



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

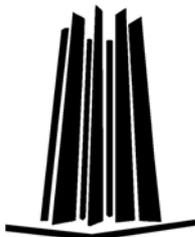
**FACULTAD DE ESTUDIOS SUPERIORES  
ARAGÓN**

**“INFORME DEL PROYECTO DE UN SISTEMA  
GESTOR DE INFORMACIÓN Y CONTROL EN EL  
SECTOR SALUD, UTILIZANDO SOFTWARE LIBRE”**

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

**L I N A R E S B A E Z A J E S Ú S  
A D R I Á N**

**ASESOR: ING. SILVIA VEGA MUYTOY**



**MÉXICO, 2006.**



Universidad Nacional  
Autónoma de México



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

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

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

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

## AGRADECIMIENTOS

**A** mis abuelos que crearon a dos seres ejemplares en mi vida y me enseñaron que de las cosas más sencillas salen formas maravillosas.

**A** mis papás que siempre creyeron en mi y me dieron más de lo que estaba en sus manos para poder alcanzar esta meta y que me siguen apoyando en todos mis proyectos, por lo que puedo decir que este logro también es de ellos.

**A** mis hermanas que siempre me apoyan y me brindan su cariño y experiencia para ser mejor persona, tanto en lo personal como en lo profesional y que seguiremos unidos como familia.

**A** mi sobrina Daniela porque me da la felicidad que de su ternura y sencillez pueden brotar.

**A** mi tía Benita y a Jorge mi cuñado que me han facilitado su ayuda y consejo.

**A** a todos mis amigos, que durante la carrera escolar, he conocido y hemos compartido buenos momentos en la aventura diaria del aprendizaje y me brindaron su amistad y compañía en el recorrido de ella.

# ÍNDICE

Introducción	Pág. 5
Capítulo 1: Informe del diplomado	
1.1 Sistema Operativo Linux	8
1.2 Instalación y Administración de Linux	18
1.3 Editores para a creación de páginas Web	33
1.4 Administradores de WWW con Linux	40
1.5 Programación con PHP	46
1.6 Interacción de WWW con Bases de Datos	55
1.7 Introducción a la seguridad en cómputo	64
1.8 Programación con Java	76
Capítulo 2: Proyecto	
2.1 Objetivo	85
2.2 Desarrollo	86
	93
Conclusión	102
Bibliografía	103

## ***INTRODUCCIÓN***

En los albores del siglo XXI los cambios en el ritmo de la vida son radicales, la globalización y sistematización han provocado el aceleramiento de cada actividad que propicia el descontrol y desorganización en todos los ámbitos, ya sean sociales, culturales, políticos, económicos y tecnológicos; siendo imprescindible una organización efectiva e todos los niveles; uno de ellos es el de la salud, factor y elemento fundamental de la vida diaria de cada individuo, pero que en los últimos años se ha visto rebasada por el aumento de solicitudes y la desorganización en los entes médicos y como consecuencia también de los directivos, administradores, proveedores, empleados, etc.

Basándose en este problema que se ha observado e indagado , se presenta el siguiente trabajo, que comprende dos capítulos llamados "Informe del Diplomado" e "Informe del Proyecto", que están basados en el diplomado "Desarrollo e Implementación de Sistemas de Software Libre, Linux" que se llevó a cabo del 11 de diciembre de 2004 al 17 de septiembre de 2005 en las instalaciones de la UNAM Campus DGSCA Mascarones, donde se proporcionaron las herramientas necesarias para poder generar desde una página web hasta la configuración y levantamiento de un servidor, desde su desarrollo hasta su implementación.

Con estas herramientas se pueden crear y administrar servidores en ordenadores personales, incluyendo su instalación, configuración y puesta en marcha, haciendo hincapié en las relaciones que deben existir desde su instalación entre cada uno de ellas y el papel que fungen en el levantamiento de un servidor completo. Estas herramientas son el sustento primordial, como Linux que brinda su soporte y manejo de dispositivos y librerías, Apache que da soporte a la publicación en Internet, MySQL que organiza le manera factible la información para su lectura, escritura y modificación de la misma, PHP que junto con Java proveen el acceso a los datos de una manera sencilla y coordinada y finalmente HTML que brinda la interfaz gráfica con la que se presentará en la red el servidor promoviendo la aceptación de los usuarios.

El primer módulo da la introducción a lo que es Linux, su origen y particularidades que lo caracterizan de los demás, incluyendo la interacción con él mismo a través de comandos y usando la interfaz consola que es con la que inicia todo Sistema Operativo para después inicializar la interfaz gráfica. En esta consola se ejecutan la mayoría de comandos para que el usuario se familiarice con la sintaxis. También se menciona la estructura de las carpetas, su jerarquía y como poder modificar las particularidades para que se apegue a nuestras necesidades de uso.

La administración de usuarios y archivos es un punto primordial, con los comandos ya establecidos, se pueden generar cuentas delimitándolas según su autorización de cada uno y controlar los permisos a las carpetas y archivos, originando a su vez distintos tipos de ligas. También se describe y administra las particularidades del gestor de arranque (LILO), que es el que rige los distintos tipos de particiones si en ellos se encuentran otros sistemas operativos.

El módulo dos describe los primeros pasos de un administrador en sus tareas básicas, desde crear copias de seguridad hasta el levantamiento de una nueva unidad en la red. Como primeras actividades se realizaron procesos del administrador, estableciendo políticas y planeo de actividades.

La instalación del sistema operativo (en este caso Linux en su distribución Slackware v10.0) se describe paso a paso, explicando cada punto esencial de lo que realiza la PC. Cabe mencionar que se observa más a profundidad el control de usuarios, en la denegación de permisos. Este módulo incluye con más detalle el manejo e instalación de dispositivos, configuración de la red y administración de memoria.

En el módulo tres se analizan el de desarrollo de sistemas con base en Linux. Aquí se utiliza la herramienta gráfica que es usada por la mayoría de los servidores para la edición de una página web: HTML. Como sus siglas en inglés (Hyper Text Markup Language) es un editor de hipertexto que por su simplicidad pero gran flexibilidad es ocupado hoy en día. Este lenguaje es solo para la generación de lo que llamamos carátula de la página, ya que no puede interactuar por si solo con una base de datos, para ello requiere de un lenguaje que lo pueda hacer por él.

El módulo cuatro es más complejo que el anterior; aquí se establece un servidor en un PC personal. Desde una introducción a servidores hasta implementación de sitios virtuales, pasando por instalación, configuración y ejecución del mismo, y ya dentro de la administración la incorporación de módulos, restricción de accesos y registro de ellos. Se ve también la ventaja que existe en un servidor Apache en comparación con otros con sus mismos objetivos.

Regresando a la parte de desarrollo de sistemas, se sabe HTML no puede interactuar con aplicaciones como por ejemplo bases de datos. Para ello el módulo cinco da una opción a esa desventaja, ya que aquí se muestra lo más básico al respecto del lenguaje PHP, orientado a la consulta de archivos que más adelante sería aplicado a manejadores de bases de datos.

El módulo seis fue dirigido a conocer el uso de un manejador de bases de datos, MySQL por ser libre de licencia y constituir un manejador seguro para la información al mantenerla íntegra y oportuna. Aquí se define como insertar, borrar, editar, y actualizar información, a partir de la creación de bases de datos, tablas y características dentro de las mismas.

Por ultimo el módulo siete que constituye el informar e implementar secuencias de seguridad que debe tener en cuenta un administrador al contemplar instalar un sistema. Cabe mencionar que por decisión conjunta del profesor y el grupo, se cambió un poco el temario para dar una mejor explicación de lo que puede existir en la red.

Programación con Java, explica que es este lenguaje y su estructura general, definiendo la diferencia entre Java con otros lenguajes de programación, partiendo del código fuente al código máquina explicando la función de la máquina virtual, que es su más grande diferencia al convertir al lenguaje en un entorno completo de programación-ejecución, lo que lo hace transportable a cualquier arquitectura de máquina.

El segundo capítulo consta del informe de un proyecto llamado "Sistema Gestor de Información y Control" (SGIC), el cual tiene la finalidad de exponer un sistema de información y control en las unidades médicas en general.

Muchos de estos sistemas pueden reflejar los factores internos y externos que como consecuencia traen consigo una buena o mala administración de los entes médicos, sin embargo la necesidad ha hecho que la planeación y el desarrollo de estos sistemas este a la par de una definición clara de cada una de las áreas operativas de los servicios médicos.

A continuación se exponen de manera clara y precisa el temario de los módulos que comprenden el diplomado y el curso adicional, además del desarrollo y descripción del proyecto resultante, esperando que sea de utilidad la información aquí vertida se describe el siguiente trabajo.

## Módulo I: Sistema Operativo Linux

Temario	
1. Introducción	4. Nociones de administración
2. Requerimientos de instalación	5. Fuentes de Información
3. Comandos y utilerías básicas	

En este primer módulo se ven algunos comandos básicos que ayudarán más adelante en los módulos subsecuentes, así como algunas nociones de administración básica, y requerimientos de instalación, también se mencionará una reseña de la historia de Linux.

Linux empieza a finales de los años setenta a través de un proyecto concertado para desarrollar nuevas técnicas de Sistema Operativo (SO). En 1968, un grupo de investigadores de la General Electric, AT&T Bell, y del Instituto Tecnológico de Massachusetts, realizó un proyecto especial de investigación de sistemas operativos llamado Multics. En 1969, Ken Thompson, de AT&T Bell Laboratories, desarrolló el SO UNIX, adaptándolo a las necesidades de un entorno de investigación.

Al año siguiente, el creador del lenguaje de programación C, Dennis Ritchie, colaboró con Ken Thompson en la tarea de pasar el código del sistema UNIX en el código del lenguaje C. Una de las ventajas de este lenguaje es que se puede acceder directamente a la arquitectura del hardware; así el SO Unix paso a ser transportable, capaz de ser ejecutado en cualquier computadora.

UNIX creció gradualmente hasta convertirse en un producto de software estándar, distribuido por muchos vendedores tales como Novell e IBM. Al principio, UNIX fue considerado como un producto de investigación. Sus primeras versiones fueron distribuidas de forma gratuita a los departamentos científicos de informática de muchas universidades de renombre. En 1972, los laboratorios Bell empezaron a emitir versiones oficiales de UNIX y a otorgar licencias del sistema a distintos usuarios. Uno de estos fue la Universidad de California, en Berkeley. Ésta añadió muchas características nuevas al sistema que, posteriormente, se convirtieron en estándar.

En 1993, AT&T vendió sus derechos sobre el UNIX a Novell. Desde entonces, Novell ha emitido sus propias versiones, llamadas UnixWare. Éstas han sido diseñadas para trabajar con el sistema NetWare de Novell.

Debido a la proliferación de versiones de UNIX en las décadas anteriores, el Instituto de Ingenieros Eléctricos y Electrónicos (IIEE) desarrollo un estándar del UNIX independiente para el American National Institute (ANSI). Este nuevo estándar ANSI del UNIX se llama Portable Operating System Interface for Computer Environments (POSIX). Este estándar define una norma universal a la cual se deben adherir todas las versiones de UNIX.

A medida que los PC fueron siendo más y más potentes, se hicieron varios intentos de desarrollar una versión de UNIX para PC. El Linux está diseñado específicamente para PC basados en Intel, según la norma POSIX. En esa época, los estudiantes utilizaban un programa llamado Minix, que incorporaba diferentes características de UNIX. Minix fue creado por el profesor Andrew Tannenbaum y ampliamente distribuido a estudiantes de todo el mundo a través de la Internet.

Linux tuvo su origen como proyecto personal de Linus Torvalds, un estudiante de la universidad de Helsinki en Finlandia. Linus inspirado por su interés en Minix, se propuso crear lo que en sus propias palabras sería un "mejor Minix que el Minix". Y después de un tiempo de trabajar por el mismo en su proyecto, realizó esta publicación en comp.os.minix: "Hola a todos los que usáis minix". Estoy haciendo un SO -gratis-(sólo es un hobby, no será grande y profesional como GNU) para clones AT 386(486).

En septiembre de 1991 lanzó la versión 0,01. Linux fue ampliamente distribuido por la Internet y en los años inmediatamente posteriores, varios programadores alrededor del mundo lo refinaron y le hicieron añadiduras incorporándole la mayoría de las aplicaciones y características estándar de un UNIX, construyendo así un sistema plenamente funcional.

Todos los principales administradores de ventanas han sido transportados a Linux. Este sistema cuenta con todas las utilidades para Internet, tales como ftp, telnet y slip. También un juego completo de utilidades de desarrollo de programas, como compiladores de C++ y depuradores. Aún con todas las prestaciones, el SO Linux sigue siendo pequeño, estable y rápido, puede ser ejecutado con tan sólo con 4 MB de memoria RAM.

La estructura de Linux, que en medida igual a UNIX se puede dividir generalmente en cuatro componentes principales: el núcleo(kernel), el shell, el sistema de archivos y las utilidades.

El núcleo es el programa medular que ejecuta programas y gestiona dispositivos de hardware tales como los discos y las impresoras.

El shell proporciona una interfaz para el usuario. Recibe órdenes del usuario y las envía al núcleo para ser ejecutadas. Se puede describir como un intérprete: traduce las órdenes que introduce el usuario y las envía al núcleo. La interfaz del shell es muy sencilla. Normalmente consiste en un inductor desde el que se teclea una orden y después se pulsa enter. En cierta forma, se está tecleando una orden en una línea. A menudo, esta línea se conoce como la línea de órdenes.

Como una alternativa a la interfaz de la línea de órdenes, Linux proporciona una interfaz gráfica de usuario(GUI) llamada X-Windows, que cuenta con varios administradores de ventanas que puede utilizar. Dos de los administradores de ventanas más populares son el Free Virtual Window Manager (fvwm) y el Open Look Window Manager (olwm).

Aunque un administrador de ventanas constituye una interfaz flexible y vistosa, hay que tener presente que tan sólo es un añadido al shell. El administrador de ventanas simplemente le pasa al shell las órdenes que recibe, éste las interpreta y las envía de nuevo al núcleo, para ser ejecutadas. A lo largo de los años se han ido desarrollando varios tipos diferentes de shell.

El sistema de archivos, organiza la forma en que se guardan los archivos en dispositivos de almacenamiento tales como los discos. Los archivos están organizados en directorios. Cada directorio puede contener un número cualquiera de subdirectorios, cada uno de los cuales puede a su vez, contener otros archivos. Todo el sistema de archivos de Linux es un gran conjunto interconectado de directorios que están organizado en una estructura jerárquica de árbol.

El núcleo, el shell y el sistema de archivos forman en conjunto la estructura básica del SO. Con estos tres elementos puede ejecutar programas, gestionar archivos e interactuar con el sistema.

Linux posee un gran número de utilidades que se pueden clasificar en tres categorías: editores, filtros y programas de comunicaciones. Y a diferencia de otros sistemas operativos se distribuye de forma gratuita bajo una licencia pública de GNU de la Free Software Foundation (Fundación de programas libres) lo que básicamente significa que puede ser copiado libremente, cambiado y distribuido dejando siempre disponible el código fuente.

El software de Linux es desarrollado por varios usuarios que deciden trabajar conjuntamente en un proyecto. Una vez que se ha completado se instala en una localización ftp de Internet. Cualquier usuario de éste puede acceder a la localización y descargar el software.

Un sistema Linux no requiere de una configuración especial de hardware (siendo una gran ventaja). Él funciona correctamente en un 386 SX con 4 MB de memoria. Pero se deberá tener en cuenta que aquellas aplicaciones con fuerte requerimiento de memoria y carga de la CPU (por ejemplo, las X Windows) irán sumamente lentas.

Una configuración mucho más utilizable sería la siguiente: 486 DX 33 MHz o superior, con 8 MB de memoria RAM y con tarjeta de vídeo VGA o superior.

Unidad de CD-ROM es altamente recomendable. Cabe decir que es imprescindible, dado que las distribuciones vienen en ese formato. En la actualidad, la mayoría de ellas están soportadas por Linux. Asimismo tiene soporte para módem y tarjeta de sonido.

Linux corre también en Laptop's o Notebooks 386 o superior, pudiendo correr también X-Windows en la mayoría de ellos.

En cuanto al espacio en disco duro, dependerá en gran medida de lo que se pretenda hacer. Con 100 MB resulta espacio suficiente para un Linux completo, incluidas las X Windows. A partir de ahí, lo que cada uno considere.

Las aplicaciones de este SO son bastas. Internet esta sostenida en UNIX y millones de servidores en el mundo operan en una computadora corriendo Linux, ya que éste hereda la fortaleza de UNIX.

Éste SO cuenta con aplicaciones de interconectividad nativas. Protocolos TCP/IP, IPX, NIS, SmallTalk, IGMP, AppleTalk, Samba, Mars\dots, Servidor de hipertexto Apache de World Wide Web, File Transfer Protocol, Gopher, Boletin Board System (BBS), Talk, Internet Relay Chat (IRC), correo electrónico E-mail, KDC, POP3, Network Time Protocol, NetWall, Name Server.

Linux tiene la facilidad de poder administrar cientos y miles de procesos simultáneos así como una multitud de usuarios en una sola computadora. Las aplicaciones llegan a superar por mucho la calidad y eficiencia de los programas y paquetes comerciales.

Se puede configurar Linux para que funcione como servidor, proporcionando así diversos servicios Internet; todo lo que necesita es el software de servidor adecuado y una organización de directorios segura. El software de servidor FTP, Web, Gopher y WAIS puede conseguirse gratis; Caldera Network Desktop instala automáticamente los servidores Web y FTP, de modo que inmediatamente puede empezar a funcionar como localizador.

Puede hacer que todos los servidores Internet se ejecuten simultáneamente; funcionan como programas demonio, esperando a recibir solicitudes de sus servicios por parte de usuarios remotos y de forma que, al recibirse una petición, la atenderá al servidor correspondiente. Así un usuario remoto podría conectarse a su servidor FTP y descargar archivos, al mismo tiempo que otro usuario esta conectado a su servidor Web, viendo sus páginas. Dependiendo de la frecuencia con que sean solicitados los servicios de cualquiera de estos servidores, convendrá ejecutarlos directamente o bajo el control del programa demonio inetd para que sean llamados únicamente cuando se reciban solicitudes de servicios. En el primer caso, sencillamente hay que invocar al programa servidor mientras que, para utilizar inetd, hay que añadir los elementos apropiados a los archivos `/etc/services` y `/etc/inetd.conf` y entonces ejecutar el programa demonio.

Otra aplicación es mediante las utilidades de correo electrónico, puede enviar mensajes a otros usuarios de su sistema. El nombre de un usuario forma parte de su dirección. Ahora hablaremos de dos utilidades con interfaces muy diferentes: Mail y Elm.

Mail es la utilidad estándar de correo electrónico que se encuentra en la mayoría de los sistemas Linux. Esta utilidad tiene una sencilla interfaz de línea de comandos con su propio juego de ordenes que trabaja en un shell propio de Mail. Se puede pensar que las diferentes ordenes de Mail sirven para enviar y recibir mensajes, no sólo es eso sino que definen las operaciones básicas de correo electrónico que están presentes en la mayoría de las utilidades de correo. Al enviar un mensaje mediante Mail, puede efectuar operaciones tales como volver a ver el mensaje que este escribiendo, guardarlo en un archivo o invocar al editor Vi para editar el mensaje. También puede crear un mensaje en un editor, guardarlo en un archivo y enviar después el contenido del archivo como un mensaje. Para recibir mensajes se invoca a la utilidad mediante la orden mail. En primer lugar se le proporciona una lista de cabeceras. Cada cabecera proporciona información sobre un mensaje, en la que se incluye la persona que lo ha enviado y el asunto del mensaje. Existen ordenes de Mail para leer, imprimir, guardar o eliminar un mensaje. Incluso puede contestar a un mensaje enviando una respuesta de inmediato.

La utilidad Elm usa una interfaz de pantalla completa con ordenes de una sola tecla, que sirven para realizar las mismas operaciones de enviar, recibir y guardar mensajes. A menudo, la interfaz de pantalla completa de Elm hace que sea mucho más fácil de utilizar que Mail. Las ordenes write y talk sirven para establecer una comunicación directa con otro usuario que actualmente este dentro del sistema. En realidad no se envía ningún mensaje, sino que todo lo que teclea aparece de inmediato en el terminal del otro usuario. La orden write es como una comunicación por radio, en la que un usuario habla y queda a la espera de la respuesta del otro. La orden talk es como una conversación telefónica en la que ambos usuarios pueden hablar simultáneamente.

Linux realiza acciones de enrutamiento de datos, conexión simultánea con redes de diversa naturaleza (topología) así como pared de seguridad para redes locales (firewalls).

Su computadora deja de ser un simple cliente y se transforma en un poderoso servidor de Internet: páginas web, correo electrónico, File Transfer Protocol, Usenet news server... aún cuando usted use una simple línea telefónica. Además que puede resguardar a su red local de posibles ataques externos (Linux sé auto protege); NO existen virus para Linux.

Puede ejecutar simultáneamente aplicaciones UNIX/Linux, MS Windows, MS-DOS, Amiga, Macintosh, 68K, Atari, Commodore hasta Nintendo en un mismo escritorio. Todo esto bajo el soporte de Linux. No se requiere reinicializar el sistema apagándolo y encendiéndolo de nuevo cada vez que se realice una modificación por muy compleja que ésta sea. Los servidores Linux pueden trabajar encendidos durante años sin requerir una reinicialización.

Linux es un sistema dedicado para las comunicaciones en redes y de óptimo intercambio de información. Al instalarlo el usuario es capaz de mantener el control del flujo de sus datos así como las operaciones realizadas por todos y cada uno de sus usuarios, creando un centro de control de trabajo y desarrollo completos

Puede administrar recursos vía local y remota. Maneja RPC para llamadas a procesos remotos. Manejo de múltiples tarjetas de red y módems simultáneos. Una sola estación Linux provee recursos de periféricos (CDROMs, módems, impresoras, multimedios, drive's, interfaces\dots) de todas y a todas las computadoras de una red. Alta seguridad. Nada se realiza sin que usted no se entere. Además de mantener una estricta auditoria de la información almacenada.

Las aplicaciones gráficas pueden ser ejecutadas vía remota usando el poderoso X11R6 Window System, el sistema de interfaz gráfico padre de todos los sistemas de ventanas, sin importar si se conecta desde estaciones basadas en UNIX System V y BSD como Silicon Graphics, Sun, Next, SCO, HP/UX, Ultrix, UNICOS, Alpha Digital UNIX, IBM AIX, súper cómputo CRAY y ambientes basados en OSF/Motif.

Comparte sistemas de archivos de variada naturaleza así como NFS (Net File System) que le permite utilizar y manipular archivos y directorios remotos como si estuvieran instalados físicamente en su propio servidor.

Su desempeño es altamente superior ante servidores de bajo desempeño tales como MS-Windows 95/NT y Novell NetWare/IntraNetWare.

Puede tener Linux corriendo en una PC basada en Intel, además en computadoras basadas en procesadores Motorola, RISC, MIPS como en Mac, Silicon Graphics, Sun, estaciones Alpha, siendo uno de los sistemas más transportables del mundo.

Para finalizar esta reseña histórica se puede decir que hay a disposición el software de la más alta calidad mundial creado por varias instituciones de investigación científica y tecnológica y compañías del mundo como la NASA, el Massachusetts Institute of Technology, Univ. de Harvard, AT&T, UNAM, Univ. Stanford, CERN, CIA.

La filosofía del sistema Linux permite al usuario tener pleno control sobre él, de tal manera que podrá modificarlo como guste, tal y como existen las limitantes en otros sistemas operativos. Desde el kernel que es la base principal de todo sistema hasta los documentos mínimos. Una ventaja de su sistema de archivos es que mantiene todas las librerías de sus respectivos programas independientes, esto significa que si algún programa provoca un error no alterara a los demás, una misma ventaja de esto es que si se desea instalar sólo hay que utilizar el comando pkgtool o en algunos casos borrar únicamente la carpeta donde esta instalado y listo, las librerías que ocupaba son eliminadas y no afecta a los demás programas ya que no se hacen dependientes de librerías de sistema.

En la tabla 1 se describe la estructura de librerías de Linux, además de incluir una breve descripción de su función o contenido:

Directorio	Descripción
/bin	Los programas user-related básicos están aquí. Cáscaras y programas del comando tales como ls.
/boot	Archivos cargador-relacionados de LILO
/dev	Bloque y archivos del dispositivo en modo carácter.
/etc	Archivos de la inicialización de la configuración y de sistema.
/home	Directorios caseros del usuario, excepto raíz.
/lib	Bibliotecas esenciales (como los módulos de la biblioteca y del núcleo del sistema C).
/mnt	Mountpoint genérico para los filesystems que fijan.
/opt	Paquetes de software opcionales. Slackware instala KDE a esta localización.
/proc	Mountpoint del filesystem de Proc para la interacción del núcleo.
/root	Directorio casero de la raíz.
/sbin	Binarios del sistema. Los programas funcionan por la raíz o en el tiempo del cargador.
/tmp	Directorio temporal. Cada uno tiene permisos de read-write aquí.
/usr	Programas user-related tales como X11, Netscape, y pino.
/var	Ficheros de diario de sistema, archivos de la cerradura, carretes del correo, y carretes de la impresora.

*Tabla 1. Estructura de Librerías de Linux*

Además de estos directorios básicos, puede haber algunos otros. Por ejemplo, usted puede tener un directorio de /mnt/cdrom para montar su impulsión de CD-ROM o /mnt/floppy para montar un disquete.

Existen diversas plataformas y distribuciones de Linux, hay que aclarar que no son la misma cosa; las plataformas son las versiones base para las siguientes distribuciones, por ejemplo RedHat es una plataforma especializada en redes, y una distribución es Fedora que resulta ser una versión apoyada en RedHat de igual manera orientada a redes. La plataforma que se utilizó durante el diplomado es llamado Slackware y un "hijo" de él es Goblinx, esa es la diferencia entre plataforma y distribución. Algunas distribuciones y plataformas se pueden encontrar en [www.isolinux.org](http://www.isolinux.org), debido al gran auge de este SO, no todos se encuentran en esta página.

