



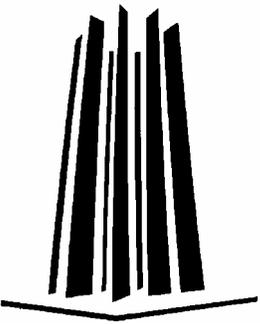
UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
CAMPUS ARAGÓN

**“INFORME DE UN PROYECTO DE SISTEMA
PARA FACTURACIÓN A TRAVÉS DE LA WEB,
UTILIZANDO SOFTWARE LIBRE”.**

**EN LA MODALIDAD DE SEMINARIOS
Y CURSOS DE ACTUALIZACIÓN Y
CAPACITACIÓN PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
I V A N G R A D A H U E R T A**

ASESOR: ING. SILVIA VEGA MUYTOY.



MÉXICO

2006



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

AGRADECIMIENTOS

Quiero agradecer a todas las personas que me han apoyado en al vida desde pequeño hasta la fecha.

En mi caso agradeceré a toda mi familia, a mis hermanos por apoyarme en las cosas que a veces no podía terminar; por brindarme tiempo y paciencia ante cualquier presión que tenía.

A mis amigos por la gran ayuda y por la compañía que tuvimos durante estos años, por las alegrías y por la convivencia que siempre nos caracterizó y que seguimos teniendo hasta ahora, por las personas que no ya no están por aquí, pero que algún día quisieron verme en esta cúspide.

A las personas que he me conocieron en este trayecto y que desde ese momento me apoyaron totalmente.

Por las personas como Victoria Juárez que apoyaron al término de este trabajo, por que sin ellos, esto no hubiese sido realidad.

Pero el agradecimiento más grande es para mis padres quienes se esforzaron por sacarme adelante con todo y los problemas que de niño se me presentaron, por sus sacrificios y por su amor hacia mí. Por mi padre que de una forma u otra me saco adelante, por educarme con gran sabiduría y por darme buenos consejos. A mi madre que fue siempre mi consejera, mi guía; por ayudarme a conseguir lo que más se dificultaba desde hacer un simple 5 hasta la forma de educarme, siempre con valores y con metas a seguir.

Es a ustedes, que son un ejemplo a seguir, a quienes les doy las gracias por todo, por que lo que ahora soy es por ustedes.

ÍNDICE

CAPITULO 1.

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

INTRODUCCIÓN	Pag.
1.1. Sistema operativo Linux.....	1
1.2. Administración de Linux.....	5
1.3. Editores para la creación de páginas Web.....	20
1.3.1. Software empleado para el desarrollo de la página.....	20
1.3.2. Práctica Web.....	31
1.4. Administración de servidores WWW con Linux.....	34
1.5. Programación con PHP.....	42
1.5.1. Práctica PHP.....	44
1.6. Interacción de WWW con Bases de Datos.;.....	47
1.6.1. Práctica Base de datos.....	52
1.7. Introducción a la seguridad en cómputo.....	55
1.8. Programación con Java.....	56
1.8.1. Práctica Java.....	66

CAPITULO 2.

PROYECTO DE SISTEMA DE FACTURACIÓN A TRAVÉS DE LA WEB CON HERRAMIENTAS LIBRES.....	68
2.1 <u>Usuario administrativo</u>	69
2.1.1 MENÚ INICIO.....	70
2.1.2 MENÚ FACTURA.....	72
2.1.3 MENÚ BUSCAR.....	75
2.1.4 MENÚ REPORTE.....	76
2.1.5 MENÚ DESCARGA XML.....	77
2.1.6 MENÚ ADMINISTRACIÓN.....	78
2.2 <u>Usuario facturador</u>	79
2.2.1 MENÚ SUBIR FACTURA.....	80
2.3 <u>Usuario consultor</u>	80
2.4 Comentarios.....	81
2.5 Bibliografía.....	82



INTRODUCCIÓN

Hoy en día es necesario tener sistemas que permitan la automatización del seguimiento y control de procesos que se llevan a cabo en diversas empresas y oficinas. Con esta automatización se tiene un uso más eficiente de los recursos y un control más preciso de los mismos, lográndose con ello, compartir información entre diversas áreas y oficinas, ya sea a través de la red local e incluso empleando Internet o la Intranet corporativa.

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

OBJETIVO GENERAL

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

OBJETIVO ESPECIFICO

Se instalará el sistema ya sea en linux o Windows de tipo PCS y se configurará en el disco duro.

DIRIGIDO A

Público en general y profesionistas interesados en desarrollar aplicaciones personalizadas de bases de datos de red.



CAPITULO 1

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

1.1 Sistema operativo LINUX

Objetivo

Se conocerá un sistema operativo que interactúe con cualquier software sin problemas, que sea libre y que contenga un gran número de librerías para hacer posible su utilización y pueda competir contra los mejores en el mundo.

Introducción

Algo de la historia de Linux

Historia del Linux: Linux fue creado originalmente por Linus Torvald en la Universidad de Helsinki en Finlandia, siendo él estudiante de informática. Pero ha continuado su desarrollo con la ayuda de muchos otros programadores a través de Internet.

Linux originalmente inicio el desarrollo del núcleo como su proyecto favorito, inspirado por su interés en Minix, un pequeño sistema Unix desarrollado por Andy Tannenbaum.

El 5 de octubre de 1991, Linux anuncio su primera versión "oficial" de Linux, versión 0.02. Desde entonces, muchos programadores han respondido a su llamada, y han ayudado a construir Linux como el sistema operativo completamente funcional que es hoy.

¿Qué es Linux?

Linux es un sistema operativo diseñado por cientos de programadores de todo el planeta, aunque el principal responsable del proyecto es Linus Torvalds. Su objetivo inicial es propulsar el software de libre distribución junto con su código fuente para que pueda ser modificado por cualquier persona, dando rienda suelta a la creatividad. El hecho de que el sistema operativo incluya su propio código fuente expande enormemente las posibilidades de este sistema. Este método también es aplicado en numerosas ocasiones a los programas que corren en el sistema, lo que hace que podamos encontrar muchísimos programas útiles totalmente gratuitos y con su código fuente. Y la cuestión es que, señores y señoras, *Linux es un sistema operativo totalmente gratuito.*

Dos características muy peculiares lo diferencian del resto de los sistemas que podemos encontrar en el mercado: A) Es libre, esto significa que no tenemos que pagar ningún tipo de licencia a ninguna casa desarrolladora de software por el uso del mismo. B) El sistema viene



acompañado del código fuente el cual nos permite modificarlo. El sistema lo forman el núcleo del sistema (kernel) mas un gran numero de programas / librerías que hacen posible su utilización.

LINUX se distribuye bajo una licencia GNU (Public License) por lo tanto, el código fuente tiene que estar siempre accesible.

Día a día, más y más programas / aplicaciones están disponibles para este sistema, y la calidad de los mismos aumenta de versión a versión. La gran mayoría de los mismos vienen acompañados del código fuente y se distribuyen gratuitamente bajo los términos de licencia GNU.

Las plataformas en las que en un principio se puede utilizar Linux son: 386-, 486-. Pentium, Pentium Pro, Pentium II, Amiga y Atari, también existen versiones para su utilización en otras plataformas, como Alpha, ARM, MIPS, PowerPC y SPARC.

Puedo asegurar que es uno de los sistemas que más aprovecha una computadora, es decir, con el se consigue ejecutar tareas mucho más rápido que con otros sistemas operativos comerciales. Y es que Linux no requiere grandes prestaciones para funcionar.

Linux no sacrifica en ningún momento la creatividad, tal y como lo hacen algunas compañías informáticas. Linux es una ventana abierta por la que es posible huir hacia un mundo donde la verdadera informática puede ser disfrutada sin límites ni monopolios.

Linux es distribuido mediante una serie de distribuciones como RedHat, Slackware, Debían ... las cuales se diferencian por su método de instalación y por los paquetes (software) que viene incluido. Es posible que encuentre a la venta versiones de Linux, todo el software de Linux esta regido por la licencia de GNU, con la cual que cualquier persona puede modificar un programa y venderlo según el desee, con la condición que la persona que compra ese producto puede realizar la misma acción o simplemente hacer copias para todos aquellos que lo quieran sin tener que pagar más (por lo tanto no se extraña si encuentra distribución comerciales). Esta licencia es la garantía que afirma la absoluta libertad de este sistema operativo.

Características de Linux

Las funciones principales de este magnífico sistema operativo son:

- **Sistema multitarea** En Linux es posible ejecutar varios programas a la vez sin necesidad de tener que parar la ejecución de cada aplicación.
- **Sistema multiusuario** Varios usuarios pueden acceder a las aplicaciones y recursos del sistema Linux al mismo tiempo. Y, por supuesto, cada uno de ellos puede ejecutar varios programas a la vez (multitarea).
- **Shells programables** Un shell conecta las ordenes de un usuario con el Kernel de Linux (el núcleo del sistema), y al ser programables se puede modificar para adaptarlo a tus necesidades. Por ejemplo, es muy útil para realizar procesos en segundo plano.
- **Independencia de dispositivos** Linux admite cualquier tipo de dispositivo (módems, impresoras) gracias a que cada una vez instalado uno nuevo, se añade al Kernel el enlace o controlador necesario con el dispositivo, haciendo que el Kernel y el enlace se fusionen. Linux posee una gran adaptabilidad y no se encuentra limitado como otros sistemas operativos.



- **Comunicaciones** Linux es el sistema más flexible para poder conectarse a cualquier ordenador del mundo. Internet se creó y desarrollo dentro del mundo de Unix, y por lo tanto Linux tiene las mayores capacidades para navegar, ya que Unix y Linux son sistemas prácticamente idénticos. Con linux podrá montar un servidor en su propia casa sin tener que pagar las enormes cantidades de dinero que piden otros sistemas.
- **Multitarea:** varios programas (realmente procesos) ejecutándose al mismo tiempo.
- **multiusuario:** varios usuarios en la misma máquina al mismo tiempo (y sin licencias para todos).
- **multiplataforma:** corre en muchas CPUs distintas, no sólo Intel.

Entre otras características están:

- Funciona en modo protegido 386.
- Tiene protección de la memoria entre procesos, de manera que uno de ellos no pueda colgar el sistema.
- Carga de ejecutables por demanda: Linux sólo lee de disco aquellas partes de un programa que están siendo usadas actualmente.
- Política de copia en escritura para la comparación de páginas entre ejecutables: esto significa que varios procesos pueden usar la misma zona de memoria para ejecutarse. Cuando alguno intenta escribir en esa memoria, la página (4Kb de memoria) se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.
- Memoria virtual usando paginación (sin intercambio de procesos completos) a disco: una partición o un archivo en el sistema de archivos, o ambos, con la posibilidad de añadir más áreas de intercambio sobre la marcha (se sigue denominando intercambio, es en realidad un intercambio de páginas). Un total de 16 zonas de intercambio de 128Mb de tamaño máximo pueden ser usadas en un momento dado con un límite teórico de 2Gb para intercambio.
- La memoria se gestiona como un recurso unificado para los programas de usuario y para el caché de disco, de tal forma que toda la memoria libre puede ser usada para caché y éste puede a su vez ser reducido cuando se ejecuten grandes programas.
- Librerías compartidas de carga dinámica (DLL's) y librerías estáticas también, por supuesto.
- Casi totalmente compatible con POSIX, System V y BSD a nivel fuente.
- Todo el código fuente está disponible, incluyendo el núcleo completo y todos los drivers, las herramientas de desarrollo y todos los programas de usuario; además todo ello se puede distribuir libremente. Hay algunos programas comerciales que están siendo ofrecidos para Linux actualmente sin código fuente, pero todo lo que ha sido gratuito sigue siendo gratuito.
- Emulación de 387 en el núcleo, de tal forma que los programas no tengan que hacer su propia emulación matemática. Cualquier máquina que ejecute Linux parecerá dotada de coprocesador matemático. Por supuesto, si tu ordenador ya tiene una FPU (unidad de coma flotante), será usada en lugar de la emulación, pudiendo incluso compilar tu propio kernel sin la emulación [matemática](#) y conseguir un pequeño ahorro de memoria.
- Soporte para muchos teclados nacionales o adaptados y es bastante fácil añadir nuevos dinámicamente.
- Consolas virtuales múltiples: varias sesiones de login a través de la consola entre las que se puede cambiar con las combinaciones adecuadas de teclas (totalmente independiente del hardware de video). Se crean dinámicamente y puedes tener hasta 64.



- Soporte para varios sistemas de archivo comunes, incluyendo minix-1, Xenix y todos los sistemas de archivo típicos de System V, y tiene un avanzado sistema de archivos propio con una capacidad de hasta 4 Tb y nombres de archivos de hasta 255 caracteres de longitud.
- Acceso transparente a particiones MS-DOS (o a particiones OS/2 FAT) mediante un sistema de archivos especial: no necesitas ningún comando especial para usar la partición MS-DOS, parece un sistema de archivos normal de Unix (excepto por algunas graciosas restricciones en los nombres de archivo, permisos, y esas cosas). Las particiones comprimidas de MS-DOS 6 no son accesibles en este momento, y no se espera que lo sean en el futuro. El soporte para VFAT (WNT, Windows 95) ha sido añadido al núcleo de desarrollo y estará en la próxima versión estable.
- Un sistema de archivos especial llamado UMSDOS que permite que Linux sea instalado en un sistema de archivos DOS.
- Sistema de archivos de CD-ROM que lee todos los formatos estándar de CD-ROM.
- TCP/IP, incluyendo ftp, telnet, NFS, etc.
- Appletalk disponible en el actual núcleo de desarrollo.
- Software cliente y servidor Netware disponible en los núcleos de desarrollo.



1.2 Administración de Linux

Objetivo.

Se llegara a brindar un funcionamiento confiable para cualquier sistema que se trabaje, para esto se conocerá los proceso de configuración, mantenimiento y actualización; así como el concepto de funcionamiento de software, hardware y periféricos con la finalidad de permitir o negar acceso.

Perfil y Actividades del Administrador.

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.

Importancia de la Administración

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

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

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.
- Auditoria e implantación de la seguridad del sistema.



INSTALACIÓN DE LINUX

Preparación de la instalación

- Slackware Linux no requiere de un sistema extremadamente potente para ejecutarse.
- Linux Slackware se puede ejecutar en equipos 386 o superiores.
- Los requerimientos mínimos para instalar y ejecutar Slackware son:

Requerimientos de instalación

Hardware Requiere

Procesador 386

RAM 16 MB – 64M

Espacio en Disco 500MB – 3G

Floppy Drive 1.44 MB

Unidad CD-ROM

Software (pagina oficial)

Nota: Para maquinas recientes consultar la lista de compatibilidad de hardware (HCL)

Antes de instalar Slackware debemos conocer las series de paquetes que se van a instalar. Linux Slackware es una de las distribuciones de Linux mas antiguas, contiene un conjunto de paquetes que se reparten en series ordenadas alfabéticamente

Series:

A - es la base contiene suficiente el kernel y utilidades básicas del sistema

AP - varias aplicaciones que no requieren del sistema X Windows.

D - Herramientas de desarrollo de programas, Compiladores, depuradores, intérpretes como C, Perl, Lisp, etc

E - El editor GNU Emacs, es tan grande que requiere su propia serie.

F - Contiene FAQs, HOWTOs, y documentación esencial para la administración.

GNOME - Contiene el ambiente de escritorio de GNOME y librerías relacionadas.

K - El código de fuente para el núcleo de Linux.

KDE - Contiene el ambiente de trabajo de escritorio KDE y QT

KDEI - Contiene el conjunto de idiomas internacionales para KDE

L - Librerías del sistema necesarias para los escritorios GNOME, KDE y más.

N - Contiene programas relacionados con la red. Demonios, clientes, etc.

T - Software de tipo TeX usado generalmente para documentos técnicos.

TCL - Contiene las herramientas del lenguaje de comandos, el Tk, el TclX, y el TkDesk.

X - Contiene la base del sistema X Window.

XAP - Contiene una colección de aplicaciones para X Por ejemplo Ghostscript y Netscape.

Y - Contiene juegos (una colección de juegos de BSD).



Creación de los discos de instalación

Discos de arranque.

En caso de que nuestro equipo sea demasiado viejo como para poder realizar una instalación desde el CD-ROM, necesitaremos crear los discos de arranque para la instalación.

Los discos de boot y de root.

Para crear los discos de arranque se siguen los mismos pasos, pero antes debemos seleccionar una imagen para cada uno de estos discos, las vamos a encontrar dentro de los directorios bootdsk.144 y rootdsk respectivamente.

Imágenes

Disco de arranque boot.

Dentro del directorio bootdsk.144 encontraremos un sin fin de imágenes para diferentes dispositivos, nosotros solo necesitaremos la imagen de bare.i.

Esta imagen contiene los controladores estándar para diferentes dispositivos (CD-ROM y discos duro), es la imagen estándar y funciona para la mayoría de las instalaciones.

Disco de arranque root.

Existen diferentes tipos de imágenes para el disco de root.

color.gz - esta imagen contiene un menú de la instalación de Slackware a color.

text.gz - algunas personas han detectados errores en la instalación utilizando color.gz, en caso de que esta imagen no funcione se puede utilizar text.gz esta imagen es la que se emplea por default.

umsdos - una versión de la imagen color.gz partición de ms-dos.

network.dsk - esta imagen es complementaria permite cargar módulos para dispositivos de red que lo necesiten. pcmcia.dsk - esta imagen contiene drivers que ayudan a configurar dispositivos PCMCIA, que vienen incluidos en computadoras portátiles.

Creación de los discos de arranque.

Las imágenes contenidas dentro de los directorios bootdsk.144 y rootdsk son archivos del tamaño exacto de un disquete de 3 1/2, estos archivos contienen información del sistema de archivos de Linux y de los datos contenidos en ellos, simulara tener instalado Linux y podremos utilizar una pequeña cantidad de comandos básicos.

La imagen no podrá ser utilizada hasta que no haya sido escrita en un disco flexible.

Para comenzar, necesitaremos un disco en blanco formateado de alta densidad (1.44MB) de 3.5 pulgadas, una computadora que nos permita poder trabajar en un ambiente en modo texto como MS-DOS.

Necesitamos una sesión MS-DOS para poder utilizar la herramienta rawrite, que viene incluida en el CD-ROM de nuestra distribución Linux, en el directorio dosutils.



Primero, etiquetemos nuestro disco (en blanco y formateado) como "Disco de Arranque" o "Disco de Instalación". Inserte el disco en la unidad de 3 1/2 y Luego, procedemos a utilizar los siguientes comandos (asumiendo que la unidad de CDROM es d:, si no es así sustituya la unidad d: por la de nuestro equipo).

```
C:\WINDOWS>D:
D:\>cd bootdsk.144
D:\BOOTDSKS>rawrite
Enter disk image source file name: ..bare.i Enter target diskette drive: a:
Please insert a formatted diskette into drive A: and
press -ENTER-- : Enter
D:\DOSUTILS>
```

- rawrite pide el nombre del archivo de una imagen de disco; teclee el directorio y el nombre de la imagen que quiere escribir (bare.i).

- Entonces, rawrite le pedirá que introduzca el nombre de la unidad en donde se escribirá la imagen; teclee a.

- Rawrite pedirá la confirmación de que existe un disco en la unidad que se ha elegido.

Por ultimo tecleamos enter y rawrite copiará la imagen en el disco flexible. Si es necesario hacer otro disco, volvemos a repetir los pasos ya mencionados con el nombre de la nueva imagen.

Una vez creado los discos de boot y root podemos comenzar a instalar Linux Slackware, solamente reiniciamos el equipo e insertamos el disco de boot y cuando el programa de instalación lo requiera insertamos el disco de root.

- En nuestro caso no será necesario crear los discos de arranque, ya que podemos iniciar la instalación directamente desde el CD-ROM de la distribución.

- Siempre y cuando contemos con esta opción en la BIOS del sistema.

Tipos de instalación:

CD-ROM

Si nuestro equipo cuenta con la opción de arranque por medio de un CD-ROM, esta sería la forma más fácil de instalar Linux Slackware.

Disco de red

Esta opción se emplea cuando se necesita hacer la instalación de varios equipos, se emplea la red para este propósito y lo podemos hacer mediante NFS, FTP y HTTP.

Disco PCMCIA

Este método se emplea cuando se desea instalar Linux en una computadora portátil, este tipo de equipos cuentan con dispositivos PCMCIA.



Una vez elegido el método de instalación procedemos a instalar Linux, arrancamos la instalación (vía CD ROM o disco de arranque), Aparecerá una pantalla de bienvenida al programa de instalación de Linux, en la parte inferior de esta pantalla aparecerá el indicador boot:

```
ISOLINUX 2.10 2004-06-18 Copyright (C) 1994-2004 H. Peter Anvin
Welcome to Slackware version 10.0 (Linux kernel 2.4.26)!

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

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

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

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

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

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

boot:
```

Cuando la instalación se realiza por CD-ROM automáticamente se cargan las imágenes de los discos boot y root.

