UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

"IMPLEMENTACIÓN DE UN SERVIDOR CON SISTEMA OPERATIVO LINUX EN BASE A LA DISTRIBUCIÓN SLACKWARE VERSIÓN 10 EN LA EMPRESA BRAINUP SYSTEMS S.A. DE C.V."

Trabajo de titulación bajo la modalidad de "Seminarios y cursos de actualización y capacitación profesional"

Para obtener el grado de:

"Ingeniero en Computación"

Presenta:

Gerardo Muñoz Medrano

Asesor

Ing. Enrique García Guzmán

San Juan de Aragón, México 2007

Agradecimientos

A mi madre ***- Por apoyarme en todos los aspectos de la vida, especialmente en mis estudios.***

A Felipe ***- Por su amistad y apoyo todos estos años.***

A Chata ***- Por estar a mi lado en los momentos difíciles.***

A Chayo ***- Por darme su apoyo siempre.***

A Jesy ***- Por estar a mi lado estos cuatro años.***

A mi asesor ***- Por ayudarme a realizar este trabajo, brindarme su confianza y su apoyo en este proyecto.***

A mi familia ***- Por estar conmigo y contar siempre con ellos.***

A todos muchas gracias

CONTENIDO

Objetivo general

Implementar en una empresa, a través de software libre, sistemas para el control de procesos e información, que funcionen de forma natural en red o por Internet a través de herramientas que han demostrado tener una alta confiabilidad, alto desempeño y funcionalidad.

Objetivo especifico

- Instalar el sistema Operativo Linux distribución Slackware en el servidor de la empresa Brainup Systems.

- Elaborar una página WWW para uso local, a través del lenguaje *HTML*, así como del lenguaje de desarrollo para servidor *PHP*, e interacción con base de datos empleando herramientas de software libre.

- Instalar, configurar y administrar un servidor WWW en base al software de libre distribución "*Apache*" válido para servir páginas Web en la Intranet, para probar el desarrollo de aplicaciones de lado del servidor y para acceder a ficheros desde un PC remoto.

- Administrar la seguridad con el fin de proteger el sistema y la información de la empresa contenida en el servidor que se implementará.

Justificación

En la actualidad es necesario contar con sistemas que permitan la automatización del seguimiento y control de procesos que se desarrollan en diversas empresas y oficinas. Con esta automatización se tiene un uso más eficiente de los recursos y un control más preciso de los mismos, lográndose con ello compartir información entre diversas áreas y oficinas, a través de la red local e incluso empleando Internet o la intranet corporativa.

El desarrollo de estos sistemas puede ser sustentado empleando software libre como lo es Linux. PHP, Apache, MySQL, PosgreSQL, que son herramientas que han demostrado tener un alto desempeño, gran estabilidad y seguridad, y por el hecho de ser libres, permiten reducir los costos que se generan en las licencias de uso de software, logrando aprovechar al máximo, los recursos de cómputo con que se cuente.

Introducción

El término "Software Libre", nació en los EE.UU., con objeto de evitar la ambigüedad del doble significado (libre y gratis) de la palabra inglesa *free.* En cualquier caso estamos hablando de programas que se distribuyen bajo una licencia que cumple con las "cuatro libertades" establecidas por Richard Stallman en su definición original de Sofwtare Libre:

La libertad de usar el programa, con cualquier propósito (libertad 0). La libertad de estudiar cómo funciona el programa y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para ella.

La libertad de distribuir copias (libertad 2).

La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie (libertad 3). El acceso al código fuente es un requisito previo para ella.

Por tanto el Software Libre se define en función de lo que los usuarios pueden hacer cuando reciben una copia de un programa y no de cómo se ha desarrollado éste, ni por quién, ni con qué intenciones. Sin embargo y aunque en la definición no se dice nada acerca de cómo

ha de producirse o comercializarse este software para que sea 'libre', esas cuatro libertades hacen posibles algunos modelos de desarrollo y de negocio, y dificultan o impiden otros.

Ésta es la razón de que sea común hablar acerca de "modelos de desarrollo de Software Libre" o de "modelos de negocio de Software Libre". Ambos términos no deben entenderse como "modelos que hay seguir para que un programa sea considerado Software Libre" sino sólo como modelos posibles, y quizás habituales, en el mundo del Software Libre.

El Software Libre (o de Código Abierto)[1] ha evolucionado durante la última década, pasando de ser un fenómeno oscuro y marginal a convertirse en un conjunto de aplicaciones relativamente bien conocido, ampliamente disponible y extensamente utilizado. Hay soluciones de Software Libre que son incluso líderes en algunos segmentos del mercado y están experimentando un enorme crecimiento en otros. Productos como OpenOffice, Linux, Apache, Firefox y muchos otros son parte de las experiencias cotidianas de millones de usuarios. Tanto las empresas como las Administraciones Públicas están teniendo en cuenta cada vez más los beneficios que el Software Libre puede proporcionar cuando se usa de forma extendida.

Sin embargo, y a pesar de su creciente popularidad, persiste una comprensión muy pobre del Software Libre. Quizás por causa de esta paradoja, durante los últimos años los investigadores han empezado a fijar su interés en el Software Libre en sí mismo: sus modelos de desarrollo, el modelo de negocio que lo circunda, la motivación de los desarrolladores, etc.

Introducción a GNU/Linux

EL origen de Linux se remonta al mes de agosto de 1991, cuando un estudiante finlandés llamado Linus Torvalds anunció en una lista de que había creado su propio núcleo de sistema operativo y lo ofrecía a la comunidad de desarrolladores para que lo probara y sugiriera mejoras para hacerlo más utilizable. Éste sería el origen del núcleo (o *kernel*) del operativo que más tarde se llamaría Linux.

Por otra parte, la FSF (Free Software Foundation), mediante su proyecto GNU, producía software (desde 1984) que podía ser utilizado libremente. Debido a lo que Richard Stallman (miembro de la FSF) consideraba software libre, es decir, como aquél del que podíamos conseguir sus fuentes (código), estudiarlas y modificarlas, y redistribuirlo sin que nos obliguen a pagar por ello. En este modelo, el negocio no está en la ocultación del código, sino en el software complementario añadido, en la adecuación del software a los clientes y en los servicios añadidos, como el mantenimiento y la formación de usuarios (el soporte que les demos), ya sea en forma de material, libros y manuales, o en cursos de formación.

La combinación (o suma) del software GNU y del *kernel* Linux, es el que nos ha traído a los actuales sistemas GNU/Linux. Actualmente, los movimientos Open Source, desde diferentes organizaciones (como FSF) y empresas como las que generan las diferentes distribuciones Linux (Red Hat, Mandrake, SuSe, etc.), pasando por grandes empresas como HP, IBM o Sun que proporcionan apoyo, han dado un empujón muy grande a los sistemas GNU/Linux hasta situarlos al nivel de poder competir, y superar, muchas de las soluciones propietarias cerradas existentes.

¿Que es Linux?

Lo que realmente se entiende bajo el término Linux es el Kernel, el núcleo del sistema operativo.

Pero el kernel por sí solo no forma todavía ningún sistema operativo. Para Linux existe una gran cantidad de software libre. Es el conjunto de todo esto (kernel y aplicaciones[2]) lo que realmente forma un sistema operativo.

[1] Para designarlo se está popularizando el uso de la sigla FLOSS (*Free / Libre / Open Source Software*).

[2] Compilador de C y C++ (*GCC*), shell *bash*, editor Emacs (GNU Emacs), intérprete *postscript* (*ghostscript*), biblioteca C estándar (*GNU C library*, o también *glibc*), depurador (*GNU gdb*) *Makefile* (*GNU make*), el ensamblador (*GNU assembler* o *gas*), el *linker* (*GNU linker* o *gld*).

En cuanto a las utilidades se trata generalmente de la versión GNU de los programas correspondientes de Unix, que la mayoría de las veces ofrecen mayor funcionalidad.

Para Unix existe una cantidad enorme de software libre, lo que permite a su vez componer una multitud de sistemas Linux. En este punto aparecen las distribuciones (SuSE, RedHat, Slackware, Debian, OpenLinux, Mandrake...), las cuales contemplan la enorme oferta de software libre y eligen los programas más adecuados para distribuirlos en forma de CD o a través de Internet, así que no hace falta comprar una distribución para tener un sistema GNU/Linux en nuestros servidores.

Razones para utilizar GNU/Linux

Una de las primeras razones que se pueden esgrimir para utilizar Linux, ya la hemos comentado: no son necesarias licencias. Aunque Linus Torvalds mantiene la marca registrada Linux, el Kernel de Linux y la mayoría del software que le acompaña se distribuye bajo licencia GPL. Esto significa que se puede modificar el código fuente y vender los programas resultantes, pero los autores originales mantienen el copyright y el usuario debe proporcionar el código fuente de los cambios.

Linux se ejecuta en más CPU's y plataformas diferentes que cualquier otro sistema operativo, ya que Linux viene con el código fuente del Kernel y es fácilmente portable. Linux representa una ventaja real, especialmente al comparar el costo de otros sistemas operativos. Además suele ser inmune a la cantidad de virus que afectan a otros sistemas operativos.

¿Por qué Linux Slackware?

Slackware, iniciado por Patrick Volkerding a fines de 1992, y liberado para el uso de todo el mundo el 17 de julio de 1993, fue la primera distribución de Linux en alcanzar un uso masivo. Esta distribución ganó popularidad rápidamente, así que Volkerding decidió llamarla Slackware y hacerla disponible públicamente.

Hay muchas razones por las cuales Slackware es la distribución vigente más antigua de Linux. No trata de emular Windows, trata de ser lo más parecida a Unix como sea posible. No trata de cubrir los procesos con interfaces gráficas lindas. En lugar de eso, pone el control en manos de los usuarios, dejándolos ver exactamente lo que están haciendo. Su desarrollo no está presionado por plazos.

Slackware disfruta en la actualidad de la fama de ser tanto un servidor sólido como una estación de trabajo coherente. Los servidores basados en Slackware dan poder a negocios, actuando en cada característica que se exige de un servidor.

Slackware es una distribución capaz de satisfacer las necesidades básicas que se buscan en un sistema operativo, ya sea de escritorio, o como un servidor. Slackware esta basado en un sistema de ficheros de inicio muy simple (BSD-like).

CAPÍTULO I

INSTALACIÓN DEL SISTEMA OPERATIVO

CAPÍTULO I INSTALACIÓN DEL SISTEMA OPERATIVO

1.1 Introducción

La Empresa Brainup Systems, esta dedicada al desarrollo de sistemas informáticos, y a la implementación de aplicaciones. A consecuencia del crecimiento que ha tenido en los últimos años, esta firma se esta viendo en la necesidad de incrementar sus recursos tecnológicos, en especial en su centro de cómputo, ya que éste se ve involucrado de manera importante en el plan de desarrollo de proyectos de la empresa.

Ante esta problemática, la empresa se ve en la necesidad de implementar una herramienta más para optimizar sus procesos. Que de acuerdo con las necesidades de los usuarios, se basa en un servidor que solvente las necesidades de conectividad, así como de herramientas donde los usuarios puedan compartir sus archivos, y una herramienta que funcione a través del Web, para publicar información de manera interna, para incrementar de esta manera la eficacia en sus procesos.

1.2 Especificaciones del Servidor de la empresa

Para instalar Linux, primero es conveniente recoger toda la información referente al hardware del servidor en donde se hará la instalación del sistema.

El servidor destinado para esta implementación es un componente de la marca DELL modelo **PowerEdge™ SC1425**

- Sistema -Procesador Intel® Xeon® 2.80GHz/2MB Cache 800MHz FSB (145282L)
- Memoria - 1GB DDR2 400MHz (2X512MB) "Single Ranked DIMMs" (1G2D4S)
- Disco Duro - 1 Disco Duro de 80GB SATA 1" (7200rpm) (80GS)
- Dispositivo óptico- Dispositivo Óptico CDRW/DVD, 24X (24CDDVD)
- Tarjeta de red -Adaptador de Red Gigabit de Cobre Intel Pro 1000MT 10/100/1000 (1000MSP)

Y con ello constatamos que el servidor es compatible, ya que los requisitos mínimos para la instalación son: Procesador AMD, Pentium o compatible, 32 de RAM para instalación en modo texto. 64 recomendados, más de 2 Gigas de disco *(Aplicable si se dispone de varios CDS),* CD-ROM, Tarjeta de video, Tarjeta de sonido[1].

1.3 Instalación del sistema operativo

Aunque existan varias distribuciones de Linux, el método de instalación en todos ellos, es similar, y se podría resumir en los siguientes pasos: 1.Elegir el método de instalación. 2. Arrancar la Instalación. 3. Crear las particiones en el disco duro. 4. Elegir e instalar los paquetes necesarios. Y 5. Instalar el gestor de arranque.

[1] Puede variar.

1.3.1 Métodos de instalación.

Aunque Linux Slackware proporciona varias maneras distintas para instalar el software. Por ejemplo desde una partición MS-DOS en el disco duro; o desde disquetes MS-DOS, aquí se usará la **instalación a través de CD ROM**, debido a que este método de instalación es el más sencillo ya que los paquetes a instalar se cargarán directamente desde el CD ROM al disco duro.

1.3.2 Arranque de la instalación

Generalmente, las computadoras pueden iniciar desde diferentes soportes de hardware, como son de disco duro, de disco flexible, de CD ROM, o de un dispositivo extraíble, generalmente vienen configuradas de fabrica, para arrancar en este orden, por lo que al insertar el CD ROM de instalación de Slackware en el ordenador y encender el sistema, debería arrancar la instalación, si no es el caso deberemos configurar la BIOS[2], para poder arrancar directamente desde este soporte. El detalle de la instalación paso a paso del sistema esta presentado en el Anexo I.

1.3.3 Particionamiento del disco duro

La forma en que Linux maneja y accede a las particiones es que en cada partición se integra en el sistema de almacenamiento necesario para formar parte de un sólo juego de archivos y directorios. Esto se consigue asociando una partición con un directorio mediante un proceso conocido como montaje. Montar una partición significa disponer de su capacidad de almacenamiento comenzando en el directorio especificado (conocido como punto de montaje).

El número y el tamaño de las particiones que utilizará el sistema operativo de la empresa son las siguientes: Se creará una partición de intercambio (*swap*). Las particiones de intercambio se usan como apoyo a la memoria virtual. El tamaño mínimo debería ser igual a la RAM presente en el ordenador, así que para esta instalación, se asignaran 1Gb de memoria. Se creará también una partición /boot. La partición montada en /boot contiene el kernel del sistema operativo, así como los archivos usados durante el arranque. A esta partición le asignaremos 2 Gb.

La partición raíz (root) es donde reside / (el directorio raíz). En este pérfil de particiones, todos los archivos (excepto los alojados en /boot) se encuentran en la partición raíz. Por ello, interesa maximizar el tamaño de la partición raíz. Por lo cual utilizaremos el espacio restante del disco duró para esta partición, es decir 73 Gb.

En el primer sector del disco esta el registro de arranque maestro junto a la tabla de particiones. El registro de arranque (como su nombre lo indica) se usa para arrancar el sistema. La tabla de particiones contiene información acerca del lugar y el tamaño de cada partición. Hay tres clases de particiones: primarias, extendidas, y lógicas. De estas, las más usadas son las primarias. Sin embargo, debido al límite del tamaño de la tabla de particiones, sólo pueden tenerse hasta cuatro particiones primarias en un disco.

La forma de superar este límite de cuatro particiones es usar particiones extendidas. Una partición extendida no tiene datos ella misma; en su lugar, actúa como "soporte" de particiones lógicas.

Por lo tanto, se puede crear una partición extendida que ocupe todo el disco, y dentro crear cualquier número de particiones lógicas. Sin embargo, sólo puede tenerse una partición extendida por disco.

[2] Basic Input-Output System (BIOS) es un código de interfaz que localiza y carga el sistema operativo en la RAM.

1.3.4 Elección de paquetes necesarios

Después de configurar las particiones y seleccionarlas para formatearlas, se está en disposición de seleccionar los paquetes para su instalación. Se pueden seleccionar componentes, que agrupan paquetes por su función, paquetes individuales, o una combinación de ambos.

Los componentes agrupan paquetes según la funcionalidad que proporcionan. Por ejemplo, Desarrollo C [C Development], Estación de Trabajo en Red [Networked Workstation], o Servidor Web [Web Server].

Muchos de los paquetes software, para trabajar correctamente, dependerán de otros paquetes software, o librerías que deben ser instaladas en el sistema.

En este punto, el programa de instalación formateará todas las particiones que asignamos, y posteriormente el programa de instalación empieza a instalar paquetes.

1.3.5 Elección del gestor de arranque

Una de las características más importantes de Linux es el método altamente configurable que se utiliza para el inicio del sistema operativo. El administrador es libre de configurar muchos aspectos del proceso de arranque, incluyendo qué programas se lanzarán en el momento del arranque. De forma parecida, la parada del sistema finaliza los procesos de forma organizada y configurable, aunque la personalización de este proceso casi nunca es necesaria. Entender el funcionamiento del proceso de arranque y parada no sólo le permite personalizarlo, sino que también facilita resolver problemas relacionados con el inicio y el cierre del sistema.

Cuando un ordenador arranca, el procesador busca al final de la memoria del sistema el programa de la *BIOS* y lo ejecuta. La *BIOS* controla no sólo el primer paso del proceso de arranque, sino que también proporciona una interfaz de bajo nivel para dispositivos periféricos.

Una vez que se haya cargado, la BIOS verifica los periféricos y localiza un dispositivo que permita arrancar el sistema. Habitualmente, en primer lugar comprueba cualquier disquete y unidades de CD-ROM presente por los medios de arranque, y a continuación si esto falla, revisa las unidades de disco duro del sistema. El orden de las unidades necesario para arrancar puede ser controlado por la BIOS. La BIOS carga en memoria cualquier programa que resida en el primer sector de este dispositivo, llamado **Master Boot Record** o *MBR*. El MBR sólo tiene 512 bytes de tamaño y contiene las instrucciones de código de máquina para el arranque del equipo, llamado gestor de arranque así como también la tabla de particiones. Una vez que la BIOS haya encontrado y cargado el gestor de arranque en memoria, le deja el control del proceso de arranque a éste.

Los gestores de arranque de Linux para la plataforma x86 se dividen en dos etapas. La primera es un pequeño código binario que se encuentra en el MBR. Su única función es la de localizar el gestor de arranque de la segunda etapa y cargar la primera parte de éste en memoria.

Para poder arrancar el sistema sin la necesidad de un disquete de inicio, normalmente se utiliza un cargador de sistemas operativos. Este cargador es un software que se ejecuta cuando la máquina arranca y es el responsable de cargar y transferir el control al kernel[3]. El kernel a su vez, inicializa el resto del sistema operativo. El proceso de instalación de Slackaware, proporciona el cargador de arranque LILO[4].

[3] El núcleo o kernel es el corazón del sistema Linux, un sistema muy básico se compone del kernel y algún interprete de comandos.

[4] LInux LOader. es también un cargador para Linux muy eficaz. No depende de un sistema de ficheros específico y puede arrancar/cargar imágenes del kernel Linux desde disquete o disco duro, así como otros sistemas operativos.