Algunas de las distribuciones más conocidas son:

Distribución	Versión	Kernel	Instalación	Nivel de Usuario	
Conectiva Linux	10.0	N/A	Media	Principiante	
Corel Linux SE	3.0	N/A	Baja	Novato	
Debian GNU/Linux	3.0r4	2.4	Media-Alta	Principiante	
Mandrake	10.1	2.4.19	Media	Principiante	
Caldera OpenLinuxServer	4.0	N/A	Media	Principiante	
Red Hat Pro	9.0	2.4.18	Media	Experto	
Slackware	10.1	2.4.19	Alta	Experto	
SuSe Linux	9.1	2.4.19	Baja	Novato	
TurboLinux Server	8.0	2.4.18	Media-Alta	Principiante	
UnitedLinux	1.0	N/A	Media	Novato	
Ubuntu	1.bz2	N/A	No instalable	Novato	
Yellow Dog	4.01	2.4.19	Alta	Experto	
Knoppix	3.7	2.4.19	No instalable	Novato	

Fedora	Core 3	2.4.19	Media	Experto	
FreeBSD	5.3	2.4.18	Media_Alta	Experto	
Gentoo	2005.0	2.4.19	No instalable	Principiante	
Lycoris	4	2.4.19	Media-Alta	Experto	
Sorcerer	1	2.4.19	Baja	Principiante	
TSL	2.2	2.4.18	Media	Novato	

Tabla 2. Distribuciones más conocidas de Linux

A continuación (Tabla 3) se verán algunos comandos básicos que ayudarán a interactuar de manera eficiente en el SO. Se muestra una lista en la cual se presentan los comandos y una descripción breve de su funcionamiento:

Comando / sintaxis	Descripción	Ejemplos
at [-l] hora [fecha]	Ejecuta un comando más tarde	at 6pm Friday miscrypt
cal [[mes] año]	Muestra un calendario del mes/año	cal 1 2025
date [mddhhmm] [+form]	Muestra la hora y la fecha	date
echo string	Escribe mensaje en la salida estándar	echo ``Hola mundo''
finger usuario	Muestra información general sobre un usuario en la red	finger nn@maquina.aca.com.co
id	Número id de un usuario	id usuario
kill [-seña] PID	Matar un proceso	kill 1234
man comando	Ayuda del comando especificado	man gcc man -k printer
passwd	Cambia la contraseña.	passwd
ps [axiu]	Muestra información sobre los procesos que se están ejecutando en el sistema	ps -ux ps -ef
who / rwho	Muestra información de los usuarios	who

cat <i>fich1</i> [... <i>fichM</i> ]	Concatena y muestra un archivo	cat /etc/passwd
cd [ <i>dir</i> ]	Cambia de directorio	cd /tmp
chmod <i>permisos fich</i>	Cambia los permisos de un archivo	chmod +x miscript
chown <i>usuario:grupo fich</i>	Cambia el dueño un archivo	chown nobody miscript
cp <i>fich1...fichN dir</i>	Copia archivos	cp foo foo.backup
diff [-e] <i>arch1 arch2</i>	Encuentra diferencia entre archivos	diff foo.c newfoo.c
du [-s <i>ab</i> ] <i>fich</i>	Reporta el tamaño del directorio	du -s /home/
file <i>arch</i>	Muestra el tipo de un archivo	file arc_desconocido
find <i>dir test acción</i>	Encuentra archivos.	find . -name ``.bak" -print
grep [-c <i>lnv</i> ] <i>expr archivos</i>	Busca patrones en archivos	grep mike /etc/passwd
head - <i>count fich</i>	Muestra el inicio de un archivo	head prog1.c
mkdir <i>dir</i>	Crea un directorio.	mkdir temp
mv <i>fich1 ...fichN dir</i>	Mueve un archivo(s) a un directorio	mv a.out prog1
mv <i>fich1 fich2</i>	Renombra un archivo.	mv .c prog_dir
less / more <i>fich(s)</i>	Visualiza página a página un archivo.	more muy_largo.c
	less acepta comandos vi.	less muy_largo.c
ls	Lista el contenido del directorio	ls -l /usr/bin
pwd	Muestra la ruta del directorio actual	pwd
rm <i>fich</i>	Borra un fichero.	rm foo.c
rm -r <i>dir</i>	Borra todo un directorio	rm -rf prog_dir
rmdir <i>dir</i>	Borra un directorio vacío	rmdir prog_dir
tail - <i>count fich</i>	Muestra el final de un archivo	tail prog1.c
vi <i>fich</i>	Edita un archivo.	vi profile
hostname (-i,-d)	Muestra el nombre del servidor	Hostname
uname (-r, -m)	Muestra la version del kernel	uname -r

Tabla 3. Comandos en Linux

Linux tiene la propiedad de poder abrir varias ventanas o sesiones en una misma máquina, con ello se podrá hacer tareas de manera independiente, o visto de otra manera que una sesión sea únicamente para un usuario. Linux no requiere de algún hardware explícito o mínimo, otra ventaja en

él. Algunas especificaciones fueron descritas con anterioridad. Un SO Linux puede correr en cualquier máquina, pero para que un servidor (que es lo que más interesa) pueda dar soporte a las exigencias de los usuarios, se debe contar por lo menos con un equipo más o menos equipado para eso; los demás usuarios pueden tener un equipo con lo mínimo para poder trabajar con el servidor, a través de los accesos remotos, los cuales permiten trabajar en un equipo convencional, pero con las propiedades de un servidor.

Con los comandos anteriores se puede crear cuentas de usuario, cambiar contraseñas de los mismos, crear y eliminar directorios, limitar o permitir el acceso a ellos por parte de los usuarios, redireccionar o filtrar información, crear ligas duras y suaves, interrumpir trabajos o procesos, etc. En fin con ellos se puede dar una administración básica a nuestro servidor.

Los programas cuando son instalados tienen un documento de configuración, al cual sólo root o cualquier usuario con permiso puede modificar, en él está contenido las instrucciones como va a funcionar el programa. Para editarlo se cuenta con las herramientas vi y pico, que son unos editores de texto, su sintaxis es *vi/pico [nombre.extensión]*. El comando pico muestra sus comandos en la parte inferior de la pantalla, presionando Ctrl + la letra de la función, en cambio el editor vi es más completo por lo cual requiere de más comandos los cuales no muestra.

Con ellos se puede editar el documento de lilo.conf que es el documento que configura el gestor de arranque LILO. Los gestores de arranque son el primer componente que se ejecuta cuando se enciende el hardware. Su misión es cargar el núcleo del sistema en memoria, y cargar el/los SO's. El Gestor LILO tiene la facilidad de poder cargar varios SO's tanto distribuciones Linux como de otras corporaciones (Windows, MAC, UNIX), el único sistema que no ha podido compaginar en su biblioteca es Windows 2003. Una vez que se haya modificado el documento se debe de ejecutar el comando *lilo* que va a leer la configuración definida por el usuario y hacer reales esos datos, esto quiere decir que en la MBR (Master Boot Record) o en el sector de arranque definida, se escribirán los datos modificados.

En el documento se distinguen claramente tres partes de estos archivos. En la primera se definen los valores generales de la configuración de donde será instalado LILO. En la segunda parte se ve cual es la partición que fue definida para Linux, con el nombre que mostrara en la lista de LILO y con la posibilidad de asignar un password, y en la tercera se encuentra la partición destinada a los demás SO.

Para reiniciar, apagar o terminar sesión se utilizan los comandos logout, reboot, halt y shutdown. El comando logout permitirá al usuario terminar sesión en el equipo sin reiniciarlo, reboot sólo reiniciara el equipo sin apagarlo, y halt lo apagara. Shutdown es un comando que funciona combinado con reboot y halt, la diferencia de utilizar este comando en lugar de los otros dos es que shutdown cerrara todos los procesos antes de continuar con el reinicio o apagado.

Otra utilidad que tiene Linux es la de montar los dispositivos cuando realmente los requiera. Los sistemas Windows (por ejemplo) montan todos los discos duros, o particiones sin que el usuario vaya a ocuparlo, además de que monta automáticamente los discos extraíbles sin previo aviso, lo que provoca el uso de memoria desaprovechada. Para montar un dispositivo sólo se debe acudir al comando mount para montar o unmount para desmontar.

En el archivo inittab se encuentra la configuración sobre el inicio de sesión, esto es si se quiere iniciar en modo consola o modo gráfico; existen otros niveles de ejecución pero son poco o nada usados. Los niveles de ejecución son 7 en total y generalmente están configurados del siguiente modo:

- 0 – Sistema apagado
- 1 – Sistema monousuario
- 2 – Sistema multiusuario con soporte NFS
- 3 – Sistema multiusuario con soporte en red
- 4 – Sin usar
- 5 – Modo multiusuario con operación gráfica
- 6 – Reiniciar el sistema

Para mantener una seguridad en el sistema se permite la configuración de los procesos ó servicios que se van a ejecutar, por ejemplo si el sistema esta destinado a conexión FTP sólo se habilita esta línea en el documento de configuración de servicios.

## Módulo II: Instalación y administración de Linux

Temario	
1. Perfil y actividades del administrador	7. Administración del área de swap.
2. Instalación	8. Realización de respaldos.
3. Dar de alta y/o baja el sistema.	9. Uso de memoria y CPU
4. Mantenimiento de claves de usuarios.	10. Interfaces gráficas en LINUX
5. Sistema de archivos.	11. Configuración y mantenimiento del uso de la red.
6. Instalación y mantenimiento de dispositivos.	

Este módulo es la complementación del módulo anterior, aquí se verá de forma más explícita la administración de Linux. Así como la instalación y configuración desde cero del sistema Linux, y algunos comandos destinados para el uso exclusivo del administrador. El administrador se encarga de configurar, mantener y actualizar el sistema en la red, además de brindar un cuidado a los periféricos para que estén listos para otro usuario.

La administración en un sistema debe de brindar seguridad en el ambiente, confianza y eficiencia. Existe la posibilidad de dividir el sistema en varios administradores por su propia extensión.

Una de las primeras tareas de un administrador es planear sus actividades así como la de crear respaldos o copias de seguridad de los documentos importantes. Una regla que se debe seguir

en seguridad de documentos y copia de estos es tener en total 4 copias como mínimo. Una copia debe de estar en la misma oficina donde se encuentra el original. Otra debe estar en el mismo edificio, la tercera en un edificio cercano, y la otra debe estar en otro estado ó ciudad (geográficamente hablando). Estas copias deben ser hechas en distintos formatos, como son la cinta magnética, disco duro, disco magnético, y cassette.

La información que por lo regular se respalda es: información de los usuarios, correos electrónicos, bases de datos, y archivos originales del sistema.

El administrador debe conocer ciertas herramientas, para la administración que viene incluidas en el sistema operativo. Un ejemplo de eso sería la tabla 4:

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, howto's, info), impresa (libros, manuales), Internet.

*Tabla 4. Comandos de administración*

Algunas utilerías ocupadas en el diplomado fueron find y cron. Find sirve para buscar algún documento en el árbol de directorios. Cron es un comando con el cual se podrá mantener un cronómetro sobre las actividades, por ejemplo, enviar un mensaje a nuestros usuarios de que el servidor se apagara en cierto tiempo, para que tengan la precaución del tiempo que les resta para terminar sus actividades y no perder información, sediendo el mando del equipo y sus actividades de cierre al comando cron con sus respectivas órdenes.

Es bien sabido que un administrador debe leer la documentación de un programa, porque el diseñador incluye en ella no sólo como debe instalarse, sino también todas las especificaciones de su programa y sus limitantes. Un ejemplo de esto es apache (programa para administración de servidores) que dependiendo de lo que se requiera hacer con él, se instalara de diferente manera.

Los conocimientos requeridos con los que debe contar un administrador son:

- 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.

Además de tener estos conocimientos, debe conocer el campo del hardware. Por ejemplo de la máquina debe conocer el modelo, características, capacidad, etc. Un punto externo en esto es la

ubicación física que va a tener el equipo, ver si es apropiado y tiene el espacio suficiente para mantenerlo en buen estado. La revisión de listas de HCL o soporte de software es otro punto dentro de la administración que no se debe pasar por alto. Y antes de comprar el equipo al por mayor debe hacer las pruebas suficientes.

Conocer los dispositivos o periféricos que tenemos interconectados es otra tarea de la administración. Si no se sabe que dispositivos se tienen instalados dentro de la máquina, se tiene la opción de abrirla, buscar sus número de serie en la placa, buscar en Internet el número y saber sus especificaciones técnicas. Si esto es un poco tedioso se puede ejecutar el comando `lspci` y la consola nos desplegará una pantalla donde se enlista los dispositivos que se tienen conectados a la motherboard. Para probar su debido funcionamiento, podemos ejecutar `modprobe` seguido del nombre que le corresponde a cada dispositivo (el comando `lspci` muestra los nombres de cada dispositivo, siendo el primero en cada fila), se debe aclarar que módulo en Linux significa driver en Windows. Si el dispositivo entrega una confirmación de su buen funcionamiento, no debe requerir más atención, pero sino se debe buscar y bajar su módulo e instalarlo.

Después de identificar el equipo y todos sus componentes, se establecen las políticas de uso y administración, lo que conlleva a dar permisos especiales a nuestros usuarios, creando grupos para cada tipo de ellos y los organizarlos según sus necesidades, además de mantener canales de comunicación, ya sea establecida o mensajería.

Cuando se establezcan las políticas, se sugiere dar a conocer los puntos de manera clara y concisa. En la administración de usuarios se debe dar cavidad a apertura de nuevas cuentas, informar sobre horas de mantenimiento, indicar sobre los respaldos y su responsabilidad, y su espacio personal.

Una forma de mantener la comunicación con los usuarios es por medio de canales seguros. Algunos de ellos pueden ser: `/etc/motd`, `news`, `wall`, `write`, `mail`, `web`.

Las tareas comunes de un administrador, ya implantado el sistema son:

- Administrar usuarios (altas, bajas o cambios).
- Instalar y configurar los dispositivos.
- Programar respaldos periódicamente.
- Capacitar usuarios para el buen manejo del sistema.
- Dar seguridad al sistema.
- Registrar los cambios que se hagan en el sistema.
- Dar asesoría a los usuarios.
- Mantenimiento de las claves de usuarios.
- Instalación y administración de software.
- Configurar y mantener las interfaces de red.
- Administración de recursos como: discos duros, `cpu`, memoria, etc.
- Monitorear constantemente el sistema.
- Detección de fallas en el sistema.
- Y realizar una auditoría constante y periódica en el sistema.

Cuando se instala en el equipo Linux (en este caso Slackware), se debe conocer los requerimientos mínimos, para Slackware no se requieren muchos, corre en máquina 386 o superiores. Los requerimientos de Slackware son los siguientes (tabla 5):

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

*Tabla 5. Requerimientos mínimos para instalación de Linux*

Linux se compone de una serie de paquetes que se van instalando, en este caso Slackware los instala alfabéticamente. Esta distribución permite seleccionar los paquetes enteros que se van a instalar o por programa individual. La tabla 6 enlista los paquetes:

<b>A</b>	Es la base que contiene el kernel y utilidades básicas del sistema
<b>AP</b>	Varias aplicaciones que no requieren del sistema X Windows.
<b>D</b>	Herramientas de desarrollo de programas, compiladores, depuradores, intérpretes como C, Perl, Lisp, etc.
<b>E</b>	El editor GNU Emacs, es tan grande que requiere su propia serie.
<b>F</b>	Contiene FAQs, HOWTOs, y documentación esencial para la administración.
<b>GNOME</b>	Contiene el ambiente de escritorio de GNOME y librerías relacionadas
<b>K</b>	El código fuente para el núcleo de Linux.
<b>KDE</b>	Contiene el ambiente de trabajo de escritorio
<b>KDE y QTKDEI</b>	Contiene el conjunto de idiomas internacionales para KDE
<b>L</b>	Librerías del sistema necesarias para los escritorios GNOME, KDE y más.
<b>N</b>	Contiene programas relacionados con la red. Demonios, clientes, etc.
<b>T</b>	Software de tipo TeX usado generalmente para documentos técnicos.
<b>TCL</b>	Contiene las herramientas del lenguaje de comandos, el Tk, el TclX, y el TkDesk.
<b>X</b>	Contiene la base del sistema X Window.
<b>XAP</b>	Contiene una colección de aplicaciones para X Por ejemplo Ghostscripty Netscape.
<b>Y</b>	Contiene juegos (una colección de juegos de BSD).

*Tabla 6. Paquetes de instalación de Linux*

Hay varios tipos de instalación: CD-ROM (es más sencilla), disco de arranque, si no se cuenta con la opción de arranque del CD-ROM, Disco red (recomendado si se instala en varios equipos simultáneamente, se hace por medio de FTP, NFS, y HTTP), Disco PCMCIA (esta opción se usa si se hace la instalación en una PC portátil). Ahora, una vez seleccionado el tipo, se instala el SO en el equipo, para eso se debe tener a la mano los discos (en este caso para la versión 10 de Slackware son 2). Antes se tiene que verificar que este configurado el inicio de arranque de los dispositivos con el CD-ROM en primer lugar.

Cuando el CD-ROM sea detectado e inicie el disco de arranque, desplegará la pantalla con el prompt del cd (boot). La presentación en pantalla sería la mostrada en la figura 1:

```
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,wpcm,irq (needed in rare cases where probing fails)
or hdx=cdrom (force detection of an IDE/ATAPI CD-ROM drive)
where hdx can be any of hda through hdt.

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

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

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

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

boot: _
```

*Figura 1. Presentación de instalación Slackware*

Sólo se dará enter aquí, el sistema seleccionará la imagen automáticamente realizando un escanéo del equipo, y dando un informe de lo ocurrido. Al inicio se define el tipo de teclado que manejará durante la instalación, para esto se selecciona el mapa de caracteres qwerty/es.map, escogiéndolo de una lista desplazándose en ella con las teclas de movimiento.

Cuando se acepte el mapa, muestra otra pantalla en la cual se verifica la posición de y localización de teclas, si está correcto, se escribe 1 para guardar cambios y salir. A continuación se selecciona el usuario raíz, en este caso root. Esto es con el fin poder seleccionar el shell con que trabajará el servidor.

Desde el prompt de root se ejecuta fdisk /dev/hda (comando /dispositivo/disco\_a\_particionar), que desplegará la pantalla donde muestra las especificaciones para la creación de particiones (fig 2).

```

root@slackware:~# fdisk /dev/hda

The number of cylinders for this disk is set to 1048.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
 1) software that runs at boot time (e.g., old versions of LILO)
 2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): _

```

*Figura 2. Creación de particiones*

Para ver los comandos con los que se cuenta se escribe m o help + Enter, si se desea usar un comando sólo lo se escribe y se pulsa Enter (fig 3):

```

and could in certain setups cause problems with:
 1) software that runs at boot time (e.g., old versions of LILO)
 2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): m
Command action
 a  toggle a bootable flag
 b  edit bsd disklabel
 c  toggle the dos compatibility flag
 d  delete a partition
 l  list known partition types
 m  print this menu
 n  add a new partition
 o  create a new empty DOS partition table
 p  print the partition table
 q  quit without saving changes
 s  create a new empty Sun disklabel
 t  change a partition's system id
 u  change display/entry units
 v  verify the partition table
 w  write table to disk and exit
 x  extra functionality (experts only)

Command (m for help): _

```

*Figura 3. Comandos para crear particiones*

Para ver el tipo de particiones se escribe " l " seguido de ENTER, la partición adecuada pa nuestro servidor sería 82 Linux Swap para el espacio reservado para la memoria Swap y 83 Linux para el espacio reservado al SO.

Ahora se define el espacio en disco para las particiones; en que partición se desea instalar el SO y que partición será destinada a la memoria SWAP. El Espacio swap en Linux es usado cuando la cantidad de memoria física (RAM) está llena. Si el sistema necesita más recursos de memoria y la memoria física está llena, las páginas inactivas de la memoria se mueven al espacio swap. Puede ser de ayuda para las máquinas con poca memoria RAM, pero no debería considerarse como algo que pueda sustituir a más RAM debido a que el espacio Swap se encuentra en discos duros, que tienen un tiempo de acceso más lento que la memoria física.

Se debe destinar suficiente espacio para la carpeta de usuarios y el home del sistema. Se recomienda tener particiones distintas para estas dos carpetas, para tener un mejor control de memoria.

Como en el equipo se cuenta con un SO instalado (Windows), se debe imprimir (ver) las particiones para no borrar las ya creadas o encimar otras. Para eso damos un p+ENTER.

Se debe mencionar que la presentación de las particiones se hace mediante la cantidad de cilindros del disco duro, por eso se debe decidir con que cilindro inicia y donde termina (Fig. 4).

```
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
```

*Figura 4. Particiones existentes*

Como se va a hacer una instalación básica sólo habrá dos particiones, una para la swap y otra para Linux nativo. Para crearla se escribe **n** seguido de enter. Dará dos opciones una primaria y otra extendida. En la primaria debe ir el sistema dominante. En la extendida se podrán crear tantas particiones como el disco lo soporte, para otros sistemas, se selecciona la primaria. Aquí se selecciona el cilindro donde se quiera comenzar, finalizar y el tamaño en MB para la partición. Esta será destinada a la memoria swap. El espacio destinado debe ser igual o el doble que la memoria RAM. Para cambiar el formato de la partición de la swap se escribe **t** seguido de enter, seleccionar el número designado a la partición y el número con el tipo de partición, en este caso 82. Para crear la partición se siguen los mismos pasos anteriores, con la diferencia que el tamaño variara y el tipo se queda por default. Se pueden crear hasta 4 particiones primarias.

Ya que se haya designado las particiones se guardan los cambios y se sale de esa pantalla (mientras no se de **w + ENTER** los cambios no serán almacenados), regresando al prompt del **cd**, posteriormente se escribe y ejecuta **setup**.

Para continuar, se define el tipo de teclado o mapa de caracteres que utilizará nuestro sistema operativo de aquí en adelante (se puede volver a modificar una vez finalizada la instalación), se da enter sobre la selección **KEYMAP** de la lista mostrada.

Es la misma selección de teclado que se había mencionado. Una vez definido, automáticamente el programa seleccionara el espacio destinado a la swap, dándole el formato. A continuación se presenta una pantalla donde estarán enlistadas las particiones listas para instalar Linux. Una vez seleccionada se escoge el tipo de sistema de archivos, en este caso se eligió **reiserfs**, a continuación la pantalla con los tres tipos a escoger por la instalación.

Las tablas 7 y 8 describen las diferencias de los tres tipos principales de sistemas de archivos, además de otros conocidos:

Sistema de archivos	Descripción	
Ext2	Solidez	Tuvo un gran número de mejoras en su tiempo, por ser el sistema con más años en el kernel de Linux. Es una herramienta que analiza el sistema de archivos completos y no sólo sectores modificados. Esta se convirtió en su debilidad, al no ser apto para servidores de alto desempeño.
	Fácil actualización	Tiene soporte con journaling Ext3
Ext3	Disponibilidad	Con su sistema journaling ya no es necesario hacer una comprobación del sistema después de un cierre no limpio. Únicamente se hace una comprobación de consistencia en los casos puntuales en los que se producen determinados errores de hardware, proporcionando una más rápida recuperación.
	Integridad de datos	Proporciona una integridad de los datos tras sufrir un cierre no limpio. Dando soporte a niveles de protección.
	Velocidad	Como puede escribir datos de manera continua, aumenta la velocidad de recuperación. Pero delimita la integridad de datos.
ReiserFS	Nueva arquitectura	A diferencia de Ext3, éste es creado desde cero. Su sistema es transaccional como ext3 pero incorpora una estructura interna que utiliza conceptos de árboles binarios, parecido a bases de datos.
Otros	XFS	Es diseñado por SGI utilizado en el SO Irix. Su estructura incorpora soporte de ancho de banda en tiempo real, extensiones y sistemas de archivos distribuidos.
	JFS	Fue diseñado por IBM. Su estructura es similar a reiserfs.

Tabla 7. Sistemas de archivos en Linux

Característica	Ext2	Ext3	ReiserFS	JFS	XFS
Estabilidad.	Excelente	Buena	Buena	Media	Buena
Herramientas para recuperar archivos borrados.	Sí (complejas)	Sí (complejas)	No	No	No
Tiempo de re-arranque después de una caída.	Largo, incluso muy extenso	Corto	Muy corto	Muy corto	Muy corto
Estado de los datos en caso de una caída.	En general bueno, pero existe el riesgo de pérdida parcial o total de los datos.	Muy bueno	Medio	Muy bueno	Muy bueno

Soporte para ACL	Si	Sí	No	No	Sí
------------------	----	----	----	----	----

*Tabla 8. Tabla comparativa de Sistemas de Archivos*

Cuando se seleccione el sistema de archivos (en este caso reiserfs), se desplegará una pantalla con la descripción de la partición. El siguiente paso será elegir por que medios vamos a instalar el SO. La opción más viable es CD-ROM, montándolo de manera automática.

La pantalla que sigue muestra los paquetes que se pueden instalar en el disco duro. Como se mencionó anteriormente estarán en orden alfabético y se puede instalar por paquetes o de manera manual (de forma manual se consume más tiempo).

Una vez instalado los paquetes, se elige el kernel que más beneficie. Estos pueden estar dentro del CD-ROM, partición o compartido en red (en este caso CD-ROM). Se elige el bare.i como selección de kernel. El disco de arranque será útil cuando no podamos entrar al sistema o se requiera hacer algunas modificaciones. La siguiente pantalla ofrece esa opción de crearlo.

La configuración de dispositivos se hace por medio de hotplug que hará un escaneo de los dispositivos de hardware y los activa de manera automática. Como paso siguiente se pide configurar el gestor de arranque. Configurar el módem es el siguiente paso. Pudiendo hacerse si es un externo o interno pero en tarjeta independiente, sino pertenece a los winmodem.

LILO es un gestor de arranque que ayudará a levantar varios SO en una misma máquina. Se elige el modo experto para poder anexar los demás SO. Se da una opción extra que por lo regular no se ocupa. Después muestra las resoluciones para la tarjeta de video, en este caso 1024\*768\*256. Lo que sigue es decidir donde instalar el gestor; se elige MBR (Master Boot Record). Después, el tiempo de elección entre sistemas (forever) y se anexan los distintos sistemas. Se empieza por Windows (es preferible la instalación Windows-Linux que vicerversa), se muestra las particiones donde esta localizado el sistema y se escribe un nombre a cada una de ellas. Seguido se anexa Linux y se escribe de igual manera el nombre con el cual se presentará en la lista. Finalmente se escribe Install y el gestor se terminara de instalar.

Ahora se instala el mouse, eligiendo su tipo de puerto. Si se cuenta con una tarjeta de red se pedirá que se configure dependiendo de cómo sea la conexión y esto por consiguiente llevará a administrar el correo electrónico y el tipo de dirección IP ya sea dinámica o estática.

Se selecciona los servicios que brindará a los usuarios el equipo servidor de una lista mostrada en pantalla (fig 5).



Figura 5. Servicios Disponibles de Linux

Para terminar, se configura la zona, el horario y el X(intertaz gráfica) con que se desea iniciar la sesión, sólo queda asignar un password a root y reiniciar el sistema y sacando el cd de arranque.



El sistema leerá el gestor de arranque, en este caso LILLO (fig 6)y se elige Linux. Cuando empiece a cargar mostrara todos los procesos que hace un SO antes de cargarse la sesión de usuario (Windows no los muestra). Desde esta interfaz modo texto se puede hacer altas, bajas o cambios de clave en los usuarios, además de crear grupos, cambiando sus permisos dependiendo de cómo organizarlos en los grupos creados. También se les puede catalogar dependiendo de sus necesidades o que interprete

```
Welcome to the LILLO Boot Loader!

Please enter the name of the partition you would like to boot
at the prompt below. The choices are:

Linux - (Linux partition)

boot: _
```

Figura 6. .Pantalla de bienvenida de LILLO

de comandos deseen utilizar.

Con el sistema activo se pueden anexar todos los dispositivos de entrada y salida como impresoras terminales, cdrom's y disquetes.

