



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

**“PROYECTO DE UN SISTEMA DE CONSULTA DE
INFORMACIÓN DE CURSOS, EN LÍNEA BASADO
EN SOFTWARE LIBRE Y TECNOLOGÍA
ORACLE”**

T R A B A J O E S C R I T O

**EN LA MODALIDAD DE SEMINARIOS Y
CURSOS DE CAPACITACIÓN Y
ACTUALIZACIÓN PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**

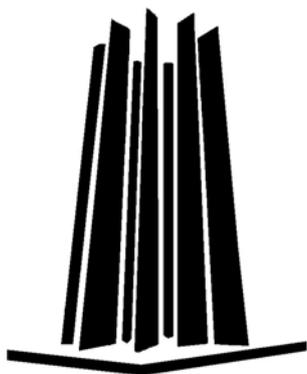
P R E S E N T A :

LIZBETH LEIJA HERNÁNDEZ

ASESOR:

ING. JOSÉ AGUSTÍN VARGAS PADILLA

SAN JUAN DE ARAGÓN, ESTADO DE MÉXICO SEPTIEMBRE DEL 2006





Universidad Nacional
Autónoma de México



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.

*A mi familia,
a Irving Cosme Hernández
y amigos.*

Agradecimientos.

Por todo su apoyo incondicional y por estar siempre junto a mí, gracias a Irving Cosme Hernández.

Agradezco a toda mi familia por su comprensión, por todos sus consejos y por estar a mi lado apoyándome a ser de mi cada día una mejor persona.

Agradezco a la Maestra Alma Ibarra Obando por la oportunidad que me dio de cursar el diplomado con el cual fue posible realizar este trabajo, por todo su apoyo y todos los consejos que me brindo para iniciarme en el desarrollo profesional.

Agradezco a todos los profesores de la Universidad Nacional Autónoma de México a los profesores de la carrera Ingeniería en Computación de la Facultad de Estudios Profesionales Aragón por su contribución en mi formación profesional.

Agradezco a mi asesor de tesis Ingeniero José Agustín Vargas Padilla.

ÍNDICE

INTRODUCCIÓN

JUSTIFICACIÓN

DEFINICIÓN DE LA PROBLEMÁTICA

ALCANCES Y LIMITACIONES

OBJETIVOS

CAPÍTULO I. INFORME GENERAL DEL DIPLOMADO: “DESARROLLO E IMPLEMENTACIONES DE SISTEMAS CON SOFTWARE LIBRE EN LINUX”.

Conceptos básicos

Iniciación a Linux

HTML

PHP

MySQL

SERVIDOR APACHE

CAPÍTULO II. PROYECTO DE UN SISTEMA DE CONSULTA DE INFORMACIÓN DE CURSOS, EN LÍNEA BASADO EN SOFTWARE LIBRE Y TECNOLOGÍA ORACLE

Descripción

Diseño del Sistema

Lenguaje de programación

Sistema Manejador de Bases de Datos

Sistema Operativo

Servidor web

CONCLUSIONES

BIBLIOGRAFÍA

INTRODUCCIÓN

El presente trabajo presenta los conceptos más importantes que fueron vistos en el diplomado “**Desarrollo e implementación de sistemas con software Libre en Linux**”, así como ejemplos y tareas más importantes de un administrador.

En cada módulo del diplomado fueron vistos los conceptos y definiciones más comunes para la implementación de la administración en Linux, es por eso que se incluye esta información, además de la administración del sistema operativo también se aplica la programación y el manejo de bases de datos, los cuales son utilizados para el desarrollo de este proyecto.

En el primer capítulo como ya hice referencia se mencionan los temas primordiales que debe conocer el administrador de Linux, ya que son conceptos, definiciones, comandos y tareas de administración básica e importante para el uso correcto del sistema operativo.

El segundo capítulo describe el desarrollo del **PROYECTO DE UN SISTEMA DE CONSULTA DE INFORMACION DE CURSOS, EN LINEA BASADO EN SOFTWARE LIBRE Y TECNOLOGÍA ORACLE** el cual es un sistema que agiliza la tarea de búsqueda de información de personas que son registradas a través de la misma aplicación, para fines propios de la institución en este caso la Dirección General de Sistemas de Cómputo Académico de la UNAM. Este sistema hace uso de herramientas de programación y conceptos de bases de datos y se compone de diferentes módulos, los cuales son descritos cada uno, en este capítulo, además de la estructura general del proyecto.

JUSTIFICACIÓN:

La creación del proyecto de un sistema de consulta de información de cursos, en línea basado en software libre y tecnología oracle, permite poner en práctica la serie de conocimientos teóricos adquiridos en las aulas, también se tiene la oportunidad de elegir el tema a desarrollar, plasmando en éste lo aprendido en los estudios profesionales y dar a conocer la forma como se aplicaron los conocimientos teóricos a un problema práctico concreto.

Además de lo anterior la importancia del desarrollo de un sistema que ponga al alcance de cualquier usuario información de los diferentes cursos impartidos por instituciones que ofrecen cursos de cómputo, facilitando al usuario realizar su registro a los cursos sin tener que salir de casa.

El usuario podrá ingresar a la página y consultar los Cursos que hay disponibles en todas las sedes y si le interesa alguno de los cursos, se podrá registrar al curso en ese momento, tendrá la opción de elegir varios cursos.

De forma personal tiene mucho significado ya que, además de poder realizar la creación del sistema dentro de un cierto límite de tiempo, me ayuda a desarrollarme profesionalmente y me sirve como ayuda para la creación de nuevos sistemas de este tipo para así poder dedicarme al trabajo profesional.

DEFINICIÓN DE LA PROBLEMÁTICA

¿Por qué de mi investigación?

Actualmente las pequeñas empresas se dedican a la construcción de sitios Web dinámicos, realizan un diseño de páginas con las cuales presentan sus servicios y sus productos. Este es utilizado como interfaz para la creación y presentación de toda la empresa, generando un sitio Web por cada categoría y subcategoría de los servicios de la misma.

El diseño de este sistema se puede lograr con métodos vistos durante el periodo académico y con ayuda del diplomado Desarrollo e Implementación de Sistemas con Software Libre en Linux

Este proyecto se puede aplicar a distintas instituciones relacionadas con cursos de cómputo e implica ayuda de profesionistas expertos en la materia para poder lograrlo.

¿Qué está pasando?

Actualmente algunas instituciones donde se imparte Cursos de Cómputo carecen de estos servicios, no ofrecen asistencia para el registro de cursos en línea y con este sistema distintas personas podrán consultar los diferentes Cursos y/o Diplomados que ofrece dicho centro, es por eso, que el propósito de esta investigación es desarrollar un sistema en el que los usuarios se puedan registrar a través de Internet y esto se podrá lograr con el diseño de una base de datos y una página electrónica que van a interactuar con el cliente.

ALCANCES Y LIMITACIONES

- La creación de una herramienta para el registro de los alumnos, que lleve el control, administración y manejo de los Cursos que desea.
- El software desarrollado permitirá recuperar información de los alumnos de forma rápida y eficiente.
- **El PROYECTO DE UN SISTEMA DE CONSULTA DE INFORMACION DE CURSOS, EN LINEA BASADO EN SOFTWARE LIBRE Y TECNOLOGÍA ORACLE** será independiente de la plataforma de cómputo a usar.
- La herramienta tendrá las opciones de alta, baja y modificación de los distintos alumnos.
- El desarrollo de este proyecto es viable ya que se requiere de este sistema para poder ofrecer los distintos cursos a través de Internet.

OBJETIVOS

Objetivo General.

Desarrollar un sistema de información, que proporcione información de los cursos que son impartidos en los centros de cómputo y además almacene los distintos datos de las personas interesadas en los cursos este registro se podrá realizar en línea a través de una página Web.

Objetivo particular:

Uno de los objetivos más importantes es, además de realizar un sistema eficiente, poder usar software libre.

Objetivos específicos.

- Realizar la creación de una base de datos con Tecnología Oracle en la plataforma Linux.
- Destacar los elementos básicos de un sistema de información.
- Practicar en función de la conectividad de la aplicación y creación de las Bases de Datos sobre el ambiente Linux.

CAPÍTULO I.

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

1.1 Conceptos básicos.

1.1.1 ¿Que es Linux?

Linux es un sistema operativo que ha sido diseñado y construido por cientos de programadores alrededor del mundo. La meta ha sido crear un clon de Unix, libre de cualquier licencia comercial (como lo es copy right), que cualquiera en el mundo lo pueda utilizar. Linux iniciado por Linus Torvalds, cuando era estudiante de la Universidad de Helsinki en Finlandia. Su meta era crear un reemplazo del sistema operativo Minix, el cual es un sistema operativo parecido a Unix para computadoras personales.

LINUX está compuesto por capas, la capa más interna está conformada por el hardware, que es el conjunto de piezas físicas del equipo de cómputo. Otra capa es el Kernel o núcleo del sistema.

El **kernel** (o **núcleo**) es la parte fundamental de un sistema operativo. Es el software responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma más básica, es la encargada de gestionar recursos, a través de servicios de llamada al sistema, este proceso consiste en recibir mensajes de diferentes fuentes y enviarlas a un destino común.

La función principal del kernel es interpretar las instrucciones proporcionadas por el usuario y convertirlas en lenguaje de máquina e indicarle al hardware lo que tiene que realizar con dicha información.

Como hay muchos programas y el acceso al hardware es limitado, el núcleo también se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo, lo que se conoce como multiplexado.

1.1.2 ¿Qué son las distribuciones?

Una de los primeros conceptos que aparecen al iniciarse en Linux es el concepto de distribución. Una distribución es un agrupamiento del núcleo del sistema operativo Linux (la parte desarrollada por Linus Torvalds) y otra serie de aplicaciones de uso general o no tan general

Las distribuciones más conocidas son:

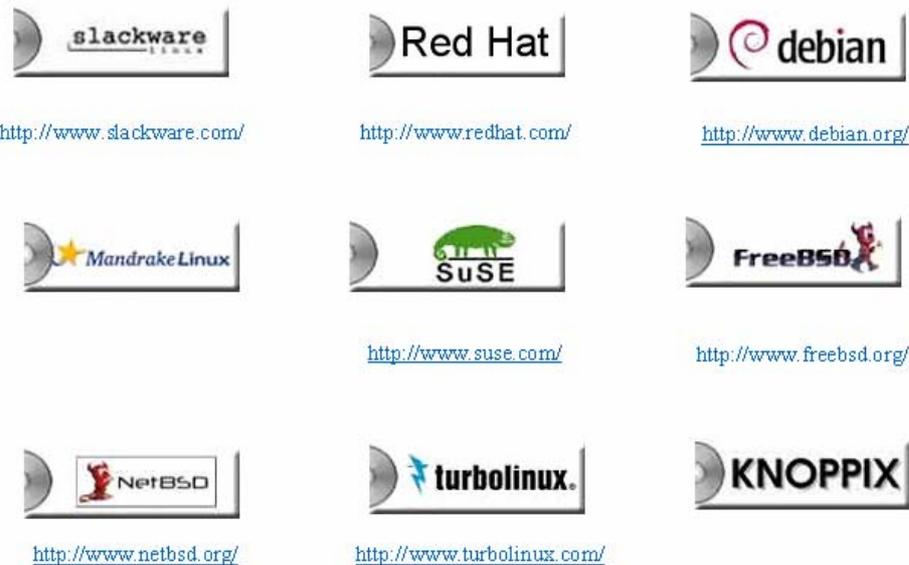


Figura 1. Distribuciones Linux

1.1.3 Características de LINUX:

Las principales características del sistema operativo linux es que es:

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

1.1.4 Entorno de trabajo: shell y X Window

Al contrario que otros sistemas operativos, por defecto el trabajo con Linux no se realiza de una forma gráfica, sino introduciendo comandos de forma manual. Linux dispone de varios programas que se encargan de interpretar los comandos que introduce el usuario y realiza las acciones oportunas en respuesta. Estos programas denominados shell son el modo típico de comunicación en todos los sistemas Unix incluido Linux.

