
```

#### 4. Se le asignan los permisos necesarios para su correcta ejecución

```
$chmod 755 info.php
```

5. Se verifica acceso al archivo abriendo una ventana del navegador y escribiendo en la barra de direcciones la siguiente dirección: <http://localhost/info.php> aparecerá una página Web que contiene la información de los parámetros básicos de configuración de PHP en el sistema.

6. El sistema está preparado para comenzar a trabajar con PHP. Todas las scripts generadas en PHP deben estar ubicadas en el directorio htdocs de Apache o bien en algún directorio que se haya destinado como virtual (virtual host) y cuya configuración este realizada dentro del archivo de configuración de Apache como se estudió anteriormente.



Figura 12. Página principal de configuración de PHP

Durante el módulo se aprendieron las técnicas de programación con PHP, partiendo del manejo de la sintaxis básica del lenguaje, como se mencionó anteriormente, PHP puede ser embebido dentro del código HTML, para interpretar un archivo, PHP simplemente interpreta el texto del archivo hasta que encuentra uno de los caracteres especiales que delimitan el inicio de código PHP. El intérprete ejecuta entonces todo el código que encuentra, hasta que encuentra una etiqueta de fin de código, que le dice al intérprete que siga ignorando el código siguiente. Este mecanismo permite embeber código PHP dentro de HTML: todo lo que está fuera de las etiquetas PHP se deja tal como está, mientras que el resto se interpreta como código.

Hay cuatro conjuntos de etiquetas que pueden ser usadas para denotar bloques de código PHP. De estas cuatro, sólo 2 `<?php. . .?>` y `<script language="php">. . .</script>` están siempre disponibles; el resto pueden ser configuradas en el archivo de configuración de PHP (php.ini) para ser o no aceptadas por el intérprete. Mientras que el formato corto de etiquetas (short-form tags) y el estilo ASP (ASP-style tags) pueden ser convenientes, no son portables como la versión de formato largo de etiquetas. Además, si se pretende embeber código PHP en XML o XHTML, será obligatorio el uso del formato `<?php. . .?>` para la compatibilidad con XML.

Las etiquetas soportadas por PHP son:

1.<?php echo("si se quiere servir documentos XHTML o XML, se hace así\n");?>

2.<? echo ("esta es la más simple, una instruccion de procesado SGML \n"); ?>  
<?= expression ?> Esto es una abreviatura de "<? echo expression ?>"

3.<script language="php">  
echo("muchos editores (como FrontPage) no aceptan instrucciones de  
procesado");  
</script>

4.<% echo ("Opcionalmente, se pueden usar las etiquetas ASP"); %>  
<%= \$variable; # Esto es una abreviatura de "<% echo . . ." %>

El método primero, <?php. . .?>, es el más conveniente, ya que permite el uso de PHP en código XML como XHTML.

El método segundo no siempre está disponible. El formato corto de etiquetas está disponible con la función `short_tags()` (sólo PHP 3), activando el parámetro del archivo de configuración de PHP **short\_open\_tag**, o compilando PHP con la opción **--enable-short-tags** del comando `configure`. Aunque esté activa por defecto en **php.ini-dist**, no se aconseja el uso del formato de etiquetas corto.

El método cuarto sólo está disponible si se han activado las etiquetas ASP en el fichero de configuración: **asp\_tags**. El soporte de etiquetas ASP se añadió en la versión 3.0.4.

No se debe usar el formato corto de etiquetas cuando se desarrollen aplicaciones o bibliotecas con intención de redistribuirlas, o cuando se desarrolle para servidores que no están bajo nuestro control, porque puede ser que el formato corto de etiquetas no esté soportado en el servidor. Para generar código portable y redistribuible, hay que asegurarse de no usar el formato corto de etiquetas.

Durante el módulo se fue diseñando un proyecto sobre una agenda telefónica en versiones inglés y español que validará el acceso a los usuarios como administradores y como simples usuarios, la característica fue que el acceso con perfil de administrador permitía realizar cambios en la agenda y poder dar de alta más usuarios, en el caso de perfil de usuario, solo podía leer el contenido de los datos de la agenda, pero no podía realizar cambios, la pantalla de inicio de dicho proyecto se muestra a continuación:

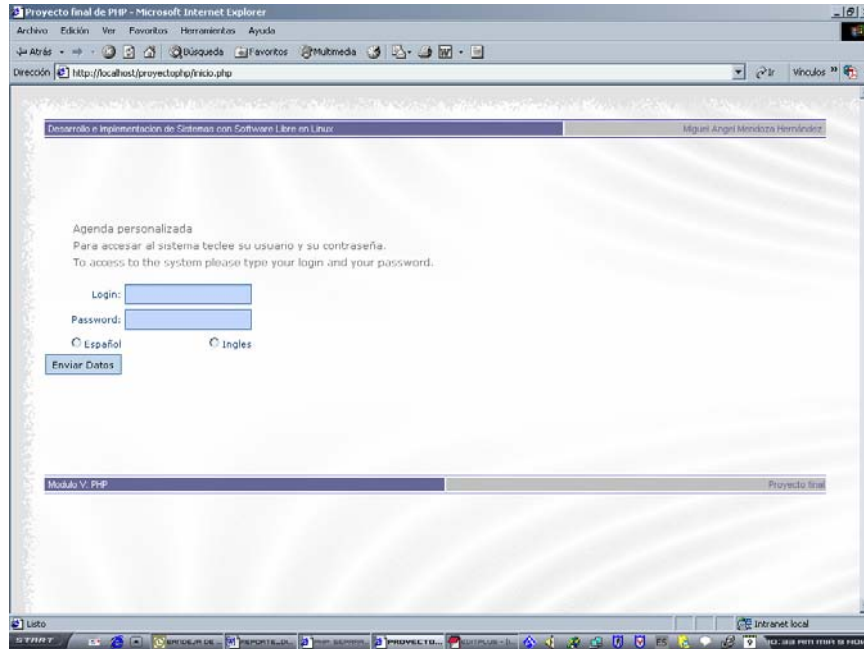


Figura 13. Pantalla de inicio del proyecto del módulo de PHP

Como se observa en la imagen, la página de inicio es un formulario con dos cajas de texto solicitando el usuario y su contraseña así como un botón de radio para seleccionar el idioma en el cual se pretende entrar al sistema y un botón para envío y validación de datos, al ingresar un usuario valido y una contraseña y seleccionar el idioma con el perfil de administrador, se validará el usuario y la contraseña y si estos son correctos aparece una ventana emergente con la leyenda “Acceso Autorizado” y pasa a la siguiente pantalla:

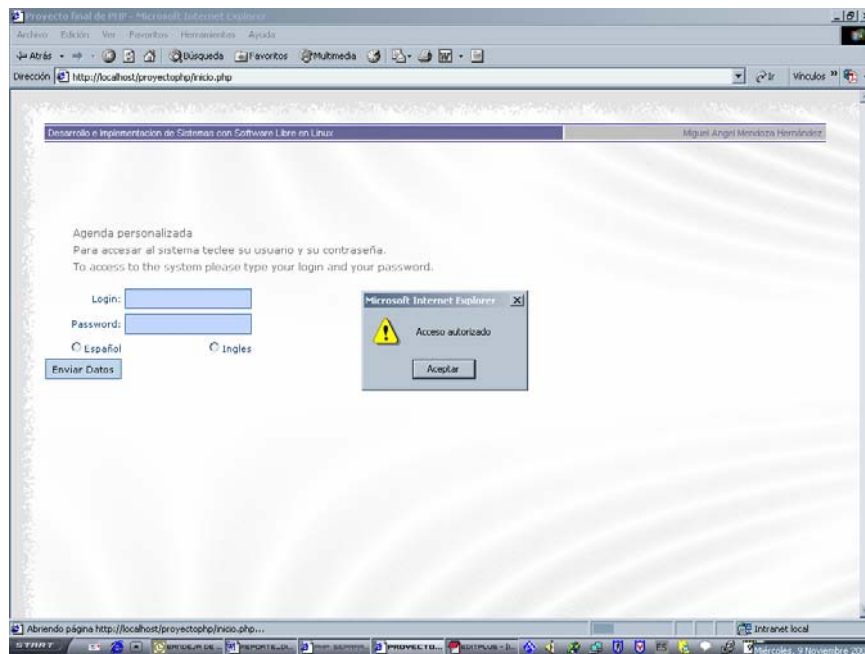


Figura 14 Pantalla de acceso autorizado al sistema

Aparece una ventana de formulario donde se teclean los datos que serán almacenados en la agenda.

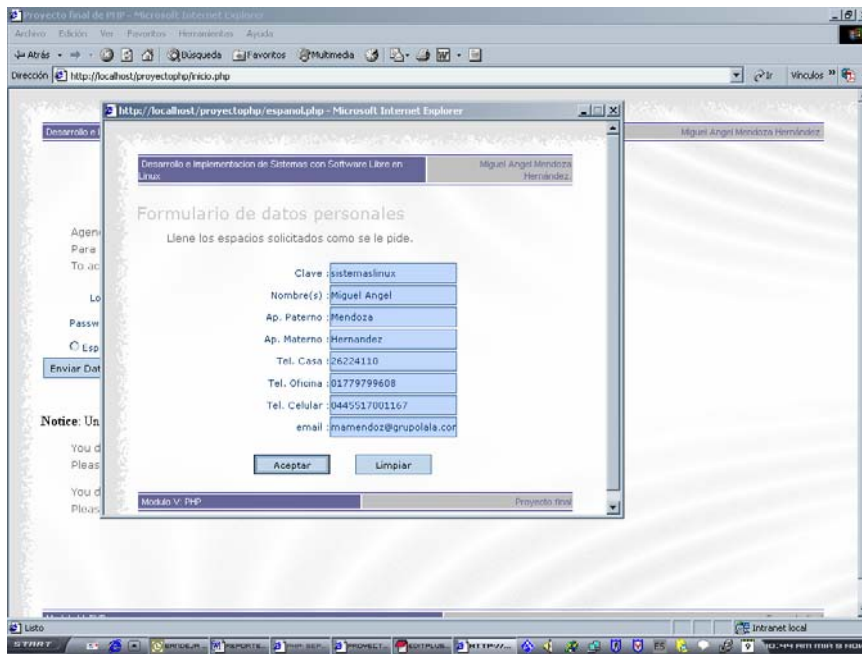


Figura 15 Formulario de captura para agenda

Una vez llenados los campos se procede a enviar la información para su almacenamiento y pasara a la siguiente pantalla que muestra la información actualizada contenida en el archivo de agenda, quedando habilitados como vínculos para ver a detalle la información de cada usuario.

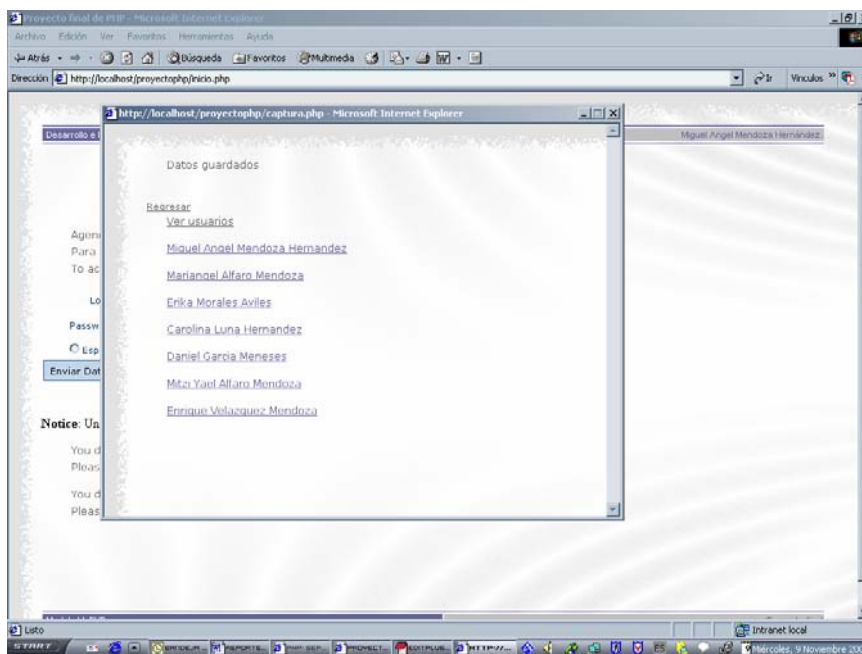


Figura 16a Datos capturados



# Informe sobre el Diplomado “Desarrollo de sistemas con Software Libre”

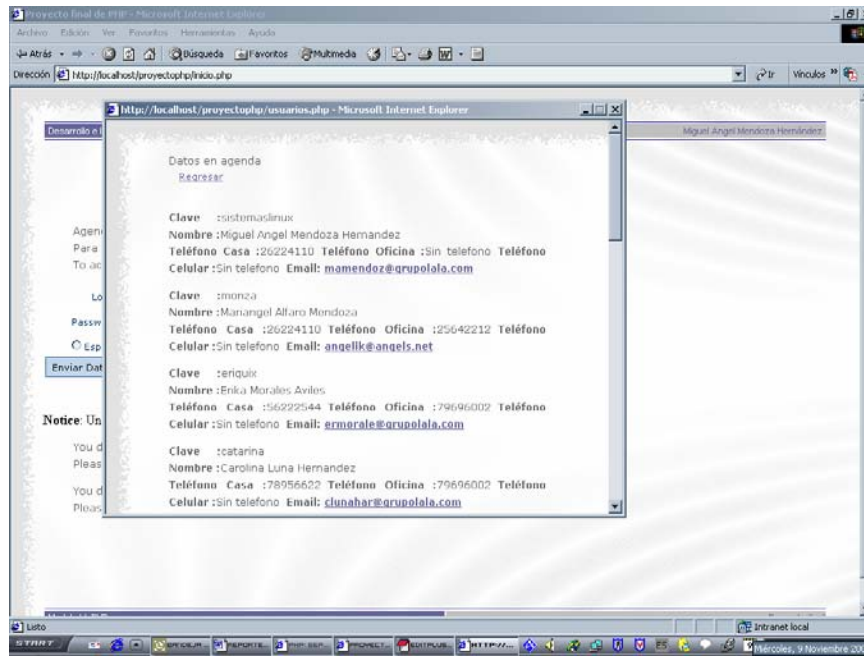


Figura 16b Datos almacenados

Si se accede al sistema como usuario normal la única ventana que mostrará el sistema será la de información de los usuarios registrados en la agenda como se muestra a continuación.

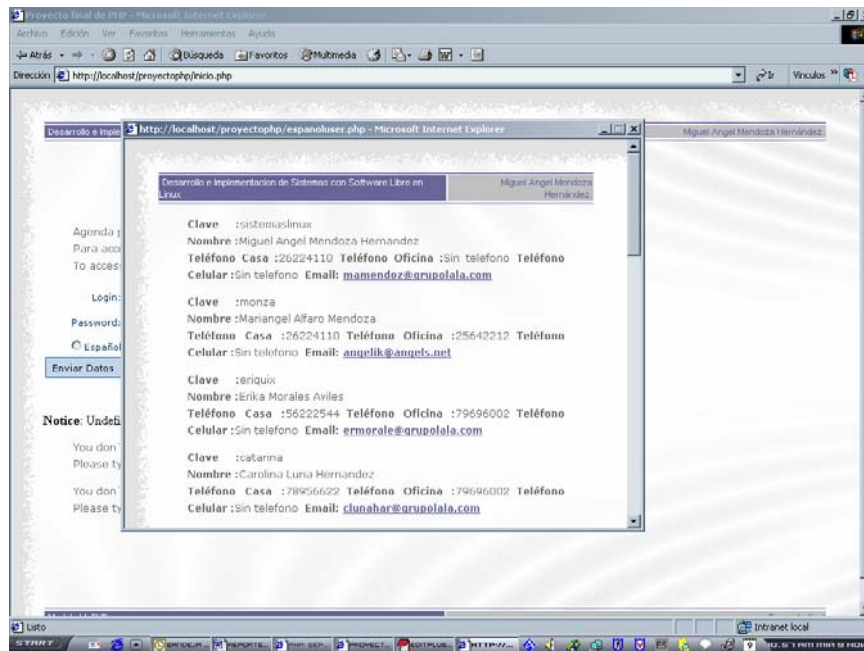


Figura 17 Datos mostrados modo usuario

Para la realización de este proyecto se consolidaron los conocimientos adquiridos en los módulos pasados utilizando las herramientas aprendidas como manejo de archivos en Linux, uso de utilerías, comandos para procesamiento de información en archivos, uso del lenguaje HTML, manejo de Apache, y programación de PHP, en conjunto se cumplieron los objetivos esperados y se obtuvieron las bases sólidas para los módulos subsecuentes.

En el siguiente punto se describe el uso de bases de datos para almacenar información de forma masiva, así como el uso de herramientas para su fácil administración y creación de reportes, y el uso de las herramientas vistas hasta el momento para la generación de documentos dinámicos.

## **1.6 Interacción de WWW con bases de datos MySQL.**

### **Objetivo General.**

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

### **Objetivo Específico.**

El alumno manejará MySQL en la construcción de bases de datos que funcionen a través del WWW y hará consultas sobre ellas.

### **Informe del Módulo VI.**

El módulo de MySQL, trató principalmente sobre el uso de esta valiosa herramienta en conjunto con herramientas de software libre aprendidas en módulos anteriores, HTML, Apache, PHP y el mismo sistema operativo Linux como plataforma, a este conjunto de herramientas se le conoce con el término Soluciones LAMP, dicho término se origina a finales del 2000 en Alemania para describir a las aplicaciones web creadas utilizando la siguiente combinación de herramientas: Linux, el sistema operativo; Apache, el servidor web; MySQL, el servidor de bases de datos; Perl, PHP, y/o Python, lenguajes de programación.

LAMP está considerada como una de las mejores herramientas disponibles para que cualquier organización o individuo pueda emplear un servidor web versátil y potente. Aunque creados por separado, cada una de las tecnologías que lo forman, disponen de una serie de características comunes. Especialmente interesante es el hecho que estos cuatro productos pueden funcionar en una amplia gama de hardware, con requerimientos relativamente pequeños sin perder estabilidad. Esto ha convertido a LAMP en la alternativa más adecuada para pequeñas y medianas empresas.

Algunas de las ventajas que se obtienen de utilizar LAMP son:

- Soporte a gran cantidad de arquitecturas, como son Intel y compatibles, SPARC, Mips y PPC (Macintosh).
- Código relativamente sencillo y con pocos cambios de una plataforma a otra.
- Parches generados en poco tiempo después de encontrarse un agujero de seguridad.
- Actualizaciones del software vía Internet.
- Posibilidad de incrementar los servicios y funciones desde el código fuente

Sin embargo, existen también una serie de desventajas que deben considerarse:

- Es muy distinto de Windows, lo que dificulta el trabajo a quienes estén acostumbrados a él.
- Las actualizaciones requieren en ocasiones tener conocimientos profundos del sistema.
- Configurar algunos servicios de red requiere de más tiempo que en Windows.
- Mayor coste del personal.
- Todos los elementos que forman LAMP son software libre, de modo que disfrutan de las siguientes ventajas propias del mismo:
  - Libertad de copia y distribución.  
Se puede conseguir gratuitamente en Internet. Hay muchísimas fuentes donde conseguir cualquiera de las distribuciones. Si no se cuenta con una conexión rápida, también regalan Linux en los CD-ROM de muchas revistas especializadas.
  - Libertad de modificación.  
Junto a los programas ejecutables, se puede obtener su código fuente. Esto, si se tienen los conocimientos necesarios, permite verificar la seguridad y eficiencia de los mismos, además de modificar y/o añadir las características y comportamientos que se deseen.

MySQL es un manejador de bases de datos que permite administrar y gestionar la información de manera eficiente satisfaciendo las necesidades de control que se tienen para tal información. Los datos manejados por MySQL son organizados de acuerdo a un proceso previo que comprende el análisis y diseño del modelo de datos, así como el diseño y configuración del mismo.

Hoy en día existen diferentes arquitecturas para los sistemas de gestión de las bases de datos, pero la más extendida y la que más ha tenido éxito y la que fue aprendida durante este módulo fue la *arquitectura relacional*. MySQL es un servidor de bases de datos relacionales muy rápido y robusto, es software libre

publicado bajo la licencia pública GNU (GPL GNU Public License) y mantenido totalmente por una compañía de origen sueco llamada MYSQL AB.

MySQL está disponible en Internet para un número enorme de sistemas operativos como son AIX, BSDi, DEC, UNIX, FreeBSD, HPUX, Linux, MAC OS X, NetBSD, OpenBSD, OS/2 WARP, SGI Irix, Solaris, SCO, OpenServer, SCO Unixware, Tru64 Unix, Windows.

La empresa gestora de MySQL (MYSQL AB), estima que hay 4 millones de servidores MySQL instalados en el mundo, lo que significa aproximadamente el 20% del mercado. Entre sus clientes destacan Yahoo!, Cisco, NASA, Lucent Technologies, Motorola, Google, Silicon Graphics, HP, Xerox, Sony Pictures, gran parte de su éxito se debe sin duda a que forma parte de la tecnología LAMP.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan.

Poco a poco los elementos faltantes en MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar: Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.

Disponibilidad en gran cantidad de plataformas y sistemas.

Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.

- ⊕ Transacciones y claves foráneas.
- ⊕ Conectividad segura.
- ⊕ Replicación.
- ⊕ Búsqueda e indexación de campos de texto.

La instalación de esta herramienta en Linux es muy sencilla y se describirá a continuación:

Se necesitará obtener la última versión estable del manejador en su sitio web, es recomendable obtener la versión source, el sitio web es <http://www.mysql.org>

Una vez obtenida la copia se procede a descomprimir el archivo a un directorio donde se realizará la instalación:

- **\$tar -xvzf mysql-4.1.11.tar.gz** version con la que se trabajo en el curso.
- **\$cd mysql-4.1.11/** cambiarse al directorio que se crea.
- **Editar el archivo INSTALL-SOURCE** y seguir paso a paso lo que indica a partir de la linea **3626**, a continuación se detalla:

### 2.8.1 Source Installation Overview Basic Commands to install MySQL

<b>shell&gt;groupadd mysql</b>	se crea el grupo mysql.
<b>shell&gt;useradd -g mysql mysql</b>	se crea usuario mysql y se asocia al grupo.
<b>shell&gt;tar -xvzf mysql-4.1.11.tar.gz</b>	se descomprime como se hizo en el paso 1.
<b>shell&gt;cd mysql-4.1.11/</b>	se cambia al directorio creado paso 2
<b>shell&gt;./configure --prefix=/usr/local/db</b>	se realiza configuración previa a la instalación con opción al directorio especificado.
<b>shell&gt;make</b>	se generan los binarios para la instalación.
<b>shell&gt;make install</b>	se copian los archivos y librerías a las rutas especificadas.

Una vez instalado el servidor se procede a copiar el archivo de configuración básica de mysql, que contiene información como el número de puerto en el que corre, tamaño de la base de datos entre otros, al directorio /etc/.

• **shell>cp support-files/my-medium.conf /etc/my.conf**

Para verificar que estaba funcionando el servidor se realizaron las siguientes pruebas.

Se cambia al directorio donde se encuentra instalado el servidor:

• **shell>cd /usr/local/db/mysql**

Se configuró el usuario que administraría las tablas y toda la base de datos, así como también se generaron las tablas de configuración y arranque de mysql

• **shell>bin/mysql\_install\_db -user=mysql**

Se cambiaron los dueños y los permisos a los directorios dentro del directorio /db/mysql/

- **shell>chown -R root .** se cambia el dueño del directorio padre a root
- **shell>chown -R mysql var** se cambia de dueño al directorio var, ya que es necesario para que pueda funcionar correctamente mysql, y es necesario que este directorio le pertenezca a mysql.

- **shell>chgrp -R mysql .** se cambia de grupo a todo el directorio padre en forma recurrente.

Posteriormente se procedió a levantar el servicio ejecutando el siguiente comando.

- **shell>bin/mysqld\_safe --user=mysql &**

Se verifico que se estuviera ejecutando el demonio.

- **shell>ps -fea | grep mysql**

Al no encontrar errores en la ejecución del servicio, se procedió a entrar a la base de datos para terminar de configurar algunos parámetros como las tablas de Grantt de configuración de MySQL, y verificar el estado del servidor.

- **shell>bin/mysql**

- **mysql>status** se verifica estado del servidor y se genera un archivo temporal de intercambio /tmp/mysql.sock que es un archivo con permisos de escritura para el usuario mysql.

Se procedió a configurar las tablas de Grantt.

- **mysql>use mysql;** se selecciona la base de datos de mysql para su configuración

- **mysql>show tables;** se pide al servidor que muestre las tablas contenidas en la base.

Se modificaron la tabla host y la tabla user, para realizar dicha tarea se ejecuto una selección de los campos por alterar de la tabla user.

- **mysql>select host,user,select\_priv,insert\_priv,update\_priv,delete\_priv from user;**

La tabla mostrada arrojo la información siguiente:

host	user	select_priv	insert_priv	update_priv	delete_priv
localhost	root	Y	Y	Y	Y
aula2-30	root	Y	Y	Y	Y
aula2-30		N	N	N	N
localhost		N	N	N	N

Se insertaron los valores necesarios en las columnas vacías del campo user con el valor **nobody**, que es el usuario que ejecuta Apache por default., con esta configuración se pretendían las interacciones vía Web de scripts realizados en PHP con conexión a la base de datos, teniendo como servidor de servicios WWW a Apache.

Posteriormente se mostrarán los pasos que se llevaron a cabo para que el conjunto de estas herramientas funcionara.