Después de arrancar la instalación Linux detectara la mayor parte del hardware que esta instalado en nuestro equipo.

El primer paso de la instalación es elegir la configuración de nuestro teclado.

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



Elegimos la opción qwerty/es.map para el mapa en español



A continuación debemos probar la localización de los caracteres en el teclado, para verificar que la distribución de los caracteres corresponde con el tipo de mapa que elegimos.



Básicamente hay que identificar los caracteres que se muestran en la imagen ya que son los que se utilizan con mayor frecuencia.



A continuación tenemos que registrarnos ante el sistema para que nos entregue un shell y seguir con la instalación, para ello hay que registrarnos como root.

```
Welcome to the Slackware Linux installation disk! (version 10.0)
##### IMPORTANT! READ THE INFORMATION BELOW CAREFULLY. #####

- You will need one or more partitions of type 'Linux' prepared. It is also
  recommended that you create a swap partition (type 'Linux swap') prior
  to installation. For more information, run 'setup' and read the help file.

- If you're having problems that you think might be related to low memory (this
  is possible on machines with 16 or less megabytes of system memory), you can
  try activating a swap partition before you run setup. After making a swap
  partition (type 82) with cfdisk or fdisk, activate it like this:
  mkswap /dev/<partition> ; swapon /dev/<partition>

- Once you have prepared the disk partitions for Linux, type 'setup' to begin
  the installation process.

- If you do not have a color monitor, type: TERM=vt100
  before you start 'setup'.

You may now login as 'root'.

:slackware login: root_
```

Ahora tenemos que crear las particiones necesarias para la instalación.

El administrador debe tener un esquema de disco pensado para una administración óptima y que sea flexible para facilitar cualquier tarea que se desee realizar, pensando en la funcionalidad del servidor.

Tener todo el sistema en una sola partición es mala idea ya que no es flexible y hace difícil de administrar el servidor. Lo más recomendable es hacer particiones pensando en la carga de archivos del sistema, de los usuarios, servicios y otras consideraciones, logrando con ello facilitar las tareas de respaldo, actualización, administración fácil y eficiente, entre muchas otras ventajas.

En la mayoría de los casos las particiones de mayor tamaño albergaran los directorios /home y /usr.



Realizaremos una instalación pensando en un equipo con dos sistemas operativos instalados. Linux y otro, para ello ya se tiene instalado el otro sistema en su propia partición por lo que resta hacer las particiones para nuestro Linux

Para consultar las particiones de disco damos el comando p

El cual imprime las particiones que se han definido hasta el momento en el disco:

```
Command (m for help): p
Disk /dev/hda: 8622 MB, 8622931968 bytes
255 heads, 63 sectors/track, 1048 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1            1           510     4096543+  2d  Unknown

Command (m for help): _
```

Pensando en una instalación básica se crearan solo dos particiones para Linux, la swap y la nativa de linux: n (crea una nueva partición)

Ahora generaremos una segunda partición de Linux en la cual se instalará el sistema:

```
/dev/hda1            1           510     4096543+  2d  Unknown
/dev/hda2            511          530      224910    83  Linux

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 3
First cylinder (539-1048, default 539):
Using default value 539
Last cylinder or +size or +sizeM or +sizeK (539-1048, default 1048): +4000M

Command (m for help): p
Disk /dev/hda: 8622 MB, 8622931968 bytes
255 heads, 63 sectors/track, 1048 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1            1           510     4096543+  2d  Unknown
/dev/hda2            511          530      224910    83  Linux
/dev/hda3            539         1025     3911827+  83  Linux

Command (m for help): _
```

Observe que las dos particiones para nuestro nuevo sistema son de tipo Linux, hay que cambiar el tipo de la segunda partición del disco para que sea de tipo swap.



Para cambiar el tipo presionamos la letra t (de tipo) e indicamos al sistema que queremos modificar la partición numero 2.

```

Device Boot      Start      End      Blocks  Id System
/dev/hda1          1        510     4096543+ 2d Unknown
/dev/hda2         511        538     224910   83 Linux
/dev/hda3         539       1025     3911027+ 03 Linux

Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): _

```

A continuación nos pide el código en formato hexadecimal, si desconocemos el código podemos consultarlo tecleando la tecla l (de listar) de la cual elegiremos el código que le corresponde:

```

0 Empty          1c Hidden W95 FAT3 70 DiskSecure Mult bb Boot Wizard hid
1 FAT12          1e Hidden W95 FAT1 75 PC/IX          be Solaris boot
2 XENIX root     24 NEC DOS        80 Old Minix      c1 DRDOS/sec (FAT-
3 XENIX usr      39 Plan 9          a1 Minix / old Lin c4 DRDOS/sec (FAT-
4 FAT16 <32M    3c PartitionMagic 82 Linux swap     c6 DRDOS/sec (FAT-
5 Extended      40 Unix 002005    03 Linux          c7 Syrix
6 FAT16         41 PPC PReP Boot   84 OS/2 hidden C: da Non-FS data
7 HPFS/NTFS     42 SFS             85 Linux extended db CP/M / CTOS / .
8 AIX           4d QNX4.x           86 NTFS volume set de Dell Utility
9 AIX bootable  4e QNX4.x 2nd part 87 NTFS volume set df BootIt
a OS/2 Boot Manag 4f QNX4.x 3rd part 8e Linux LUM      e1 DOS access
b W95 FAT32     50 OnTrack DM      93 Amoeba        e3 DOS R/O
c W95 FAT32 (LBA) 51 OnTrack DM6 Aux 94 Amoeba BBT     e4 SpeedStor
e W95 FAT16 (LBA) 52 CP/M           9f BSD/OS        eb BeOS fs
f W95 Ext'd (LBA) 53 OnTrack DM6 Aux a0 IBM Thinkpad hi ee EFI GPT
10 DPU          54 OnTrackDM6     a5 FreeBSD      ef EFI (FAT-12/16/
11 Hidden FAT12  55 EZ-Drive       a6 OpenBSD      f0 Linux/PA-RISC b
12 Compaq diagnost 56 Golden Bow     a7 NeXTSTEP     f1 SpeedStor
14 Hidden FAT16 <3 5c Priam Edisk    a8 Darwin UFS    f4 SpeedStor
16 Hidden FAT16  61 SpeedStor     a9 NetBSD       f2 DOS secondary
17 Hidden HPFS/NTF 63 GNU HURD or Sys ab Darwin boot   fd Linux raid auto
18 AST SmartSleep 64 Novell Netware b7 BSDI fs        fe LANstep
1b Hidden W95 FAT3 65 Novell Netware b8 BSDI swap      ff BBT

Hex code (type L to list codes): 82_

```

Aquí se muestra el listado donde vemos que el código 82 es el de Linux Swap.

Una vez que se ha cambiado el tipo queda de la siguiente manera:

```

10 Hidden W95 FAT3 65 Novell Netware b8 BSDI swap      ff BBT
Hex code (type L to list codes): 82
Changed system type of partition 2 to 82 (Linux swap)
-----
Command (m for help): p

Disk /dev/hda: 8622 MB, 8622931968 bytes
255 heads, 63 sectors/track, 1048 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device Boot      Start      End      Blocks  Id System
-----
/dev/hda1          1        510     4096543+ 2d Unknown
/dev/hda2         511        538     224910   82 Linux swap
/dev/hda3         539       1025     3911027+ 83 Linux
-----
Command (m for help): _

```



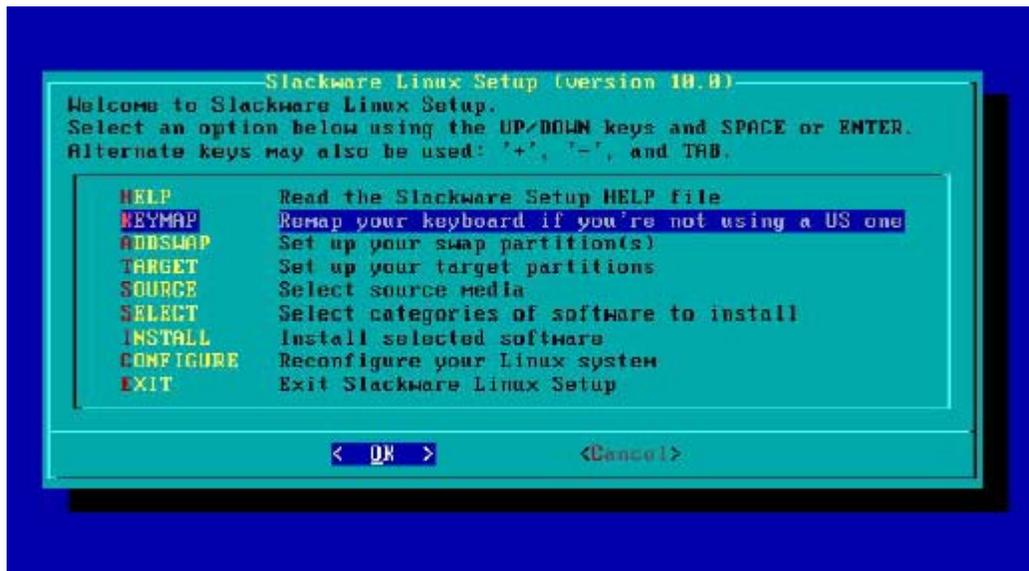
Ahora tenemos que guardar los cambios realizados en las particiones y comenzar la instalación con el comando setup.

```
Device Boot      Start          End      Blocks      Id System
/dev/hda1        1             510     4096543+    2d Unknown
/dev/hda2        511           538       224918     82 Linux swap
/dev/hda3        539          1025     3911827+    83 Linux

-----
Command (M for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
root@slackware:/# setup
```

Se despliega el asistente de instalación ahora en formato gráfico en forma de menú:



Aquí tenemos que seguir el procedimiento de arriba hacia abajo, para lo cual nos desplazamos a KEYMAP para elegir el tipo de teclado como anteriormente se realizó.

Ya que se ha definido el tipo de teclado el sistema automáticamente detecta la partición de swap, nos solicita darle formato y la prepara para la instalación:





```
          FORMATTING SWAP PARTITION
Formatting /dev/hda2 as a Linux swap partition (and
checking for bad blocks)...
```

```
          SWAP SPACE CONFIGURED
Your swapspace has been configured. This information will
be added to your /etc/fstab:

/dev/hda2      swap          swap          defaults      0      0
(100%)
< OK >
```

Una vez que ha preparado el área de swap el proceso de instalación reconoce las particiones de Linux, la partición que elijamos en este momento es donde se va a instalar el sistema.

Ya que elegimos la partición solicita darle formato rápido, formato con revisión de sectores dañados o montar la partición sin formatear. Si elegimos la opción de dar formato nos solicita el tipo de Sistema de Archivos con el cual va a ser formateada la partición.

Ya terminado el formato de todas las particiones entrega un reporte con información de la(s) particiones.

El siguiente paso es elegir el recurso desde el cual se va a realizar la instalación.

En este caso le indicamos que vamos a instalar desde los CD, el asistente nos mostrará el siguiente mensaje solicitando auto montaje o montaje del CD en forma manual

Lo mas conveniente es decirle que aplique un montaje del CD en forma automática, ya que de otra forma necesitaríamos saber en que IDE se encuentra conectada la unidad de CD y si esta como maestro o como esclavo, información que solo la podemos obtener abriendo el CPU o revisando la configuración en el BIOS

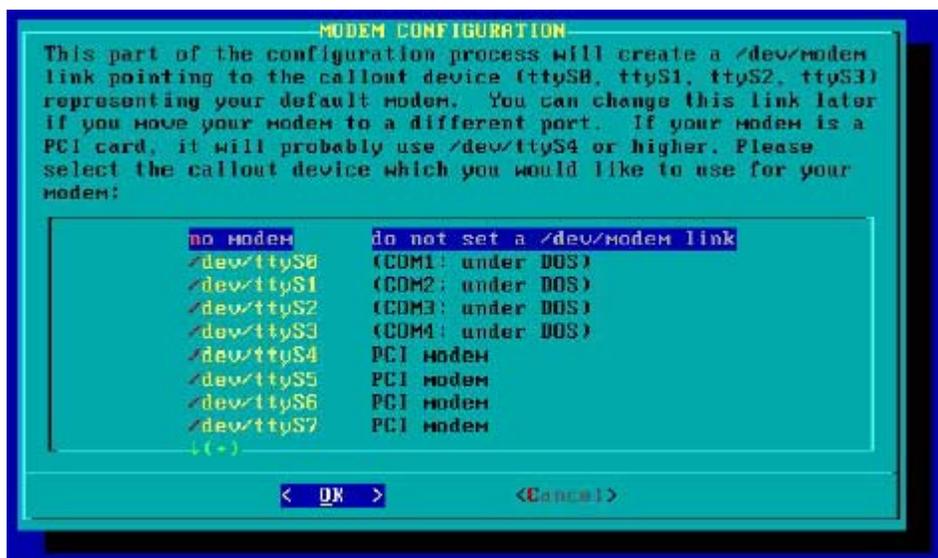
Ya que se ha detectado el CD el asistente nos pedirá elegir las series de paquetes que se van a instalar.

Con la barra espaciadora marcamos o desmarcamos las series de paquetes. Ahora tenemos que elegir el modo de instalación, el modo Full es automático, los otros modos solicitan la instalación de los paquetes ya sea por series o aplicando algún otro criterio.

Después de haber terminado la instalación de paquetes continuamos con la configuración del sistema, el siguiente paso es indicarle desde que medio deseamos instalar el Kernel para el sistema, obviamente en este caso es desde CD.



Ahora tenemos que elegir cual kernel es el que vamos a instalar, este depende de la arquitectura y de las características de la computadora donde se esta instalando. El asistente también nos permite configurar un modem, si es que contamos con uno externo o que se encuentre en una ranura de expansión podremos configurarlo fácilmente, si es un MODEM integrado a la tarjeta principal generalmente no es detectado ya que pertenece a la familia de los winmodem.



El siguiente paso es habilitar el hotplug, esta opción reconoce gran variedad de dispositivos de hardware y ayuda a la activación automática de la mayoría de ellos.

LiLo es el gestor de arranque el cual nos ayuda a generar un menú de inicio para equipos multisistemas, entre otras cosas, como la instalación esta contemplada para una maquina que tiene dos sistemas operativos se realizará la configuración, de este menú elegimos la opción en modo experto.

El cual nos lleva a un menú. Este menú nos permite iniciar la configuración de LILO con las propiedades principales, para ello tenemos que iniciar con la opción Begin:



Enseguida podemos elegir del menú el tipo de resolución que deseamos para modo texto o consola, para ello necesitamos saber las características de la tarjeta de video y que tipos de resolución soporta, hoy en día casi todas las tarjetas soportan una resolución de 1024x768x256.

El siguiente paso es indicarle en donde deseamos instalar LILO, lo recomendable es que se instale en el MBR del disco duro. Para lo cual el asistente nos preguntara en que disco deseamos que se instale, esto por si contamos con más de un disco.

Ahora tenemos que elegir un tiempo de espera antes que inicie con el sistema primer sistema que tiene por default.

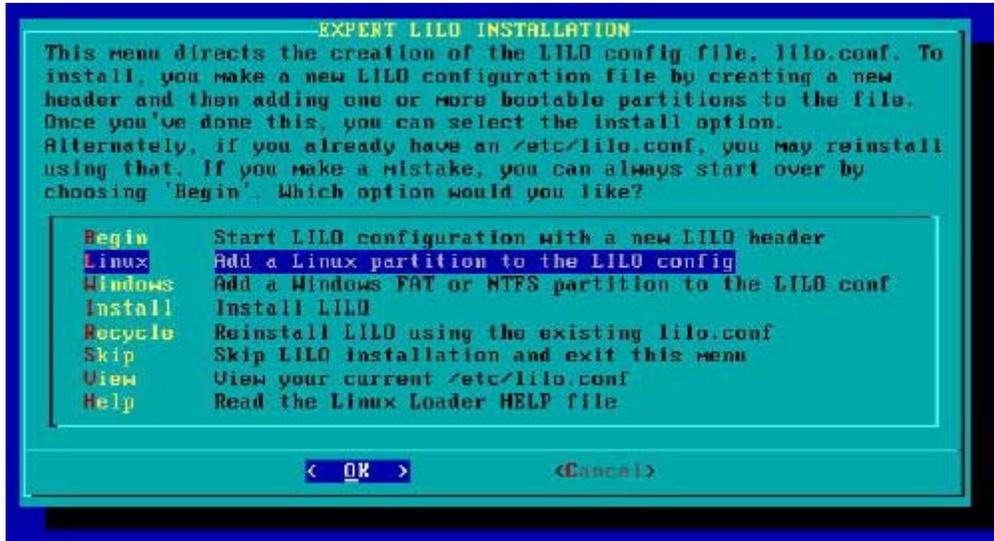
Hemos terminado de configurar la parte inicial de LILO, ahora hay que agregar los sistemas que deseamos sean visibles en el menú de booteo de los sistemas operativos.

Ahora tenemos que elegir un tiempo de espera antes que inicie con el sistema primer sistema que tiene por default.

El asistente nos mostrará un listado de las particiones que son de sistemas tipo Windows, solicitándonos la elección de la partición apropiada.

Ahora nos preguntará por la etiqueta con que queremos que se muestre este sistema en el menú de booteo, así de esta forma hemos agregado el sistema Windows, ahora tenemos que agregar Linux siguiendo este mismo procedimiento pero usando la opción de Linux del menú de configuración.

En seguida se configura la partición de Linux, para ello del menú elegimos dicha opción.



Nos muestra las *particiones* de tipo linux de la cual elegimos la partición apropiada

De igual forma tenemos que insertar la etiqueta como nombre que llevará en el menú de booteo

De esta forma hemos terminado de agregar los sistemas a nuestro gestor de arranque, si tuviéramos más sistemas el procedimiento sería el mismo.

1.3 Editores para la creación de páginas web

Objetivo:

Se adquirirán los conocimientos necesarios para el diseño de páginas, así como Software empleado para el desarrollo de la página, manipulación y tratamiento de imágenes.

Información necesaria para el desarrollo de la página

Es importante definir el objetivo principal de la página WEB a crear, ya que este parámetro permitirá al diseñador el establecer las prioridades en su desarrollo, así como el enfoque de la información y la presentación de la misma.

Una vez establecido el objetivo de la página, se procede a hacer una recopilación del material escrito y gráfico que permitirá conformar el contenido de la misma, dicha información puede obtenerse haciendo un estudio del tema o bien apoyándose en información y archivos que pueden ser proporcionados por la empresa o de la propia Internet.

1.3.1 Software empleado para el desarrollo de la página

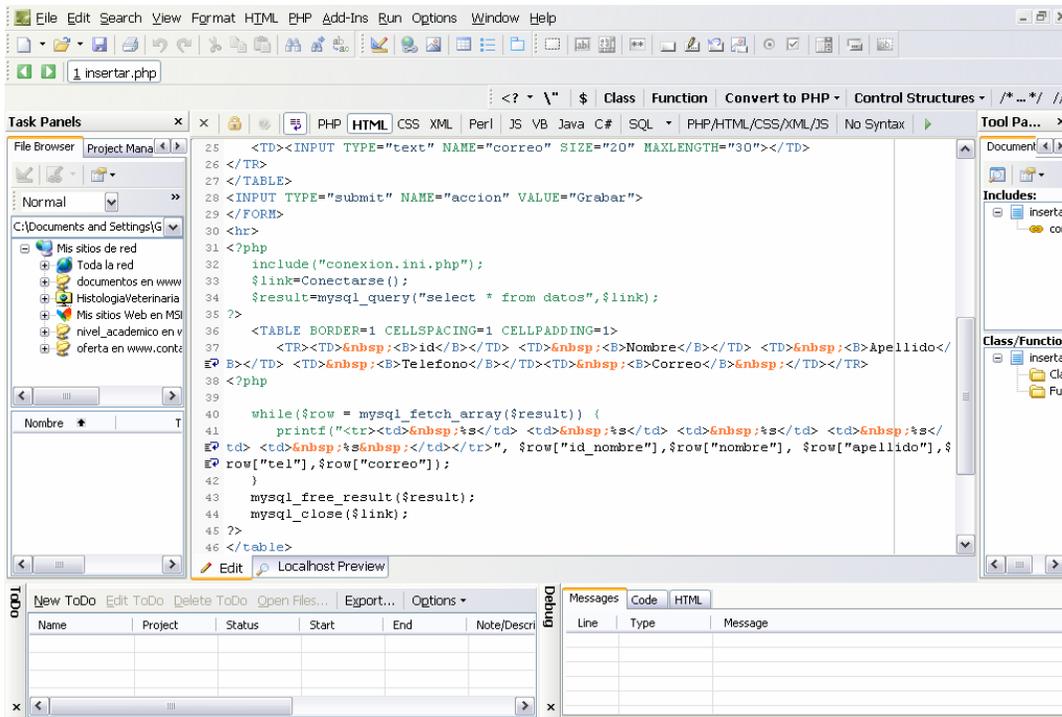
El diseño de una página Web se ve apoyado en gran medida al emplear un editor de HTML, el cual facilita las tareas a realizar ya que asiste al diseñador, construyendo por sí solo, las etiquetas de HTML, además de brindarle una referencia rápida de etiquetas, así como la versatilidad de manipular la información empleando la filosofía de los procesadores de texto. Algunos ejemplos de editores de HTML son:



- Front-page (Microsoft)
- Hot Dog
- Hot Metal

Como ejemplo se tiene HOT DOG EDITOR DE HTML

El editor HOTDOG tiene la filosofía de ir construyendo las páginas mediante una serie de asistentes que le van dando al usuario opciones para crear cada una de las etiquetas de HTML.