Para muchas personas el hecho de tener que introducir los comandos de forma manual les puede parecer intimidante y dificultoso, aunque como se verá más adelante los comandos de Linux son relativamente simples y muy poderosos.

No obstante, casi todas las distribuciones más recientes incluyen el sistema X Window, el cual es el encargado de controlar y manejar la interfaz grafica de usuario.

1.1.5 Usuarios y grupos

Linux es un sistema operativo multitarea y multiusuario. Esto quiere decir que es capaz de ejecutar varios programas (o tareas) de forma simultánea y albergar a varios usuarios de forma simultánea.

Por lo tanto, todos los usuarios de Linux deben tener una cuenta de usuario en el sistema que establezca los privilegios del mismo. A su vez Linux organiza a los usuarios en grupos de forma que se puedan establecer privilegios a un determinado grupo de trabajo, para el acceso a determinados archivos o servicios del sistema.

1.1.6 Instalación

Al contrario que lo que ocurre con Microsoft Windows la instalación de Linux no es un proceso sencillo, puesto que Linux permite el control y la personalización de una cantidad mayor de parámetros y opciones. Pese a todo se están realizando grandes progresos buscando que la instalación de Linux sea un proceso mas sencillo, dependiendo la distribución que se emplee.

Uno de los conceptos principales a tener en cuenta antes de la instalación es el de partición. Cada sistema operativo organiza la información de los archivos que contiene de forma diferente, utilizando cada uno su propio sistema de archivos. Como referencia se indica a continuación el nombre del sistema de archivos de diferentes sistemas operativos:

Sistema Operativo	Sistema de archivos
MS-DOS	FAT
MS Windows 95	VFAT
MS Windows 95 OSR2 y Windows 98	FAT32
MS Windows NT	NTFS
IBM OS/2 Warp	HPFS
Linux	Ext2

Esto en general impide que se puedan instalar varios sistemas operativos mezclados en un mismo disco duro. Para solucionar este problema existen las llamadas particiones con las que se divide un determinado disco duro de forma que pueda contener ambos sistemas de archivos. A todos los efectos realizar una partición es equivalente a que el disco duro se divida en dos.

Existe un concepto fundamental a la hora de instalar y usar Linux que es el de Súper Usuario o usuario root. Este usuario es el administrador del sistema y se crea durante la instalación. Como administrador que es, puede acceder y modificar (así como destruir) toda la información del sistema, por lo que hay que evitar en la medida posible trabajar como usuario root.

1.2 Iniciación a Linux

Existen tres formas de acceder a un sistema Linux:

A través de una consola de texto. El usuario se conecta directamente a la computadora que tiene instalado Linux y accede mediante un sistema no gráfico.

Desde un gestor de sesiones gráfico (X Window). El usuario se conecta directamente a la computadora que tiene instalado Linux y accede al sistema mediante un programa gráfico.

Desde una computadora remota mediante telnet o secure shell.

1.2.1 Entrada y salida del sistema

En cualquiera de las formas para acceder a un sistema Linux en la pantalla aparecerá lo siguiente:

Login: (Se teclea el nombre del usuario)

Password: (Se teclea la contraseña.)

Por motivos de seguridad la contraseña debe cumplir ciertas condiciones tales como:

- Contener al menos seis caracteres.
- Contener al menos un carácter numérico o especial y dos alfabéticos.
- Ser diferente del nombre de login.

La primera vez que se accede al sistema, la contraseña empleada será la proporcionada por el administrador del sistema.

Salida Del Sistema

Es necesario indicarle al sistema que queremos terminar la sesión de trabajo en el servidor, para ello existen diversas formas para terminar la sesión de trabajo en Linux, dependiendo de si estamos en modo gráfico o de texto.

Existen varios comandos, entre ellos se encuentran los comandos exit y logout o la combinación de teclas Ctrl+d. Ambos comandos nos permiten terminar la sesión en el servidor, al momento que lo tecleamos y presionamos enter se cierra la conexión en el servidor. La salida de X Window depende del gestor de ventanas que se esté ejecutando.

El modo texto se basa en la ejecución de una serie de comandos, que son interpretados por un programa o shell. Linux dispone de varios de estos programas pero el más habitual es conocido como bash o Bourne Shell. Si Linux se ha arrancado en modo texto el sistema arranca de forma directa el shell y queda a la espera de introducción de nuevos comandos. Si se ha arrancado en modo gráfico se puede acceder al shell de dos formas:

Se puede acceder al shell del sistema presionando alguna de las siguientes combinaciones de teclas:

```
<ctrl>+<alt>+<F1>  
<ctrl>+<alt>+<F2>  
<ctrl>+<alt>+<F3>  
<ctrl>+<alt>+<F4>  
<ctrl>+<alt>+<F5>  
<ctrl>+<alt>+<F6>
```

Esto hace que el sistema salga del modo gráfico y acceda a alguna de las seis consolas virtuales de Linux, a las cuales también se puede acceder cuando se arranca en modo de texto. Para volver al modo gráfico hay que presionar <ctrl>+<alt>+<F7> o <ctrl>+<alt>+<F8>.

1.2.2 Sistema De Archivos

LINUX está compuesto de un sistema de archivos jerárquico en el cual no existen unidades de disco, en su lugar cada unidad de almacenamiento así como cada dispositivo de hardware son reconocidos como un archivo o directorio dentro del sistema. Existe sólo una raíz en el sistema la cual representa la parte más alta de la estructura de directorios y es conocida como root o raíz, es representada por el símbolo: / a partir de este punto se desprenden diferentes ramas de directorios como las siguientes:

Directorios Principales:

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

1.2.3 Comandos Básicos

Existen una serie de nociones básicas que hay que tener en cuenta a la hora de introducir los comandos. En primer lugar citaremos las siguientes:

- Los comandos hay que teclearlos correctamente.
- Las letras mayúsculas y minúsculas se consideran como diferentes.

En su forma más habitual (los shells de Bourne o de Korn), el sistema operativo utiliza un signo de \$ como prompt para indicar que está preparado para aceptar comandos, aunque este carácter puede ser fácilmente sustituido por otro u otros elegidos por el usuario. En el caso de que el usuario acceda como administrador este signo se sustituye por #.

Cuando sea necesario introducir el nombre de un archivo o directorio como argumento a un comando, Linux, permite escribir las primeras letras del mismo y realiza un autorrellenado al presionar la tecla del tabulador. Si no puede distinguir entre diversos casos rellenará hasta el punto en el que se diferencien. Por **Ejemplo**, supongamos una carpeta con los siguientes directorios:

```
Programas
Documentos_proyecto
Documentos_privados
```

Al escribir `cd Pr <tab>` Linux rellenará el resto del contenido hasta escribir `cd`

`Programas`. Por el contrario al escribir `cd D <tab>` escribirá `cd Documentos_`

Estructura básica de los comandos. (Comando -opciones argumento1 argumento2...)

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

Las opciones modifican el funcionamiento de un comando, generalmente se utilizan para agregar información a la salida de un comando o aumentar el potencial del comando a ejecutar, las opciones no son necesarias para la ejecución del comando, siempre son precedidas de un guión y se puede utilizar más de una opción.

Los argumentos son requisito para la ejecución del comando, si no damos el argumento apropiado para la ejecución del comando, éste no se ejecutará y mandará un mensaje de error el sistema.

Algunos Comandos Sencillos de LINUX.

Comando	Descripción
date	Muestra por pantalla el día y la hora.
cal 1949	Muestra el calendario del año 1949.
cal 05 1949	Muestra el calendario de mayo de 1949.
who	Indica qué usuarios tiene el ordenador en ese momento, en qué terminal están y desde qué hora.
whoami	Indica cuál es la terminal y la sesión en la que se está trabajando.
man comando	Todos los manuales de Linux están dentro del propio sistema operativo, y este comando permite acceder a la información correspondiente al comando comando. Por Ejemplo con man who aparecerá por pantalla y de forma formateada por páginas, la explicación del comando who. Se puede navegar a través de estas páginas con los cursores del teclado, y presionando q para salir.
clear	Este comando limpia la consola.
Pwd	Imprime en pantalla el directorio actual de trabajo, es decir, donde nos encontramos posicionados dentro del sistema de directorios indicándonos la ruta absoluta.
touch	Este comando nos permite crear archivos vacíos, de tamaño 0.
mkdir	Éste comando nos permite crear directorios,
rmdir (remove directory)	Éste comando borra directorios vacíos.
ls (list)	Muestra un listado del contenido del directorio actual de trabajo.
ls -l	Muestra un listado con formato largo indicando todas las propiedades de los archivos y directorios del directorio actual de trabajo.
ls -a	Muestra todo el contenido del directorio, incluyendo los archivos y directorios ocultos.
ls -i	Muestra el listado con el inodo donde se encuentra almacenado el archivo en el disco.
ls -r	Muestra el listado en orden inverso alfabético.

ls -R	Muestra un listado en forma recursiva, es decir lista el contenido de directorios y subdirectorios a partir de la ruta donde le indiquemos.
ls -t	Muestra un listado tomando como opción la fecha de modificación del archivo o directorio.
Cat	Nos permite ver el contenido de uno o más archivos, su función original es concatenar archivos. Ejemplo: cat archivo1 archivo2
Cd	Este comando nos permite movernos entre directorios, es necesario que le indiquemos la ruta del directorio al que nos queremos cambiar. Ejemplo: cd dir1/dir2/dir3
cp (copy)	El comando cp nos permite hacer copias de archivos uno por uno, múltiples archivos o incluso hacer copias de estructuras de directorios. Ejemplo: cp ruta_origen ruta_destino

Ejemplos de cp:

cp arch1 arch2 arch3 arch4 /ruta_destino

Aquí se esta haciendo una copia de múltiples archivos del directorio actual a una ruta destino enumerando uno a uno.

cp /dev/*.res .

Aquí se esta haciendo una copia de todos los archivos que terminen en .res dentro del directorio actual de trabajo.

cp -r informe respaldo

Aquí se esta haciendo una copia recursiva del directorio informe dentro del directorio respaldo.

cp -i * respaldo/

La opción **-i** indica al sistema que realice una copia interactiva de tal forma que si se esta copiando un archivo con un nombre de un archivo que ya existe, el sistema pedirá confirmación para sobrescribir el archivo existente.

cp -v * respaldo .

La opción **-v** nos va informando del nombre de los archivos a los cuales se les esta haciendo la copia.

Comando	Descripción
rm (Remove)	El comando rm borra archivos, puede ser borrado un solo archivo o varios haciendo uso de los meta caracteres *
rm -r directorio	La opción -r hace un borrado en forma recursiva, esta opción se debe usar con mucho cuidado ya que se borrará el directorio especificado y todo su contenido.
rm -f *	La opción -f fuerza a que los elementos sean borrados aunque nosotros no tengamos permisos sobre ellos.
mv (move)	El comando mv tiene dos funciones básicas, la primera es mover archivos o estructuras de directorios de un lugar a otro, la segunda es renombrar archivos o directorios.

Ejemplos:

Aquí se está moviendo el archivo saludo dentro del directorio carta.

```
mv saludo carta/
```

Aquí se está cambiando el nombre del archivo saludo, ahora se llamará hola.

```
mv saludo hola
```

Comando	Descripción
more	Este comando nos permite ver el contenido de uno o mas archivos imprimiéndolos en pantalla, para avanzar una pantalla se presiona la barra espaciadora, para avanzar una línea se presiona la tecla de enter. Ejemplo: <code>ore archivo1 archivo2...</code>
less	La orden more es bastante útil, pero a menudo pasa que se ha adelantado más allá de la pantalla que se quería visualizar, more no proporciona una manera de retroceder. El comando less proporciona esa función. Se usa de la misma forma que el comando more.
head	Nos muestra las primeras 10 líneas contenidas en un archivo Ejemplo: <code>head nom_archivo.</code>
Head -n	Esta opción le indica al comando que despliegue las primeras n, donde n es igual al número de líneas, en lugar de 10. Ejemplo: <code>head -20 nom_archivo</code>
tail	Nos muestra las últimas 10 líneas contenidas en un archivo: Ejemplo: <code>tail nom_archivo</code>
tail -n	Esta opción le indica al comando que despliegue las últimas n número de líneas en lugar de 10. Ejemplo: <code>tail -20 nom_archivo</code>

Filtros (Grep)

Este comando busca patrones en archivos. Por defecto devuelve todas las líneas que contienen un patrón determinado en uno o varios archivos. Utilizando las opciones se puede variar mucho este comportamiento. Si no se le pasa ningún archivo como argumento hace la búsqueda en la entrada estándar.