El administrador debe ser capaz de dar de alta o baja el sistema de ello dependerá la confiabilidad y rendimiento del sistema. Para ello hay que saber lo que ocurre dentro del mismo. Cuando arranca el proceso de Linux, carga una copia del kernel que se encuentra en el directorio /boot, seguido detecta los dispositivos de hardware y los pone a disposición del sistema, busca en

los scripts de configuración para leer el nivel de inicio para el cual esta configurado por default, luego busca los scripts del runlevel, finalmente ejecuta el scriptrc.local. aquí como en el módulo anterior, se dio un vistazo a los diferentes modos de inicio del sistema. Para apagar el sistema existen varios comandos :

Comando	Descripción	Sintaxis	
Poweroff	Hace un apagado inmediato del sistema.	Poweroff -f	
Halt	Funciona de igual manera que el comando anterior. Este comando halt cierra cada uno de los procesos antes de apagar el equipo.	Halt -f	
Reboot	Reinicia el equipo, pero es igual de dañino que los otros dos al no verificar procesos.	reboot	
Init	Sirve para cambiar el modo de inicio de ejecución del sistema, colocando después del comando el nivel de inicio.	INIT [0...6]	
Telinit	Funciona igual que el anterior con la diferencia que éste lo hace de modo remoto, es decir al equipo que se está conecta.	TELINIT [0...6]	
Shutdown	Es el más eficiente para cierre de sesión. Ya que da una baja de sistema de manera eficiente. Este comando manda una señal de sincronización para cerrar todos los programas en ejecución y otra para verificar que estén cerrados. Después de esto manda una señal para dar de baja todos los daemons y cerrar el sistema.	Shutdown -h now	Apaga el sistema e forma inmediata
		Shutdown -r now	Reinicia el sistema en forma inmediata
		Shutdown -t [segundos] [acción]	Realiza la acción después de un determinado tiempo
		Shutdown -k	No apaga el sistema sólo envía un mensaje a todos los demás usuarios
		Shutdown -f [acción]	Evita que se ejecute un fsck al reinicio
		Shutdown -F [acción]	Fuerza que se realice un fsck al reinicio
		Shutdown -c	Cancela la instrucción shutdown

		Shutdown +m [acción]	Espera m minutos antes de realizar la acción.
--	--	----------------------	---

*Tabla 9. Comandos de apagado y reinicio de sistema*

Una de las principales tareas de un administrador es el manejo de cuentas de usuario, para ello debe contar con un plan que regularice y organice todas las claves, tanto en baja, alta o cambios de las mismas. Debe manejar sus password instruyéndolos en su correcto uso y proporcionándoles una clave fuerte, asignándoles permisos adecuados. La siguiente lista (tabla 10) son comandos para administrar usuarios:

Comando	Descripción		
Adduser	Agrega usuarios modo comando. Agrega de forma transparente con la posibilidad de seleccionar grupos, shell y demás características propias del usuario.		
Useradd	Agrega usuarios modo comando. Agrega el usuario con características más complicadas. Aquí se le agregan los privilegios.		
Userdel	Elimina usuario. Se puede agregar -r al término del comando para borrar de forma recursiva toda la información que compete a este usuario.		
Usermod	Modifica las propiedades individuales del usuario sin necesidad de borrarlo y crearlo de nuevo. Tiene vario tipos variables dependiendo del uso que se le dé.	-u	Modifica el UID del usuario
		-g	Modifica el grupo principal del usuario
		-G	Modifica los grupos adicionales del usuario
		-d	Modifica el directorio HOME del usuario
		-s	Modifica el shell por default
		-c	Modifica el comentario (campo de GECOS)
		-f	Asigna / modifica fecha de inactividad
		-e	Modifica / asigna fecha de caducidad de cuenta
		-p	Cambia el password del usuario
		-L	Bloquea el acceso de la cuenta (lock)
		-U	Desbloquea una cuenta bloqueada (unlock)
		-u	Modifica el UID del usuario
Groupadd	Agrega grupos administrando determinadas características.		
Groupdel	Elimina grupos.		
Groupmod	Modifica grupos cambiando las características que sean agregadas o borradas de los usuarios.		
Groups	Lista los grupos a los que pertenece un usuario.		

Passwd	Modifica propiedades de usuarios / grupos.
Kuser	Administra usuarios y grupos en modo gráfico.

*Tabla 10. Comandos para administrar usuarios*

Un administrador debe ver por la distribución del espacio en disco. Con el comando `df` se puede ver como se encuentra el disco duro particionado. Para prevenir ciertos inconvenientes se debe separar algún espacio libre para ciertas prioridades, de tal forma que si se satura un área de almacenamiento no se pierda en control de acceso al sistema para que los usuarios puedan depurar sus cuentas. Éste comando se puede usar dentro del directorio `/HOME` y saber cuanto espacio esta ocupando cada usuario. Para ver cuanto espacio ocupa cada `/HOME` de usuario se usa la modificación de `df` a `du`.

Linux cuenta con la facilidad de crear particiones sin la necesidad de salir del sistema o de requerir discos de instalación. Cuando se instala por primera vez el sistema se puede hacer de la misma manera, siguiendo los mismo pasos. Claro se debe tener cuidado de no borrar las particiones ya creadas a menos que se conozca lo que se esta haciendo. En Slackware existe la propiedad de poder crear particiones primarias y extendidas. El límite, como ya se dijo, de particiones primarias es de cuatro, pero dentro de una partición extendida se pueden crear un sin número de particiones, en ellas se pueden meter otros sistemas o dejarlo como espacio reservado para otro uso.

Para un servidor es conveniente formatear los discos con revisión de sectores dañados, aunque esta tarea puede llevarse mucho tiempo por las características del hardware y tamaño de la partición.

Ahora que se han creado las nuevas particiones, hay que añadirlas al documento que monta estas unidades automáticamente, sino cada vez que se deseen verlas será necesario editarlas y montarlas. Para ello se abre el documento de texto `/etc/fstab` (fig 7). En él se encuentran las particiones ya creadas en la instalación, como muestra la figura:

```
root@nilo:~# cat /etc/fstab
/dev/hda2      swap          swap          defaults      0    0
/dev/hda3      /             reiserfs      defaults      1    1
/dev/cdrom     /mnt/cdrom    iso9660       noauto,owner,ro 0    0
/dev/fd0       /mnt/floppy   auto          noauto,owner   0    0
devpts        /dev/pts     devpts        gid=5,mode=620 0    0
proc          /proc        proc          defaults      0    0
root@nilo:~# █
```

*Figura 7. Archivo fstab*

Al modificar el documento (fig 8) se debe agregar las líneas requeridas para el montaje, el resultado de tal modificación sería la siguiente:

```

/dev/hda2      swap          swap          defaults      0  0
/dev/hda3      /             reiserfs      defaults      1  1
/dev/hda5      /respaldo    ext3          defaults      0  0
/dev/hda6      /bases       ext3          defaults      0  0
/dev/cdrom     /mnt/cdrom   iso9660       noauto,owner,ro 0  0
/dev/fd0       /mnt/floppy  auto          noauto,owner   0  0
devpts        /dev/pts     devpts        gid=5,mode=620 0  0
proc          /proc        proc          defaults      0  0

```

Figura 8. Archivo fstab modificado

Nótese que el archivo tiene nuevas líneas denominadas particiones, mostrando la dirección donde se encuentra, el nombre que se les da, el formato su tipo de lectura y secuencia.

La siguiente tabla (11) explica las diferentes columnas que contiene el archivo ya modificado:

Columna	Descripción
/dev/hda#	Partición o dispositivo
/nombre	Punto de montaje en el sistema
Ext3, reiserfs, ext2,xfs,nfs	Tipo de sistema (formato)
defaults	Opciones de montaje asociadas al dispositivo
0	Sistemas que requieren dump
0	Orden en que son revisados los sistemas

Tabla 11. Descripción del archivo fstab

Además de adicionarlos al documento se debe crear el punto de montaje, para que se tenga acceso a él, esto es en la carpeta /mnt y aquí se deben crear las carpetas asignándoles el mismo nombre como se les llamó en el documento fstab.

La ventaja de Linux sobre otros sistemas, es que tiene la facilidad de reconfigurar el área de SWAP incrementándola si es necesario o disminuyéndola dependiendo de las necesidades. Este espacio se puede determinar de la siguiente manera:

- 1.-Determinar el espacio del archivo utilizando la siguiente formula:  
 (Tamaño de bloque) X 64Mb = 65536  
 Suponiendo que vamos a hacer un archivo de 64 MB con bloques de 1024 bites cada uno.
- 2.-Ahora se crea el archivo con el comando dd  
 ddif=/dev/zeroof=/swapfilebs=1024count=65536
- 3.-Indicar al sistema que este archivo será utilizado como swap  
 mkswap/swapfile
- 4.-activar el archivo swap para que comience a ser utilizado

Swapon/seapfile

5.-Para que se active al inicio del sistema hay que agregar el registro al archivo /etc/fstab  
/swapfileswapswapdefaults0 0

Si se tiene un área swap activada, ésta tiene que desactivarse antes de poner en funcionamiento el archivo:

Swapoff/dev/recurso

Con el comando free podemos ver si esta instalado y funcionando. Ahora se puede configurar el equipo para que inicie en modo gráfico, para esto no se hace ningún cambio hasta que se haga un respaldo de todos los documentos de configuración que se deban instalar, una vez hecho esto se inicia la configuración. El documento inittab contiene la información sobre el inicio y los niveles de inicio (ya explicado anteriormente), elegir el modo gráfico sustituyendo el número anterior por el respectivo al que se desea. Ahora se copia el archivo xorg.conf-isa por xorg.conf. Finalmente se reinicia el equipo y se vera como el equipo inicia normalmente pero cuando cargue la interfaz, cargará la gráfica pidiendo de igual manera que el modo texto nombre y password.

Los respaldos son una utilidad muy requerida para asegurar y salvaguardar los datos. Para crear respaldos se utiliza el comando tar, que sirve para empaquetar una estructura de directorios en forma recursiva, el cual tiene varias opciones, como se muestra a continuación (tabla 12)

tar [opciones] archivo.tar archivo.

Opciones	Descripción
-c	Crea un nuevo archivo tarareado
-v	Verbose, imprime en pantalla todo aquello que esta afectado.
-f	Indica al sistema que afectará a un archivo
-x	Extrae el contenido de un archivo tar.
-t	Lista todos los archivos contenidos en el archivo tar.
-u	Update, actualiza todo el contenido en el archivo tar comparándolo con los directorios en donde fue creado, modificando todos sus archivos contenidos por las nuevas actualizaciones.
-w	Pide una confirmación cada vez que el archivo vaya a ser tarareado o viceversa
-d	Hace una búsqueda de todos los archivos y directorios que hayan sufrido alteraciones después de haber sido creado el archivo tar.

Tabla 12. Opciones de compresión el comando tar

Gzip crea una compresión del archivo tar, manejando nueve niveles de compresión (1 el más bajo y 9 el más alto), agregando al final de este archivo la extensión gz. Su sintaxis es:

gzip [opción] nombre\_archivo.tar

Que al igual que tar tiene diferentes opciones (tabla 13):

Opción	Descripción
-n	Indica el nivel de compresión
-d	Descomprime un archivo gzipeado
-f	Fuerza la compresión o descompresión del archivo gzipeado sin importar permisos o propiedades.
-t	Hace una revisión de integridad del archivo, mandando un mensaje en caso de que no este completamente integro.
-l	Muestra las propiedades del archivo gzipeado.

*Tabla 13. Opciones de compresión del comando gzip*

Para terminar este módulo, hay varios modos de acceder al equipo vía remota, por ejemplo telnet y ssh vía texto, y VNC en modo gráfico. También se definió que sistema de transferencia de archivos se desea que ocupen los usuarios, entre los conocidos están: sistema NFS, Samba, FTP y SCP entre otros.

## Módulo III: Editores para la creación de páginas web

<b>Temario</b>	
1. Introducción.	
2. Etiquetas (Markup Tags)	
3. Formato de caracteres.	
4. Ligas.	
	5. Imágenes.
	6. Tablas.
	7. Elementos avanzados.

Este módulo comprende el uso de 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 software libre

Como introducción se dio un vistazo al origen de World Wide Web. WWW como es llamada comunmente, fue creada en 1989 por Tim Barners Lee en el Instituto Europeo de Investigación de la Física de Partículas (CERN) en Ginebra (Suiza), aunque hasta el año 1990 no recibió el nombre definitivo. Surgió como sistema para compartir los trabajos realizados por los físicos del Instituto a través de un entorno hipertexto basado en una arquitectura cliente-servidor, y se finalizó con la creación de un tipo de servidor llamado WWWLIB proporcionando una nueva forma de tratar la información depositándose en dominio público en 1991.

En 1992 ya se disponía del primer cliente de World Wide Web y su difusión se realizaba mediante FTP anónimo, distribuyéndose con gran velocidad y tomando mucha popularidad en el mundo de la red Internet.

En 1993, WWW recibió un gran impulso con el desarrollo del Mosaic. En 1994 se celebró la primera conferencia de World Wide Web y ya se encontraban registrados, de manera oficial, unos 1500 servidores de este sistema. Actualmente, se calcula que existirán unos 12000 servidores.

El Sistema de Nomenclatura de Dominios (DNS), permite otorgar direcciones a todos los usuarios de Internet, basándose en unas sencillas normas (parecidas al sistema de identificación de correo electrónico). Con este sistema, todos los ordenadores quedan identificados.

El lenguaje HTML (Hyper Text Makeup Language) permite integrar en un mismo documento textos, imágenes, sonidos y video (multimedia), a la vez de acceder a otros documentos del mismo tipo mediante los "enlaces de hipertexto" (hyperlinks). HTML es una aplicación del lenguaje estándar para la creación de todo tipo de hipertextos: SGML (Standard Generalized Markup Language). Los documentos HTML soportan los caracteres del alfabeto ISO Latin-1 y no distingue entre mayúsculas y minúsculas exceptuando en las direcciones URL. La última versión de este estándar es la 3.0.

HTML emplea URL (Uniform Resource Locator) para especificar uniformemente la dirección hacia la que apuntan los enlaces. La estructura de una dirección URL es: servicio://máquina/ruta. Servicio puede ser cualquiera de los que están disponibles en Internet (http ---otra página Web, ftp, gopher, telnet, news o file ---fichero local).

La máquina emplea el sistema TCP/IP para componer su dirección: nombre, ciudad, provincia y país. La ruta nos señala la posición en la que se encuentra el archivo deseado dentro de la máquina.

Los documentos HTML emplean etiquetas para indicar los elementos a insertar en ellos, así como la estructura y presentación de los mismos. Las etiquetas delimitan elementos tales como cabeceras, tablas, listas, imágenes... Todos los elementos de HTML están delimitados por las etiquetas de comienzo " <etiqueta> " y, las de final " </etiqueta> " aunque no todos poseen esta última (hay comandos que sólo poseen etiqueta de comienzo). Si se quiere poner un comentario dentro del documento, tendrá que introducirse entre las etiquetas "<!--" y "-->"

Además del nombre indicativo del tipo de elementos que son, llevan unos atributos que pueden variar las características de los mismos. Para ser reconocidos como tales, los elementos tienen que ir inmediatamente detrás del delimitador de comienzo de etiqueta "<". Los atributos consisten, normalmente, en un nombre de atributo, un signo igual y un valor. Tienen que aparecer siempre en la etiqueta de comienzo y detrás del nombre del elemento, pudiendo existir varios atributos para el mismo elemento.

Los documentos HTML se distinguen porque están comprendidos entre las etiquetas <HTML> y </HTML>. Incrustadas dentro de estas etiquetas se tienen otras dos partes principalmente: la cabecera (delimitada por las etiquetas <HEAD> y </HEAD>) y el cuerpo

(delimitado por las etiquetas <BODY y </BODY>) del documento; en esta última se puede definir una imagen de fondo para mostrar. Estas partes, a su vez, poseen sus propios elementos.

Esos elementos están conformados por títulos, párrafos, listas, texto preformateado, indexación, direcciones de correo, rompimiento de línea, separadores, etc.

Todos estos caracteres dentro del código HTML pueden ser formateados según como quieren que se presenten en la pantalla.

Un documento HTML (en su presentación) está compuesto por títulos, párrafos, saltos de línea, etc., la tabla 14 muestra las etiquetas básicas contenidas dentro de un código:

Etiqueta	Descripción
<html>	Define un documento del HTML
<body>	Define el cuerpo del documento
<h1> to <h6>	Define entre el tamaño 1 a 6 de cabecera
<p>	Define un párrafo
 	Inserta una sola línea rotura
<hr>	Define una regla horizontal
<!-->	Define un comentario

Tabla 14. Etiquetas en HTML

Para definir el aspecto que mostrará ante el usuario el formato de texto se hace mediante etiquetas, las principales son (tablas 15,16 y 17):

Etiqueta	Descripción
<b>	Define el texto en negrilla
<big>	Define el texto grande
<em>	Define el texto acentuado
<i>	Define el texto del itálico
<small>	Define el texto pequeño
<strong>	Define el texto fuerte
<sub>	Define subscripted el texto
<sup>	Define el texto superscripted
<ins>	Define el texto insertado
<del>	Define el texto suprimido
<s>	Desaprobado. Utilice el < del > en lugar de otro
<strike>	Desaprobado. Utilice el < del > en lugar de otro
<u>	Desaprobado. Utilice los estilos en lugar de otro

Tabla 15. Etiquetas del formato texto

Etiqueta	Descripción
<code>	Define el texto de código de computadora
<kbd>	Define el texto del teclado
<samp>	Define código de computadora de la muestra
<tt>	Define el texto del teletipo
<var>	Define una variable
<pre>	Define el texto preformateado
<listing>	Desaprobado. Utilice < pre > en lugar de otro
<plaintext>	Desaprobado. Utilice < pre > en lugar de otro
<xmp>	Desaprobado. Utilice < pre > en lugar de otro

*Tabla 16. Etiquetas de salida de computadora*

Etiqueta	Descripción
<abbr>	Define una abreviatura
<acronym>	Define siglas
<address>	Define un elemento de la dirección
<bdo>	Define la dirección del texto
<blockquote>	Define una cita larga
<q>	Define una cita corta
<cite>	Define una citación
<dfn>	Define un término de la definición

*Tabla 17. Citas y etiquetas de definición*

Dentro de las ligas existen cuatro tipos: rutas, URL's, anclas y los mail-to. Las rutas se distinguen por utilizar direcciones relativas a distintos documentos o archivos definidos por el programador. Estos programas son anexados de manera manual a la carpeta de acceso que tienen los usuarios, sólo colocando una liga a ellos. Las URL's como su nombre lo dice se usan para ligar la página a otras direcciones de Internet que estén relacionadas con la información. Las anclas son usadas para ligar textos extensos dentro de la misma página. Es como un índice que nos ligará a cualquier lugar dentro de la misma. Los mail-to conectan al administrador de correo previamente instalado, o usando el default, para poder enviar correo al administrador de la página.

Estas ligas pueden ser implementadas con imágenes, que en lugar de hacer que la liga aparezca como texto subrayado, sea una imagen o combinados.

Dentro del código se pueden incluir tablas, para mantener un orden en la visualización final. Con estas tablas se puede anexar imágenes consecutivas a un texto, que sin ellas no sería posible, además que combinado con php se pueden generar tablas dinámicas, que en bases de datos son convenientes, pero eso se tratará más adelante. Las tablas se definen con la etiqueta < table >. Una tabla se divide en filas (con la etiqueta del < tr >), y cada fila se divide en las células de datos (con la

etiqueta del < td >). El td de las letras está parado para los "datos de la tabla," que es el contenido de una célula de datos. Una célula de datos puede contener el texto, imágenes, listas, párrafos, formas, reglas horizontales, las tablas, etc. Las etiquetas usadas para eso son (tabla 18):

Etiqueta	Descripción
<table>	Define una tabla
<th>	Define un jefe de la tabla
<tr>	Define una fila de la tabla
<td>	Define una célula de la tabla
<caption>	Define un subtítulo de la tabla
<colgroup>	Define grupos de columnas de la tabla
<col>	Define los valores de la cualidad para unas o más columnas en una tabla
<thead>	Define una cabeza de la tabla
<tbody>	Define un cuerpo de la tabla
<tfoot>	Define un pie de la tabla

*Tabla 18. Etiquetas que definen las propiedades de las tablas*

Existen tres tipos de listas que se pueden incorporar a la página: listas pérdidas, desordenadas, definición. Las listas pérdidas son aquellas que para marcarlas usan un número de identificación, en cambio las listas desordenadas, como su nombre lo dice no tienen ninguna marca personal, van denotadas con un punto. Las listas por definición, utilizan la definición de sí mismas para ordenarlas (tabla 19).

Etiqueta	Descripción
<ol>	Define una lista pedida
<ul>	Define una lista desordenada
<li>	Define un artículo de la lista
<dl>	Define una lista de la definición
<dt>	Define un término de la definición
<dd>	Define una descripción de la definición
<dir>	Desaprobado. Utilice la < UL > en lugar de otro
<menu>	Desaprobado. Utilice la < UL > en lugar de otro

*Tabla 19. Etiquetas que definen las propiedades de las listas*

Los formularios son los elementos que permiten que el usuario incorpore la información (como campos del texto, campos del textarea, menús drop-down, botones de radio, checkboxes, etc.) en un formulario, la pregunta que se hace con esa información es "¿Qué se deseamos hacer con ella?", se tiene que dar la opción al usuario de escoger entre cambiarla, borrarla o enviarla (guardarla). Con los botones que se le anexan, se le dará la oportunidad que él decida que hacer con ella. La tabla 20 muestra las etiquetas usadas en un formulario:

Etiqueta	Descripción
<form>	Define una forma para la entrada del usuario
<input>	Define un campo de la entrada
<textarea>	Define un texto-área (un control multilínea de la entrada de texto)
<label>	Define una etiqueta a un control
<fieldset>	Define un fieldset
<legend>	Define un subtítulo para un fieldset
<select>	Define una lista seleccionable (una caja drop-down)
<optgroup>	Define a grupo de la opción
<option>	Define una opción en la caja drop-down
<button>	Define un botón de empuje
<isindex>	Desaprobado. Utilice la etiqueta < INPUT > en lugar de otro

*Tabla 20. Etiquetas que definen las propiedades en formularios*

Pero para que funcione se debe meter todo dentro de un formulario, para que dependiendo de la elección el programa haga lo suyo. En la etiqueta del formulario, se le indicara a donde se debe de enviar esa información, ya sea correo o a una BD. La siguiente etiqueta indica el inicio y fin de un formulario:

```
<form method="post, get" action="accion_a_realizar">
```

```
Demás_etiquetas
```

```
</form>
```

En un formulario de HTML se puede enviar la información de distintas maneras, una de ellas es con el método GET que envía la información a la vista en la barra de direcciones, y el método POST que oculta la información enviada.

Cuando se hacen los formularios es lógico dejar los cuadros de texto que el usuario va ocupar, el administrador deberá colocar los necesarios para recopilar la información necesaria. Este es el método de inserción de cuadros de texto y de captura:

```
<input type="text" name="" size="" value="" maxlength="">
```

La descripción de lo anterior indica que capturará texto (type), y la almacenará en una variable (name) con cierto tamaño (size). También indica la longitud máxima de caracteres que pueden ser insertados (maxlength).

Existen otras etiquetas que son utilizadas dentro del mismo formulario. Ya es bien conocido el password, que al igual que la inserción de texto se almacenan passwords, pero mantenerlos secretos a la vista de los demás se cambia la etiqueta type="text" por type="password", sustituyendo los caracteres por asteriscos.

En algunas oportunidades se usan los INPUT RADIO y los INPUT CHECKBOX que son los iconos de selección simple o múltiple.

Otra etiqueta es la INPUT BUTTON que es similar a RESET y a SUBMIT, pero esta es una etiqueta programable. Las anteriores se encuentran ya definidas, esta no; hay que definirla, aplicarle una función existente que deberá ejecutar, por ejemplo una función. Existen otras etiquetas ya definidas como: INPUT HIDDEN, INPUT FILE, INPUT IMAGE, TEXTAREA, ETC.

Una etiqueta muy útil es SELECT que lo que hará será enlistar, con la propiedad de poder hacer una selección de lo que presenta (dependiendo de la selección, realizara una acción programada, invisible para el usuario).

Los frames son subdivisiones de la pantalla, es como si fueran varias ventanas en una misma lo que da una presentación más profesional a las aplicaciones, con este método se puede hacer consultas en la misma ventana sin tener que abrir otra como por ejemplo cuando se hacen ligas comunes. Con sólo indicarle a que frame de la ventana principal se quiere que aparezca el link seleccionado. Existen de dos tipos: los frames simples o los enlazados. Los simples sólo tienen un FRAMESET dentro del código, y los enlazados pueden tener dos o más dentro de uno. Las etiquetas (tabla 21) usadas en frames son:

Etiqueta	Descripción
<frameset>	Define un sistema de frames
<frame>	Define una ventana secundaria (un frame)
<noframes>	Define una sección del noframe para los browsers que no manejan frames
<iframe>	Define una ventana secundaria en línea (frame)

*Tabla 21. Etiquetas que definen las propiedades de los frames*

Se pueden insertar imágenes, para amenizar más la navegación y presentación que se brinda, se recomienda utilizar tablas para mantener un orden en la página, estas son las etiquetas que se usan para una imagen (tabla 22).

Etiqueta	Descripción
< img >	Define una imagen
< map >	Define un mapa de imagen
< área >	Define un área clickable dentro de un mapa de imagen

*Tabla 22. Etiquetas que definen las propiedades de las imagenes*

Existen otras etiquetas de gran utilidad pero poco conocidas, la tabla 23 las describe:

Etiqueta	Descripción
<style>	Define un estilo
<link>	Define una referencia del recurso
<div>	Define una sección en un documento
<span>	Define una sección en un documento
<font>	Desaprobado. Utilice los estilos en lugar de otro
<basefont>	Desaprobado. Utilice los estilos en lugar de otro
<center>	Desaprobado. Utilice los estilos en lugar de otro

*Tabla 23. Etiquetas que dan formato a la página*

Para darle un toque personal se pueden agregar etiquetas de información, esto es, dar información sobre lo que se está viendo, estas son sus etiquetas:

Etiqueta	Descripción
<head>	Define la información sobre el documento
<title>	Define el título del documento
<base>	Define un URL de la base para todos los acoplamientos en una página
<link>	Define una referencia del recurso
<meta>	Define la información del meta
<!DOCTYPE >	Define el tipo del documento. Esta etiqueta va antes de que la etiqueta del comienzo del < HTML >.

*Tabla 24. Etiquetas que definen la cabecera del programa*

Prácticamente se vio lo básico con respecto al lenguaje HTML, dando cavidad al usuario de investigar más sobre el tema, y crear páginas más competitivas y profesionales.

## Módulo IV: Administración de servidores WWW con Linux

Temario	
1.Introducción a los servidores www	5. Incorporación de módulos.
2.Instalación del servidor www.	6. Accesos restringidos.
3.Configuración del servidor.	7. Registro de accesos.
4.Ejecución del servidor	8. Manejo de sitios virtuales.

Un servidor WWW es un programa encargado de ofrecer comunicación mediante el protocolo http (HyperText Transfer Protocol). La forma en que se transmite información por la red con este protocolo es de cliente-servidor, en el se publican servicios electrónicos en los cuales están los populares email, chat, etc... . El cliente http es quien consulta estos servicios para su uso personal.