En el área de trabajo del HOT DOG se van escribiendo las etiquetas de HTML o bien se crean a través de asistentes que se activan al presionar el botón de la etiqueta deseada. Para generar una etiqueta con asistente se hace clic en la posición en donde se va a insertar la nueva etiqueta y seguido de esto se hace clic en el botón de asistente perteneciente al nuevo elemento que se va a insertar.

Manipulación de imágenes.

Al trabajar con páginas WEB es necesario manejar alguna herramienta que permita convertir archivos de gráficos de un formato a otro, además de permitir cambiar tamaño, color, rotar o hasta editar la imagen; para hacer esto es posible apoyarse en diversas utilerías que le permitan al usuario realizar todos estos trabajos con las imágenes. Por mencionar algunos ejemplos de herramientas que permiten manipular estos atributos existe software como LVIEW, COREL PHOTO PAINT, GRAPHIC WORKSHOP, entre otros.



Cambio de formato de imágenes

Normalmente las utilerías tienen la capacidad de abrir los archivos de gráficos en los formatos más comunes, por lo que una vez abierto el archivo para cambiarlo de formato basta con activar la opción de “Guardar Como” y seleccionar en la opción de tipo de archivo el formato GIF o JPG.

Las imágenes pueden ser cambiadas fácilmente de tamaño para que aparezcan en la proporción deseada en las páginas que se diseñen. Las utilerías traen una opción para cambiarle el tamaño a la imagen activa en ese momento, normalmente con el nombre de expandir/comprimir o RESIZE, en donde simplemente se escribe el nuevo tamaño de la imagen.

Las utilerías permiten también hacer cambios o modificaciones en las imágenes tales como agregar texto a la imagen, modificarla, rotarla y cambiarla de color, basta con familiarizarse con las opciones del software para aplicar los efectos deseados.

Concepto de gif's animados

Un elemento que le da mucha vista a las páginas WEB es la incorporación de GIF's animados, los cuales se construyen dibujando una serie de imágenes que se unen en un solo archivo gif mediante una herramienta de software o bien algunas herramientas ya cuentan con utilerías que aplican efectos a una imagen fija dando origen a todos los cuadros del gif animado. Algunos ejemplos de utilerías de software son el GIF-ANIMATOR, Gif Construcción, Ani magic y Giftrans.

Actualmente existen herramientas de software que permiten aplicar efectos a las imágenes para generar las animaciones, como por ejemplo el GifAni que permite abrir un archivo de imagen y después basta con seleccionar una opción del menú de efectos para que genere los cuadros de la animación con base en el efecto elegido.

Visualizadores de páginas.

Algunos de los visualizadores más actuales, compatibles y con mayor número de opciones integradas son;

- Netscape
- Internet Explorer
- Mosaic

Estos visualizadores se pueden bajar de Internet haciendo un ftp anónimo a;

<ftp://ftp6.netscape.com/pub>

<ftp://ftp.ncsa.com>

Etiquetas de HTML.

Las etiquetas de HTML son el conjunto de instrucciones o comandos que después son interpretados por los visualizadores de WEB, desplegando como resultado de esta interpretación las páginas WEB. Las Etiquetas de HTML se interpretan en el orden en que se ponen y van encerradas entre pico paréntesis (< >). Estas etiquetas no son sensitivas de tal forma que se pueden escribir con mayúsculas o minúsculas indistintamente.

Estructura básica de una página de HTML.



Las páginas de HTML pueden escribirse en cualquier editor que maneje formato de solo texto o bien con un procesador de texto que permita guardar en el formato de solo texto, o bien apoyándose en un editor de HTML que maneja ya el formato del archivo de texto por default. La estructura básica de una página debe llevar el siguiente esqueleto o estructura base:

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

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

```
<head>.. </head>
```

La etiqueta encabezado identifica la sección inicial del documento html. Entre otras cosas escribe el título del documento de esa sección.

```
<title>...</title>
```

La etiqueta titulo contiene el título que aparece en la barra de título del navegador web

```
<body>...</body>
```

La etiqueta de cuerpo abarca la parte más grande de una página Web. Contiene todo lo que el usuario ve cuando entra a esa página

La mayoría de las etiquetas se colocan dentro de BODY.

```
<hn>...</hn> para n=1..6
```

La etiqueta encabezado, crea varios tamaños de encabezados. La n se reemplaza por un número entre 1 y 6 cuanto menor sea el número mayor será el encabezado.

```
<hr>
```

La etiqueta de regla horizontal, dibuja una línea horizontal en medio de la página Web, haciendo más fácil su lectura.

```
<br>
```

La etiqueta de interrupción origina un salto de línea, con excepción de que coloca una línea en blanco como separador obligadamente

```
<p>
```

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

```
<i>....</i>
```

Etiqueta de texto itálico o cursivas

```
<b>...</b>
```

Etiqueta de texto bold o negritas

```
<u>...</u>
```

Etiqueta de texto subrayado

```
<tt>...</tt>
```



Etiqueta de texto en tipo máquina de escribir (monoespaciada)

```
<a href="direccion"> ... comentario ...</a>
```

Esta etiqueta de vínculos permite incorporar ligas o vínculos hacia otras páginas del WWW u otros URL's (URL localizador uniforme de recursos) .

```

```

Esta etiqueta permite incorporar imágenes.

```
<blink> ... </blink>
```

Esta etiqueta hace que el texto aparezca con un parpadeo.

```
<pre>... </pre>
```

Etiqueta de texto preformateado, esta etiqueta despliega el texto exactamente como fue escrito, respetando múltiples espacios en blanco y saltos de línea, sin embargo el texto aparece con formato de letra monoespaciada.

Encabezados o headers

La etiqueta de encabezados permite resaltar un texto que se representa en la página como un título o encabezado. La etiqueta puede ser <H1>, <H2>, <H3>, <H4>, <H5> y <H6>, donde <H6> le aplica al texto el tipo de letra más pequeño y <H1> le aplica el tipo de letra más grande.

```
<H1> Titulo de una página </H1>
```

Titulo de una página

```
<H6> nota aclaratoria </H6>
```

Etiquetas de formato

Las etiquetas de formato más importantes son;

```
<CENTER> ..... </CENTER>
```

Coloca alineado al centro de la página todos los elementos que se coloquen entre las etiquetas.

```
<I>...</I>, <B>...</B>, <U>...</U>, <TT>...</TT>
```

Dan atributos al texto que esta encerrado dentro de las etiquetas, empleando ITALICAS, NEGRITAS, SUBRAYADO y MONOESPACIADO respectivamente.

Manejo de colores

Los colores se establecen con un número hexadecimal (0-F) de 6 dígitos, de los cuales los dos primeros representan la cantidad de ROJO, el tercero y cuarto la cantidad de VERDE, y los dos últimos la cantidad de AZUL.

Por ejemplo algunos colores serian;

#FFFFFF Blanco

#FF0000 Rojo

#00FF00 Verde

#0000FF Azul

#FFFF00 Amarillo

#00FFFF Cian

#FF00FF Magenta





Colores de la página

Los colores de la página se pueden definir en la etiqueta de <body> tal y como se ve en el ejemplo siguiente;

```
<BODY TEXT="#FFFFFF" LYNK="#FF00FF" VLINK="#FFFF00" ALINK="#FFFFFF" BGCOLOR="#000000">
```

Donde TEXT define el color del texto de la página, LYNK define el color de los vínculos nuevos, VLINK el color de los vínculos ya visitados, ALINK el color de los vínculos que se están accedando en ese momento y BGCOLOR que define el color del fondo o BACKGROUND de la página.

Imagen como fondo de la página

La página puede tener como fondo una imagen en formato GIF, de tal forma que incluyendo el modificador BACKGROUND se puede incluir la imagen como mosaico en el fondo de la página. Cabe mencionar que no se pueden usar GIF's animados, hasta el momento, como fondo de la página.

La etiqueta se emplea entonces como se ve en el ejemplo siguiente;

```
<BODY BACKGROUND="FONDO.GIF" >
```

Hiperligas

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

```
<A HREF="dirección del recurso"> Texto o imagen del vínculo</A>
```

Este tipo de ligas puede acceder a recursos como otra página WEB, Correo Electrónico, FTP, Gopher, Usenet, Telnet y otros. Por ejemplo se puede hacer referencia a estructuras como las siguientes.

```
<A HREF="http://hermes máscarones.unam.mx">WWW de máscarones </A>
```

```
<A HREF="ftp://pegaso máscarones.unam.mx">Archivos a transferir </A>
```

```
<A HREF="gopher://gopher.unam.mx"> Gopher de la UNAM </A>
```

Listas de elementos con HTML

Las listas en HTML permiten elaborar rápidamente relaciones de elementos que son mostrados en diferentes formatos en las páginas de WEB dependiendo del tipo de lista que se emplea.

Listas numeradas

Estas listas van numerando automáticamente los elementos conforme se van agregando, además es posible elegir el tipo de numerador para elaborar la lista. Para cambiar el tipo de numerador se debe agregar en la etiqueta <OL TYPE=1>, que inicia la lista, el modificador TYPE que puede tener los siguientes valores dependiendo del resultado deseado en el numerador de la lista;

1	NUMERACIÓN ESTANDARD
a	LETRAS MINÚSCULAS
A	LETRAS MAYÚSCULAS
i	NÚMEROS ROMANOS PEQUEÑOS
I	NÚMEROS ROMANOS GRANDES



Listas con viñetas

Estas listas colocan una marca o viñeta en cada uno de los elementos definidos de tal forma que para agregar un nuevo elemento basta con colocar una nueva etiqueta , antes de cerrar la lista con , y seguido de la etiqueta el texto que se quiere mostrar como nuevo elemento de la lista.

```
<UL>
```

```
<LI> Primer elemento
```

```
<LI> Segundo elemento
```

```
</UL>
```

Imágenes

La etiqueta permite incorporar imágenes a una página WEB, las imágenes deben estar en formato GIF o JPG ya que estos son los únicos formatos que reconoce el visualizador. Una imagen se inserta escribiendo una etiqueta HTML como la siguiente;

```
<IMG SRC="Archivo.gif"> ó <IMG SRC="Archivo.jpg">
```

Es posible cambiar el tamaño de la imagen empleando los modificadores de esta etiqueta WIDTH y HEIGHT que se definen en píxeles o por porcentaje como se ve en el siguiente ejemplo

```
<IMG SRC="Archivo.gif" WIDTH=200 HEIGHT=200>
```

ó

```
<IMG SRC="Archivo.gif" WIDTH=90%>
```

También se puede definir la alineación de una imagen respecto al texto escrito a continuación, de tal forma que se puede tener alineado en la parte superior (TOP), alineado a la parte inferior (BOTTOM) o al centro (MIDDLE).

```
< IMG SRC="Archivo.gif" ALIGN=TOP>
```

```
< IMG SRC="Archivo.gif" ALIGN=BOTTOM>
```

```
< IMG SRC="Archivo.gif" ALIGN=MIDDLE>
```

Además, en el caso de los visualizadores que no pueden desplegar imágenes, se tiene un modificador (ALT) para desplegar mensajes de texto alternos como se ve en el siguiente ejemplo.

```
< IMG SRC="Archivo.gif" ALT="Texto Opcional">
```

Las Tablas permiten incorporar a las páginas de WEB las estructuras compuestas por renglones y columnas que tradicionalmente se manejan en los procesadores de texto. Para formar una tabla se escribe una estructura base como la siguiente;

PR,PC	PR,SC
SR,PC	SR,SC

```
<TABLE BORDER>
```

```
<TR>
```

```
<TD> PRIMER RENGLON, PRIMER COLUMNA </TD>
```

```
<TD> PRIMER RENGLON, SEGUNDA COLUMNA </TD>
```

```
</TR>
```

```
<TR>
```

```
<TD> SEGUNDO RENGLON, PRIMER COLUMNA </TD>
```

```
<TD> SEGUNDO RENGLON, SEGUNDA COLUMNA </TD>
```

```
</TR>
```

```
</TABLE>
```



Tablas

La Tabla anterior es de dos renglones por dos columnas y en lugar de los textos escritos entre las etiquetas `<TD>..</TD>` se escribe el contenido que el usuario desea que aparezca en los cuadros (o celdas) de la tabla. El modificador `BORDER` que va en la etiqueta `<TABLE>` determina que la tabla lleva dibujadas las líneas divisorias y bordes de la tabla por lo que si se deseara la tabla sin borde basta con eliminar este modificador. La tabla tendrá tantas columnas como etiquetas `<TD>` escriba el usuario dentro de las `<TR>` y tantos renglones como etiquetas `<TR>` escriba dentro de las etiquetas `<TABLE>...</TABLE>`

Formularios

Los formularios emplean una serie de instrucciones que definen acciones o elementos dentro del formulario tales como las siguientes etiquetas de HTML;

```
<FORM METHOD="POST" ACTION="../cgi-bin/programás/datos">
```

Contenido de la forma

```
</FORM>
```

La etiqueta `<FORM>` define el inicio de la forma y lleva los modificadores `METHOD`, que puede ser `POST` o `GET` según el método de envío al servidor, así como `ACTION` que indica el nombre del programa a ejecutar para procesar la información enviada por el formulario.

```
<INPUT TYPE="..." NAME="...">
```

La etiqueta `<INPUT>` indica un elemento del formulario, donde `TYPE` define la clase de elementos tales como `"TEXT"` para campos de captura de texto junto con `SIZE=#` donde `#` define el número de caracteres, `"RADIO"` para casillas de opciones únicas, `"CHECK"` para casillas de verdadero o falso, `"submit"` para botones junto con `VALUE="TEXTO"` que define el texto que aparece en el botón, `"RESET"` que genera un botón que borra los valores introducidos y regresa a los valores por default y que junto con `VALUE="TEXTO"` indica el nombre del botón.

El atributo `NAME` indica el nombre del elemento.

Elementos básicos de un formulario

```
<INPUT TYPE="TEXT" NAME="NOMBRE" SIZE=25 VALUE="ESC. AQUÍ">
```

Esta etiqueta crea un campo para que se capture información mediante el teclado, asignándole el contenido de este campo a una variable `NOMBRE` de tamaño 25 y con valor inicial "ESC. AQUÍ".

```
<INPUT TYPE="PASSWORD" NAME="PASSW" SIZE=10>
```

Esta etiqueta crea un campo para que se capture información mediante el teclado, ocultándola al escribirla, asignándole el contenido de este campo a una variable `PASSW` de tamaño 10.

```
<INPUT TYPE="RADIO" NAME="OPCION" VALUE="UNO">
```

Esta etiqueta crea un campo de selección de opción única y se escribe con una lista de opciones, permitiendo seleccionar solo una, asignándole el contenido de este campo a una variable `OPCION` y dándole el valor de "UNO".

```
<INPUT TYPE="CHECKBOX" NAME="MULTI" VALUE="AZUL">
```

Esta etiqueta crea un campo de selección de opciones múltiples y permite seleccionar varias a la vez, asignándole el contenido de este campo a una variable `MULTI` dándole el valor de "AZUL".



```
<INPUT TYPE="SUBMIT" NAME="BOTON" VALUE="REGISTRAR">
```

Esta etiqueta crea un botón en el formulario de nombre BOTON y con el texto escrito en el botón de REGISTRAR.

```
<INPUT TYPE="RESET" NAME="BORRAR" VALUE="BORRAR DATOS">
```

Esta etiqueta crea un botón en el formulario de nombre BORRAR y con el texto escrito en el botón de BORRAR DATOS.

Menús desplegables en un formulario

Es posible crear listas de opciones que aparecen en menús desplegables, para esto se utiliza la etiqueta `<SELECT>` que define el nombre de la variable con `NAME` y las opciones se escriben con la etiqueta `<OPTION>`, de tal forma que para crear una lista con los días de la semana se puede seguir el ejemplo siguiente;

```
<SELECT NAME="DIA">  
<OPTION>Lunes  
<OPTION>Martes  
<OPTION>Miércoles  
<OPTION>Jueves  
<OPTION>Viernes  
</SELECT>
```

Áreas de texto en un formulario

Los campos de área de texto pueden contener varias líneas de texto a diferencia de los campos de tipo `TEXT`; por lo que cuando la entrada es una serie de líneas continuas son muy útiles. La etiqueta empleada para incluir estos elementos se llama `<TEXTAREA>` y se escribe como sigue;

```
<TEXTAREA NAME="COMENTARIOS" ROWS="20" COLS="20">
```

```
</TEXTAREA>
```

La etiqueta anterior define un área de texto de 20 renglones por 20 columnas y el contenido de este campo se lo asigna a una variable de nombre `COMENTARIOS`.

Frames

Los Frames son divisiones en forma de ventanas que se pueden definir desde `HTML` con la etiqueta `<FRAMESET>`. Actualmente la mayoría de los visualizadores permiten la incorporación de frames, sin embargo se debe tener presente que todavía existen algunos que no cuentan con esta capacidad por lo que se debe incluir la etiqueta `<NOFRAMES>` para poder considerar a estos visualizadores.

Para incluir frames en una página se sustituye el `<BODY>` y se trabaja con `<FRAMESET>`, de tal forma que se pueden crear frames Horizontales o Verticales dependiendo si se usa `<FRAMESET COLS="">` o `<FRAMESET ROWS="">`, en cada Frame se puede incorporar una página de `WWW` de tal forma que se emplea la etiqueta `<FRAME SRC="ARCHIVO.HTM" Name="">` para incorporar el contenido de cada `FRAME`.



Como ejemplo de una página con Frames se puede considerar el siguiente código de HTML;

```
<HTML>  
<FRAMESET COLS= "30%,70%">  
  <FRAME NAME= "UNO" SRC= "PRIMERA.HTM">  
  <FRAME NAME= "DOS" SRC= "DOS.HTM">  
</FRAMESET>  
</HTML>
```

En el caso de incorporar ligas de hipertexto se puede incorporar la etiqueta TARGET="NOMBREFRAME" para que se despliegue el contenido en el Frame especificado en TARGET.

```
<A HREF= "pagina.HTM" TARGET= "UNO">pagina UNO </A>
```

Actualización y Mantenimiento de la página

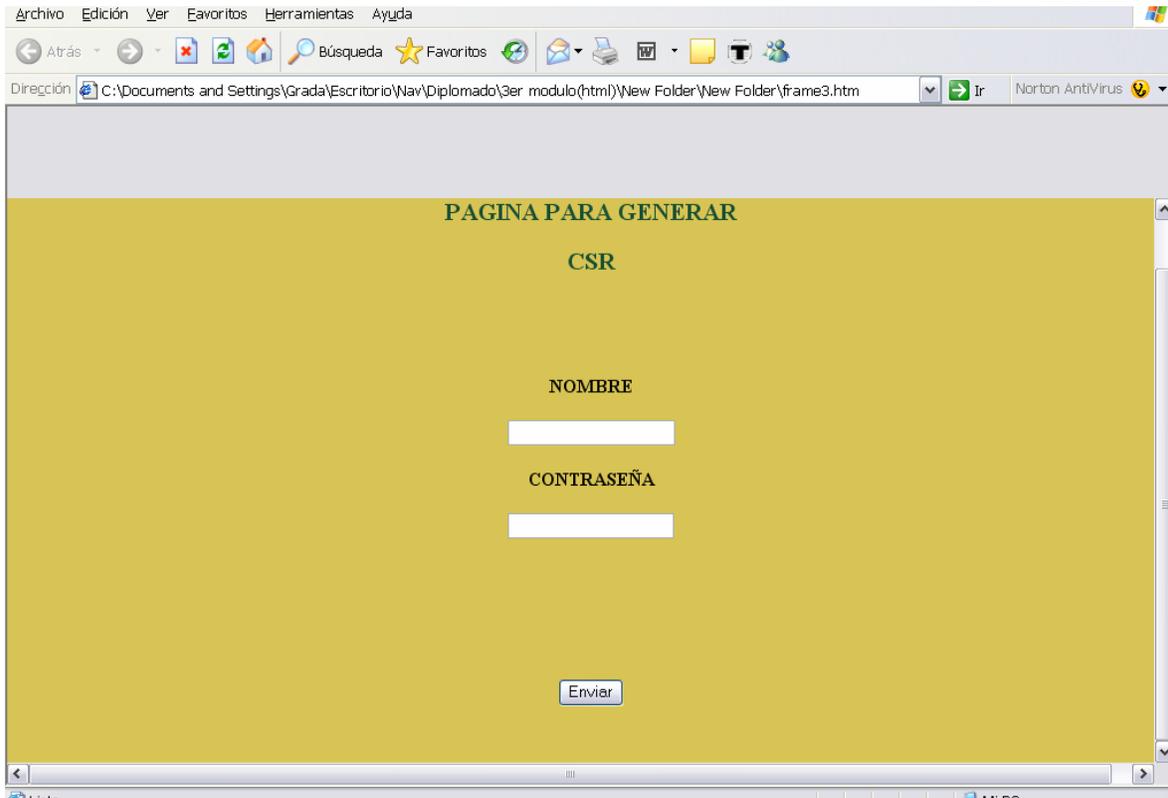
Finalmente, una vez terminada la página, es posible que se tengan que programar una serie de actividades encaminadas a estar actualizando y agregando información a la página según las necesidades de la empresa y la capacidad para actualizar esta información. Además es conveniente considerar que, por lo menos cada año, se debe plantear un nuevo diseño u actualización de la información tomando en cuenta los comentarios y cambios que puedan mejorar la página (hacer que cada año se proponga un nuevo diseño de página y se mejore en cada nueva versión).