Sintaxis:

```
grep [opciones] <patrón> [archivos]
```

Algunas opciones:

-c :	Devuelve sólo la cantidad de líneas que contienen al patrón.
-i :	Ignora las diferencias entre mayúsculas y minúsculas.
-H :	Imprime además de las líneas, el nombre del archivo donde se encontró el patrón. Es así por defecto cuando se hace la búsqueda en más de un archivo.
-l :	Cuando son múltiples archivos sólo muestra los nombres de aquellos donde se encontró al patrón y no las líneas correspondientes.
-v :	Devuelve las líneas que no contienen el patrón.
-r :	Busca en un directorio de forma recursiva.
-n :	Imprime el número de cada línea que contiene al patrón.

Ejemplos:

```
$ grep -H -r linux /usr/share/doc
```

```
# grep -n error /var/log/messages
```

```
$ grep -i ftp /etc/passwd
```

```
$ grep -c root /etc/group
```

```
$ grep -l -r -i img /var/www/icons/*
```

1.2.4 Rutas (Paths)

En cualquier sistema operativo moderno la estructura de archivos es jerárquica y depende de los directorios. En general la estructura del sistema de archivos se asemeja a una estructura de árbol, estando compuesto cada nudo por un directorio o carpeta, que contiene otros directorios o archivos.

En **Windows** cada unidad de disco se identifica como una carpeta básica que sirve de raíz a otras, y cuyo nombre es especial **a:**, **c:**, **d:** etc. En los sistemas **Unix**, y por lo tanto en **Linux**, existe una única raíz llamada / de la que se originan todos los archivos y directorios, y que es independiente de qué dispositivos estén conectados a la computadora.

El camino o path de un archivo o directorio es la secuencia de directorios que se ha de recorrer para acceder a un determinado archivo separados por /. Supongamos la estructura de archivos de la siguiente figura.

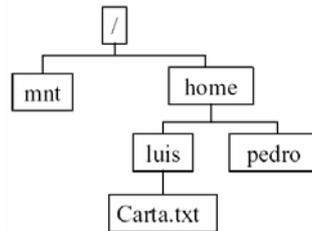


Figura 2. Path.

Existen dos formas del path o camino:

- El camino absoluto que muestra toda la ruta a un archivo, /home/luis/Carta.txt.
- El path relativo a un determinado directorio, por **Ejemplo** si no encontramos en el directorio /home, el path relativo al archivo Carta.txt es luis/Carta.txt

Es posible utilizar rutas absolutas o rutas relativas según sea nuestra conveniencia.

Las rutas absolutas siempre inician con el símbolo de / que indica el punto más alto de la estructura de directorios, a partir de aquí se debe de poner la ruta completa para llegar al punto que se quiera afectar.

Ejemplo:

```
cd /var/spool/mail
```

Las rutas relativas nunca inician con el símbolo de raíz / en lugar de ello inician la ruta a partir del directorio actual de trabajo.

1.2.5 Redireccionamiento

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

- Redireccionamiento de salida

El carácter mayor que (>) nos permite redireccionar la salida de un programa hacia un archivo,

Ejemplo:

```
ls > milista
```

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

- Redireccionamiento de entrada

El carácter menor que (<) nos permite redireccionar un archivo hacia un programa,

Ejemplo:

```
cat < milista
```

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

```
mail usuario < milista
```

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

- Redireccionamiento de un programa a otro programa

El carácter pipe (|) funciona como un filtro mandando la salida de un programa hacia la entrada de otro programa, enlazando la salida de un programa con la entrada de otro programa,

Ejemplo:

```
ls -l | more
```

Aquí se está enlazando la salida del comando ls con el programa more, el cual nos mostrará el resultado del comando ls por pantalla cada vez que presionemos la tecla espaciadora.

- Redireccionamiento no destructivo o de adición.

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

Ejemplo:

```
ls >> milista
```

- Redireccionamiento condicionado

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

Ejemplo:

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

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

1.2.6 Atributos de los archivos.

Tipos de archivos:

Dentro del sistema operativo vamos a encontrar archivos de diferentes tipos, los cuales los vamos a distinguir por el primer bit de sus propiedades.

El atributo del primer bit nos indica si el elemento listado es directorio o archivo, así mismo nos indica de qué tipo de archivo se trata.

Archivos más comunes.

Archivo de bloque (acceso por bloques en paralelo como la impresora).

d	Nos indica que el elemento es un directorio.
-	Archivo ordinario.
l	El elemento listado es una liga suave.
c	Archivo de tipo carácter (Archivos de acceso de hardware mediante un bit como el mouse).
b	Archivo de bloque (acceso por bloques en paralelo como la impresora).

Permisos

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

r	lectura
w	escritura
x	ejecución

Debemos asignar los permisos a los archivos de acuerdo a la función que van a realizar dependiendo para lo que fueron creados.

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

Estos permisos se encuentran asignados en tres bloques, los cuales representan al usuario o dueño del elemento listado, al grupo al que pertenece el usuario y a los otros usuarios del sistema, de ello que los atributos se componen de 10 bits, el primer bit indica el

tipo de elemento que está listado (archivo o directorio), los siguientes 9 bits representan los tipos de permisos asignados a cada elemento listado.

Ejemplo de listado de permisos

```
dante% ls -l
total 426
-rw-r--r--  1 carlos  becas  30208 Sep 29 23:54 apa1.jpg
-rw-----  1 carlos  becas   1170 Sep 29 20:14 dead.letter
-rw-r--r--  1 carlos  becas  86796 Sep 29 23:53 Guerra1.jpg
drwxr-xr-x  2 carlos  becas   512 Sep 30 01:16 hola
lrwxrwxrwx  1 root    other    5 Sep 28 15:11 hola1 -> hola1
-rw-r--r--  2 carlos  becas    0 Sep 28 15:11 liga
-rw-r--r--  2 carlos  becas    0 Sep 28 15:11 liga1
-rw-----  1 carlos  becas  13167 Sep 29 23:56 mbox
-rw-r--r--  1 carlos  becas  40007 Sep 30 01:11 mi1
-rw-r--r--  1 root    other  40007 Sep 28 13:44 nied
drwxr-xr-x  3 carlos  becas   512 Sep 30 01:17 respaldo
dante% █
```

Figura 3. Permisos de los archivos.

Comando chmod (permisos)

Este comando nos permite modificar los permisos a los archivos y directorios, la asignación de permisos puede hacerse de dos formas, mediante el método simbólico o mediante el método octal.

Asignación de permisos:

chmod indica al sistema el cambio de permisos.

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

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

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

Método octal

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

X	vale	1
W	vale	2
R	vale	4

Quedando nuestra tabla de la siguiente forma:

Asignación	Permisos
0 _ _ _	Ninguno
1 _ _ x	Ejecución
2 _ w _	Escritura
3 _ w x	Escritura y Ejecución
4 r _ _	Lectura
5 r _ x	Lectura y Ejecución
6 r w _	Lectura y Escritura
7 r w x	Lectura, escritura y ejecución

De tal forma que si deseamos poner todos los permisos para el usuario, permisos de lectura y ejecución para el grupo y ninguno para los otros el comando quedaría de la siguiente forma:

Ejemplo:

```
chmod 750 hola
```

Sustituyendo el número correspondiente para cada bloque de permisos al usuario, al grupo y a los otros.

1.2.7 Comandos básicos de red

Comunicación con los usuarios

Dentro del sistema operativo LINUX existen diferentes comandos que nos permiten mantenernos en constante comunicación con los usuarios, sin importar el lugar donde se encuentren ni el servidor donde se encuentre alojada su cuenta, a continuación se describen algunos de estos programas.

who muestra un listado sencillo de quien se encuentra conectado en el servidor, a que hora se conecto y que esta haciendo.

w Muestra un listado con detalles de los usuarios conectados en el servidor.

whoami me indica con que login estoy registrado dentro del servidor, este comando es útil ya que es posible que un usuario se convierta en otro diferente utilizando el comando su

finger nos muestra información de los usuarios conectados en el servidor, entre ella el nombre real del usuario.

finger login o **finger nombre** nos muestra información en específico del usuario que le estamos indicando, entre la que se encuentra el nombre real del usuario, directorio \$HOME, última fecha de acceso al servidor, etc.

finger usuario

Correo electrónico

mail cuenta@mail.com El comando mail nos permite enviar correo electrónico a cualquier usuario que tenga acceso a una cuenta de e-mail.

mail usuario@atena.mascarones.unam.mx

Todos los correos revisados con este programa son almacenados en un archivo llamado mbox que se encuentra en nuestro directorio \$HOME

mail tecleando el comando sólo revisa la lista de correos, en caso de que no tengamos correo mandará un mensaje de que no hay correo en nuestra cuenta.

mail cuenta@correo.com

Revisión del correo.

mail. Tecleando este comando sólo nos dará acceso a una lista del correo que hayamos recibido, aquí se describen algunas opciones para la revisión de nuestro correo.

- h nos mostrará una lista numerada de los títulos de los correos recibidos.
- # Tecleando el número que aparece en el listado de correos veremos el contenido del correo deseado.
- r manda una replica del correo que estamos visualizando en ese momento.
- q sale del programa mail guardando todos los cambios que hayamos realizado como borrado de correos, marcado de correos ya leídos, etc.
- x sale del programa mail sin salvar cambios en la lista, es decir que no marcara los correos que ya consultamos y los mantendrá marcados como nuevos
- d# Borra el correo marcado en la lista con el número que le estamos indicando.

pine

Al igual que mail nos permite trabajar con el servicio de correo electrónico pero este programa tiene muchas ventajas en comparación con el comando mail, presenta un menú de opciones, configuración del servicio de correo, envío de archivos adjuntos, etc.

Para ejecutar este programa basta con teclear pine desde el símbolo de sistema y presionar la tecla de enter.

Todos los correos que son revisados con este programa son almacenados en un archivo que lleva el mismo nombre que nuestro login y se encuentra en la ruta `/var/spool/mail/`

Esta ruta puede cambiar dependiendo de la distribución y la versión del sistema operativo que estemos usando.

write Este comando nos permite enviar un mensaje a un usuario, siempre y cuando el usuario se encuentre conectado en el servidor.

write usuario

Terminar mensaje

Ctrl+d

talk. Este programa permite realizar una conversación interactiva en tiempo real, es posible mantener una conversación con más de dos usuarios al mismo tiempo, es necesario que los usuarios se encuentren conectados en el servidor.

talk login1 login2 login3 etc

Es necesario que haya un mínimo de tres líneas para cada usuario, el número de usuarios para acceder en una conversación esta limitado por el número de líneas disponibles en pantalla.

Es recomendable que la terminal este configurada como vt100 ya que con esta terminal trabaja apropiadamente éste programa.

El comando echo **\$TERM** nos mostrará el tipo de configuración que tiene actualmente nuestra terminal, si nuestra terminal no tiene el tipo apropiado podemos definirlo mediante el comando

TERM=vt100

Así cambiaremos el valor de la variable y la configuración de la terminal por la que consideremos más apropiada.

ytalk. Éste comando tiene funciones muy parecidas a talk, sólo que este comando permite realizar una solicitud de conversación a más de un usuario, como único requisito es que se tengan un mínimo de tres líneas para cada usuario.

ytalk usuario@yahoo.com

En versiones más recientes es posible que talk tenga el mismo funcionamiento que ytalk

wall. Este comando nos permite enviar un mensaje a todos los usuarios que se encuentran conectados al servidor, el mensaje es escrito directamente en pantalla.