En la figura 9 se muestra la arquitectura básica de un servidor WWW:

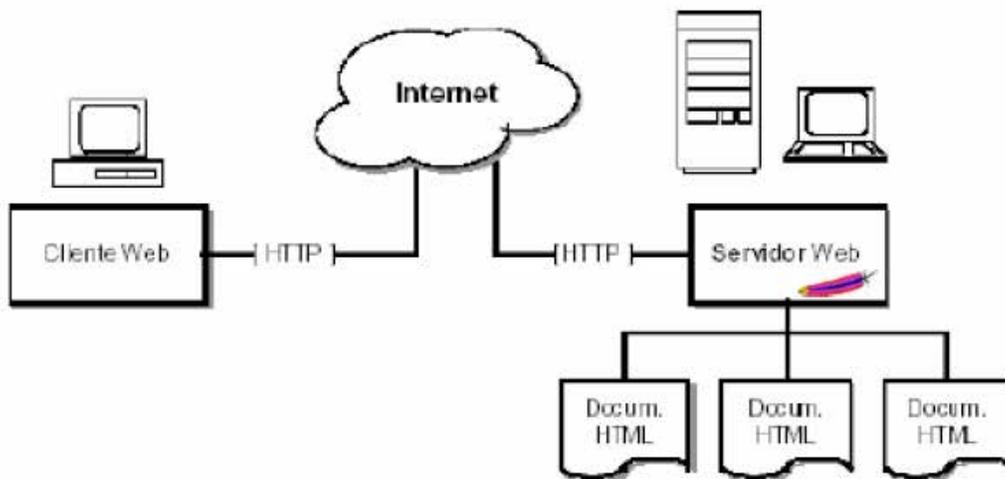


Figura 9. Arquitectura de la WWW

La forma en que trabaja un servidor es, cuando un cliente http abre una conexión, el servidor manda un acknowledge, notificando que ha abierto una sesión. El cliente solicita un recurso que ofrece el servidor (request message), y como respuesta el cliente recibe un response message el cual contiene el servicio solicitado. Esta es la estructura básica de una conexión cliente-servidor.

Entre los servidores WWW más populares se encuentran:

Servidor	Distribuidor
Internet Information Server (IIS)	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

Tabla 25. Servidores Web

Cuando uno va a decidir que servidor es el que desea instalar debe tener en cuenta ciertos puntos. El administrador debe tomar en cuenta los servicios que va a brindar. Para ello su funcionalidad es primordial, puesto que es la manera en que va a tener rentabilidad su servidor. Ver para que plataformas esta destinado o cuales puede soportar.

Una cosa que no puede pasar por alto, es la cantidad de usuarios que podrán acceder a él, esto conlleva al número de conexiones que puede tener habilitadas y el número de transacciones por minuto que puede soportar. Contando con esto el costo computacional por transacción y la proyección a un crecimiento futuro.

Como costo adicional, el soporte que se le debe de dar para la tecnología seleccionada. Y el análisis por el retorno de inversión (ganancia).

En este caso en el diplomado se ocupó el servidor Apache ya que es robusto (soporta un gran número de transacciones), es configurable y con esta ventaja ofrece mayor seguridad. Es adaptable a diferentes plataformas. Brinda soporte para proxys y granjas de servidores.

Soporta varios lenguajes de programación, además de soportar Scripting Languages incluyendo su propio código. Soporta la configuración para accesos restringidos y encriptamiento de información SSL, además de que es gratuito.

En el mercado, los administradores se deciden más por apache, ya que por sus cualidades es más seguro. Apache se puede obtener gratuitamente de la página [www.apache.org](http://www.apache.org). Existen dos modos en los que se puede instalar apache, uno por el camino fácil (rpm) o por el mejor camino. En el modo fácil sólo se ejecuta el paquete y el sólo se configura, pero como se desea tener un mejor control lo se compila, de modo que se pueda dar soporte a diferentes lenguajes de programación. Para compilarlo (que es el mejor modo), se inicia con la siguiente secuencia de instrucciones:

```
tar zxvf apache-1.3.27.tgz
cd apache-1.3.27
```

**nota:** si se desea dar una configuración más adecuada se abre archivo de texto que tiene esa información y que esta contenido en la carpeta actual. Si se prefiere que se configure con las directivas actuales de la máquina, se escribe a continuación el código siguiente:

```
./configure
make
```

```
make install
```

En apache existen diferentes directivas de configuración. Apache es administrado por más de 200 directivas las cuales son configuradas para dar el respectivo servicio. Estas directivas se dividen en tres grupos: Global Environment, Main Server y Virtual Server.

La sección Global Environment como su nombre lo dice, administra todas las directivas generales de apache. Dentro de esta división están las directivas acordes al módulo, entre ellas están:

**Server Name:** define el nombre del servidor, el cual será utilizado cuando se construyan los URL's , y cuando el usuario requiera de otros servicios web. También es usada en la directiva ServerConfig y VirtualHost.

Server Type: antes era usada para definir el esquema mediante el cual opera apache, sus posibles opciones eran standalone e inetd. Fue eliminado en la versión 2.0 de apache, debido a que la opción inetd requería usar el superdemonio inetd o xinetd para generar un proceso de httpd cada vez que se generara o abriera una conexión.

User: define el usuario ID mediante el cual operara apache. Por default se encuentra en User#-1

Group: al igual que User define el grupo mediante el cual apache operara. Por default se encuentra en User#-1.

Port: antes de ser reemplazado por Listen en la versión 2.0, era utilizado para definir porque puerto operara el servidor web. Su valor por default era el 80.

ServerAdmin.: esta directiva define el correo electrónico del administrador del servidor web e indica la dirección mail en los mensajes de error que arroje la página. Para que funcione debe estar habilitada la directiva ServerSignature Email.

DocumentRoot: define la ruta absoluta donde se encontraran los archivos html que se van a publicar. Su ruta default es /usr/local/apache2/htdocs, aunque varia según la versión o decisión del mismo administrador. Esta directiva también es usada por ServerConfig y VirtualHost.

ServerRoot: define la ruta absoluta donde los directorios logs y conf podrán ser encontrados. Su valor por default es /usr/local/apache2, aunque puede variar. Al igual que la anterior se usa en las mismas directivas.

MinSpareServers: define el número mínimo de procesos que son mantenidos en spare. Si el número de conexiones es menor que el valor de esta directiva, se levantan los daemons necesarios. Su valor por default son 5.

MaxSpareServers: es similar al anterior con la diferencia de marcar el máximo número de procesos en spare. Si sobrepasa el número de conexiones en espera elimina los daemons necesarios, siempre y cuando no estén ocupados.

StartServers: define el número máximo de servidores que iniciaran cuando se levante apache. Su valor por default es 5.

MaxClients: define el número máximo de procesos servidores corriendo en el servidor. Su valor por default es 20.

ErrorDocument: en caso de que la petición del usuario no pueda ser atendida, esta directiva (según su configuración) enviara las siguientes respuestas.

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

Los siguientes son los códigos de mensajes más usados en la red:

200 OK  
102 Processing  
204 No content  
400 Bad Request  
404 Not Found  
403 Forbidden  
405 Method not allowed  
505 HTTP version not supported  
507 Insufficient storage

PidFile: define el lugar donde se almacenará el PidFile que contiene el PID, número que identifica el http padre.

Este documento se encuentra en la misma carpeta donde residen los Logs o bitácoras. Dos documentos importantes son los access.log y el error.log. El primero registra los accesos al servidor, y el segundo registra los errores al intentar conectarse al servidor.

ErrorLog<filename>: define el nombre del archivo en el cual el servidor almacenara la bitácora de los errores que puedan presentarse..

LogLevel: define el nivel de mensajes de error que serán registrados en la bitácora. Los tipos de errores son: Emerg, Alert, Crit, Error, Warn, Notice, Info, Debug. Cuando se especifica el nivel, se reportan los mensajes de todos los niveles incluyendo el nivel definido. Si la bitácora es manejada por syslog el mensaje notice, será almacenado en la bitácora.

Esta fue la figura 10 describe lo ya mencionado:

Level	Description	Example
emerg	Emergencies - system is unusable.	"Child cannot open lock file. Exiting"
alert	Action must be taken immediately.	"getpwuid: couldn't determine user name from uid"
crit	Critical Conditions.	"socket: Failed to get a socket, exiting child"
error	Error conditions.	"Premature end of script headers"
warn	Warning conditions.	"child process 1234 did not exit, sending another SIGHUP"
notice	Normal but significant condition.	"httpd: caught SIGBUS, attempting to dump core in ..."
info	Informational.	"Server seems busy, (you may need to increase StartServers, or Min/MaxSpareServers)..."
debug	Debug-level messages	"Opening config file ..."

Figura 10. Mensajes LogLevel

CustomLog<filename>: define el nombre del archivo en el cual se almacenaran los accesos.

LogFormat "format string"<filenick>: define el tipo de información que se almacenara en la bitácora. La tabla 26 muestra los "format string" definidos:

%h	Registra la IP del cliente, hostname.
%l	Si el daemon identd corre en el cliente, reporta la información que el identd devuelva.
%u	Si se requiere de un username y password para acceder, en este campo se registra.
%t	Fecha y hora del request en el formato [dia/mes/a_o:hora:minuto:segundo tzoffset].
\"%r\"	Recurso solicitado por el cliente, entre comillas, request.
%>s	Un código de tres dígitos donde se muestra el valor devuelto al cliente, status.
%b	El número de bytes devueltos exceptuando encabezados.

*Tabla 26. Tipos de format string*

Existen una serie de LogFormat establecidos:

- Common- Mostrados en la lista anterior.
- Referred- Orientado a llevar un registro de las rutas de los usuarios dentro del sitio.
- Agent- Orientado a registrar el nombre de los navegadores de los clientes
- Combined- Combina las características de los anteriores.

HostNameLookup [on/off]: habilita la resolución de nombres en el DNS cuando se establece una conexión.

La sección Main Server controla las directivas del servidor principal o estándar de apache. Y la sección Virtual Server administra las directivas de los procesos de apache con controlan las ip's o nombres de dominios.

Se debe verificar la configuración, cuando todo quede correcto, guardar los cambios e iniciar el servidor. Para corroborar que todo este funcionando correctamente, se abre un navegador del equipo y en la dirección web se escribe localhost, si aparece la ventana de bienvenida de apache estará configurado bien el servidor.

El control de acceso es una parte fundamental en cualquier servidor. Apache ofrece la funcionalidad de ACL, con lo cual es factible determinar las direcciones IP's de los usuarios que tendrán acceso a los recursos del servidor web.

La realización de estos niveles de acceso depende de las directivas Allow, Deny, y Order. Order define el orden de la implementación de las directivas anteriores. Deny define los hosts que no tendrán acceso, y Allow define los que si tendrán acceso al recurso.

El orden en que sea leído será concluyente, de manera que si el orden en que son ejecutados es Allow-Deny significará " que todo lo que no este explícitamente permitido esta prohibido", si es el caso contrario Deny-Allow será "que todo lo que no esta explícitamente prohibido esta permitido".

En Apache existen módulos que son los encargados de dar funcionalidad adicional al servidor, agregando más directivas. Existen de dos tipos: los estáticos y los dinámicos. Los estáticos se agregan de manera automática al instalarse apache, y los dinámicos se agregan durante el ambiente productivo, según las necesidades.

Un host virtual nos ayudara mucho al tener que brindar un servicio. Es más fácil recordar una dirección web que una IP cuando se desea recurrir a un servicio. No sólo tiene esta función, sino que además, puede utilizarse para mantener una seguridad y una barrera más contra ataques hackers. Esto se puede implementar de la siguiente manera: Si uno desea que sus usuarios se conecten por el puerto 1000, sólo se debe crear un servidor host con la conexión a ese puerto. Cuando un intruso desea conectarse a nuestro servidor lo hará por el puerto más común que es el 80 u 8080, siendo ese el problema que ese puerto esta desactivado.

También se pueden tener varios servidores virtuales y brindar un deferente servicio o presentación en un sólo servidor real. Estos puntos son implementados en el archivo de configuración de apache llamado httpd.conf.

## Módulo V: Programación con PHP

Temario	
1. Introducción al lenguaje PHP	4. Como diseñar una aplicación CGI
2. Herramientas elementales	5. Algunas funciones útiles
3. Common Gateway Inteface	6. PHP y la programación orientada a objetos

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

Los primeros conocimientos adquiridos 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, existen cuatro:

- `<? ... ?>`
- `<?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, y el punto anterior a este para los que les gusta programar con JavaScript.

Algunos comandos son parecidos a otros lenguajes; para los que trabajaron con lenguaje C es más fácil relacionarlo.

Los comentarios son muy útiles, ya que con ellos se pueden hacer anotaciones dentro del código para mantener un orden y saber para que sirve cada bloque de código, si éste es muy extenso. Para poder marcarlos se utilizan las siguientes sintaxis (tabla 27):

Sintaxis	Descripción
/* Aquí va el comentario */	Comentario de varias líneas. El contenido está dentro de estos símbolos, el símbolo final va al término del comentario total no de cada línea.
// Aquí va el comentario	Comentario de una sola línea. Se utiliza cuando el comentario no es muy extenso. Sólo afecta al texto que se encuentra delante de él sin salto de línea.
# Aquí va el comentario	Igual al anterior, tiene el mismo resultado. Si se desea cubrir más líneas es mejor usar la primera sintaxis.

*Tabla 27. Sintaxis de comentarios*

La sintaxis de un código PHP, como se mencionó antes, va insertado dentro de un código HTML, un ejemplo sería el siguiente:

```
<html>
<head>
<title>Ejemplo de código PHP</title>
</head>
<body>
<!--Probando comentarios en PHP-->

<h4>Algo de código HTML</h4>
<?
/*Este ejemplo tendrá la misma salida que
el ejemplo anterior*/
    print "Esto es PHP";
?>
<br>
<b> Más de HTML </b>
<br>
```

```

<?
//Esta línea de texto es un comentario de una sola línea
    print ("Esto también es PHP");
#Esto también es un comentario
?>
</body>
</html>

```

Como se observa en el código anterior existe la posibilidad de insertar todos los códigos PHP que se necesiten, con la aclaración que se debe abrir y cerrar cada bloque de PHP si se desea tener varios en un mismo documento HTML. Esta es su sintaxis básica. También como caso curioso dentro de un código PHP se puede meter código HTML, por ejemplo para la generación de tablas dinámicas la sintaxis sería la siguiente:

```

<html>
<head><title>Código HTML dentro de PHP</title></head>
<body>
<table border="1">
    <tr>
    <?
        print "<td>Esto es código PHP</td>";
    ?>
    </tr>
</table>

```

Este código lo que hace es imprimir el texto dentro del print insertado en una tabla. Con un ciclo de repetición se pueden hacer que cada vez que imprima el contenido lo haga dentro de una tabla.

En PHP las variables que se ocuparon en la programación. Van a ir precedidas del símbolo \$. Al igual que JavaScript, PHP es tipeado, esto quiere decir que no se necesita especificar el tipo de variable antes de poder almacenar algún dato. Dentro del mismo código se pueden cambiar el valor de la variable, sin que esto afecte su función.

Existen 4 tipos de datos en PHP: los boolean o boléanos que sólo 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 encuentren dentro de comillas dobles serán evaluadas.

Los enteros se dividen a su vez en a tipos: los positivos, los negativos, octales y hexadecimales. Los doubles son representados por números con decimales.

Las cadenas pueden contener dentro de su contexto un carácter especial, para poder ser evaluados se antepone diagonal invertida (\), con los que se cuenta son los siguientes (tabla 28):

Carácter especial	Descripción
\n	Nueva línea. Insertara una nueva línea, o llamado salto de línea comúnmente.
\r	Retorno de carro sobre la misma línea entregara una salida, eliminando lo anterior
\t	Tabulador horizontal. Insertara un tabulador mostrándolo en pantalla.
\\	Diagonal invertida
\\$	Signos de pesos
\"	Doble comilla

*Tabla 28. Caracteres especiales*

Cuando se utilizan comillas simples, la cadena no será evaluada, por lo cual los anteriores caracteres no tendrán efecto.

Cualquier expresión es una serie de caracteres o funciones, que de acuerdo a su sintaxis entregara un valor. Un tipo muy común de estas expresiones son las comparaciones(0-1 o TRUE-FALSE). Para hacer estas expresiones se ocupan operadores (tabla 29), existen 5 tipos de ellos:

Operador y Grupo	Descripción	Ejemplo	
+	Operadores aritméticos	Suma	$\$a + \$b$
-		Resta	$\$a - \$b$
*		Multiplicación	$\$a * \$b$
/		División	$\$a / \$b$
%		Módulo	$\$a \% \$b$
=	Operadores de asignación	Asignación	$\$a = \$b$
+=		Suma y Asignación	$\$a += \$b (\$a = \$a + \$b)$
-=		Resta y Asignación	$\$a -= \$b (\$a = \$a - \$b)$
*=		Multiplicación y Asignación	$\$a *= \$b (\$a = \$a * \$b)$
/=		División y Asignación	$\$a /= \$b (\$a = \$a / \$b)$
%=	Módulo y Asignación	$\$a \% = \$b (\$a = \$a \% \$b)$	
==	Operadores relacionales	Igualdad	$\$a == \$b$
!=		Distinto	$\$a != \$b$
<<		Menor que	$\$a << \$b$
>		Mayor que	$\$a > \$b$
<=	Menor o igual que	$\$a <= \$b$	
&&	Operadores lógicos	AND	$\$a \&\& \$b$
		OR	$\$a    \$b$
!		NOT	$!\&a$

++	Operadores especiales	Incremento	\$a++ (postincremento) ++\$a (preincremento)
--		Decremento	\$a-- (postdecremento) --\$a (predecremento)
(tipo)expresión		Cast	\$a = (int) \$b
.		Concatenación de cadenas	\$a = "cad1" . "cad2"
()		Agrupación de expresiones	\$a = (\$a + \$b) * \$c
[]		Acceso a elementos de arreglo	\$a = \$arr[7];
New		Instanciación de objetos	\$a = new AlgunaClase;
->		Acceso a variables y métodos	\$a = \$obj->var1

Tabla 29. Operadores lógicos y de asignación

Cabe mencionar que los operadores de asignación se leen de derecha a izquierda. Los anteriores operadores tienen un orden de precedencia, así para no cometer errores de lectura, enlistamos la precedencia (tabla 30):

Operador	Número de operandos	Operador	Número de operandos
()	Unario	< > <= >=	Binarios
new	Unario	== !=	Binarios
[]	Binario	&&	Binario
++ --	Unarios		Binario
* / %	Binarios	? :	Ternario
+ - .	Binarios	= += -= *= /= %=	Binarios

Tabla 30. Descripción de operadores de asignación

Las sentencias es una llamada a una función específica o un ciclo. 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; existen los ya clásicos en lenguajes de programación if, for, while, do while, switch también llamados ciclos de repetición o evaluación.

Las sentencias if, evalúan su información, si devuelve un TRUE ejecuta lo que contiene, sino hace lo que sigue. La sentencia for crear un ciclo delimitado por las veces que asignemos, al inicializar la variable se compara, e incrementa, paso siguiente ejecuta la sentencia contenida en él. La sentencia while, evalúa su variable si devuelve un TRUE continua sino salta al siguiente código.

La sentencia do-while es similar a la anterior con la diferencia que este ejecuta al menos una vez su sentencia y después evalúa.

La sentencia switch sirve para comparar una variable con una serie de valores distintos. En caso de que no sea ninguno de estos valores podremos definir un valor default y darle una sentencia y simplemente salir del switch. Cuando una variable entra es evaluada, si queremos que termine de evaluarla cuando salga de su elección de sentencia o CASE deberemos poner al final de este un BREAK para que termine de ejecutar el switch.

Además de la expresión BREAK existen otras más con un uso en particular: La sentencia CONTINUE es similar a la anterior, con la excepción de forzar a terminar, este fuerza a continuar la siguiente iteración saltando cualquier código intermedio. RETURN se usa para provocar la salida del método actual, esto es, regresará al código donde fue evaluada. También tiene la facilidad de devolver o no un valor sólo colocando el valor seguido de el (RETURN valor;).

Una función es un bloque de código que puede ser ejecutado tantas veces sea requerido en diferentes partes del programa. Una función se define con la siguiente sintaxis:

```
function nombreFuncion( /* parametros */)
{
...
cuerpo de la función
...}

```

Los argumentos o parámetros son entradas de datos en una función. Estos serán evaluados dentro y serán enviados ya sea a pantalla, según la función, o retornados para seguir siendo evaluados fuera de ella.

Al crear funciones se debe de fijar que variables serán de uso exclusivo por la función y cuales por el código externo. Algunas funciones ya se encuentran definidas para el manejo de variables, estas son las básicas, aunque existen más:

Función	Descripción
gettype	Determina el tipo de dato de una variable. Regresa los siguientes valores: <i>integer, double, string, array, object, unknow type</i> .
settype	Pone el tipo de dato de una variable. Regresa los siguientes valores: <i>array, double, integer, object o string</i>
isset	Determina si existe una variable, es decir, si tiene almacenado algún valor. Regresa como valores TRUE si lo encuentra y si no lo encuentra regresa FALSE.
unset	Destruye una variable y libera la memoria asignada a esta. Regresa TRUE si es un éxito si no regresa un FALSE.

Tabla 31. Funciones básicas

Las constantes son iguales a las variables con la diferencia que éstas no pueden ser redefinidas con otro valor. Para definir las se utiliza define("constante", valor de la constante), para poder usarlas sólo se escribe el nombre sin anteponer el símbolo de \$.

Una vez visto esto empezamos a hacer arreglos, manejo de cadenas e inclusión de archivos. Al trabajar con arreglos se pudieron ver las ventajas no sólo en el área matemática, sino como apoyo en estructuras más avanzadas, con el se pueden hacer rotaciones de puntos y crear la modificación de archivos de imagen. Con manejo de cadenas se puede hacer una valoración de los datos escritos, digamos por ejemplo un formulario. Hay dos tipos de arreglos, los escalares y los asociativos. Los escalares es aquel que maneja dentro de sí los datos en forma numérica, y los asociativos no los maneja en forma numérica. La tabla 32 define la sintaxis de un array:

Arreglo lineal	Arreglo lineal asociativo y escalar	Arreglo tridimensional
<code>\$arreglo = array("uno", "dos", "tres");</code>	<code>\$arreglo[0] = "Primero";</code> <code>\$arreglo[1] = "Segundo";</code> <code>\$arreglo[2] = "Tercero";</code>  ó  <code>\$arreglo[] = "Primero";</code> <code>\$arreglo[] = "Segundo";</code> <code>\$arreglo[] = "Tercero";</code>	<code>\$arreglo[0] = (1,2,3);</code> <code>\$arreglo[1] = ("perro", "gato", "raton");</code> <code>\$arreglo[2] = ("uno", "dos", "tres");</code>  ó  <code>\$arreglo=array(1,2,3),array(4,5,6),array(7,8,9);</code>

Tabla 32. Sintaxis de array's

La tabla 33 enlista las funciones más utilizadas para el manejo de arreglos:

Función	Descripción
<code>int sizeof (array var)</code>	Regresa el número de elementos del arreglo.
<code>mixed array_pop (array array)</code>	Saca y regresa el último elemento del arreglo, reduciendo el tamaño del arreglo en un elemento.
<code>int array_push (array array, mixed var [, mixed ...])</code>	Trata el arreglo como una pila y pone las variables pasadas como argumento al final del arreglo. Regresa el nuevo número de elementos en el arreglo.
<code>bool in_array (mixed needle, array haystack)</code>	Busca un elemento en un arreglo y regresa TRUE si lo encuentra o FALSE en caso contrario.

Tabla 33. Funciones para aplicaciones en array's

El manejo de cadenas es muy conveniente cuando se desea evaluar de forma especial la entrada de datos en un formulario. Para un control sobre las cadenas tenemos funciones que son explicadas en la tabla 34:

<b>Función</b>	<b>Descripción</b>	<b>Sintaxis</b>
strlen	Regresa la longitud de la cadena	int <b>strlen</b> (string str)
echo	Manda a la salida todos los argumentos ya sea en cadenas individuales o concatenación de estas.	<b>echo</b> (string arg1, string [argn]... )
print	Manda a la salida sus argumentos.	<b>print</b> (string arg)
printf	Manda a la salida los argumento con un cierto formato. El formato es el que se describe con sprintf().	int <b>printf</b> (string format [, mixed args...])
sprintf	Devuelve una cadena con formato.	string <b>sprintf</b> (string formato [, mixed args...])
strstr	Regresa una cadena formada desde la primera ocurrencia de cierta subcadena hasta el final de dicha cadena.	string <b>strstr</b> (string str, string substring)
implode	Regresa una cadena que es la representación de los elementos del arreglo unidos por un cierto delimitador.	string <b>implode</b> (string delim, array piezas)
explode	Regresa un arreglo de cadenas, cada una de las cuales es una subcadena formada al romper una cadena original de acuerdo a cierto delimitador. Se puede especificar el máximo número de elementos en el arreglo utilizando el argumento limit.	array <b>explode</b> (string delim, string source [, int limit])

*Tabla 34. funciones en el manejo de cadenas*

Y la inclusión de archivos, es una mirada rápida a una base de datos pequeña, si los datos no son suficientemente grandes para una base de datos, pueden ser almacenados en un documento de texto.

Para poder abrir cualquier documento de texto se usa la siguiente expresión:

int **fopen** (string filename, string mode)

Como si fuera una asignación de privilegios, se le puede tratar a estos documentos como datos limitados, asignándoles parámetros para su uso, estos son los parámetros de lectura-escritura de un documento (tabla 35):

<b>Modos</b>	<b>Descripción</b>
r	Abre para lectura únicamente.

r+	Abre para lectura y escritura.
w	Abre para escritura únicamente. Si el archivo no existe intenta crearlo.
w+	Abre para lectura y escritura. Si el archivo no existe intenta crearlo.
a	Abre para escritura únicamente. Si el archivo no existe intenta crearlo.
a+	Abre para lectura y escritura. Si el archivo no existe intenta crearlo.

*Tabla 35. Parámetros en la lectura-escritura de documentos de texto*

La diferencia entre los modos w, w+ y a, a+ es que estos últimos colocan el puntero al final del archivo, todos los demás lo colocan al principio de archivo.

Además de abrir los archivos también podemos cerrarlos, o pedir información de ellos, estas son algunas de esas funciones (tabla 36):

<b>Función</b>	<b>Descripción</b>	<b>Sintaxis</b>
filesize	Entrega el tamaño en bytes del documento, o FALSE si no lo encuentra.	int <b>filesize</b> (string filename)
feof	Regresa TRUE si el puntero de archivo se encuentra al final, en caso de no entrega FALSE.	int <b>feof</b> (int fp)
fread	Lee una cantidad de bytes de un archivo. La lectura termina cuando se ha leído la cantidad especificada o cuando se ha alcanzado el final de archivo (EOF), lo que suceda primero.	string <b>fread</b> (int fp, int length)
fwrite	Escribe una cadena a un archivo, puede especificarse opcionalmente cual es la longitud de la cadena que se va a escribir.	int <b>fwrite</b> (int fp, string string, int [length])
fclose	Cierra el archivo, en caso de que no pueda entrega FALSE.	int <b>fclose</b> (int fp)

*Tabla 36. Funciones en la lectura-escritura de documentos*

En lo que respecta a programación con CGI's se vio la posible relación que existirá entre un servidor web y un programa, de tal que modo que éste pueda interactuar con la Internet. CGI se definió como una interfaz que permite a los servidores web comunicarse con aplicaciones externas o acceder a otras fuentes de información. En las aplicaciones CGI, se utilizan las variables de ambiente las cuales ayudan a transmitir datos acerca de una petición de http de un servidor a una aplicación CGI. Algunas de estas variables son:

REQUEST\_METHOD: Indica cual es el método que se utiliza al ejecutar un programa CGI, este puede ser GET o POST.

SCRIPT\_NAME: Nombre por el que fue llamado el programa CGI.

QUERY\_STRING: Variable que contiene la información pasada a un programa CGI cuando se utiliza el método GET.

CONTENT\_LENGTH: Indica la longitud de la cadena que será leída de la entrada estándar cuando se utiliza el método POST.

REMOTE\_HOST: Contiene el nombre de la máquina remota que ejecutó el programa CGI.

REMOTE\_ADDR: Indica la dirección IP de la máquina remota que ejecutó el programa CGI.

SERVER\_NAME: Indica el nombre del servidor en el cual se ejecuta el programa CGI.