```
mysql> insert into user(host, user, select_priv, insert_priv,  
update_priv, delete_priv) values("localhost","nobody","Y","Y","Y","Y");  
mysql> insert into user(host, user, select_priv, insert_priv,  
update_priv, delete_priv) values("aula2-30","nobody","Y","Y","Y","Y");
```

Se verificó que los datos fueran insertados en la tabla con la siguiente instrucción sql.

```
mysql>select host,user,select_priv,insert_priv,update_priv,delete_priv  
from user;
```

La tabla mostrada con los datos agregados fue la siguiente:

host	user	select_priv	insert_priv	update_priv	delete_priv
localhost	root	Y	Y	Y	Y
aula2-30	root	Y	Y	Y	Y
aula2-30	nobody	Y	Y	Y	Y
localhost	nobody	Y	Y	Y	Y

Con esto, quedó configurado completamente el servidor listo para comenzar a crear nuevas bases de datos y tablas.

En la práctica se creó una base de datos con un par de tablas para verificar como funcionaba MySQL. Se creó una base de datos llamada agenda y en ella se creo una tabla llamada nombres que tenía el siguiente formato.



Tabla nombres.


id_nombre	id_tipo	id_tamano
cont	int	entero corto
nombre	text	
email	char	60
telefono	char	15
edad	char	3

Los tipos de datos manejados en MySQL para un valor de tipo texto no hay un tamaño definido para el campo, para un valor de tipo char, el tamaño del campo es limitado.


Con los datos antes descritos se procedió a crear la base de datos.

 **mysql>create database agenda;**

Se accedió a la base.

 **mysql>use agenda;**

Se crearon las tablas.

 **mysql>create table nombres(cont int default '0' not null auto\_increment, nombre text, email char(60), telefono char(15), edad char(3), primary key (cont), unique id (cont));**  
**mysql>table created;**

Se procedió a insertar algunos datos en la tabla recién creada.

 **mysql>insert into nombres(nombre, email, telefono, edad) values("Miguel", "mamendez@grupolala.com", "26224110", "28");**

Se realizaron algunas consultas a dicha tabla

 **mysql>select \* from nombres;**

 **mysql>select \* from nombres where cont='1';**

 **mysql>select \* from nombres where edad='28';**

Hasta este punto se tenía instalado el servidor MySQL y propiamente configurado, pero el objetivo del curso era conjuntarlo con otras herramientas aprendidas en el diplomado como son Apache y PHP, para lo cual se llevó a cabo lo siguiente:

- Se compiló nuevamente Apache

```
$ tar zxvf httpd-2.0.54.tar.gz
$ cd httpd-2.0.54
$ ./configure --prefix=/usr/local/apache2
$ make
$ make install
```

- Se compila nuevamente PHP y se configura para que funcione con MySQL como manejador de bases de datos

```
$ tar zxvf php-5.0.4.tar.gz
$ cd php-5.0.4
$ ./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-mysql
$ make
$ make install
$ cp php.ini-dist /usr/local/lib/php.ini
```

- Editar httpd.conf y agregar las siguientes líneas:

```
LoadModule php5_module modules/libphp5.so
AddType application/x-httpd-php .php .phtml
AddType application/x-httpd-php-source .phps
```

El archivo de configuración de PHP, php.ini, se tiene que modificar en algunas líneas para la conexión y transferencia de datos de MySQL.

- Se buscan las siguientes líneas y se modifican:

```
safe_mode=On
Data Handling
register_globals=On           el valor original es Off
register_argc_argv=On
```

Hechos los cambios, se reinician los servicios y se realizan las pruebas necesarias para verificar su funcionamiento. Se diseñó un programa en PHP que realizaba una conexión a la base de datos llamada agenda y solicitaba información de la tabla nombres, el script es el siguiente:

```
<html>
<body>
<?php
$link = mysql_connect("localhost", "nobody");
mysql_select_db("agenda", $link);
$result = mysql_query("SELECT * FROM nombres", $link);
echo "Nombre: ".mysql_result($result, 0, "nombre")."<br>";
echo "Telefono: ".mysql_result($result, 0, "tel")."<br>";
echo "Email :".mysql_result($result, 0, "email")."<br>";
echo "Direccion :".mysql_result($result, 0, "dir")."<br>";
?>
</body>
</html>
```

Como proyecto final se quedó abierto a algún sistema propuesto por cada participante, para lo cual se implementó para este módulo el proyecto creado en el módulo de PHP, el cual se describió en el capítulo 1.5, se adaptó para que los datos se manejaran por medio de una base de datos y no por medio de archivos de texto como se manejaron en el anterior proyecto. Las pantallas de acceso y presentación de datos mantuvieron el mismo esquema y diseño.

La seguridad en cómputo es primordial en todo sistema, durante el diplomado se trataron temas de seguridad en el sistema y en las aplicaciones y sistemas desarrollados, a continuación se verán algunas características de seguridad computacional y como pueden ser aplicadas al entorno de desarrollo de sistemas con software libre.

## **1.7 Introducción a la Seguridad en Cómputo.**

### **Objetivo general.**

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

### **Objetivo específico.**

El participante reconocerá la importancia de la seguridad e identificará los elementos que le permitirán proteger el sistema y la información.

### **Informe del módulo VII.**

Con el desarrollo de las redes, locales y globales, la seguridad toma un lugar muy importante en la computación. Las telecomunicaciones y la computación se han venido uniendo cada vez más. Las compañías telefónicas utilizan, cada vez más, sistemas computacionales en las estaciones telefónicas, mientras que cada día se conectan mas computadoras a la red. Si bien las telecomunicaciones ofrecen muchas ventajas, también es cierto que crean los principales problemas de seguridad.

En la actualidad con el auge de Internet y del comercio electrónico, existe una gran cantidad y diversidad de sistemas conectados a Internet, entre los que están servidores, estaciones de trabajo y PC's. En este entorno cada sistema es una victima potencial de un ataque. Existe también una gran diversidad de sistemas operativos en todos estos sistemas y una buen parte de servidores en Internet esta corriendo alguna distribución de Linux o alguna versión de UNIX.

La velocidad con que ha evolucionado y crecido Internet y la amplia variedad de servicios que se proporcionan a través de ella, han provocado que muchas veces no se le de a la seguridad la importancia adecuada en el desarrollo de sistemas que estarán disponibles por la red. Linux mismo no fue planeado con la seguridad como una meta fundamental, pero cuenta con los elementos para poder implementar mecanismos de seguridad.

En este módulo se manejó la importancia que tiene la seguridad en un sistema Linux, se analizaron los problemas que puede provocar un ataque o una intrusión en este tipo de sistemas, se comprobó la eficacia de Linux en comparación a otros sistemas operativos comerciales, los problemas típicos que pueden presentarse en cuestiones de seguridad son los siguientes:

- Negación de servicio
- Robo de contraseñas
- Necesidad de un análisis forense
- Reinstalación y/o reconfiguración del sistema

Todos los puntos anteriores fueron estudiados en el módulo, se analizó que estos implican tiempo del administrador y tiempo durante el que no se proporcionan los servicios para los que esta destinado el sistema con las consecuencias que esto implica hacia los usuarios, como la pérdida de confianza, y de información.

También existe variedad entre los atacantes o los intrusos y la razón por la que esto sucede.

Los atacantes pueden ser adolescentes curiosos “sobre cuestiones de seguridad en los sistemas conectados a Internet”, o bien pueden ser personas que si tienen como objetivo provocarle algún daño a la organización o pretenden robar datos del sistema, pueden ser intrusos que quieren usar el sistema como puente para lanzar ataques a otros sistemas que son su objetivo principal, o puede tratarse de crimen organizado, espionaje y demás.

Lo que se tiene que tomar en cuenta es que la mayoría de los ataques a sistemas de cómputo son ataques internos. Es decir, son ataques realizados por alguna persona que pertenece a la organización o bien que ha dejado la organización y lo hace con algún animo de venganza.

A menudo se usa el termino hacker para referirse a un intruso o un atacante, sin embargo, el uso de esta palabra es erróneo, ya que existen algunos otros términos que definirían mejor a un atacante o intruso, tal es el caso de cracker, cuyo término es empleado para definir a aquel intruso que accede a algún sistema para provocar algún daño. Otro término utilizado también es lamer que define a un individuo que no tiene conocimientos amplios sobre los sistemas de cómputo, sino que solo limita.

Hoy en día, los ataques son cada vez más frecuentes. Los motivos de los ataques varían desde aquellos que irrumpen en un sistema para probar solamente que lo pueden hacer, hasta el espionaje y el sabotaje. Los daños causados también varían desde el simple detalle de detectar un intruso hasta la pérdida invaluable de información y de dinero.

A lo largo de la historia de Internet, ha habido una gran cantidad de problemas de seguridad en los sistemas de cómputo, algunos de los cuales han sido muy famosos ya sea por la dimensión del daño que provocaron o bien por el renombre de la organización víctima del ataque o intrusión.

Actualmente los ataques más frecuentes son:

- ^ Irrumpir en un servidor de WWW y modificar sus páginas
- ^ Sabotaje en máquinas a través del ataque SYN, en el cual las máquinas no pueden operar. Es decir, no permiten que el sistema proporcione el servicio que debería.
- ^ Utilizar las máquinas atacadas como servidor de FTP. Poniendo, generalmente, software pirata o contenido pornográfico.
- ^ Robo de información.
- ^ Uso gratuito e indebido de los recursos.
- ^ Ataques a equipos caseros convirtiendo a estos en máquinas “zombies” para comenzar un ataque masivo de tipo DOS (Negación de servicio) contra algún servidor de WWW (caso concreto al sitio de Microsoft <http://www.microsoft.com>), mediante el uso de gusanos y troyanos, para envío de spam (correo masivo no deseado) y correos de phishing (fraude por Internet).

Los huecos que se suelen utilizar son:

- ^ Cuentas sin contraseñas.
- ^ Explotación de vulnerabilidades en los sistemas
- ^ Uso de software P2P (Persona a Persona - Person to Person) para bajar y compartir música de Internet.
- ^ Sitios de diarios en línea gratuitos (Blogs) ya que ofrecen grandes cantidades de almacenamiento gratuito, utilizándolos para colocar código viral o software keylogging, utilizando el spam o la mensajería instantánea, para atraer una gran cantidad de usuarios.
- ^ Cuentas con contraseñas débiles (principalmente las del sistema).
- ^ Uso de sniffers (programas que monitorean la red para atrapar las contraseñas u otra información valiosa).
- ^ Huecos de seguridad reportados en el CERT (Equipo de Respuesta a Emergencias de Cómputo – Computer Emergency Response Team), o en BBS underground, sendmail es el programa que por si solo cuenta con más huecos.
- ^ Aprovechar una mala configuración de permisos de algún sistema de archivos.

Se define a continuación el concepto de seguridad en cómputo analizado en el curso.

No existe una definición de seguridad en cómputo o seguridad computacional universalmente aceptada. Esto se debe a que es un tema relativamente nuevo y un tanto ambiguo al momento de definir sus fronteras. La siguiente definición es una de las más aceptadas.

*Un sistema de cómputo es seguro si se puede confiar en que se comportará como se espera que lo haga.*

Esto quiere decir que la seguridad del sistema depende de lo que se espera de él. Cada centro de cómputo tiene distintos sistemas, con distintas aplicaciones, necesidades y políticas de uso. En algunos sistemas, como puede ser un servidor FTP anónimo, en el que la información puede ser leída por todo el público se puede considerar seguro, mientras que en una máquina que mantiene información sobre una fórmula para hacer bombas no se puede considerar segura. De manera que antes de definir lo que es la seguridad de un sistema en particular, hay que definir claramente que es lo que se espera de él.

La seguridad en cómputo no se restringe a los accesos externos ilegales. De hecho, la mayoría de los problemas de seguridad son internos como se menciono anteriormente. Por ejemplo, casi siempre se espera que la información que almacena un usuario permanezca inalterada. Si se borra o modifica dicha información, se tiene un problema de seguridad, independientemente de que haya sido eliminada o alterada por algún intruso o por un error del administrador, o un problema de software o de hardware.

En el curso se resaltaron algunos tipos de seguridad, para entender un poco más a fondo que es seguridad computacional, resaltando que existen otros tipos de seguridad de los que aquí se presentan, como la seguridad física.

- Confidencialidad, el objetivo principal es proteger la información para que no pueda ser leída o copiada por personas no autorizadas. La información confidencial no se debe poder obtener a través de ningún método.
- Integridad de los datos, se refiere a que la información no se modifique o se borre, ya sea por un error de software, hardware o humano o por alguien mal intencionado sin el permiso del dueño.
- Autenticación, consiste en comprobar una identificación, para comprobar que el usuario es quien dice ser.
- Disponibilidad, consiste en que la información pueda ser accedida en el momento y con la velocidad requerida. De nada sirve mantener la información confidencial e íntegra si no se puede acceder a ella en el momento que se necesita.
- Control, consiste en regular quien tiene acceso al sistema. El objetivo es evitar que no entren al sistema personas no autorizadas.
- Auditoria, consiste en mantener registros que ayuden a averiguar quien hizo qué.

Se concluye que cada centro de cómputo debe definir, de acuerdo a sus necesidades, el orden de prioridad de estos tipos de seguridad, el administrador debe conocer los requerimientos y necesidades de su centro de cómputo e identificar cuales son las prioridades.

Lo ataques hoy en día son mucho más sofisticados y se usan herramientas y técnicas cada vez más complejas además de que ahora atentan contra la integridad y bienes de cada persona, y brindan una perspectiva diferente a lo que antes se entendía como seguridad computacional, la tecnología ya no es suficiente, la seguridad informática corre ahora por cuenta del usuario, quien tiene que generar conciencia, porque es el ahora, el responsable de su propia protección.

En general, el concepto de antivirus es reconocido por los usuarios; la mayoría de las personas ubican que un virus es algo malo, incluso aquellas personas que no utilizan una computadora. Pero existen otras amenazas cuyo daño se extiende más allá de un aparato. Además de los virus, el correo no deseado (spam) es un problema que afecta la productividad de las empresas. Los correos saturados de mensajes con publicidad innecesaria no son tan inocentes como aparentan porque en su mayoría pueden estar cargados con software espía (spyware) para robar datos confidenciales.

El usuario ya no debe confiar en todo lo que le llega a su buzón de correo electrónico, ya que puede generar consecuencias lamentables porque, aparte del robo de información, el exceso de correos bloquea la entrada o niega el servicio a sus contactos personales, sin descartar la posibilidad de adquirir malware que pueda contener más de 25 ataques en un solo archivo adjunto.

A continuación se mostrará una gráfica que muestra lo que más temen los usuarios en referencia a la seguridad informática personal.

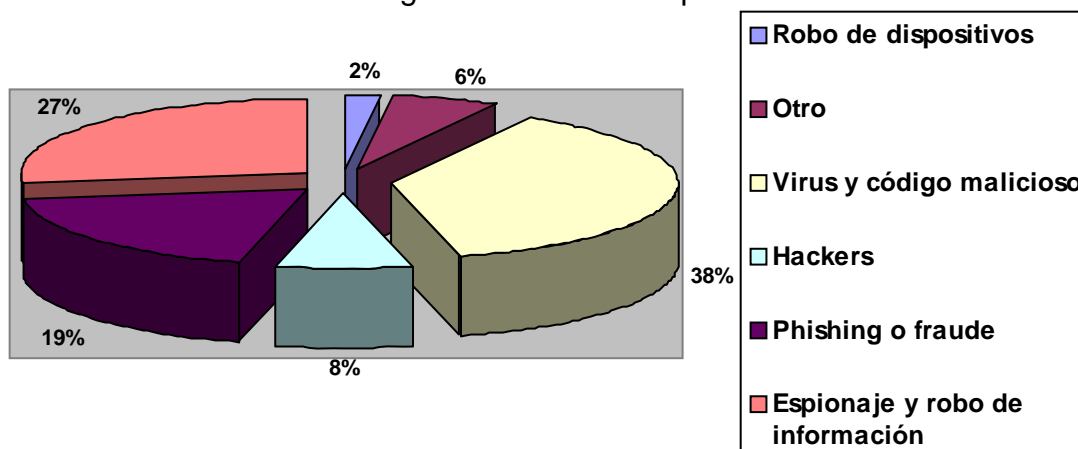


Figura 18. Fuente: Resultados de encuesta aplicada en el portal Netmedia.info a 130 usuarios



Como se observa, el panorama es poco alentador, pero como se analizó durante el curso, esto se puede resolver con algunos consejos prácticos que hay que seguir y en su mayoría sólo requieren de tiempo, a continuación algunos consejos que se manejaron durante el curso:

- Mantener la computadora actualizada. Lo más recomendable es tener equipos que lo hagan automáticamente, pero aun así hay que estar pendiente de que esas actualizaciones anti malware se haga constantemente. Hay que recordar que los bichos evolucionan a diario.
- Utilizar las herramientas. Si se cuenta con un firewall o antivirus, entre otros, se recomienda encenderlos, ya que no solo son artículos de decoración en el site.
- Rechazar los mensajes desconocidos. Las fallas humanas siempre sobre pasan a las técnicas, muchas veces el mismo usuario es el que abre la puerta a estos bichos al dar clic a mensajes cuyo remitente es desconocido.
- Borrar toda la basura. Los buzones saturados pueden traer verdaderos problemas incluso la pérdida de información importante.
- Antes de comprar, hay que informarse. El hecho de adquirir miles de herramientas de seguridad por moda no es recomendable, por lo tanto antes de proceder a la compra se tendrá que analizar lo que se necesita y conforme a ello, establecer el pedido.
- Respaldo de información. Una computadora de escritorio casera tiene el mismo valor que una de un gran corporativo porque la información de cada uno no tiene precio por muy compleja o sencilla que sea, nadie puede ponerle valor. La mayoría de los usuarios no sabe realizar un respaldo de su información.
- El Filtrado de contenido. Filtrar contenido y trabajar con un sistema detector de intrusos es una practica muy buena, que evitará los molestos problemas que puede ocasionar cuando algún intruso se infiltre a la red.
- Encriptación de información delicada. Otra manera más de proteger la información delicada critica ya que solo el usuario emisor y el usuario destinatario tendrán las llaves de acceso para abrir el mensaje.
- Si es pública no se usa. Evitar hacer transacciones bancarias o de tipo confidencial en una PC pública (café Internet, computadora de la escuela).
- Verificación del uso de los portales. Es recomendable que antes de hacer banca en línea se verifiquen cuales son las medidas de seguridad que ofrecen los bancos, como los sistemas anti-intrusos. Los bancos tienen algunos dispositivos especiales para evitar los fraudes como los OTP (Password de una Sola Vez – One Time Password) que dan una contraseña con vigencia de un minuto, Tokens que se componen de una firma digital que se conecta a la computadora o las tarjetas de coordenadas que ya no son tan funcionales.

La parte que corresponde estudiar ahora es sobre como proteger al sistema, este proceso fue analizado durante el curso y se comentó que para asegurar el sistema se debe de tener en cuenta el objetivo principal de llevar a

cabo dicha acción, se deben minimizar y analizar los riesgos potenciales de seguridad, analizar también las amenazas potenciales que se pueden sufrir, así como las pérdidas que se pueden generar y la probabilidad de su ocurrencia, su análisis no sólo lleva a establecer un nivel adecuado de seguridad, sino que permite conocer mejor el sistema que se va a proteger, se tiene que tomar en cuenta el diseño de políticas de seguridad y definir responsabilidades y reglas a seguir para evitar tales amenazas o minimizar sus efectos en caso de que se produzcan.

Para llevar a cabo dicha implementación se deben usar mecanismos de seguridad. Se convierten en la herramienta básica para garantizar la protección de los sistemas o de la propia red. Se dividen en: detección, prevención, recuperación.

Si no se cuenta con un plan adecuado pueden existir problemas serios, pérdidas representativas como pérdida de clientes, pérdida de imagen, pérdida de ingresos por beneficios, pérdida de ingresos por ventas y cobros, pérdida de ingresos por producción, pérdida de competitividad, pérdida de credibilidad en el sector.

Se presenta una gráfica que representa todo lo mencionado anteriormente.



Figura 19 Mecanismos de seguridad empleados

Durante el curso se estudiaron algunas herramientas para entender mejor aun los conceptos vistos en clase.

La creación de políticas de seguridad tiene como fin el aterrizar una estrategia integral de seguridad, establecer e implementar políticas para garantizar el cumplimiento adecuado de la estrategia. Las políticas son la parte que atañe al recurso humano y a la administración de sistemas, y es el eslabón que permite

que la infraestructura física y de sistemas funcionen y no sean vulneradas por un error de recursos humanos o de procesos.

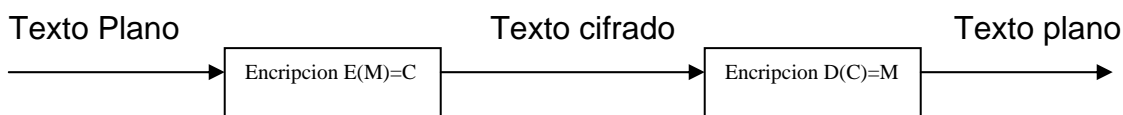
Es el marco que sirve de apoyo para contar con un sistema confiable. Una política de seguridad se enuncia normalmente en términos de sujetos y objetos. Un sujeto es algo activo en el sistema; usuarios, procesos y programas. Un objeto es algo que esta en función a la acción de un sujeto; archivos, directorios, dispositivos, etc.

Una política de seguridad es un código de conducta para usarse en un sistema de cómputo. Esta política detalla las actividades que están o no permitidas, los pasos a seguir para obtener la protección adecuada, así como los pasos a seguir en caso de presentarse un incidente de seguridad, además establece responsabilidades y derechos. Una política de seguridad explica las sanciones que se impondrán a los que infrinjan una regla.

Escribir una política de seguridad no es suficiente. El paso más importante es hacer que cada usuario del sistema este enterado de dicha política. La manera más fácil de hacer que cada usuario del sistema se entere de las políticas es entregarles una copia de las mismas al momento de recibir su cuenta en el sistema.

Durante el curso, se manejaron otros conceptos de seguridad computacional, como el concepto de criptología, que es la ciencia que estudia los aspectos y contenidos de información en condiciones secretas. La criptología se divide en criptografía, que es el *arte* de construir códigos secretos mediante un conjunto de técnicas o procedimientos que alteran los símbolos de información sin alterar el contenido, convirtiendo a la información modificada en un conjunto de símbolos sin contenido para las partes que no disponen de las técnicas (receptor del mensaje) y criptoanálisis, que son las metodologías y técnicas que permiten recuperar la información que ha sido previamente tratada por un procedimiento criptográfico, sin conocer la técnica utilizada para la criptografía.

Otro concepto manejado y que es de gran importancia es el cifrado, que se define como un algoritmo criptográfico empleado para ocultar el mensaje en “texto plano”. Es la función matemática usada para la encripción y decodificación del mensaje original. El cifrado se usa en criptosistemas mediante un proceso de encripción-decripción del mensaje original llevándose a cabo de la siguiente manera.



Donde M es el mensaje en texto plano, E es el método de encriptación, C es la clave que se obtiene en el criptosistema y D es el método de decriptación. El procedimiento es sencillo, al encriptar un mensaje en texto plano mediante un criptosistema, se obtiene una llave o clave, que se convierte en el texto cifrado y así viaja por el medio hasta llegar a su destino, en donde se realiza la decriptación de la clave mediante un criptosistema, para obtener el mensaje original en texto plano.

De esto se puede concluir que la criptografía tiene como objetivos principales mantener la confidencialidad del mensaje, es decir que la información contenida en el mensaje permanezca secreta, para de esta manera garantizar la autenticidad tanto del mensaje, que sea realmente el mensaje enviado, como del par remitente/destinatario, que sean quienes dicen ser y no remitente/destinatario fraudulentos.


En la práctica la seguridad que ofrece un criptosistema consiste en mostrar que *“cualquier ataque que tiene una probabilidad de romper la llave requiere de una cantidad infinita de computación”*. Un sistema criptográfico se dice *inseguro* cuando los contenidos de encriptación pueden ser descifrados en un tiempo polinomial.

Para comprender el funcionamiento de las llaves para autenticación, se instalaron algunas herramientas para ejemplificar dichos conceptos, como a continuación se muestra una aplicación para realizar firmas digitales de autenticación.


## GNU PG


Gnu Privacy Guard (GnuPG o GPG) es un sistema de codificación de código libre y desarrollo abierto. Es un reemplazo libre a PGP (*Pretty Good Privacy*) originalmente creado por Phil R Zimmermann.

Para instalar la herramienta se lleva a cabo lo siguiente:

 Descargar el archivo *gpg* de la página de gnupg:  
[http://www.gnupg.org/\(es\)/download/index.html](http://www.gnupg.org/(es)/download/index.html)

 Descompactar el archivo  
`tar -zxvf gnupg *.tar.gz`

 Configurar el archivo  
`./configure --prefix=/usr/remote/gpg`

 Generar archivos previos de configuración  
`make`

- Instalar paquete gnupg

```
make install
```

- Generar una referencia en el entorno de configuración del usuario para que se ejecute gpg sin importar donde se esté ubicado en el sistema

```
PATH = /usr/remote/gpg/bin : $PATH
```

Para empezar a utilizar *gpg*, es necesario generar un usuario cualquiera, y cambiarse de sesión

```
su - usuario
```

Una vez dentro del home del usuario se ejecuta el comando

```
gpg
```

La ejecución del comando anterior genera un nuevo par de llaves (privada y pública) para el usuario. Ahora se inicia la creación de la llave

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

Este comando despliega una serie de datos que conforman la generación de dicha llave:

**Selection → 1**

**Keysize de 1024 a 4096 bits → 2048**

**Key is valid for ( 0 ) → 10 (días) Periodo de Validez para que tendrá la llave generada**

**Correct → Yes Datos Correctos.**

**ID:**

**Real Name : Miguel Angel Mendoza\_\_**

**e-mail : mamendoz@grupolala.com**

Se ejecuta el siguiente comando que muestra las llaves con las que cuenta el “llavero”.

```
gpg --list-keys
```

Se exporta la llave pública del usuario a un archivo.

```
gpg --export --armor > MiLlave.key
```

Se importa la llave pública de otro usuario

```
gpg --import < LlaveDeUnAmigo.key
```

Se firma la llave pública de otro usuario y se considera confiable.