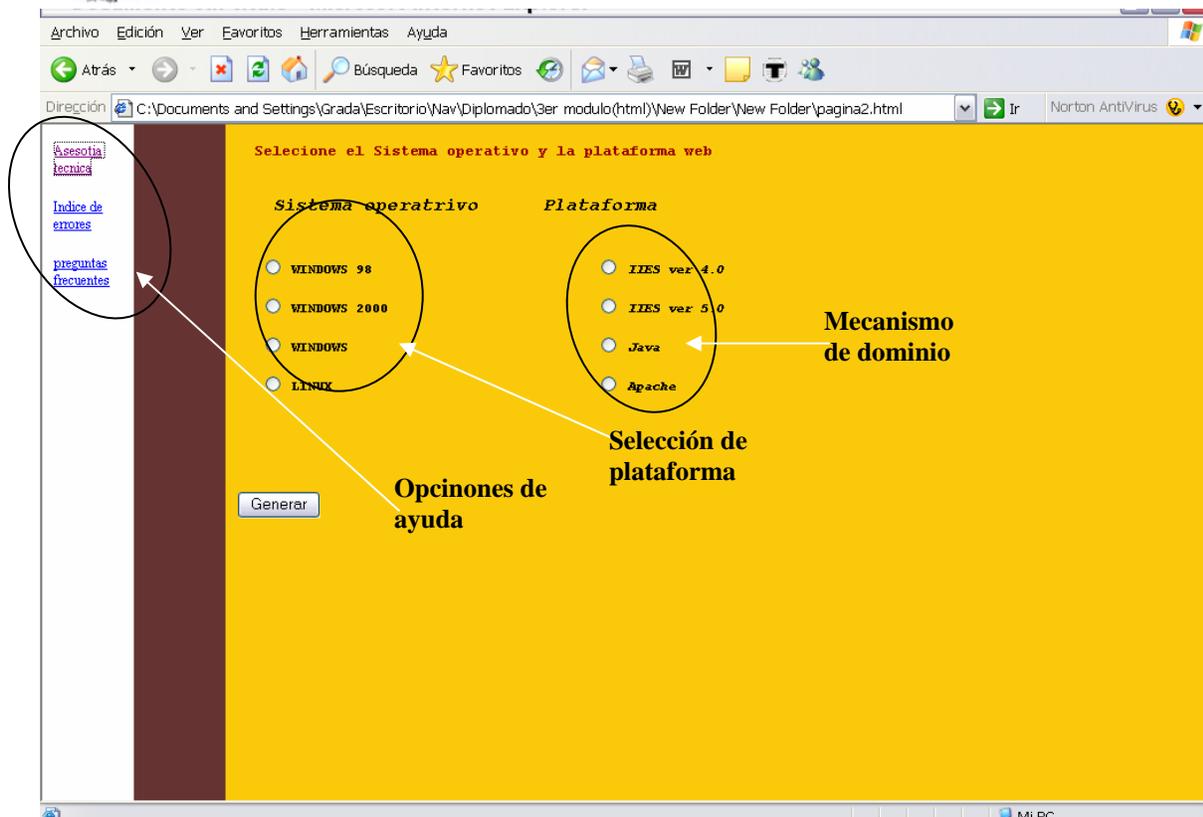
1.3.2 Práctica.

Aquí se muestra la práctica realizada con las herramientas que se proporcionaron.

Se realizó una página en la cual, ayuda a un usuario a generar una petición (CSR) a una entidad emisora de certificados digitales.



Como página inicial, se tiene una ventana donde nos pide nombre y contraseña, la cual si es correcta nos muestra la siguiente ventana.



En esta página nos permite seleccionar algunas de las plataformas que hay en el mercado, al igual que los diferentes mecanismos para el “levantamiento” de páginas.

También contiene opciones adicionales como lo es:

1. Asesora técnica.
2. Índice de errores.
3. Preguntas frecuentes.

Las cuales puede apoyar al usuario a resolver dudas con respecto a lo que se quiere realizar.

Ya que se tiene las opciones adecuadas se oprime el botón **Generar** y pasara a la siguiente ventana.



Archivo Edición Ver Favoritos Herramientas Ayuda

← Atrás → Búsqueda Favoritos

Dirección: [nts%20and%20Settings/Grada/Escritorio/Nav/Diplomado/3er%20modulo/\(html\)/New%20Folder/New%20Folder/pgin3.html#](#) Ir Norton AntiVirus

Generacion del CSR Asesoría Técnica | Índice de errores | Preguntas Frecuentes

[Ir a pag principal](#) | [VeriSign](#) | [Atras](#)

Contraseña / Seleccion de Servidor / Instrucciones CSR /

Instrucciones para generar su CSR

Este es un ejemplo de archivo de Petición de Firma de Certificado (CSR):

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBCTBTAIBADBPQoswCQYDQQGEwJVUzEQMA4GA1UECBMRmxvcmkYTEYMBYg
A1UEChMPRllyb3VibUaUGUgV2VlMRQwEgYDVQDFAT3d3cuZR3Lm5ldDBcMA0G
CSqGSIb3DQEBAAQsAMEgCQQCeojtnHqG0GTxp+XZ56RaSe1IZWpumXjU6Sx7
v1FdXzsY1oLOqa09Dtnu1WsQRHh0YDS+445oncjKm1zCG/IZAgMBAAGgADANBgkq
hkiG9w0BAQQFAANBAFBj9g+NIUh8YWPrFGntg4miUd/wqUshptjy4PjdsD3ugy
5avvuh3G//PpGh2ayXIJHpJXTUBQyzxSEIINYtc=
-----END NEW CERTIFICATE REQUEST-----
```

EJEMPLO DE CSR

Quando usted genere el CSR en el Paso 3, ya sea con su software de servidor que envia por correo electrónico que es la Petición de Firma de Certificado (CSR) o el que crea en forma de archivo con la solicitud en su disco duro (como por ejemplo: key.req). Abra este archivo de Petición de Firma de Certificado (CSR) con un editor de texto ASCII como el NotePad. (No utilice un procesador de texto como Word que inserta caracteres de formato y de control.)

PAGINA EN CONSTRUCCION

<p>Confirmar el Nombre de Dominio</p> <p>Advantage Security Systems sólo podrá emitir un Global Server ID al titular del nombre de dominio. Esta información será incluida en su Global Server ID como parte de su nombre distintivo.</p> <p>Esciba el nombre y la dirección del propietario del nombre de dominio registrado. Usted necesitará saber el nombre del propietario registrado y el dominio y ser capaz de introducirlo exactamente de la manera en que aparece il appears en los resultados de la petición de WHOIS.</p> <p>Descripción</p> <p>Búsqueda del nombre de dominio: Los nombres de dominio son registrados en una organización conocida como NIC (o InterNIC). Para verificar los datos relativos a su dominio vaya a la página de búsqueda apropiada de la columna derecha. Esciba su nombre de dominio y realice su búsqueda.</p> <p>Por ejemplo, si el sitio web de su compañía se encuentra en www.advantage-security.com, su nombre de dominio es advantage-security.com. Los resultados de la búsqueda serán, al menos, los siguientes:</p> <p>Resultados de la búsqueda Advantage Security Systems, S. de R.L. de C.V. VERISIGN-DOM Av. Prolongacion Reforma 625, Desp. 402 Torre Lexus Paseo de las Lunas, Santa Fe Mexico, DF 01330</p> <p>Nombre de Dominio: ADVANTAGE-SECURITY.COM</p> <p>En este ejemplo, la búsqueda nos muestra que el titular de ADVANTAGE-SECURITY.COM es</p>	<p>Páginas de búsqueda WHOIS</p> <ul style="list-style-type: none"> • Network Solutions (Dominios .com, .edu, .org, and .net) • Guatemala (Dominios .gt) • Honduras (Dominios .hn) • El Salvador (Dominios .sv) • Costa Rica (Dominios .cr) • Nicaragua (Dominios .ni) • Panama (Dominios .pa) • Venezuela (Dominios .ve) • Mexico (Dominios .mx) • Colombia (Dominios .co) • Todos los otros países
---	--

VerySign

[Contactenos](#)

[Atras](#)

[Section Link](#)

[Section Link](#)

Related Link Category

[Deportes](#)

[Entretenimieno](#)

[Seguridad](#)

[Descargas](#)

.MX es la nueva forma de decir México

Encuentra tu dominio pra consulta administrativa

Solo da un ckick a la imagen

Ya en esta parte de la página Web, proporciona las instrucciones con respecto a lo que se solicito en la anterior pagina.



También contiene un ejemplo de como debe verse la petición, ayuda (como la pagina anterior) y algunos links que servirán para realizar la certificación y la seguridad de la pagina que lo esta requiriendo.

1.4 Administración de servidores www con linux.

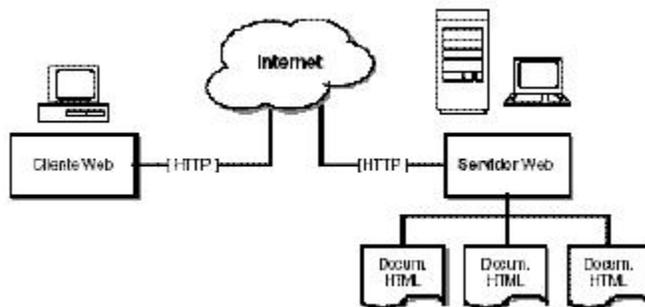
En la actualidad es necesario contar con sistemas que permitan la automatización del seguimiento y control del proceso que se desarrollan en diversas empresas y oficinas.

El desarrollo de estos sistemas pueden ser sustentados empleando software, libre como LINUX, PHP, apache-www-server y MYSLQ, que son herramientas que han demostrado se un alto desempeño, gran estabilidad y seguridad por el hecho de ser libres permiten reducir los costos que se generan por las licencias de uso de software, logrando aprovechar al máximo los equipos de computo con que se cuenta.

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

Objetivo general

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



¿Qué es un servidor WWW?

Es un servidor que ofrece servicios dentro del Word Wide Web en Internet.

Servidor WWW: es un programa encargado de ofrecer comunicación mediante el protocolo http. (HTTP – Hypertext Transfer Protocol.)

Servidores de WWW.



Los más populares son:

Internet Information Server - Microsoft
Sun Java Web Server - Sun Microsystems
Roxen Web Server – Open Source
Public Domain HTTP Daemon - NCSA
Zeus Web Server - Zeus
Apache Web Server – Open Source

Algunos criterios que se pueden seguir para una selección adecuada a nuestras necesidades pueden ser:

Función del Servidor de web
Expertise de los administradores
Plataforma disponible
Numero de conexiones concurrentes
Numero transacciones por segundo
Costo computacional por transacción
Proyección del crecimiento esperado.
Soporte para la tecnología utilizada para el desarrollo.
Análisis del retorno de Inversión.

En base a estos criterios el más indicado es Apache

HTTP

HTTP es el protocolo de red para el WWW, basa su operación en la arquitectura “Cliente-Servidor”. El servidor http es el encargado de publicar “recursos” electrónicos.

El cliente http hace una consulta los recursos que el servidor ofrece, simple y poderoso.

¿ Como funciona HTTP?

El cliente HTTP abre una conexión, el server manda un “acknowledge” notificando que se ha abierto una sesión; El cliente envía su “request message” solicitando un recurso.

El servidor responde con “response message” que contiene el recurso solicitado despues cierra la conexión.

¿Por qué Apache?

Robusto, Soporte de un gran numero de transacciones.
Configurable para diferentes entornos de trabajo.
Con un alto nivel de seguridad.
Disponible para una gran variedad de plataformas.
Soporte para servicio de proxy
Soporte para granjas de servidores
Soporte para Scripting languages integrados como modulos (por ejemplo PHP, mod_perl)
Incluye el codigo fuente del servidor
Soporte para accesos restringidos



Soporte para SSL
Y además es gratuito.

¿Por qué no IIS ?

Por que este puede contener Virus, Red Code, Nimda además un debil sistema de seguridad, Hackers.

Al ser basado en Windows es inestable e inseguro
Por su tecnología propietaria es imposible portar

Forma de instalación de un apache.

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

- RPM “The easy way”
- rpm -iv apache-1.3.27.rpm

Compilación “The best way”

```
tar zxvf apache-1.3.27.tgz
./configure
make
make install
```

Información básica del Servidor de Web Apache

Directivas

Apache es administrado por más de 200 directivas las cuales permiten que determinada funcionalidad puedan ser incluidas.

En Linux el administrador controla que directivas estarán disponibles de acuerdo a los módulos con los que se compila Apache.

Grupos de Directivas

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

Global Environment
Main Server
Virtual Servers

Global Environment

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

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



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

ServerName

Define el nombre para el servidor Apache, el cual será utilizado cuando se construyan los URLs cuando el cliente solicite otros recursos web (páginas, phps, cgis, etc.) del servidor.

Esta etiqueta se utiliza tanto en ServerConfig como en VirtualHost.

ServerType (Eliminado en 2.0)

Define el esquema mediante el cual opera apache, las posibles opciones son:

-standalone
-inetd

No se recomienda inetd debido a los recursos de cómputo que se consumen al obligar al superdemonio (inetd o xinetd) a generar un proceso httpd cada vez que se genere una nueva conexión.

User

Esta directiva define el user ID mediante el cual apache operará.

Valor por Default: User #-1

Group

Esta directiva define el group ID mediante el cual apache operará.

Valor por Default: User #-1

Port (Reemplazado por Listen en 2.0)

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

Valor por Default: 80

ServerAdmin

Esta directiva define el correo electrónico del administrador del servidor web e indica la dirección a incluirse en los mensajes de error que serán enviados al cliente.

Es necesario tener habilitada la directiva **ServerSignature Email**

DocumentRoot

Esta directiva define la ruta absoluta donde se almacenarán los archivos html que se desean publicar.

ServerRoot



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

Valor por Default: /usr/local/apache

Esta etiqueta se utiliza tanto en ServerConfig como en VirtualHost.

StartServers

Esta directiva define cual es el número máximo de servidores que se iniciaran cuando se arranca apache.

Valor por Default: 5

MaxClients

Esta directiva define cual es el número máximo de procesos servidores que como máximo podran ser iniciados, con el objeto de atender las solicitudes de los clientes.

Valor por Default: 20

ErrorDocument

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

Enviar un mensaje de error (default).

Enviar un mensaje personalizado.

Redirigir a un URL local para manejar el problema o error.

Redirigir a un URL externo quien manejara el problema o error.

error.log

Registra los errores que genere un acceso al sitio o que los procesos de Apache reporten.

Logging

HostNameLookup

HostNameLookup [on|off]

Esta directiva habilita la resolución de nombres en el DNS cuando se establece una conexión.

Nota: Por razones de rendimiento se sugiere no habilitarla. Para la resolución de nombres en las bitácoras existe el programa logresolve, tambien de apache, que puede realizar la resolución de nombre fuera de línea.



Configuración Servidor de Web Apache

<Directory> y <DirectoryMatch>

```
<Directory /usr/local/apache/htdocs>  
...  
</Directory>
```

La directiva Directory permite aplicar otras directivas de forma específica a todos los recursos dentro de la ruta definida por “dir”.

Nota: Se deben considerar rutas absolutas.

Options

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

Options [+/-] option [[+/-]option]
Sitios Web Dinámicos

- PHP
- JSP
- Servlets
- ASP
- ColdFusion
- ActiveX
- JavaScript
- Applets
- CGIs

Sitios Web Dinámicos





CGIs

La tecnología CGI “Common Gateway Interface” define un modelo de programación que puede ser implementado por múltiples lenguajes.

Los CGIs son programas que siguen un estandar definido, corren en el servidor, reciben parametros desde el cliente y su salida es enviada al navegador.

Fueron la primera alternativa para generar dinamismo a un sitio web.

AddHandler cgi-scripts .cgi .pl

AddHandler permite que determinada extensión sea relacionada a un evento en particular, en el caso de los CGIs lo que se indica es que las extensiones .cgi y .pl quedan identificada como un script.

Apache las interpretará en lugar de solo enviar el contenido del archivo al cliente.

Directivas requeridas:

➤ **AddType text/html .shtml**

Relaciona los archivos con extensión shtml como aquellos que contienen etiquetas SSI.

➤ **AddOutputFilter INCLUDES .shtml**

Habilita el filtro para los archivos .shtml

Nota: SSI representa un riesgo y debe ser usado con cautela, para evitar un incidente de seguridad.

VirtualHost

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

```
<VirtualHost servername>
```

```
</VirtualHost>
```

Ejemplo:

```
NameVirtualHost 192.168.40.5
<VirtualHost 192.168.40.5>
ServerAdmin root@web.ejemplo.com
ServerName www.micasa.com
DocumentRoot /home/micasa/htdocs/
</VirtualHost>
```

```
<VirtualHost 192.168.40.5>
ServerAdmin root@web.ejemplo.com
```



```
ServerName www.miempresa.com
DocumentRoot /home/miempresa/htdocs/
</VirtualHost>
```

Control de Acceso

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

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

Order

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

Allow

Define los hosts que tendrán acceso al recurso.

Deny

Define los hosts que no tendrán acceso al recurso.

Apache ofrece la funcionalidad al estilo “ACL”, con lo cual es factible determinar asignar un login y password por usuario, quienes tendrán acceso sobre recursos específicos del servidor Web.

AuthUserFile <archivo de autenticación>

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

Indica cual es el archivo que contiene la lista de usuarios y passwords para realizar la autenticación.

Nota: Es importante que este archivo no sea accesible por los clientes web, pues tendrían acceso a la lista de passwords y podrían atacarla por métodos de diccionario o incluso fuerza bruta y con ello conseguir nuevas claves del sistema.

Require valid-user

Esta directiva indica que se requiere de un usuario y una clave de acceso para un recurso determinado.

Modulos

En Apache los módulos son los encargados de agregar funcionalidad adicional incrementando el número de directivas disponibles.

Modulos estaticos

Se agregan al momento de compilar apache.

Modulos dinamicos

Se agregan durante el ambiente productivo, permiten adicionar funcionalidad sin necesidad de recompilar apache.



Ejemplo de Compilación

Modulo Estatico mod_php

```
$ tar -zxvf apache_1.3.x.tar.gz
$ cd apache_1.3.x
$ ./configure
$ cd ..