HTTP\_USER\_AGENT: Indica que tipo de cliente ejecutó el programa CGI.

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.

Algunas funciones útiles son: 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.

## Módulo VI: Interacción de WWW con Bases de Datos

<b>Temario</b>	
1. Introducción	
2. Manejo de formularios como Front-End	
3. Instalación y configuración de la base de datos en Linux	
	4. Introducción al desarrollo de CGI's
	5. Desarrollo del Back-End con MySQL
	6. Desarrollo de una aplicación personalizada

En este módulo se describe el desarrollo de una aplicación de bases de datos que funcionará a través del servidor en la WWW.

Aquí se establecen los conceptos de la base de datos (datos, información y esquemas), siendo los datos, caracteres que por si solos no tienen valor. La información se originará a partir de

la suma de todos esos caracteres, y toman un valor importante. Los esquemas conforman la estructura de toda la base de datos.

Un DBMS o manejador de bases de datos por sus siglas en inglés es aquel programa que junto con una colección de datos, se accede a los datos. Su objetivo primordial es la de brindar la información pronta y precisa, además de asegurar su completa confidencialidad. En este caso se usara el DBMS MySQL.

Los modelos son herramientas que van 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.

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

Esta técnica de normalización consiste en transformar una tabla en varias fases. Su objetivo es facilidad en actualización de los mismos. El ciclo de vida de un sistema consta de seis fases: análisis, diseño, implementación, pruebas, puesta en marcha y mantenimiento.

En el análisis se entrevista a los clientes 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 crea el sistema que administrará la base de datos, se hace la base de datos y se cargan o importan los datos.

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

Utilizar como manejador de BD MySQL se debe a que es el DBMS libre más usado para sitios web, además de tener la practicidad de ser usado en otras funciones necesarias para entornos, conservando su velocidad. Algunas características de este manejador son las siguientes:

**Costo:** Es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.

**Asistencia:** Ofrece contratos de asistencia a precios razonables y existe una activa comunidad Misal.

**Velocidad:** es mucho más rápido que la mayor parte de sus rivales.

**Funcionalidad:** dispone de varias funciones que exigen los desarrolladores, como la compatibilidad para duplicación, funciones SSL e integración en la mayor parte de los lenguajes de programación.

**Portabilidad:** se ejecuta en la mayoría de los SO y los datos se pueden transferir de un sistema a otro sin presentar dificultad.

**Facilidad de uso:** Resulta fácil de utilizar y administrar. Las herramientas de Misal son potentes y flexibles, sin sacrificar su capacidad de uso.

Sólo basta con descargar una versión actual de Misal y descomprimirla en cualquier parte del equipo (se debe verificar que versión tiene ya instalada si es que existe, y desinstalarla). Una vez descomprimido, introducirse en su carpeta y leer el instructivo de instalación. Existirán varias formas de hacerlo pero se debe escoger la más adecuada a las necesidades. Una vez hecho, se levanta el servicio no como usuario root sino como otro usuario:

```
mysqld_safe -user=mysql &
```

con esta línea se levanta el servicio como usuario root. Para establecer contacto con la BD ejecutamos lo siguiente:

```
mysql -u root -p
```

Para finalizar el servicio bastara con ejecutar el siguiente comando:

```
mysqladmin shutdown -u root -p
```

Si la BD no tiene password se debe establecer uno como usuario root, para que sólo el administrador tenga acceso a ella. Para hacerlo se escribe el siguiente comando:

```
mysqladmin -u root password 'mi_password'
```

Como administradores, siempre se debe contar con un respaldo de nuestra BD. Para esto existen tres maneras de hacerlo, dependiendo de que se quiere respaldar:

Respaldo de una base de datos

```
mysqldump base_datos -u root -p > /respaldos/base_resp.sql
```

Respaldo de una tabla de una base de datos

```
mysqldump base_datos tabla -u root -p > /respaldos/tabla.sql
```

Restauración de la base de datos

```
mysql basedatos -u root -p > /respaldos/base.sql
```

Con este último comando se restaura la base de datos original (sí sufrió algún percance la base de datos), con la base respaldada.

Algunos comandos básicos para proporcionar información acerca de la base de datos, son ocupados por los administradores para verificar que la base junto con sus tablas no hayan sido alteradas, son los siguientes (tabla 37):

Comando	Significado
USE base_datos	Decidir que base de datos se va a ocupar.
SELECT database();	Muestra la base con la que se va a trabajar.
SELECT user();	Muestra con que usuario se va a trabajar.
SELECT VERSION();	Muestra la versión de MySQL que se está usando.
SHOW TABLES;	Muestra un listado de las tablas que están contenidas en la base de datos.
SHOW TABLES STATUS;	Muestra además del listado los estados de las tablas.
SHOW COLUMNS FROM NOMBRE_TABLA; DESCRIBE NOMBRE_TABLA;	Muestra las columnas de una tabla en particular.

*Tabla 37. Comandos básicos*

La tabla 38 de comandos son de gran utilidad para la creación, y administración de la base de datos:

Comando	Significado
CREATE TABLE nombre_tabla ( nombre_campo1 tipo_dato, nombre_campo2 tipo_dato, CONSTRAINT pk_key PRIMARY KEY(nombre_campo1) );	Creación de una tabla
CREATE TABLE uno ( campo1 tipo_dato, campo2 tipo_dato, CONSTRAINT pk_key PRIMARY KEY(campo1) ); CREATE TABLE dos ( n_campo1 tipo_dato, n_campo2 tipo_dato, n_campo3 tipo_dato, CONSTRAINT pk_key PRIMARY KEY(n_campo1), CONSTRAINT fk_key FOREIGN KEY(n_campo1) REFERENCES uno(campo1) );	Añadir llaves foráneas
ALTER TABLE nombre_tabla ADD nuevo_campo tipo_dato;	Agregar columnas.
ALTER TABLE nombre_tabla CHANGE campo nuevo_campo VARCHAR(30); Ó ALTER TABLE nombre_tabla MODIFY nuevo_campo integer;	Modifica una definición de

	columna, ya sea el nombre o tipo de variable.
ALTER TABLE nombre_tabla RENAME TO nuevo_nombre_tabla; Ó ALTER TABLE nombre_tabla RENAME nuevo_nombre_tabla	Cambiar el nombre de una tabla.
ALTER TABLE nombre_tabla ADD CONSTRAINT pk_tabla PRIMARY KEY (campo); Ó ALTER TABLE nombre_tabla ADD PRIMARY KEY (campo);	Añadir llaves primarias.
ALTER TABLE nombre_tabla ADD FOREIGN KEY (campo) REFERENCES tabla_externa (campo_externo);	Añadir llaves foráneas.
ALTER TABLE nombre_tabla DROP PRIMARY KEY;	Borrar llave primaria.
ALTER TABLE nombre_tabla DROP campo_borrar;	Borrar un campo de una tabla.
INSERT INTO nombre_tabla (campo1, campo2) VALUES (valor_campo1, valor_campo2); Ó INSERT INTO nombre_tabla <u>VALUES</u> (valor_campo1, valor_campo2);	Insertar datos dentro de una tabla.
UPDATE nombre_tabla SET nombre_campo=valor_campo WHERE comparación_expresión;	Actualizar datos.
DELETE FROM nombre_tabla WHERE comparación_expresión;	Borrar datos
SELECT campo1, campo2 FROM nombre_tabla WHERE condicion_expresión;	Seleccionar datos
DROP TABLE nombre_tabla;	Borrar tabla.
DROP DATABASE nombre_base_datos;	Borrar base de datos.

*Tabla 38. Comandos administradores de bases de datos*

Para las expresiones de comparación o condicionamiento se utilizan símbolos o constantes para poder ejecutarlas (tabla 39):

Expresión	Ejemplo	Significado
=		Igual que
!, NOT		Invierte el valor de verdad de la expresión.
!=, <>		Diferente que
>		Mayor que
<		Menor que
>=		Mayor igual que
<=		Menor igual que
IS NULL	a IS NULL	Verdad si a contiene un valor NULL.
IS NOT NULL	a IS NOT NULL	Verdad si a no contiene ningún valor NULL.
BETWEEN	a BETWEEN b AND c	Verdad si a esta entre los valores de b y c.
NOT BETWEEN	a NOT BETWEEN b AND c	Verdad si a no esta entre los valores de b y c.
LIKE	a LIKE b	Verdad si a equivale con b en una

		correspondencia de patrón SQL.
NOT LIKE	a NOT LIKE b	Verdad si a no equivale con b en una correspondencia de patrón SQL.

*Tabla 39. Funciones de condicionamiento*

Algunas funciones ya vienen incorporadas a MySQL, las cuales no pueden proporcionar información requerida sin tener que programar el evento, algunas de estas son (tabla 40):

Comando	Descripción
CURRENT_DATE();	Devuelve la hora actual con este formato AAAA-MM-DD hh:mm:ss.
DAYOFWEEK(AAAA-MM-DD);	Devuelve el día de la semana de la fecha proporcionada (1-domingo, 7-sábado).
DAYOFYEAR(AAAA-MM-DD);	Devuelve el día del año de la fecha proporcionada (1-366).
ASCII(carácter)	Devuelve el valor ASCII del primer carácter.
BIN(número)	Devuelve el valor binario del número especificado.
BIT_LENGTH(cadena)	Devuelve la longitud de la cadena especificada.
CHAR(número1, número2,...)	Devuelve los caracteres que se obtendrían si cada número fuera un entero convertido desde código ASCII, ignorando valores nulos.
CONCAT(cadena1, cadena2,...)	Concatena las cadenas y devuelve la cadena resultante.
REVERSE(cadena)	Invierte el orden de los caracteres y devuelve el resultado.
LOWER(cadena)	Devuelve una cadena con todos los caracteres convertidos a minúsculas.
UPPER(cadena)	Devuelve una cadena con todos los caracteres convertidos en mayúsculas.
ABS(número)	Devuelve el valor absoluto de un número.
MOD(número 1, número 2)	Devuelve el módulo de número 1 y 2.
PI()	Devuelve una representación próxima al valor $\pi$ .
POW(número 1, número 2)	Esta función eleva número 1 a la potencia de número 2 y devuelve resultado.
RAND([número])	Devuelve un número aleatorio (como flotante) comprendido entre 0 y 1.
BENCHMARK(número _ repeticiones, expresión)	Esta función brinda una idea general sobre la diferencia de los equipos aunque su principal objetivo es ayudar a optimizar las funciones

*Tabla 40. Funciones incorporadas a MySQL*

Al instalar MySQL, de manera automática se instala la base de datos mysql. Esta base de datos dará la facilidad de poder administrar de una manera más eficiente a los usuarios, que tendrán acceso a la base de datos correspondiente (tabla 41). Existen cinco tablas principales en esta base que afectan el acceso al sistema:

Tabla	Descripción
User	Enumera la usuarios, los equipos y contraseñas asociadas que pueden acceder al servidor así con los permisos globales que tienen.
db	Enumera las bases de datos a las que pueden acceder los usuarios. Los permisos otorgados se aplican a todas las tablas de la case de datos.
host	Junto con la tabla de db, permite un acceso más controlado en función de un determinado equipo.
tables_priv	Enumera el acceso a tablas concretas. Los permisos otorgados se aplican a todas las columnas de la tabla.
columns_priv	Enumera el acceso a columnas específicas

*Tabla 41. Tablas de control en MySQL*

Para poder agregar usuarios, revocar permisos, cambiar passwords, se utilizan los siguientes comandos (los usuarios pueden ser locales o remotos; tabla 42):

Comando	Significado
GRANT tipo_privilegio1,tipo_privilegio2 ON base_datos.tabla usuario@direccion IDENTIFIED BY 'contraseña';	Otorga determinados permisos sobre la base de datos.
REVOKE tipo_privilegio1 ON base_datos FROM usuario@direccion;	Revoca determinado privilegio o permiso sobre la base de datos.
SET PASSWORD=PASSWORD ('new_password');	Cambia el password del usuario en sesión.

*Tabla 42. Funciones de control de permisos*

La tabla 43 denota los principales privilegios con los que un usuario puede contar en su poder:

Privilegio	Descripción
ALL	Concede todos los permisos básicos(excepto GRANT).
ALL PRIVILEGES	Igual que ALL.
ALTER	Permiso para cambiar la estructura de una tabla a excepción de los índices.
CREATE	Permiso para crear bases de datos y tablas, a excepción de índices.
DELETE	Permiso para eliminar registros de una tabla
DROP	Permiso para eliminar bases de datos o tablas, a excepción de índices.
INDEX	Permiso para crear, modificar, borrar índices.

INSERT	Permiso para añadir nuevos registros a una tabla.
SHOW DATABASES	Permiso para ver todas las bases de datos.
SELECT	Permiso para devolver datos de una tabla.
SHUTDOWN	Permiso para cerrar el servidor.
UPDATE	Permiso para modificar datos de una tabla.

*Tabla 43. Funciones de Administración de Permisos*

Como se observa en esta tabla, lo que se hace es otorgar permisos o revocarlos sobre los comandos que se utilizan en MySQL. Dependerá de las necesidades de cada usuario estas limitantes. Para poder 'activar' estos cambios se pueden ejecutar los siguientes comandos: FLUSH PRIVILEGES, mysqladmin flush-privileges o mysqladmin reload.

A continuación se muestran las expresiones utilizadas para otorgar y revocar permisos:

Tabla	Descripción
*.*	Todas las tablas de todas las bases de datos.
*	Todas las tablas de la base de datos actual.
NombreBD.*	Todas las tablas de la base de datos "NombreBD".
NombreBD.NombreTabla	La tabla "NombreTabla" de la base de datos "NombreBD".

*Tabla 44. Expresiones para las funciones de administración de usuarios*

Los registros que están almacenados en una tabla deben estar de manera "accesible", esto quiere decir ordenado. Para ello se ocupan índices; variando del tipo que se necesite, existen en MySQL cuatro tipos de índices (45):

Tipo	Descripción
Clave primaria	Es un índice establecido sobre un campo en el que cada valor es exclusivo y ninguno de los valores pueden ser nulos.
Índice exclusivo	Es lo mismo que un índice ordinario con la diferencia de que no se permiten duplicaciones. Estos índices permiten valores nulos.
Índice de texto completo	Igual que el índice ordinario pero que ordena los valores comparando toda la cadena dentro del campo.
Índice ordinario	Estos índices permiten el uso de valores duplicados.

*Tabla 45. Tipos de índices*

En la siguiente tabla (46) se muestra la manera de crear cada uno de estos índices:

Tipo	Comando
Clave primaria	CREATE TABLE nombre_tabla(

	Campo1 tipo_dato, Campo2 tipo_dato, CONSTRAINT pk_nombre_tabla PRIMARY KEY (campo1));
Índice exclusivo	CREATE TABLE nombre_tabla( campo1 tipo_columna, campo2 tipo_columna, UNIQUE (nombre_campo));
Índice ordinario	CREATE TABLE nombre_tabla( Campo1 tipo_dato, Campo2 tipo_dato, INDEX nombre_indice (Campo1), INDEX nombre_indice (Campo2));

*Tabla 46. Creación de índices*

En la creación de tablas es bueno ocupar índice, esto acelera la búsqueda de información, pero también es debido solo usar los necesarios ya que el abuso de estos puede provocar que se alente el envío de información.

Los campos de autoincremento son muy útiles, cuando se requiere incrementar un campo definido cada vez que se inserte un registro. Este campo debe ser de tipo numérico para evitar conflictos. Con la siguiente sintaxis se crea un campo de autoincremento:

```
CREATE TABLE nombre_tabla(
    Nombre_campo1 INT AUTO_INCREMENT PRIMARY KEY,
    Nombre_campo2 tipo_dato);
```

Para crear un campo de incremento automático en una tabla ya existente se utiliza el siguiente comando:

```
ALTER TABLE nombre_tabla MODIFY nombre_campo tipo_columna AUTO_INCREMENT;
```

Saber hacer transacciones entre registros, es muy útil cuando se necesita intercambiar información y actualizarla. Las funciones BEGIN, COMMIT, ROLLBACK, UPDATE, son las funciones que se deben utilizar para hacer las transacciones.

La función BEGIN indica el inicio de la transacción, COMMIT que la transacción a terminado y ROLLBACK que en caso de error deshaga la operación de transacción. UPDATE indicará que datos deben realizar la transacción y en que tabla.

```
BEGIN;
UPDATE nombre_tabla SET campo1=valor_campo1;
UPDATE nombre_tabla SET campo2=valor_campo2;
COMMIT;
```

o

```

BEGIN;
UPDATE nombre_tabla SET campo1=valor_campo1;
UPDATE nombre_tabla SET campo2=valor_campo2;
ROOLBACK;

```

El administrar una base de datos sin una interfaz gráfica requiere de mucha paciencia, mas aun el estructurarla desde su inicio, pero con lo observado durante el módulo se puede tener una idea de investigación y así poder conocer mas a fondo sobre las bases de datos y sus manejadores.

## Módulo VII: Introducción a la seguridad en cómputo.

<p><b>Temario</b></p> <p>1. Introducción</p> <p>2. Control de acceso</p> <p>3. Sistema de archivos</p>	<p>4. Administración básica de seguridad</p> <p>5. Seguridad en red</p> <p>6. Políticas de seguridad</p>
--	--

En este último módulo del diplomado, se llego a un acuerdo entre alumnos y el maestro, para modificar el temario propuesto para mejorar su comprensión y conocer en mejor manera la manera de ataques en lugar de sólo defenderse de ellos.

El primer punto observado fueron las definiciones de algunas características que pueden estar contenidas en un sistema en su programación. Algunas de ellos son:

Tabla	Definición
Seguridad	Es la característica de un sistema que garantice que opere como se espera que lo haga, que sea ajeno a todo riesgo y amenaza, que no posee vulnerabilidades, que es confiable y que funciona sin fallo. Se debe orientar a ofrecer tranquilidad.
Amenaza	Circunstancia que puede causar daño a un sistema. Frecuentemente aprovecha vulnerabilidades.
Vulnerabilidad	Ausencia de una contramedida o debilidad de la misma, haciendo propenso al sistema a ser afectado por un agente perturbador. La debilidad surge en el diseño, la implementación o procedimientos para operar y administrar un sistema. En seguridad informática se define como hoyo en el sistema.
Peligro	Es la probabilidad en que se presente una amenaza.
Exposición	Es el grado en que esta el nivel de afectación en ser atacados.
Riesgo	Tiene una formula muy fácil de comprender:

Riesgo = Peligro * Exposición * Vulnerabilidad Es la probabilidad de que una vulnerabilidad sea explotada, de acuerdo a su grado de exposición y peligro que involucra.
--

*Tabla 47. Definiciones*

Tomando las definiciones anteriores se puede definir un Sistema Seguro: es un sistema en el que se puede confiar porque se comportara como debería hacerlo, la información se mantendrá inalterada y accesible durante el tiempo que esta sea requerida por el dueño.

Para lograr este objetivo se debe contar con políticas de seguridad y servicios. Un servicio de seguridad y una política están vinculados, porque el servicio de seguridad debe satisfacer una política de seguridad. Y 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, autenticación(algo que se sabe, algo que se tiene, algo que se es).

Algo que se sabe, es lo que el usuario guarda de manera confidencial en su cerebro, lo que podría ser un nip, password, claves de acceso, etc. Algo que se tiene, es lo físico que brindará el acceso, ya sean tarjetas de identificación, memorias USB, llaves Bluetooth, etc. Y algo que se es, son las maneras en que se puede identificar con el propio cuerpo, ya sea huellas digitales, patrones físicos, lectura de iris, etc.

Dentro de los servicios de seguridad se puede encontrar la integridad(que la información se mantenga inalterada desde su creación o modificación), autorización(se asegura que la información este accesible sólo a usuarios determinados con la posibilidad de establecer niveles de seguridad a cada uno), no repudio(se asegura que el emisor no puede negar que la información no fue enviada), disponibilidad(que el sistema este disponible en cualquier momento que se requiera), y auditoria(llevar a un registro de los accesos a esa información marcando eventos exitosos y fallidos y entregando una bitácora).

Existen diferentes niveles de confianza, entre los más conocidos están los niveles establecidos por la National Computer Security Center en su libro U.S. Department of Defense Trusted Computer System Evaluation Criteria mejor conocido como el Libro Naranja, este divide sus niveles en 4 grupos y siete niveles de confianza para ambientes de computo:

D	Mínima
C1	
C2	
B1	
B2	
B3	
A	Se realizan pruebas formales de seguridad en el sistema

*Tabla 48. Niveles de seguridad del Libro Naranja*

Y la Common Criteria Organization proporcionando siete niveles desconocidos como Evaluation Assurance Leves (EALs):

EAL1	Funcionalidad probada
EAL2	
EAL3	
EAL4	
EAL5	
EAL6	
EAL7	Formalmente verificado, diseñado y probado

*Tabla 49. Niveles de seguridad de EAL's*

Se define la criptografía como la forma de cifrar datos en forma oculta, 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.

La criptografía simétrica requiere que el emisor y el receptor, compartan una llave secreta la cual es utilizada para cifrar el mensaje y descifrarla cuando llegue a su destino. Como se menciona arriba existe el inconveniente de extravío o robo de la llave, para ello se recomienda un canal seguro de intercambio por ejemplo el DES, 3DES AES, blowfish o IDEA. En cambio la criptografía asimétrica resuelve ese problema, al utilizar dos llaves. Tanto el emisor como el receptor tiene dos llaves, una pública y otra privada. La privada es guardada en secreto y la pública se comparte con el otro individuo, para que sea capaz de cifrar la información con ella.

Actualmente existen dos tipos de algoritmos de cifrado, el Rivest Shamir Adleman (RSA), y Diffie-Hellman (D-H). Este último utiliza el llamado exponencial de claves en su algoritmo y el RSA debe su solidez a factorizar números primos grandes.

Para poder marcar (autenticarlo) el paquete o mensaje, existe el Message Digest que es el resultado de un algoritmo unidireccional que permite generar una huella digital, que es cifrar el paquete con su propia llave. Esto puede garantizar la integridad de un mensaje. En el módulo se cifra 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.

Cuando se cifra con la llave privada todo aquel que tenga la llave pública puede conocer el contenido del mensaje. Con esto se obtiene la firma digital, esto genera la autenticación ya que sólo el propietario pueden cifrar con su propia llave.

Durante el curso se generó 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 pública. Siendo este certificado avalado por un tercero. Siendo HTTPS un nuevo formato de comunicación que protegerá y dará seguridad al cifrar toda información corriendo por Internet.

Las siguientes son algunas de las recomendaciones que se sugieren para una mejor seguridad:

- Mantener en producción una versión actualizada.
- Evitar el uso de permisos no necesarios en el directorio de apache.
- Restringir el uso de SSI y de CGIs.
- Revisar con frecuencia las bitácoras del web server.
- Seleccionar passwords fuertes para los accesos restringidos.
- Implementar el soporte para SSL(mod\_ssl).
- Agregar módulos orientados a incrementar el nivel de seguridad(mod\_access, mod\_auth, mod\_security).

Se pueden administrar las directivas dirigidas a la seguridad, algunas directivas básicas son: Order, Allow, Deny. Estas ya fueron explicadas anteriormente, por eso sólo se mencionaran. Con estas directivas se pueden definir los clientes que tendrán o no acceso al servidor.

Se puede hacer de dos maneras la configuración, como Allow-Deny o Deny-Allow. La primera dice que "Todo lo que no esta explícitamente permitido esta prohibido" y la segunda dice que "Todo lo que no esta explícitamente prohibido esta permitido".

Hay que considerar que los medios de transmisión no son seguros por lo que hay que utilizar canales más seguros como: SSH, SFTP, IMAPs, POPs, etc. HTTPS es la implementación de http con SSL para cifrar su transmisión. SSL proporciona seguridad criptográfica, esto es, que todo el contenido que sea transmitido por ese canal será cifrado; interoperabilidad y extensibilidad.

Otro punto 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.



Figura 11. Tipos de ataques

Los pasivos se caracterizan por ser menos dañinos, esto es que pueden sólo 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 cuatro tipos de sniffers, los activos, los pasivos, los spoofing y lo que envenenan la tabla de arp. Los pasivos sólo capturan paquetes, y los activos intentan apoderarse de las sesiones. Estos sniffers tienen la propiedad de ser casi invisibles, como no generan bitácoras (son pasivos), utilizan un mínimo de memoria y cpu, son difíciles de detectar.

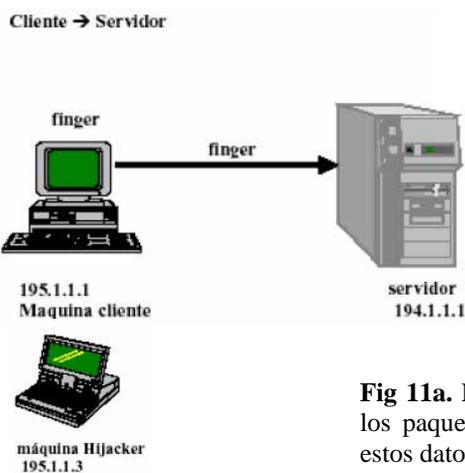
Los sniffers más utilizados son Dsniff, Sniffit, Snifsol, Ethereal, Tcpdump, snort, aunque éste último no es un sniffer en concreto sino que tiene la facilidad de usarse como uno.

Entre los ataques activos se encuentran los ya conocidos:

- Virus
- Caballo de troya
- Worms
- Fuerza bruta
- Sniffers
- Spoofing
- Spam
- Graffiti
- Negación de servicio, etc.
- Spoofing

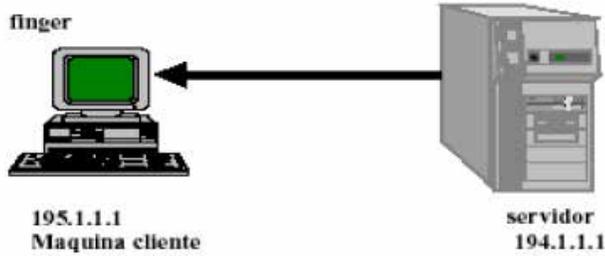
No sólo se puede atacar de esta manera, existe lo que es la ingeniería social, la cual consiste en obtener información importante a través del conocimiento de los Whois. Esto también con lleva a la falsificación y a la usurpación. La utilización de debilidades en el servidor (Backdoors, stack overflow, 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 se sabe por donde atacar. Esto se hizo 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 o 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 haciéndose pasar por el usuario verdadero (fig 11<sup>a</sup>, 11b, 11c, 11d).



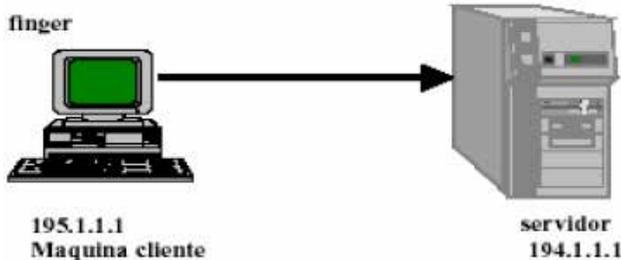
**Fig 11a.** Los datos de este campo son usados para definir la secuencia de los paquetes enviados. Esta en hexadecimal y el server lo que hará con estos datos es ponerlos en el campo ACK el paquete que envíe al cliente.

Servidor → Cliente



**Fig 11b.** El servidor traduce el contenido del paquete y lo ejecuta, escribiendo la “i” en el shell y envía otro paquete al cliente, que en realidad es un “echo” del recibido por el cliente para que el cliente se cerciore de que el paquete a llegado y ha sido ejecutado.