Desactivar mensajes en pantalla

mesg Este comando nos indica el estado en que se encuentra configurado para nosotros como usuario, sólo tiene dos estados, y ó n.

La opción **yes**, indica que está habilitada la recepción de mensajes mientras que la opción **no**, indica que no está habilitada la recepción de mensajes, en ocasiones es conveniente deshabilitar la recepción de mensajes como cuando estamos realizando tareas que requieren toda nuestra atención como cuando editamos un archivo con vi o cuando estamos elaborando algún programa.

Para habilitar la recepción de mensajes debemos dar la instrucción

mesg y

Para deshabilitar la recepción de mensajes debemos dar la instrucción

mesg n

diff Este comando activa o desactiva la recepción de mensajes proporcionados por el sistema como son la recepción de correo electrónico, etc.

Al igual que mesg tiene solo dos estados: activo y desactivo.

fortune Elige de una base de datos el mensaje del día como el que aparece en pantalla cada vez que iniciamos sesión.

date muestra la hora del sistema.

df Muestra el estado de las particiones de los discos duros indicándonos el espacio utilizado y el estado libre en disco.

du muestra el espacio utilizado entre archivos y directorios a partir del directorio desde donde se ejecutó el comando.

last muestra las últimas conexiones de los usuarios al servidor.

last -20 muestra los últimos 20 registros al servidor.

last -20 usuario muestra los últimos 20 registros al servidor del usuario carlos

sort Ordena el contenido de un archivo por líneas en orden alfabético.

sort -r ordena el contenido de un archivo en forma inversa alfabéticamente.

wc cuenta las líneas, palabras y caracteres contenidos en un archivo.

wc -l Cuenta únicamente las líneas

wc -w Cuenta únicamente las palabras

wc -c Cuenta únicamente los caracteres

banner este programa permite hacer carteles de tipo banner, por default tiene un ancho de 132 líneas lo que impide que el banner pueda ser leído directamente en pantalla; para ello se debe utilizar la opción **-w#** para cambiar el número de columnas que abarcará el mensaje.

banner -w35 saludos a todos.

Aquí se está desplegando un banner en pantalla con el mensaje saludos a todos teniendo un ancho de 35 columnas.

cal el sistema cuenta con un calendario el cual comprende desde el año 1 hasta el año 9999, por default el comando cal nos va a entregar el calendario del mes en curso.

cal 2005 nos mostrará el calendario del año 2005

cal 6 1975 nos mostrará el calendario del sexto mes del año 1975

cal -j 2005 nos mostrará la numeración de los 365 días del año 2005 en lugar de desplegar la numeración de cada uno de los meses.

cal -y mostrará el calendario del año en curso.

cal 9 1752 Aquí se observará que falta la numeración entre los días 2 y 14 de Septiembre de 1752, esto es debido a que el Papa Gregorio XIII decretó en el año 1582 que se debían ajustar todos los calendarios debido al ajuste de los años bisiestos.

1.2.8 Procesos

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

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

Cuando nosotros iniciamos una sesión dentro del servidor en realidad estamos ejecutando una copia del interprete de comandos que tenemos por default, a esta copia del interprete se le asigna un número el cual está asociado al proceso en ejecución, por cada sesión que nosotros iniciemos se va a ejecutar una nueva copia del shell asignándole un nuevo número de identificación para este proceso.

Es importante recalcar que todo proceso tiene una dependencia del shell principal del sistema, en nuestro caso como usuarios todos los procesos que generemos van a depender de la copia del shell que nosotros estamos ejecutando.

ps Este comando permite ver los procesos que se están ejecutando desde nuestra sesión abierta, y que se encuentra en uso en este momento, muestra un listado de los procesos en formato corto.

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

ps -f entrega un listado de los procesos que se están ejecutando en nuestra sesión, muestra los procesos en formato largo, donde podemos ver la dependencia de los procesos por el UID, PID y el PPID

El UID es el identificador del Usuario dueño del proceso

El PID es el número del identificador del Proceso en ejecución.

El PPID es el identificador del Proceso Padre del Proceso en ejecución.

ps -e Muestra un listado en formato sencillo de todos los procesos ejecutándose en el servidor.

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

Los procesos pueden encontrarse en diferente estado dependiendo de la atención que estén teniendo en un momento en específico.

ps -h nos indica el estado en que se está ejecutando los procesos.

Estados de los procesos:

- Z** Zombie, cuando un proceso corre en forma independiente, incluso del shell o del proceso que lo invocó.
- R** En ejecución, se encuentra en la CPU.
- T** Stopped o detenido.
- S** Sleeping Durmiendo en espera de recibir una solicitud para iniciar su trabajo.
- D** Dormido sin interrupción posible, normalmente relacionado con entrada/salida

Procesos en primer y segundo plano.

Una vez que se esta ejecutando un programa, nosotros podemos enviarlo a segundo plano presionando la combinación de teclas **ctrl+z**

jobs Éste comando nos permite ver los programas que se están ejecutando en segundo plano, nótese que a cada proceso en segundo plano se asigna un número que nos sirve para identificar el proceso.

jobs -l Muestra los procesos en segundo plano con formato largo.

Es posible enviar un proceso a segundo plano en forma automática desde el momento en que éste se inicia, esto se logra colocando el carácter de ampersan al final de la línea del comando **&**.

find / -name shadow 1> usuarios &

Generalmente los procesos que escriben directamente en pantalla no pueden enviarse a segundo plano como el **ls**.

Para mandar un proceso a primer plano debemos dar la instrucción **fg #** donde el **#** representa el número que nos entrega el comando **jobs**.

De esta forma podemos interactuar con nuestros procesos mandándolos a primer o segundo plano según nuestras necesidades de trabajo.

kill Éste comando nos permite entre otras opciones matar procesos, nosotros como usuarios únicamente podemos matar nuestros propios procesos.

kill -9 PID mata el proceso indicado

kill -15 PID mata el proceso pero al mismo tiempo lo reinicia asignándole el mismo PID, generalmente se utiliza esta opción para procesos que necesitan estar siempre en ejecución.

1.2.9 RespalDOS

tar nos permite empaCar una estructura de directorios, su función principal es almacenar el contenido de un directorio en forma recursiva dentro de un archivo.

Sintaxis: **tar [opciones] archivo.tar archivo**

Principales opciones de tar

- c** crea un nuevo archivo tarareado.
- v** verbose, imprime en pantalla todo aquello que esta afectando.
- f** Indica al sistema que va a afectar a un archivo
- x** Extrae el contenido de un archivo .tar

tar -cvf archivo.tar archivo

Aquí se esta creando un archivo utilizando la opción -c la opción -v muestra en pantalla todos los archivos y directorios que esta afectando, la opción -f indica que se va a enviar la salida del comando a un archivo, a continuación de las opciones se especifica el nombre que va a tener el archivo tarareado y por último el nombre del directorio que va a ser afectado.

tar -xvf archivo.tar

Aquí se esta recuperando la información almacenada dentro de un archivo tarareado, con la opción -x, una de las principales características de tar es que no le importa quien es el dueño del archivo, el grupo al que pertenece ni los permisos que tuvo el archivo al momento de ser tarareado, ya que cuando realizamos el proceso de destararear toda la información afectada toma los permisos que nosotros tenemos por default.

Como regla general cualquier opción es combinada con las opciones -vf ya que es recomendable ver qué esta realizando el sistema, así como para indicarle que se está enviando la salida del comando a un archivo, de lo contrario tomará por default la unidad de cinta.

-t Lista todos los archivos contenidos en un archivo .tar en formato largo mostrando todos sus detalles.

tar -tvf archivo.tar

-u update, actualiza el contenido del archivo tar realizando una búsqueda dentro de la estructura de directorios del sistema desde donde se inició el proceso de tarareo, añadiendo

todos los archivos y directorios que hayan sido creados o sufrido modificaciones después de haber sido creado el archivo tarreado.

tar -uvf archivo.tar

-w Esta opción nos pide confirmación para tarrear o destarrear cada elemento que va a ser afectado.

tar -cwvf archivo.tar

tar -xwvf archivo.tar

-d Opción no disponible en todas las versiones, hace una búsqueda de todos los archivos y directorios que hayan sufrido modificaciones después de haber sido creado el archivo .tar

tar -dvf archivo.tar

gzip Permite realizar una compresión de archivos, manejando 9 niveles de compresión que van desde el -1 al -9 donde el nivel 1 es el nivel de compresión más bajo mientras el nivel -9 es el nivel de compresión más alto.

Al momento de ejecutar esta utilera el programa le añade la extensión **.gz** al archivo comprimido, por default maneja el nivel de compresión 6.

Sintaxis: gzip [-n] archivo

La opción **-n** indica el nivel de compresión que se quiera realizar, el nivel óptimo es el -9 ya que realiza la compresión más alta que puede tener el archivo afectado.

-d Esta opción permite descomprimir un archivo gziado.

Sintaxis: gzip -d archivo

-f Fuerza la compresión o descompresión de la información afectada sin importar los permisos o propiedades que tenga el archivo o directorio al momento de ser afectado.

Sintaxis: gzip [-fn] archivo

-t Ésta opción hace una revisión del archivo tarreado para verificar su integridad, en caso de que el archivo no se encuentre íntegro el sistema mandara un mensaje indicando el tipo de incongruencia del archivo.

Sintaxis: gzip -t archivo.gz

-l Ésta opción muestra las propiedades del archivo tarreado indicando el tamaño del archivo cuando esta comprimido, el tamaño de la información cuando se encuentra descomprimida y el porcentaje de compresión de la información.

Sintaxis: gzip -lv archivo.tar.gz

1.2.10 Protocolo de Transferencia de Archivos (FTP)

FTP es un programa especialmente diseñado para la transferencia de archivos a través de Internet, no es confiable ya que deja pasar la información en texto claro, en dado caso de que algún paquete no sea recibido en forma íntegra el sistema recibe un mensaje de error y automáticamente reenvía el paquete sin que haya pérdida de información.

Para trabajar con el servicio de ftp se debe realizar un proceso de registro al servidor que se quiere acceder el cual se compone el inicio de sesión ftp, abrir la llamada al servidor, el registro del nombre del usuario y la contraseña de usuario para acceder al sistema mediante ftp.

A continuación debemos posicionarnos dentro del directorio en el cual queremos extraer o depositar la información.

1.2.10.1 Comandos básicos de ftp

ftp Inicia una sesión de ftp.

open 132.248.75.120 Abre la conexión de ftp en el servidor que se le indica.

Name (132.248.75.120:carlos): Solicita el nombre de usuario o login

Password: Solicita el password de acceso para el usuario.

Si todos los datos insertados son correctos tendremos acceso al servidor para realizar la transferencia de archivos.

ls -la |more nos mostrará el contenido del directorio por pantallas en formato completo.

cd para movernos entre directorios en el servidor.

lcd para movernos entre directorios en la máquina local.

hash Manda un símbolo de # cada 2048 bits transferidos.

binary Indica que la información será transferida en formato binario.

ascii Indica que la información será transferida en formato ascii o de tipo texto.

prompt Activa o desactiva el modo de interactividad con el servidor.

get permite descargar un archivo del servidor

mget permite descargar múltiples archivos del servidor, permite el uso de metacaracteres.

put nos permite subir un archivo al servidor.

mput permite subir múltiples archivos del servidor, permite el uso de metacaracteres.

!comando nos permite ejecutar un comando del shell de la máquina local.

help nos muestra una lista de los comandos disponibles en la sesión de ftp

help comando nos muestra la ayuda del comando que estamos consultando.

close o quit cierran la conexión de ftp en el servidor.

1.2.11 Ligas

Existen dos tipos de ligas, una de ellas es la de tipo suave y la otra es de tipo dura.