$ tar -zxvf php-4.x.y.tar.gz
$ cd php-4.x.y
$ ./configure --with-mysql --with-apache=../apache_1.3.x
$ make
$ make install
$ cd ../apache_1.3.x
$ ./configure --prefix=/www --activate-module=src/modules/php4/libphp4.a
(La linea de arriba es correcta, El archivo libphp4.a no existe en este momento el objeto es crearlo.)
$ make
$ make install
Copiar el archivo php.ini
$ cd ../php-4.x.y
$ cp php.ini-dist /usr/local/lib/php.ini
```

Agregar el tipo de archivo .php a la configuración de apache

```
httpd.conf
AddType application/x-httpd-php .php
tar -zxvf apache_xxx.tar
tar -zxvf php-xxx.tar
cd apache_xxx
./configure --prefix=/usr/apache2 --enable-module=so
make
make install
cd ../php-xxx
```

1.5 Programación con PHP

En este capítulo se analizaron las expresiones básicas para la creación de un sitio dinámico, en el cual hicimos accesos, modificaciones, eliminaciones a documentos de texto a través de una interfaz grafica.

El lenguaje PHP es un lenguaje de programación de estilo clásico, con esto quiero decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

Pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría



ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML.



Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

Los primeros conocimientos son la sintaxis básica, que cubre la incorporación de un código PHP en un texto de HTML. Con estas simples expresiones se inicia un código php:

- `<? ... ?>`
- `<?php ... ?>`
- `<SCRIPT LANGUAGE="php"> ... </SCRIPT>`
- `<% ... %>`

En los puntos suspensivos va el código que se vaya a ejecutar, siendo el inicio el primer conjunto de símbolos y el final el último conjunto. Hay que aclarar que el último método es para aquellos que conocen de programación ASP.

En la parte de variable se nos explica la diferencia y utilización de las variables que se ocuparán en la programación. Cada variable va a ir precedida del símbolo \$. Al igual que JavaScript, PHP es tipado, esto quiere decir que no se necesita especificar el tipo de variable antes de poder almacenar algún dato.

Existen 4 tipos de datos en PHP: los booleanos o boléanos que solo recibirán o enviarán una respuesta cierta o falsa. Los double también conocidos como de punto flotante, son usados para representar un número con punto decimal o exponente. Los enteros que son todos aquellos que son números cerrados (sin decimal). Y los string o cadenas que son aquellos que recibirán caracteres no numéricos. Estas últimas son especificadas entre comillas dobles(") o sencillas('). Cuando se encuentre dentro de comillas dobles serán evaluadas, y dentro de ellas se pueden representar caracteres especiales.

Se que cualquier expresión es una serie de caracteres y operadores llamado finalmente funciones. Las sentencias son de una llamada a una función específica. Estas a su vez pueden ser agrupadas en bloques delimitadas por {}. PHP ya tiene contenido dentro de su lenguaje un grupo de librerías que incluyen algunas funciones básicas, para eliminar la redundancia y simplificar el código, por



ejemplo la función print que imprime en pantalla el texto o resultado solicitado, o los ya clásicos en lenguajes de programación for, while, do while, también llamados ciclos de repetición. Las constantes son iguales a las variables con la diferencia que estas no pueden ser redefinidas con otro valor.

El uso de arreglos, manejo de cadenas e inclusión de archivos, se pueden ver las ventajas no solo en el área matemática, sino como apoyo en estructuras mas avanzadas, con el se pueden hacer rotaciones de puntos y crear la medicación de archivos de imagen. Con el manejo de cadenas se puede hacer una valoración de los datos escritos, digamos por ejemplo un formulario. Y la inclusión de archivos, es una mirada rápida a una base de datos pequeña, si los datos no son suficientemente grandes para un base de datos, pueden ser almacenados en un documento de texto.

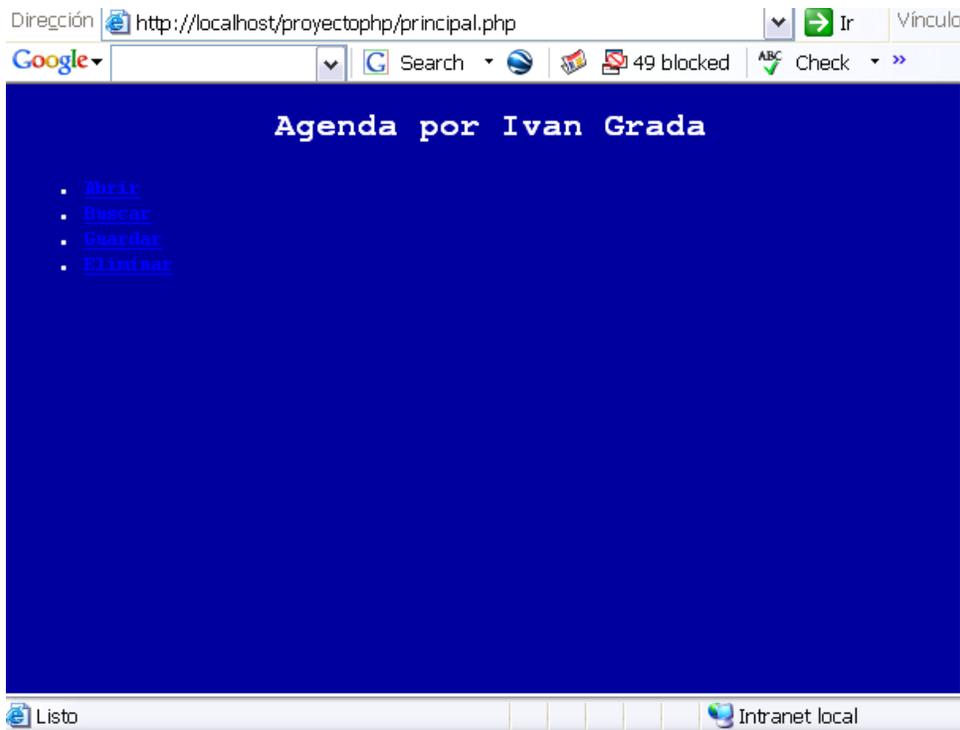
1.5.1 Práctica

La práctica consistía en una agenda, la cual debe contener: agregar, borrar, buscar, y mostrar cualquier dato en la agenda. Toda esta información debe estar contenida en texto plano.

La práctica se realizo de la siguiente manera:

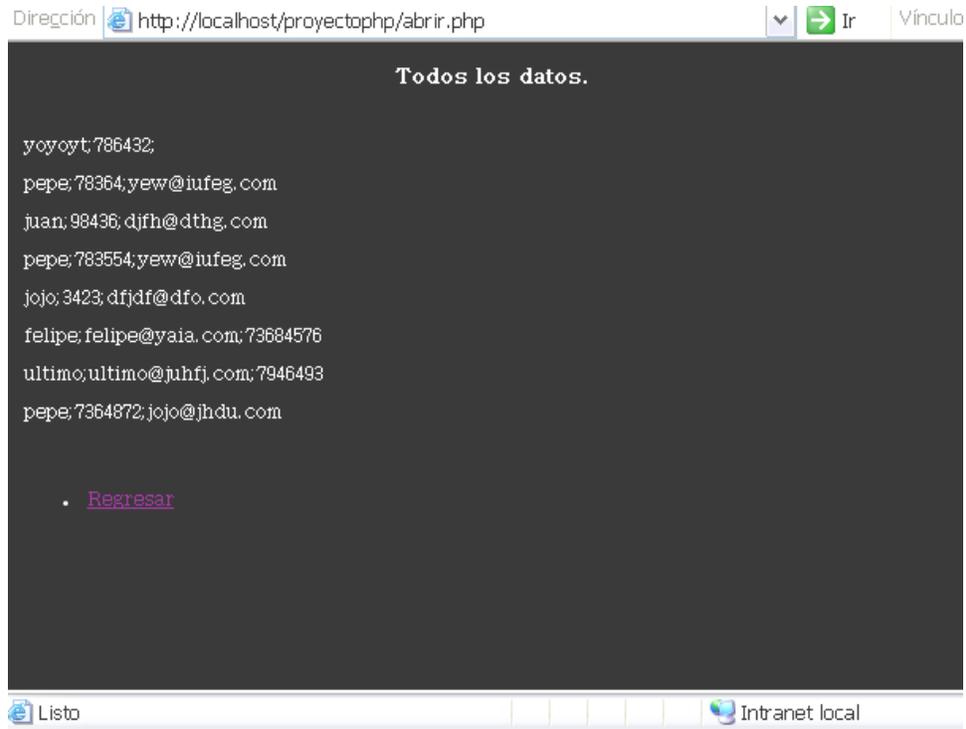
7 páginas tipo php que realizaran cada función del programa a pedir.

PANTALLA PRINCIPAL





PANTALLA ABRIR



PANTALLA BUSCAR



Dirección http://localhost/proyectophp/buscar.php Ir Vinculos >>

Buscar.

agendita

nombre:

[Regresar](#)

Listo Intranet local

PANTALLA GUARDAR

Dirección http://localhost/proyectophp/guardar.php Ir Vinculos >>

AÑADIR.

agendita

nombre:

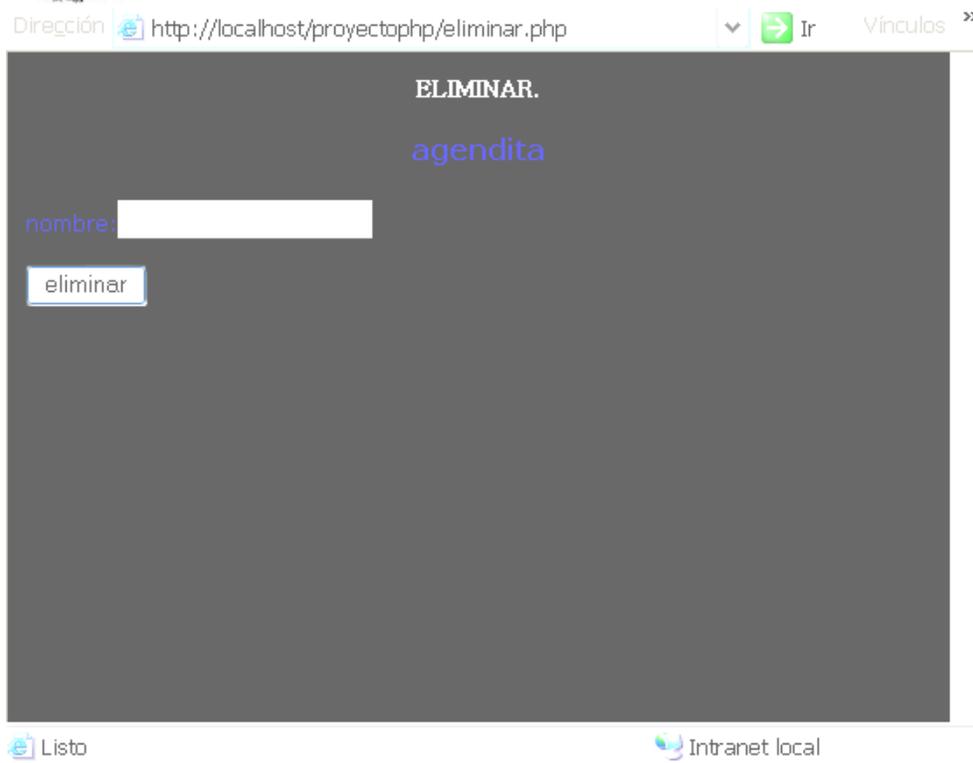
telefono:

correo:

[Regresar](#)

Listo Intranet local

PANTALLA ELIMINAR



En lo que respecta a programación con CGI's solo realiza lo básico con respecto a esta norma. Esto sirve en la relación que existe entre un servidor web y un programa, de tal que modo que este pueda interactuar con la Internet.

Unos datos importantes visto dentro de PHP fueron las funciones GET y POST (dentro de la norma CGI), explicando que con ellas se puede decidir si se desea visualizar los datos enviados a un formulario o no. Esto tiene el fin de darle una mayor seguridad pues eso evita que metan código al visualizar el acceso.

Algunas funciones útiles vistas fueron el Manejador de Correo. URL's y File-Uploads. Con ellos se pueden dar servicios comunes a nuestro servidor, ahorrando algunas tareas y tiempo en la programación de funciones independientes.

1.6 Interacción de WWW con Bases de Datos

En este modulo se desarrolla una aplicación de bases de datos que funcionaran a través de nuestro servidor en la WWW.

Principalmente en este modulo solo se ve en teoría como son los conceptos de la base de datos (datos, información y esquemas), siendo los datos caracteres que por si solo no tienen valor. La información se originara a partir de la suma de todos esos caracteres, y que toman un valor importante y los esquemas que va a ser el esquema de toda la base de datos.

Un DBMS o manejador de bases de datos por sus siglas en Ingles es aquel programa que junto con una colección de datos, se accede a los datos. El objetivo primordial de un DBMS es la de brindar la



información pronta y precisa, además de asegurar su completa confidencialidad. En este caso se usará el DBMS MySQL.

Los modelos son herramientas que ayudan a describir los datos, las relaciones entre ellos y su semántica. El modelo entidad-relación, es una manera de organización de datos más en forma con el mundo real, buscando una relación entre los datos, para evitar redundancias.

Los lenguajes son el punto de estudio en el que más nos enfocamos, ya que con estos comandos manejamos y administramos nuestra base de datos. Y a través de estos comandos, los implementaremos a una página del servidor.

La normalización, que ocuparemos en este caso, es una técnica que evita la aparición de anomalías en los datos así como otras cuestiones relacionadas con la administración de datos. Lo que se quiere evitar con eso es la redundancia de los datos, así como facilitar su control y actualización de datos, y generar una estructura fácilmente comprensible muy parecida a la situación que permitan sus datos y que permita su crecimiento.

Esta técnica de normalización consiste en transformar una tabla en varias fases. Su objetivo es eliminar redundancia de datos, facilidad en actualización de datos.

El ciclo de vida de una base de datos consta de seis fases: análisis, diseño, implementación, pruebas, puesta en marcha y mantenimiento.

En el análisis se entrevista a los accionistas y se examinan todos los sistemas existentes para identificar los problemas, las posibilidades y los límites. En esta fase se determinan los objetivos y el ámbito del nuevo sistema.

El diseño es la fase donde se crea el diseño conceptual a partir de las necesidades determinadas anteriormente. También se crea el diseño lógico para la implementación de la base de datos.

La implementación es la fase donde se instala el sistema de administración de la base de datos, se crea la base de datos y se cargan o importan los datos.

Las pruebas y la puesta en marcha son dependientes, porque como las pruebas entregan sus resultados, la forma normal de operación debe de generar información para los usuarios. Su mantenimiento requerirá de mantenerlo estable y actualizado, evolucionando a cada necesidad que se requiera.

Como Práctica final se desarrolló una aplicación en PHP que interactuara con una base de datos desarrollada en MySQL, en la cual se pudiera, anexar, borrar, buscar y actualizar datos. Parecido a la agenda realizada en PHP pero esta vez con interacción con una base de datos.

Antes de ver la práctica se tienen que contar con ciertos comandos.

A continuación se hace mención de algunos conceptos por el funcionamiento de mysql y sus comandos básicos que ayudan a generar la práctica con PHP.

mysql



Mysql es una base de datos cliente/servidor, lo cual quiere decir que tenemos un server (o daemon) que se ejecuta en segundo plano, a la escucha de las peticiones del programa cliente. En mysql, el daemon o servidor es **mysqld** y el cliente, **mysql**.

Ayudas:

En una sesión de DOS navegamos hacia `c:/mysql/bin` (o la ruta donde tengamos instalado mysql y escribimos

```
c:/mysql/bin:>mysqld --help
```

Para obtener la lista de comandos disponibles a `mysqld`. Lamentablemente verás que la información desborda la pantalla ; y que en DOS no hay barras de scroll vertical !. No hay problema. Un nuevo truco de DOS: el carácter *pipe*, es decir, la barra vertical, acompañada de la palabra *more*, así:

```
c:/mysql/bin:>mysqld --help|more
```

Verás que ahora la información se muestra pantalla a pantalla, pulsando la barra espaciadora para ir avanzando.



Arrancando el servidor

Nosotros utilizaremos `mysqld-opt`. `opt` no es una propiamente una opción de `mysqld` (por eso no te saldrá listada si has realizado la prueba anterior), sino que es un ejecutable distinto a `mysqld` (mas rápido, a costa de algunas opciones avanzadas) que es recomendado para los principiantes como nosotros.

```
c:/mysql/bin:>mysqld-opt
```

Bien, aunque la pantalla no muestra nada, ya tenemos el daemon `mysqld` corriendo en segundo plano. Para comprobarlo, puedes pulsar `ctrl-alt-delete` y veras `mysqld-opt` en la lista de tareas. O haz un ping:

```
C:/mysql/bin>mysqladmin ping
mysqld is alive
```

Utilizamos la herramienta de administración `mysqladmin` para hacer un ping, y el server nos contesta

```
mysqld is alive.
```

Para cerrar el servidor usamos de nuevo `mysqladmin`, sin olvidar nuestro nombre de usuario y password !!:

```
C:/mysql/bin>mysqladmin -u nombredeusuario -p shutdown
Enter password: *****
```

Recuerda que el nombre de usuario se introduce siempre precedido de `-u` y que si escribes a continuación `-p`, al pulsar enter, `mysql` te permitirá que teclees la contraseña.

Verificamos que el proceso `mysqld` ya no está corriendo (`ctrl-alt-supr` de nuevo) y ya sabemos encender y apagar `mysqld`.

Comandos

- `-p|--password [=pwd]`
La contraseña para conectar con el server. Si usamos `--p` sin proporcionar la contraseña, `mysql` nos la preguntará antes de ejecutar el comando.
- `-u|--user [=nombre de usuario]`
El nombre de usuario. Es necesario especificarlo si no hemos hecho aun log en el server
- `-h|--host=hostname`
Conecta al servidor. Si no lo especificamos, intentará `localhost`
- `-P|--port=pnum`
Especifica el puerto por el que conectara al server. Si no se especifica usará el puerto por defecto
- `-f|--force`
Mediante esta opción podemos hacer `drop` en una base de datos (eliminarla) sin que pida confirmación. Si usamos esta opción en una orden que incluya varios comandos, obligamos a la ejecución de todos ellos incluso aunque haya algun error.
- `-?|--help`
muestra la ayuda y sale de la ejecución



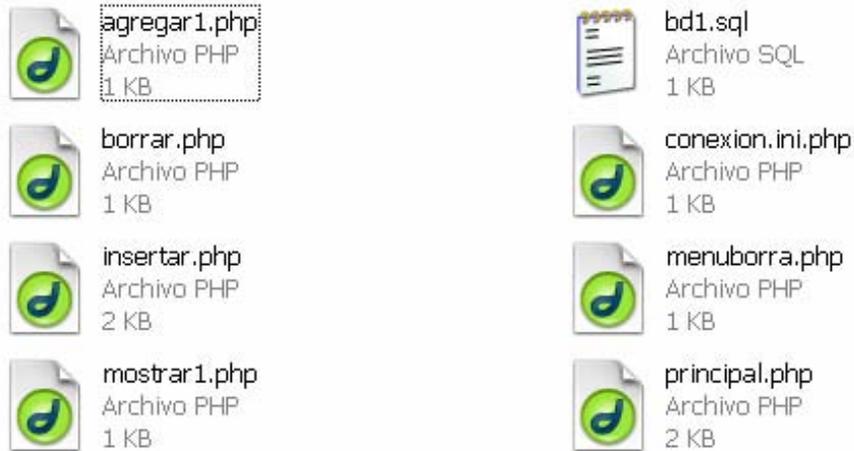
- -s|--silent
Cierra la ejecución si no puede conectar con el server
- -v|--verbose
Especifica que el server debe contestar con toda la información disponible a cada comando que ejecutemos
- -V|--version
Muestra la versión y sale
- -w|--wait [=num de intentos]
Reintenta la conexión si el server esta caido



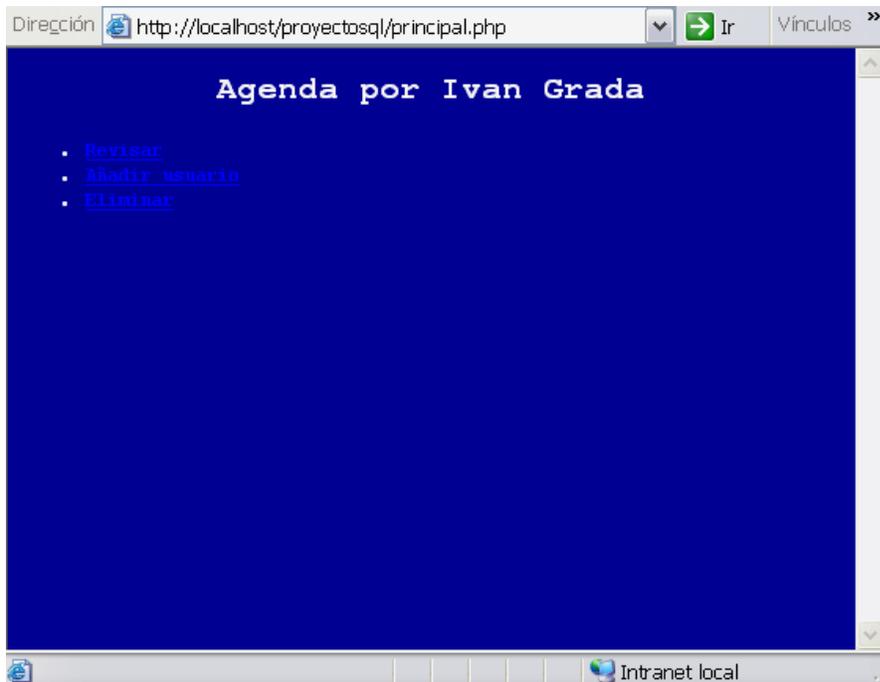
1.6.1 PRÁCTICA.

Con lo adquirido se logro hacer lo siguiente:

Se crearon instancias diferentes para cada una de las pantallas.



PANTALLA PRINCIPAL





PANTALLA REVISAR

Dirección http://localhost/sql/mostrar1.php Ir Vínculos

Conexión con la base de datos conseguida.

id	Nombre	Apellido	Telefono	Correo
1	Ernesto	Avila	56655663	eavila@basedatos
2	Jenny	Zaguri	58974000	jzaguri@basedatos
3	Adriana	Lopez	81233333	alopez@basedatos
4	Daniel	Ortega	42080000	dortega@basedatos
5	Raul	Uribe	92895005	ruribe@basedatos
6	Fernanda	Hlin	2124563000	fhlin@basedatos
7	Carmen	Maldonado	50124899	cmal@basedatos
8	Yoyo	Gonzales	65477788	correo@bd.com

[Regresar a inicio](#)

Listo Intranet local

PANTALLA AÑADIR USUARIOS



Dirección <http://localhost/sql/insertar.php> Ir Vinculos >>

Ejemplo de uso de bases de datos con PHP y MySQL

id:
Nombre:
Apellidos:
Telefono:
Correo:

Conexión con la base de datos conseguida.

id	Nombre	Apellido	Telefono	Correo
1	Ernesto	Avila	56655663	eavila@basedatos
2	Jenny	Zaguri	58974000	jzaguri@basedatos
3	Adriana	Lopez	81233333	alopez@basedatos
4	Daniel	Ortega	42080000	dortega@basedatos
5	Raul	Uribe	92895005	ruribe@basedatos
6	Fernanda	Hlin	2124563000	fhlin@basedatos
7	Carmen	Maldonado	50124899	cmal@basedatos
8	Yoyo	Gonzales	65477788	correo@bd.com

[Regresar a inicio](#)

Listo Intranet local

PANTALLA ELIMINAR

Dirección <http://localhost/proyectosql/menuborra.php> Ir Vinculos >>

Ejemplo de uso de bases de datos con PHP y MySQL

Conexión con la base de datos conseguida.

Nombre	Apellido	Telefono	Correo	Borrar
Ernesto	Avila	56655663	eavila@basedatos	Borrar
Jenny	Zaguri	58974000	jzaguri@basedatos	Borrar
Adriana	Lopez	81233333	alopez@basedatos	Borrar
Daniel	Ortega	42080000	dortega@basedatos	Borrar
Raul	Uribe	92895005	ruribe@basedatos	Borrar
Fernanda	Hlin	2124563000	fhlin@basedatos	Borrar
Carmen	Maldonado	50124899	cmal@basedatos	Borrar
Yoyo	Gonzales	65477788	correo@bd.com	Borrar

Listo Intranet local

1.7 Introducción a la seguridad en cómputo.



El primer punto observado fueron definiciones y métodos de seguridad existentes. Seguridad se define como un sistema que no contiene vulnerabilidades, no existe riesgo o amenaza, que funciona sin fallo, etc. Un servicio de seguridad y una política están vinculados, porque el servicio de seguridad debe satisfacer una política de seguridad. Una política de seguridad son las especificaciones de seguridad que una organización debe tener, observar y proveer con el fin de salvaguardar la información.

Según la norma ISO 17799 considera la información como un activo promoviendo: confidencialidad, y autenticación (algo que se sabe, algo que se tiene, algo que se es).

Se vio lo que es la criptografía, existiendo dos tipos. La asimétrica y la simétrica diferenciándolas, porque una ocupa un par de llaves para su cifrado y descifrado (simétrica). Y la otra requiere de un intercambio de llaves lo que provocaría un robo o extravío de la llave. Para poder marcar (autenticarlo) el paquete, existe el Message Digest que es el resultado de un algoritmo unidireccional que permite generar una huella digital, o la Firma digital que es cifrar el paquete con su propia llave. En el modulo se cifro con dos programas distintos, estos son GNU PG que el código libre y que fue creado por PHIL R Zimmermann, y OpenSSL (Open Secure Socket Layer) que igual es de código abierto y fue desarrollado por Eric A. Young y Tim J. Hudson.

Con esto ya bien definido se procede a generar un certificado digital para poder dar de alta el servidor. Explicando que la norma X.509 es el estándar para formatos de certificados con llave publica. Siendo este certificado avalado por un tercero. Siendo HTTPs un nuevo protocolo de comunicación que protege y da seguridad al cifrar toda información corriendo por Internet.

El segundo punto que se ve son los ataques y sus categorías. Un ataque es aquel que altera, elimina, o provoque un retraso en la información sin consentimiento del administrador directo de esa información. Existen dos tipos de ataques los activos y los pasivos. Los pasivos se caracterizan por ser menos dañinos, esto es que pueden solo recopilar información del servidor o de usuarios. Esto se hace a través de Whois que son los registros de información de una página, la cual es accesible a cualquier usuario de Internet y con esa información poder conseguir más. Los DNS o NIC, que son las tablas de ip y nombres de dominio y transferencias en la zona. Trazando la ruta destino. Los sniffers y analizadores de tráfico, que tienen casi las mismas funciones, olfatear la red y ver lo que pasa a través de ella. Los sniffers leen las tramas de redes (IP, IPX, etc). Existen dos tipos de sniffers, los activos y los pasivos. Los pasivos solo capturan paquetes, y los activos intentan apoderarse de las sesiones. Estos sniffers tiene la propiedad de ser casi invisibles, como no generan bitácoras (son pasivos), son difíciles de detectar.

Entre los ataques activos se encuentra los ya conocidos virus, caballos de troya, gusanos, spoofing, algunos sniffers, spam. No solo se puede atacar de esta manera, existen lo que es la ingeniería social, la cual consiste en obtener información importante a través del conocimiento de los Whois. Esto también conlleva a la falsificación y a la usurpación. La utilización de debilidades en el servidor (Backdoors, stack overflow, formatos de cadena, y bombas lógicas, entre otras) también son entradas abiertas a los hackers.

El escaneo de puertos es un arma indispensable a cualquier ataque, sabiendo los puertos abiertos sabremos por donde atacar. Esto se realiza utilizando el programa nmap.

Un ataque por fuerza bruta se trata de obtener información por cualquier medio y descifrar información. Una manera de atacar de forma directa es por medio de un diccionario de claves. Con



un programa apropiado se puede obtener passwords de interés, comparando el password original con una lista diccionario de posibles claves.

Otro ataque es el secuestro de sesiones. Su función es robar el control de una sesión, suplantando al usuario y engañar al servidor haciéndonos pasar por el usuario verdadero.

Un ataque mas que para obtener información es para dejarlo fuera de servicio, es el Denial of Service. Este consiste en infectar a maquinas inocentes y hacer que estas infecten al servidor mandándole ataques Dos (Ping de la muerte, Inundación Sync, spoofing, smurt, Fraggle, entre otros). Un ping de la muerte lanza un paquete que excede los 65535 bytes provocando un reinicio. Inundación del Sync, como su nombre lo dice inunda un servidor enviando puros sync en lugar de ack's, sobrepasando el numero de sync permitidos y derriba el servidor. Un smufing es un ataque de los mas temidos. Requiere de tres "personajes" para que funcione: una victima, un atacante, y una red amplificadora. El atacante originara un paquete ICMP hacia el broadcast amplificador de red, haciendo parecer que su origen es una interfaz de red de la victima. Cada interfaz del broadcast enviara respuestas a la supuesta interfaz, originando un trafico que la victima no soportara.

Para poder defendernos existen los Firewalls, de estos existen dos tipos los fisicos y los lógicos. Los fisicos son usualmente ruteadores, o switch de capa 3. un lógico son programas que diferencia entre paquetes nocivos y paquetes permitidos.

Entre los firewalls existen 4 tipos:

- Packet filters
- Circuit Level gateways
- Application level gateways
- Stateful multilayer inspection firewall

En los firewalls se debe implementar según su nivel requerido, y el tráfico que vaya a entrar y salir. En linux los firewalls existentes son las iptables y los netfilters. Que son completamente configurables desde su naturaleza. Otra forma de protección son los proxy servers, que son los intermediarios entre la red interna e Internet.

Como evaluación se realiza un examen que consiste en un examen teórico-práctico y un proyecto en casa que consistía en implementar un formulario con codificación de texto almacenado en una base de datos con PHP y MySQL.

1.8 Programación con Java.

OBJETIVO:

El participante identificará los principales componentes del lenguaje java, que le permitirán ir incorporando los conocimientos adquiridos sobre el lenguaje.

1.8.1 LENGUAJE DE PROGRAMACIÓN



Java es un lenguaje sencillo y orientado a objetos que nos permiten desarrollar aplicaciones en muy diversas áreas tales como: seguridad, aplicaciones clientes-servidor, interfaces gráficas, páginas Web interactivas, entre otras.

Una de sus principales características es la creación de módulos reutilizables que funcionan sin necesidad de conocer su estructura interna, permitiendo al usuario añadir nuevos módulos. Así como la obtención de programas independientes de la plataforma en la cual fueron desarrollados.

Hay que mencionar algunas definiciones para poder entender como se programa con java.

Clase: Es un molde que nos sirve para definir los atributos y el comportamiento de los objetos.

Polimorfismo: Es cuando un método es invocado pero depende de cómo se usa para cada uno.

También hay que tomar en cuenta las diferentes versiones que existen de java, como los son:

Java 2 standar edition.

Java 2 Enterprise edition.

Java 2 Micro (usado para palms o celulares).

Además, en una instrucción simple pueden aparecer operadores, los hay de dos tipos: los que actúan sobre un operador, o los que lo hacen sobre dos. Los operadores los clasificaremos por su empleo:

Operadores aritméticos

Operador	Uso	Descripción
+	op1 + op2	Suma op1 y op2 (*)
-	op1 - op2	Resta op2 de op1
*	op1 * op2	Multiplica op1 y op2
/	op1 / op2	Divide op1 por op2
%	op1 % op2	Obtiene el resto de dividir op1 por op2
++	op ++	Incrementa op en 1; evalúa el valor antes de incrementar
++	++ op	Incrementa op en 1; evalúa el valor después de incrementar
--	op --	Decrementa op en 1; evalúa el valor antes de decrementar
--	-- op	Decrementa op en 1; evalúa el valor después de decrementar



Operadores relacionales:

Operador	Uso	Devuelve true si
>	op1 > op2	op1 es mayor que op2
>=	op1 >= op2	op1 es mayor o igual que op2
<	op1 < op2	op1 es menor que op2
<=	op1 <= op2	op1 es menor o igual que op2
==	op1 == op2	op1 y op2 son iguales
!=	op1 != op2	op1 y op2 son distintos

Operadores lógicos:

Operador	Uso	Devuelve true si
&&	op1 && op2	op1 y op2 son verdaderos
	op1 op2	uno de los dos es verdadero
!	! op	op es falso (niega op)

Nota: Los operadores lógicos siempre devuelven un valor booleano.

Operadores a nivel de bit:

Operador	Uso	Descripción
>>	op1 >> op2	desplaza a la derecha op2 bits de op1
<<	op1 << op2	desplaza a la izquierda op2 bits de op1
>>>	op1 >>> op2	desplaza a la derecha op2 bits de op1 (sin signo)
&	op1 & op2	operación and
	op1 op2	operación or
^	op1 ^ op2	operación xor
~	~ op	operación complemento a 1

Los operadores a nivel a bit toman los operadores, los transforman a binario y realizan las operaciones trabajando con los bits uno a uno.

Sintaxis del lenguaje

El lenguaje se basa en pensar que hay en el mundo real objetos y esos objetos tienen un tipo, o clase. Por ello el lenguaje se basa en clases, que describen como son los objetos. Por ejemplo, el lenguaje tiene una clase que describe ficheros, una que describe cadenas de texto, o bien nosotros podemos crear clases, como por ejemplo la clase Persona, que describe los datos que interesan de una persona.

Por ello siempre para comenzar a trabajar con un programa java hay que crear una clase:

```
public class TablaMultiplicar{  
  
}
```

Además se deben de cumplir las siguientes características:



- La clase se debe de llamar exactamente igual que el fichero que la contiene.
- La clase que se llama igual que el fichero debe de estar precedida de la palabra public.
- Cuando se intenta ejecutar una clase java, la máquina virtual lo que hace es llamar a un método especial llamado main que debe estar dentro de la clase a ejecutar:

```
public class TablaMultiplicar{  
public static void main(String arg[]){  
}  
}
```

Y es dentro de la función main donde escribiremos el código que queremos que se ejecute:

```
public class TablaMultiplicar{  
public static void main(String arg[]){  
int numero = Integer.parseInt(arg[0]);  
for(int i = 1 ; i<=10 ; i++){  
System.out.println(""+numero+" * "+i+" = "+(i*numero));  
}  
}
```

Objetos y clases

Java es un lenguaje totalmente orientado a objetos, esto representa que posee instrucciones y sintaxis específicas para la programación orientada a objetos. Además, en java existe el concepto de jerarquía de herencia, es decir, que todas las clases deben de heredar de otra clase para formar todas ellas un árbol invertido. La clase raíz de este árbol es la clase java.lang.Object y todas las clases heredan de ella directa o indirectamente. Adicionalmente las clases son colocadas en carpetas para facilitar su ordenación y el trabajo con ellas, dichas carpetas (paquetes es el término más exacto) formarán parte del nombre de la propia clase, por ejemplo, java.lang.Object significa que existe una carpeta llamada lang, dentro de la cuál existe otra carpeta llamada lang, dentro de la cuál existe una clase llamada Object.



Estructura de un fichero java

Antes de ver la estructura de un fichero, veamos primero como se realizan comentarios que no se compilan, en este lenguaje hay dos tipos: hasta fin de línea (precedido por //) y el comentario de

Definición de una clase

Aquí tenemos la sintaxis para definir una clase:

```
modifAcceso modifClase class nombreClase [extends nombreBase] [implements listaInterfaces]
```

```
{  
Atributo 1  
Atributo N  
método 1  
método N  
}
```

Donde **nombreClase** es el nombre de la clase, cualquier nombre, pero respetando las reglas de nomenclatura del lenguaje.

ModifAcceso puede ser uno de los siguientes valores:

Public: indica que la clase es pública, y por tanto que puede ser utilizada desde cualquier otra clase, con independencia de si están en el mismo paquete o no.

Sin especificar: indica que la clase tiene visibilidad de paquete, es decir, sólo la pueden usar las clases que se encuentren en el mismo paquete que dicha clase.

ModifClase indica características específicas de la clase que estamos construyendo, los posibles valores son:

Abstract: indica que a la clase le falta, al menos uno, el código de algún método. Posee el método (abstracto), pero no tiene el código de ese método, siendo responsabilidad de las clases derivadas proporcionar el código de dicha clase. Una clase abstracta no se puede instanciar.

Final: se emplea para evitar que esta clase pueda ser derivada.

Extends se utiliza para indicar que la clase hereda de nombreBase, en java sólo se permite heredar de una única clase base. En caso de no incluir la cláusula extends, se asumirá que se está heredando directamente de la clase java.lang.Object

Implements indica que esta clase es de los tipos de interfaz indicados por listaInterfaces, pudiendo existir tantos como queramos separados por comas. Esta cláusula es opcional.



Definición de atributos de una clase

Es decir, en java las variables de tipo básico son el nombre de una zona de memoria en la cuál podemos almacenar valores, pero que en cambio, las variables de tipo objeto son en realidad referencias (punteros o alias) de objetos.

Una variable de tipo objeto no es un objeto completo, sino tan solo almacena la situación del objeto en la memoria del equipo. Esto es muy similar a lo que ocurre con las casas y las direcciones de dichas casas: la dirección calle Alcalá 950 es una dirección válida, pero no podemos mandar cartas a dicha dirección.

Lo mismo sucede con los objetos, podemos tener una variable para referirnos a objetos, pero la variable puede que no apunte a ningún objeto y por tanto no la puedo emplear para intentar acceder a un método o a un atributo del objeto referenciado por la variable, sencillamente porque no existe el objeto referenciado.

Una variable que no apunta a un objeto se asume que tiene un valor especial llamado null, e incluso podemos asignar el valor null a la variable:

```
Thread t = null;
```

Es por ello que se deben construir objetos y asignárselos a las referencias, usando la palabra clave new. new permite crear un objeto a partir de la descripción de la clase que le pasamos como argumento, por ejemplo:

```
new Persona()
```

Conseguimos crear un objeto de la clase Persona, los paréntesis permiten especificar qué constructor estamos llamando al crear el objeto (veremos constructores más adelante).

Pero al crear un objeto persona como en el código anterior lo estamos creando como un objeto anónimo, es decir sin asignar el objeto a una variable de tipo referencia, desde la cuál poder referirnos al objeto y poder llamar a sus métodos y atributo, por ello lo más habitual será asignar el objeto a una variable como en:

```
Persona p = new Persona();
```

y así poder acceder al objeto Persona recién creado:

```
p.nombre = "Alberto";
```

Definición de métodos de una clase

Para definir los métodos se emplea la siguiente sintaxis:

```
[modifVisibilidad] [modifFunción] tipo nombreFunción (listaParámetros) [throws listaExcepciones]  
{  
}
```



Para **modifVisibilidad** se aplica las mismas normas que para atributos:

public: indica que es un método accesible a través de una instancia del objeto.

private: indica que a través de una instancia no es accesible el método. Al heredar el método se convierte en inaccesible.

protected: indica que a través de una instancia no es accesible el método. Al heredar si se puede usar desde la clase derivada.

Sin especificar: indica visibilidad de paquete, se puede acceder a través de una instancia, pero sólo de clases que se encuentren en el mismo paquete.

nombreFunc debe de ser un identificador válido en el lenguaje.

tipo es el tipo del valor devuelto por la función, pudiendo ser:

Un tipo básico.

Un objeto de una clase o interfaz. En este tipo de objetos se incluyen las matrices o vectores.

void, en el caso de no devolver ningún valor.

Instancias

Podemos interpretar que una clase es el plano que describe como es un objeto de la clase, por tanto podemos entender que a partir de la clase podemos fabricar objetos. A ese objeto construido se le denomina **instancia**, y al proceso de construir un objeto se le llama **instanciación**.

Cuando se construye un objeto es necesario dar un valor inicial a sus atributos, es por ello que existe un método especial en cada clase, llamado **constructor**, que es ejecutado de forma automática cada vez que es instanciada una variable. Generalmente el constructor se llama igual que la clase y no devuelve ningún valor. Análogamente, **destructor** es un método perteneciente a una clase que es ejecutado de forma automática cuando un objeto es destruido. **Java** no soporta los destructores. Es posible que exista más de un constructor en una clase, diferenciados sólo en los parámetros que recibe, pero en la instanciación sólo será utilizado uno de los constructores.

Es recomendable emplear el constructor para inicializar las variables internas del objeto, o para obtener recursos, mientras que el destructor se suele emplear para liberar esos recursos obtenidos en el constructor.

Imaginemos ahora un televisor, nosotros que poseemos el mando a distancia podemos manejar el volumen del televisor, pero no nos haría gracia que otra persona pudiese manejar el volumen a su antojo. Lo mismo sucede con los objetos: un objeto que no quiere que otro objeto llame a un método o acceda a un atributo, debe definir a dichos atributos y métodos como privados. En cambio si los definimos como públicos, cualquier objeto podrá utilizarlos.



Pensemos ahora en un coche deportivo y en un coche utilitario, ambos son coches, pero hacemos la distinción deportivo y utilitario porque son diferencias lo suficientemente importantes como para ser tenidas en cuenta, pero no tanto, como para pensar que uno es un coche, pero el otro no. Esta situación se interpreta pensando en que existen dos clases: Deportivo y Utilitario, pero que ambas, además de sus propias características (atributos y métodos), poseen las de la clase coche, esto es la herencia. Denominaremos clase base a la clase de la cuál heredamos y derivada a la clase que hereda de la clase base.

Herencias

Existen dos tipos de **herencia sencilla y múltiple**. Sencilla significa que sólo heredamos de una clase base, mientras que múltiple indica que tenemos varias clases base (por ejemplo un hidroavión hereda de barco y de avión). Java sólo soporta herencia simple.

Al utilizar la herencia aparecen dos conceptos: **super y this**, **this** representa al objeto completo, en cambio la herencia **super**, sólo representa la parte heredada de la clase base.

Cuando se hereda nos encontramos frente a un pequeño problema: ¿Qué sucede cuando se hereda un método de la clase base, el cuál estamos redefiniendo en la clase derivada? Esto es un caso de sobrescritura de métodos. La solución es simple, cuando estemos ejecutando el método de un objeto derivado se llamará al método de su propia clase, es decir el redefinido. Si lo que se quiere es emplear el método de la clase base, hay que emplear una técnica que consiste en usar: `super.método()`.

Los constructores no son heredados, pero sí llamados. Es decir, cuando se construye un objeto de la clase derivada se llama al constructor de la clase derivada, pero antes de comenzar a ejecutarse se llama al constructor de la clase base, que tras ejecutarse continua la ejecución del constructor de la clase derivada.

Se puede elegir qué constructor de la clase base es llamado, generalmente llamando al método `super()`, que representa al constructor de la clase base, pero al pasar parámetros distintos, seleccionamos qué constructor de la clase base queremos llamar.

Supongamos ahora que tenemos algo que es capaz de encenderse, de apagarse, de iniciar una reproducción, de parar una reproducción, sin duda todos pensamos en un reproductor, pero por esa descripción encajan objetos como reproductor de cassette, reproductor de CD, el vídeo,...

Llamaremos a la descripción **interfaz**, y los objetos que cumplen ese interfaz (es decir, tienen todas las funciones que definen el interfaz) diremos que **implementan el interfaz**. Pensemos ahora en el hecho que una persona que sabe iniciar una reproducción de un CD también sabe iniciar una reproducción de vídeo, en ambos casos debe de encender el objeto, iniciar la reproducción, parar la reproducción y apagar el reproductor. Eso significa que para la persona es transparente el tipo (clase) real del objeto reproductor que posea, ya que la persona sabe que puede ponerlo en marcha, apagarlo, sin necesidad de conocer la clase real, tan sólo debe de saber que es un objeto del tipo reproductor.



Polimorfismo

Este hecho de que un objeto pertenezca a una clase, pero que pueda cumplir uno o más interfaces es muy similar al hecho de que una persona (un objeto) es hija (un interfaz), posiblemente tía (otro interfaz) o incluso abuela (otro interfaz). Con esto lo que se indica es que un interfaz sólo tiene interés para quien lo necesite usar, es decir, el interfaz tío sólo lo necesitan los objetos sobrinos, no un objeto de tipo policía.

Esta característica de que varios objetos de distintas clases puedan recibir el mismo mensaje y ser capaces de responderlo, es conocida como **polimorfismo**.

Interfaces

Un **interfaz** es una lista de acciones que puede llevar a cabo un determinado objeto. Sorpresa, ¿eso no eran los métodos que se definen en una clase? Casi, en una clase además de aparecer los métodos aparecía el código para dichos métodos, en cambio en un interfaz sólo existe el prototipo de una función, no su código.

Veámoslo con un ejemplo: Pensemos en un interfaz en el que en su lista de métodos aparecen los métodos despegar, aterrizar, servirComida y volar. Todos pensamos en un avión, ¿verdad? El motivo es sencillamente que avión es el concepto que engloba las acciones que hemos detallado antes, a pesar que existen muchos objetos avión diferentes entre sí, por ejemplo Boeing 747, Boeing 737, MacDonell-Douglas.

Lo realmente interesante es que todos ellos, a pesar de pertenecer a clases distintas, poseen el interfaz avión, es decir poseen los métodos detallados en la lista del interfaz avión.

Esto significa también que a cualquier avión le podemos pedir que vuele, sin importarnos a que clase real pertenezca el avión, evidentemente cada clase especificará como volará el avión (porque proporciona el código de la función volar).

En java un interfaz define la lista de métodos, pero para que una clase posea un interfaz hay que indicar explícitamente que lo implementa mediante la cláusula implements. Pero veamos primero la estructura de un interfaz:

```
[modif.visibilidad] interface nombreInterfaz [extends listaInterfaces]
```

```
{  
prototipo método1;  
.....  
prototipo método1;  
}
```

Donde modif.visibilidad puede ser public o bien sin especificar, es decir visibilidad pública (desde cualquier clase se puede emplear el interfaz) o de paquete (sólo se puede emplear desde clases del mismo paquete).



NombreInterfaz por convenio, sigue las mismas reglas de nomenclatura que las clases, y en muchos casos acaba en `able` (que podíamos traducir como: 'ser capaz de').

La cláusula opcional `extends`, se emplea para conseguir que un interfaz herede las funciones de otro/s interfaces, simplemente lista Interfaces es una lista separada por coma de interfaces de los que se desea heredar.

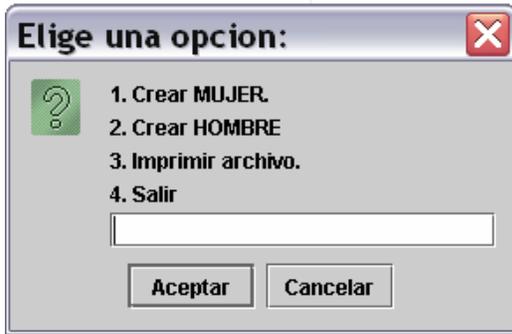
En muchas ocasiones un interfaz es empleado para definir un comportamiento, que posteriormente será implementado por diversas clases, que podrán no tener nada que ver entre ellas, pero que todas se comportarán igual de cara al interfaz. Es decir, todas tendrán las funciones indicadas por el interfaz.

Cuando varios objetos de distintas clases pueden responder al mismo mensaje (función), aún realizando cosas distintas se denomina **polimorfismo**.

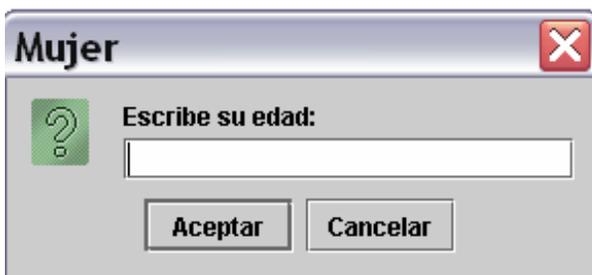
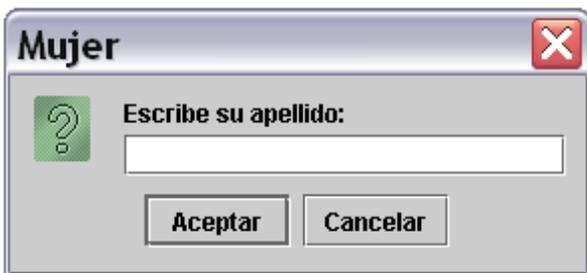
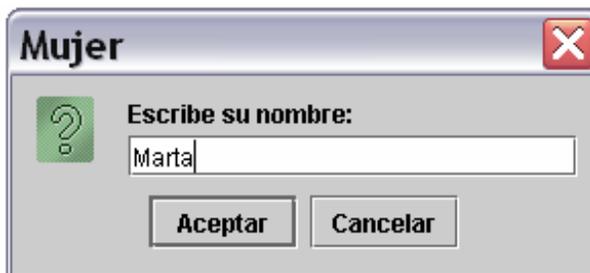


1.8.2 Práctica

Para esta parte, se procede con una práctica en al cual se exponga los condimentos adquiridos, en este caso se realiza una base de datos en al cual se pude mostrar algunas de las herramientas mas usadas en la programación de JAVA.



Aquí se trata de realizar una base para hombres y otra para mujeres.

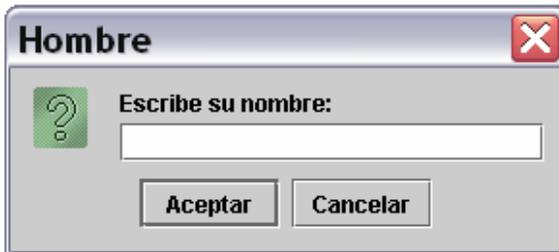




Al finalizar manda un mensaje como este.



Este proceso es igual a la de de opción hombre.





CAPITULO 2

PROYECTO DE SISTEMA DE FACTURACIÓN A TRAVÉS DE LA WEB CON HERRAMIENTAS LIBRES.

Este sistema de facturación permitirá realizar facturas oficialmente, con ello se podrá reducir ampliamente los lugares donde se archivan estos documentos, además de que pondrán ser enviados inmediatamente por vía Web, lo que aras mas rápido las entregas de facturas sin demoro o perdida.

El sistema se encuentra dividido en la siguiente forma

1. El administrador.
2. El facturador.
3. El consultor

Al ingresar al sistema de FACTURA ELECTRÓNICA muestra la pantalla para acceso, ya sea para cualquiera usurario dado de alta (imagen 1.1).

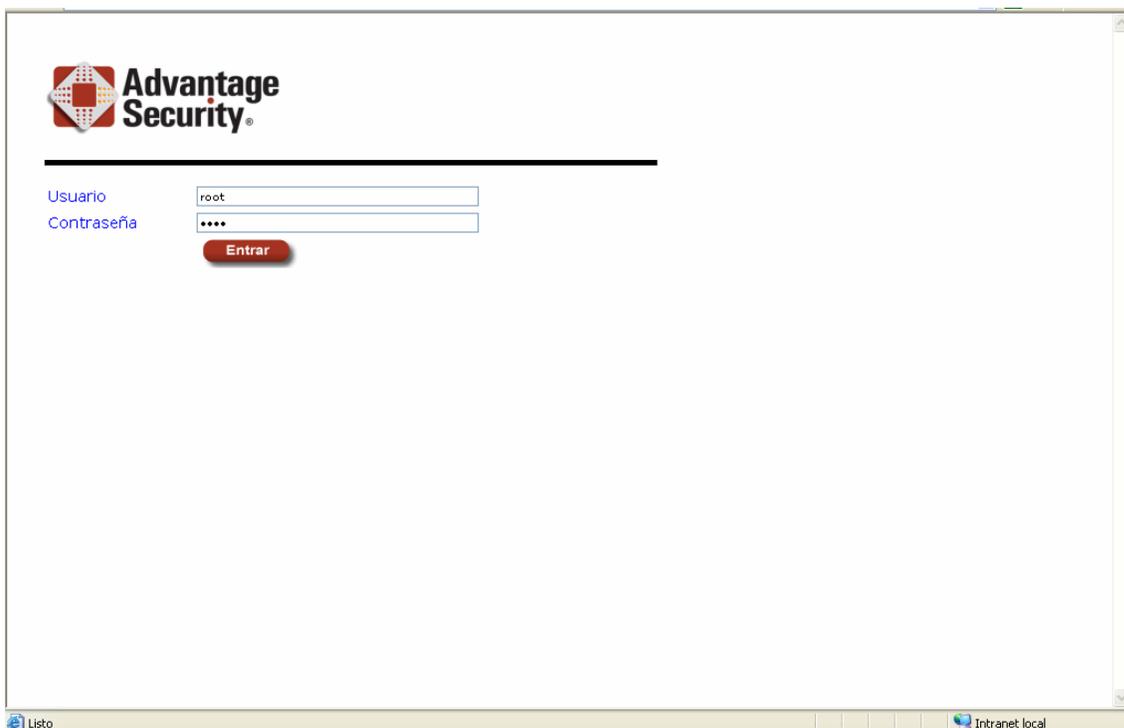


Imagen 1.1



Como se puede apreciar en la imagen 1.2 se tiene que ingresar el usuario y dar clic en el botón entrar; de ahí en adelante se tendrán funciones independientes dependiendo de los privilegios que se tenga.

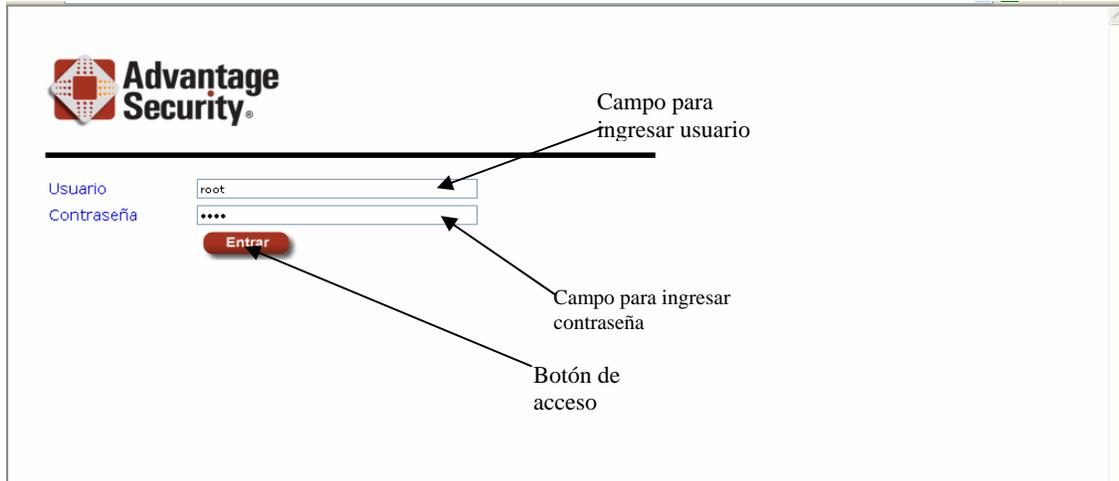


Imagen 1.2

Usuario administrativo

Se le conoce así al usuario que tendrá acceso para agregar, modificar o eliminar tanto usuarios como empresas.

Al acceder como usuario administrativo nos aparece una pantalla como la imagen 2.1

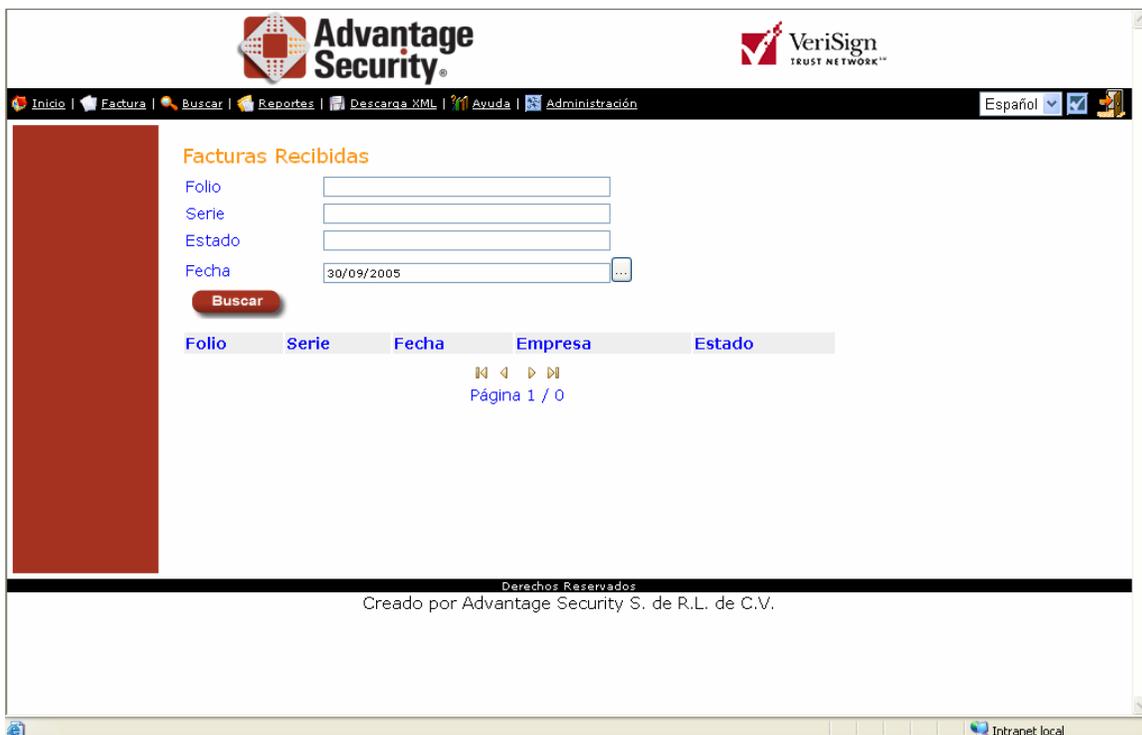


Imagen 2.1.

En esta imagen se puede apreciar las diferentes herramientas.



1. Inicio.
2. Factura.
3. Buscar.
4. Reportes.
5. Descarga xml
6. Administración

MENÚ INICIO.

El menú de inicio que ofrece el sistema contiene un acceso fácil a **facturas recibidas** las cuales serán aprobadas o canceladas según sea el caso, la búsqueda puede ser por folio, serie, estado o fecha.

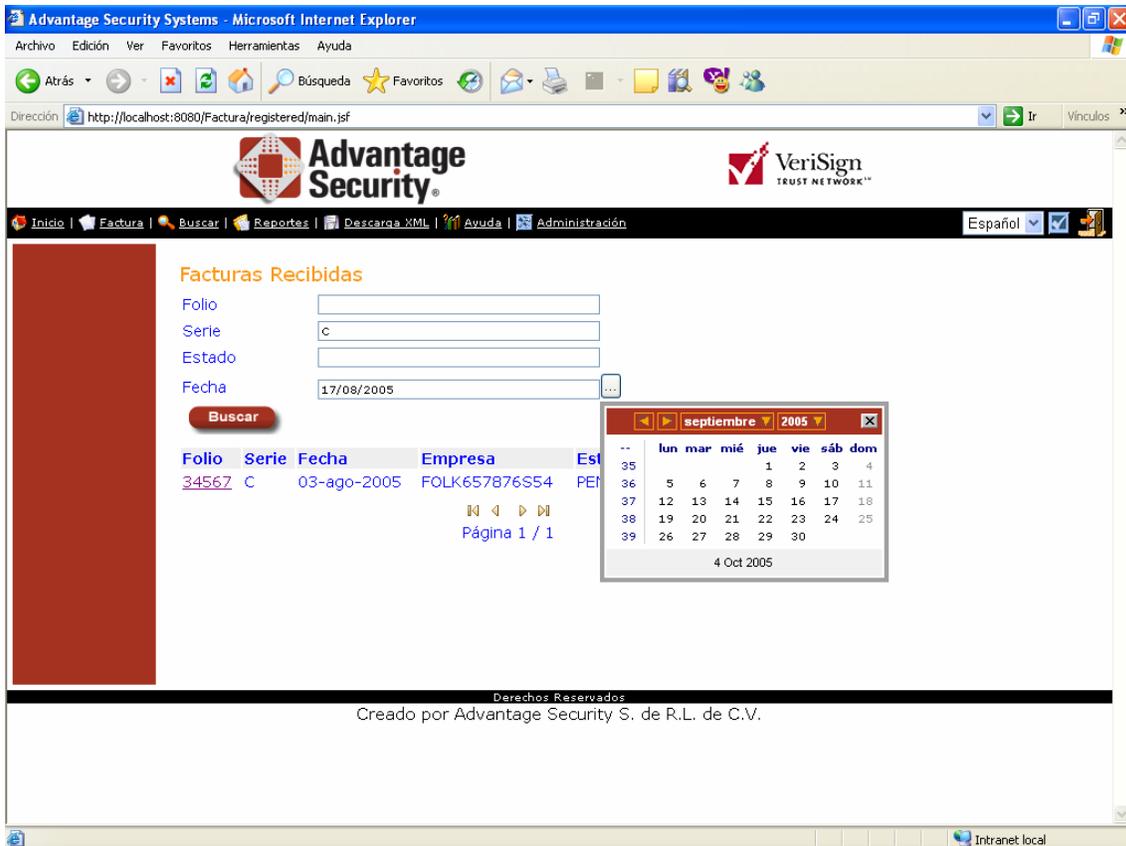


Imagen 2.2

Si ya se cuentan con algunas facturas por revisar, el sistema nos proporciona información general de ellas (véase la imagen 2.3) como:

- Folio: Numero de folio que se la asignado
- Serie: La serie de la factura.
- Fecha: Fecha en el cual se realizo la factura.



- Empresa: Nombre de la empresa que solicita.
- Estado: El estatus que se tiene de la factura mostrada



Imagen 2.2

Para poder ver más detallada la información de una factura podemos dar clic en vínculo de folio.

Folio **Serie**
[34567](#) **C**

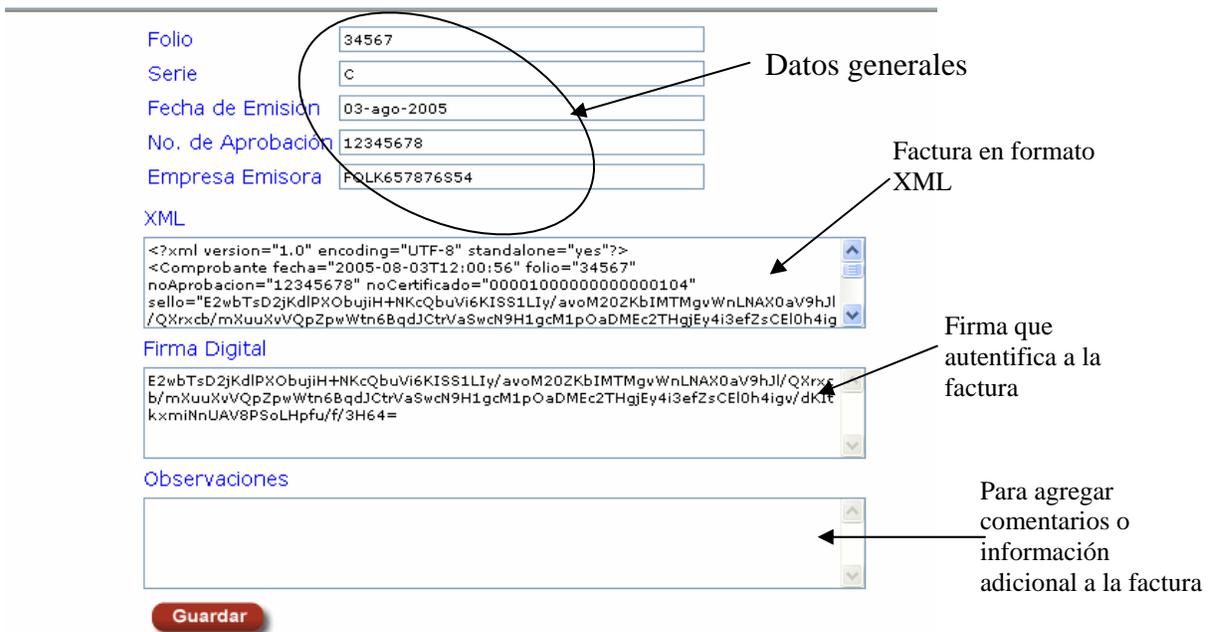


Imagen 2.3

MENÚ FACTURA.



En este menú se pueden importar facturas que se tengan o que se deseen investigar, agregándolas a la base de datos del sistema.(Imagen 2.4)

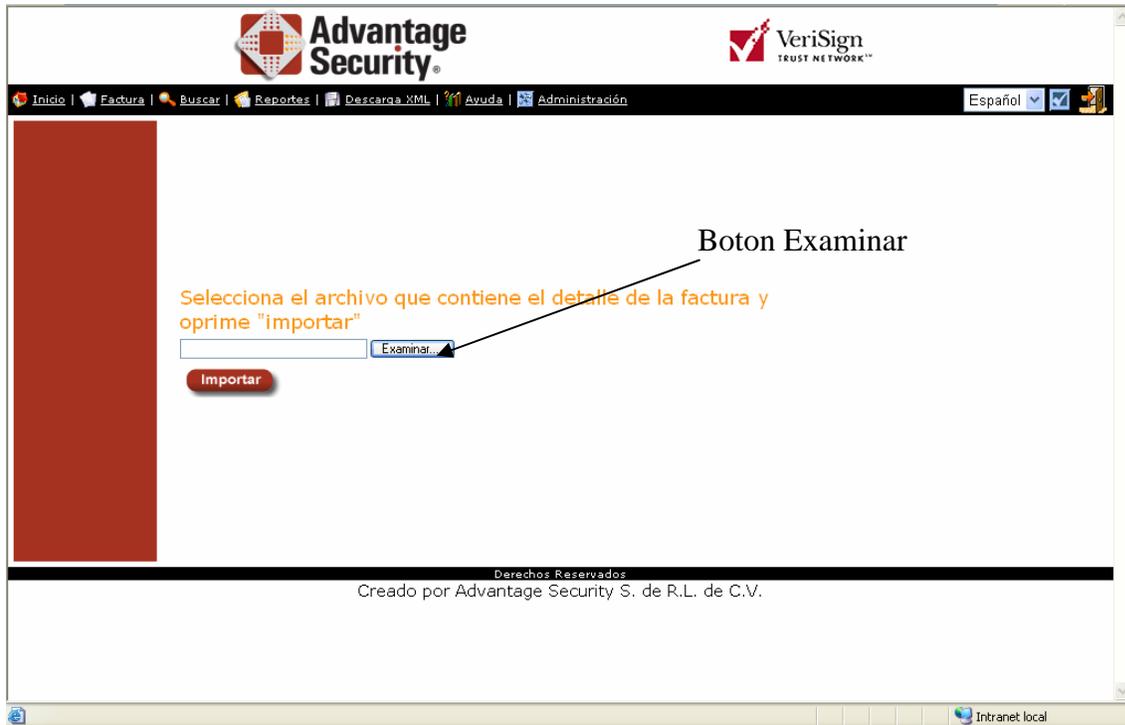


Imagen 2.4

Para esto, contiene un botón llamado EXAMINAR en el cual podremos hacer la búsqueda ya sea en el disco duro o en alguna unidad externa a la PC. Como aparece en la imagen 2.5. Este archivo debe de ser con extensión .txt o .xml.



Imagen 2.7

Al confirmar, nos pedirá el certificado emitido por el SAT y su contraseña que sellara a las facturas que se emitan. Después dar clic al botón continuar.



Imagen 2.8



Imagen 2.9

Con esto se finaliza el proceso de importación, se recomienda que después de hacerlo resguardar la factura en el menú de **DESCARGA**.



NOTA

Hay que tener en cuenta que si se importa una factura que ya se encuentra en el sistema, mandar un error, mencionando que la factura ya se encuentra. (Vea el ejemplo de la imagen 2.10.)

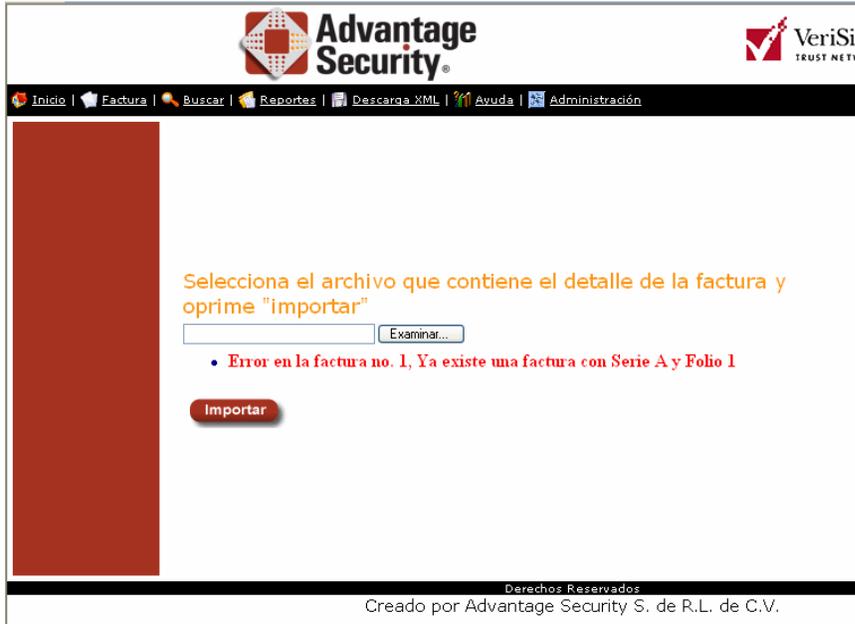


Imagen 2.10

MENÚ BUSCAR

En este menú, se puede hacer búsquedas de facturas que se han generado por el sistema.

La forma de búsqueda es similar a una búsqueda de facturas recibidas (Ya sea por el número de serie, fecha generada, folio o empresa).

De igual manera, al obtener la factura deseada, podremos acceder a ella dando clic a cualquiera de las características que aparecen, como se muestra en la imagen 2.11

Folio	Serie	Fecha	Empresa	
<u>1</u>	A	03-oct-2005	FOLK657876S54	<u>Cancelar</u>

Imagen 2.11

La diferencia a comparación de de la búsqueda de facturas recibidas, es de que este ya no trae el campo ACEPTAR. Solo contiene el campo cancelar.



MENÚ REPORTES



Este menú como su nombre lo dice sirve para generar reportes en la factura que se requiera. Para esto debe de buscar el mes y el año en que se realizó la factura, al seleccionar esto nos mandara todas las facturas y solo se tendrá que escoger la indicada para realizar dicho reporte.

Si hay un gran numero de facturas en el mes seleccionado, nos pedirá que bajemos el reporte para poder hacer lo mencionado anteriormente.



Imagen 2.12

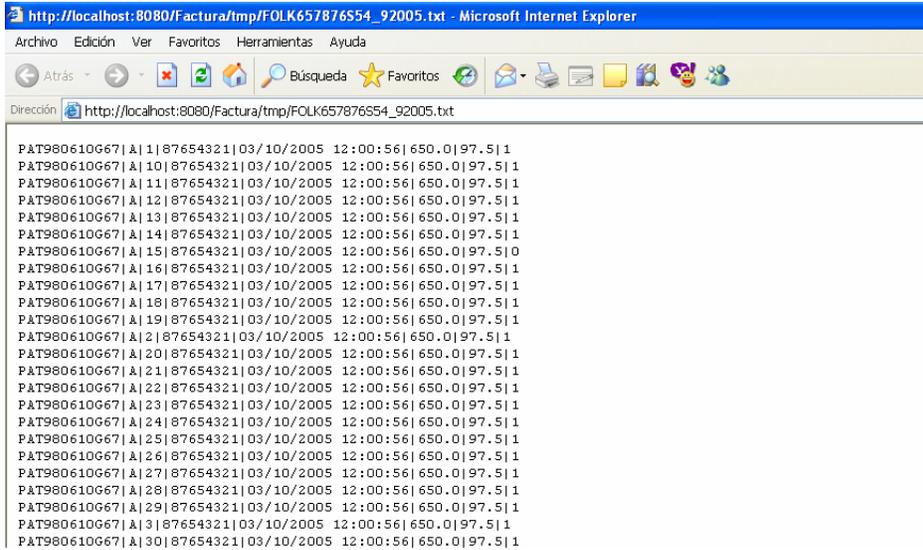


Imagen 2.13

MENÚ DESCARGA XML



En este menú se generan los respaldos de las facturas generadas por el sistema en formato XML.

La forma de realizar es seleccionar el mes y año en el que se requiere el respaldo, dar clic en el botón GENERAR.

Al terminar nos mandan un el reporte en la pantalla de que se realizo el archivo.



Imagen 2.14



Preguntar que tipo de certificado es el que se tiene que poner

MENÚ ADMINISTRACIÓN



Este menú solo es habilitado cuando se acceda al sistema como administrador. Este menú tiene la característica de agregar al sistema usuarios y empresas.



Imagen 2.15

Empresas

En la opción de empresa, su puede realizar la búsqueda ya sea por el **RFC** o por el **Nombre Fiscal**, si existe dicha empresa la desplegara por el RFC, el Nombre fiscal y la opción de eliminar (imagen 2.15).

Si se quiere agregar una nueva empresa dar clic en el botón nuevo, el cual nos manada una pantalla con los datos a llenar (imagen 2.16).



ps || [Salir Administración](#)

Rfc

Nombre

Calle

Código Postal

Colonia

Estado

Localidad

Municipio

No. Exterior

No. Interior

País

Referencia

Guardar

Imagen 2.16

Usuario facturador

Este usuario tiene algunos privilegios como el realizar la firma en el documento de alguien que colcita alguna factura.

Contiene la herramienta buscar que funciona de la misma manera como la que se encuentra el usuario administrativo. (Imagen 3.1)

[Inicio](#) | [Buscar](#) | [Subir Factura](#) | [Ayuda](#)

Facturas Recibidas

Folio	Serie	Fecha	Empresa	Estado	
34567	C	03-ago-2005	FOLK657876S54	APROBADA	Notificar

⏪ ⏩

Página 1 / 1

Imagen 3.1

La diferencia es que en este tipo de usuario se encuentra el menú SUBIR FACTURA.



MENÚ SUBIR FACTURA



Si el facturador requiere agregar una factura para ser sellada, usara la opción SUBIR FACTURA. Para esto debe de tener a disposición la factura en formato '.XML' y su certificado personal. (Imagen 3.2)

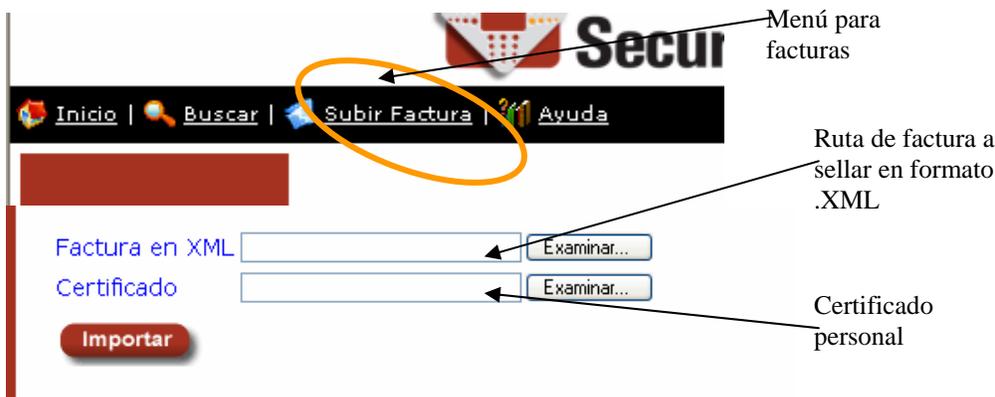


Imagen 3.2

Usuario consultor

Este tipo de usuario tiene los mínimos privilegios para el sistema ya que el solo puede consultar el tipo de factura y checar si ya ha sido aprobada, al igual puede ver las observaciones en cada factura que el requiera ver. (Imagen 4.1)

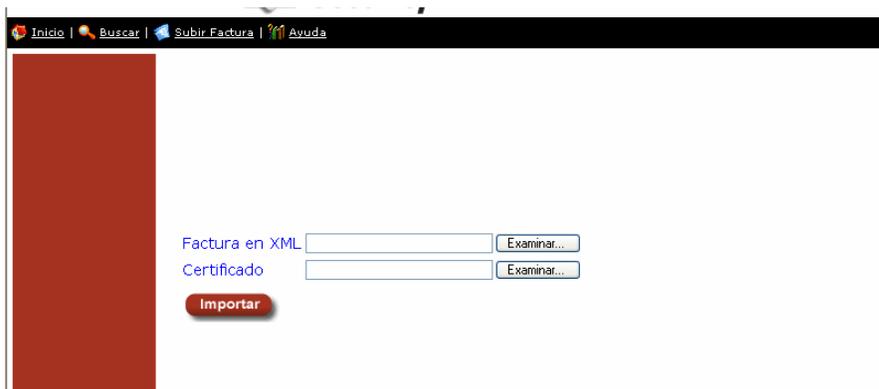


Imagen 4.1

Este usuario tiene los menús: buscar, subir factura y ayuda; su funcionalidad es igual a las ya explicadas anteriormente.



COMENTARIOS

Este proyecto fue realizado con las herramientas (anteriormente mencionadas) y conocimientos proporcionados.

El mecanismo fue levantado con apache por todas las características que se tiene, el diseño grafico con html; operaciones, procesos y programación avanzada se realizo ya con una programación mas avanzada como lo es php y java. Todas estas herramientas son libres de licenciamiento, por lo tanto, son de uso público; con esto y el conocimiento adecuado se logro llegar a realizar este proyecto.

Cabe mencionar que este sistema se usara en producción y se espera tener mejoras para adecuarse a los sistemas del mercado, siempre confiando en la utilización de estos software's.

Por mi parte ha sido una gran experiencia el tener la oportunidad de utilizar todo este tipo software al igual del saber que muchas aplicaciones en Web son realizadas con todo esto, el finalizar con un sistema practico de emotividad de seguir usándolo al igual de poder realizar lo que la imaginación de muchos de nosotros tenemos, pues aun creo yo que falta mucho por hacer y mejor que realzarlo en forma conjunta, menciono conjunta pues todo esto se ha hecho por muchas personas que han querido aportar algo a la humanidad.

Espero que la mayoría de los que empiezan a programar les guste utilizar toda esta gama gratuita de productos tan buenos como los productos en los que cobran sus licencias.



BIBLIOGRAFÍA

BLANCO Vicente J.

76063 Linux: instalación, administración
B5 y uso del sistema de México Alfa omega

42/02 LEBLANC Dee-Amm,

MACA Administración de empresas LAINUX Madrid anaya 2001 . 864 p

FACUNDO ARENA, Héctor.

76063 linux avanzado guía del administrador.
A72 Argentina MP ediciones 2000 279 p.

MOHR, James.

76063 linux recursos para el usuario.
M6418 México: prentice hall, 1999 789p.

CAMPIONE, Mary.

73J38 The Java tutorial. Boston. Addison. Wesley,
C36 2000.569p

McFEDRIES, Paúl.

76H94 Creando una pagina Web con HTLM fácil.
M3418 México: prentice-hall. 19996 212p.

SORIA MOMPARDER,R.

.76H94 Navegar en Internet: diseño y creación de paginas web html.

S67 México alfaomega: 1999 .201p.

TK5105 HOBBS, Lilian.

BARKAKATI, Naba Nabakakati.

76063 Red hat linux a fondo. Madrid:
B6718 Anaya Multimedia, 1999. 780p.

CARLING, M.

76063 Guía avanzada administración de sistemas linux.
C36961 Madrid: Anaya 2001. 864p.

LEBLANC, Dee-Ann.

76063 La Biblia de administración de sistemas linux.
L424718 Madrid: Anaya 2001. 864p.

Edición especial linux máxima. Seguridad.

.9A25 México: prentice-hall

M3818 1999. 482p.



- FACUNDO ARENA, Héctor.
76063 Linux avanzado guía del administrador.
A72 Argentina: MP Ediciones 2000. 709.p
- JUNCAR, José Antonio.
76063 Todo sobre linux Read hat 6.1
B65418 RAMA, 2000. 176p.
- HARLOW, Erick.
76063 Guía avanzada: desarrollo de aplicaciones linux con gtk + gdk.
H3618 Madrid. Prentice-Hall, 1999.496p.
- PITSS,D.
.76063 La biblia de red hay linux.
R43518 Mexico: Anaya Multimedia, 1998. 780p.
- RICARD MANUEL, Alberto.
.76063 Linux demitificado.
R494 Buenos aires: pearson educación, 2001.400p.
- SÁNCHEZ PRIETO SEBASTIÁN.
.8u65 Unix y linux guía práctica.
S36 Mexico: Alfaomega: Rama, 2000. 798p.
- STEVE, Shan.
.7603 Manual de administración de linux.
S423518 Madrid: McGraw Hill, 2000. 546p.
- CASTAGNETTO, Jesús.
.73P224 Profesional php Programming.
P76 USA: Wrox, 1999. 909p.
- TRIGOS GARCIA, Esteban.
373P224 php4.
T75 Madrid: Anaya 2000. 284p.
- WLLING, Luke.
.73p224 php and sql web development.
W45 Indianapolis: SAMS, 2001. 867p.
- MOMJIAN Bruce.
.9D Posgre SQL Introduction. And Concep. EUA, Adison- Wesley , 2001 462p.
- 43/05 PEREZ, Cesar
MA Desarrollo de paginas Web Dinamicas con PHP y MySQL México, Alfa Omega Ra-Ma,
2004, 513p



KLANDER, Lars

.9°25 prueba de hackers. España:
H3318 Anaya Multimedia 1998.569p

APPLIGER, Rolf.

.59 Sistemas de autenticación para O6618 seguridad en redes. Colombia : Rama.

RAYA, JOSE LUIS.

.59 La seguridad de una red con netware 5
R39 México: Alfa omega, 200 .576p.

SIYAN, Karanjit.

.875157 Internet y seguridad de redes.
H3718 Mexico: Prentice hall, 1995.407p.

RUSELL, Deborah.

.9A25 Computer security basics
R87 USA O`Reilly asosociates, Inc. 441p.

GRATTON, Pierre.

.9°25 Protección informática
G7318 México DF McGraw-hill Inter. Americana, 2002 739p.

Lawrence, Kansas

.9A25 UNIX Security
U55 Lawrence, Kansas: R D Books 1997 245p.

<http://biblioteca.dgsca.unam.mx>

<http://www.webestilo.com/php/php09a.phtml>