De manera general, este es el procedimiento para la instalación de cualquier sistema Linux. Y en el caso de la implementación en el servidor de la empresa se podrá observar a detalle en el Anexo I.

CAPÍTULO II
ADMINISTRACIÓN BÁSICA DEL SISTEMA OPERATIVO

CAPÍTULO II ADMINISTRACIÓN BÁSICA DEL SISTEMA OPERATIVO

2.1 El administrador de sistemas

Las grandes empresas y organizaciones dependen cada vez más de sus recursos de computación y de cómo éstos son administrados para adecuarlos a las tareas. El gran incremento de las redes distribuidas, con sus equipos servidores y clientes, ha creado una gran demanda de un nuevo pérfil laboral: el llamado *administrador de sistemas*.

El administrador es la persona encargada de que las tecnologías utilizadas por los usuarios funcionen adecuadamente, o sea, que los sistemas cumplan las perspectivas de los usuarios, así como las tareas que éstos quieran realizar.

Debido a la gran cantidad de conocimientos, no es extraño que aparezcan a su vez diferentes subperfiles de la tarea del administrador.

En una gran organización puede ser habitual encontrar a los administradores de sistemas operativos (UNIX, Mac, o Windows), que suelen ser diferentes: administrador de bases de datos, administrador de copias de seguridad, administradores de seguridad informática, administradores encargados de atención a los usuarios, etc.

En una organización más pequeña como lo es Brainup Systems, varias o todas las tareas pueden estar asignadas a uno o pocos administradores. Los administradores de sistemas UNIX (o de GNU/Linux) serían una parte de estos administradores (cuando no el administrador que tendrá que hacer todas las tareas). Normalmente, su plataforma de trabajo es UNIX (o GNU/Linux en nuestro caso), y requiere de bastantes elementos específicos que hacen este trabajo único. UNIX (y variantes) es un sistema operativo abierto y muy potente, y, como cualquier sistema software, requiere de cierto nivel de adecuación, configuración y mantenimiento en las tareas para las que vaya a ser usado. Configurar y mantener un sistema operativo es una tarea seria, y en el caso de LINUX puede llegar a ser bastante frustrante.

Algunas áreas importantes por tratar son:

a) Que el sistema sea muy potente también indica que habrá bastantes posibilidades de adaptarlo (configurarlo) a las tareas que queremos hacer. Habrá que evaluar las posibilidades que se nos ofrecen y cuán adecuadas son para nuestro objetivo final.
b) Un sistema abierto y ejemplo claro de ello es nuestro GNU/Linux, que nos ofrecerá actualizaciones permanentes, ya sea en la corrección de errores del sistema, como en la incorporación de nuevas prestaciones. Y, evidentemente, todo esto tiene unos impactos directos importantes en costes de mantenimiento de las tareas de administración.
c) Los sistemas se pueden utilizar para tareas de coste crítico, o en puntos críticos de la organización, donde no se pueden permitir fallas importantes, que hagan lenta o paren la marcha de la organización.
d) Las redes son actualmente un punto muy importante (si no el que más), pero también es un área de problemas potenciales muy crítica, tanto por su propia naturaleza distribuida como por la complejidad del sistema para encontrar, depurar y solucionar los problemas que se puedan presentar.
e) En el caso particular de los sistemas UNIX, y en nuestros GNU/Linux, la abundancia, tanto de versiones como de distribuciones diferentes del sistema, incorpora problemas adicionales a la administración, ya que es necesario conocer las problemáticas y diferencias de cada versión y distribución.

2.2 Tareas de administración local del sistema

Las principales tareas que un administrador de sistemas, en este caso el administrador del servidor con sistema operativo LINUX Slackware son las siguientes:

2.2.1 Arranque y apagado del sistema

Cualquier sistema basado en UNIX tiene unos sistemas de arranque y apagado valorables, de manera que podemos configurar qué servicios ofrecemos en el arranque de la máquina y cuándo hay que pararlos, o programar el apagado del sistema para su mantenimiento.

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

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

El nivel de inicio por default se encuentra definido en el archivo /etc/inittab, para ello necesitamos conocer los diferentes niveles que están predefinidos en el archivo de inicio:
Cabe mencionar que dependiendo del nivel de inicio la rutina de encendido apagado puede variar.

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

Para apagar el sistema podemos usar varios comandos, dichos comandos se introducen en una terminal o consola:

Poweroff

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

Halt

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

Shutdown

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

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

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

Reboot

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

Para saber en que nivel se esta ejecutando el sistema se utiliza el comando **runlevel**:

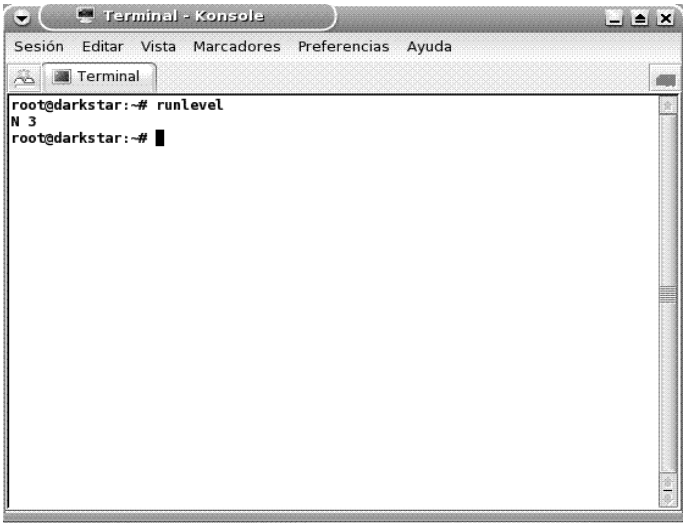


FIGURA 2.1 Niveles de ejecución.

Cabe recalcar que la gran mayoría de las tareas administrativas tienen que realizarse en modo monousuario o de mantenimiento para evitar que se inicien los dispositivos de red y los servicios del sistema, de esta manera se evita que cualquier usuario se conecte vía remota impidiendo que se realicen las tareas administrativas eficientemente.

Init

Si deseamos cambiar el modo de ejecución del sistema podemos hacerlo directamente con el comando init y el número de inicio, por ejemplo si deseamos cambiar a nivel de ejecución 1.

El sistema ejecutará los scripts correspondientes para cerrar los demonios, los dispositivos de red y quedará en modo monousuario.

2.2.2 Gestión de usuarios y grupos

Dar cabida a los usuarios es una de las principales tareas de cualquier administrador. Habrá que decidir qué usuarios podrán acceder al sistema, de qué forma y bajo qué permisos; y establecer comunidades mediante los grupos.
En el caso particular de Brainup Systems, se crearán 45 cuentas de usuario, que son la cantidad de empleados de la empresa.

Comandos para administrar usuarios:

Comandos para administrar los usuarios:

Adduser Agrega usuarios modo comando.
Useradd Agrega usuarios modo comando.

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

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

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

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

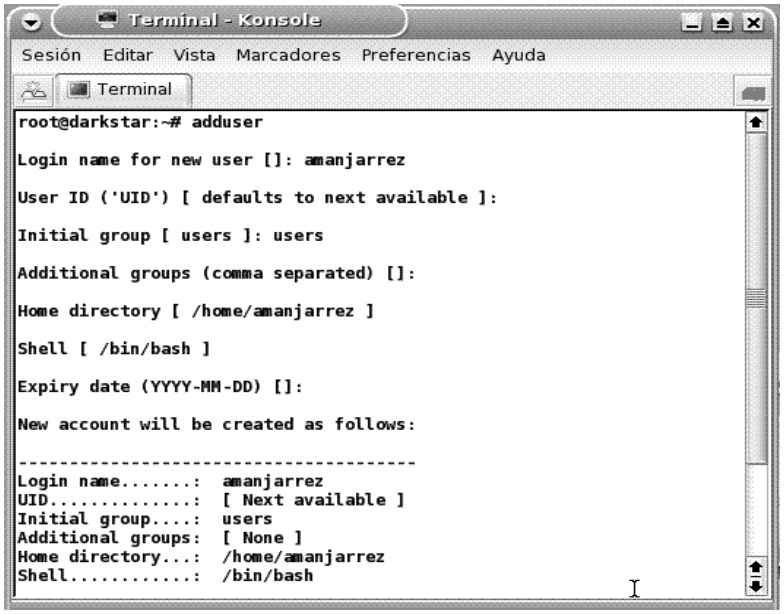


FIGURA 2.2 Alta de un nuevo usuario.

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

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

Grupos

Para agregar un grupo nuevo se usa el comando groupadd, todos los registros de los grupos se encuentran en el archivo /etc/group.

Algunos sistemas permiten agregar un grupo privado para cada usuario, este no es el caso de Slackware, ya que tiene un grupo llamado users al cual pertenecen todos los usuarios por default, aunque es posible modificar las propiedades para adoptarlo a las necesidades administrativas y funcionales del sistema.

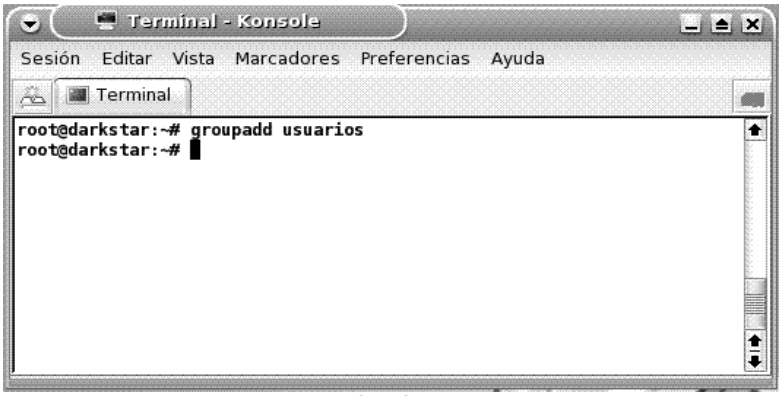


FIGURA 2.3 Alta de un nuevo grupo.

2.2.3 Gestión de recursos del sistema

Consiste básicamente en qué ofrecemos, cómo lo ofrecemos a quién damos acceso.

Uso de recursos:

du (Disk Usage)

Permite monitorear el estado del sistema de discos:

Una buena planeación para la implementación de un servidor permite prevenir ciertos inconvenientes al separar los espacios de almacenamiento para ciertas prioridades, de tal forma que si se satura un área de almacenamiento no se pierda el control de acceso al sistema para que los usuarios puedan depurar sus cuentas; una buena opción es configurar cuotas de disco.

df

Muestra el espacio ocupado por directorios entregando un total de espacio ocupado, este comando se puede aplicar sobre los directorios de cada usuario y saber rápidamente cuanto espacio esta usando cada usuario en total.

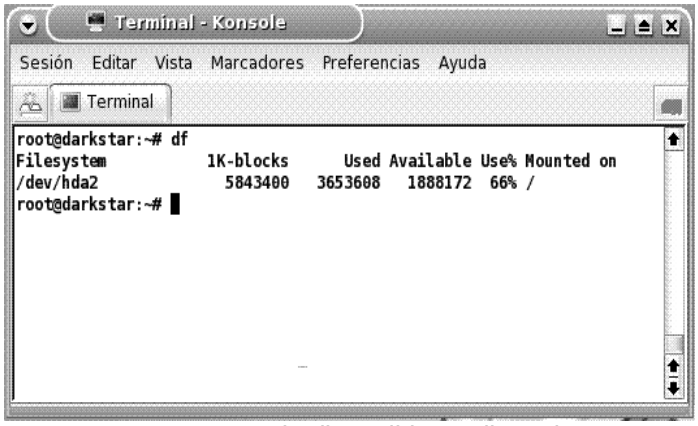


FIGURA 2.4 Espacio disponible en disco duro.

2.2.4 Gestión de los sistemas de archivos

El ordenador puede disponer de diferentes recursos de almacenamiento de datos y dispositivos (disquetes, discos duros, ópticos, etc.) con diferentes sistemas de acceso a los ficheros. Pueden ser permanentes o extraíbles o temporales, con lo cual habrá que modelar y gestionar los procesos de montaje y desmontaje de los sistemas de ficheros que ofrezcan los discos o dispositivos afines.

En Linux hay varios tipos de sistema de archivos, cada uno con características variadas las cuales determinan la funcionalidad del sistema, algunas ofrecen ventajas sobre otras teniendo desde el estándar ext2 hasta sistemas mejorados como Reiser o journal.

El tipo de sistema de archivos lo podemos elegir en el momento de formatear las particiones determinando el tipo de sistema de archivos que deseamos que tenga dicha partición.

2.2.5 Cuotas del sistema

Cualquier recurso que vaya a ser compartido tiene que ser administrado, y según la cantidad de usuarios, habrá que establecer un sistema de cuotas para evitar el abuso de los recursos por parte de los usuarios o establecer clases (o grupos) de usuarios diferenciados por mayor o menor uso de recursos. Suelen ser habituales sistemas de cuotas de espacio de disco, o de impresión, o de uso de CPU (tiempo de cómputo usado).

CAPÍTULO III
CONFIGURACIÓN Y ADMINISTRACIÓN DEL SERVIDOR DE RED DE LA EMPRESA

CAPÍTULO III CONFIGURACIÓN Y ADMINISTRACIÓN DEL SERVIDOR DE RED DE LA EMPRESA

3.1 Introducción

Una de las tareas más tediosas del administrador de redes, es la de recorrer los puestos de trabajo para configurar números de IP ya sea por la llegada de un equipo nuevo, o por modificaciones en la estructura de la red. Para facilitar este proceso se configuran los servicios de asignación dinámica de IPs.

El uso de IPs estáticas en redes bajo la tutela conlleva siempre inconvenientes de mantenimiento sin importar las dimensiones de la misma; puede tratarse de nuestra pequeña red doméstica, un cibercafé, oficina o una universidad. Los nuevos nodos que se conecten a la red o los cambios vitales como servidores de nombres, pasarela (gateway), servidores WINS, etc. obligan al administrador de cuando en cuando a dejar su puesto y recorrer los cubículos para realizar estos cambios.

Dado que cuanto más grande es la red es más difícil mantenerla de este modo, el grupo de trabajo de la Internet Engineering Task Force resolvió el problema desarrollando un Protocolo para Configuración Dinámica de Terminales, conocido como DHCP. Este protocolo, basado en otro llamado BOOTP (de Boot Protocol) pero con mejores características, está descrito en el RFC 2131 y en anteriores como el 1541 y el 1531, ya obsoletos por la publicación del primero. Imaginando que los interesados en esta nota serán administradores de redes haciendo sus primeras armas.

3.2 DHCP

DHCP (sigla en inglés de **D**ynamic **Host C**onfiguration **P**rotocol) es un protocolo de red que permite a los nodos de una red IP obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme estas van estando libres, sabiendo en todo momento quien ha estado en posesión de esa IP, cuanto tiempo la ha tenido, a quien se la ha asignado después.

3.2.1 Características

Provee los parámetros de configuración a las computadoras conectadas a la red informática que lo requieran (máscara, puerta de enlace y otros) y también incluyen mecanismo de asignación de direcciones de IP.

Este protocolo apareció como un protocolo estándar en octubre de 1993. En RFC 2131 (Inglés) (http://www.faqs.org/rfcs/rfc2131.html) se puede encontrar la definición más actualizada. Los últimos esfuerzos describiendo DHCPv6, DHCP en una red IPv6, fue publicado como RFC 3315 (Inglés) (http://www.faqs.org/rfcs/rfcs3315.html).

3.2.2 Asignación de direcciones IP

Sin DHCP, cada dirección IP debe configurarse manualmente en cada ordenador y, si el ordenador se mueve a otro lugar en otra parte de la red, se debe de configurar otra dirección IP diferente. El DHCP le permite al administrador supervisar y distribuir de forma centralizada las direcciones IP necesarias y, automáticamente, asignar y enviar una nueva IP si el ordenador es conectado en un lugar diferente de la red.

El protocolo DHCP incluye tres métodos de asignación de direcciones IP:

- **Asignación manual**: donde la asignación se basa en una tabla con direcciones MAC (pares de direcciones IP ingresados manualmente por el administrador). Sólo las computadoras con una dirección MAC que figure en dicha tabla recibirá el IP que le asigna dicha tabla.

- **Asignación automática**: donde una dirección de IP libre obtenida de un rango determinado por el administrador se le asigna permanentemente a la computadora que la requiere.
- **Asignación dinámica**: el único método que permite la reutilización dinámica de las direcciones IP. El administrador de la red determina un rango de direcciones IP y cada computadora conectada a la red está configurada para solicitar su dirección IP al servidor cuando la tarjeta de interfaz de red se inicializa. El procedimiento usa un concepto muy simple en un intervalo de tiempo controlable. Esto facilita la instalación de nuevas máquinas clientes a la red.

Algunas implementaciones de DHCP pueden actualizar el DNS asociado con los servidores para reflejar las nuevas direcciones IP mediante el protocolo de actualización de DNS establecido en RFC 2136 (Inglés) (http://www.faqs.org/rfcs/rfc2136.html).

El DHCP es una alternativa a otros protocolos de gestión de direcciones IP de red, como el BOOTP (*Bootstrap Protocol*). DHCP es un protocolo más avanzado, pero ambos son los usados normalmente. Cuando el DHCP es incapaz de asignar una dirección IP, se utiliza un proceso llamado "Automatic Private Internet Protocol Addressing".

3.2.3 Parámetros configurables

Un servidor DHCP puede proveer de una configuración opcional a la computadora cliente. Dichas opciones están definidas en RFC 2132 (Inglés) (http://www.ietf.org/rfc/rfc2132.txt).

Lista de opciones configurables:

- Dirección del servidor DNS
- Nombre DNS
- Puerta de enlace de la dirección IP
- Dirección de Publicación Masiva (*broadcast address*).
- Máscara de subred.
- Tiempo máximo de espera del ARP (*Protocolo de Resolución de Direcciones* según siglas en inglés)
- MTU (*Unidad de Transferencia Máxima* según siglas en inglés) para la interfaz
- Servidores NIS (*Servicio de Información de Red* según siglas en inglés)
- Dominios NIS
- Servidores NTP (*Protocolo de Tiempo de Red* según siglas en inglés))
- Servidor SMTP
- Servidor TFTP
- Nombre del servidor WINS

3.3 Instalación del servidor DHCP

El servidor DHCP se obtiene descargándolo de la Internet, aunque ya viene precargado en la versión de slackware que se está utilizando.