```
gpg --sign-key <id>
```

Para enviar mensajes cifrados con esta herramienta a algún determinado usuario.

Se cifra un Mensaje.txt con la llave de un usuario.

**gpg -e Mensaje.txt**

Se descifra el Mensaje.txt con la llave privada del usuario.

**gpg -d Mensaje.txt.gpg**

El uso de esta herramienta garantiza que se cumplan los objetivos principales de la criptografía, su uso es sencillo, práctico y fue muy ilustrativo, el uso de este tipo de herramientas disminuye significativamente el robo de información por identidad y garantiza que la información enviada sea confidencial, pero involucra tanto a remitentes como destinatarios, además de que cada usuario tendrá la responsabilidad de llevar a cabo estas prácticas para que el robo de información y abuso de usuarios fraudulentos sea menor, no se menciona que vayan desapareciendo, porque cada día inventan nuevas técnicas, pero si garantizar una transacción de información confiable al menos entre el par que emplea esta herramienta.

Otra herramienta utilizada en el curso utilizada para crear cifrado de datos por bloques fue openssl, se manejo un módulo adicional que trabaja en conjunto con Apache mod\_ssl, que permite implementar un socket seguro en el servidor de WWW. La instalación, configuración y empleo se verá a continuación.

El proyecto OpenSSL es un desarrollo de código abierto con el propósito de desarrollar un Toolkit de SSL y TLS con el nivel de una herramienta profesional. El proyecto está basado en el código fuente de SSLeay desarrollado por Eric. A Young y Tim J. Hudson. Para mayor información <http://www.openssl.org/>

• Obtener una copia del software

<http://www.openssl.org/source/openssl-0.9.7e.tar.gz>

• Configurar la aplicación

**./configure --prefix=/usr/remote/openssl**

• Compilar para la creación de los archivos binarios

**make**

• Ejecutar la instalación de la aplicación.

**make install**

Una vez instalado el software se procedió a realizar algunos ejemplos de cifrado de mensajes ejecutando el siguiente comando:

```
openssl enc -e -des3 -in Ejemplo.txt -out Ejemplo.des3
```

Para descifrar los mensajes se aplica el siguiente comando:

```
openssl enc -d -des3 -in Ejemplo.des3 -out Ejemplo.txt
```

La utilización de esta herramienta permite la creación de Algoritmos de Resumen del Mensaje 5 (Message Digest Algorithm 5 - MD5) que es un algoritmo de reducción criptográfico de 128 bits ampliamente usado en la criptografía, y se muestra a continuación:

Para crear un hash MD5 (huella digital, que es una salida producida por una función hash aplicada a un documento de dicho documento. Cualquier cambio en el documento produce una huella diferente) de algún archivo determinado.

```
openssl dgst -md5 Ejemplo.txt  
MD5(Ejemplo.txt)= 08a4ebfb7bf8f68a4e169a9806cb3a3b
```

Se estima que es imposible obtener dos mensajes con la misma huella digital, también es imposible producir un mensaje que arroje una huella predefinida.

Verificando la huella digital concerniente a determinado archivo:

```
md5sum Ejemplo.txt  
08a4ebfb7bf8f68a4e169a9806cb3a3b Ejemplo.txt
```

Con openssl se pueden crear mensajes cifrados utilizando SHA (Secure Hash Algorithm, Algoritmo de Hash Seguro) que es un sistema de funciones hash criptográficas relacionadas de la Agencia Nacional de Seguridad estadounidense y publicadas por el National Institute of Standards and Technology (NIST). Hoy en día se le conoce como SHA-0 para prevenir equivocaciones con los demás algoritmos de su familia.

Para crear un hash tipo SHA en openssl se ejecuta el siguiente comando:

```
openssl dgst -sha Ejemplo.txt  
SHA(Ejemplo.txt)= 37ad34ca6b2d8447a7504b3c76aa7d8bb319644f
```

En openssl se generan llaves de tipo RSA, que es un sistema criptográfico con clave pública y recibe su nombre por la inicial del apellido de sus inventores:

Ronald Rivest, Adi Shamir y Leonard Adleman. Todo usuario de dicho sistema hace pública una clave de cifrado y oculta una clave de descifrado. Una llave es un número de gran tamaño, que una persona puede conceptuar como un mensaje digital, como un archivo binario o como una cadena de bits o bytes. Cuando se envía un mensaje, el emisor busca la clave pública de cifrado del receptor y una vez que dicho mensaje llega al receptor, éste se ocupa de descifrarlo usando su clave oculta. Los mensajes enviados usando el algoritmo RSA se representan mediante números y el funcionamiento se basa en el producto de dos números primos grandes (mayores que  $10^{100}$ ) elegidos al azar para conformar la clave de descifrado. La seguridad de este algoritmo radica en que no hay maneras rápidas de factorizar un número grande en sus factores primos utilizando computadoras tradicionales. RSA es considerado como un criptosistema de bloque

Generando una llave RSA de 128 bits.

**openssl genrsa 128**

Generando una llave RSA de 128 bits cifrada con 3DES.

**openssl genrsa -3des 128**

Para generar una llave DSA de 1024 bits cifrada con 3DES se lleva a cabo lo siguiente:

Primero se generan los parametros que DSA requiere:  
**/usr/remote/bin/openssl dsaparam -out parametros.pem 128**

Con los parametros generados se realiza la llave:  
**/usr/remote/bin/openssl gendsa -out privkey.pem parametros.pem**

De esta manera se obtiene una llave privada con los parámetros establecidos por DSA.

Como se mencionó anteriormente, openssl cuenta con módulo para trabajar en conjunto con Apache y crear sockets de conexión seguros. El modulo mod\_ssl proporciona el soporte necesario para implementar SSL sobre HTTP.

A continuación se describirá el proceso de instalación y configuración de la herramienta, en conjunto con Apache en su versión 2.0.52, así como la generación de certificados y llaves privadas que garantizan una conexión y transacción de datos segura entre el cliente y el servidor al momento de validar datos personales como números de créditos de tarjetas, contraseñas de usuario en sistemas seguros y demás.



Para Apache 2.0.52:

```
gzip -dc httpd-2.0.52.tar.gz | tar xvf -  
cd httpd-2.0.52  
./configure --prefix=/usr/remote/apache --enable-so --enable-ssl --with-  
ssl=/usr/remote/openssl  
make  
make install  
apachectl start  
apachectl stop  
/usr/remote/apache/bin/httpd -l | grep mod_ssl
```

Para generar el certificado necesario para realizar la comunicación vía HTTPS se debe realizar:

Se genera la llave privada

```
openssl genrsa -out server.pem 2048
```

Se genera un certificado digital (Certificate signing request), que es un documento que asienta que una llave pública y su correspondiente llave privada pertenecen a un individuo en particular, certificando de esta manera la identidad de dicho individuo. Tiene el propósito de hacer disponible a otras personas una llave pública personal, el cual se enviará a la Autoridad Certificadora (CA), que son entidades responsables de certificar la identidad de un individuo y su posesión de una llave pública, estas generan y administran certificados y los publican en repositorios.

```
openssl req -new -key server.key -out cert.csr
```

O si se utiliza un certificado auto firmado, se realiza:

```
openssl req -new -x509 -key server.key -out cacert.pem -days 30
```

El uso de este software permitió entender el concepto de emitir información de manera segura, el uso de las firmas y de los certificados digitales hoy en día es muy amplio, son empleados para una gran variedad de transacciones electrónicas como correo electrónico, comercio electrónico, trabajo en grupo y transferencia de fondos electrónicos. Como por ejemplo, la "Enterprise Server" de Netscape necesita un Certificado Digital para cada uno de los servidores seguros. Así, un cliente que realiza una compra en una tienda virtual a través de un servidor de software de Netscape solicita el Certificado Digital del servidor para verificar la identidad de la tienda y el contenido ofrecido por ésta.

Sin la verificación del servidor, el comprador no confiará al vendedor información confidencial, como el número de la tarjeta de crédito. El Certificado

Digital es un instrumento que establece un canal seguro para comunicar información confidencial al operador de comercio electrónico.

Por otro lado, las tiendas virtuales, la banca electrónica y otros servicios electrónicos se están convirtiendo en lugares de encuentro que ofrecen al cliente un servicio flexible y personalizado 24 horas al día, directamente en su casa. No obstante, la preocupación acerca de la privacidad y la seguridad puede limitar el aprovechamiento de estas nuevas formas de intercambio. Por sí sólo, el sistema de encriptación no es suficiente, porque no ofrece prueba alguna sobre la identidad del remitente que envió la información encriptada. Sin ninguna garantía especial, se arriesga a que un tercero se haga pasar por otro en línea. Los Certificados Digitales solucionan este problema, ofreciendo un medio electrónico para verificar la identidad de una persona. Al utilizar los Certificados Digitales en conjunto con el sistema de encriptación, se proporciona una solución más completa para la seguridad, asegurando la identidad de todas las partes involucradas en una transacción.

Igualmente, un servidor seguro debe tener un Certificado Digital para asegurar que el servidor está administrado por la organización que dice poseerlo y para asegurar que el contenido que ofrece es auténtico.

Se concluye que al aplicar lo analizado durante el curso en la práctica, se brinda un panorama claro del funcionamiento de la emisión de certificados digitales para transacciones seguras, se garantiza que la información enviada viaje de forma segura y confidencial, y que esta sea enviada por quien dice ser y que sea recibida por un destinatario autenticado.

Durante el curso se hizo hincapié de tener accesos seguros y garantizados a los servidores y servicios en línea llevándose a cabo mediante el uso de programas de conexión seguros, como es el caso de SSH (Secure Shell – Shell Seguro), que es un programa para acceder a otra computadora a través de una red, se utiliza para ejecutar comandos en una máquina remota (interprete de comandos), o bien mover archivos de una computadora a otra. Este programa esta disponible para varias plataformas y es el sustituto ideal para otros clientes de conexión remota como telnet, ftp, rlogin, rsh, rcp y rdist. Provee una fuerte autenticación y comunicaciones seguras por medio de encriptación sobre canales inseguros.

El procedimiento de autenticación de usuario usando ssh es descrito a continuación:

Puede ser autenticado de diferentes formas.

- Dialogo dirigido por el cliente. Se envían peticiones al servidor.
- Primera petición: siempre solicita al usuario su login name.
- Servidor responde peticiones: un éxito o falla.

Los métodos soportados actualmente son:

- autenticación de password tradicional
- combinación de autenticación .rhost con RSA basada en host
- autenticación RSA pura
- autenticación kerberos V5 y autenticación servidor TIS
- se incluye soporte para otros métodos

Se estudiaron también las diferencias entre SSH y SSL. SSH corre sobre la capa de aplicación del modelo OSI y su método de autenticación son llaves, mientras que SSL se ejecuta en la capa de transporte y su método de autenticación son certificados.

Otro concepto estudiado y analizado durante el curso fue esteganografía, que es una área similar a la criptología, que es un conjunto de técnicas que permiten ocultar o camuflar cualquier tipo de datos, la información (documentos electrónicos) puede esconderse dentro de imágenes utilizando los bits menos significativos (espacios en blanco), o también aprovechar los campos no usados de los paquetes de protocolos de redes. Una de las técnicas más complejas usadas en la esteganografía en textos es la denominada “Null Cipher – Cifra Nula”, a continuación un ejemplo del uso de esta técnica:

El remitente envia un mensaje de este tipo

*Fishing freshwater bends and saltwater coasts rewards anyone feeling stressed. Resourceful anglers usually find masterful leapers fun and admit swordfish rank overwhelming anyday.*

Si se toma la tercera letra de cada palabra diría

- **Send Lawyers, Guns, and Money.**

Se hizo una comparativa entre la criptografía y la esteganografía de lo cual se resume lo siguiente:

Cuando se encripta un mensaje, el mensaje encriptado da sospechas de que algún tipo de información confidencial esta viajando en la red.

En la esteganografía el mensaje nunca dará sospechas ya que puede viajar en imágenes totalmente inofensivas. Desafortunadamente, muchos individuos ocupan la esteganografía con fines meramente delictivos o de espionaje, utilizándola principalmente para el tráfico de información confidencial y la pornografía infantil.

En la esteganografía la imaginación de las partes involucradas es la única limitante.

Lo ideal es enviar el mensaje oculto por medio de esteganografía y aparte ese mensaje debe ir encriptado.

La constante evolución de las amenazas en Internet, plantea el enorme reto de educar no solo a los empleados, sino también a todos los usuarios de computadoras conocidos, para comenzar a cimentar una cultura de seguridad computacional sólida en todos los niveles. Que cada usuario tenga conocimiento que el primer paso para mejorar la seguridad en un sistema es analizar los riesgos respondiendo las siguientes preguntas básicas.

Identificación de recursos

• ¿Qué se está tratando de proteger?

Identificación de amenazas

• ¿Qué se necesita para protegerlo?

Cálculo de riesgos.

• ¿Cuánto tiempo, dinero y esfuerzo se está dispuesto a destinar para obtener un nivel adecuado de protección?

Estas tres preguntas forman la base del proceso conocido como “Análisis de riesgos”. Que permitirán al usuario establecer sus propias políticas de acceso y uso de su sistema o bien del sistema usado en su trabajo, y garantizar un acceso y uso confiable de sus recursos.

A continuación se expondrá una historia ficticia basada en el clásico cuento de Caperucita y el Lobo Feroz con un tono ligero pero con un claro enfoque de seguridad en la información.

*Había una vez una usuaria de Internet, llamada K-perucita, que estaba con sus amigos chateando en el canal #bosque del IRC. De pronto recibió un correo electrónico de sus mamá, que decía así: “Hija, te anexo unos archivos para el documento HTML de tu abuelita. Por favor, conéctate a su FTP y se los pasas para que ella pueda montar su página web”.*

*Y así, K-perucita, se dispuso a abrir una ventana de navegación, para subir al FTP de su abuelita los archivos que le habían mandado. Mientras descargaba los archivos desde su cuenta de correo web, le llegó un mensaje de ICQ, el usuario era*

*e-lobo@hacker.bosque.com.*

*K-perucita respondió el mensaje de ICQ y se entabló la siguiente conversación:*

*e-lobo: Hola K-perucita, ¿qué haces en la web? ¿cuál es tu destino?*

*K-perucita: voy a la cuenta de mi abuelita, a subir vía FTP un documento HTML para que monte su página web.*

*Y así, e-lobo hizo un atajo vía telnet y llegó a la cuenta de la abuelita primero. Cuando la cuenta de la abuelita le pidió su cuenta de acceso (login ID), el lobo tecleó "K-perucita", posteriormente, obtuvo la contraseña utilizando técnicas de crackeo; y así logró entrar al sistema.*

*La abuelita, al ver que no era K-perucita sino otra persona, trató de matar el proceso. Pero e-lobo fue más veloz: hizo un ICMP flood a los puertos que el firewall de la abuelita no estaba controlando, y cuando cayó le cambió la contraseña.*

*Luego se tomó privilegios de raíz o administrador (root) en la máquina, y cambió el sistema operativo por uno diferente, que se parecía en todo, hasta en la interfaz, al de la abuelita. Finalmente, se metió a la cuenta de la abuelita, y se hizo pasar por ella.*

*Poco después llegó K-perucita, y cuando entro, noto un poco cambiada la cuenta de su abuelita. Inmediatamente inició un proceso talk y le preguntó:*

*K-perucita: Abuelita, ¿por qué tienes esa cuota tan grande en disco?*

*e-lobo: es para almacenar mis archivos mejor.*

*K-perucita: Abuelita, ¿por qué tienes esa interfaz gráfica tan novedosa?*

*e-lobo: es para administrar mis archivos mejor.*

*K-perucita se dio cuenta que algo raro sucedía ahí, y preguntó:*

*K-perucita: Abuelita, ¿por qué tienes privilegios de root?*

*e-lobo: ¡Para crackearte mejor!*

*K-perucita se dio cuenta de que no se trataba de su abuelita... pero, ¿quién podría ser? Lanzó un proceso who is y descubrió que su "familiar" estaba conectado con la cuenta e-lobo@hacker.bosque.com.*

*Inmediatamente, mandó un correo a security@cyberspace.cop.org para delatar al impostor. Este trato de bloquear su servidor POP3 haciéndole un overload de memoria, pero K-perucita ya había dado clic en el botón de enviar.*

*Poco tiempo después, se conectó a la máquina uno de los investigadores de la policía cibernética, cyberspace.cop, que rápidamente obtuvo la dirección IP de e-lobo, le hizo un override a la máquina, se tomó privilegios de root y antes de que e-lobo se diera cuenta, el ciberpolicía mató el proceso y colocó un ban para todo el dominio. Del basurero del sistema operativo de e-lobo se pudo recuperar la tabla de partición del sistema de la abuelita, por lo que se logró salvar toda su información.*

*La abuelita, tuvo suerte: recuperó su trabajo y subió su página web a un promedio de 10 kb/seg de transferencia. El site fue admirado por todos en el ciberespacio recibiendo numerosos hits en poco tiempo.*

Como se observó en la historia anterior, el robo de identidades es común en usuarios de Internet, los usuarios se hacen pasar por terceros ocasionando robos o pérdidas de información, daños a sistemas, ataques programados etc., nadie está garantizado de estar 100% protegido, se tienen que establecer reglas y políticas propias de uso de los servicios disponibles en Internet y del mismo equipo, manteniendo actualizados los servicios de un software antivirus, manejo seguro de contraseñas y principalmente mantener actualizado el sistema de todas las vulnerabilidades que día con día aumentan.

El siguiente capítulo describirá el uso de más herramientas de software libre para la manipulación de datos, empleando PHP y PostgreSQL y algunos generadores de moldes para una administración mucho más sencilla.

## 1.8 Desarrollo de aplicaciones con Postgresql y PHP.

### Objetivo General.

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

### Objetivo específico.

El alumno manejará Postgresql en la construcción de bases de datos que funcionen a través del WWW, utilizará herramientas adicionales de generación de Templates y hará consultas sobre ellas.

### Informe del Módulo VIII

La primera parte del trabajo desarrollado durante el curso fue implementar el entorno de trabajo, lo que se hizo fue analizar cada una de las herramientas que serían utilizadas, para el desarrollo del curso.

El curso está enfocado al desarrollo de aplicaciones Web usando herramientas de software libre, al igual que en el módulo VI de MySQL, donde se estudio la arquitectura de trabajo LAMP (Linux, Apache, MySQL, PHP), durante este módulo se trabajo con la misma arquitectura, pero utilizando a PostgreSQL en lugar de MySQL, además de utilizar un par de herramientas más para el desarrollo de aplicaciones dinámicas, como es el caso de Smarty, que es un motor de plantillas para PHP. Específicamente, esta herramienta facilita la manera de separar la aplicación lógica y el contenido en la presentación, y de la cual se hablará más adelante. También se utilizo otra herramienta muy poderosa para realizar interacciones con la base de datos de manera sencilla y directa; ADODB.

Así como el uso de herramientas ya conocidas como es el caso del servidor de páginas Web Apache, y el lenguaje de programación para scripts de lado del servidor PHP. Todas estas herramientas en conjunto fueron la base para el desarrollo del curso.

Se comenzará a hablar sobre PostgreSQL. PostgreSQL es un servidor de bases de datos vía SQL de dominio público, es un manejador de base de datos objeto-relacional, altamente escalable, compatible con SQL y con mas de 15 años

de desarrollo, surge en la Universidad de Berkeley, basado en Ingres y toma el nombre de postgres bajo el liderazgo de Michel Stonebraker, en 1996 cambia de nombre de postgres95 a PostgreSQL. El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto PostgreSQL es distribuido bajo la clásica licencia BSD. No tiene restricciones acerca de la manera en que puede usarse el código fuente.

PostgreSQL puede ser ejecutado en distintas plataformas, en general, cualquier plataforma moderna compatible con Unix o Linux debería ser capaz de ejecutar PostgreSQL. Las plataformas que han sido probadas al momento de la distribución son listadas en las instrucciones de instalación.

PostgreSQL también corre nativamente sobre sistemas operativos basados en Microsoft Windows NT tales como Win2000, WinXP y Win2003. Se encuentra disponible un instalador preempacado <http://pgfoundry.org/projects/pginstaller>. Las versiones de Windows basadas en MSDOS (Win95, Win98, WinMe) pueden correr PostgreSQL usando Cygwin.

Para obtener una copia del software se puede obtener directamente de su sitio Web mediante <http://www.postgresql.org> o bien por medio de <ftp://ftp.postgresql.org/pub/>

Así mismo el soporte brindado para esta aplicación se puede encontrar en muchas listas comunitarias y el excelente manual documental que se encuentra disponible en el mismo sitio Web.

PostgreSQL incluye una amplia documentación, incluyendo un extenso manual, páginas man y algunos ejemplos. O bien, se pueden consultar los manuales en línea en <http://www.postgresql.org/docs>. Ahí hay dos libros de PostgreSQL disponibles en línea en <http://www.postgresql.org/docs/books/awbook.html> y <http://www.commandprompt.com/ppbook/>. También existen algunos libros de PostgreSQL disponibles para su compra. Uno de los más populares es de Korry Douglas. Una lista de revisiones de libros puede ser encontrada en <http://techdocs.postgresql.org/techdocs/bookreviews.php>. Además existe una colección de artículos técnicos de PostgreSQL en <http://techdocs.postgresql.org/>.

El uso de PostgreSQL hoy en día da pauta a una comparación con otros DBMS (Data Base Management System – Sistemas Manejadores de Bases de Datos). Existen varias maneras de medir el software: características, desempeño, fiabilidad, soporte y precio.



## Características

PostgreSQL tiene la mayoría de las características presentes en grandes DBMS comerciales, tales como transacciones, subconsultas, triggers, vistas, integridad referencial con llaves externas, y bloqueo sofisticado. También cuenta con algunas características que no tienen los otros DBMS, como tipos definidos por el usuario, herencia, reglas, y control de concurrencia multiversión para reducir el bloqueo de controversias.

### • Desempeño

El desempeño de PostgreSQL es comparable con el de otras bases de datos comerciales y de código abierto. Es más rápida para algunas cosas, más lenta para otras. El desempeño es usualmente +/-10% comparado con otras bases de datos.

### • Fiabilidad

Un DBMS debe ser fiable, o es inútil. Los desarrolladores de PostgreSQL procuran liberar código estable y bien probado que tenga mínimos errores. Cada versión tiene al menos un mes de pruebas beta, y la historia de liberaciones muestra que se pueden proveer versiones estables y sólidas que se encuentran listas para su uso en producción.

### • Soporte

Las listas de correo, proporcionan contacto con un gran grupo de desarrolladores y usuarios que ofrecen ayuda para resolver cualquier problema encontrado. El acceso directo a desarrolladores, la comunidad de usuarios, manuales y el código fuente suelen hacer que el soporte de PostgreSQL sea superior al de otras DBMSs. También hay soporte comercial disponible por incidente para aquellos que lo necesiten.

### • Precio

PostgreSQL está disponible para cualquier uso, ya sea comercial o no. Se puede agregar el código al producto sin limitación alguna, excepto aquellas mencionadas en el estilo de licencia BSD.

La interfaz de trabajo de PostgreSQL con algún lenguaje de programación a nivel servidor está integrada en su mayoría con lenguajes de tipo script como es el caso de PHP, Perl, Python, TCL. Al instalar PostgreSQL se incluyen solamente las interfaces de C y C incrustado. Todas las demás interfaces son proyectos independientes que son descargados por separado; el ser separado permite que

se tenga una agenda de liberación propia y equipos de desarrollo, como es el caso de PHP.

PostgreSQL esta ampliamente considerado como el manejador de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto rendimiento.

A continuación se detallará el procedimiento empleado para la instalación, configuración y puesta en marcha de un servidor de bases de datos con PostgreSQL, y se mostrará también la configuración del resto de aplicaciones utilizadas durante el curso.

La instalación de Postgresql se puede realizar de dos formas:

– “Sencillo y Practico”

Por paquetes de la distribución favorita.

– “En cualquier Unix o Linux”

Se obtiene una copia de la ultima versión del software del sitio oficial <http://www.postgresql.org>. Se continua con el proceso de compilación

- Se descomprime el archivo para obtener un directorio donde se ejecutará la instalación

**\$ tar zxvf postgresql-8.0.3.tar.gz**

- Una vez descomprimido el directorio se tendrá que posicionar dentro de el

**\$ cd postgresql-8.0.3**

- Se escriben los parámetros de configuración indicándole al sistema donde será instalada la aplicación

**\$ ./configure --prefix=/usr/local/pgsql**

- Se crean las dependencias y los archivos binarios de instalación

**\$ make**

- Se ejecuta la instalación

**# make install**

- Una vez terminada la instalación se procede a crear el usuario que manejará el proceso de carga de postgresql

**#adduser postgres**

- Se crea el directorio donde se colocarán las bases de datos  
**#mkdir /usr/local/pgsql/data**
- Se cambia de dueño al directorio  
**# chown postgres /usr/local/pgsql/data**
- Se cambia al usuario creado para manejar los procesos de postgresql  
**# su – postgres**
- Se inicia la base de datos indicando al sistema donde se encuentra situado el directorio de las bases de datos, cualquiera de las dos opciones aquí mostrada es válida