Cliente → Servidor



**Fig 11c.** El cliente enviara un paquete ACK, como respuesta al paquete que acaba de recibir del servidor. Este paquete no contendrá DATA, sólo sirve para confirmar que ha llegado el paquete del servidor.



**Fig 11d.** Ahora es cuando se debe enviar nuevos datos al servidor haciéndose pasar por el cliente. Para lograrlo se debe calcular la secuencia del paquete spoofed que se enviaran de los números SEQ y ACK, basándose en el primer paquete que se ha interceptado (¿sniffado?:-?). se debe enviar datos al servidor para que no se ejecute el comando que el cliente estaba introduciendo(que por ahora sólo había enviado “i” del finger , con lo cual será más fácil), para que esto ocurra se envían retornos de carro.

Un ataque más para obtener información y para dejarlo fuera de servicio, es el Denial of Service (fig 15). Este consiste en infectar a máquinas inocentes y hacer que estas infecten al servidor mandándole ataques DoS (Ping de la muerte, Inundación Sync, spoofing, smurt, Fraggle, entre otros).

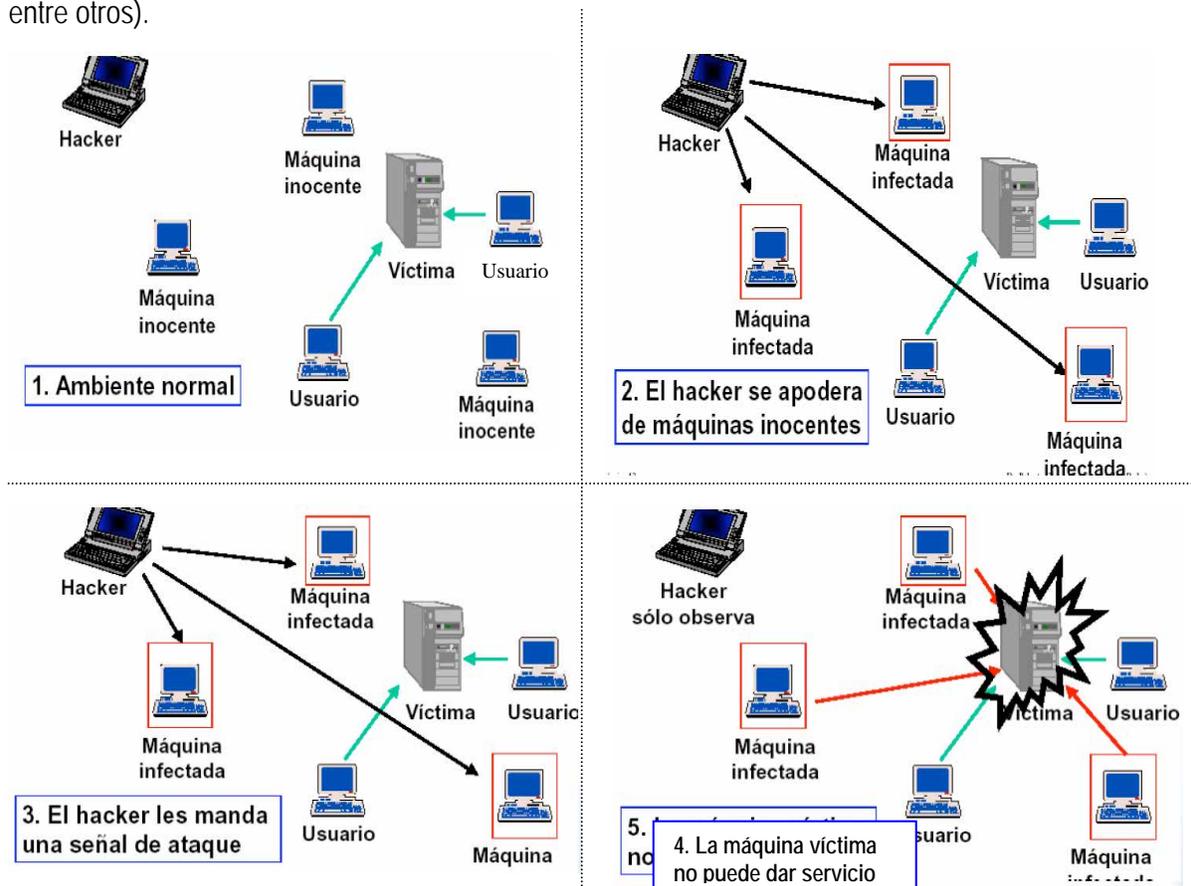


Figura 12. Ataque Denial Of Service

Un ataque Sync flood, consiste en lanzar paquetes SYNC al equipo a atacar, cuando los recibe la víctima lanza SYNC-ACK en espera de ACKs, pero al recibir puros SYNC satura la capacidad del equipo de recibir nuevas conexiones, dejándolo fuera de servicio.

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 número de sync permitidos y derriba el servidor.

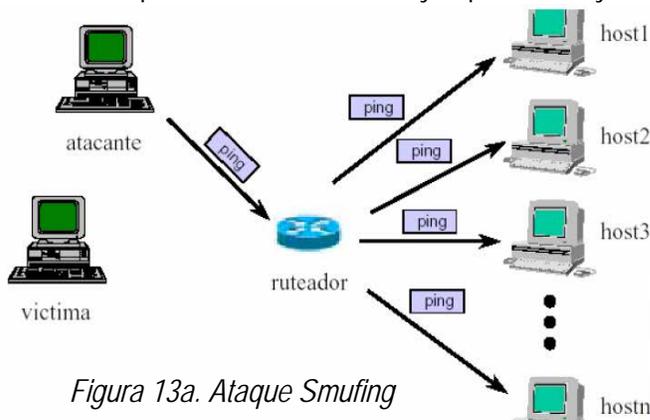


Figura 13a. Ataque Smurfing

Un smufing (fig 13a y 13b) es un ataque de los más temidos. Requiere de tres "personajes" para que funcione: una víctima, 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 tráfico que la victima no soportara.

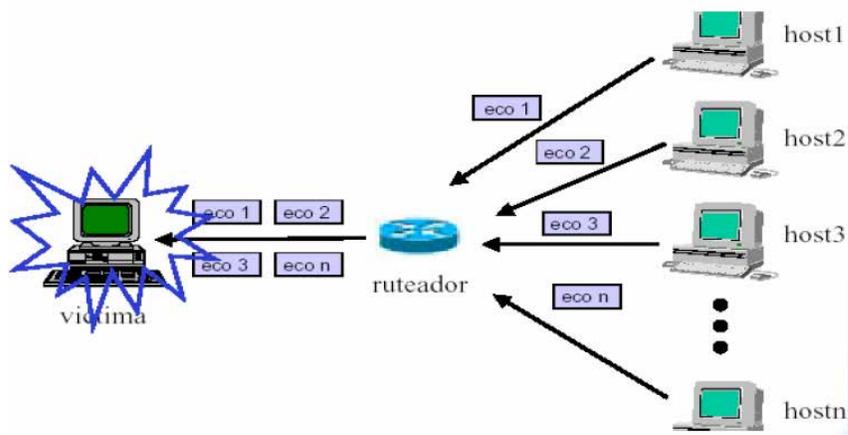


Figura 13b. Ataque Smuffig

Un spoofing es simplemente la creación de paquetes de comunicación TCP/IP usando la dirección IP de alguien más. La ingeniería social requiere de usar trucos psicológicos para hacerse pasar por un usuario autorizado. Consiguiendo información importante puede navegar de equipo a equipo con esa información.

Al hablar de virus se debe adentrar mucho a este campo, porque existe una gran variedad de variantes, estas son: troyanos, gusanos, bomba lógica, spyware, adware, etc. Los más dañinos de todos estos son los gusanos, puesto que existe una gran variedad de ellos. Estos programas se reproducen o hacen copias de sí mismos a través de la red, provocando enormes sobrecargas de procesamiento en las máquinas que se instalan y reduciendo la disponibilidad de los sistemas. Los virus en general se ejecutan solos, viviendo dentro de otro programa, va escalando memoria. Si el ataque es por medio de una vulnerabilidad en el sistema se le llaman exploits, pero si el ataque es enviando un cierto comando al equipo y que sea ejecutado causando un daño se llaman bombas lógicas.

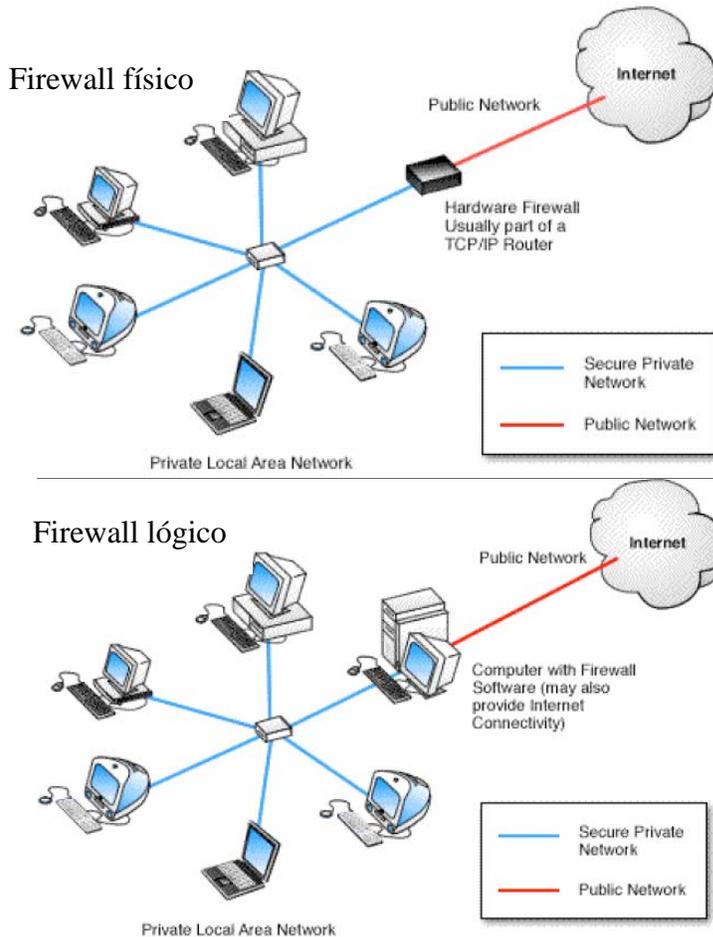
Existen tres tipos de buscadores de vulnerabilidades (tabla 50), estos ocupan esa información ya sea para uso personal o para alertar a los demás:

Protagonista	Descripción
White Hat	Busca vulnerabilidades en el sistema y crea el PoC (parche), y avisa de esto a los fabricantes.
Gray Hat	Encuentra nuevas vulnerabilidades, desarrolla el PoC, pero no avisa a los fabricantes sólo pública su el PoC en la red.
Black Hat	Busca nuevas vulnerabilidades y la utiliza para beneficio personal.

Tabla 50. Tipos de hackers

Existe un grupo llamado Script Kiddies, que no son más que personas que saben descargar y ejecutar programas, son grafiteros por excelencia. Las consecuencias de sus actos son mínimos y pueden ser reparados.

Para poder defendernos existen los Firewalls, de estos existen dos tipos los físicos y los lógicos (fig 14). Los físicos son usualmente ruteadores, o switch de capa 3. Un lógico son programas que diferencia entre paquetes nocivos y paquetes permitidos.



Entre los firewalls físicos 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.

Los iptables son tablas de Ips que controlaran el tráfico de información en la red dependiendo de la administración que tenga. Con ciertas instrucciones se puede bloquear el acceso de cierta información a diversos usuarios, filtrando así la red de paquetes

dañosos.

Figura 14. Tipos de Firewalls

Otra forma de protección son los proxy servers (fig 15), que son los intermediarios entre la red interna e internet. Cuando el usuario busca cierta información y esta búsqueda es enviada un proxy recibe esa información, si la tiene la envía pero si no la tiene pública esta búsqueda en Internet. Se puede configurar más de un proxy en la red como en el ejemplo siguiente:

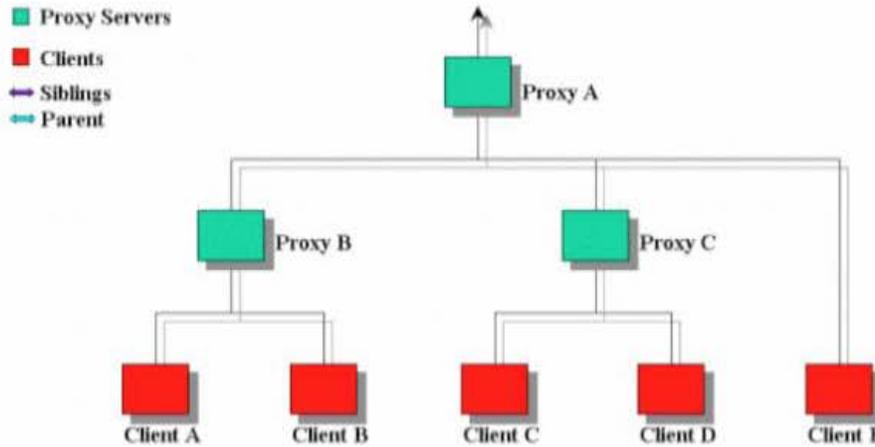


Figura 15. Estructura básica de un proxy

Como tema adicional se tratará de la seguridad que se puede implementar en PHP. Al ser un lenguaje de programación muy flexible, esta misma se puede convertir en su misma vulnerabilidad, ejemplo:

```
<?php
function usuario_autorizado()
{ //Instrucciones para validar al usuario
  return false;
}

// Define $autorizado = true Únicamente si el usuario es auténtico.
if ( usuario_autorizado() )
{
  $autorizado = true;
}
// Como no se inicializa $autorizado = false, esté podría recibir un si
// valor vía el URL y en consecuencia, cualquiera podría autenticarse. GETrs
if ( $autorizado == "true" )
{
  print "Bienvenido a la sección Financiera del Sistema";
  //include "/finanzas/estadoFinanciero.php";
}
else
{
  print "Acceso Denegado";
}
?>
```

•Descripción:

Permite que las variables recibidas por el código PHP desde una forma HTML, automáticamente se conviertan en variables. Desde PHP 4.2.0 ésta variable está deshabilitada.

•Utilidad:

Facilita el uso del lenguaje, es posible acceder a los valores de la forma con sólo hacer referencia a \$<nombreEnLaForma>

•Problema:

Si un usuario malicioso, envía un parámetro al navegador éste lo recibirá y declarará una variable en el código con ese nombre.

Este es sólo un ejemplo de cómo serían los errores de código en php. Estos son los dos tipos de ataque por vulnerabilidad en código más conocidos: SQL Injection y Cross Site Scripting XSS.

SQL Injection es un ataque que aprovecha la mala validación de los datos entrantes por parte del usuario, el cual puede introducir instrucciones SQL que no son válidas en el formato normal del sistema. Esto puede permitir el acceso a bases de datos prohibidas. Este usuario no autorizado podrá hacer con la información de la base de datos lo que él desee, como si fuera un administrador.

Con lo siguiente se explica lo anterior, tomemos por ejemplo la siguiente tabla que contiene a los usuarios de un sistema cualquiera:

·id	·nombre	·edad	·tipo	·clave
·1	·Alejandro	·29	·admin	·secreto
·2	·Cesar	·30	·user	·Mexico23

En esta tabla se observa que sólo contiene dos usuarios, uno que es el administrador y otro que es el usuario común. Cabe mencionar que el administrador sólo se puede insertar por medio de la interfaz SQL y esto corre a cargo del administrador principal. En el formulario que se creó sólo se solicitan datos y se insertaron en la base de datos, sin pedir validación.

```
INSERT INTO usuarios  
( nombre, edad, tipo, clave ) VALUES ( 'Alejandro',29,'admin','secreto' )
```

```
http://localhost/inserta.php?nombre=Cesar&edad=30&tipo=user&clave=Mexico23
```

Este es el código para insertar un usuario con la interfaz de SQL. Y la dirección URL de abajo ejemplifica que se envía a la base de datos por medio del formulario. El resultado de arriba sería el segundo dato visto en la tabla de arriba.

Pero este mismo código puede ser insertado mediante los cuadros de texto destinados a capturar datos. Sólo hay que conocer como insertar caracteres especiales que identifican al código SQL (tabla 51):

Texto	Equivalencia
%27	'
%2C	,
%28	(
%29	)

Tabla 51. Equivalencias de texto en HTML

Si se inserta de manera correcta en el formulario los caracteres seguido del nombre de usuario ficticio se puede crear un usuario con las propiedades de un administrador, esto es lo que se vería si se envía de manera correcta:

```
http://localhost/inserta.php?nombre=hacker%27%2C30%2C%27admin%27%2C%27hacker%27%29%2C%28%27usuario&edad=36&tipo=user&clave=seguro
```

y la tabla resultante sería:

id	nombre	edad	tipo	clave
1	Alejandro	29	admin	secreto
2	Cesar	30	user	Mexico23
3	hacker	30	admin	hacker
4	usuario	36	user	seguro

este es el código SQL enviado a la base de datos:

```
INSERT INTO usuarios ( nombre, edad, tipo, clave ) VALUES  
( 'hacker',30,'admin','hacker'),('usuario',36,'user','seguro' )
```

Como se observa el resultante es la creación de dos usuarios uno inofensivo con propiedades determinadas y otro con control total sobre la base de datos con las mismas propiedades del administrador.

Para este tipo de problemas existe una configuración incluida en php llamada "magic\_quotes\_gpc=off", que es a grandes rasgos una validación de caracteres en el código, es decir sólo evalúa si existen estos tipos de caracteres especiales y los convierte en caracteres permitidos. Cuando se instala por primera vez php esta aparece deshabilitada, sólo tendría que cambiar off por on y listo. Claro hay un sacrificio y esta recae en la velocidad, pero es un riesgo que hay que correr.

El XSS(Cross Site Scripting) busca mostrar un contenido distinto al original a un visitante específico de un sitio. Esto quiere decir que se altera la presentación del contenido más no lo modifica y el usuario lo toma como una nueva actualización permitiéndolo obtener información personal por parte de él.

También se vio la seguridad que puede ofrecer MySQL. Un problema muy común es no asignarle un password al manejador.

Otra inquietud es si se brinda al usuario integridad en sus datos, es decir, si se ofrece seguridad y confidencialidad en sus passwords o en toda la información. En MySQL se puede implementar de manera sencilla ya que cuenta con un modo de cifrado de información.

Hay de dos tipos de cifrado implementados en MySQL y son el modo password que da una cadena de 41 bytes basados en el algoritmo SHA1 y el modo MD5 basado en el algoritmo del mismo nombre, entregando una cadena de 128Bits

Existen herramientas que son de uso web, ya sea para analizar la misma red o las vulnerabilidades existentes. Una de ellas se llama NIKTO que es un analizador de vulnerabilidades para sitios web.

Apache cuenta con su propia seguridad, esta está localizada en el módulo mod\_security que implementa un nivel de seguridad, que permite el análisis y filtrado de los request que recibe un servidor, busca evitar los SQL inyection, y permite correr apache en ambientes Jail.

## Curso Complementario: Programación con Java.

Temario	
1. La tecnología Java	4. Trabajando con objetos en Java
2. Estructura del lenguaje	5. Excepciones
3. Conceptos de la programación orientada a objetos	6. AWT creación de applets y aplicaciones

Java se define como un lenguaje orientado a objetos que permite desarrollar aplicaciones en muy diversas áreas tales como seguridad, animación, acceso a bases de datos, aplicaciones cliente servidor, interfaces gráficas, páginas de Web interactivas, etc.

Una de sus principales características es la creación de módulos reutilizables que funcionan sin la necesidad de conocer la estructura interna, permitiendo al usuario añadir nuevos módulos. De esta manera se pueden obtener programas independientes de la plataforma en la cual fueron desarrollados.

Siendo java un lenguaje de programación, se requiere de traductores, que son los que interpretan el código de alto nivel a código máquina. Existen dos tipos de ellos:

Compiladores: traducen los programas fuente escrito en lenguaje de alto nivel (C/C++, FORTRAN) a lenguajes máquina (0,1); fig 16 .

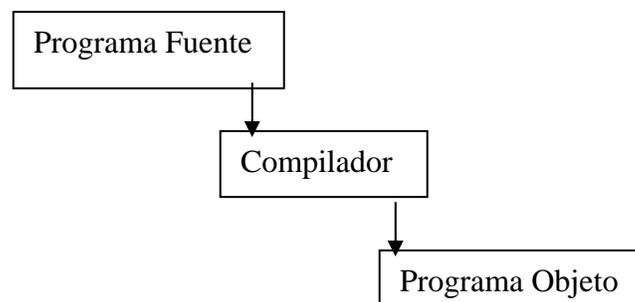


Figura 16. Estructura básica de la compilación

Interpretes: Un intérprete es un traductor que toma un programa fuente, lo traduce y a continuación lo ejecuta. Java es un lenguaje interpretado que traduce los códigos de bytes traducidos a su vez por el compilador Java correspondiente (fig 17).

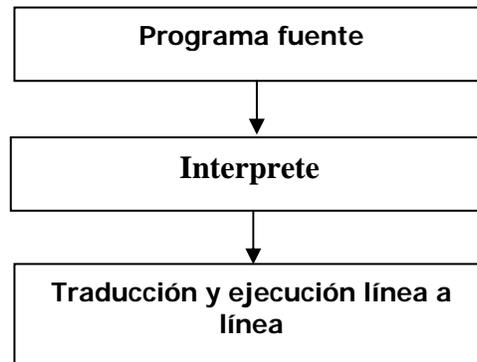


Figura 17: Estructura de un Interprete

Hasta los años sesenta, relativamente poca estructura se había impuesto a la forma en que escribían código los programadores. No había estructuras bien definidas, y en consecuencia, el control de flujo a través de cientos de líneas de código resultaba casi imposible este tipo de programación se le llamo monolito (mono-uno y lithos-piedra).

Ante esta necesidad surgieron dos tipos de programación: la estructurada y POO o programación orientada a objetos.

En la programación estructurada, surgieron las funciones, que son los bloques de la programación, estas funciones se denominaron la extinción del "goto". Al no solicitar la búsqueda por líneas sino por bloques se redujo de manera considerable el código, además de dar un mayor orden en la programación.

La POO utiliza objetos, que son bloques constructores con la cualidad de ser reutilizables, y que dieron surgimiento a los conceptos como clases, objetos herencia, polimorfismo, encapsulamiento.

Clase: es un prototipo o modelo que define las variables y métodos comunes a todos los objetos de un cierto tipo. En pocas palabras, es el molde que se utiliza para crear objetos.

Objeto: es una instancia (un ejemplar, un caso concreto) de una clase. Un objeto tiene dos características: *estado* y *comportamiento*. El estado de un objeto son sus atributos (propiedades) y el comportamiento (operaciones). Por lo tanto, un objeto es una realización concreta de una descripción de una clase. El proceso de creación de objetos se denomina instanciación (crear instancias) de una clase.

Encapsulamiento u ocultamiento de información: que consiste en la combinación de los datos y las operaciones que se pueden ejecutar sobre esos datos en un objeto, impidiendo usos indebidos al forzar que el acceso a los datos se efectúe siempre a través de los métodos del objeto.

Herencia: es la capacidad para crear nuevas clases (descendientes) que se construyen sobre otras existentes, permitiendo que éstas les transmitan sus propiedades. En programación orientada a objetos, la reutilización de código se efectúa creando una subclase que constituye una restricción o extensión de la clase base, de la cual hereda sus propiedades.

Polimorfismo: consigue que un mismo mensaje pueda actuar sobre diferentes tipos de objetos y comportarse de modo distinto. El polimorfismo adquiere su máxima expresión en la derivación o extensión de clases; es decir, cuando se obtienen nuevas clases a partir de una ya existente mediante la propiedad de derivación de clases o herencia.

Java cuenta con su propia plataforma lo que lo hace más flexible al momento de ejecutar en cualquier sistema (fig 18).

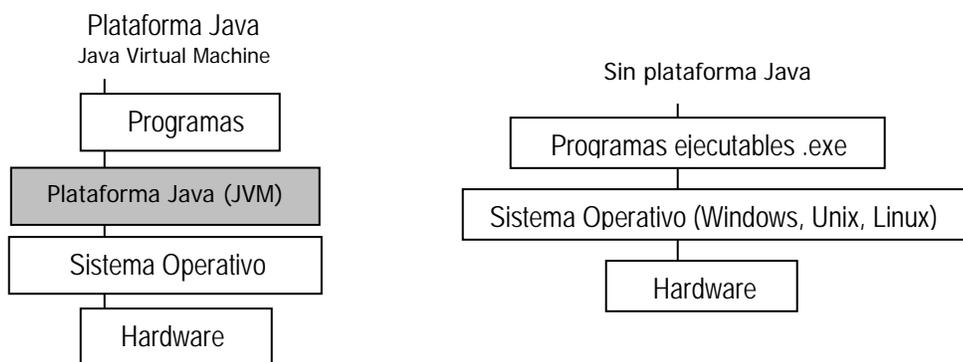


Figura 18. Comparación de plataformas de programación

Este programa es independiente de su plataforma, y no se puede ejecutar directamente por el procesador. En su lugar, una máquina virtual Java (JVM, Java Virtual Machine) ejecuta (interpreta) los bytecodes. Existen numerosas JVM disponibles para una gran variedad de plataformas que permite a los programas Java ser independientes de la plataforma. Por ejemplo, un programa Java compilado en una estación de trabajo UNIX puede ejecutarse en un Macintosh o en Windows 2000. Otra característica de Java proviene de sus bibliotecas incorporadas. Los paquetes que vienen con el entorno de desarrollo Java (JDK, Java Development Kit) contienen muchos centenares de clases integradas, con muchos millares de métodos.

El Java 2 SDK era antiguamente conocido como Java Development Kit (JDK). Hasta la versión 1.1 se conocía el entorno de desarrollo como JDK, y aun así con las versiones de Java 2, muchos programadores siguen conociendo el entorno por JDK.

El Kit de Desarrollo de Software de Java (Software Development Kit) contiene todas las clases que se necesitan para desarrollar programas Java, herramientas de compilación, ejecución y depuración de código. Para desarrollar Java se necesita obtener un SDK, pero está disponible para su descarga gratuita del sitio Java de Sun. Entre otras cosas, el SDK viene con:

**javac** : Un compilador controlado por línea de órdenes.

**java** :Una copia de JVM, el motor que corre códigos de bytes compilados adecuados para su plataforma (Windows, UNIX, Linux).

**jar**: programa para crear ficheros tipo jar, que son archivos comprimidos de java.

**javadoc** : programa para documentar nuestras clases.

**appletviewer** : programa para poder visualizar applets.

La principal ventaja del SDK es su carácter gratuito, autorizado y actualizado. Su principal desventaja es que proporciona sólo herramientas de desarrollo estilo línea de órdenes en vez de un entorno gráfico de programación visual.

Java, según sus creadores, se define como sencillo, orientado a objetos, distribuido, interpretado, robusto, seguro, arquitectura neutra, alto rendimiento, multihilo, dinámico, portable y enfocado a la Red Internet.

Existen diferentes plataformas en la programación con java, dependiendo de la orientación y necesidad del programa. Estas se dividen en tres categorías: J2ME, J2SE, J2EE.