En el caso de que se quisiera instalar manualmente se sigue el siguiente procedimiento:

1. Descargar la versión actualizada del archivo desde *www.linux.org,* dicho archivo tiene la extensión .tar.gz
2. Copiarlo al directorio deseado, en este caso será en /tmp.
3. Dentro de /tmp ejecutar tar –zxvf dhcp -2.0.tar.gz para descomprimirlo.
4. Acceder al directorio de las fuentes (/temp/ dhcp -2.0) y ejecutar. /configure. Este comando sirve para detectar el sistema operativo y el kernel utilizado.
5. Se ejecuta el comando make para compilar las fuentes.
6. Al finalizar la compilación, ejecutar make install para copiar los binarios a directorios ejecutables

Para iniciar el servicio de DHCP se ejecuta

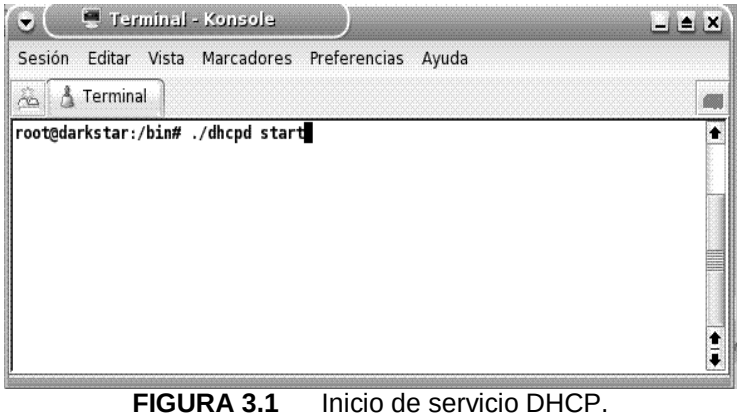


FIGURA 3.1 Inicio de servicio DHCP.

Instalado el software del servidor, nuestra atención caerá sobre dos archivos importantes: /etc/dhcpd.conf, y /var/state/dhcp/dhcpd.leases. El primero nos permitirá configurar el servidor a según nuestros requerimientos, mientras que el segundo es una base de datos creada por el servidor, con las asignaciones de IP que se van realizando.

Este último archivo es importante, porque permite en primer lugar verificar la actividad del servidor, y en segundo (y más importante que nuestra curiosidad), permite al servidor llevar cuenta de las IPs que va prestando a los distintos clientes, para conservarlas en caso de caídas. La capacidad de asignar IPs dinámicamente a equipos desconocidos junto a la de recuperar las IPs asignadas con anterioridad, sobre una base de tiempos de caducidad renovables, constituyen la ventaja más notable sobre el predecesor de DHCP, BOOTP. Cabe destacar respecto al archivo leases, que el servidor udhcpd, pensando en sistemas embebidos que no siempre cuentan con discos rígidos, no solo puede actualizar su .leases en base a tiempos, sino también a señales, lo cual le hace ideal para instalaciones tipo Disk-on-a-chip.

Visto esto, nuestro archivo de configuración debe quedar de esta manera a fin de ajustar las IPs a la red, estos cambios se deben insertar en el archivo dhcpd.conf ubicado debajo de /etc/:

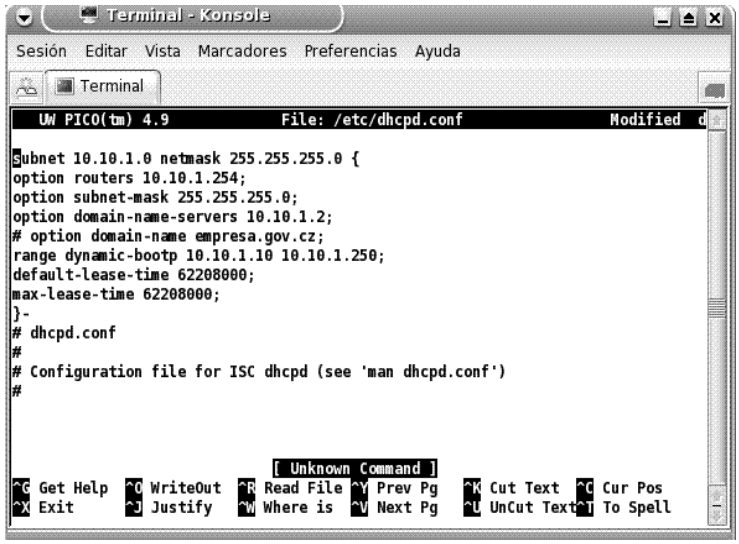


FIGURA 3.2 Edición del archivo dhcpd.conf.

La primera línea, describe la red y es posible definir más de una subred, en caso de que compartan la misma infraestructura física. Definida la red y la máscara, indicaremos la ruta por defecto (gateway, puerta de enlace o pasarela son términos equivalentes), el o los servidores de nombres, opcionalmente un nombre de dominio y el rango de IPs que estarán disponibles para asignarlas a los clientes que las soliciten. Nótese que se han reservado los

primeros 10 IPs y los últimos 4, por si necesitamos ocuparlas de modo estático con routers, por ejemplo.

Las dos líneas restantes indican el tiempo por defecto y máximo de vencimiento de la asignación de IPs en segundos, y en el ejemplo equivalen a cuatro años. Para que se ahorren cálculos, sírvanse una lista de valores que pueden resultarles de interés:

Con la configuración anterior, las el servidor ya se encuentra en funcionamiento. Aunque desde luego el protocolo es mucho más potente. Es posible por ejemplo asignar IPs fijas por MAC address, agregando la siguiente declaración:

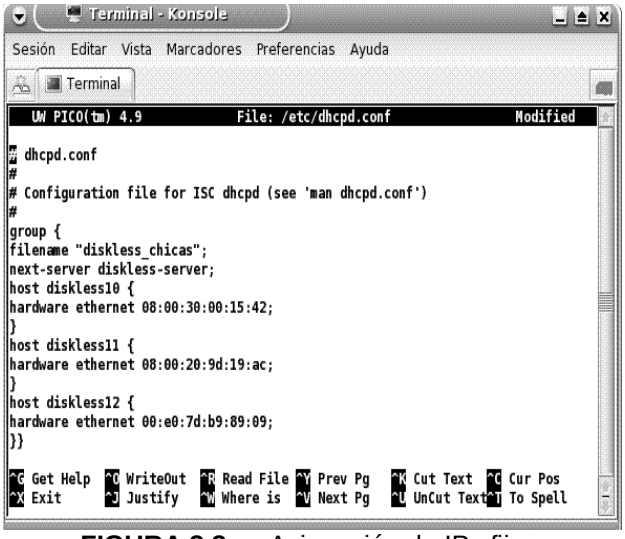


FIGURA 3.3 Asignación de IPs fijas.

La línea comentada se refiere a una imagen de sistema operativo que deberá obtener la máquina cliente en cuestión, en caso de que se trate de una máquina sin disco. El tema de las terminales sin disco, involucra otros protocolos como tftp y nfs.

Es posible también agrupar varios hosts para indicarles opciones a todos, como sigue:

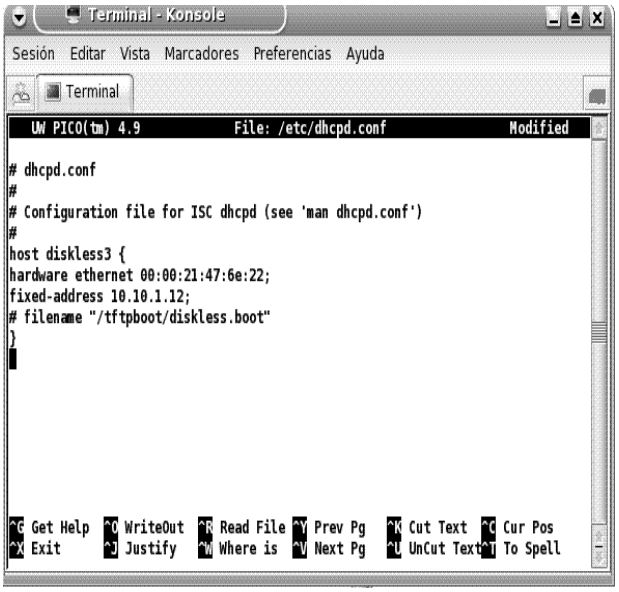


FIGURA 3.4 Agrupación de hosts.

A continuación se configurarán los clientes.

De nada sirve tener andando el servidor DHCP, si las máquinas cliente están configuradas con IPs estáticos. De modo tal que si es su caso, éstas tendrán que ser reconfiguradas.

En el caso de terminales Windows bastará seguir la configuración por defecto. De la versión de Windows dependerá como llegar a él (Panel de Control/Red/TCP-IP para Win9x, o Conexiones de Red/Red de Área Local/Propiedades/TCP-IP para versiones subsiguientes):

- Obtener IP automáticamente
- DNS desactivado
- WINS desactivado

Si se trata de una terminal Linux, según su distribución dispondrá de un interfaz (control-panel, linuxconf, netconf, redhat-config-network, drake). Para el caso de Slackware, se edita el archivo /etc/network/interfaces, indicando dhcp como método en lugar de static, si es que no ha sido cambiado ya. Cualquiera que sea la distribución, los scripts de inicio (en rc.) deben invocar al demonio dhcpcd o a pump, indicándoles el dispositivo que necesitan configurar.

CAPÍTULO IV
ADMINISTRACIÓN Y CONFIGURACIÓN DEL SERVIDOR DE ARCHIVOS

CAPÍTULO IV ADMINISTRACIÓN Y CONFIGURACIÓN DEL SERVIDOR DE ARCHIVOS

4.1 Introducción

Ante la necesidad de compartir archivos existe el servicio de FTP el cual permite intercambiar ficheros en red. Pero presenta serios problemas de integración:

Su uso no es transparente, es decir, cambia según tratemos con estaciones de trabajo Unix o Windows.

Por otro lado, existe también el protocolo NFS, que es una solución limitada a máquinas UNIX Entre máquinas Unix, es posible usar el protocolo NFS para compartir ficheros. Se trata de una gran solución puesto que permite conservar todas las funcionalidades del sistema de ficheros Unix. Aún así, presenta una serie de inconvenientes:
NFS presenta problemas de seguridad.
No existe una buena implementación libre de NFS para equipos Windows.

En lugar de usar una solución, costosa, en los equipos Windows, es más económico y lleva menos trabajo utilizar el protocolo utilizado nativamente por las maquinas Windows. Este protocolo, llamado Common Internet FileSystem (CIFS), tiene implementaciones sobre un gran número de plataformas.

Existe una implementación libre de este protocolo llamada SaMBa, que permite utilizarlo sobre servidores Unix

4.2 Samba

Samba es una implementación bajo Unix de los protocolos CIFS y NetBIOS (antiguamente llamado SMB, de allí el nombre de SAMBA). Samba es una implementación libre del protocolo de archivos compartidos de Microsoft Windows (antiguamente llamado SMB, renombrado recientemente a CIFS) para sistemas de tipo UNIX. De esta forma, es posible que ordenadores con Linux o Mac OS X se vean como servidores o actúen como clientes en redes de Windows. Samba también permite validar usuarios haciendo de Controlador Principal de Dominio (PDC), como miembro de dominio e incluso como un dominio Active Directory para redes basadas en Windows; aparte de ser capaz de servir colas de impresión, directorios compartidos y autenticar con su propio archivo de usuarios.

Entre los sistemas tipo Unix en los que se puede ejecutar Samba, están las distribuciones GNU/Linux, Solaris y las diferentes variantes BSD entre las que podemos encontrar el Mac OS X Server de Apple.

4.2.1 Características

Samba es una implementación de una docena de servicios y una docena de protocolos, entre los que están NetBIOS sobre TCP/IP (NetBT), SMB (también conocido como CIFS), DCE/RPC o más concretamente, MSRPC, el servidor WINS también conocido como el servidor de nombres NetBIOS (NBNS), la suite de protocolos del dominio NT, con su Logon de entrada a dominio, la base de datos del gestor de cuentas seguras (SAM), el servicio Local Security Authority (LSA) o autoridad de seguridad local, el servicio de impresoras de NT y recientemente el Logon de entrada de Active Directory, que incluye una versión modificada de Kerberos y una versión modificada de LDAP. Todos estos servicios y protocolos son frecuentemente referidos de un modo incorrecto como NetBIOS o SMB.

Samba configura directorios Unix/Linux (incluyendo sus subdirectorios) como recursos para compartir a través de la red. Para los usuarios de Microsoft Windows, estos recursos aparecen como carpetas normales de red. Los usuarios de Linux pueden montar en sus sistemas de archivos estás unidades de red como si fueran dispositivos locales, o utilizar la orden smbclient para conectarse a ellas muy al estilo del cliente de la línea de órdenes ftp.

Cada directorio puede tener diferentes permisos de acceso sobrepuestos a las protecciones del sistema de archivos que se esté usando en Linux. Por ejemplo, las carpetas home pueden tener permisos de lectura y escritura para cada usuario, permitiendo que cada uno acceda a sus propios archivos; sin embargo, deberemos cambiar los permisos de los archivos localmente para dejar al resto ver nuestros archivos, ya que con dar permisos de escritura en el recurso no será suficiente.

El paquete de SAMBA incluye utilidades para controlar el acceso de los archivos con la misma soltura que un WindowsNT. Además Samba puede colaborar con un servidor NT existente, o reemplazarlo del todo.

Con samba es posible: Proteger por contraseña el acceso a un directorio compartido. Proteger con una contraseña personificada para cada usuario, y dotar de permisos de acceso individualizados.

La configuración de Samba se consigue editando un solo archivo, accesible en */etc/smb.conf* o en */etc/samba/smb.conf*.

4.3 Paquetes a instalar

Los paquetes de samba suelen ser 3 (al menos en la distribución slackware):
1. samba-common
2. samba
3. samba-client

El primer paquete tiene los elementos que van a permitir el buen funcionamiento de los otros dos: Las herramientas de conversión de tablas de caracteres Windows, los ficheros de configuración y la documentación.

El segundo paquete contiene todos los programas del servidor, es decir: aplicaciones que permiten hacer accesible los recursos a los usuarios, herramientas de configuración y la documentación esencial de Samba.

El último paquete contiene los programas clientes, que permiten acceder a los recursos compartidos

4.3.1 Proceso de instalación

Para instalar Samba, descargarnos el código fuente desde www.samba.org para compilarlos, posteriormente en una consola tecleamos los siguientes comandos:

```
user@shell:~$ tar zxfv samba-a.b.c-iy86.tar.gz
user@shell:~$ cd samba-a.b.c-iy86
user@shell:~$ ./configure
user@shell:~$ make
user@shell:~$ su
password:
root@shell:~# make install
```

4.4 Presentación de los servidores

4.4.1 Los dos demonios

Dos demonios se encargan de ofrecer los servicios de la conjunto de aplicaciones del Samba. El primero es el smbd y el segundo de ellos es el nmbd. smbd es el demonio que se encarga de la compartición de recursos: ficheros, impresoras, etc. pero también del control del acceso a los recursos. Gestiona los permisos de los diferentes clientes una vez que estos han sido identificados.

El demonio nmbd se ocupa de anunciar servicios. Es decir, se encarga de informar a las máquinas presentes en la red sobre cuales son los recursos disponibles. Este demonio maneja también la resolución de nombres de NetBIOS[1]. Puede para ello comunicarse con un servidor WINS (Windows Internet Naming Service) presente en la red.

4.5 Las herramientas de configuración

Existen dos formas para realizar la configuración:

Es posible editar directamente los ficheros de configuración con un editor de texto, pero podemos configurar esos mismos ficheros con la ayuda de una interfaz gráfica, obteniendo idéntico resultado.

Nosotros veremos aquí el manejo de Swat (Samba Web Administration Tool). Se trata de una interfaz que se comporta como un servidor Web, conectándose a la máquina por medio de un simple navegador[2]. Es posible leer la documentación, cambiar la configuración y realizar las demás tareas administrativas después de habernos validado con un usuario y una contraseña.

4.5.1 Las herramientas del cliente

Las herramientas para el cliente bajo Microsoft Windows son aquellas utilizadas habitualmente para trabajar con servidores NT. No hay que cambiar nada en este sentido. El funcionamiento para las maquinas Windows es totalmente transparente. Para GNU/Linux, existen en el paquete samba-client programas cliente para los servicios CIFS que sean proporcionados por un servidor Windows o por un servidor Unix usando Samba.

4.5.2 Configuración con SWAT

La herramienta SWAT es el ejemplo de una buena interfaz de administración gráfica. Intenta de forma relativamente transparente poder proporcionar todas las funcionalidades de la configuración en modo texto.

4.5.3 Presentación de los ficheros en modo texto

El fichero /etc/smbpasswd contiene los passwords de los usuarios de Samba, de forma cifrada.

El fichero /etc/lmhosts es un interfaz entre los nombres de máquinas NetBIOS y las direcciones IP numéricas. Su formato es parecido al de /etc/hosts.

El fichero /etc/smbusers contiene una lista de usuarios del sistema, seguida de una lista de usuarios de Samba que disponen de los derechos de esos usuarios. De esta forma es posible crear varios usuarios Samba sin tener que crear para cada uno de ellos un usuario del sistema.

4.5.4 Los menús de SWAT

4.5.4.1 Las secciones del smb.conf

Los menús GLOBALS, SHARES, PRINTERS son parecidos a los de las secciones existentes en el fichero /etc/smb.conf, que se presenta como un fichero .ini habitual del mundo Windows.

---

[1] La resolución de nombres consiste en obtener una equivalencia entre la dirección IP de la LAN y el nombre de la máquina.

[2] El servidor Swat suele ejecutarse en el puerto 901, para no entrar en conflicto con el servidor de http (Servidor Web) que suele escuchar el puerto 80.

El menú GLOBALS contiene variables generales que se aplican al total de los recursos puestos a disposición del servidor de SMB. Esta sección contiene también información de identificación del servidor dentro de la red NetBIOS: grupo de trabajo, nombre e identificador. Esta sección contiene también los modos de funcionamiento de Samba.

El menú SHARES contiene la lista de comparticiones de disco efectuadas por la máquina. Se aconseja primero crear la partición compartida y después precisar para cada partición sus propiedades particulares[3].

El menú PRINTERS es casi idéntico al anterior, pero permite compartir impresoras en lugar de particiones de disco.

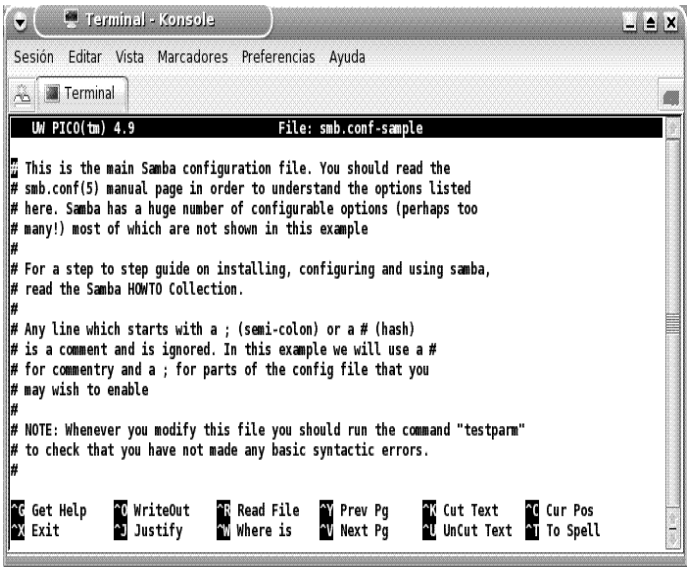


FIGURA 4.1 El archivo smb.conf.

4.5.4.2 Los otros menús

El menú HOME permite acceder a la versión HTML de la documentación de Samba.