```
$ /usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
```

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

- Se crea una base de datos de prueba  
**\$ /usr/local/pgsql/bin/createdb test**

- Se accede a la base de datos recién creada para verificar su funcionamiento

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

PostgreSQL ha sido instalado en el sistema. A continuación se muestra la forma de acceder a postgresql y su pantalla de inicio desde línea de comandos.

Se accede al manejador con el usuario creado para tal fin:

```
$ su - postgres
```

```
$ createdb prueba
```

```
Welcome to the psql, the PostgreSQL interactive terminal.
```

```
Type \copyright for distribution terms
```

```
\h for help with SQL commands
```

```
\? For help on internal slash commands
```

```
\g or terminate with semicolon to execute query
```

```
\q to quit
```

```
prueba=#
```

Se crea una tabla de prueba para comprobar funcionamiento:

```
prueba=#  
create table MiTabla3 (  
idMiTabla integer,  
logico boolean,  
fecha date,  
punto point,  
circulo circle,  
direccionIP inet,  
cantidad double precision,  
dinero money,  
texto character varying (100) );
```

Los tipos de datos que soporta Postgresql 8.0, se pueden consultar en:  
<http://www.postgresql.org/docs/8.0/interactive/datatype.html>

A continuación se insertan datos en la tabla creada:

```
prueba=#insert into MiTabla3 values (1, true, '1/1/2005','(3,5)', '<(3,3),5>',  
, '132.248.75.140', 3.14159, '$2134.56', 'Texto, esto es un ejemplo de un insert  
sobre una tabla');
```

De esta manera se comprobó el funcionamiento de postgresql y se procedió a instalar y configurar las demás aplicaciones con las que trabajará en conjunto.

Instalando Apache

La instalación de apache en Linux, se puede realizar de dos formas:

– “Sencillo y Practico”

Por paquetes de la distribución favorita

apt-get install (Debian)

emerge (Gentoo)

rpm (Red Hat, Mandrake, SuSe)

installpkg (Slackware)

– “En cualquier Unix o Linux”

Se obtiene una copia del software de su sitio Web oficial <http://www.apache.org>

Se continúa con el proceso de compilación

- Se descomprime el archivo para obtener un directorio desde donde se ejecutará la instalación

```
$ tar zxvf httpd-2.0.54.tar.gz
```

- Una vez descomprimido el directorio se tendrá que posicionar dentro de el  
**\$ cd httpd-2.0.54**
- Se escriben los parámetros de configuración indicándole al sistema donde será instalada la aplicación  
**\$ ./configure --prefix=/usr/local/apache2**
- Se crean las dependencias y los archivos binarios de instalación  
**\$ make**
- Se ejecuta la instalación.  
**# make install**

Apache se encuentra instalado los parámetros de configuración básica de la aplicación fueron tratados en el módulo correspondiente, los parámetros que se necesiten cambiar para una correcta función con PostgreSQL serán tratados en este apartado. A continuación se muestra el proceso de instalación y configuración de PHP.

Instalando PHP.

La instalación de PHP también se puede realizar de dos formas:

- “Sencillo y Practico”  
Por paquetes de la distribución favorita.
- “En cualquier Unix o Linux”  
Se obtiene una copia del software de su sitio Web oficial <http://www.php.net>

Se continua con el proceso de compilación

- Se descomprime el archivo para obtener un directorio desde donde se ejecutará la instalación  
**\$ tar zxvf php-5.0.4.tar.gz**
- Una vez descomprimido el directorio se tendrá que posicionar dentro de el  
**\$ cd php-5.0.4**
- Se escriben los parámetros de configuración indicándole al sistema donde será instalada la aplicación, indicando que trabajará en conjunto con postgresql  
**\$ ./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-pgsql**
- Se crean las dependencias y los archivos binarios de instalación  
**\$ make**

- Se ejecuta la instalación.

**# make install**

- Se copia el archiva de configuración básica de PHP al directorio de librerías del sistema

**# cp php.ini-dist /usr/local/lib/php.ini**

**Editar httpd.conf y agregar las siguientes líneas:**

**LoadModule php5\_module modules/libphp5.so**

**AddType application/x-httpd-php .php .phtml**

**AddType application/x-httpd-php-source .phps**

Para validar que Apache y PHP han sido completamente instalados utilice:

1. Arrancar el Servidor de Web

**/usr/local/apache2/bin/apachectl start**

2. Validar que mod\_PHP ha quedado integrado a Apache

**Generar y probar el script info.php:**

**<?**

**phpinfo());**

**<?**

En capítulos anteriores se han mostrado las pantallas de verificación de funcionamiento de las aplicaciones aquí descritas. Hasta este punto se verifico que las aplicaciones funcionaran correctamente, se procedió a realizar la integración de Apache, PHP y PostgreSQL, como se muestra a continuación.

Para acceder a Postgres desde PHP

Para disponer de la funcionalidad requerida se debio compilar php habilitando el soporte para postgres, mediante la opción: `--with-pgsql`

El proceso de conexión a una base de datos Postgres desde PHP se realiza de forma similar a la conexión con bases de datos Mysql.

Se utilizan funciones del tipo "pg\_<action>" donde cada una de ellas define una acción en específico.

Para facilitar el proceso de conexión a la base de datos se tiene que parametrizar la base de datos se suelen utilizar las siguientes variables:

```
PGHOST=pgsql.ejemplo.com  
PGPORT=7890  
PGDATABASE=web-system  
PGUSER=web-user  
PGPASSWORD=secret  
PGDATESTYLE=ISO  
PGTZ=JST  
PGCLIENTENCODING=EUC-JP  
export PGHOST PGPORT PGDATABASE PGUSER PGPASSWORD  
PGDATESTYLE PGTZ PGCLIENTENCODING
```

Se crea una cadena (string) de conexión, algunas de las variables definidas conforman el string de conexión necesario para conectarse a Postgres.

```
$conn_string = "host=localhost port=5432 dbname=midb user=miguelmh  
password=linuxtux";
```

Con el string de conexión definido se puede proceder a establecer una conexión a la base de datos.

```
$con = pg_connect($conn_string);
```

A continuación se muestra un ejemplo de un string de conexión incrustado en una script de PHP.

```
<?php  
/*  
* Definimos los parámetros de conexión a la BD Postgres  
*/  
define('PGHOST','132.248.75.140');  
define('PGPORT',5432);  
define('PGDATABASE','ejemplo');  
define('PGUSER', 'miguelmh');  
define('PGPASSWORD', 'linuxtux');  
define('PGCLIENTENCODING','UNICODE');  
define('ERROR_ON_CONNECT_FAILED','Error: No fue posible conectarse al RDBMS');  
/*  
* Definimos entonces el string de conexión.  
*/  
pg_pconnect('host=' . PGHOST . ' port=' . PGPORT . ' dbname=' . PGDATABASE . '  
user=' . PGUSER . ' password=' . PGPASSWORD);  
?>
```

Para realizar accesos desde PHP hacía postgres, se disponen de las siguientes funciones de uso común:

```
$con = pg_connect($con);
```

Se conecta a la Base de datos

```
$result = pg_query ($con, "SELECT * FROM prueba order by id desc");
```

Realiza un Select sobre la bse de datos

```
$res = pg_insert($con, 'tabla', $arreglo);
```

Realiza el insert del contenido del arreglo \$arreglo en la tabla "tabla"

```
$arr = pg_fetch_array($result);
```

Recupera el contenido del arreglo \$result

```
pg_close($con);
```

Cierra la conexión a la base de datos

Para más información visitar la página Web de referencia  
<http://www.php.net/manual/en/ref.pgsqldb.php>

Se creo un usuario para manejo de una nueva base de datos distinto a postgres, se creo una base de datos y una tabla para ejemplificar el uso de PHP y postgresql.

Se cambia de usuario a postgres

```
$su – postgres
```

Se ejecuta el comando para creación del usuario

```
/bin postgres$ ./createuser -P miguelmh  
Shall the new user be allowed to create databases? (y/n) y  
Shall the new user be allowed to create more new users?(y/n) n  
CREATE USER  
exit
```

Se accede a la base de molde con el usuario que se ha creado

```
linux$ psql template1 miguelmh
```



Se crea la base de datos

```
create database primera ;
```

Se accede a la base recién creada para crear una nueva tabla

```
\c primera
```

```
primera=> create table prueba (id int, nombre varchar(100) ) ;
```

Se insertan datos

```
primera=> insert into prueba values (1, 'Miguel Angel Mendoza');  
primera=> insert into prueba values (2, 'Carmen Martinez Acevedo');  
primera=> insert into prueba values (3, 'Uriel Alfaro Mendoza');  
primera=> insert into prueba values (4, 'Gabriel Velázquez Mendoza');  
primera=> insert into prueba values (5, 'Gerardo Mendoza de la Rosa');
```

La tabla ha sido creada y los datos insertados.

El script que se programo para que se mostraran los datos en formato Web.