J2ME (Micro Edition):

Proporciona un ambiente robusto, flexible para los usos que funcionan en los dispositivos del consumidor, tales como teléfonos móviles, PDAs, y las cajas de la fijar-tapa de la TV, así como una amplia gama de dispositivos encajados. J2ME incluye las máquinas virtuales de Java y un sistema de Java estándar (APIs) definido como el proceso de la comunidad de Java, por los grupos de expertos y que miembros incluyen a fabricantes del dispositivo, a vendedores del software, y a abastecedores de servicio principales.

J2ME entrega las ventajas de la tecnología de Java al consumidor y a los dispositivos destinados. Incluye interfaces de usos flexibles, un modelo robusto de la seguridad, una amplia gama de los protocolos de red incorporados, y ofrece ayuda extensa para los usos networked y fuera de línea que se pueden descargar dinámicamente. Los usos basados en especificaciones de J2ME se escriben para una amplia gama de dispositivos, y con ello explotan las capacidades nativas de cada dispositivo.

J2SE (Standard Edition):

La interfaz de programación J2SE (API) define la manera prescrita por la cual un applet o una clase, pueden hacer peticiones y utilizar las funcionalidades disponibles en las bibliotecas compiladas de la clase de J2SE. Las bibliotecas de la clase de J2SE son también parte de la plataforma de J2SE.

El J2SE API consiste en tecnologías que son organizadas en dos grupos: Base Java y tablero del escritorio Java.

La base Java proporciona la funcionalidad esencial para los programas empresariales dignos de gran alcance de la escritura en las áreas dominantes tales como acceso de base de datos, seguridad, invocación alejada del método (RMI), y comunicaciones, para nombrar algunos.

Java de escritorio proporciona una gama completa de características para uso de escritorio; la ayuda que proporciona brinda una experiencia total al usuario. Java de escritorio consiste en productos del despliegue tales como plug-in de Java, APIs que modelan el código escrito tal como JavaBeans, interfaz utilizando gráfico (GUI) APIs como las clases de la fundación de Java (JFC) y oscilación, y las multimedias APIs tal como Java3D.

### J2EE (Enterprise Edition):

Esta plataforma ofrece un conjunto de especificaciones y técnicas que proporcionan soluciones completas, seguras, estables y escalables para el desarrollo, despliegue y gestión de aplicaciones en múltiples niveles de funcionalidad basadas en servidores. J2EE reduce el costo y la complejidad de desarrollo, lo cual redundando en rapidez de desarrollo.

La plataforma J2EE define un estándar para el desarrollo de aplicaciones de múltiples niveles (servidores Web, de aplicaciones, de base de datos, etc.). Gracias a que su funcionamiento se basa en componentes modulares que incluyen un conjunto de servicios predefinidos, se simplifica la tarea de la producción de sistemas. J2EE extiende las ventajas de la plataforma Java 2 (como por ejemplo, seguridad, la portabilidad de programas, el acceso a las bases de datos, etc.) con la integración de recursos como Enterprise JavaBeans, Servlets Java, JavaServer Pages, y la tecnología XML.

En este módulo se ocupa la plataforma standard por ser la más usada a nivel comercial por los desarrolladores de software.

Los IDE's son herramientas que proporcionan un entorno de desarrollo que permite la creación rápida de programas. Entre las herramientas más importantes de desarrollo se encuentran las siguientes:

- Café de Symantec
- Sun Java Workshop
- Visual Age for Java by IBM
- Jfactory de Roge Wave
- JCreator
- JBuilder de Borland
- Visual J++ de Microsoft
- Forte de Sun

Los programas java se dividen en dos categorías: las aplicaciones y los applets. Las aplicaciones son programas independientes, como cualquier escrito por un lenguaje de alto nivel, comúnmente llamado programa estructurado y pueden ser ejecutados en cualquier máquina con un interprete java. Los applets son programas incrustados en otro lenguaje, con la propiedad de poder ejecutarse en cualquier navegador web compatible con java. Los applets no pueden leer o escribir en el sistema de archivos de la computadora, pues en caso contrario podrían producir daños en

archivos y propagar virus. No pueden establecer conexiones entre la computadora de un usuario y otra computadora, excepto que sea el servidor donde están almacenados los applets, y no se pueden ejecutar programas de la computadora donde reside el navegador, dado que podría originar daños.

En general, se puede convertir un applet Java para ejecutarse como una aplicación sin pérdida de funcionalidad. Una aplicación no siempre se puede convertir para ejecutarse como un applet, debido a las limitaciones de seguridad de los applets.

Para poder utilizar las herramientas proporcionadas por SDK se debe compilar las variables de entorno path. Como las instrucciones fueron hechas en Linux, el siguiente código muestra las líneas que se tiene que seguir para hacerlo:

```
cd /opt  
PATH = $PATH:/opt/jdk.../bin
```

Ya instalado este paquete se puede ejecutar algunas sentencias, si se desea saber la versión de java que se tiene solo se ejecuta `java -versión` en el prompt de Linux.

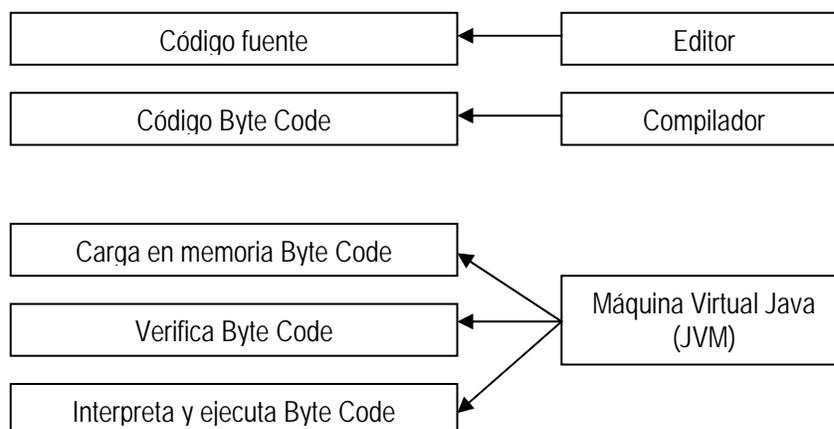
Para poder compilar un applet sin la necesidad de un IDE sólo se escribe `javac` seguido de la ruta del programa que deseamos compilar: El applet es creado a partir de un documento de texto con la extensión `.java`, en el contendrá una línea para imprimir la línea hola mundo. Esto quedaría así suponiendo que se encuentra en una carpeta que se llama `JavaApplets` en la carpeta de `root`:

```
javac /root/JavaApplets/MyApplet.java
```

Ahora para poder visualizar el código de instrucciones, al ejecutar `javaviewer` seguido de la ruta del nuevo archivo creado tras la compilación tendrá una extensión de `.class`:

```
javaviewer /root/Javaapplets/MyApplet.class
```

El siguiente esquema muestra (fig 19) como sería un método de programación en java:



*Figura 19. Comparativos de programación Java con otros*

Un programa en java esta compuesto por clases. Uno de los métodos obligatorios en la clase principal es el main(). Un método en una clase es un grupo de instrucciones que realiza una o más acciones. Este tipo de programas debe de contener al inicio de cada código algunos import que incluirán a su vez varios applets y librerías.

La siguiente estructura representa un applet básico:

```
import java.lang.*; //Archivo de clases básicas en JAVA.
Modificador_de_acceso class NombrePrograma //Nombre de la clase
{
    public static void main(String ar[]) //Cabecera del método
    {
        //Sentencias
        System.out.println("Bienvenido a la programación en JAVA");
    }

    //Definición de otros métodos
    modificador _de_acceso tipo_devuelto nombre_metodo(parámetros)
    {
        //Declaraciones locales
        //Sentencias
        //return expresión
    }
}
```

Siempre el nombre del archivo fuente ha de ser el nombre de la clase con la extensión *java*, de lo contrario el compilador marcara error.

Los métodos en Java se definen en la clase a la que pertenecen.

```
Tipo_retorno nombreMétodo (lista_de_parámetros)
{
    sentencias
    return expresión
}
```

tipo\_retorno            Es el tipo de valor devuelto por la función, si es de tipo void, el método no devuelve nada.

nombre\_Método        Nombre del método

lista\_de\_parámetros   Lista de parámetros, pasados al método. Un método puede no recibir ningún parámetro.

Un comentario es cualquier información que se añade en las líneas del programa para proporcionar información de cualquier tipo. El compilador ignora los comentarios, no se realiza ninguna tarea concreta. En Java, los comentarios de un programa se puede introducir de dos formas:

```
//           para comentarios en línea.  
/* .....*/ para más de una línea.  
/** ...*/   para poner comentarios en la documentación creada por el programa javadoc.
```

Para poder utilizar IDE's debemos instalar el paquete JDK para eso, descargamos el paquete bin (ejecutable) y hacemos lo siguiente desde la consola de mando:

```
chmod 755 jdk-1_5_0_02-linux-i586.bin  
cp jdk-1_5_0_02-linux-i586.bin /opt/  
./jdk-1_5_0_02-linux-i586.bin  
cd /opt  
./jdk-1_5_0_02-linux-i586.bin  
PATH=$PATH:/opt/jdk1.5.0_02/bin/
```

Estas instrucciones nos refieren a cambiar el modo del archivo haciéndolo ejecutable. Una vez hecho esto se ejecuta y se espera a que termine de instalarse, y para terminar se adiciona al path del sistema (la copia del archivo a la carpeta /opt es opcional).

Ahora se instala el IDE, para éste, se descarga netbeans 3.6, con la siguiente serie de comandos se instala el IDE:

```
chmod 755 nb-3_6-as8pe-linux.bin  
./nb-3_6-as8pe-linux.bin  
cd /opt/NetBeans3.6/  
ls  
./bin/runide.sh &  
cd bin/  
ls  
ln -s /opt/NetBeans3.6/bin/runide.sh /root/Desktop/
```

Con esto queda configurado el IDE teniendo un acceso directo al escritorio. Durante el curso se llevaron a cabo practicas para poner en practica la teoría que se iba explicando haciendo énfasis en las propiedades del lenguaje Java.

## *Objetivo*

La problemática que se encuentra aún latente en algunas unidades médicas, es la de no contar con un sistema que agilice las actividades del personal que llevan a cabo la recepción y canalización de pacientes y todo lo referente a su atención. La complicada fluidez de trámites, provoca que se atrase de manera considerable la atención a pacientes, desde que ingresa, ya sea a consulta externa o urgencias pasando por admisión, hasta su salida de la unidad médica. Entre los múltiples problemas que existen están:

- La admisión
- La atención de urgencias
- El manejo de citas
- Y el control de insumos

Con respecto a la admisión se encuentra el problema del largo tiempo que tiene que esperar el paciente para su atención. Por ejemplo, en algunas unidades de servicios médicos, al solicitar el servicio en el área de urgencias, el paciente debe registrarse en el departamento de admisión para que sus datos personales sean capturados. Si ya se encuentra afiliado y dado de alta en su base de datos, ¿porque deben de recapturar los datos cuando vuelve a acudir a solicitar servicios médicos?. Una vez capturados los datos personales hay que esperar a que el personal entregue la hoja a urgencias y esta sea repartida a los distintos consultorios existentes, habiendo la posibilidad de extraviarse o traspapelarse provocando que se haga más largo el tiempo de espera.

En el servicio de urgencias se maneja una cantidad importante de pacientes, con lo que pueden traspapelarse las hojas sueltas, por parte de cualquiera de los que tienen acceso a éstas, por otro lado, como es un servicio casi impredecible, debe existir un estricto orden para no perder la secuencia entre la atención de un paciente a otro, ya que el tiempo de atención de cada uno de los pacientes difiere de la causa de solicitud de consulta y/o de sus antecedentes, además de los ingresos de ambulancias que se atienden en este servicio.

Debe existir un control óptimo en cuanto a las entradas y salidas de materiales y medicamentos para un mejor servicio, en base a esto se debe tener en cuenta tres puntos básicos:

- Abastecimiento, porque es importante conseguir que los proveedores surtan los recursos materiales y medicamentos cuando son requeridos, teniendo en cuenta que la demanda sea diferente a la pronosticada y así evitar el desabasto de los que son de mayor demanda.
- Caducidad, porque se debe tener un inventario de seguridad del producto calendarizado para así promoverlo en la unidad y/o el cambio oportuno con el proveedor y así no rebasar la fecha de caducidad, y por lo tanto no desperdiciar estos insumos.
- El evitar fugas de medicamentos o materiales proporcionando así un ahorro importante y limitar el mal uso de estos insumos que pueden provocar el desabastecimiento.

El Sistema Gestor de Información y Control propone ser una herramienta que proporcione elementos de información para el manejo y control de los servicios y recursos materiales de la unidad médica, optimizando la atención médico-paciente y la mejora en la utilización y resguardo de los insumos de la misma.

## Estructura interna

La estructura del Sistema Gestor de Información y Control se encuentra conformado por 4 grupos básicos: administrativo, personal, insumos y pacientes. La idea de dividir así la base de datos es porque se trata de cubrir los puntos esenciales que cumplan con las expectativas de este sistema y su objetivo: el dar un buen servicio. El diagrama de datos general queda graficado de la siguiente manera:

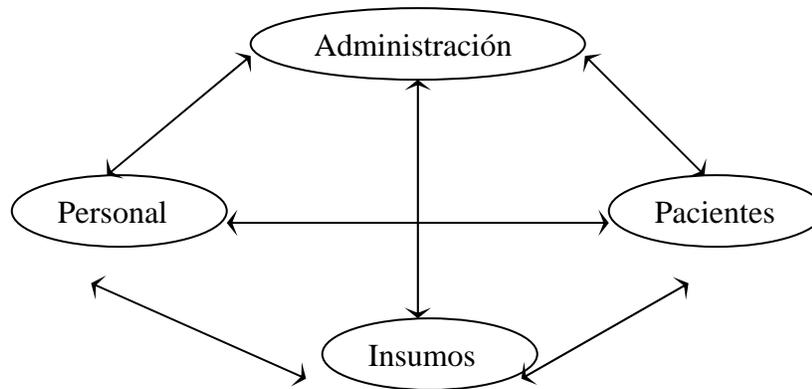


Figura 20. Esquema general de la base de datos

La forma que adquiere la estructura es en forma de red, ya que de esta manera se tendrá un mejor y más ágil manejo de los datos.

El grupo de insumos esta constituido por todo tipo de material ya sea farmacéutico o de tipo consumible (gasas, vendas, jeringas, lancetas, etc). Para tener un control sobre este inventario se crearon tres tablas: In\_T1, In\_T2, In\_T3 (Insumo tabla #). La primer tabla registra las características únicas de cada insumo asignándole un ID propio y brindando la información sobre su nombre, marca registrada (que es la sustancia activa con respecto a los medicamentos), su registro en la secretaria de salud y su código de barras para agilizar su captura en caso de contar con escáner. La segunda tabla esta constituida por campos que ayudaran a controlar los insumos con el proveedor (ya sea fabricante o laboratorio), esta tabla asigna un id por lote según como venga descrito por el proveedor, marca su fecha y hora de provisión, la cantidad que contiene cada lote, el nombre del proveedor, y un campo de descripción, en este campo se anota cualquier dato relevante sobre el lote o característica del insumo. Existe un campo más que se llama Alta/Baja que indica si ese lote se encuentra en óptimas circunstancias para su uso y distribución, esto es debido a que si por alguna razón sufra algún percance, ya sea en su transporte o ya dentro de almacén.



La tabla dos tiene en su deber el trabajo de controlar el insumo de manera individual con apoyo de las otras dos ya mencionadas. Guarda la fecha de elaboración, caducidad y cantidad por lote. El campo de caducidad sirve para dar referencia a un insumo que esté próxima su fecha y por medio del programa sea descartado al momento de capturarlo y darlo de baja del sistema. Con esto se evita su distribución y que el material no llegue a causar daños a terceros. El campo de cantidad lleva un conteo sobre los insumos distribuidos, comparando su resultado con el inicial y sacando estadísticas.

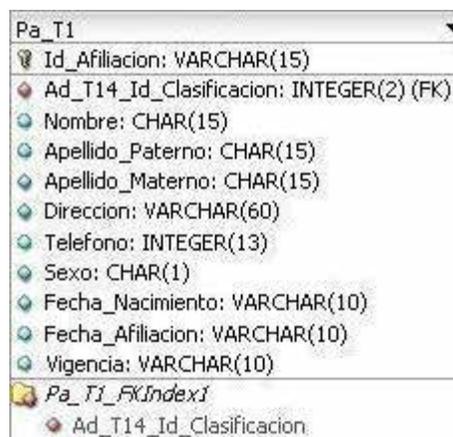
Este grupo de tablas trata de eliminar la necesidad de dejar materiales inservibles dentro de almacén, y tener un mejor control sobre ellos, además de agilizar la entrega de estos a los usuarios finales.



*Pe\_T1. Tabla de personal*

El siguiente grupo es de personal. Este grupo tiene la finalidad de mantener un orden sobre la información pertinente a los empleados que laboran ahí. En esta tabla se muestra los datos generales de cada personal como lo es el identificador único, su número de empleado, nombre, apellidos, sexo, dirección, teléfono, fecha de nacimiento (que ayudara a obtener su edad dinámicamente), su identificador de cargo, que con la ayuda de una tabla externa se obtiene el nombre del cargo, ahorrando espacio en memoria, su identificador de la unidad médica donde labora proporcionado por la secretaria de salud, su fecha de adscripción a esa unidad y su identificador de área dentro de la unidad. Las tablas aledañas apoyan a la tabla principal para evitar redundancia y mal manejo de memoria, pero estas pertenecen al grupo administrativo. La sigla de esta tabla es Pe\_T1(Personal Tabla 1).

El grupo de pacientes contiene cinco tablas básicas: datos personales, urgencias, consulta externa, carnet de citas, terapias y estudios, y sus notas médicas. La tabla 23 (Pa\_T1) almacena los datos personales de cada paciente que requiera de un seguimiento en su caso clínico, esto significa que se la hace su historia clínica en esa institución. En caso de requerir urgencias pero no de llevar un seguimiento o visita única, sus datos serán igualmente capturados pero en un tabla (NoExp) completamente independiente de este sistema salvo la id del empleado que lo



*Pa\_T1. Captura de datos de nuevo expediente*

atendió. Los datos ahí capturados llevarán un cronómetro de vida no mayor a seis meses (se ejecutara una limpieza automática y quedarán eliminados para ahorrar espacio), en caso de recibir solicitud por parte de un agente externo de una hoja clínica, se le podrá entregar una copia con los datos que solicite.

*NoExp. Solicitud de urgencias sin expediente*

acompañada la persona deben de capturarse datos personales de el como son su nombre, parentesco, domicilio, teléfono y edad.

La tabla Pa\_T2 trata de eliminar la recaptura de datos de un mismo paciente reservando memoria a datos clínicos. Esta tabla es de urgencias y en ella se captura datos necesarios al ingreso del

*Pa\_T2. Eliminación de recaptura de datos*

paciente como son el folio (número de atención de ese año), su fecha y hora de admisión a la institución, si viene como son su nombre, parentesco, domicilio, teléfono y edad.

El campo consultorio es dinámico, porque elige de manera equitativa el consultorio donde será llamado y atendido, evitando el traspapeleo del paciente y agilizando su atención. La fecha y hora de alta será capturada por el personal de admisión hospitalaria. Si este campo se mantiene vacío, la id de afiliación del paciente se encontrara bloqueada para evitar su uso por diferentes personas.

*Pa\_T3. Control de consulta externa*

urgencias con la diferencia que esta manejada no por admisión hospitalaria sino por su carnet. La tabla PA\_T5 contiene los datos relacionados con las citas de cada paciente subdividiéndolo en terapias, consultas y auxiliares de diagnóstico. Como se observa a diferencia de las demás tablas que maneja la fecha como una sola línea esta la

La tercer tabla (Pa\_T3) es la de consulta externa que es similar a la tabla de

*Pa\_T5. Control de citas (carnet)*

clasifica en dos campos: mes y día, con la finalidad de darle un mayor dinamismo a la hora de consultar la agenda del médico como se verá más adelante en el desarrollo del sistema.

Pa_T4	
F/H_Respuesta:	VARCHAR(15)
Pa_T2_Id_Folio:	VARCHAR(10) (FK)
IDX:	VARCHAR
Nota:	VARCHAR
PTA:	INTEGER(1)
PTS:	INTEGER(1)
<i>Pa_T4_FKIndex1</i>	
Pa_T2_Id_Folio	

*Pa\_T4. Control de cambio de turno*

La tabla Pa\_T4 da el beneficio al problema del cambio de turno y redundancia a datos ya capturados. Los campos que tiene esta tabla son: la fecha y la hora de respuesta (cuando entra a consultorio), su IDX (causa o motivo de visita), nota (descripción del caso), y PTA (Pendiente Turno Anterior), PTS (Pendiente Turno Siguiente). Estos últimos dos campos ayudan a controlar el cambio de turno, si el caso es de un caso anterior este se verá reflejado en la tabla y si es requerido dejarlo a observación durante el turno siguiente se

almacenara en el campo PTS. La razón para la creación de esta tabla es, eliminar la búsqueda innecesaria de las notas anteriores de ese paciente ya impresas, al tenerlas a la mano en el momento que lo demande.

Finalmente el grupo de administración maneja todo respecto a papeleos internos, ya se hable de licencias o de refrendos de personal. La función de esta tablas es la de evitar duplicaciones y el mal uso de papeles dentro del instituto. Este grupo es el más amplio en tablas que los anteriores, el cual tiene en su haber trece tablas. Dentro se dividen en subgrupos: licencias, recetas y referencia de la unidad. Las tablas de licencias comprende desde su introducción al instituto como su distribución al usuario final.

Ad_T5	
Id_Lote_LM:	VARCHAR(6)
Ad_T3_Id_UM:	INTEGER(2) (FK)
Id_LM_j:	INTEGER(5)
Id_LM_f:	INTEGER(5)
<i>Ad_T5_FKIndex1</i>	
Ad_T3_Id_UM	

*Ad\_T5. Registro por lote de licencias médicas*

Ad_T6	
Id_LM:	INTEGER(5)
Pe_T1_Id_Personal:	VARCHAR(6) (FK)
No.Serie_j:	VARCHAR(12)
No.Serie_f:	VARCHAR(12)
Alta/Baja:	INTEGER(1)
<i>Ad_T6_FKIndex1</i>	
Pe_T1_Id_Personal	

*Ad\_T6. Control de licencias médicas por libro*

Ad_T7	
No.Serie:	VARCHAR(12)
Pa_T1_Id_Afiliacion:	VARCHAR(15) (FK)
Ad_T10_Id_SO:	INTEGER(2) (FK)
Ad_T9_Id_CLM:	INTEGER(2) (FK)
Ad_T8_Id_MLM:	INTEGER(2) (FK)
Fecha:	VARCHAR(10)
Dias_Otorgados:	INTEGER(12)
Fecha_Inicial:	VARCHAR(10)
Fecha_Terminal:	VARCHAR(10)
Alta/Baja:	INTEGER(1)
<i>Ad_T7_FKIndex1</i>	
Ad_T8_Id_MLM	
<i>Ad_T7_FKIndex2</i>	
Ad_T9_Id_CLM	
<i>Ad_T7_FKIndex3</i>	
Ad_T10_Id_SO	
<i>Ad_T7_FKIndex4</i>	
Pa_T1_Id_Afiliacion	

*Ad\_T7. Control de licencias médicas individuales*

La tabla Ad\_T5 almacena la identificación única por lote y contiene dos campos para saber de que número de id a que id van las licencias, además de contener a que unidad médica fueron enviados. La siguiente tabla (Ad\_T6) indica a que médico fue encomendado tal número de licencia y de que número de serie inicial final le toco, adjunto a estos campos tiene uno para bajas o altas, en caso de extraviar el libro de licencias este será cancelado de la base de datos y no podrá ingresar ninguna de sus números

de licencias para usarlo. Cuando sea confirmado por la unidad será dado de baja permanente. La tabla subsecuente (Ad\_T7) en orden jerárquico es la de los datos de cada licencia, esto es a que paciente fue dada, su fecha y hora, los días otorgados, la fecha inicial y final de los días dados, y un campo para dar de baja en caso de que se requiera cancelar esta licencia, los otros tres campo son llaves foráneas que apuntan a los motivos de licencia, su carácter de licencia y el servicio otorgado (Ad\_T8, Ad\_T9, Ad\_T10).

Ad_T8	
Id_MLM: INTEGER(2)	
Motivo_Licencia: CHAR(20)	

*Ad\_T8 Motivo de solicitud de licencia*

Ad_T9	
Id_CLM: INTEGER(2)	
Caracter_Licencia: CHAR(20)	

*Ad\_T9 Caracter de solicitud de licencia*

Ad_T10	
Id_SO: INTEGER(2)	
Servicio_Otorgado: INTEGER	

*Ad\_T10 Servicio otorgado en urgencias*

Las tablas Ad\_T11, Ad\_T12 y Ad\_T13 agrupan las tablas respectivas a las recetas médicas. Tiene el mismo manejo que las tablas especializadas; administran la entrada del lote de recetas incluyendo los números de serie iniciales que contiene, la asignación de un libro de recetas a cada médico y su número de serie inicial final, lo único que cambia es que en la receta se incluyen el nombre el id del medicamento además de indicar la cantidad de estos. También por seguridad se incluye el campo de baja para dar de baja el libro completo o sólo la receta cancelada, y el campo donde se captura quien surtió esa receta.

Ad_T11	
Id_Lote_RM: VARCHAR(6)	
Ad_T3_Id_UM: INTEGER(2) (FK)	
Id_RM_i: INTEGER(5)	
Id_RM_f: INTEGER(5)	
<i>Ad_T11_FKIndex1</i>	
Ad_T3_Id_UM	

*Ad\_T11. Registro por lote de recetas médicas*

Ad_T12	
Id_RM: INTEGER(5)	
Pe_T1_Id_Personal: VARCHAR(6) (FK)	
No.Serie_i: VARCHAR(12)	
No.Serie_f: VARCHAR(12)	
Alta/Baja: INTEGER(1)	
<i>Ad_T12_FKIndex1</i>	
Pe_T1_Id_Personal	

*Ad\_T12. Control de recetas médicas por libro*

Ad_T13	
No.Serie: VARCHAR(12)	
Pe_T1_Id_Personal: VARCHAR(6) (FK)	
In_T1_Id_Clave: INTEGER(7) (FK)	
Pa_T1_Id_Afiliacion: VARCHAR(15) (FK)	
Cantidad_Med: INTEGER(2)	
Alta/Baja: INTEGER(1)	
<i>Ad_T13_FKIndex1</i>	
Pa_T1_Id_Afiliacion	
<i>Ad_T13_FKIndex2</i>	
In_T1_Id_Clave	
<i>Ad_T13_FKIndex3</i>	
Pe_T1_Id_Personal	

*Ad\_T13. Control de recetas médicas individuales*

La tabla Ad\_T1 sólo maneja las fechas de refrendo del personal que se debe realizar cada año o cada vez que sea necesario. Ad\_T2 indica los distintos tipos de cargo que puede tener un empleado en la dependencia y Ad\_T4 el área donde puede laborar. La fecha en que fue adscrito el empleado esta almacenado en Ad\_T3.