La liga de tipo suave es un vínculo hacia un archivo pero la liga es de tamaño muy pequeño ya que sólo hace referencia al archivo que se está ligando, funciona como un acceso directo y su función generalmente es asociar a un programa de nombre complejo con una liga la cual tiene un nombre más fácil de recordar.

Sintaxis de la liga suave

In -s nom_archivo nom_liga

El comando In nos permite generar las ligas, en este caso con la opción -s se indica al sistema que se está generando una liga de tipo suave.

La liga dura al igual que la liga suave mantiene una relación con un archivo el cual está ligado, la diferencia es que al comparar el tamaño de la liga con el archivo, ambos tienen el mismo tamaño, al editar el contenido de uno se actualiza automáticamente el otro, pero a comparación de un archivo común es que la liga está asociada al mismo inodo del archivo con el que está ligado.

Sintaxis de la liga dura

In nom_archivo nom_liga

En este caso se está generando una liga dura, al aplicar el comando In sin opciones el sistema interpreta el comando como la creación de una liga dura.

1.2.12 Reconocimiento del servidor y del sistema

hostname

nos muestra el nombre del servidor.

Ejemplo: atena

-i Esta opción nos muestra la dirección ip que tiene asignada el servidor de trabajo.

Ejemplo: 132.248.75.120

-d Nos muestra el dominio del servidor.

Ejemplo: mascarones.unam.mx

-f Muestra un listado completo incluyendo el nombre del servidor y su dominio.

Ejemplo: atena.mascarones.unam.mx

uname

Nos indica el tipo de LINUX que se esta utilizando.

Ejemplo: linux

-n Indica el nombre del servidor

Ejemplo: atena

-r Muestra la versión del kernel

Ejemplo: 2.4.18

-m Indica la arquitectura del procesador

Ejemplo: i586

-a Muestra todos los datos antes mencionados

1.2.13 Administración de Linux

El administrador de sistemas es la persona responsable de configurar, mantener y actualizar el sistema o conjunto de sistemas que forman una red.

Cuidando el funcionamiento del software, hardware y periféricos de forma que estén disponibles para ser utilizados por los usuarios.

1.2.13.1 Tareas del Administrador.

El Administrador debe:

- Proporcionar un ambiente seguro, eficiente y confiable
- Brindar un funcionamiento confiable del sistema.
- Dividir el trabajo entre varios administradores, dependiendo del tamaño del sistema.

Planear las actividades.

- Absolutamente todas las actividades de administración se planean.
- Guardar copias de seguridad.
- Jamás modificar sin respaldar previamente.

Tener copias de seguridad

Absolutamente todo lo que se almacene o realice en el sistema debe estar respaldado:

- Información de los usuarios
- Bases de datos
- Correo electrónico
- Archivos originales del sistema

Conocer las utilerías del sistema:

Utilerías del sistema:

- Básicas: cut, sort, paste, diff, comm, tail, head, grep, egrep, compress, etc.
- Control de tareas: at, crontab.
- RespalDOS: dump, dd, restore, tar, cpio.

Herramientas de programación:

- c
- shell.

Documentación:

- En línea (man, apropos, info).
- Impresa (libros, manuales).
- Internet.

Conocimientos requeridos

- 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.

Conocer el hardware

- El hardware de la máquina:
- Características, modelo, capacidad, etc.
- Ubicación física.
- Consultar las listas de hardware soportado publicadas por la distribución que se desea instalar (HCL)
- Hacer pruebas antes de adquirir el equipo

Establecer políticas de uso y administración

- Apertura de cuentas.
- Horas de mantenimiento.
- Responsabilidad de los respaldos.
- Borrado de archivos temporales.
- Cuotas de disco.
- Seguridad del sistema.

Mantener canales de comunicación con los usuarios:

Hoy en día existen varias formas y herramientas para estar en comunicación constante con los usuarios:

- /etc/motd
- News
- Wall
- Write
- Mail
- Web

1.2.13.2 Tareas administrativas comunes:

- Administración usuarios.
- Configuración de dispositivos.
- Programar respaldos periódicamente.
- Capacitar usuarios.
- Asegurar el sistema.
- Registrar los cambios del sistema.
- Asesorar 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).
- Monitoreo del sistema.
- Detección de fallas.
- Auditoría e implantación de la seguridad del sistema.
- Dar de alta y/o de baja el sistema.

El administrador de sistemas debe conocer las diferentes formas para dar de alta o de baja el sistema, de ello dependerá en gran medida el rendimiento y confiabilidad del sistema incluso después de un reinicio, también puede ser muy útil para rescatar un sistema cuando no inicia apropiadamente.

Primero es necesario comprender lo que sucede cuando encendemos el equipo y el proceso de arranque del sistema. Lo primero que realiza el proceso de arranque de Linux es cargar una copia del kernel que se encuentra en el directorio / boot, a continuación detecta todos los dispositivos de hardware y los pone a disposición del sistema, busca en los scripts de configuración para leer el nivel de inicio para el cual está configurado por default, luego busca los scripts del runlevel, finalmente ejecuta el script rc.local.

El nivel de inicio por default se encuentra definido en el archivo /etc/inittab, para ello necesitamos conocer los diferentes niveles que están predefinidos en el archivo de inicio:

```
# 0 = halt
# 1 = single user mode
# 2 = unused (but configured the same as runlevel 3)
# 3 = multiuser mode (default Slackware runlevel)
# 4 = X11 with KDM/GDM/XDM (session managers)
# 5 = unused (but configured the same as runlevel 3)
# 6 = reboot
```

Cabe mencionar que dependiendo del nivel de inicio la rutina de encendido apagado puede variar.

En la siguiente línea se muestra la configuración del nivel de inicio por default en Linux Slackware, como puede verse se encuentra definido el nivel 3 de inicio:

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

Si deseara que Linux iniciara por default en algún otro nivel solo tiene que cambiar el número 3 por el identificador numérico del modo en el que desea que inicie su sistema, por **Ejemplo**, si deseara que su computadora inicie en modo gráfico puede asignar el nivel 4 de inicio.

Para apagar el sistema podemos usar varios comandos:

- poweroff

- halt
- shutdown
- reboot
- init
- telinit

Aunque estos comandos nos permiten apagar o reiniciar el proceso puede variar, aquí hay que determinar cual es la forma más apropiada.

Poweroff:

Este comando hace un apagado inmediato del sistema, muy peligroso para ciertas tareas ya que no verifica que los procesos se hayan terminado en forma apropiada, da de baja todos los servicios y apaga el sistema, ahora este programa se ha actualizado para no causar daños al sistema o información por un apagado deficiente.

`poweroff -f` Fuerza el apagado del sistema.

Halt

Muy parecido a `poweroff`, solo que éste para apagar el sistema hace un llamado al comando `halt` quien en realidad es quien apaga el sistema, la desventaja de usar `halt` es que funciona muy parecido o `poweroff`.

`halt -f`

Fuerza el apagado del sistema, esta opción solo se debe utilizar como ultima alternativa.

Reboot

Este comando sirve para reiniciar el sistema, igualmente este programa no realiza un apagado eficiente ya que no verifica que los programas sean terminados eficientemente.

Shutdown

Este es considerado el comando más eficiente para detener, reiniciar el sistema, ya que su proceso de apagado es muy eficiente, de hecho "da de baja el sistema en una forma segura".

Se tiene control del apagado del sistema ya que se puede definir un tiempo, todos los usuarios son avisados de que el sistema se va a dar de baja mediante la instrucción SIGTERM, deshabilita el login para que nadie se pueda logear en el sistema, se comunica con el programa init y le solicita el nivel 0 de ejecución.

Este comando envía una señal de sincronización de discos, manda la señal de termino a todos los programas y los cierra eficientemente siguiendo su proceso normal de las aplicaciones, una vez que se han cerrado todos los procesos manda una nueva señal de sincronización para verificar que no hay tareas en ejecución, termina baja los demonios y finalmente apaga el sistema en forma segura.

shutdown -h now Apaga el sistema en forma inmediata

shutdown -r now Reinicia el sistema en forma inmediata

shutdown -t sec acción Realiza la acción con t numero de segundos de espera

shutdown -k No apaga el sistema, solo envía un aviso a todos los usuarios conectados.

shutdown -f acción Evita que se ejecute fsck al reinicio

shutdown -F acción Fuerza a que se ejecute fsck al reinicio

shutdown -c Cancela la instrucción shutdown

Shutdown +m acción Espera m minutos antes de realizar la acción

Mantenimiento de claves de usuarios

Una de las principales tareas es la administración de las cuentas de usuario, para ello se debe tener un plan administrativo de claves contemplando altas, bajas y cambios de las claves, también es importante definir los grupos a los que va a pertenecer uno o más usuarios así como se debe tener un especial cuidado en la capacitación de los usuarios para elegir contraseñas fuertes, asignar permisos adecuados y administrar apropiadamente los recursos para no tener un abuso de ellos.

Comandos para administrar los usuarios:

- Adduser Agrega usuarios modo comando
- Useradd Agrega usuarios modo comando
- Userdel Elimina usuario
- Usermod Modifica las propiedades del usuario
- Groupadd Agrega grupos
- Groupdel Elimina grupos
- Groupmod Modifica grupos
- Groups Lista los grupos a los que pertenece un usuario
- Passwd Modifica propiedades de usuarios/grupos

- Kuser Administra usuarios y grupos en modo grafico

Adduser. Es un programa que trabaja un asistente el cual permite agregar usuarios de forma rápida y fácil, realiza todos los procesos de forma transparente.

Useradd. Es el comando para agregar usuarios genéricos para todos los linux, es más complejo y se requiere conocimiento más profundo sobre administración de usuarios y grupos.

Se debe conocer los grupos que existen en el sistema, la secuencia de UID del sistema de usuarios entre otras cosas.

A parte de agregar los registros hay que crear el directorio del usuario, otorgarle privilegios de usuario y grupo y asignarle un password a la cuenta.

Userdel

Permite eliminar usuarios, aquí tenemos que utilizar la opción `-r` para que haga un borrado recursivo de todo lo que pertenece al usuario a eliminar, de otra forma solo se eliminarán los registros del usuario de los archivos `passwd`, `shadow` y `group`, posteriormente el administrador tendrá que eliminar todo lo que pertenecía al usuario a mano.

En ocasiones es necesario modificar el perfil de un usuario, para ello el sistema cuenta con el comando `usermod`, aunque tiene muchas opciones solo vamos a describir las más importantes.

- `usermod -u` modifica el UID del usuario
- `usermod -g` modifica el grupo principal de usuario
- `usermod -G` modifica los grupos adicionales del usuario
- `usermod -d` modifica el directorio HOME del usuario
- `usermod -s` modifica el shell por default
- `usermod -c` Modifica el comentario (campo de GECOS)
- `usermod -f` Asigna/modifica fecha de inactividad
- `usermod -e` modifica/asigna fecha de expiración de la cuenta
- `usermod -p` Cambia el password del usuario
- `usermod -L` Bloquea el acceso de la cuenta (Lock)
- `usermod -U` Desbloquea una cuenta bloqueada (Unlock)

1.3 HTML

Introducción

El World Wide Web (WWW) es un sistema que contiene mucha información. Pero esta información debe estar ordenada de alguna forma, de manera que sea posible encontrar lo que se busca. La unidad básica donde está almacenada esta información son las páginas Web. Estas páginas se caracterizan por contener texto, imágenes, animaciones e incluso sonido y video.

Una de las características más importantes de las páginas Web es que son hipertexto. Esto quiere decir que las páginas no son elementos aislados, sino que están unidas a otras mediante los links o enlaces hipertexto. Gracias a estos enlaces el navegante de Internet puede pulsar sobre un texto de una página para navegar hasta otra página.

En algunos ambientes se discute que el World Wide Web ya no es un sistema hipertexto sino hipermedia. Los que defienden el cambio se apoyan en que aunque en sus orígenes el WWW constaba únicamente de texto en la actualidad es un sistema principalmente gráfico y se puede hacer que las zonas activas sean, no sólo texto, sino imágenes, videos, botones, etc. en definitiva cualquier elemento de una página. Aún así, el término original no ha sido reemplazado todavía.