```
<?php  
$conn_string = "host=localhost port=5432 dbname=primera user=miguelmh  
password=mendoza75";  
$con = pg_connect($conn_string);  
$result = pg_query($con, "SELECT * FROM prueba order by id asc");  
?>  
<table border=1>  
  <tr>  
    <td>Id</td>  
    <td>Nombre</td>  
  </tr>  
<?>  
while ($row = pg_fetch_array($result))  
{  
  print "<tr>";  
  print "<td>$row[id]</td>";  
  print "<td>$row[nombre]</td>";  
  print "</tr>";  
}  
pg_close($con);  
?>
```

Como se comentó al inicio de este capítulo, durante el curso se trabajó con otras herramientas de software libre para el desarrollo de aplicaciones dinámicas, como es el caso de Smarty, que es un motor de plantillas para PHP. Específicamente, esta herramienta facilita la manera de separar la aplicación lógica y el contenido en la presentación, permitiendo a los programadores un mayor control sobre la información que requiera presentar. Smarty permite la creación de moldes o plantillas (templates) que facilitan el proceso de edición, eliminando el código PHP incrustado, sin eliminar la funcionalidad, estas plantillas son archivos HTML.

Smarty tiene muchas ventajas ya que un programador no tiene que preocuparse por la presentación, ya que está en un archivo distinto. Para el diseñador se minimiza el riesgo de al modificar alguna imagen se pierda la funcionalidad por una modificación involuntaria.

Esto implica que puede rediseñar el sitio, de forma mas sencilla sin modificar la funcionalidad.

Pero también existen algunas desventajas ya que para el programador la tarea de desarrollo es ligeramente más complicada. Se requiere de un preproceso para generar el HTML final, lo cual consume tiempo de procesamiento, CPU. Y para el diseñador debe considerar aun algunas etiquetas y trabajar con frecuencia con archivos fraccionados.

El funcionamiento de los motores generadores de plantillas para PHP (PHP Template Engine) es sencillo, se diseña sobre un HTML convencional, se insertan “tags” particulares que representan las secciones dinámicas de un sitio. El Engine busca por los tags y los reemplaza por código dinámico, definido previamente. Se genera “al vuelo” el código HTML que se mostrará al visitante. Algunos “Engines” incorporan cache o buffers con el fin de evitar regenerar una página cada vez que se solicita, la intención es optimizar el desempeño.

Smarty es uno de los Engines de templates para PHP mas ampliamente utilizado. Es posible descargarlo de <http://smarty.php.net/download.php>. Las características de Smarty son:

- Dispone de Cache
- Gran variedad de tags
- Ampliamente documentado en varios idiomas incluyendo el Español
- Cuenta con soporte para Plug-ins
- En realidad, es un pequeño pero potente lenguaje de “programación”

Se muestra a continuación el proceso de instalación, configuración y puesta en marcha de Smarty.

- Se obtiene una copia de la última versión del software de su sitio Web <http://smarty.php.net/download.php>

- Se descomprime el archivo para obtener un directorio desde donde se ejecutará la instalación

```
$> tar -zxvf Smarty-2.6.7.tar.gz
```

- Se recomienda crear un directorio para las librerías de Smarty

```
$> mkdir /usr/local/lib/php/Smarty
```

- Una vez que se descomprime el archivo se procede a copiar el contenido del directorio `libs` al directorio creado para las librerías de Smarty

```
$> cp -r Smarty-2.6.7/libs/* /usr/local/lib/php/Smarty
```

También se puede realizar el mismo proceso, solo que ahora en un directorio para librerías dentro del directorio `htdocs` de Apache.

- Se crea el directorio y se copia el contenido del directorio `libs`.

```
$>mkdir /usr/local/apache/htdocs/smarty/lib
```

```
$> cp -r Smarty-2.6.7/libs /usr/local/apache/htdocs/smarty/lib
```

- Se cambia al directorio `htdocs`

```
$> cd /usr/local/apache/htdocs
```

- Se crean los directorios necesarios para que Smarty funcione, el proceso de configuración de Smarty se encuentra publicado en el manual oficial de la aplicación.

```
$> mkdir smarty
```

```
$> mkdir smarty/templates
```

```
$> mkdir smarty/templates_c
```

```
$> mkdir smarty/cache
```

```
$> mkdir smarty/configs
```

- Se cambian de dueño los directories, por el usuario que corre el procesos de Apache

```
$> chown nobody:nobody smarty/templates_c  
$> chown nobody:nobody smarty/cache
```

- Se cambian permisos a los directorios

```
$> chmod 775 smarty/templates_c  
$> chmod 775 smarty/cache
```

La instalación de Smarty no tuvo problemas, se realizaron algunos ejercicios para ejemplificar el uso y funcionamiento de la herramienta a continuación se muestra el código estudiado en clase. Los archivos o plantillas de Smarty tienen la extensión .tpl y deben de ser guardados dentro del directorio templates. Los archivos PHP que hagan referencia a una plantilla deberán contener las rutas completas de los directorios y clases que ocupa Smarty.

Los directorios son para definir las propiedades de la clase de Smarty, `$template_dir`, `$compile_dir`, `$config_dir`, y `$cache_dir` respectivamente. Los directorios de Smarty solo son accesados por la librería de Smarty y nunca son accesados directamente por el navegador. Por consiguiente para evitar cualquier preocupación con la seguridad, es recomendado colocar estos directorios *fuera* del document root.

A continuación se muestra un ejemplo sencillo de lo comentado en el párrafo anterior.

```
<html>                                     <?php  
<head>                                     // La ruta completa donde se ubique Smarty.class.php  
<title>Smarty</title>                     require('/usr/local/lib/php/Smarty/Smarty.class.php');  
</head>                                     $smarty = new Smarty();  
<body>                                     $smarty->template_dir = '/usr/local/apache/htdocs/smarty/templates';  
Hola, {$nombre}!                           $smarty->compile_dir = '/usr/local/apache/htdocs/smarty/templates_c';  
</body>                                     $smarty->cache_dir = '/usr/local/apache/htdocs/smarty/cache';  
</html>                                     $smarty->config_dir = '/usr/local/apache/htdocs/smarty/configs';  
                                             $smarty->assign('nombre', 'Miguel');  
                                             $smarty->display('index.tpl');  
                                             ?>
```

index.tpl

index.php

Como se observa en el código, el archivo HTML o plantilla posee la extensión .tpl y la referencia al valor de la variable \$nombre se encuentra asignada con un valor determinado en el script php, donde se definen todos los directorios de configuración de Smarty.

También se utilizó otra herramienta muy poderosa para realizar interacciones con la base de datos de manera sencilla y directa; ADODB. ADOdb es una librería para PHP ADOdb es un conjunto de librerías de bases de datos que permite conectarse a varias bases de datos de una manera portable. Se puede obtener una copia de: <http://adodb.sourceforge.net/#download>.

ADOdb tiene Características especiales:

Facil para programadores Windows debido a que muchas de las convenciones son similares al ADO de Microsoft.

A diferencia de otras clases PHP de base de datos que se enfocan unicamente en el enunciado SELECT. ADOdb soporta código para manejar INSERT y UPDATE que son rápidamente adaptables a múltiples bases de datos.

Tambien hay métodos para manejo de fechas, concatenacion de cadenas y entrecomillado de cadenas para diferentes bases de datos.

Tiene un systema de metatipos (metatype) para poder determinar cuales tipos como CHAR, TEXT and STRING son equivalentes en diferentes bases de datos.

Es facil de portar debido a que todo el codigo que depende de la base de datos esta en funciones. No se tiene que portar la logica principal de las clases. La creacion de tablas e indices es portable con las clases de diccionario de datos datadict.

Posee un monitor de rendimiento de base de datos y ajuste de enunciados SQL con la clase de performance monitoring.

Manejo de sesiones en base de datos con la clase session management. Manejo de notificacion por sesion vencida.

Para instalar ADOdb

Hay que asegurarse de estar usando PHP 4.0.4 o posterior. Descomprimir todos los archivos en un directorio accesible por el servidor Web.

Asegurarse de corregir los parametros de conexión a las necesidades. Se puede depurar el código usando `$db->debug = true` como se muestra a continuación:

```
<?php
include('adodb/adodb.inc.php');
$db = ADONewConnection($dbdriver); # eg 'mysql' o 'postgres'
$db->debug = true;
$db->Connect($servidor, $usuario, $contraseña, $database);
$rs = $db->Execute('select * from alguna_tabla_pequeña');
print "<pre>";
print_r($rs->GetRows());
print "</pre>";
?>
```

### Instalación Mínima

Para los desarrolladores que quieran distribuir una versión mínima de ADOdb, se requieren los siguientes archivos:

**adodb.inc.php**  
**adodb-lib.inc.php**  
**adodb-time.inc.php**  
**drivers/adodb-\$database.inc.php**  
**license.txt (por razones legales)**  
**adodb-php4.inc.php**  
**adodb-iterator.inc.php**

Opcionalmente:

**adodb-error.inc.php y lang/adodb-\$lang.inc.php (si se usa MetaError())**  
**adodb-csvlib.inc.php (si se usan recordsets en memoria - CacheExecute(), etc)**  
**adodb-exceptions.inc.php y adodb-errorhandler.inc.php (si se usa el manejo de errores de ADOdb o excepciones de PHP5).**

### Ejemplos de Código de Inicio

Al ejecutar ADOdb, al menos se cargan dos archivos. Primero `adodb/adodb.inc.php`, que contienen todas las funciones usadas por todas las clases de bases de datos. El código específico a una base de datos en particular está en el archivo `adodb/driver/adodb-????.inc.php`.

Por ejemplo, para conectarse a una base de datos postgresql:

```
include('adodb/adodb.inc.php');  
$conn = &ADONewConnection('postgres');
```

Cada vez que se necesite conectar a una base de datos, se debe de crear un objeto de conexion usando la funcion `ADONewConnection($driver)`. `NewADODConnection($driver)` es un nombre alterno para la misma funcion.

En este momento, no se está conectado a la base de datos. Primero se tiene que decidir si se va a usar una conexión *persistente* o *no persistente*.

La ventaja de las conexiones *persistentes* es que son más rápidas, debido a que la conexion no es cerrada nunca (aun si se usa `Close()`). Las conexiones *No persistentes* consumen menos recursos reduciendo el riesgo de sobrecargar la base de datos o el servidor Web.

Para conexiones persistentes, se usa `$conn->PConnect()`, o `$conn->Connect()` para conexiones no persistentes. Algunas bases de datos tambien manejan `NConnect()`, la cual forza la creacion de una nueva conexion.

*Problemas con Conexiones:* Si se crean dos conexiones, pero ambas usan el mismo usuario y contraseña, PHP compartira las conexiones. Esto puede causar problemas si se supone que las conexiones son a diferentes bases de datos. La solucion es usar siempre diferentes usuarios para diferentes bases de datos o usar `NConnect()`.

PostgreSQL acepta conexiones usando:

a. La cadena estandard de conexion:

```
$conn = &ADONewConnection('postgres');  
$conn->PConnect('host=localhost port=5432 dbname=maria');
```

b. Los clasicos 4 parametros:

```
$conn->PConnect('localhost','usuario','contraseña','database');
```

Para ejemplificar el uso de ADOdb se colocará un ejemplo que muestra los campos de la tabla que se creó anteriormente:

```
<?
require("adodb/adodb.inc.php");

$db = NewADOConnection('postgres');
$db -> debug = false;
$db -> Connect("localhost", "miguel", "mendoza75", "primera");
$rs = $db -> Execute("Select * from prueba");

while ($arreglo = $rs->FetchRow() )
{
    print "<table border=1><tr>";
    print ("<td>id:</td><td>" . $arreglo['id'] . "</td>");
    print ("<td>Nombre:</td><td>" . $arreglo['nombre'] . "</td>");
    print "</tr></table>";
}

?>
```

Como puede observarse en el código la conexión a la base de datos es manejada por una variable de ADOdb. Como se comentó anteriormente, se tiene que crear una nueva instancia de conexión de ADOdb y se le indica al sistema que el nuevo objeto de conexión creado hace referencia a postgres, después es ejecutada la petición a la base de datos y finalmente se imprimen los resultados en una tabla de HTML por medio de un arreglo predefinido de ADOdb.

Se muestran a continuación ejemplos de inserción de datos desde un formulario y paginación de datos con ADOdb.

```
<html>                                     formu.html
<head>Formulario de prueba</head>
<body>
<form action="insertadatos.php" method="get">
Escribe un valor numerico<input type="text"
name="txtdato"><br>
Escribe un nombre <input type="text"
name="txtnombre"><br>
<input type="submit" value="Enviar datos">
<input type="reset" value="Cancelar datos">
</form>
</body>
```

El formulario posee dos cajas de texto que contienen la información que se almacenará en la base de datos. Dicha información es procesada, para su almacenamiento en la tabla usando ADOdb, mediante la script mostrada abajo. Dicha script contiene un referencia a un paginador que utiliza clases de ADOdb



```
<?
require("adodb/adodb.inc.php");
require("adodb/tohtml.inc.php");

$db = NewADOConnection('postgres');
$db -> debug = false;
$db -> Connect("localhost","miguel","mendoza75","primera");

if ($_GET=="")
{
}
else
{
$rs = $db -> Execute("Insert into prueba(id,nombre) values('" . $_GET["txtdato"] . "', '" . $_GET["txtnombre"] . "')");
$rs = $db -> Execute("Select * from prueba");
$rs2html($rs,'border=1 cellpadding=3',array('Identificador','Nombre'));

print "<BR><p>Sus datos han sido actualizados, por favor de clic en el link para regresar a la pagina de captura</p><A
HREF=javascript:history.go(-1)'>[Regresar]</A>";
print "<a href='paginador.php' target='_self'>Ir a Paginador</a>";
}

?>
```

**insertadatos.php**

```
<?
require("adodb/adodb.inc.php");
require("adodb/tohtml.inc.php");
require("adodb/adodb-pager.inc.php");

$db = NewADOConnection('postgres');
$db -> debug = false;
$db -> Connect("localhost","miguel","mendoza75","primera");

$rs = $db -> Execute("Select * from prueba");

$pager = new ADODB_Pager($db,"select * from prueba",'referenciade',true);
$pager->Render($rows_per_page=2);

$rs2html($rs,'border=1 cellpadding=3',array('Identificador','Nombre'));