Ad_T1	
Pe_T1_Id_Personal	VARCHAR(6) (FK)
Fecha_Refrendo	VARCHAR(10)
<i>Ad_T1_FK.Index1</i>	
Pe_T1_Id_Personal	

*Ad\_T1. Refrendo de credenciales de personal*

Ad_T2	
Id_Cargo	INTEGER(2)
Cargo	CHAR(20)

*Ad\_T2 Cargo que ocupa en la institución*

Ad_T3	
Id_UM	INTEGER(2)
Adscripcion	VARCHAR(20)

*Ad\_T3. Fecha de adscripción*

Ad_T4	
Id_Area	INTEGER(2)
Area	CHAR(20)

*Ad\_T4. Área donde labora dentro de la institución*

Ad\_T14, Ad\_T15, Ad\_T16, Ad\_T17 y Ad\_T18 se dirigen al paciente. La primera al clasificar el tipo e estudio que ofrece la institución, la segunda al servicio que ofrece. La tercera y cuarta sirven de apoyo a Pa\_T5, que es el carnet del paciente; la suma de estas tres tablas da como resultado la agenda del médico, misma que servirá al momento de la consulta.

Ad_T14	
Id_Clasificacion	INTEGER(2)
Clasificacion	CHAR(15)

*Ad\_T14. Clasificacion del paciente*

Ad_T15	
Id_Estudio	INTEGER(2)
Estudio	CHAR(10)

*Ad\_T15. Tipos de estudios que se ofrecen*

Ad_T16	
Id_Servicio	INTEGER(2)
Servicio_2	CHAR(10)

*Ad\_T16. Tipos de servicio que se ofrecen*

Ad_T17	
Id_Mes	INTEGER(12)
Mes	CHAR(9)

*Ad\_T17. Control de meses para manejo de citas*

Ad_T18	
Id_Dia	INTEGER(5)
Dia	INTEGER(2)

*Ad\_T18. Control de días para manejo de citas*

Para concluir el diagrama de la base de datos final quedaría de la siguiente manera, ya con sus ligas incluidas que manejarán la información:

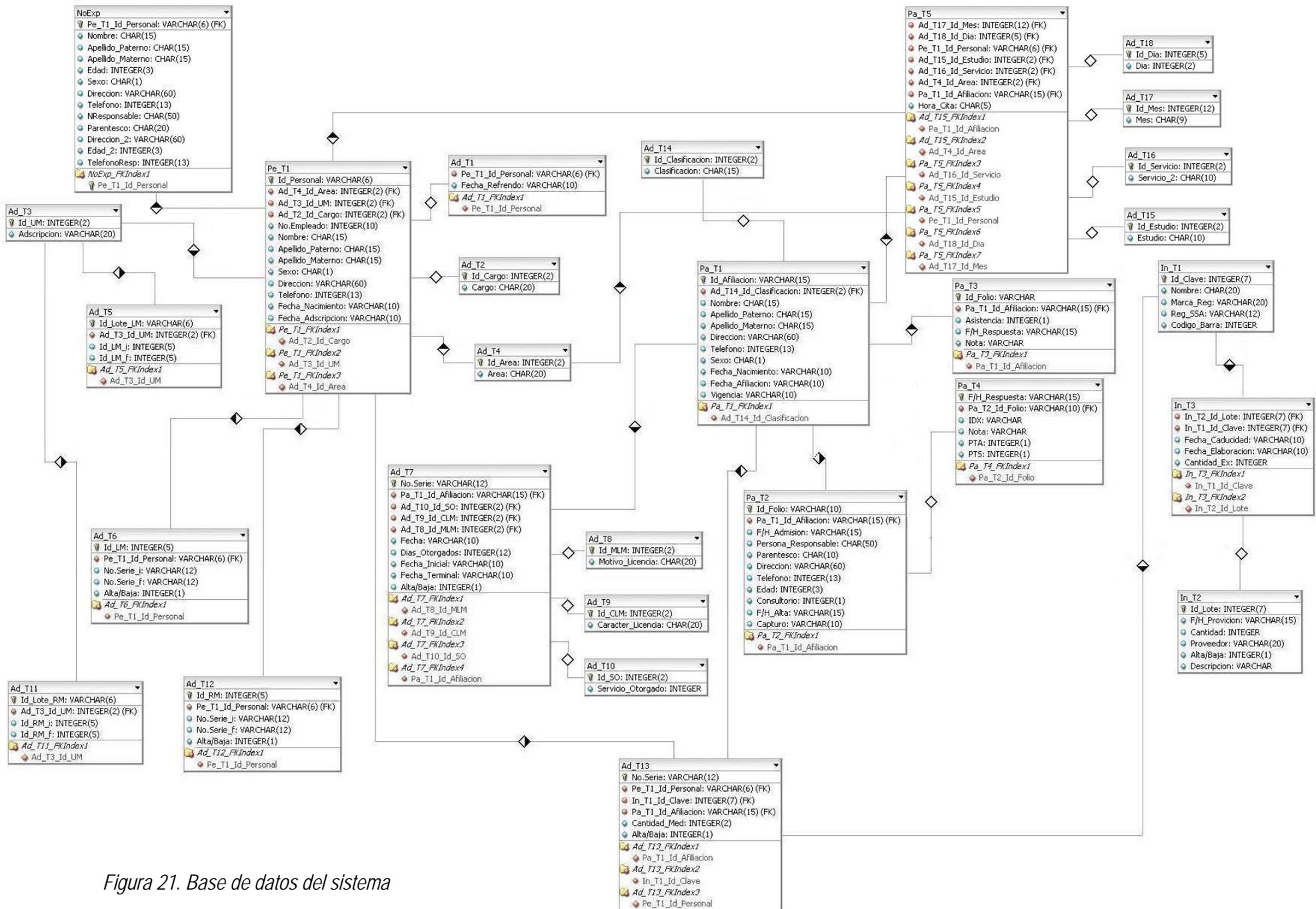


Figura 21. Base de datos del sistema

## Sistema

Para continuar el sistema será desarrollado por obvias razones, con el software que fue utilizado durante el diplomado.

Como Sistema Operativo en el servidor se utilizara Linux Slackware, que por descripción ya mencionada, es robusto y flexible para las aplicaciones que se requiera, además de contar con licencia gratuita. Como interfaz dentro de la Intranet se utilizará HTML y para captura y manejo de datos se utilizara PHP; sin embargo para la verificación de password y seguridad se hará a través de Java. Como administrador de servidor se contará con Apache implementando la seguridad correspondiente y finalmente como manejador de la base de datos se usará MySQL.



Figura 22. Ingreso al sistema

usuario conectado a él:

La Intranet tendrá una interfaz gráfica que facilitara de manera notoria su manejo, además de ser intuitivo para el usuario. Cuando el personal requiera de ingresar su nip, se ejecutara una aplicación java de seguridad (fig. 22) para verificar su autenticidad, si ésta es denegada simplemente el sistema seguirá bloqueado. En cambio si es verídica la clave este ingresará a la aplicación correspondiente a su función. La siguiente imagen (fig. 23) es la presentación del sistema a todo



Figura 23. Pantalla general del sistema

En ella se puede apreciar que se cuenta con un menú respectivo a las actividades que controla el sistema, ya dependa de la actividad a realizar, el usuario sólo deberá seleccionar posicionando el puntero del mouse y dar clic sobre el link. Si una persona no esta autorizada para usar esa aplicación aunque sea empleado no podrá usar esa aplicación, por ejemplo si un médico desea ver el sistema de farmacia no podrá hacerlo.

Tomando primeramente al personal farmacéutico. Cuando se ingrese su nip autorizado, aparecerá una pantalla como se muestra en la fig. 24:

The screenshot displays the 'SGIC Farmacia' web application. At the top, there are logos for 'SESA' and 'SALUD'. Below the header, there are two main navigation buttons: 'ADMINISTRAR LOTE' and 'IMPRIMIR INFORME'. The interface is divided into several functional areas:

- Search Section:** Includes 'Clave Med.' and 'Nombre:' input fields, a 'BUSCAR' button, and a 'Verificar' button next to the 'No. Serie:' field.
- Medicamentos Bajo ese Nombre:** A list of search results for 'Ambroxol', showing dosage (250 mg or 500 mg), form (Tabletas or Frasco), and availability (Agotado or available). The selected item is 'Ambroxol 250 mg Frasco GU3217'.
- Datos Receta Medica:** A table displaying prescription details:
 

No. Serie:	RM123456789
Médico:	Escalante Morales Ruben
Expediente:	ELAJ121212999
Nombre:	Escutia Lazcano Jose David
Medicamento:	Ambroxol 250mg frasco
Cantidad:	2
- Capturo:** A field for 'ID o Nombre de Empleado' with an 'Aceptar' button.

Figura 24. Sistema del área de farmacia

En ella el usuario ingresará el número de serie de la receta que esta recibiendo este será verificado en el sistema y si es correcto le aparecerá a quien pertenece esta receta (personal médico) y a quien fue entregada por medio del ID de afiliación. De ser necesario al no conocer la clave del medicamento podrá ingresar el nombre del mismo y seleccionar de un menú que aparezca a su lado izquierdo cual es que corresponde a la búsqueda. Ese menú entregará nombre, tipo (tabletas o frasco) y si se encuentra agotado o no.

Si por alguna razón esta receta fue cancelada pero aún así fue presentada en ventanilla de farmacia el sistema la reconocerá como cancelada y no podrá ser surtida para evitar distribución ilegal de medicamento. Cada vez que ingrese (por teclado o si cuenta con escáner) un medicamento su id de empleado será almacenado en la base de datos para saber quien surtió esa receta. Una sola pantalla pero funcional en su momento.

También existe una sección en la que se podrá dar un "mantenimiento" al control de los lotes de cada medicamento, como muestra (fig. 25) la pantalla siguiente:



Figura 25. Impresión en pantalla del informe por lotes

En esta pantalla sólo se dará un informe completo por búsqueda específica, esto es, entregara información respectiva a cierto insumo desde la primer entrega de lote por parte del laboratorio o fabricante.

Además en esta área se contará con una pantalla extra para poder ingresar los nuevos lotes de medicamentos con sus

características ya especificadas en la estructura interna o darlos de baja por alguna razón ya sea el lote completo o por medicamento individual sin requerir de ingresar un número de serie válido para hacerlo (fig. 26).



Figura 26. Ingreso de nuevos lotes de insumos

El área de admisión hospitalaria tendrá acceso a su propia información. La información que a él le compete es el ingreso de pacientes tanto a urgencias como a nuevo ingreso y asignación de clasificación. Cuando ingrese su clave aparecerá esta pantalla (fig. 27):

The screenshot shows the 'SGIC Admision' web application interface. At the top, there are logos for the Secretaría de Salud (SSA) and the SGIC. Below the logos, there are navigation tabs: 'AFILIADO', 'NO AFILIADO', and 'ACEPTAR'. The main content area is divided into several sections:

- EXP.:** A text input field.
- NOMBRE:** A text input field containing 'Escutia' and a 'BUSCAR' button.
- PACIENTES POSIBLES:** A list of search results with checkboxes:
  - ELOD121565000 Escutia Lopez David, Camelia 51 #765 Col.Aragon CP 07645
  - ELAJ125482999 Escutia Lazcano Jose David, Portales #45 Col.Neza CP 04545
  - ELAR549145002 Escutia Lara Roberto, Camelia 51 #765 Col.Aragon CP 07645
 An 'Aceptar' button is located below this list.
- Paciente:** A form with the following fields:
  - Folio: UM123548
  - Expediente: ELAJ121212999
  - Nombre: Escutia Lazcano Jose David
  - Dirección: Camelia 654 #54 Col. Guerrero CP 164564
  - Teléfono: 55555555
  - Edad: 17
  - Sexo: Masculino
  - Clasificación: 1
  - Vigencia: 12/12/2005
  - Consultorio: 3
- Responsable:** A form with the following fields:
  - Nombre: [input field]
  - Parentesco: [input field]
  - Dirección: [input field]
  - Teléfono: [input field]
  - Edad: [input field]
  - Capturo: [input field] with a dropdown menu showing 'Id\_Empleado o Nombre'
 An 'Aceptar' button is located at the bottom right of this section.

Figura 27. Pacientes con expediente médico

Si es usuario ya adscrito sólo se hallará necesario ingresar su id de afiliación, en caso contrario deberá seleccionar nuevo ingreso y sus datos serán recopilados. Si se cuenta con acompañante o sin él, esta pantalla permitirá ingresarlo temporalmente y mantenerlo accesible durante su tratamiento (como se menciono arriba sólo durante seis meses). Esta sería la pantalla de no contar con expediente (fig. 28):

The screenshot shows the 'SGIC Admision' web application interface for admission without a medical record. It features the same top navigation and logos as Figure 27. The main content area is divided into:

- Paciente:** A form with the following fields:
  - Folio: UM123548
  - Consultorio: 3
  - Nombre: [input field]
  - Dirección: [input field]
  - Teléfono: [input field]
  - Edad: [input field]
  - Sexo: [input field]
- Responsable:** A form with the following fields:
  - Nombre: [input field]
  - Parentesco: [input field]
  - Dirección: [input field]
  - Teléfono: [input field]
  - Edad: [input field]
  - Capturo: [input field] with a dropdown menu showing 'Id\_Empleado o Nombre'
 An 'Aceptar' button is located at the bottom right of this section.

Figura 28. Admisión hospitalaria sin expediente médico

Ahora se pasa a la presentación por parte del médico de urgencias. Este es más complejo que el anterior pues cuenta con un control de licencias médicas y pacientes. Cuando ingrese su password tendrá esta pantalla en el monitor (fig. 29):



Figura 29. Consulta de urgencias

Como se observa aquí, tendrá una lista de los pacientes correspondientes a su consultorio, que se mantendrá visible todo el tiempo para llevar su control personal. El médico sólo necesita llamar al paciente correspondiente y seleccionarlo con el mouse para que aparezcan sus datos personales en pantalla. Después de revisión anotara su IDX que no es más que el diagnóstico resumido y a continuación ingresar en el campo correspondiente su nota.

En la parte inferior-central se cuenta con un checkbox (fig. 29) que sirve para indicar si el paciente queda pendiente al siguiente horario por cambio de turno.



Figura 30. Alta o baja de licencias médicas

Si el paciente requirió de licencia médica, el personal médico deberá ingresar el número de serie de la licencia médica. Si el médico requiere cancelar esa licencia deberá ingresar a una aplicación (fig. 30) que se encuentra en el menú y de nuevo ingresar su password y cancelarla, si por alguna razón su nip y la id de la licencia no son correlacionales, el sistema simplemente denegará la acción.



Figura 31. Alta o baja de recetas médicas.

Igualmente sucede con las recetas médicas, el médico únicamente dentro de la misma pantalla (fig. 31) ingresará el número de serie de la receta o de que número a que número de serie corresponde por paciente y ésta será relacionada con su id de afiliación así sólo el podrá "cobrar" esa receta. De la misma manera si desea cancelar esa receta deberá ingresar al menú correspondiente y digitar su clave para cancelarla, contando con la misma seguridad que la recetas.

Ahora, si el paciente no responde al llamado por tres ocasiones será cancelada su admisión y será dado de alta.

Si por alguna situación el médico requiere de ver la historia clínica del paciente tendrá la facilidad de hacerlo sólo si el paciente cuenta con ella. En ella se presentara (fig. 32) los datos completos del paciente y las notas clínicas que haya tenido, incluyendo los datos de los médicos de cada nota y fecha y hora en que fueron anotados. Si la necesita esta sería la pantalla que vería:



Figura 32. Historia Clínica

Para facilitar la tarea del médico al final de su día se incluye un botón para imprimir su informe diario de actividades el lugar de tener que volver a capturar todo.

En lo que respecta a la consulta interna esta es controlada no por admisión sino por la agenda personal de cada médico. En ella esta contenida todos los pacientes que tienen cita organizados por día y hora. Su pantalla (fig. 33) correspondiente es la siguiente:

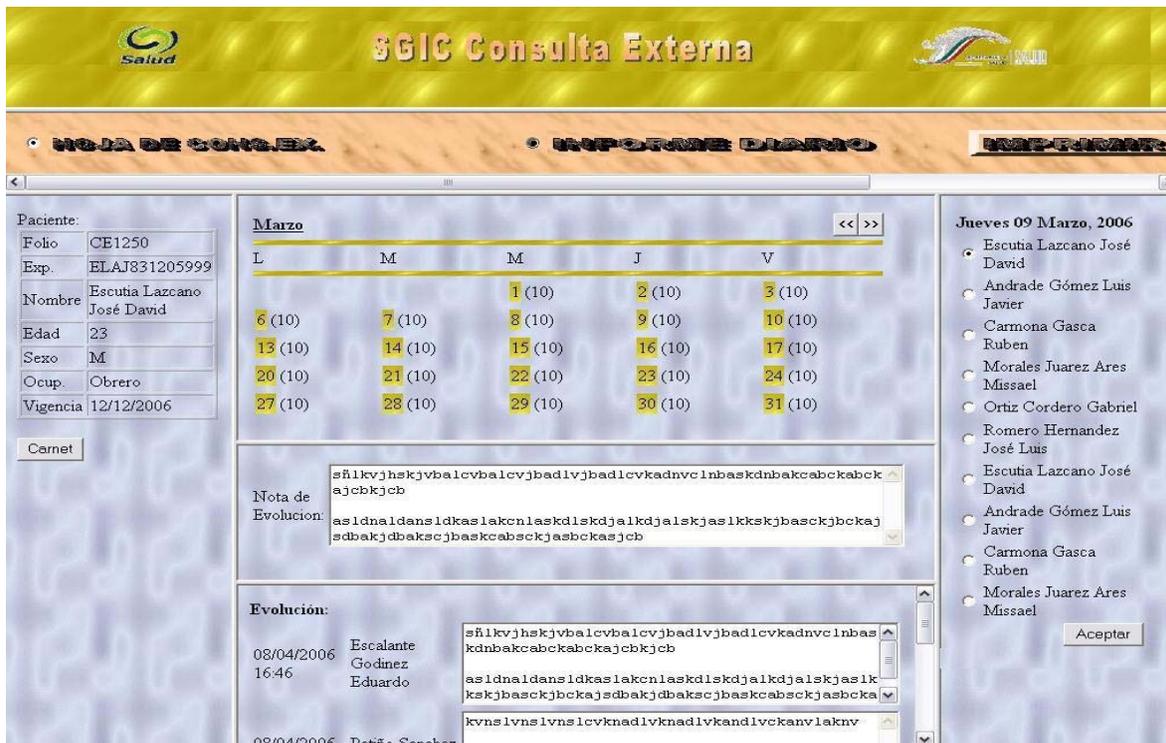


Figura 33. Agenda personal del médico de consulta externa

Como se observa cuenta con un calendario, la cual esta linkeada con la tabla que contiene su agenda personal. Al seleccionar un día al lado derecho se mostrara la lista de pacientes que tiene citados para ese día.

Carnet - Microsoft Internet Explorer - [Trabajar sin conexión]

Consultas:						
Área	Fecha	Hora	Cons.	Médico	Servicio	
CE	20/12/2000	10:30	3	Oropeza Reyes Mario	MR	
CE	25/02/2001	11:00	3	Oropeza Reyes Mario	MR	
CE	25/04/2001	11:00	3	Oropeza Reyes Mario	MR	
CE	25/06/2001	11:00	3	Reyes Alcantara Ruben	CH	

Terapias:						
Área	Fecha	Hora	Servicio	Terapista	Cubiculo	
TP	20/12/2000	10:30	TO	Ordoñez Ruiz Claudia	10	
TP	25/02/2001	11:00	TF	Reyes Cabello Araceli	2	
TP	25/04/2001	11:00	TF	Reyes Cabello Araceli	4	
TP	25/06/2001	11:00	TR	Campos Mendoza Omar	6	

Auxiliares de Diagnostico:			
Estudio	Fecha	Hora	Medico
RX	12/12/2002	12:00	Campos Ortega Sergio
TAC	12/12/2003	12:50	Labastida Billar Cecilia
RM	21/05/2003	12:00	Lavarez Lun Rocio
MN	30/02/2003	16:00	Martinez Sanchez Joel

Figura 34. Carnet del paciente

varios días al mismo lugar.

En el calendario con unas flechas de apoyo podrá moverse sin ningún problema por todos los meses. Al lado de cada digito señalando el día, esta la cantidad que tiene citados para esa fecha

Para ver al paciente en turno sólo debe seleccionarlo y sus datos aparecerán en pantalla. Únicamente deberá anotar sus nuevas observaciones. Si requiere de ver la notas anteriores de con respecto a su evolución lo podrá hacer en la parte central inferior.

Para comodidad del médico podrá dar un vistazo al carnet (fig. 34) del paciente para saber en que fechas podrá dar su próxima cita con respecto a las terapias o estudios del paciente, esto con el fin de dar comodidad al paciente para que en un día pueda realizar todas las citas pendientes sin tener que ir

y ver si les es conveniente citar más o no. algo muy conveniente para no conglomerar ese fecha de actividades y repartirlas equitativamente.

Si el paciente no se presenta al final del turno sólo será visto como cancelado. Con la misma facilidad que el médico de urgencias, cuando termine el turno, sólo tendrá que seleccionar Imprimir Informe y éste será impreso dos veces para que sea firmado por admisión y el médico tenga una copia con que ampararse si es necesario.

Por lo regular las citas están manejadas por Ventanilla o Control de Enfermeras, para facilitar esas actividades engorrosas y cantidades de papel enormes, van a contar con su propia pantalla de actividades (fig. 35):

**Consultas:**

Area	Fecha	Hora	Cons.	Médico	Servicio
CE	20/12/2000	10:30	3	Oropeza Reyes Mario	MR
CE	25/02/2001	11:00	3	Oropeza Reyes Mario	MR
CE	25/04/2001	11:00	3	Oropeza Reyes Mario	MR
CE	25/06/2001	11:00	3	Reyes Alcantara Ruben	CH
CE	25/11/2001	11:00	3	Crespi Muñoz Luis	O

**Terapias:**

Area	Fecha	Hora	Servicio	Terapeuta	Cubículo
TP	20/12/2000	10:30	TO	Ordoñez Ruiz Claudia	10
TP	25/02/2001	11:00	TF	Reyes Cabello Araceli	2
TP	25/04/2001	11:00	TF	Reyes Cabello Araceli	4
TP	25/06/2001	11:00	TR	Campos Mendoza Omar	6
TP	20/12/2000	10:30	TO	Ordoñez Ruiz Claudia	10

**Auxiliares de Diagnostico:**

Estudio	Fecha	Hora	Medico
RX	12/12/2002	12:00	Campos Ortega Sergio
TAC	12/12/2003	12:50	Labastida Billar Cecilia
RM	21/05/2003	12:00	Lavarez Lun Rocio
MN	30/02/2003	16:00	Martinez Sanchez Joel
US	23/08/2003	17:00	Cordova Rio Alejandro

Figura 35. Control de citas de pacientes

Cuando un paciente llegue con su carnet, la enfermera sólo ingresará el número de afiliado(expediente) y seleccionar que tipo va a ser, si es cita para terapia, consulta o auxiliar de diagnóstico.

Cada selección es diferente dependiendo del tipo de cita y claro su pantalla de captura será diferente.

Figura 36. Control de estudios

Figura 37. Control de terapias

EXP.:

Consulta

Area: CH

Fecha:

Hora:

Cons.:

Medico:

Servicio:

*Figura 38. Control de citas*

La primer imagen (fig. 36) corresponde al formulario de Auxiliar de Diagnóstico. Los que son estudios requeridos por el médico. La imagen superior (fig. 37) captura la información para citas de terapia y finalmente la imagen del lado izquierdo (fig. 38) recoge la información de las citas para consulta externa.

Esta pantallas aparecerán del lado izquierdo de la pantalla principal dejando el lado derecho para ver la anotaciones nuevas y que hayan sido actualizadas.

Para finalizar con el área de citas, se anexo un botón para imprimir directamente las hojas respectivas a cada cita y ser entregadas a

las secciones correspondientes.

Para mejorar la seguridad todos los usuarios se encontraran restringidos con respeto a sus funciones, esto es, sólo los de farmacia podrá surtir recetas, los médicos son los únicos que pueden realizar consultas médicas y los de trabajo social y admisión hospitalaria exclusivamente pueden ingresar un nuevo paciente y clasificarlo.

Estas son las funciones dentro de la Intranet y únicamente pueden funcionar dentro de ella. Como último punto las tablas que no son usadas por las pantallas aquí mostradas, pertenecen a áreas que no manejan datos de manera constante, por tal hecho, estas son llenadas por medio del manejador de bases de datos a cargo del ingeniero en turno, por medio de la administración de esa unidad a cargo.

## ***CONCLUSIÓN***

Ante la alta demanda de sistemas computacionales que cumplen con las tareas que exige el movimiento social–tecnológico que actualmente existe en estos últimos años, y que, va evolucionando día con día, es necesario que estos sistemas vayan más allá de sus simple aplicación, también deben permitir su crecimiento paralelo con el organismo por el que fue ideado e implementado, dejando ver claramente su desarrollo y expansión a varios ámbitos.

El proyecto SGIC (Sistema Gestor de Información y Control) ha propuesto una alternativa para los problemas que existen actualmente en el sector de la salud, específicamente en las unidades médicas, al convertirse en una herramienta que agiliza notablemente los procesos internos manteniendo íntegra y oportuna la información al momento de ser requerida, porque elimina actividades repetitivas ahorrando espacio donde se requiera logrando de esta manera optimizar considerablemente la calidad en los servicios del sector salud.

SGIC al ser implantado en las unidades médicas, cumplirá con las ventajas antes descritas, ya que es un programa que tiene como factores la flexibilidad, pues se adapta a las necesidades de cada área y permite su modificación; es seguro porque evalúa los datos introducidos por parte del personal de la unidad médica, evitando accesos mal intencionados o cadenas de texto dañino; es conciso porque solo puede utilizarlo dentro de la misma área, y este solo puede solicitar estos datos para generar un buen diagnóstico; mantiene la información oportuna permitiendo ser consultada en cualquier momento que se necesite y es confidencial porque mantiene jerarquías dentro de los niveles de acceso originando bitácoras que describen los accesos que cualquier personal haya tenido al sistema, siendo esto independiente dependiendo del área en la que se labore.

Por lo tanto, el SGIC es un programa robusto porque cumple con las características para ser un sistema competitivo y eficiente todo esto representa un esfuerzo para cumplir con las pretensiones de un sustenta especializado de información, destacando nuevas aplicaciones que la hacen diferente a otros sistemas.

Pero así como se necesita de estos sistemas, también se requieren de Ingenieros en Computación capaces de brindar nuevas soluciones a los problemas en la actualidad, deben estar preparados y estar a la vanguardia que continuamente van cambiando y exigiendo más aptitudes en ellos. El Diplomado de Software Libre abre un nuevo panorama a una nueva filosofía que con el paso del tiempo ha tenido más auge en los programadores del mundo. Actualmente el Software Libre, ya sea, Open Source y Free Source ha remontado en la carrera computacional dando nuevas opciones a las industrias y ofreciendo Software de alta calidad.

Por esta razón se puede considerar al Ingeniero en Computación como una pieza fundamental para poder hacer más sólida la relación entre el hombre y la sociedad tecnológica que evoluciona aceleradamente en el extenso mundo de los sistemas computacionales.

## ***BIBLIOGRAFÍA***

[www.php.net](http://www.php.net)

[www.linuxiso.org](http://www.linuxiso.org)

[www.w3school.com](http://www.w3school.com)

[www.wikipedia.com](http://www.wikipedia.com)

[www.linuxpowered.org](http://www.linuxpowered.org)

FACUNDO, Héctor Arena  
Linux a fondo\*, 1ra edición  
MP Ediciones  
Buenos Aires, Argentina

DE LUIS, Mario  
Programación con Java\*  
Prensa Técnica, 1ra edición  
Madrid, España

GARCIA, Ramón Pelayo  
Diccionario Enciclopédico Larousse\*  
Ediciones Larousse, 3ra edición  
D.F., México