El menú VIEW nos permite ver el fichero smb.conf tal cual ha sido redactado por SWAT. Es posible ver también la totalidad de las opciones posibles, incluso las que SWAT no ha cambiado, pero que tienen un valor por defecto.

El menú PASSWORD permite al usuario cambiar su contraseña. Se trata de un interfaz gráfico para el programa smbpasswd. Sirve también al administrador para añadir nuevos usuarios.

4.5.5 Teoría de funcionamiento de CIFS

Sobre una misma red, varias máquinas pueden poner recursos a disposición de otras. CIFS dispone de un sistema para anunciar servicios (browsing), que permite saber que recursos compartidos hay disponibles.

Cada máquina que desea anunciar sus recursos compartidos a las otras máquinas contacta con una máquina en particular, la Servidora de Anuncios (Master Browser) que se encarga de centralizar estas notificaciones de presencia. Es posible configurar el servidor Samba para que sea el mismo Servidor de Anuncios o dejar esta tarea a una máquina Windows.

[3] Aquí hablo de particiones, pero también vale para carpetas/directorios compartidos.

4.5.6 Acceso a recursos

El acceso a los recursos puede controlarse de dos formas:

- Escondiendo el recurso, es decir, no anunciándolo a ciertas máquinas de la red.
- Estableciendo un sistema de validación basado en contraseña, para restringir el acceso.

El anuncio de servicios esta limitado al "grupo de trabajo". Cada máquina Windows puede ser miembro de un solo grupo, y por tanto solo puede pertenecer a un conjunto de máquinas que compartan los mismos recursos.

Es posible de este modo separar conjuntos de recursos compartidos, creando distintos grupos de trabajo. Si lo que deseamos es tener máquinas accediendo a los recursos de varios grupos distintos, es necesario pasar por un sistema de autentificación.

Existen formas distintas de autentificación, cada una con sus ventajas e inconvenientes.

La autentificación por usuario/contraseña. Se trata del método por defecto. Representa la ventaja de permitir una gestión fina de los permisos. Para cada usuario es posible definir el acceso o no a unos recursos. Este método presenta un inconveniente: cada usuario debe disponer de una cuenta en la máquina Unix, para permitir la autentificación.

El control de acceso por comparticiones. Se trata de un método más global: cada recurso compartido es protegido por un password propio. Para ello es necesario que varios usuarios conozcan el mismo password y que recuerden la contraseña adecuada para cada recurso compartido al que accedan. Este método presenta la ventaja de que no son necesarias tantas cuentas de usuario como usuarios haya, sino tantas como recursos se compartan.

Autentificación contra otro servidor. Existen también dos métodos indirectos de control de acceso. El primero, el método server, consiste en consultar con otro servidor CIFS, que se encargara de la autentificación. El segundo método, domain, consiste en validarse contra el servidor de dominio NT[4].

4.5.7 Configuración de los parámetros globales

Identificar el servidor

Primero hay que elegir algunos parámetros de funcionamiento del servidor, para que se integre bien en la red.

El campo server string, permite elegir la descripción que acompaña al nombre del servidor en la lista de recursos anunciados.

El campo netbios name, permite definir el nombre de la máquina, no como un nombre de DNS, sino como un nombre de resolución de nombres propio del protocolo NetBIOS. Es importante entender que son dos cosas totalmente diferentes.

El campo workgroup, permite elegir el grupo de trabajo del que el servidor Samba hace parte. En este caso el grupo de trabajo al que están integradas los equipos de la empresa cuyo nombre es BRAINUP.

El campo interfaces permite identificar la o las tarjetas de red que enlazan el servidor con el grupo de trabajo.

[4] Un dominio NT es un conjunto de máquinas que comparten a la vez recursos y un proceso de autentificación común.

El campo security permite elegir el método de autentificación, podemos elegir uno de los vistos anteriormente. Que en este caso utilizaremos de control de acceso por comparticiones.

Los menús hosts allow y host deny permiten controlar el acceso a los recursos de ciertas máquinas. Las configuraciones hechas en esta sección se aplican a la totalidad de los recursos compartidos, independientemente de la configuración especifica.

4.5.7.1 La sección [GLOBAL]

Las configuraciones realizadas por Swat se reflejan en el fichero de configuración /etc/smb.conf. Si editamos dicho fichero podremos ver algo de este estilo:

```
[global]
workgroup = BRAINUP
server string = Servidor Samba
security = SHARE
log file = /var/log/samba/log.%m
max log size = 50
```

4.5.8 La autentificación por cada usuario

Las contraseñas encriptadas.

Por defecto, Samba no utiliza contraseñas cifradas. Esta elección le permite interoperar con clientes de Windows 3.x y Windows 95.

Pero por culpa de esta compatibilidad perdemos seguridad y es necesario tocar el registro del sistema de Windows en máquinas Windows 98 y posteriores para que todo funcione. Si en la red no hay maquinas Windows 95 o anteriores se aconseja configurar el servidor de Samba para que use contraseñas cifradas. Esto último se hace de esta forma, añadiendo en el fichero smb.conf la siguiente línea:

Dentro del [global] de smb.conf :

```
encrypt passwords = Yes
```

Estas contraseñas son almacenadas dentro del fichero /etc/smbpasswd. Las máquinas clientes contactan con el servidor y reciben una clave codificada usando la contraseña cifrada. El resultado es reenviado al servidor, que hace la misma operación. Si los dos resultados son idénticos la autentificación es correcta. Esto impide a un usuario “malicioso” hacerse con los passwords que atraviesan la red camino al servidor en busca de la autentificación.

4.5.9 Limitar el acceso a ciertos usuarios.

Para cada recurso es posible restringir el acceso a ciertos usuarios. Para cada una de las líneas de recursos compartidos en /etc/smb.conf, podemos añadir la línea:

```
valid users = gmedrano, amanjarrez,
```

En su ausencia, el recurso es accesible por todos los usuarios del servidor Samba. Si esta línea esta presente el acceso esta reservado únicamente a los usuarios mencionados.

4.5.10 Autorizar ciertos accesos en modo sólo lectura

La opción read only, permite impedir a los usuarios que escriban en el directorio compartido. Podemos también limitar este acceso a unos usuarios concretos, para ello tenemos dos posibilidades:

- Autorizar el acceso de escritura y bloquear ciertos usuarios con derecho de solo lectura, colocando su nombre en la seccion read list= del recurso compartido.

- Autorizar el acceso en sólo lectura y dar el privilegio de escritura a ciertos usuarios gracias a la sección write list= del recurso compartido.

4.5.11 Autorizar la conexión de las máquinas NT

Las máquinas NT intentan conectarse directamente al servidor, y no a un recurso en concreto. Es por tanto preciso autorizarlas para ello. Es necesario que las máquinas (y no los usuarios) dispongan de una cuenta. Las máquinas no van a conectarse al shell, asi que no es necesario darles un usuario del sistema con su directorio personal y demás.

El identificador de una máquina es su nombre NetBIOS, seguido del carácter $. Así por ejemplo la máquina icerberg, tendrá como identificador iceberg$. Hecho lo cual hay que añadir esta cuenta de usuario a la base de datos de los usuarios de Samba, con el comando:
smbpasswd -a -m máquina

4.5.12 Los clientes

Acceder a los recursos compartidos: smbclient

Este comando permite acceder, desde un cliente GNU/Linux, a recursos puestos a disposición a través de servidores CIFS, bien se trate de un servidor Samba o de un servidor basado en Microsoft Windows. La interfaz es parecida a la del ftp, es de este modo posible transferir ficheros sin esfuerzo. La sintaxis es:
smbclient //máquina/recurso

El recurso puede ser bien un directorio o bien una impresora, o un disco compartido al que se desea acceder.

El nombre de la máquina es su nombre de NetBIOS, que puede (y suele) ser diferente de su nombre de DNS.

La opción -R permite elegir el modo de resolución del nombre de la máquina:

-R lmhosts permite consultar el fichero /etc/lmhosts, que resuelve nombres de IP contra nombres de NetBIOS de la máquina.

-R wins permite lanzar la consulta a un servidor WINS para obtener dicha conversión.

Una vez conectado al servicio en cuestión, disponemos de una interfaz de transferencia de ficheros idéntica a la del FTP.

Disponemos de algunas opciones extra, tales como print fichero, para imprimir un fichero local en el servidor. Integrar un recurso compartido en nuestra jerarquía de directorios:

El comando smbmount nos permitirá movernos de una manera más cómoda por los recursos compartidos vía CIFS. Se comporta de una forma similar a los montajes vía NFS: el recurso compartido CIFS se monta en un punto de nuestra jerarquía de directorios y podemos movernos por el usando los comandos Unix habituales. smbclient se encarga de gestionar las interacciones entre los ficheros presentes en el servidor.

Para desmontar un recurso compartido usamos el comando smbumount.

4.5.13 Guardar datos de un recurso compartido

Para guardar datos en un recurso compartido, se utiliza el comando smbtar. El comando smbtar es muy similar al comando tar de Linux. Permite realizar copias de seguridad de los archivos del servidor desde la máquina cliente Samba. La sintaxis es la siguiente:
smbtar -s servidor -x recurso -t lugar_de_almacenamiento

CAPÍTULO V
INSTALACIÓN, CONFIGURACIÓN Y ADMINISTRACIÓN DEL SERVIDOR WEB

CAPÍTULO V INSTALACIÓN, CONFIGURACIÓN Y ADMINISTRACIÓN DEL SERVIDOR WEB

5.1 Servidores Web y Transferencia de Hipertexto

Un servidor Web es un programa que implementa el *protocolo HTTP* (*hypertext transfer protocol*). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

Sin embargo, el hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un formato de archivo y HTTP es un protocolo.

Cabe destacar el hecho de que la palabra *servidor* identifica tanto al programa como a la máquina en la que dicho programa se ejecuta. Existe, por tanto, cierta ambigüedad en el término, aunque no será difícil diferenciar a cuál de los dos nos referimos en cada caso.

5.2 Protocolo

Un servidor Web se encarga de mantenerse a la espera de *peticiones HTTP* llevada a cabo por un *cliente HTTP* que solemos conocer como *navegador*. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. El cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Sobre el servicio web *clásico* podemos disponer de aplicaciones web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

Aplicaciones en el lado del cliente: el cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas *scripts*). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje *javascript* y *java*, aunque pueden añadirse más lenguajes mediante el uso de *plugin*s

Aplicaciones en el lado del servidor: el servidor Web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP. Las aplicaciones de servidor suelen ser la opción por la que se opta en la mayoría de las ocasiones para realizar aplicaciones Web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad adicional, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador Web básico puede utilizar este tipo de aplicaciones. Algunos conceptos relacionados con las aplicaciones Web son:

- *PHP*
- *ASP*
- *Perl*
- *CGI*
- *.NET*
- JSP (Tecnología Java)

Algunos servidores Web importantes son:

- *Apache*
- *IIS*
- *Cherokee*

5.3 El servidor apache

El servidor HTTP Apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor *parcheado*).

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

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

Apache tiene amplia aceptación en la red: en el 2005, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios Web en el mundo y creciendo aún su cuota de mercado.[1]

5.4 Instalación del Servidor Web

Para el caso del servidor de la Empresa, se instalará el servidor Apache, un manejador de base de datos de código abierto, llamado MySQL; que podrá interactuar con este servidor. Y un lenguaje de programación, que ejecutará aplicaciones del lado del servidor llamado PHP. Todo esto con la finalidad de crear una aplicación dinámica para la empresa.

5.4.1 MySQL

MySQL es un sistema de gestión de base de datos, multihilo y multiusuario. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso. Está desarrollado en su mayor parte en ANSI C[2].

Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL está poseído y patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius.

5.4.2 PHP

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. Las siglas significan "PHP Hypertext Pre-processor" (inicialmente PHP Tools, o, *P*ersonal *H*ome *P*age Tools), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la biblioteca GTK+[3].

[1] (estadísticas históricas y de uso diario proporcionadas por Netcraft).
[2] estandarización del lenguaje de programación C
[3] grupo importante de bibliotecas o rutinas para desarrollar interfaces gráficas de usuario (GUI).

5.4.3 Instalación de Apache

Para la instalación de apache, primero se descarga el código fuente desde el sitio oficial: *www.apache.org,* y posteriormente en una consola, ubicarnos en el directorio en donde se encuentra este archivo. Y a continuación ejecutar estos comandos:

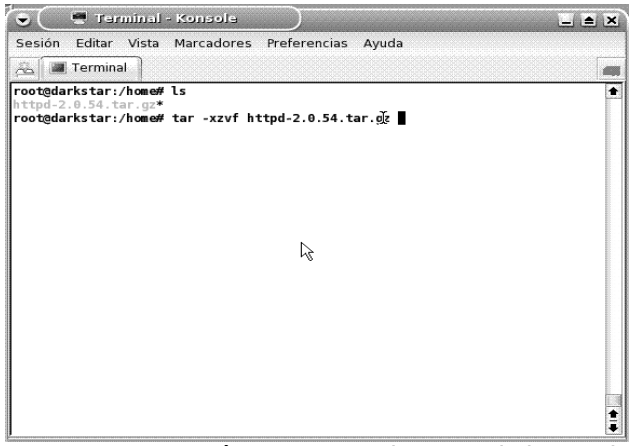


FIGURA 5.1 Parámetros para descomprimir Apache.

Para comprobar las dependencias:

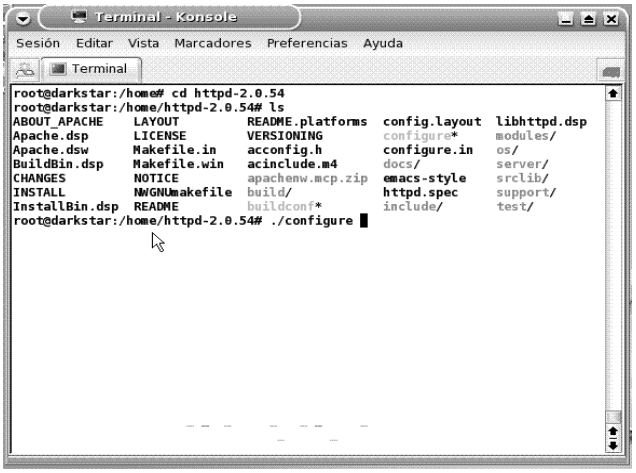


FIGURA 5.2 Comandos para comprobar dependencias entre módulos.

Se teclean los siguientes comandos para compilar e instalar las fuentes:

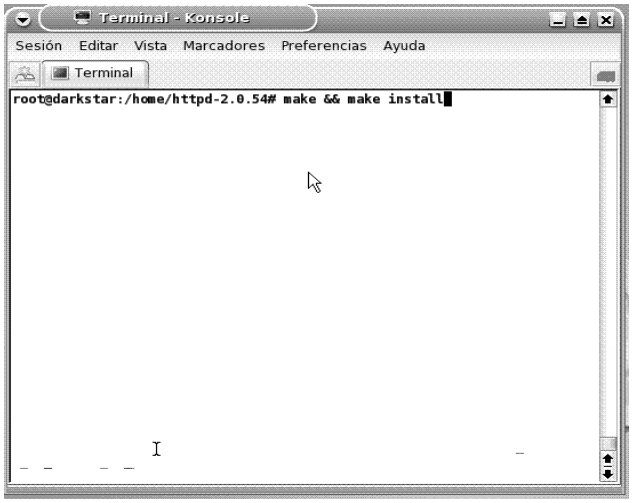


FIGURA 5.3 Instrucciones para compilar e instalar el código fuente.

Para iniciar el servicio de apache se ejecuta el siguiente comando:

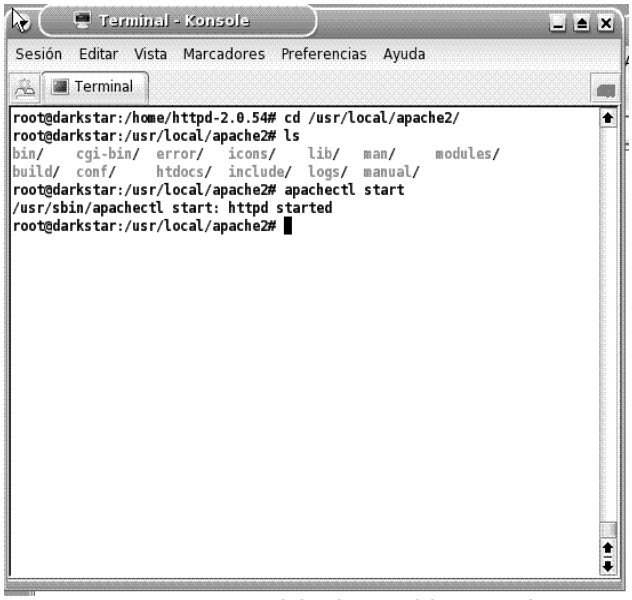


FIGURA 5.4 Inicio de servidor Apache.

Para comprobar que el servidor esta funcionando adecuadamente, en cualquier navegador web, se teclea: http://localhost/ y se debe mostrar la siguiente página:

FIGURA 5.5 Servidor Apache en ejecución.

5.4.4 Instalación de PHP

Se deben descargar el código fuente desde el sitio oficial: www.php.net, y posteriormente en una consola, ubicarnos en el directorio en donde se encuentra este archivo, y a continuación ejecutar estos comandos

Se descomprime el código fuente:

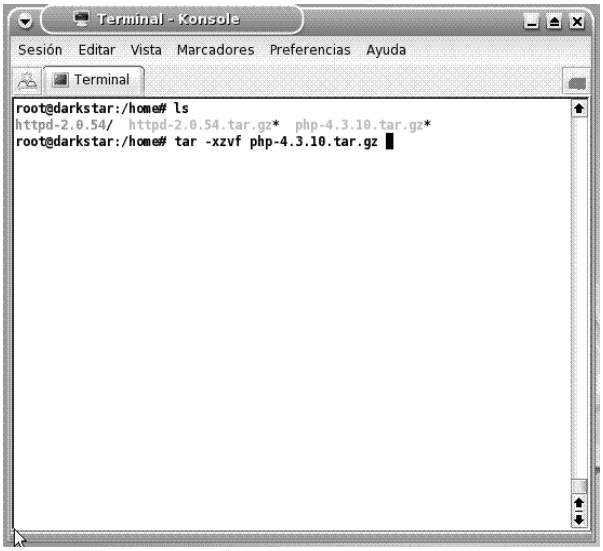


FIGURA 5.6 Parámetros para descomprimir PHP.

Para comprobar dependencias, e indicarle el destino de instalación, se ejecuta el siguiente comando:

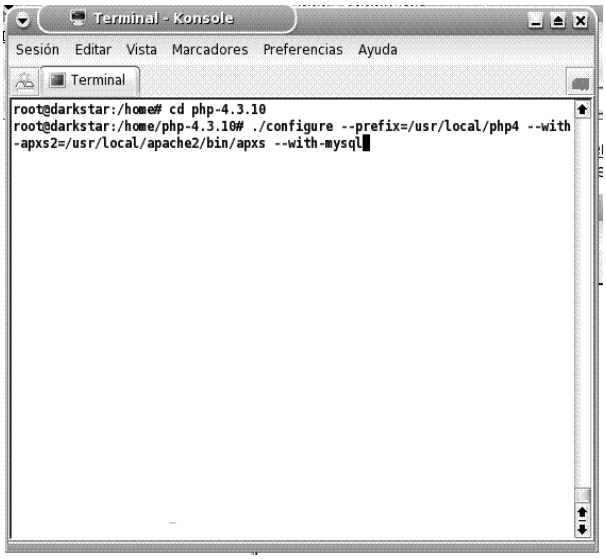


FIGURA 5.7 Comandos para comprobar dependencias entre módulos.