?>
```

**paginador.php**

Como proyecto final del curso se trabajo en un mini portal de noticias utilizando todas las herramientas estudiadas en el curso, el proyecto consistio en la publicación de noticias por medio de moldes o templates, permitiendo a un administrador y a varios usuarios la modificación, eliminación o añadido de nuevas noticias y articulos.

Se notaron diferencias muy marcadas en el uso de las herramientas vistas durante este módulo en comparación con la programación tradicional que se venía manejando, y el manejo de las interfases de acceso a base de datos y presentación de la información de forma dinámica. La inclusión de este tipo de herramientas, facilitan el trabajo del administrador de sistemas, pues se maneja un estandar proporcionado por los moldes o templates, pues el uso de PHP con código embebido en el HTML puede llegar a hacer insostenible el mantenimiento de un sitio web cuando se quiere modificar el diseño, para ello existe una serie de clases para separar el código HTML del PHP. En otras palabras, el uso de templates permite separar elegantemente el codigo PHP de su presentacion, hay varios motores de templates, en este módulo se utilizó Smarty, sin dudas la opcion mas poderosa, y cuyas características ya fueron tratadas anteriormente.

A continuación se mostrará la pantalla principal del sistema de portal de noticias.

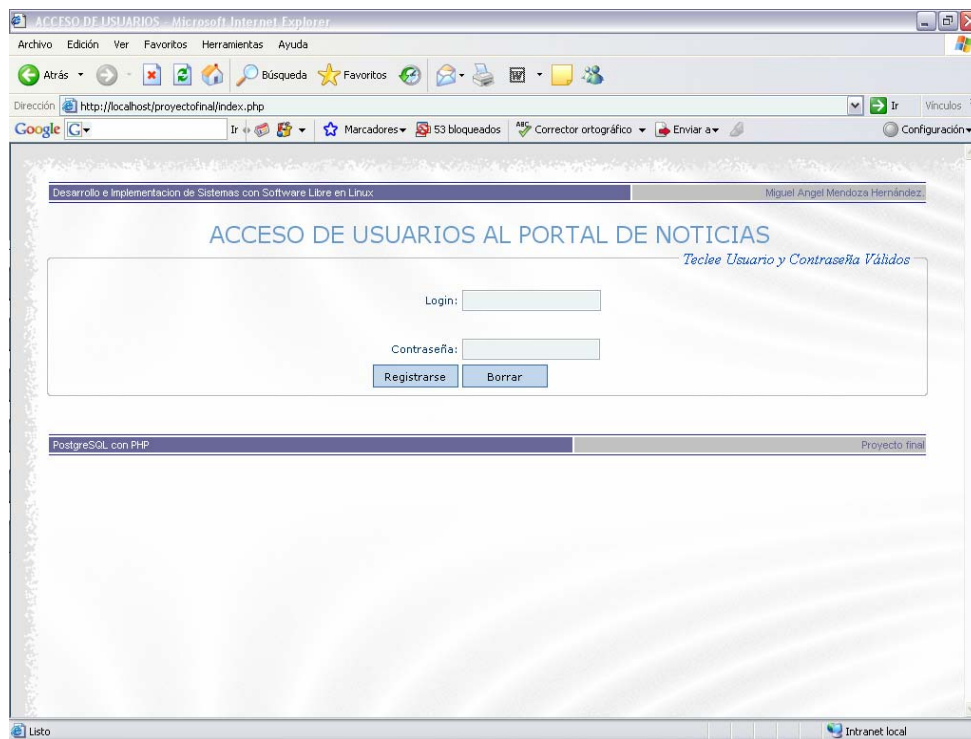


Figura 20 Entrada al sistema del módulo VIII.

## 2. Desarrollo de un sistema de control de inventarios con software libre.

La empresa Mexilac Planta Tizayuca, cuenta con una infraestructura en el área de sistemas de última generación, sin embargo, el desarrollo de aplicaciones está enfocado a cuestiones asociadas con las áreas de ventas, costos, contabilidad y asuntos humanos, y están desarrollados para correr sobre clientes con Windows 2000 como sistema operativo residente, las aplicaciones corren sobre la Intranet del corporativo y todos los servicios y servidores se encuentran ubicados en Torreón, que es la sede corporativa.

En el grupo no se maneja algún sistema para el manejo de inventarios, para control y gestión de los equipos de cómputo, usuarios asignados por zona, nomenclatura de los equipos por zona, equipos asignados por área (sus características de software y hardware), por tal razón se propone el siguiente proyecto para que pueda ser usado por los responsables del área de sistemas de cada planta, cada centro de distribución (CEDIS), y cada sede que forme parte del grupo Lala.

En el corporativo se cuenta con software propietario, todos y cada uno de los equipos utilizados en todo el corporativo tiene como sistema operativo de base alguna de sus versiones de Microsoft Windows, en su gran mayoría Windows 2000, los sistemas y aplicaciones corren sobre servidores UNIX o bien sobre Windows Server, el corporativo no cuenta con servidores dedicados que tengan como plataforma alguna distribución de Linux, salvo un equipo que se encuentra funcionando como proxy y firewall, el cual provee la salida a Internet y filtra contenidos tanto salientes como entrantes.

El sistema propuesto está basado en software libre utilizando las herramientas estudiadas durante el diplomado de “Desarrollo de Sistemas con Software Libre”, cuyo informe se detalla en el capítulo 1.

### 2.1 Características de hardware y software utilizado en este proyecto.

Se utilizó un equipo Pentium IV marca Dell con las siguientes características:



Procesador Pentium IV 2.8 GHz



Memoria RAM 256 MB






Disco duro 40 GB











Tarjeta de video integrada chip Intel



Tarjeta de red integrada chip Intel PRO

-  Monitor pantalla plana 17" digital
-  Teclado PS2 102 teclas
-  Mouse optico USB Scroll

## 2.2 La plataforma de software utilizado en este proyecto fue la siguiente:

-  Linux Slackware versión 10.1 como sistema operativo.
-  Apache 2.0 como servidor de páginas Web.
-  MySQL como servidor de bases de datos.
-  MySQLCC como centro de control del gestor de la base de datos
-  PHP como lenguaje de programación de lado del servidor.
-  Quanta para el manejo y edición de los scripts.
-  Webmin para la administración vía Web del servidor.
-  PICO y VI para la edición de archivos de configuración y scripts.

Una vez instalado, configurado y probado el sistema operativo y todo el software mencionado anteriormente se procedió a realizar un analisis de los requerimientos de información para el sistema, se tomó en cuenta que es un sistema de control de inventarios por lo que los datos estan enfocados de la siguiente manera:

La información contenida en la base de datos está basada en un estandar que se maneja en todas las áreas de sistemas del corporativo. La información contenida en la base de datos se basa en los siguientes campos: **inventario** este campo fue diseñado creando una nomenclatura generalizada basada en 3 subcampos de contenido como se muestra a continuación:

Equipo	Departamento	Código de referencia	Nomenclatura
CPU	Sistemas	01	CPUDS01
MONITOR	Sistemas	02	MONDS02
MOUSE	Sistemas	03	MOUDS03
TECLADO	Sistemas	04	TECDS04
IMPRESORA LASER	Sistemas	05	IMLDS05
IMPRESORA MATRIZ	Sistemas	06	IMMDS06
NOBREAK	Sistemas	07	NOBDS07
REGULADOR	Sistemas	08	REGDS08

Observe que para el campo nomenclatura se están tomando las tres primeras letras del tipo de equipo, más la letra D(departamento), más la(s) inicial(es) del área, más el código asignado al tipo de dispositivo, ej.: **CPUDS01**  
En caso de que haya más de un equipo en el área asignado, se le agregará a la nomenclatura la letra E (equipo) más un número indicador del equipo (1,2,3), ej. **CPUDS01E1** ó bien **CPUDS01E2**  
Tomando como ejemplo el registro para un cpu y un monitor del área de Recursos humanos quedaria de la siguiente manera:  
para el cpu **CPUDRH01** y para el monitor **MONDRH02**

**serie** » este campo basa su información en el service tag de los equipos o bien el número de serie que trae cada dispositivo, es un estándar utilizado en el corporativo.

**asignado a** » campo utilizado para definir a quien ha sido asignado el equipo por departamento, este valor es susceptible a cambios por los departamentos existentes o no en las plantas o cedis.

**dirección ip** » campo utilizado para colocar la dirección IP asignada por el DHCP, este campo es opcional pero se incorpora a la base de datos debido a que la mayoría de los routers en el corporativo "recuerdan" la dirección física de los equipos y de esta manera las IP's asignadas se mantienen aún renovandolas, se maneja el último octeto para captura, ej.: del segmento 200.200.151.0-255 (correspondiente a Tizayuca) se extrae el último octeto, así para una dirección de este segmento válida como lo es la 200.200.151.59 el valor que se capturaría en la base de datos sería 059, debido a que hay direcciones que tienen 3 dígitos como es el caso de 200.200.151.100 donde 100 sería el dato a capturar en la base de datos.

**nomenclatura** » campo utilizado para definir la nomenclatura del equipo ej. para Tizayuca se tiene MLTIZPTA051 (ML acrónimo de MultiLacteos, TIZ acrónimo de Tizayuca, PTA acrónimo de planta, 051 número de equipo asignado), para Campofrío se tiene MLCAFOFC098 (ML acrónimo de MultiLacteos, CAF acrónimo de Campo Frío, OFC acrónimo de oficinas, 098 número de equipo asignado).

**usuario** » campo utilizado para definir a que usuario ha sido asignado el equipo  
**lugar de ubicación** » campo utilizado para definir la zona (planta o cedis) donde esta ubicado el equipo.



Existen otros campos que servirán para definir el tipo de equipo que se tiene o se pretende dar de alta, la marca de tales equipos o simplemente para modificar registros existentes, ejemplos de estos registros se muestran a continuación:

Serie	Usuario	Nomenclatura	Dirección IP	Tipo de equipo	Marca	Departamento
CPUDPRA01	RUTH	MLTIZPTA019	200.200.151.122	CPU PIII-10 GB DD-64 RAM	Compaq Deskpro PIII	CONTABILIDAD
CPUDCD01E3	Mariana Damaris Paulin	MLTIZPTA006	200.200.151.97	CPU PIV-20 GB DD- 256 RAM	DELL GX260	CONTABILIDAD
CPUDCD01E2	Maria de Jesus Gomez Reyes	MLTIZPTA002	200.200.151.110	CPU PIV-20 GB DD- 256 RAM	DELL GX260	CONTABILIDAD
CPUDCD01E1	Ruth Massiel Gonzalez	MLTIZPTA008	200.200.151.58	CPU PIV-20 GB DD- 256 RAM	DELL GX260	CONTABILIDAD
MONDPRA02	RUTH	MLTIZPTA019	200.200.151.122	MONITOR 15" S500	COMPAQ S500	CONTABILIDAD
MONDCD02E1	Ruth Massiel	MLTIZPTA008	200.200.151.5	MONITOR 17"	DELL 772P	CONTABILIDAD

Informe del Proyecto de un Sistema de Control de Inventarios  
con Software Libre

	Gonzalez		8	E772P		
MONDCD02E3	Mariana Damaris Paulin	MLTIZPTA006	200.200.151.97	MONITOR 17" E772P	DELL 772P	CONTABILIDAD
MONDCD02E2	Maria de Jesus Gomez Reyes	MLTIZPTA002	200.200.151.10	MONITOR 17" E772P	DELL 772P	CONTABILIDAD
MOUDCD03E1	Ruth Massiel Gonzalez	MLTIZPTA008	200.200.151.58	MOUSE OPTICO USB DELL	DELL USB	CONTABILIDAD
MOUDCD03E3	Mariana Damaris Paulin	MLTIZPTA006	200.200.151.097	MOUSE OPTICO USB DELL	DELL USB	CONTABILIDAD
MOUDCD03E2	Maria de Jesus Gomez Reyes	MLTIZPTA002	200.200.151.10	MOUSE PS2 DELL	DELL PS2	CONTABILIDAD
MOUDPRA03	RUTH	MLTIZPTA019	200.200.151.122	MOUSE PS2 GENIUS	GENIUS PS2	CONTABILIDAD
TECDPRA04	RUTH	MLTIZPTA019	200.200.151.122	TECLADO 102 TECLAS PS2	COMPAQ PS2	CONTABILIDAD
TECDCD04E1	Ruth Massiel Gonzalez	MLTIZPTA008	200.200.151.58	TECLADO 102 TECLAS PS2	DELL PS2	CONTABILIDAD
TECDCD04E3	Mariana Damaris Paulin	MLTIZPTA006	200.200.151.97	TECLADO 102 TECLAS PS2	DELL PS2	CONTABILIDAD
TECDCD04E2	Maria de Jesus Gomez Reyes	MLTIZPTA002	200.200.151.10	TECLADO 102 TECLAS PS2	DELL PS2	CONTABILIDAD

El sistema cuenta con un módulo adicional para la impresión de etiquetas de control que contienen la información básica de cada equipo y el cual es pegado en cada dispositivo de tal manera quede a la vista del personal y de los usuarios para manejo de los datos. A continuación se presenta el modelo de etiqueta lista para impresión:

Inv:CPUDS01E1 serie:8RNP41		Inv:MONDS02E1 serie:CN0Y-432BJHP	
Area:SISTEMAS Ub:TIZAYUCA		Area:SISTEMAS Ub:TIZAYUCA	
Tipo:P IV-40 GB DD-256 RAM		Tipo:MONITOR 17" UVGA	
IP:059 Marca:DELL GX270		IP:059 Marca:DELL E773S	
Nom:MLTIZPTA051		Nom:MLTIZPTA051	

Dichas etiquetas son impresas en el modelo de etiquetas Avery 5162. Más adelante se tratará este tema a fondo.

Con los datos establecidos se procedió al diseño de la base de datos. El diseño de la base de datos así como la programación de los módulos de altas bajas y cambios se crearon de acuerdo a las necesidades en el área.

La base de datos consta de las siguientes tablas:

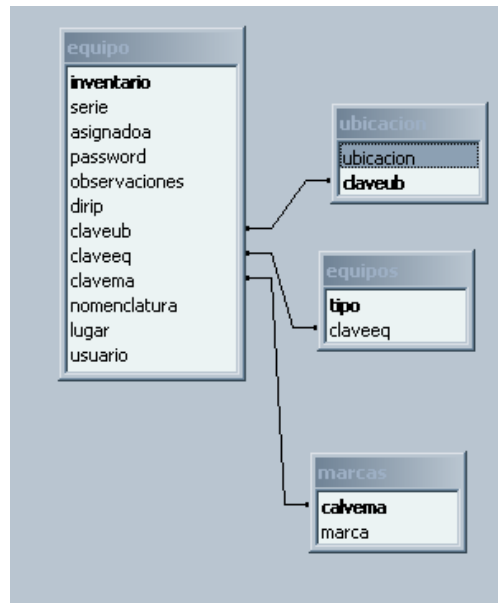


Figura 1 Tablas de datos de la base Principal

### Tabla equipo

Nombre del campo	Tipo de Datos	Descripción
inventario	Texto (12) PK	Numero de inventario asignado a cada equipo.
Serie	Texto (25)	Numero de serie de línea de cada equipo
asignadoa	Texto (25)	Nombre del departamento a quien se tiene asignado el equipo.
password	Texto (15)	Contraseña para protección de datos
observaciones	Texto (40)	Campo de observaciones a la información contenida
Dirip	Texto (3)	Último número de las direcciones IP por segmento 200.200.151.n
claveub	Numerico (doble)	Numero definido para clave de ubicación del equipo.
claveeq	Numerico (doble)	Numero definido para clave del tipo de equipo.
clavevema	Numerico (doble)	Numero definido para clave de la marca del equipo.
nomenclatura	Texto (11)	Nombre asignado a cada equipo para identificarlo en red.
lugar	Texto (50)	Lugar fisico donde se encuentra el equipo.
usuario	Texto (50)	Nombre del usuario vigente a quien esta asignado el equipo

### Tabla equipos

Nombre del campo	Tipo de datos	Descripción
Tipo	Texto (30)	Descripción de las características de los equipos
claveeq	Autonumerico (entero largo)	Clave asignada conforme se van insertando los datos en los campos para definir el tipo de equipo

### Tabla marcas

Nombre del campo	Tipo de Datos	Descripción
marca	Texto (50)	Descripción de las marcas de los equipos
clavema	Auto numérico (entero largo)	Clave asignada conforme se van insertando nuevos datos, en los campos para definir la marca del equipo.

### Tabla ubicación.

Nombre del campo	Tipo de datos	Descripción
ubicación	Texto (30)	Descripción de la ubicación por departamento donde se encuentran los equipos.
claveub	Auto numérico (entero largo)	Clave asignada conforme se van insertando nuevos datos en los campos para definir la ubicación del equipo

Las tablas fueron creadas por medio de archivos sql en modo comando, y contenían la siguiente información:

#### Archivo equipo.sql

```
create table equipo (inventario text(12) primary key, serie text(25), asignadoa text(25), password text(15), observaciones text(40), dirip text(3), claveub numeric double, claveeq numeric double, clavema numeric double, nomenclatura text(11), lugar text(50), usuario text(50));  
commit work;
```

#### Archivo equipos.sql

```
create table equipos (tipo text(30), claveeq auto numeric);  
commit work;
```

#### Archivo marcas.sql

```
create table marca (marca text(50), clavema auto numeric);  
commit work;
```

#### Archivo ubicacion.sql