El lenguaje HTML

Este lenguaje estructura documentos. La mayoría de los documentos tienen estructuras comunes (títulos, párrafos, listas) que pueden ser definidas por el lenguaje mediante tags. Cualquier cosa que no sea un tag es parte del documento mismo.

`<tag> texto afectado </tag>`

Características del lenguaje HTML

El diseño del WWW debe cumplir entre otras, las siguientes características:

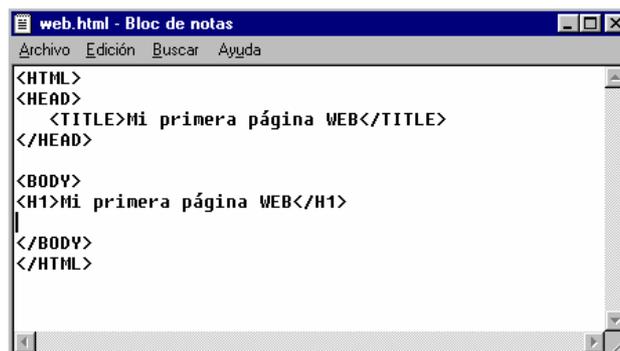
- El Web tenía que ser distribuido: La información repartida en páginas no muy grandes enlazadas entre sí.
- El Web tenía que ser hipertexto y debía ser fácil navegar por él.
- Tenía que ser compatible con todo tipo de computadoras (PCs, Macintosh, estaciones de trabajo) y con todo tipo de sistemas operativos (Windows, MS-DOS, UNIX, MAC-OS).

- Debía ser dinámico: el proceso de cambiar y actualizar la información debía ser ágil y rápido.

Estas características son las que marcaron el diseño de todos los elementos del WWW incluida la programación de páginas Web. Como respuesta a todos estos requisitos se creó el lenguaje HTML (HiperText Markup Language), cuyas siglas significan "lenguaje hipertexto de marcas".

Este lenguaje es el encargado de convertir un archivo de texto inicial en una página web con diferentes tipos y tamaños de letra, con imágenes impactantes, animaciones formularios interactivos, etc.

Una de las características de este lenguaje más importantes para el programador es que no es necesario ningún programa especial para crear una página Web. Gracias a ello se ha conseguido que se puedan crear páginas con cualquier computadora y sistema operativo. El código HTML, no es más que texto y por tanto lo necesario para escribirlo es un editor de texto.



```
web.html - Bloc de notas
Archivo Edición Buscar Ayuda
<HTML>
<HEAD>
  <TITLE>Mi primera página WEB</TITLE>
</HEAD>
<BODY>
<H1>Mi primera página WEB</H1>
</BODY>
</HTML>
```

Figura 4. Editor de texto simple.

Una vez que hemos escrito el código deberemos guardar el archivo con extensión .html o .htm. Los siguientes son nombres válidos de archivos que contengan código HTML: index.html, index.htm, principal.html, PRINCIPAL.htm

Estructura básica de un documento HTML

Cabecera y cuerpo del documento

Tres son las tags que describen la estructura general de un documento y dan una información sencilla sobre él. Estas tags no afectan a la apariencia del documento y solo interpretan y filtran los archivos HTML.

1. <HTML>: Limitan el documento e indica que se encuentra escrito en este lenguaje.

2. <HEAD>: Especifica el prólogo del resto del archivo. Son pocas las tags que van dentro de ella, destacando la del título <TITLE> que será utilizado por los marcadores del navegador e identificará el contenido de la página. Solo puede haber un título por documento, preferiblemente corto aunque significativo, y no caben otras tags dentro de él. En head no hay que colocar nada del texto del documento.
3. <BODY>: Encierra el resto del documento, el contenido.

```
<HTML>
  <HEAD>
    <TITLE>Ejemplo </TITLE>
  </HEAD>
  <BODY>
    Hola mundo
  </BODY>
</HTML>
```

En primer lugar deberemos especificar que el archivo de texto que estamos escribiendo es un documento HTML, para ello usamos las instrucciones de inicio y fin de la etiqueta <html> al principio y al final de la página respectivamente:

```
<html>
  Código de la página
</html>
```

El código de la página esta formado a su vez por dos grandes bloques, la cabecera y el cuerpo.

La cabecera de la página está delimitada por las instrucciones de inicio y fin de la etiqueta <head>.

Estas instrucciones deben estar dentro de la etiqueta HTML de la siguiente manera:

```
<html>
  <head>
    Elementos de la cabecera
  </head>
  ... Resto de código de la página...
</html>
```

En la cabecera de la página se introduce toda aquella información que afectará a toda la página. En un principio esta información se limitará al título. Este título se indicará con la etiqueta <title> usando la siguiente sintaxis:

```
<html>
<head>
  <title> Mi primera página WEB</title>
</head>
</html>
```

El resultado de ver esta página con nuestro navegador es el que se observa en la figura. Como vemos, el área de la pantalla donde suele estar la página está todavía vacía, pero si nos fijamos en la cabecera de la ventana del navegador vemos que el título ha pasado de ser "Mi primera página WEB. Es decir el título que le demos a nuestra página con la etiqueta <TITLE> pasará a ser el título de la ventana del navegador.

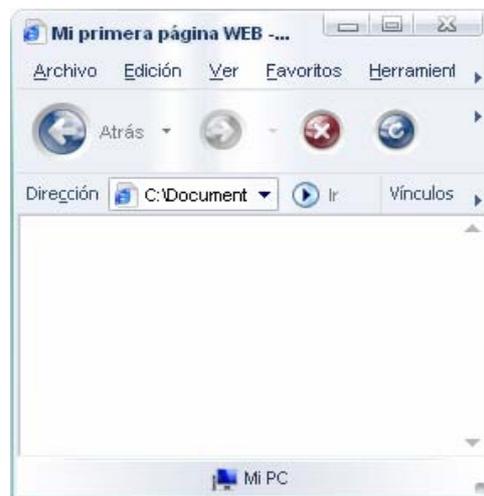


Figura 5. El título de la página es mostrado en la parte superior de la ventana del navegador.

Esta es la forma de crear páginas Web con HTML, considerando que existen otras etiquetas para dar el formato que se desee, de acuerdo al diseño con que se cuenta.

1.4 PHP

El PHP es un lenguaje embebido en páginas HTML y que se ejecutan en el servidor.

PHP significa Hypertext Preprocessor, aunque originalmente significaba Personal Home Page Tools. Los archivos PHP normalmente se denominan con la extensión php, php3 o phtml.

¿Que se puede hacer con PHP?

Al nivel más básico, PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies.

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir un interfaz vía Web para una base de datos es una tarea simple con PHP.

Las siguientes bases de datos están soportadas actualmente por PHP:

- Oracle
- PostgreSQL
- MSQL
- Sybase
- DB2
- MySQL
- Informix

PHP también soporta el uso de otros servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados. También se pueden abrir sockets de red directos (raw sockets) e interactuar con otros protocolos.

Sintaxis básica de PHP

Las instrucciones PHP están embebidas en HTML. Una página PHP es una página normal HTML que con unas marcas especiales le indican al servidor que deben interpretarse.

Por ejemplo:

ejemplo1.php

```
<html>
<head>
<title>Saludos</title>
</head>
<body>
<p>Hola,
  <?
      /* Ahora pasamos a modo PHP */
      $nombre = "Lizbeth";
      $apellido = "Leija";
      echo "soy $nombre $apellido";
  ?>
</p>
</body>
</html>
```

Saliendo de HTML

Hay cuatro formas de salir de HTML y entrar en el "modo de código PHP":

Ejemplos. Formas de salir de HTML

1. `<? echo ("esta es la más simple, una instrucción de procesado SGML \n"); ?>`
2. `<? php echo("si quiere usar documentos XML, haga esto \n"); ?>`
3. `<script language="php">`
`echo ("a algunos editores (como FrontPage) no aceptan las instrucciones de`
`procesado");`
`</script>`
4. `<% echo ("Puedes también usar etiquetas tipo ASP"); %>`
`<%= $variable; # Esto es una forma abreviada de "<?echo .." %>`

La primera forma sólo está disponible si se han habilitado las etiquetas cortas. Esto se puede hacer a través de la función `short_tags()`, habilitando la opción de configuración `short_open_tag` en el archivo de configuración de PHP, o compilando PHP con la opción `--enable-short-tags` en `configure`.

La cuarta manera está disponible sólo si se han habilitado las etiquetas tipo ASP usando la opción de configuración `asp_tags`.

La etiqueta de cierre de un bloque incluirá el carácter de nueva línea final si hay uno presente.

Separación de instrucciones

Las instrucciones se separan igual que en C o perl - terminando cada sentencia con un punto y coma.

La etiqueta de cierre (?>) también implica el fin de la sentencia, así lo siguiente es equivalente:

```
<?php
    echo "Esto es una prueba";
?>

<?php echo "Esto es una prueba" ?>
```

Comentarios en PHP

PHP soporta comentarios tipo 'C', 'C++' y shell de Unix. Por ejemplo:

```
<?php
    echo "Esto es una prueba"; // Esto es un comentario tipo c++ para una línea
    /* Esto es un comentario multilínea
    otra línea más de comentario*/
    echo "Esto es otra prueba";
    echo "Una Prueba Final"; # Este es un comentario tipo shell
?>
```

El tipo de comentario de "una línea" sólo comenta, en realidad, hasta el fin de la línea o el bloque actual de código PHP, lo que venga primero.

<h1>Esto es un <?# echo "simple";?> ejemplo.</h1>

<p>La cabecera de arriba dirá 'Esto es un ejemplo'.

Se deberá tener cuidado para no anidar comentarios de tipo 'C', lo cual puede ocurrir cuando se comentan grandes bloques.

```
<?php
    /*
    echo "Esto es una prueba"; /* Este comentario causará un problema */
    */
?>
```

Cuando un cliente solicita esta página, el servidor web la procesa en forma secuencial desde el principio al final buscando secciones PHP limitadas por <? y ?>. En caso de encontrarlas, las compila y ejecuta.

- PHP puede ser agregado rápidamente al código HTML producido por editores HTML interactivos.
- PHP facilita la interacción entre diseñadores y programadores.
- No se necesita re-escribir cada línea de código HTML en un lenguaje de programación.

Inclusión de archivos PHP

Otra forma de agregar código PHP al HTML es poniendo el código PHP en otro archivo e invocarlo mediante la función include. Por ejemplo, un archivo llamado ejemplo.inc contiene solamente el siguiente código:

ejemplo.inc

```
<?
```

```
$novia = "Mafalda";  
print $novia;
```

```
?>
```

Ejemplo.phtml

```
<HTML>  
<HEAD>  
<TITLE>Saluda</TITLE>  
</HEAD>  
<BODY>  
<P>Buen Día  
    <? include("ejemplo.inc"); ?>.  
    Me gusta"<? include("ejemplo.inc"); ?>" .  
</BODY>  
</HTML>
```

1.5 MySQL

MySQL es un sistema manejador de bases de datos.

Una base de datos es una colección de datos estructurado. Puede ser cualquier simple lista de compras o una galería de pinturas o la información de los productos de una empresa de redes, etc. Para agregar, acceder y procesar los datos almacenados en la base de datos, se necesita un sistema manejador de bases de datos como MYSQL.

Entre la base de datos física (es decir, los datos tal y como están almacenados en la realidad) y los usuarios del sistema, existe un nivel de programas, denominado, manejador de bases de datos (MBD) o, en la mayoría de los casos, el sistema administrador de bases de datos relacional RDBMS (Relational Data Base Management System).

Un RDBMS es el conjunto de programas que permiten la definición, manipulación y control de acceso para una o varias bases de datos.