Se compila e instala el código fuente con las siguientes sentencias:

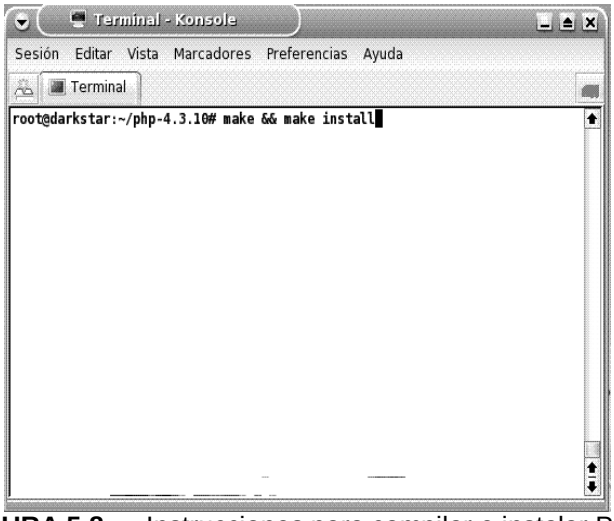


FIGURA 5.8 Instrucciones para compilar e instalar PHP.

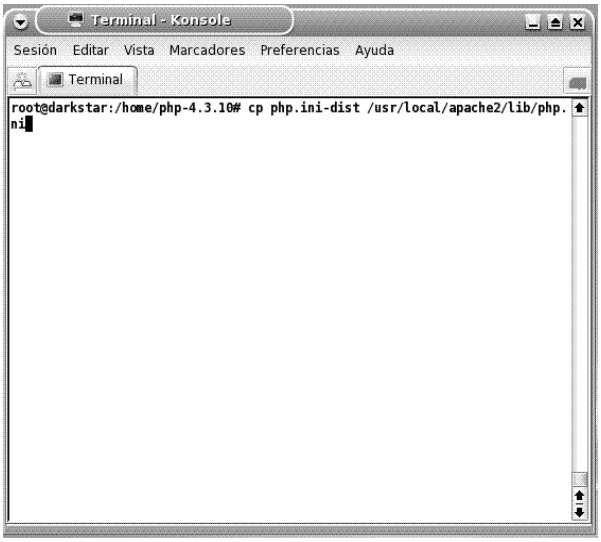


FIGURA 5.9 Sentencia para vincular PHP con el servidor Apache.

A continuación y con un editor de texto cualquiera, se deben agregar las siguientes líneas al archivo httpd.conf que es el archivo de configuración de apache; este lo encontramos en la ruta /usr/local/apache

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

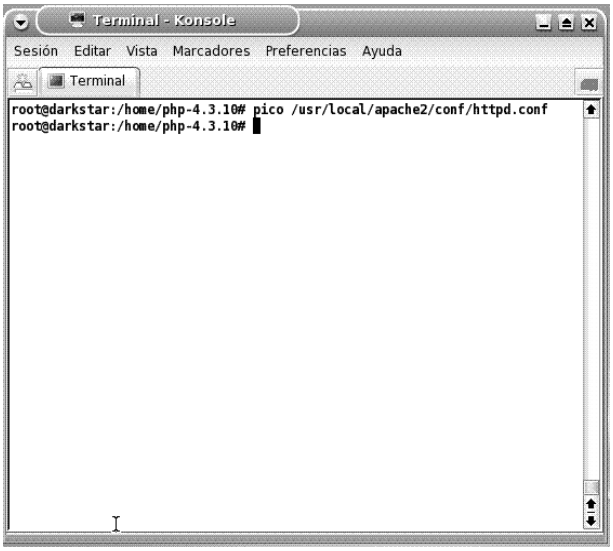


FIGURA 5.10 Edición del archivo de configuración de Apache.

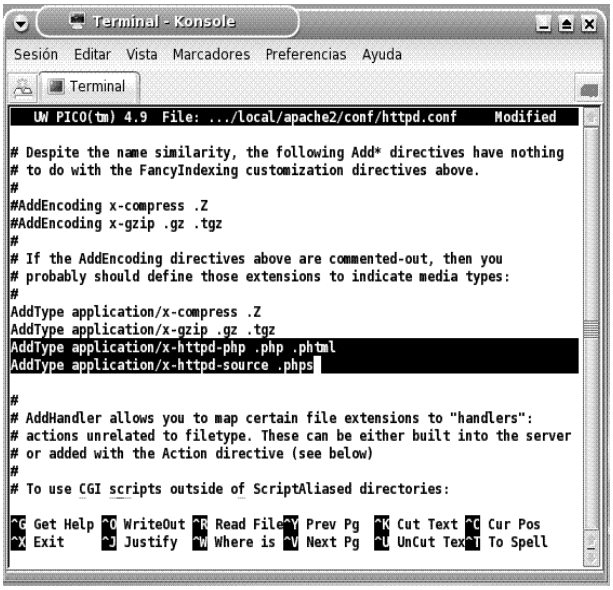


FIGURA 5.11 Edición del archivo httpd.conf.

En el mismo archivo de configuración, se debe verificar que exista la línea

LoadModule php4_modules/libphp.so

Crear un archivo llamado info.php y guardarlo en el document root de apache /usr/local/apache2/htdocs. Con el siguiente contenido:

<¿ phpinfo();?>

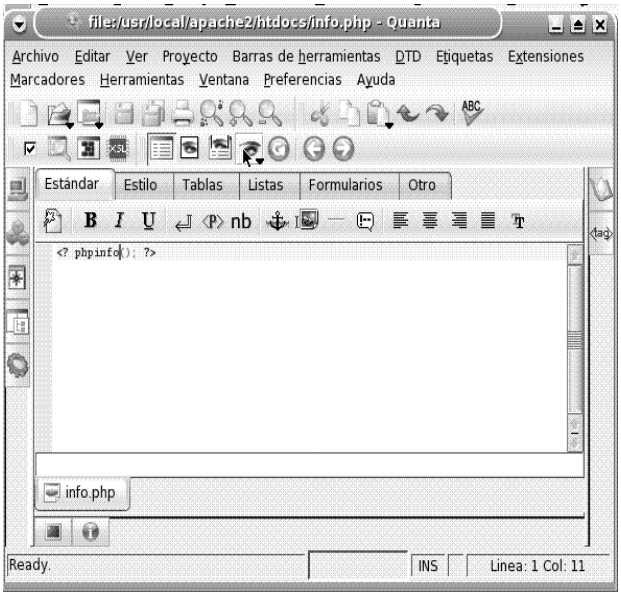


FIGURA 5.12 Creación de archivo info.php.

Reiniciar apache con el siguiente comando:

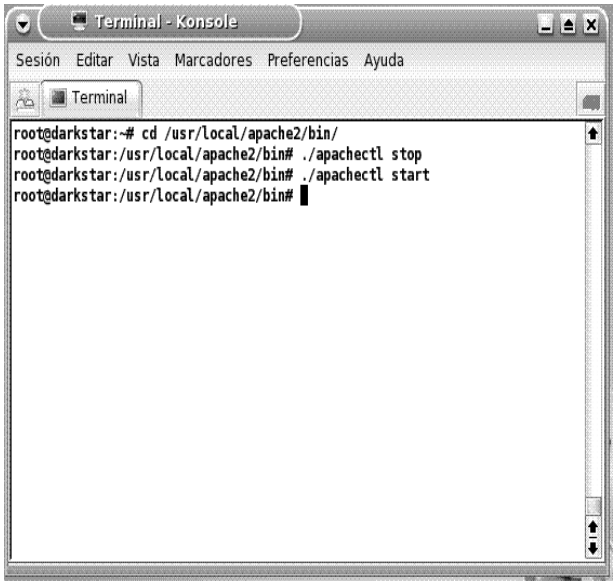


FIGURA 5.13 Reinicio de servidor Web.

Teclear en un navegador cualquiera:

http://localhost/info.php

Aquí deberá aparecer una página, con las características del lenguaje PHP, esto es únicamente para cerciorarnos que el servidor está funcionando adecuadamente.

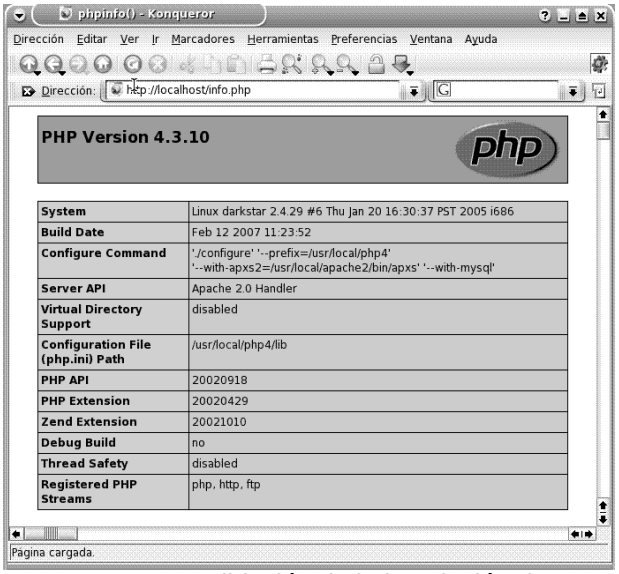


FIGURA 5.14 Validación de la instalación de PHP.

5.5.5 Instalación de Mysql

Descargar el código fuente desde www.mysql.com

Descomprimir con el comando

Tar –xzvf mysql.tar.gz

Ejecutar el siguiente comando para comprobar dependencias:

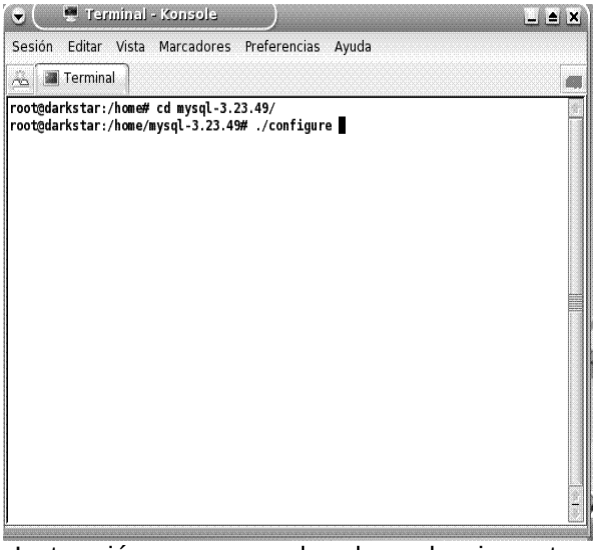


FIGURA 5.15 Instrucción para comprobar dependencias entre módulos.

Compilar e instalar el código fuente con las siguientes sentencias:

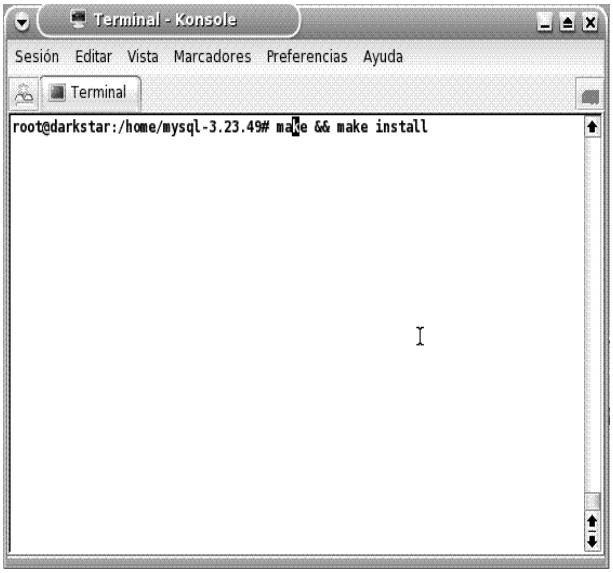


FIGURA 5.16 Instrucciones para compilar e instalar el MySQL.

Ejecutar el siguiente commando

usrl/local/mysql/bin /./mysql_install_db –user=mysql

chown –R root (En el directorio mysql4)

chown –R mysql var

chgrp –R mysql

Par levantar el servidor se ejecuta el siguiente comando:

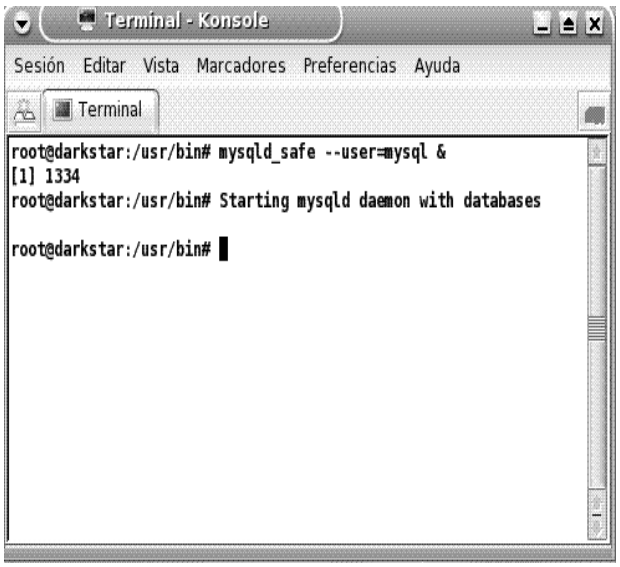


Figura 5.17 Inicio del servicio MySQL.

Y de esta manera queda instalado el servidor apache, con soporte para el lenguaje PHP y el servidor de base de datos MySQL.

Es importante señalar, que para el caso particular de la empresa, este servidor sólo estará disponible en la intranet y con posibilidades para publicarlo en Internet, siempre y cuando se adquiera una dirección IP fija y un dominio.

CAPÍTULO VI
DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN DINÁMICA PARA EL WEB APOYADO EN HTML, EL LENGUAJE PHP, Y EL MANEJADOR DE BASE DE DATOS MYSQL

CAPÍTULO VI DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN DINÁMICA PARA EL WEB APOYADO EN HTML, EL LENGUAJE PHP, Y EL MANEJADOR DE BASE DE DATOS MYSQL

6.1 Introducción

El departamento de informática, requiere de un sistema que pueda manejar el inventario de equipo de cómputo de la empresa; a fin de tener el control de las computadoras existentes de la misma. En este sistema, se requiere dar de alta o de baja los equipos de cómputo, así como el conocer las características de estos (sistema operativo, memoria, número de serie, etc.) Se desarrollara un aplicación, en la cual, los usuarios de la empresa, debidamente validados, podrán acceder a información correspondiente a este equipamiento tecnológico, está aplicación se desarrollará en el lenguaje PHP, HTML; y el manejador de bases de datos Mysql, que ya hemos instalado en el servidor.

Existe un programa, en la distribución Slackware; llamado Quanta *plus,* que es un editor que facilita la creación de páginas Web, es de distribución libre; y por ende ocuparé en la creación de esta aplicación Web.

6.2 Funcionamiento del sistema

El sistema funcionará a través de la intranet de la empresa, ya que la aplicación estará basada en el Web, se accederá por medio de un portal, en el cual los usuarios validados, podrán ejecutar cuatro acciones principales:

Insertar un equipo.- Dar de alta un equipo en el inventario de la empresa.
Eliminar un equipo.- Dar de baja un equipo en el inventario de la empresa.
Actualizar un equipo.- Modificar los datos de un equipo.
Consultar equipos.- Desplegar un listado de todos lo equipos dados de alta.

Tanto los usuarios validos, así como los equipos de cómputo, se guardarán en una base de datos en MySQL y se accederá a ella por medio del lenguaje de programación PHP, embebido con HTML. Todo ello, montado sobre el Servidor Apache, que ya se implementó en el capítulo anterior.

1.3 Diseño de la Base de Datos

La base de datos llamada “Pixup” contará con dos tablas, una para los usuarios que accederán al sistema, y otra para almacenar las características de los equipos de cómputo. La tabla de los usuarios se denominara “Admin”, y contará con los siguientes campos:

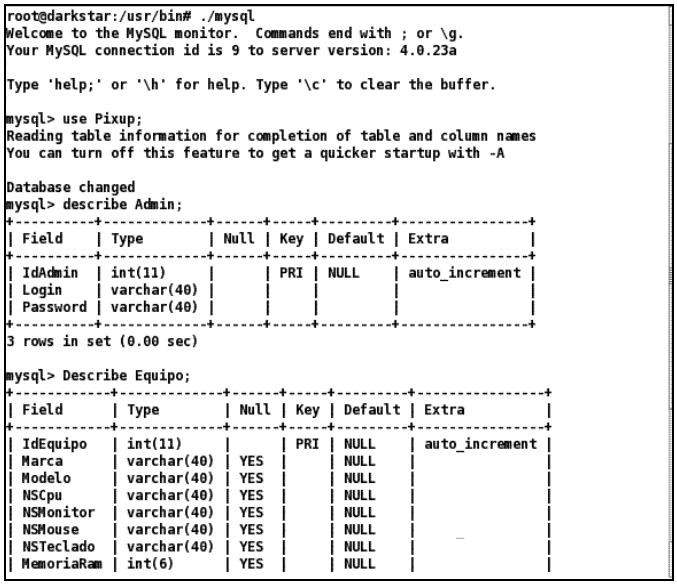
Id.- Identificador único de usuario
Login.- nombre del usuario que accederá al sistema.
Password.- Contraseña de usuario

Para la tabla de los equipos de cómputo se creará una tabla llamada Equipo, que poseerá los campos siguientes:

IdEquipo.- Identificador único del equipo de cómputo.
Marca.- Marca de fabricante de equipo de cómputo.
Modelo.- Modelo del equipo.
NSCpu.- Número de serie del CPU.
NSMonitor.- Número de serie del Monitor.
NSMouse.- Número de serie del mouse.
NSTeclado.- Número de serie del teclado.
MemoriaRam.- Cantidad de memoria RAM del equipo.
SistOp.- Tipo de Sistema operativo de la PC.
Disco Duro.- Capacidad del disco duro.

Usuario.- Usuario al que está asignado el equipo.

El Script para la creación de esta base de datos se incluye en el anexo II, para verificar que se haya creado adecuadamente la base de datos, se teclea en una consola los siguientes comandos:



```
root@darkstar:/usr/bin# ./mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9 to server version: 4.0.23a

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use Pixup;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> describe Admin;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| IdAdmin  | int(11)     |      | PRI | NULL    | auto_increment |
| Login    | varchar(40) |      |     |         |                |
| Password | varchar(40) |      |     |         |                |
+----------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> Describe Equipo;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| IdEquipo   | int(11)     |      | PRI | NULL    | auto_increment |
| Marca      | varchar(40) | YES  |     | NULL    |                |
| Modelo     | varchar(40) | YES  |     | NULL    |                |
| NSCpu      | varchar(40) | YES  |     | NULL    |                |
| NSMonitor  | varchar(40) | YES  |     | NULL    |                |
| NSMouse    | varchar(40) | YES  |     | NULL    |                |
| NSTeclado  | varchar(40) | YES  |     | NULL    |                |
| MemoriaRam | int(6)      | YES  |     | NULL    |                |
```

FIGURA 6.1 Descripción de la base de datos.