```
create table ubicacion (ubicacion text(30), claveub auto numeric);  
commit work;
```



De esta manera fueron creadas las tablas y se procedió a programar los módulos de altas, bajas, cambios y reportes utilizando PHP como lenguaje de programación, la lista de scripts se muestra a continuación.

presenta.php. La página de inicio del sistema. Contiene un menú dinámico que permite la navegación en el sistema y muestra una breve introducción sobre lo que trata. A continuación se muestran algunas pantallas de la página principal.

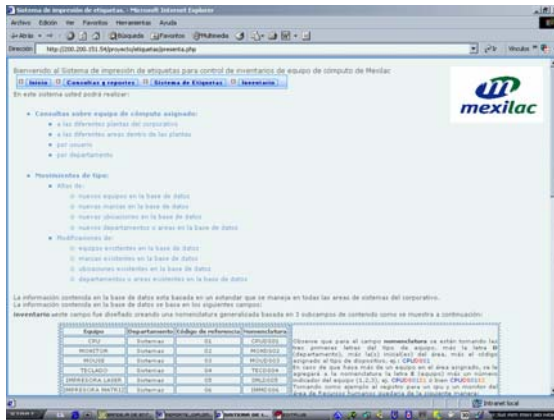


imagen 1a



imagen 2a

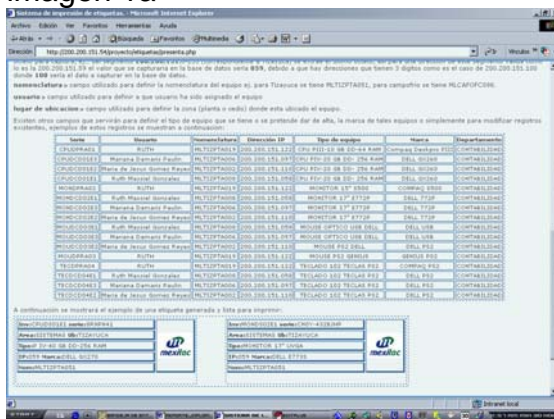


imagen3a

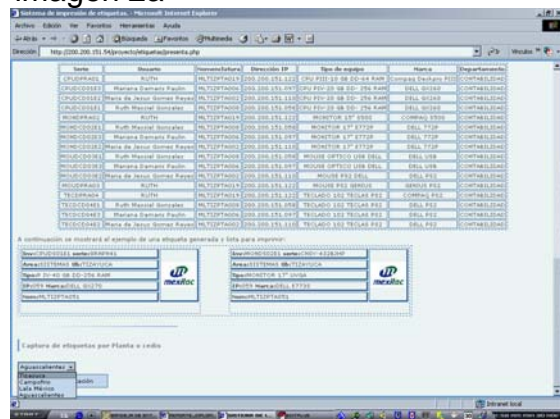


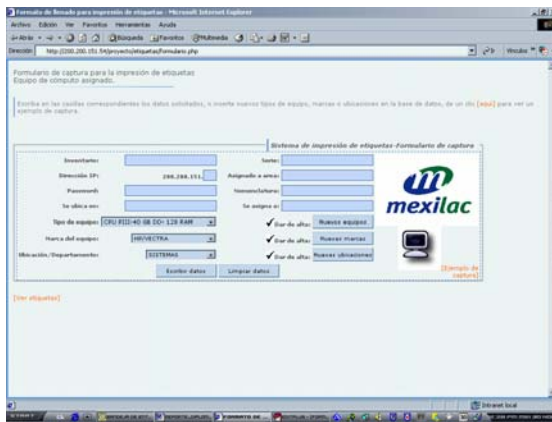
imagen4a

Las anteriores pantallas muestran la entrada principal al sistema, muestran contenido dinámico con interacción directa a la base de datos MySQL con PHP, así como una muestra de impresión de etiquetas de control y menús para selección de captura por zona.

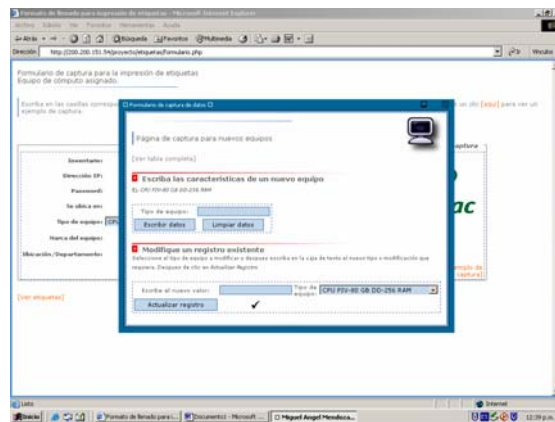
En la imagen 2a se encuentra un menú dinámico que brinda acceso a la captura de información por zonas, en el caso de estudio se selecciono la entrada de datos para la zona de Tizayuca, al dar clic en el objeto del menú seleccionado el sistema redirecciona a un formulario escrito en php llamado formulario.php que permite la captura de nuevos datos para insertarse en la base de datos, esta página contiene vinculos hacia otras paginas para insertar nuevos equipos o bien

realizar cambios en esa categoría , insertar nuevas marcas o realizar cambios en esa categoria y por ultimo, insertar nuevas ubicaciones o realizar cambios en esa categoria, asi como un enlace para ver un ejemplo de captura del sistema.

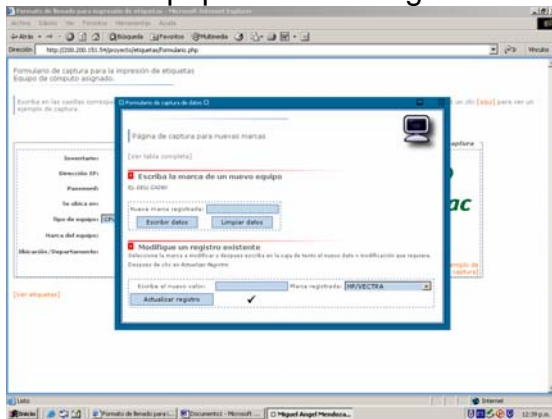
Cada vínculo está accionado por un botón de formulario que abrirá una ventana emergente dinámica programada en JavaScript y que contiene hojas de estilo en cascada (CSS – Cascade Style Sheets). Las páginas a las que hacen referencia dichos vínculos son, escribe Equipos.php que es una script programada para realizar inserciones o cambios en la base de datos en la tabla de equipos, escribe marcas.php, que realiza inserciones o cambios en la base de datos en la tabla de marcas, y escribe ubicaciones.php, que realiza inserciones o cambios en la base de datos en la tabla ubicación.



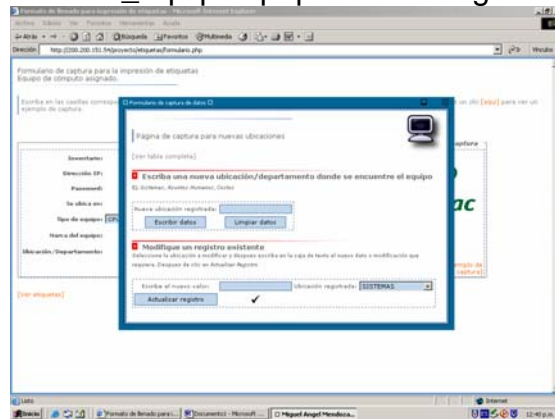
formulario.php fig 1.



escribe Equipos.php fig 2



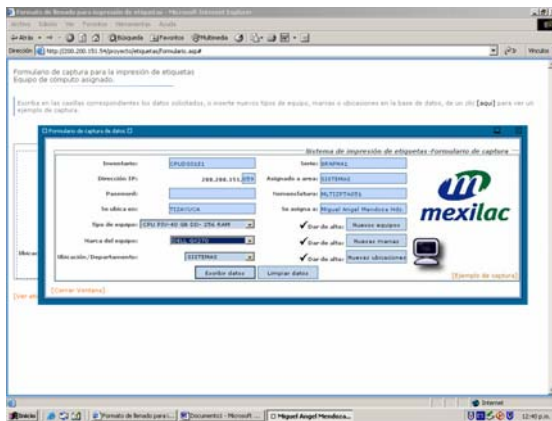
escribe marcas.php fig 3.



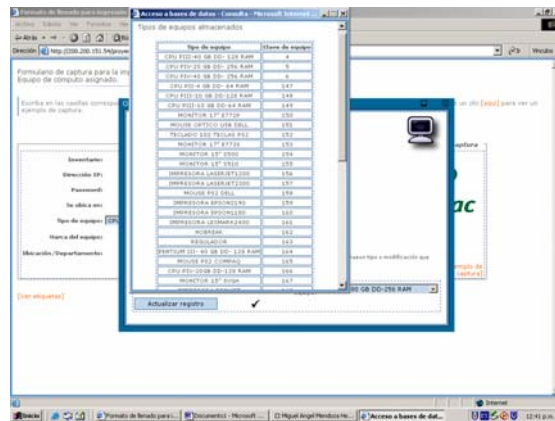
escribe ubicaciones.php fig 4.

Se muestra a continuación las pantallas de muestra de inserción de datos y la pantalla que muestra los equipos capturados en la base de datos cuyo vinculo se encuentra en las ventanas emergentes.

Desarrollo de un sistema de control de inventarios con software libre



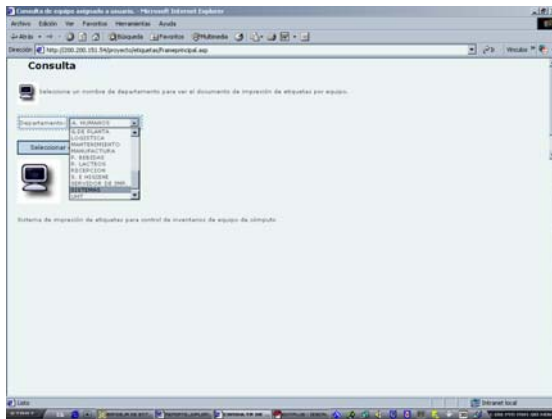
formulario de muestra para captura fig 5.



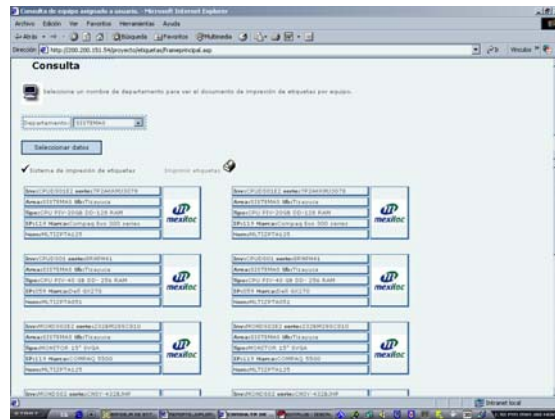
ventana emergente con información fig 6. contenida en la base de datos por tabla.

Una vez que se capturan, actualizan o eliminan los datos necesarios, el sistema cuenta con un módulo de reportes para que el encargado del área de sistemas por zona pueda generar reportes con información actualizada del equipo de cómputo con el que cuenta en su zona, así como filtros para presentación de informes detallados por departamento, usuario, o nomenclatura de equipos, así como para acceder al módulo de impresión de etiquetas de control, dicho vinculo se encuentra ubicado en la fig. 1 y esta marcado con la leyenda ver etiquetas, y es el punto en el cual nos centraremos.

Al dar clic en el vinculo [Ver etiquetas](#) se abrirá una nueva ventana de tipo frame llamada frameprincipal.php que contiene un filtro por departamento para tener acceso a las etiquetas de control listas para imprimirse como se muestra en las pantallas siguientes.

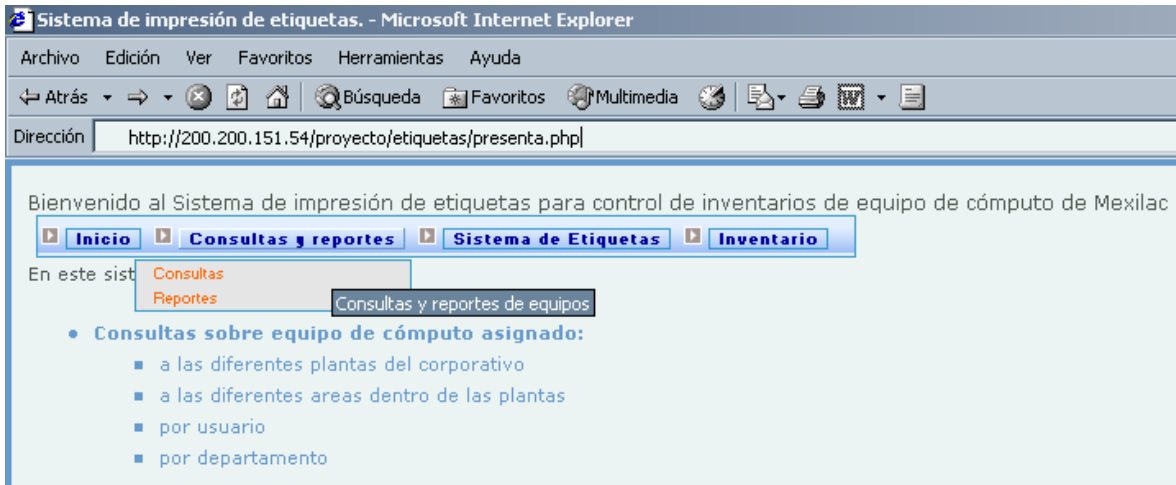


frameprincipal.php fig 7.

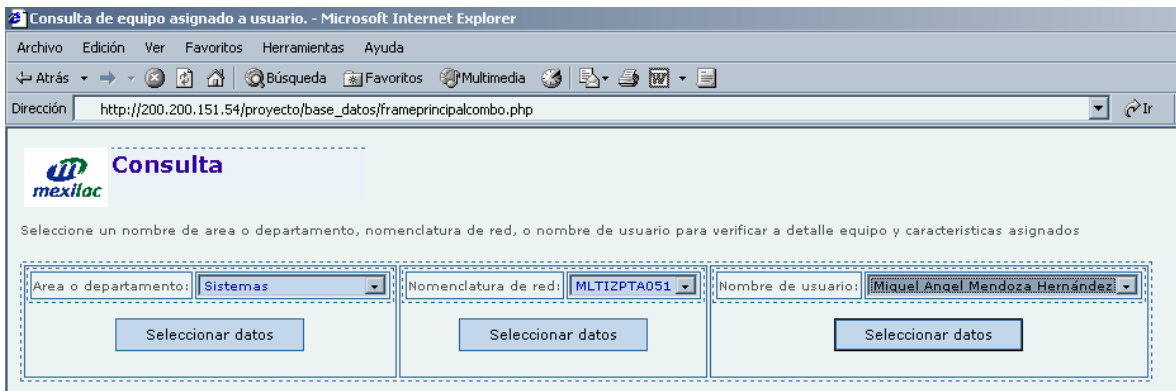


selección de impresión de etiquetas para el departamento de sistemas fig 8.

Para realizar informes o reportes del sistema obtenidos de la información contenida en la base de datos, existe un vínculo en la página principal dentro del menú dinámico consultas y reportes



Al seleccionar alguna de las dos opciones el sistema redirecciona a una página que contiene filtros para ver información de la base de datos, ya sea que filtre la información por área o departamento, por nomenclatura de red del equipo o por usuario, como se muestra a continuación.



El filtrado de información se muestra en un frame dentro de la misma página como se muestra a continuación:

Consulta específica por área o departamento

Tabla de datos de equipo asignado a usuario					
Ubicación	Área	Nombre de usuario	Usuario de red	Sistema Operativo instalado	Versión de Service Pack
Planta Tizayuca	Sistemas	resguardo sistemas	Administrador	Microsoft Windows 2000 Profesional	Service Pack 4
Planta Tizayuca	Sistemas	Miguel Ángel Mendoza Hernández	mamendez	Microsoft Windows 2000 Profesional	Service Pack 4

Ver el tipo de equipo asignado a estos usuarios  
 Ver el software instalado

Regresar a la pantalla de **Tabla de usuarios**

Filtro de información por área o departamento, observese que se generan vinculos que funcionan de manera dinamica para generar información acerca del tipo de equipo que tiene asignado el usuario o bien para saber el software que tiene instalado. Como se muestra a continuación:

Consulta específica por área o departamento

Tabla de datos de equipo asignado a usuario						
Ubicación	Área	Nombre de usuario	Usuario de red	Sistema Operativo instalado	Versión de Service Pack	
Planta Tizayuca	Sistemas	resguardo sistemas	Administrador	Microsoft Windows 2000 Profesional	Service Pack 4	
Planta Tizayuca						
		<b>Tipo de Procesador</b>	<b>Memoria RAM instalada</b>	<b>Modelo de computadora</b>	<b>Usuario Asignado</b>	<b>Nomenclatura</b>
		Intel Pentium 4A, 2800 MHz (21 x 133)	253 MB (PC2100 DDR SDRAM)	Dell OptiPlex GX270	Resguardo sistemas	MLTIZPTA054

Ver el tipo de equipo asignado  
 Ver el software instalado

Regresar a la pantalla de **Tabla de usuarios**

En todos estos informes que se generan existe la posibilidad de poder imprimir la información obtenida de la base de datos para un mejor control de la misma, o bien tener un control de inventario coordinado entre plantas y cedís.

La impresión de la etiquetas es responsabilidad de cada persona encargada del área de sistemas por localidad, por lo que cada cual deberá tener actualizada la base de datos y deberá colocar en cada equipo o dispositivo dichas etiquetas para identificación y manejo de los mismos.

La implementación de este sistema garantizó el control del inventario de equipo de cómputo por planta y por CEDIS perteneciente al grupo LALA, que como se había comentado se controlaba mediante un archivo creado en Excel y no garantizaba un acceso confiable a la información y era difícil de manipular.

## Conclusiones

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

Desarrollos tan robustos que se comparan con aplicaciones de carácter propietario y que brindan un mejor rendimiento y una mejor respuesta al usuario final ya sea que su aplicación este ejecutándose vía WEB o localmente. La flexibilidad de los lenguajes de programación empleados, embebidos al lenguaje universal con el que están creadas las páginas WEB, y permitiendo la adaptación a estándares de cada empresa o lugar donde se planea implementar sistemas utilizando esta tecnología.

El desarrollo de estos sistemas puede ser sustentado empleando software libre como LINUX, PHP, Apache-WWW-server y MySQL o PostgreSQL, que son herramientas, como se ha demostrado en este informe, tienen un alto desempeño, gran estabilidad y seguridad y, por el hecho de ser libres, permiten reducir los costos que se generan por las licencias de uso de software, logrando aprovechar al máximo los equipos de cómputo con que se cuenta.

La instalación de Linux como plataforma principal garantiza al usuario final un sistema operativo confiable, brindando una menor posibilidad de errores y riesgo por vulnerabilidades de software, haciendo su administración sencilla y eficaz a través de aplicaciones enfocadas a tareas específicas.

Linux cuenta con programas para casa y oficina hasta aplicaciones complejas para desarrollo de Software y Hardware, manejadores de bases de datos, integración de servicios. Todo en un solo sistema.

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

La implementación de sistemas usando software libre en las empresas públicas y privadas esta creciendo cada vez más, y es común encontrar en las empresas equipos con el sistema operativo Linux, haciendo funciones varias, como puede ser un servidor de bases de datos, de archivos, de aplicaciones, de páginas WEB, de correo, de servicios de Terminal, proxy o como equipo ruteador.

## **Bibliografía**

Título: Manual Avanzado de Linux  
Autor: Raúl Montero Rivero  
Editorial: Anaya Multimedia

Título: La Biblia de Administración de sistemas Linux  
Autor: Dee-Ann Leblanc  
Editorial: Anaya Multimedia

Título: Administración de Servicios de Información en Internet  
Autor: Cricket Liu y otros  
Editorial: Osborne/McGraw Hill

Título: Administración de Sistemas Linux  
Autor: M. Carling, S. Degler, J. Dennis  
Editorial: Prentice Hall/Pearson

Título: MySQL Edición Especial  
Autor: Paul Dubois  
Editorial: Prentice Hall/Pearson

Título: MySQL in a nutshell  
Autor: Russell Dyer  
Editorial: O'REILLY & ASSOCIATES

Título: WEB Database applications with PHP and MySQL  
Autor: Lane, David; Williams, Hugh E.  
Editorial: O'Reilly Vlg. GmbH & Co.

Título: PostgreSQL  
Autor: Korry Douglas, Susan Douglas.  
Editorial: Sams.

Título: Aggressive Network Self-Defense (Paperback)  
Autor: Neil R. Wyler, Bruce Potter, Chris Hurley  
Editorial: Syngress Publishing

## Mesografía

<http://www.slackware.org>  
<http://www.apache.org>  
<http://www.php.net>  
<http://www.mysql.org>  
<http://www.postgresql.org>  
<http://smarty.php.net/>  
<http://adodb.sourceforge.net/>  
<http://www.cert.org>  
<http://www.snort.org>  
<http://www.insecure.org>  
<http://www.linuxparatodos.net>  
<http://www.google.com>  
<http://es.wikipedia.org>  
<http://www.seguridad.unam.mx>  
[http://www.isocmex.org.mx/seg\\_info.html](http://www.isocmex.org.mx/seg_info.html)