Algunas características del RDBMS MYSQL son:

- Facilitan la integridad, seguridad y acceso de los datos.
- Los datos se almacenan como mínima redundancia.
- Las aplicaciones son independientes del almacenamiento físico de los datos.
- Trabaja en diferentes plataformas de sistema operativo.
- Soporta los lenguajes de programación C, C++, Eiffel, Java, Perl, PHP, Python y Tcl APIs.
- funciones SQL.

MYSQL permite las siguientes condiciones en una base de datos como la mayoría de los RDBMS:

- Los datos han de estar almacenados juntos.
- Tanto los usuarios finales como los programas de aplicación no necesitan conocer los detalles de las estructuras de almacenamiento.
- Los datos son compartidos por diferentes usuarios y programas de aplicación; existe un mecanismo común para la inserción, actualización, borrado y consulta de los datos.

- Los procedimientos de actualización y recuperación, comunes, y bien determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de datos.
- Tanto datos como procedimientos pueden ser transportables conceptualmente a través de diferentes RDBMS.
- Soporta los estándares ANSI.
- Contiene esquemas de almacenamiento independiente que se pueden seleccionar desacuerdo a las necesidades.

Arquitectura cliente/servidor del RDBMS MYSQL

Conceptualmente lo que sucede en un RDBMS cuando un usuario realiza alguna petición, se presenta lo siguiente:

- El usuario solicita alguna petición a la base de datos empleando algún sublenguaje de datos determinado (SQL).
- El RDBMS MYSQL interpreta esa solicitud y la analiza.
- El RDBMS MYSQL inspecciona en orden el esquema externo de ese usuario, la correspondencia externa/conceptual asociada, el esquema conceptual, la correspondencia conceptual/interna y la definición de la estructura de almacenamiento.
- El DBMS ejecuta las operaciones necesarias sobre la base de datos almacenada y devuelve una respuesta al usuario.

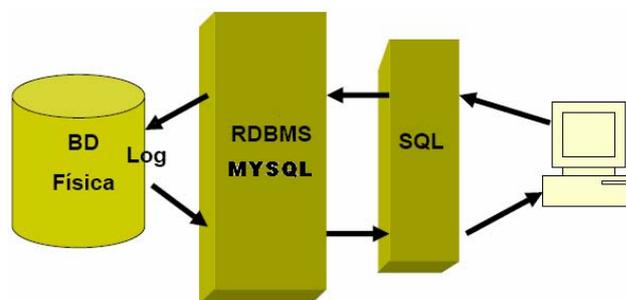


Figura 6. Arquitectura cliente/servidor del RDBMS MYSQL

MYSQL utiliza el Lenguaje estructurado de Consultas SQL, por sus siglas en ingles "Structured Query Language" el cual representa un método estricto y más general de almacenamiento y manejo de los datos.

La estructura básica de una base de datos relacional con SQL es muy simple. Una instalación de base de datos puede contener múltiples bases de datos, cada base de datos puede contener un conjunto de tablas. Cada tabla está compuesta de un conjunto de columnas cuidadosamente diseñadas y cada elemento de la tabla es una fila.

Hay cuatro sentencias de manipulación de datos (DML) soportado por la mayoría de los servidores SQL y que constituyen una gran parte de todas las cosas que se pueden hacer sobre una base de datos.

1. SELECT
2. INSERT
3. UPDATE
4. DELETE

Los cuatro tipos de sentencias permiten la manipulación de datos, pero no de la estructura de la base de datos. En otras palabras, se pueden usar para agregar o modificar la información almacenada en la base de datos, pero no para definir o construir una nueva base de datos. Para modificar la estructura, o agregar tablas y base de datos se usan las sentencias

1. DROP
2. ALTER
3. CREATE

Estas son las características más importantes del Manejador de Bases de Datos Relacional MYSQL.

1.7 SERVIDOR APACHE

El nombre Apache se origina de "A PAtCHy server".

Desde su origen ha evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad.

El servidor Apache es el servicio que se encarga de resolver las peticiones de páginas de Internet de los clientes utilizando originalmente codificación e ideas basadas en el protocolo de Internet http, es uno de los mejores servidores de Web utilizados en la red Internet desde hace mucho tiempo. Es un servidor de Web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP).

Sus principales características son:

- Implementa los últimos protocolos, aunque se base en el HTTP.
- Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo y con la API de programación de módulos.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para solución de los mismos.
- Funciona sobre muchas plataformas (Unix, Linux, Win32, OS2)
- Módulos cargados dinámicamente.
- CGI, Perl (ejemplo: Formularios, diccionarios en línea, etc)
- Php + Bases de datos
- SSL: transacciones seguras
- Soporte para host virtuales
- Alto desempeño

Configuración del servidor apache

Dentro de la configuración del Apache las tareas que realizan son:

- habilitar a los usuarios que puedan habilitar sus páginas web dentro de su propia cuenta en internet.

- configuración de maquinas virtuales.
- configuración del servidor seguro apache http.

El servidor Apache es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- **Módulos Base:** Módulo con las funciones básicas del Apache
- **Módulos Multiproceso:** son los responsables de la unión con los puertos de la máquina, acepando las peticiones y enviando a los hijos a atender a las peticiones
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

Estas son las características más importantes del servidor Apache.

CAPÍTULO II.

PROYECTO DE UN SISTEMA DE CONSULTA DE INFORMACIÓN DE CURSOS, EN LÍNEA
BASADO EN SOFTWARE LIBRE Y TECNOLOGIA ORACLE.

2.1 Descripción

En el Centro de Cómputo de la Dirección General de Servicios de Computo Académico sede palacio de la Autonomía se imparten alrededor de 60 cursos y las fechas de inicio de los cursos son programadas mensualmente, estos cursos son ofrecidos para todo el público.

En este centro, se requiere almacenar de forma organizada la información de las personas que desean tomar alguno de estos cursos.

Las personas interesadas deben llenar una hoja con un formato en particular, para que la institución cuente con sus datos y se les pueda dar aviso cuando el curso de su interés se vaya abrir

Procedimiento del registro de personas interesadas en tomar cursos

Los cursos cuentan con una fecha de inicio programada pero no siempre se abre en esta fecha ya que se requiere de un mínimo de 3 personas interesadas en el mismo curso para que sea abierto; las personas interesadas acuden a la sede para solicitar información y dependiendo si hay una fecha de inicio próxima confirmada, realizan su inscripción al curso, pero sino se les pide llenar el formato de registro, para que cuando se cuente con una fecha segura de inicio del curso se le informe vía telefónica de acuerdo a la información con que se cuenta en estas hojas de registro.

Forma Actual de Trabajo

Actualmente la información de las personas interesadas en tomar cursos se encuentra registrada en hojas con el formato siguiente.

Fecha: _____ Hora: _____
Solicitó la información: Via telefónica Visita personal
Nombre: _____
Colonia: _____ Delegación o Municipio: _____
Tel.Particular: _____ Tel.Oficina: _____ Celular: _____
Correo Electrónico: _____
 Estudia ¿Dónde? _____ Col. _____
 Trabaja ¿Dónde? _____ Col. _____
Cursos de interés: Horario: Sabatino L-V _____
1. _____
2. _____
3. _____
Medio por el cual se enteró de los cursos: _____
Observaciones: _____

Figura 7. Formato de registro

Esta información es necesaria para poder llamar por teléfono e informarles que el curso de su interés va a ser impartido.

Desventajas de la forma actual de trabajo.

- Al personal encargado de esta tarea, le resulta complicada la búsqueda de todas las personas que se encuentran registradas, ya que la cantidad de información es muy amplia, y tienen que realizar la búsqueda de registro en registro verificando el nombre del curso y horario en común de cada uno.
- Existe redundancia en la información.

Requerimientos del sistema.

Con el fin de agilizar estos movimientos se ha propuesto que desarrolle un sistema que por medio de una base de datos se almacene toda la información de las personas interesadas de acuerdo al formato de registro con que cuenta la institución y a través de una aplicación Web se consulten los datos de todas las personas, seleccionando el nombre de un curso y horario.

El sistema debe:

- Automatizar el proceso de búsqueda.
- Listar todos los registros de un curso en común.
- Permitir altas, bajas y actualizaciones de los registros.
- Mantener segura y accesible la información.

2.2 Diseño del Sistema de consulta de información de cursos.

Descripción del diseño para este sistema

Para que se logren los requerimientos anteriores, debe existir una base de datos que esta basada en el modelo relacional ya que este modelo permite a los usuarios obtener información de manera más eficiente, además brinda mayor seguridad e integridad en la información.

Además necesitamos una interfaz gráfica de usuario que interactúe con la base de datos mediante la cual se podrá almacenar y consultar la información, y realizar las tareas requeridas por la institución.

Este sistema será desarrollado sobre una plataforma linux y una base de datos Oracle para un mejor servicio y seguridad.

2.2.1 Diseño de la base de datos.

En el diseño de la base de datos se debe realizar un modelo de datos que ayude a entender el significado de los datos y que facilite la comunicación en cuanto a los requisitos de información.

La primera etapa es el diseño conceptual, en donde se construye un esquema de la información de acuerdo al formato de registro, independientemente de todas las consideraciones físicas. Después viene el diseño lógico, en el que el esquema anterior se transforma según el modelo de base de datos que se vaya a utilizar para implementar el sistema. Por último, en la etapa del diseño físico, se produce una descripción de la implementación de la base de datos en memoria secundaria.

El modelo relacional, es el modelo lógico en el que se basan la mayoría de los SGBD comerciales en uso hoy en día, nuestro caso Oracle por sus características, como todo modelo de datos, tiene que ver con tres aspectos de los datos:

- Estructura de datos.
- Integridad de datos.
- Manejo de datos.

Es por estas razones que se optó este modelo de datos para el sistema.

Modelo lógico del sistema.

Diseño lógico de la base de datos.

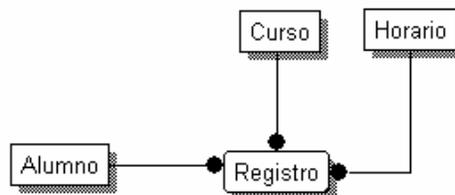


Figura 8. Diseño lógico de la base de datos.

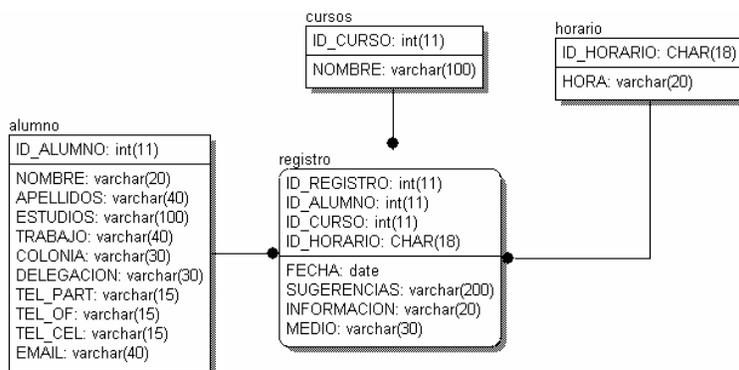


Figura 9. Diseño físico de la base de datos.

Tabla alumno

Esta tabla contiene todos los datos que corresponden a las personas interesadas en los cursos; como su nombre, apellidos estudios, trabajo, colonia, delegación, teléfono particular, teléfono de oficina, teléfono celular y por ultimo su e-mail si cuenta con uno. No toda la información es indispensable, pero se requiere su nombre completo y teléfono para poder localizarlo en caso de apertura de un curso.

Tabla cursos

Esta tabla contiene el catalogo de cursos con que cuenta la institución.

Tabla registro

Esta tabla contiene la información que falta de acuerdo al formato de registro, es decir fecha de registro, el medio por el que se entero de los cursos, el horario de su preferencia, las sugerencias y la forma en que lleno este formato puede ser de manera personal o vía telefónica

Tabla horario

Esta tabla contiene los diferentes horarios con que cuenta la institución para impartir los cursos.

El diseño de las aplicaciones, una fase que se debe llevar a cabo en paralelo con el diseño de la base de datos, está compuesta por dos actividades: el diseño de las transacciones y el diseño de las interfaces de usuario de informes y formularios.