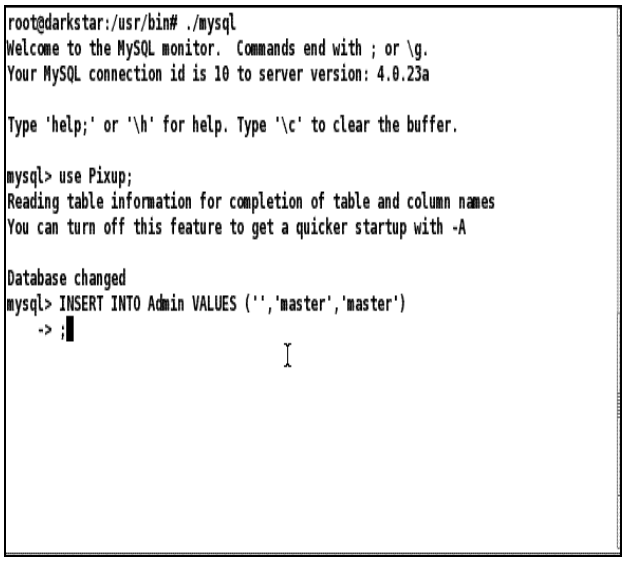
6.4 Diseño de la aplicación

El sistema estará compuesto de dos páginas principales, la primera será la página de autentificación, en la cual se le pedirá al usuario introduzca su nombre de usuario y contraseña, la segunda será la de administración del inventario, en esta se podrán seleccionar los opciones de dar de alta, actualizar, eliminar o listar los equipos existentes en la base de datos.

En el anexo II se incluye el código fuente de la aplicación que se desarrolló para este proyecto.

6.4.1 Página de acceso

El primer paso para autorizar el acceso a un usuario al sistema, será darlo de alta directamente en el manejador de bases de datos, es decir introducir manualmente el usuario y la contraseña en la tabla “Admín” de la base de datos “Pixup” que es la que estamos utilizando para el sistema:



```
root@darkstar:/usr/bin# ./mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10 to server version: 4.0.23a

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use Pixup;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> INSERT INTO Admin VALUES ('','master','master')
    -> ;
```

FIGURA 6.2 Inserción de datos en la tabla Admin.

De esta manera, ya hemos dado de alta al usuario master para su acceso a la página.

La página de acceso esta programada para consultar a la base de datos y hacer una comparación entre lo que tecleamos como nombre de usuario y contraseña, y los datos almacenados en la Tabla “Admin” de la base de datos “Pixup”, si el resultado de esta comparación es verdadero, se podrá acceder al sistema:

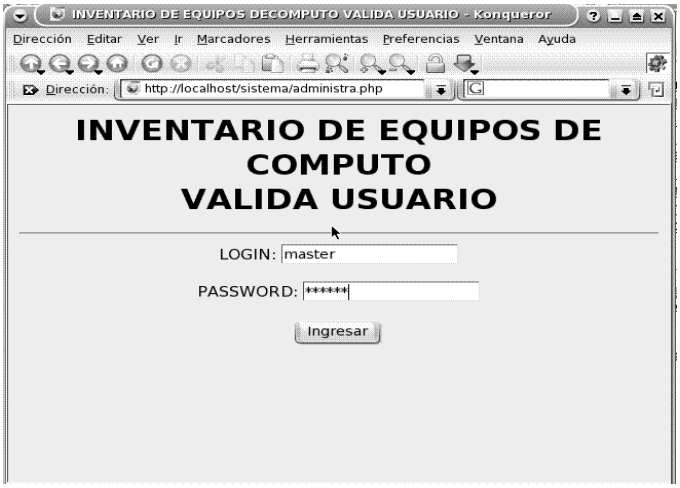


FIGURA 6.3 Página de acceso.

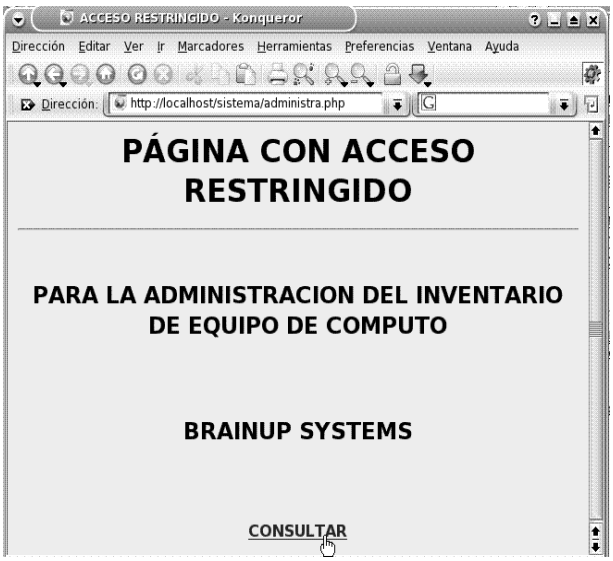


Figura 6.4 Página de inicio.

A continuación tenemos otra página en la que se nos indica, el nombre de usuario con el que estamos accediendo, y el link para acceder a la administración de los equipos de cómputo. Tenemos también otro link que da la opción de salir del sistema por completo.

FIGURA 6.5 Acceso a administración de equipos.

6.4.2 Administración de equipos

Al acceder a la página “Administrar equipos”, se podrá elegir entre las opciones de insertar, actualizar, eliminar y listar los equipos de cómputo.

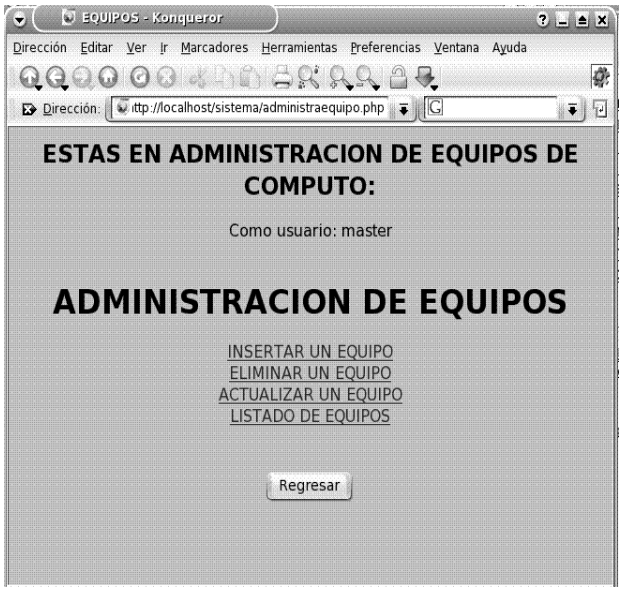


FIGURA 6.6 Página de administración de equipos.

6.4.3 Inserción de equipos

En esta página se podrán dar de alta equipos, al llenar el formulario, y guardar los datos, aparece una página que indica los cambios efectuados en la base de datos:

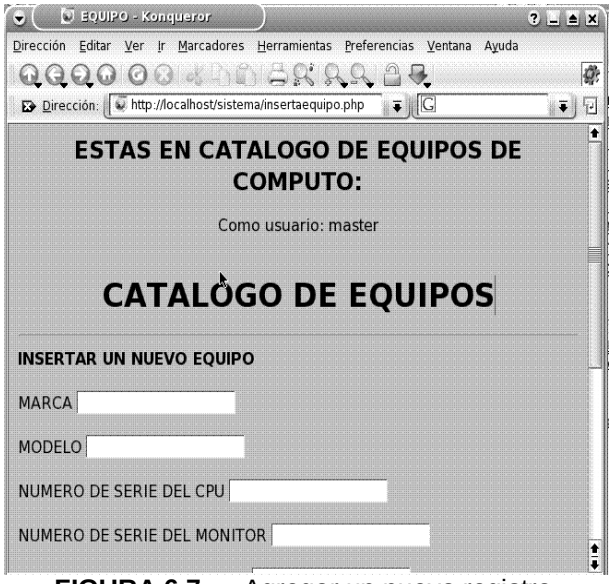


FIGURA 6.7 Agregar un nuevo registro.

6.4.4 Eliminación de equipos

Para eliminar de la base de datos un registro, bastará con elegir uno de ellos de la caja de selección y hacer click en el botón eliminar equipo. A continuación se desplegara una página en donde se informará que el cambio en la base de datos se realizó adecuadamente.

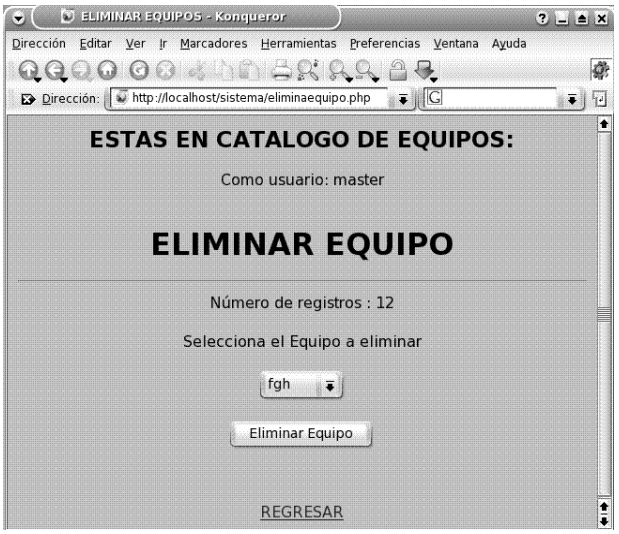


FIGURA 6.8 Eliminacíon de un registro.

6.4.5 Actualización de equipos

Al seleccionar esta opción, se podrán realizar cambios en un registro que este dado de alta en la base de datos, para realizar la actualización, se deberá de elegir un registro de la caja de selección y seleccionar en el botón “Actualizar Equipo”, esta acción nos llevara a un formulario en el que podremos modificar los parámetros del registro seleccionado. Al hacer la modificación aparece una página en la que se indica que el registro ha sido actualizado.

FIGURA 6.9 Actualización de un registro.

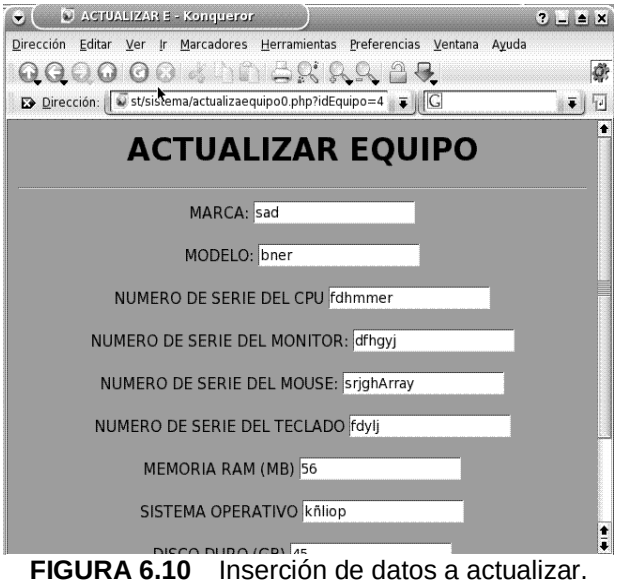


FIGURA 6.10 Inserción de datos a actualizar.

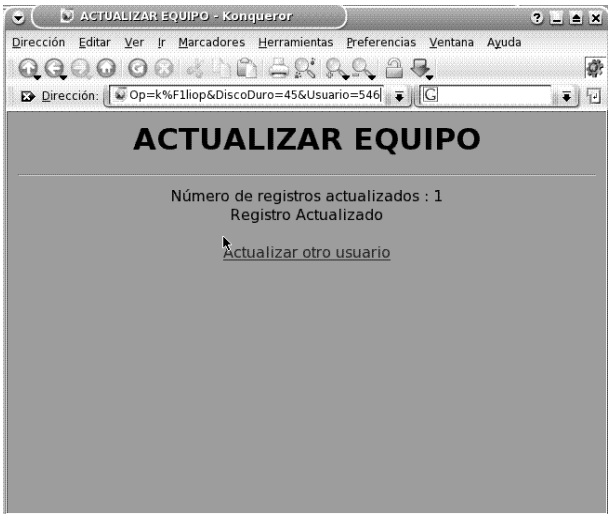


FIGURA 6.11 Validación de cambios efectuados.

6.4.6 Listado de equipos

Al elegir la opción LISTADO DE EQUIPOS, se desplegará un listado de todos los equipos dados de alta en el sistema.

FIGURA 6.12 Listado de registros.

6.5 Implementación del sistema en el servidor

Para que la aplicación funcione en el servidor, los archivos generados durante el desarrollo de la aplicación, se deberán copiar en la carpeta /usr/local/apache2/htdocs que es la carpeta que destina la aplicación para guardar las páginas Web a publicar. Para esto se creará una carpeta llamada sistema en donde se introducirán los archivos con extensión .php que contienen el código fuente del sistema:

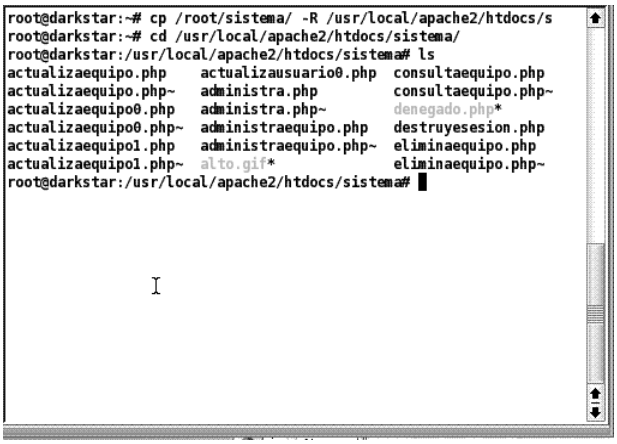


FIGURA 6.13 Implementación del programa en el servidor Apache.

Finalmente, para acceder al sistema de cualquier computadora de la intranet de la empresa, bastará con teclear en un navegador:
http://10.73.1.3/sistema/administra.php[1]

[1] Los dígitos 10.73.1.3, es la dirección IP del servidor, y la ruta sistema/administra.php es el archivo de entrada al sistema, es decir la página de validación de usuario.

CAPÍTULO VII
SEGURIDAD EN EL SERVIDOR – FIREWALL

CAPÍTULO VII SEGURIDAD EN EL SERVIDOR – FIREWALL

7.1 Seguridad en Internet

La seguridad ha sido el principal concerniente a tratar cuando una organización desea conectar su red privada al Internet. Sin tomar en cuenta el tipo de negocios, se ha incrementado el número de usuarios de redes privadas por la demanda del acceso a los servicios de Internet tal es el caso del World Wide Web (WWW), Internet Mail (e-mail), Telnet, y File Transfer Protocol (FTP). Adicionalmente los corporativos buscan las ventajas que ofrecen las páginas en el WWW y los servidores FTP de acceso público en el Internet.

Los administradores de red tienen que incrementar todo lo concerniente a la seguridad de sus sistemas, debido a que se expone la organización privada de sus datos así como la infraestructura de su red a los Expertos de Internet (Internet Crakers). Para superar estos temores y proveer el nivel de protección requerida, la organización necesita seguir una política de seguridad para prevenir el acceso no-autorizado de usuarios a los recursos propios de la red privada, y protegerse contra la exportación privada de información. Todavía, aun si una organización no esta conectada al Internet, esta debería establecer una política de seguridad interna para administrar el acceso de usuarios a porciones de red y proteger sensitivamente la información secreta.

7.2 Firewalls

Un Firewall en Internet es un sistema o grupo de sistemas que impone una política de seguridad entre la organización de red privada y el Internet. El firewall determina cual de los servicios de red pueden ser accesados dentro de esta por los que están fuera, es decir quien puede entrar para utilizar los recursos de red pertenecientes a la organización. Para que un firewall sea efectivo, todo tráfico de información a través del Internet deberá pasar a través del mismo donde podrá ser inspeccionada la información. El firewall podrá únicamente autorizar el paso del tráfico, y el mismo podrá ser inmune a la penetración. Desafortunadamente, este sistema no puede ofrecer protección alguna una vez que el agresor lo traspasa o permanece entorno a éste.

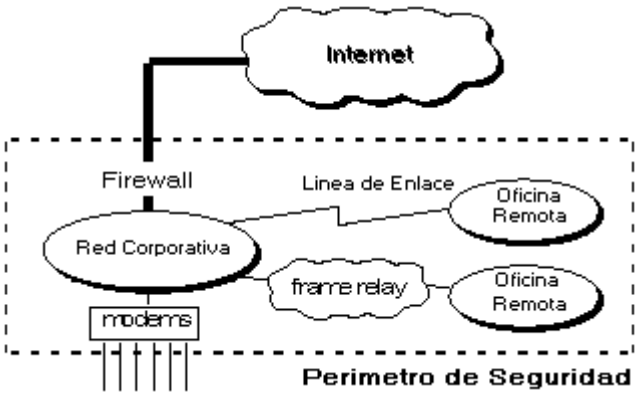


FIGURA 7.1 Diagrama general de un Firewall.

La Política De Seguridad Crea Un Perímetro De Defensa. Esto es importante, ya que debemos de notar que un firewall de Internet no es justamente un ruteador, un servidor de defensa, o una combinación de elementos que proveen seguridad para la red. El firewall es parte de una política de seguridad completa que crea un perímetro de defensa diseñada para proteger las fuentes de información. Esta política de seguridad podrá incluir publicaciones con las guías de ayuda donde se informe a los usuarios de sus responsabilidades, normas de acceso a la red, política de servicios en la red, política de autenticidad en acceso remoto o local a usuarios propios de la red, normas de dial-in y dial-out, reglas de encriptación de datos y discos, normas de protección de virus, y entrenamiento. Todos los puntos potenciales de

ataque en la red podrán ser protegidos con el mismo nivel de seguridad. Un firewall de Internet sin una política de seguridad comprensiva es como poner una puerta de acero en una tienda.

7.3 Beneficios de un firewall en Internet

Los firewalls en Internet administran los accesos posibles del Internet a la red privada. Sin un firewall, cada uno de los servidores propios del sistema se exponen al ataque de otros servidores en el Internet. Esto significa que la seguridad en la red privada depende de la "Dureza" con que cada uno de los servidores cuenta y es únicamente seguro tanto como la seguridad en la fragilidad posible del sistema.

El firewall permite al administrador de la red definir un "choke point" (envudo), manteniendo al margen los usuarios no-autorizados (tal, como., hackers, crakers, vándalos, y espías) fuera de la red, prohibiendo potencialmente la entrada o salida al vulnerar los servicios de la red, y proporcionar la protección para varios tipos de ataques posibles. Uno de los beneficios clave de un firewall en Internet es que ayuda a simplificar los trabajos de administración, una vez que se consolida la seguridad en el sistema firewall, es mejor que distribuirla en cada uno de los servidores que integran nuestra red privada.

El firewall ofrece un punto donde la seguridad puede ser monitoreada y si aparece alguna actividad sospechosa, este generara una alarma ante la posibilidad de que ocurra un ataque, o suceda algún problema en el tránsito de los datos

Un firewall de Internet es el punto perfecto para auditar o registrar el uso del Internet. Esto permite al administrador de red justificar el gasto que implica la conexión al Internet, localizando con precisión los cuellos de botella potenciales del ancho de banda, y promueve el método de cargo a los departamentos dentro del modelo de finanzas de la organización.

Un firewall de Internet ofrece un punto de reunión para la organización. Si una de sus metas es proporcionar y entregar servicios información a consumidores, el firewall de Internet es ideal para desplegar servidores WWW y FTP.

Finalmente, el firewall puede presentar los problemas que genera un punto de falla simple. Enfatizando si este punto de falla se presenta en la conexión al Internet, aún así la red interna de la organización puede seguir operando - únicamente el acceso al Internet esta perdido - .

7.4 Limitaciones de un firewall

Un firewall no puede protegerse contra aquellos ataques que se efectúen fuera de su punto de operación. Por ejemplo, si existe una conexión dial-out sin restricciones que permita entrar a nuestra red protegida, el usuario puede hacer una conexión SLIP o PPP al Internet. Los usuarios con sentido común suelen "irritarse" cuando se requiere una autenticación adicional requerida por un Firewall Proxy server (FPS) lo cual puede ser provocado por un sistema de seguridad circunvecino que esta incluido en una conexión directa SLIP o PPP del ISP.

Este tipo de conexiones derivan la seguridad provista por firewall construido cuidadosamente, creando una puerta de ataque. Los usuarios pueden estar consientes de que este tipo de conexiones no son permitidas como parte de integral de la arquitectura de la seguridad en la organización.

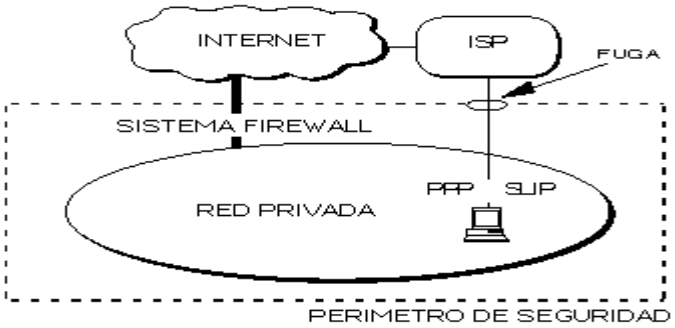


FIGURA 7.2 Diagrama de vulnerabilidad de un Firewall.

El firewall no puede protegerse de las amenazas a que esta sometido por traidores o usuarios inconscientes. El firewall no puede prohibir que los traidores o espías corporativos copien datos sensitivos en disquetes o tarjetas PCMCIA y substraigan estas del edificio.

El firewall no puede proteger contra los ataques de la "Ingeniería Social", por ejemplo un Hacker que pretende ser un supervisor o un nuevo empleado despistado, persuade al menos sofisticado de los usuarios a que le permita usar su contraseña al servidor del corporativo o que le permita el acceso "temporal" a la red.

Para controlar estas situaciones, los empleados deberían ser educados acerca de los varios tipos de ataque social que pueden suceder, y a cambiar sus contraseñas si es necesario periódicamente.

El firewall no puede protegerse contra los ataques posibles a la red interna por virus informativos a través de archivos y software. Obtenidos del Internet por sistemas operativos al momento de comprimir o descomprimir archivos binarios, el firewall de Internet no puede contar con un sistema preciso de SCAN para cada tipo de virus que se puedan presentar en los archivos que pasan a través de él.

La solución real esta en que la organización debe ser consciente en instalar software anti-viral en cada despacho para protegerse de los virus que llegan por medio de disquetes o cualquier otra fuente.

Finalmente, el firewall de Internet no puede protegerse contra los ataques posibles en la transferencia de datos, estos ocurren cuando aparentemente datos inocuos son enviados o copiados a un servidor interno y son ejecutados despachando un ataque.

7.5 Políticas del firewall

Las posturas del sistema firewall describen la filosofía fundamental de la seguridad en la organización. Estas son dos posturas diametralmente opuestas que la política de un firewall de Internet puede tomar:

"No todo lo específicamente permitido esta prohibido"
"Ni todo lo específicamente prohibido esta permitido"

La primera postura asume que un firewall puede obstruir todo el tráfico y cada uno de los servicios o aplicaciones deseadas necesariamente para ser implementadas básicamente caso por caso. Esta propuesta es recomendada únicamente a un limitado número de servicios soportados cuidadosamente seleccionados en un servidor. La desventaja es que el punto de vista de "seguridad" es más importante que facilitar el uso de los servicios y estas limitantes numeran las opciones disponibles para los usuarios de la comunidad. Esta propuesta se basa en una filosofía conservadora donde se desconocen las causas acerca de los que tienen la habilidad para conocerlas.

La segunda postura asume que el firewall puede desplazar todo el tráfico y que cada servicio potencialmente peligroso necesitara ser aislado básicamente caso por caso. Esta propuesta crea ambientes más flexibles al disponer más servicios para los usuarios de la comunidad. La desventaja de esta postura se basa en la importancia de "facilitar el uso" que la propia seguridad del sistema. También además, el administrador de la red esta en su lugar de incrementar la seguridad en el sistema conforme crece la red. Desigual a la primer propuesta, esta postura esta basada en la generalidad de conocer las causas acerca de los que no tienen la habilidad para conocerlas

7.6 Tipos de Cortafuegos

Hay dos tipos de cortafuegos.

1. Cortafuegos de filtrado de paquetes - que evitará el acceso no autorizado a determinados paquetes de la red.

2. Servidores Proxy (a veces llamados cortafuegos) - encargados de establecer las conexiones a la red.

7.6.1 Cortafuegos de Filtrado de Paquetes

El Filtrado de Paquetes es el tipo de cortafuegos integrado en el núcleo de Linux.
Un cortafuegos de filtrado trabaja a nivel de red. Los datos salen del sistema sólo si las reglas del cortafuegos se lo permiten. Cuando los paquetes llegan son filtrados atendiendo al protocolo utilizado, la dirección fuente y destino, y la información que sobre el puerto viene contenida en cada paquete. Muchos encaminadores o routers de red tienen la posibilidad de desarrollar servicios cortafuegos. Los cortafuegos de filtrado nos los podemos imaginar como un tipo de encaminador. Por este motivo se necesita tener un profundo conocimiento de la estructura de los paquetes IP para trabajar con uno.

Puesto que son muy pocos los datos que se analizan y registran, los cortafuegos de filtrado de paquetes requieren menos CPU y crean menos latencia en la red.

Los cortafuegos de filtrado no prevén los controles mediante el uso de contraseña. Los usuarios no pueden identificarse. Lo único que identifica a un usuario es el número IP asignado a su estación de trabajo.
Los cortafuegos de filtrado resultan más transparentes para el usuario, que no tiene que establecer reglas en sus aplicaciones para acceder a Internet. No sucede lo mismo con la mayoría de los servidores proxy. Para la implementación en el servidor de la empresa se utilizará este tipo de cortafuegos.

7.6.2 Servidores proxy

Este tipo de servidores se usa principalmente para controlar, o supervisar, el tráfico hacia el exterior.
Algunos proxy de aplicación almacenan en una memoria de almacenamiento intermedio una copia local
de los datos solicitados. Esto reduce el ancho de banda preciso y acelera el acceso a los mismos datos
para el siguiente usuario. Ofrece una inequívoca prueba de lo que fue transferido.
Existen dos tipos de servidores proxy
1. Servidores proxy de aplicación - son los que hacen el trabajo por Usted.
2. Servidores proxy SOCKS - establecen conexiones entre puertos.

7.7 Introducción a *iptables*

Iptables es como se conoce al módulo *Netfilter*, la herramienta estándar actual de cortafuegos bajo el sistema operativo estándar Linux de cortafuegos.

La primera pila IP para Linux la desarrolló Ross Biro (NET-1). Al mismo tiempo, Orest Zborowski y Laurence Culthane trabajaban en la API sockets y en los controladores SLIP. Sin embargo, la verdadera integración de Linux a la red llegó de la mano de Alan Cox con NET-2 y NET-3. Esta última, es la actual pila de protocolos TCP/IP en la que además de Cox participaron muchos otros como Donnald Becker.

El sistema operativo Linux, ha contado con herramientas de filtrado de paquetes (IPFW) incorporadas en su núcleo desde la versión del kernel 1.1. Esta primera versión con filtrado, contaba con una adaptación de la herramienta *ipfw* del sistema operativo BSD llevada a cabo por Alan Cox por el año 94. El holandés Jos Vos junto a otras personas, mejoró el sistema de filtrado para las series 2.0 del kernel, e introdujo la utilidad de configuración *ipfwadm*.

A mediados de 1998, de la mano de Michael Neuling y Rusty Russell aparece la herramienta *ipchains*, incorporada en los kernels de la serie 2.2 y que todavía hoy es utilizada en gran parte de los sistemas Linux, aunque sólo se asegurará su compatibilidad en el núcleo hasta el año 2003.

A mediados de 1999 Rusty Russell aparece en escena con una nueva herramienta de filtrado *iptables*. Como lo fue ipchains sobre ipfw, iptables es una modificación que permite la construcción de reglas más precisas y un mejor aprovechamiento de los recursos.

Además de realizar un mejor aprovechamiento de los recursos del sistema, la principal característica del módulo *netfilter-iptables,* es la integración de las herramientas de filtrado (el cortafuegos propiamente dicho), de *NAT* y de manipulación (*MANGLING*). NAT es el acrónimo de *Network Addres Translation*. Lo que hace NAT básicamente es alterar las cabeceras de los paquetes IP, principalmente las direcciones) y mantener un "registro de entrada y salida" de los paquetes modificados, para poder alterar los paquetes respuesta de igual forma. Las aplicaciones de NAT son muchísimas, aunque actualmente, la aplicación más conocida del NAT, es lo que se conoce como enmascaramiento o *masquerading*. El enmascaramiento, se utiliza principalmente para dos cosas:

- *La conexión de varios equipos a través de una sola dirección IP.* Como ejemplos, las pequeñas LAN domésticas, un CyberCafé, y en general cualquier red con más equipos que IPs. En lugar de instalar un servidor proxy y configurar las distintas aplicaciones para que hagan uso del mismo, se instala NAT, y no hay necesidad de configurar las aplicaciones, ya que al trabajar en un nivel más bajo de la pila, resulta totalmente transparente.

No debe confundirse el NAT con un "proxy", aunque tengan aplicaciones parecidas. NAT no es un proxy, los proxies trabajan en un nivel superior de la pila de protocolos, bien en el nivel de TCP/UDP o incluso en el nivel de aplicación, lo que implica que los clientes deben configurarse para hacer uso del servicio. Por el contrario, NAT, al igual que el filtrado, trabaja en un nivel más bajo, a nivel IP, por lo que es "transparente". ¿Por qué se confunden muchas veces el NAT y un proxy? La respuesta puede deberse a que tanto el uso de un proxy como el uso de NAT se emplean actualmente [1] para "compartir una conexión a Internet".

El filtrado de paquetes y *NAT* están ampliamente ligados, ya que comos ya hemos visto, ambos servicios constituyen el módulo Netfilter y se configuran haciendo uso de la herramienta *iptables*. Antes de *iptables*, la herramienta Linux utilizada para configurar y ofrecer servicios NAT era *ipmasqadm*. A diferencia de *ipchains / iptables* cuyo manejo es conceptualmente similar, entre *ipmasqadm* e *iptables*, existen diferencias de uso.

[1] Tanto la definición de Proxy como el protocolo NAT, contemplan muchas más posibilidades que el simple "compartir" de la conexión a Internet. Pero, un usuario normal de un PC, es lo único con lo que suele asociarlos, si es que ha oído hablar de ellos.

7.8 Instalación de *iptables*

Lo primero que debemos hacer para instalar iptables, es conseguirlo. Podemos descargar iptables de *http://www.netfilter.org/.* En cualquier distribución actual se instala por defecto y el kernel "precompilado" soporta la configuración, a pesar de eso, vamos a describir brevemente el proceso necesario para una instalación.

7.8.1 Parámetros del kernel

Como ya hemos dicho, para poder utilizar iptables, necesitamos un kernel superior a la versión 2.3.15. Para configurarlo correctamente deberemos recompilar el mismo (*make config*) e incluir distintas opciones dentro del kernel, o bien habilitar la posibilidad de que puedan cargarse posteriormente como módulos.

7.8.2 Instalación de iptables.

Lo primero que debemos hacer tras descargar el paquete iptables, es descomprimirlo. Para descomprimirla, utilizaremos la orden:

bzip2 -cd iptables-1.2.5.tar.bz2 | tar -xvf

Con ello, tendremos el paquete descomprimido en un directorio iptables-1.2.3. Posteriormente se siguen los siguientes pasos:

cd /root/iptables-1.2.2

make KERNEL_DIR=/usr/src/linux BINDIR=/usr/bin LIBDIR=/usr/lib MANDIR=/usr/man

make install KERNEL_DIR=/usr/src/linux BINDIR=/usr/bin LIBDIR=/usr/lib MANDIR=/usr/man

Finalmente, para arrancar el servicio iptables en el arranque del sistema ejecutaremos el comando siguiente:

hkconfig --level 235 iptables on; service iptables start

Naturalmente, no hay ninguna regla activa. Las reglas creadas con el comando iptables15 se almacenan solamente en RAM, es decir, que si reiniciamos el sistema tras haber configurado varias reglas de iptables, éstas se perderán y tendremos que volver a teclearlas. Para incluir las reglas al inicio del sistema, podemos hacer varias cosas. La primera, editar el archivo script /etc/rc.d/init.d/iptables, éste script se ejecutará cada vez que se inicie el servicio iptables, que hemos configurado antes para iniciarse con el sistema. Otra opción, es introducir las reglas mediante el comando iptables y cuando el cortafuegos funcione como es debido, salvarlas en el fichero /etc/sysconfig/iptables. Es decir, introducimos las reglas
de filtrado, y cuando el cortafuego funcione como esperamos, ejecutamos: /sbin/service iptables save

Esto hace que el script de inicio de iptables (rc.d) ejecute el programa /sbin/iptables-save y escriba la configuración actual de iptables en el fichero /etc/sysconfig/iptables. Este fichero debería ser de sólo lectura para el usuario root, para que las reglas de filtrado de paquetes no sean visibles por el resto de los usuarios. La próxima vez que se inicie el sistema, el script de inicio de iptables volverá a aplicar las reglas guardadas en /etc/sysconfig/iptables usando el comando /sbin/iptables-restore.

7.8.3 Estructura y funcionamiento de iptables.

Comenté anteriormente, que el módulo netfilter, integraba tres posibilidades en el manejo de los paquetes, cada una de esas posibilidades, se corresponde con una tabla donde se aplican las reglas. Con la opción iptables –t *tabla*, especificamos la tabla sobre la que

queremos trabajar. Estas tablas son *filter*, *nat* y *mangling*. Veamos que podemos hacer sobre cada una de ellas:

- *nat:* La tabla *nat* se utiliza para configurar el protocolo de Network Address Translation. Cuando un flujo de paquetes (una conexión TCP) atraviesa la tabla, el primer paquete es admitido, el resto, son automáticamente identificados como parte del flujo de ese primer paquete y de manera automática se llevan a cabo sobre ellos las operaciones NAT o de enmascaramiento. Esta es la razón por la cual, no se lleva a cabo ningún tipo de filtrado en esta tabla. La tabla de *nat* tiene tres *chains* o cadenas sobre las que podemos añadir reglas. La cadena PREROUTING se utiliza para alterar los paquetes tan pronto llegan al cortafuegos (DNAT o NAT del destino). La cadena OUTPUT, se utiliza para alterar los paquetes generados localmente en el cortafuegos, antes de tomar ninguna decisión de enrutado. Finalmente tenemos la cadena POSTROUTING para alterar los paquetes que acaban de dejar el cortafuegos (SNAT o NAT en el origen).

- *mangle:* La tabla de *mangling* o "manipulación", permite manipular otros elementos de los paquetes, como el TTL, el TOS, etc…, ha excepción del NAT, que se realiza en la otra tabla. La funcionalidad de esta tabla está en expansión y aunque potencialmente puede ser muy valiosa, no tiene demasiada utilidad (salvo a hackers). Consta de dos cadenas, PREROUTING y OUTPUT.

filter: Esta es la reina de la casa. En la tabla *filter*, se llevan a cabo la funcionalidad principal de iptables, el filtrado de paquetes. Como las anteriores, consta de varias chains predefinidas, en este caso INPUT, FORWARD y OUTPUT. La primera hace referencia a los paquetes entrantes cuyo destino es el propio cortafuegos. La segunda se emplea para decidir que hacer con los paquetes que llegan al cortafuegos y tienen como destino otro host, así podemos decidir si encaminarlos o no. Por último, la cadena OUTPUT se utiliza para filtrar paquetes generados en el propio host con destinos externos. Cuando un paquete entra en el cortafuegos, lo hace a través de alguna interfaz (tarjeta de red, módem, etc,). El paquete se dirige al Kernel, entrando en las distintas cadenas de las tablas sólo si procede.

En la figura que se ofrece posteriormente, puede observarse el esquema general del procesado de paquetes en iptables. En la tabla siguiente podemos ver también ejemplos de las desventuras de los paquetes en su tránsito por el módulo netfilter del kernel.

Paso	Tabla	Cadena	Comentario
1			En el aire, por ejemplo Internet
2			Llega a la interfaz de red, por ejemplo eth0
3	MANGLE	PREROUTING	Si casa con alguna de las reglas de *mangling*, se actúa según indique la acción.
4	NAT	PREROUTING	Si casa con alguna de las reglas de *NAT*, se actúa según indique la acción.
5			Decisión de enrutado
6	FILTER	INPUT	Se procede a realizar el filtrado del tráfico con destino local.
7			Aplicación local (cliente o servidor)

FIGURA 7.3 Procesado de paquetes en iptables.

7.8.4 El comando iptables

En este apartado, presentaré el uso del comando iptables, que es la herramienta para crear las reglas de nuestro cortafuegos.

Como ya se había mencionado, iptables, funciona mediante tres tablas, a su vez, cada una de esas tablas, tiene definidas unas "*chains*" o cadenas. Cada una de estas chains se compone de una lista de reglas de filtrado. Cada regla no es más que un par condición/acción sobre atributos del paquete IP. El paquete, irá pasando secuencialmente por cada una de las reglas de la *chain* hasta encajar en el patrón de alguna de ellas. Cuando esto ocurra, el paquete se tratará según indique la acción de la regla con cuya condición el paquete ha hecho matching[2]. Si tras recorrer toda la lista, el paquete no encaja con ninguna de las reglas, se ejecutará la acción por defecto asociada a esa *chain*.