2.2.2 Módulos del sistema

El sistema de consulta de información de cursos en línea, tiene la finalidad de almacenar la información proporcionada por las personas interesadas en cursos. Fue diseñada de tal forma que para el personal encargado de esta tarea le resulte más fácil realizar búsquedas eficientes de la información.

El sistema inicia con la página de registro y se enlaza con páginas con un una tarea en específico.

Pantalla de Registro

La primera pantalla es para almacenar en la base de datos, la información del interesado como son: sus datos personales, los cursos de su interés y otras referencias útiles para la institución.

La información que se introduce en esta pantalla será almacenada en la base de datos mediante programación con java y sentencias SQL.

solicitó la información
 personal vía telefónica

nombre apellidos
colonia delegación

estudia ¿dónde?
trabaja ¿dónde?

cursos de interés:
curso 1: Seleccione el curso
curso 2: Seleccione el curso
curso 3: Seleccione el curso
horario: Seleccione el Horario

teléfono
particular:
trabajo:
celular:
e-mail:

medio por el cual se enteró:
observaciones:

UNAM

Inicio Registrar consulta actualiza datos consultar

Figura 10. Pantalla de Registro

La pantalla de registro, como ya se mencionó cuenta con 3 ligas que están direccionadas hacia las páginas de:

- 1) Consulta. 
- 2) Actualiza Datos. 
- 3) Catálogo. 

1. Consulta

En la pantalla de Consulta se podrá ver la información de todas las personas registradas.

Para poder realizar una consulta en la base de datos debe seleccionar un curso y un horario para que cuando de clic en el botón "Ver Lista" el sistema genere la lista de la gente registrada de acuerdo a la información seleccionada.



Figura 11. Pantalla de Consulta



Figura 12. Lista de personas registradas

2. Actualiza Datos.

Esta ventana cuenta con la lista de todas las personas que se encuentran registradas en la base de datos y tiene la finalidad de actualizar toda la información o parte de ella de acuerdo al nombre que seleccionen de esta lista, también cuenta con la opción de Borrar el registro por completo.

Podrá ver la información de cada persona, seleccionando un nombre de la lista desplegable "Nombre" y después dar clic sobre el botón "Seleccionar" para que muestre la información del registro en cada una de las cajas con que cuenta este módulo, estos datos son los que fueron proporcionados por el usuario cuando se registró.

Selecciona Registro

Nombre: Se eccione el Registro [v] [Seleccionar]

Para eliminar el registro presione "Borrar"

[BorrarRegistro]

nombre: Hanz Maec apellidos: Ziñga Guico

colonia: Conto delegación: Cuautémoc

estudia ¿dónde?: UVI

trabaja ¿dónde?:

teléfono particular: []

trabajo: []

celular: 044516959223

e-mail: []

cursos de interés:

curso 1: Reg sto Contable Con C0L [v]

curso 2: Sistema de Administración Empresarial SAE [v]

curso 3: [v]

horario: Indiferente [v]

medio por el cual se enteró: gaceta

[Actualizar] [consulta] [actualiza datos]

Figura13. Actualiza Datos

Si desea borrar el registro definitivamente, de la base de datos, tendrá que pulsar sobre el botón "BorrarRegistro".

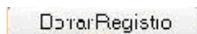


Figura 14. Botón Borrar registro

Cuando presiona el botón “Borrar Registro” lanza la siguiente ventana para confirmar que quiere borrar la información.



Figura 15. Confirmación Borrar

Si lo que se desea es actualizar algún dato, deberá ingresar la nueva información y dar clic en el botón Actualizar.

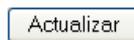


Figura 16. Actualizar

3. Catálogo de cursos.

Esta pantalla tiene las funciones de:

- Agregar Curso
- Borrar Curso
- Actualizar Curso

Podemos elegir cualquiera de las tres opciones que nos da la pantalla seleccionando el círculo y después pulsar sobre el botón de seleccionar.



Figura 17. Catálogo de Cursos.

La opción Agregar Curso nos permite insertar un nuevo registro a la lista de cursos con que se cuenta en el sistema.

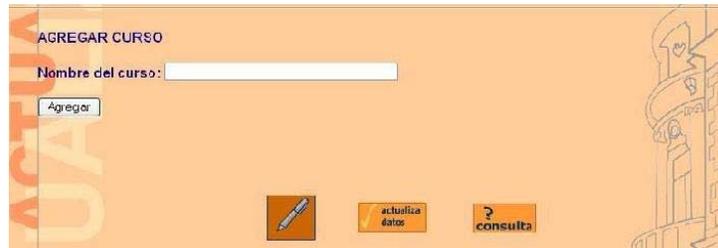


Figura 18. Agregar un nuevo curso al catalogo.

La opción de borrar curso muestra la lista de todos los cursos, de la cual tenemos que seleccionar el que se desea borrar, después de presionar el botón de borrar, envía un mensaje para confirmar esta operación.



Figura 19. Borrar un Curso del catalogo.

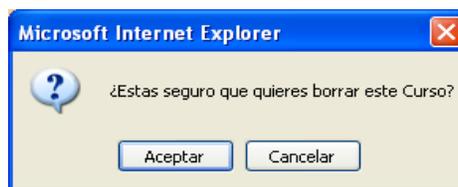


Figura 20. Confirmación Borrar Curso.

La opción "Actualizar Curso" también muestra la lista de todos los cursos disponibles en el sistema, una vez seleccionada esta opción debemos elegir el registro que se quiere modificar, ya realizados los cambios pulsamos sobre el botón de Actualizar para que la modificación se realice.



Figura 21. Actualizar nombre del Curso.

2.3 Lenguaje de programación

El lenguaje de programación que se utilizara para este proyecto es Java ya que es un lenguaje con el que podemos realizar cualquier tipo de programa y cada vez cobra más importancia tanto en el ámbito de Internet como en la computación en general.

Una de las principales razones por que el uso de Java, es por que es un lenguaje independiente de la plataforma. Es decir que si hacemos un programa en Java podrá funcionar en cualquier computadora con sistema operativo Windows, Linux, macintosh, etc. Esto lo consigue porque se ha creado una Máquina Virtual de Java para cada sistema que sirve de puente entre el sistema operativo y el programa de Java.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso desde distintos lugares hacia la aplicación.

Además Java es compatible con otros lenguajes como HTML, Java Script y SQL, esto permite mayor facilidad de conexión con distintas bases de datos como por **Ejemplo** Oracle e Interactividad con aplicaciones web.

2.4 Sistema Manejador de Bases de Datos

Para este sistema de registro se utilizará tecnología Oracle ya que actualmente se encuentra en muchas industrias, por sus características de compatibilidad, portabilidad, conectividad y capacidad.

Oracle proveen grandes beneficios a sus usuarios, protegiendo inversiones presentes y futuras en tecnología informática.

Oracle se considera como uno de los sistemas de bases de datos más completos, rápidos, integrable y disponible en todas las plataformas (Unix, Linux y Windows) destacando su:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Multiplataforma.

Esta base de datos ha sido desarrollada desde hace más de 20 años desde las bases de datos relacionales hasta la era del Internet. Ha sido diseñada para proveer la solución más completa y de menor costo para cualquier requerimiento de información de negocio y actualmente es la única herramienta capaz de:

- Tener información crítica disponible en cualquier momento.
- Dar protección y seguridad a información privilegiada.
- Integrar información de diversas fuentes, incluyendo bases de datos externas y archivos de sistemas.
- Consolidar y administrar todo el contenido de Internet.
- Reducir el tiempo que le toma a un negocio tomar mejores decisiones analizando la información más rápido.
- Permitir a la organización desarrollar e implementar “business solutions” rápidamente.

Aunque es una base de datos comercial, se tomo la decisión de utilizar este manejador por estas características y por su amplio uso en los sistemas de información. Además que cuenta con herramientas propias para administrar y monitorear las bases de datos como lo es la herramienta Enterprise Manager.

2.5 Sistema Operativo

El sistema operativo que se utilizará para esta aplicación es una distribución Red Hat Linux ya que brinda al usuario y desarrollador de software, total control y legalidad sobre el software que ejecuta, es seguro y además es compatible con el manejador de base de datos Oracle.

2.6 Servidor web

Para este sistema se utilizará el servidor web Tomcat que es un contenedor de servlets, es decir un shell basado en plataforma Java, que nos permite ejecutar aplicaciones web Java Server Pages (JSP) y servlets.

Incluye numerosas particularidades que hacen que sea una plataforma de predilecto para el desarrollo de aplicaciones y servicios web.

Tomcat está integrado al proyecto Apache-Jakarta, por lo que se trata de la solución idónea para implementar especificaciones Apache, mejor que con cualquier otro servidor de aplicaciones web disponibles en el mercado.

Conclusiones

La Universidad Nacional Autónoma de México, a través de la Dirección General de Servicios de Cómputo Académico, imparte el Diplomado de Desarrollo e Implementación de Sistemas con Software Libre en Linux, con el objetivo de formar profesionales que cuenten con los elementos necesarios para realizar una adecuada y productiva administración de Sistemas.

El principal objetivo de este trabajo, ha sido presentar una alternativa de administración mediante el uso de herramientas de programación y en combinación con otro tipo de tecnologías lograr un acoplamiento que de cómo resultado un sistema funcional y eficiente para cualquier organización. Además una alternativa de desarrollo de aplicaciones mediante el concepto de "Software Libre".

El software libre constituye un fenómeno único en la historia de la tecnología ya que es posible producir software de calidad, es posible encarar proyectos de alta complejidad por que ofrece cualidades de funcionalidad calidad y seguridad.

Las empresas gastan mucho dinero en tecnologías de información sin obtener los resultados esperados, esto hace que cada una cuente con un área de desarrollo en la que su recurso principal es el uso de esta tecnología, por que les permite:

1. La libertad de usar el programa con cualquier propósito.
2. La libertad de estudiar como funciona su aplicación y adaptarla a sus necesidades.
3. La libertad de distribuir copias sin necesidad de pagar licencias de uso.
4. La libertad de mejorar la aplicación de modo que toda la empresa se beneficie.
5. El software libre permite una casi total libertad de elección de plataforma y hardware, pudiendo llegar también al automantenimiento.

El diplomado de Desarrollo e Implementación de Sistemas con Software Libre en Linux es una buena opción para introducirse al mundo de la administración de Sistemas, por que se exponen diferentes temas como: programación, servidores web, seguridad, bases de datos, esto otorga una amplia visión en las áreas de sistemas en general, ya que puedes aportar opiniones, ayuda o soluciones a problemas que no competen solo a tu área de trabajo. Este diplomado abre las puertas al manejo de entornos Unix y Linux, por lo que, por mi parte lo puedo recomendar a personas que inician a manejar este tipo de tecnología.

Bibliografía.

David Bandel. "Linux Edición Especial"
México, Sexta Edición, Prentice Hall

Steve Maxwell, "UNIX System Administration – A Beginner's Guide"
New York, McGraw-Hill/Osborne, 2000

J. Purcell, "Linux Complete Command Reference"
Indianapolis, RedHat Press, 2002

RedHat, "Manual de Referencia RadHat Linux"
USA, RedHat Inc., 2003

Víctor Gracia, "Curso Completo de HTML"
México, Quita Edición, Alfaomega

Rasmus Lerdorf, "Programación PHP"
O'REILLY, 2002

MySQL AB, "MySQL Referente Manual"
MySQL AB, 2001

Referencias Web

Introducción a Linux.

<http://www.linuxparatodos.net/linux/menu-introduccion.php>

Conceptos Básicos

<http://www.todo-linux.com/tl.php?file=manual.todo-linux.com/linuxgeneral.htm>

Kernel. "El núcleo del sistema"

<http://es.wikipedia.org/wiki/Kernel>

Linux. ¿Que es Linux?

<http://www.grulic.org.ar/linux.html>

Linux. Distribuciones.

<http://www.linux-es.org/distribuciones.php>