Ahora, podemos adentrarnos ligeramente en las opciones de la herramienta *iptables*.
Primero, veremos como manipular las *chains*. La utilización del comando, presenta siempre el siguiente patrón:
iptables [-t tabla] comando [match] [objetivos/saltos]

Las tablas son siempre una de las tres siguientes *filter*, *nat*, o *forward.* Si no se indica tabla alguna, por defecto, nos referimos a la tabla *filter*. Para añadir y manipular cadenas utilizaremos siempre los comandos siguientes:

- *iptables –N*: Crear una nueva chain o cadena vacía (sin reglas).
- *iptables –X*: Elimina una chain que esté vacía (a excepción de las tres internas)
- *iptables –F*: Vacía una chain. Es decir, elimina todas las reglas de una chain.
- *iptables –P*: Cambia la política por defecto de una chain.
- *iptables –L*: Lista las reglas de una chain.
- *iptables –Z*: Pone a cero las variables de auditoría de una chain (número de paquetes, de bytes, etc…)

Con los comandos siguientes conseguimos manejar las reglas de esas cadenas:

- *iptables –A*: Inserta al final de una cadena una nueva regla.
- *iptables –I*: Inserta al comienzo de una chain una nueva regla.
- *iptables –R*: Reemplaza una regla de una chain.
- *iptables –D*: Elimina una regla de una chain (podemos indicar el orden o su condición).

Por último, sólo queda mostrar como podemos establecer las condiciones y acciones sobre cada regla, es decir como establecer las condiciones de *match* entre paquetes y reglas:

- *iptables –s*: Indica un dominio o IP (rango de IPs) de origen sobre el que se evalúa la condición de la regla.
- *iptables –d*: Indica un dominio o IP (rango de IPs) de destino el que se evalúa la condición de la regla.
- *iptables –i (--in-interface)*: Indica la interfaz de entrada sobre la cual se evalúa la condición de la regla.
- *iptables –o (--out-interface)*: Indica la interfaz de salida sobre la cual se evalúa la condición de la regla.
- *iptables –p*: Especifica el protocolo del datagrama que concordará con esta regla.

Los nombres válidos de protocolos son *tcp, udp, icmp*, o un número, si se conoce el número del protocolo de IP. Cada protocolo lleva asociados sus propios modificadores a través de las extensiones correspondientes:

[2] El recorrido por la lista de reglas es secuencial, por lo que es muy importante el orden en el cual coloquemos las reglas. Haremos hincapié en ello más adelante.

- **Extensiones de TCP:**

    - *sport*: Especifica el puerto que debe utilizar el origen del datagrama para concordar con esta regla. Se pueden especificar los puertos en la forma de un rango, especificando los límites inferior y superior con los dos puntos ":" como delimitador. Por ejemplo, 20:25 describe todos los puertos que van desde el 20 hasta el 25 incluyendo ambos. De nuevo, el signo "!" puede utilizarse para negar los valores.
    - *dport*: Igual que la opción anterior pero para el puerto destino.
    - *tcp-flags*: Especifica mediante una máscara si los bits indicadores de TCP del datagrama concuerden con ella. La máscara es una lista separada por comas de los indicadores que deben examinarse en la comprobación.(SYN, ACK, FIN, RST, URG, PSH, ALL o NONE).
    - *syn*: La regla casa con los datagramas cuyo bit SYN valga 1 y cuyos bits ACK y FIN valgan ambos 0. Esta opción es una abreviatura de: --tcp-flags SYN,RST,ACK SYN

- **Extensiones UDP:**

    - *sport*: Como en TCP, especifica el puerto que debe utilizar el origen del datagrama para concordar con esta regla.
    - *dport*: Igual que la opción anterior pero para el puerto destino.

- **Extensiones ICMP:**

    - *icmp-type*: Puede especificarse el tipo de mesaje ICMP tanto por su número como por los siguientes identificadores: echo-request, echoreply, source-quench, time-exceeded, destination-unreachable, network-unreachable, host-unreachable, protocol-unreachable, y port-unreachable.

- **Extensiones MAC:**

    - *mac-source*: Se especifica la dirección MAC (ethernet) Sólo tiene sentido en la cadena INPUT y FORWARD.
    - *iptables –f*: Cuando se fracciona un datagrama porque supera el MTU de la red, podemos utilizar esta opción para especificar acciones sobre el segundo y restantes fragmentos del datagrama.
    - *iptables ¡*: Invierte el valor lógico de la condición de la regla.

Existen más posibilidades de filtrado, y como ya hemos dicho, podemos diseñar las propias por la facilidad de extensión de iptables. Estas extensiones, son librerías compartidas que normalmente están en el directorio /usr/local/lib/iptables/, aunque, según distribuciones, pueden aparecer en /lib/iptables/ o en /usr/lib/iptables/. Sirviéndonos de ellas podemos incluir filtrado sobre multitud de nuevas opciones o realizar acciones tremendamente exóticas con los paquetes filtrados (por ejemplo, el módulo REJECT incluye la nueva acción REJECT, cuyo efecto es similar al de DROP exceptuando los mensajes de error ICMP que si son tratados), incluso podemos listar el número de reglas similares que se dispararán por minuto, para evitar ataques DOS. Como siempre, a medida que las extensiones demuestran su utilidad, se añaden a la distribución estándar de iptables, y si no hay ninguna que nos ofrezca lo que buscamos, podemos hacerla nosotros.

Asumiremos que la red interna es confiable, es decir, que de existir alguna amenaza, ésta vendrá desde Internet, no desde dentro. Tenemos que tener en cuenta que la IP es dinámica, esto puede traernos algunos inconvenientes, en el caso que deseemos (como es nuestro caso, ya que corremos servicios en el cortafuegos) filtrar utilizando nuestra dirección IP pública. Hay diversas soluciones, pero dado que el ISP ofrece cierta "estabilidad" en la duración de las IP. La solución más sencilla, es obtener la IP mediante la ejecución de:

```
EXT_IP=`ifconfig $EXT_IF | grep inet | cut -d : -f 2 | cut -d \ -f 1`
```

O de algún método equivalente como:

```
EXT_IP="`ifconfig $EXT_IF | grep 'inet addr' | awk '{print $2}' | sed -e 's/.*://'`"
```

Resuelto este problema, sólo queda estructurar el script en función de las necesidades de la empresa
.
Para permitir accesos al servidor HTTP, FTP o SSH de *canyizares* desde cualquier lugar, basta con ejecutar los siguientes comandos:

```
#Servicio FTP

iptables -t filter -A INPUT -p TCP -s 0/0 --dport 21 -j allowed

#Servicio SSH

iptables -t filter -A INPUT -p TCP -s 0/0 --dport 22 -j allowed

#Servicio HTTP

iptables -t filter -A INPUT -p TCP -s 0/0 --dport 80 -j allowed


#Servicio HTTP dentro de la LAN. Se exporta al exterior

iptables -t nat -A PREROUTING –i $EXT_IF -p tcp --dport 80 -j \ DNAT --to-destination 192.168.0.2
```

Tras ver la sintaxis de iptables y comprender como es su funcionamiento, el configurar un cortafuegos a medida es una cuestión relativamente sencilla. A media que nos volvemos "paranoicos" o necesitamos una mayor seguridad y a la vez el poder exportar más servicios, e cuando las cosas se van complicando.

Si no se desea una seguridad o perfilado extremo, actualmente existen herramientas gráficas que permiten generar scripts generales de configuración, a pesar de que puede que no excesivamente eficientes, pueden servir de base para "retocarlo" después a manualmente y evitar teclear demasiado.

CONCLUSIONES

Linux se ha convertido en uno de los sistemas operativos más fuertes del mercado, una de las principales ventajas es su costo. Hoy estamos obligados de alguna forma a pagar el alto costo de otras herramientas que se comportan de alguna forma tiranamente, ya que no solo pagamos un precio más que alto, sino que estamos obligados a amoldarnos a estas herramientas de las cuales conocemos muy poco internamente, lo que nos genera un problema a la hora de hacer un seguimiento de nuestros procesos.

Todo citado es lo contrario de Linux, ya que no solo es mucho más barato sino que el producto se amolda a nuestras necesidades, no obligándonos a trabajar con algo que desconocemos, así como también nos ofrece más escalabilidad, estabilidad y robustez.

Hoy Linux no sólo ahorra costos en materia de licencias sino que también en materia de disponibilidad de los sistemas productivos, ya que cuando mayor disponibilidad tenemos y menor cantidad de errores de sistema, incide en un beneficio directo para las empresas que lo implementan.

Con este proyecto se demuestra que existen en el mundo Linux, potentísimas herramientas libres para acometer proyectos de gran envergadura. Se demuestra también que con Linux lo que hay que poner son recursos humanos a la hora de crear nuevos proyectos, sin necesidad de recurrir a caros sistemas propietarios.

Es importante tener en cuenta que la solución propuesta para esta empresa puede no ser una generalidad, debido a que hoy en día existen una infinidad de herramientas tanto de software libre como comercial y el administrador de sistemas tiene la libertad de elegir las que más se adecúen a las necesidades y presupuesto de la empresa.

La situación respecto al soporte del sistema es aún incipiente, pero se está desarrollando.
Sin embargo los profesionales de Linux ven una interesante oportunidad de nuevos negocios ante esta situación.

La posibilidad de optar por el sistema GNU/Linux dentro del sistema empresarial no sólo es una gran oportunidad para el país, sino también una decisión trascendental por los peligros que encierra la dependencia tecnológica de Microsoft. Lo que está en juego es la posibilidad de disponer de la tecnología al servicio de todos, no sólo al servicio de las necesidades económicas de esa gran corporación,

Dentro del universo del software libre existen gran cantidad de paquetes diferentes a los que se utilizaron en este proyecto con los que se puede implementar un sistema semejante a la propuesta presentada.

Por tanto, y de acuerdo a los objetivos planteados en un principio, es posible implementar completamente un servidor con herramientas de software libre, que solvente las necesidades de automatización y control de procesos tecnológicos de una empresa.

Este proyecto me sirvió para fortalecer los conocimientos en cuanto al uso del software libre, así como las debilidades teóricas que poseía antes de concluido el proyecto. Y en consecuencia, mejorar en el desarrollo de la práctica profesional.

BIBLIOGRAFÍA:

- Using Samba. Eckstein Robert, et al. 1ra ed. O'Reillly & et; Associates, 1999. 416 pp. Collection O'Reilly System.
- Configuración de sistemas Linux. Daniel L. Morrill, Ed. Anaya Multimedia, 2002.
- Sitios Web bajo Linux: Usuarios Expertos. Hector Facundo Arena, MP Ediciones, 2001.
- La Biblia de Administración de sistemas Linux. Dee-Ann Leblanc, col. La Biblia de, Ed. Anaya Multimedia.
- Guía Avanzada Firewalls Linux. Robert Ziegler y José Ignacio Sánchez, Prentice Hall PTR, 1.ª edición, 2001.
- De Windows a Linux - Para Distribuciones Red Hat. Michel Martin, Marcombo, 2001.
- Linux. Guía de referencia y aprendizaje. Matt Welsh, Matthias Kalle Dalheimer y Lar Kaufman, col. O'Reilly, Ed. Anaya Multimedia, 2000.
- Linux Facil.. Hector Facundo Arena, MP Ediciones, 2000.
- Linux - Guía del Administrador. Hector Facundo Arena, MP Ediciones, 2000.
- Apuntes del Diplomado en diseño e implementación de sistemas con software libre Linux

DIRECCIONES ELECTRÓNICAS:

- www.apache.org
- www.php.net
- www.mysql.com
- www.linux.org

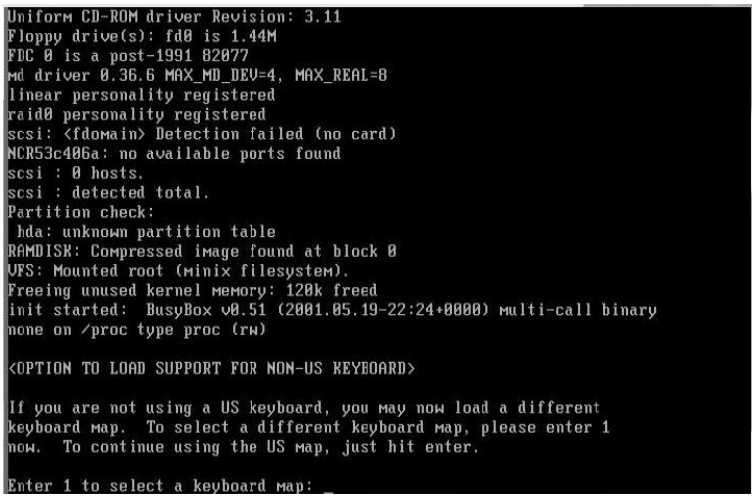
ANEXO I INSTALACIÓN DE LINUX SLAKCWARE

Una vez elegido el método de instalación procedemos a instalar Linux, arrancamos la instalación. Aparecerá una pantalla de bienvenida al programa de instalación de Linux, en la parte inferior de esta pantalla aparecerá el indicador boot.

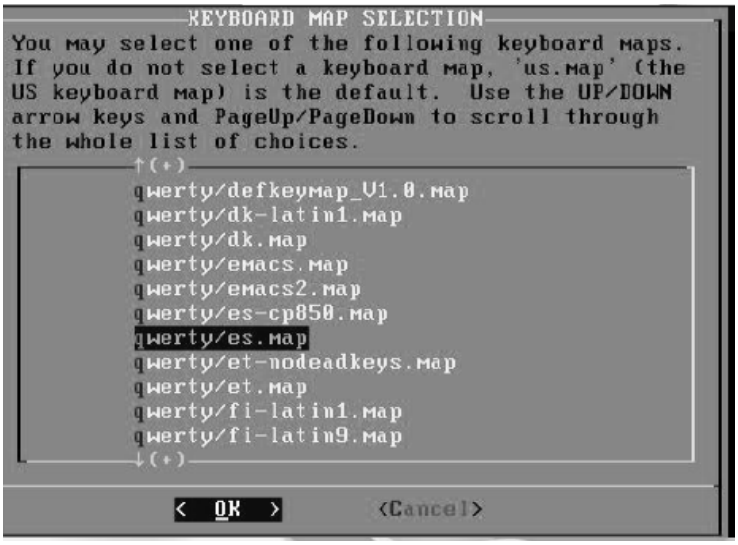
Cuando la instalación se realiza por CD-ROM automáticamente se cargan las imágenes de los discos boot y root. Después de arrancar la instalación Linux detectara la mayor parte del hardware que esta instalado en el equipo.

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

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

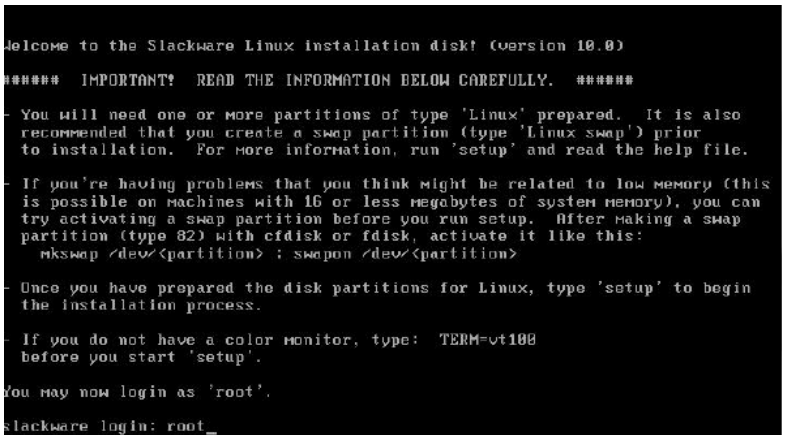


Elegir la opción qwerty/es.ma para el teclado en español.



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

Posteriormente se deberá que registrarse ante el sistema para que nos entregue un Shell y seguir con la instalación, para ello hay que registrarnos como root.



Ahora tenemos que crear las particiones necesarias para la instalación.

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

Tener todo el sistema en una sola partición es mala idea ya que no es flexible y hace difícil de administrar el servidor.

Lo más recomendable es hacer particiones pensando en la carga de archivos del sistema, de los usuarios, servicios y otras consideraciones, logrando con ello facilitar las tareas de respaldo, actualización, administración fácil y eficiente, entre muchas otras ventajas.

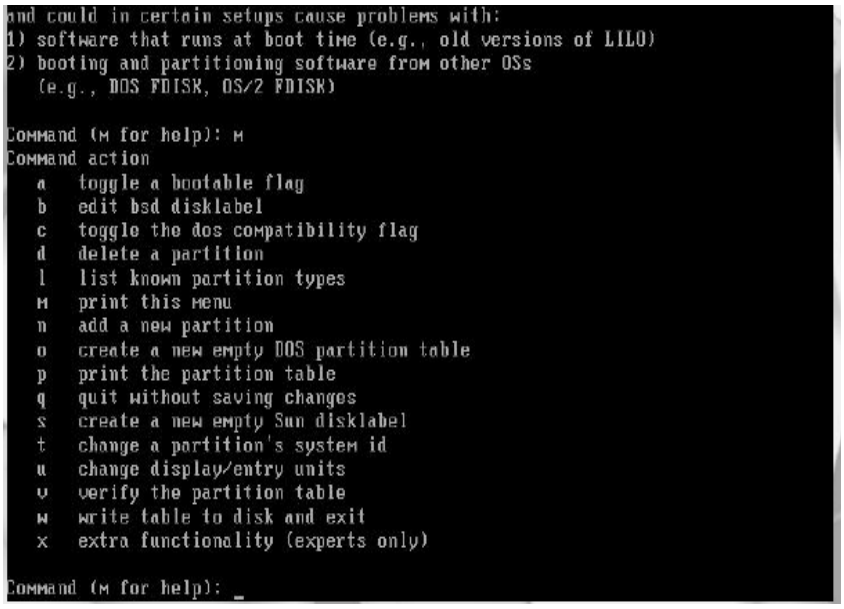
En la mayoría de los casos las particiones de mayor tamaño albergaran los directorios /home y /usr.

Para particionar el disco usamos el comando:

fdisk/dev/hda

Si utilizamos el comando indicándole el disco apropiado nos muestra la siguiente pantalla:

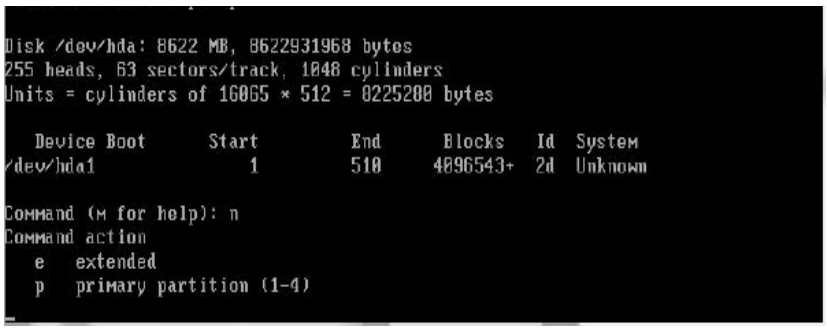
Para apoyarnos con la ayuda de las opciones de fdisk presionamos la letra m consiguiendo el despliegue del siguiente menú:



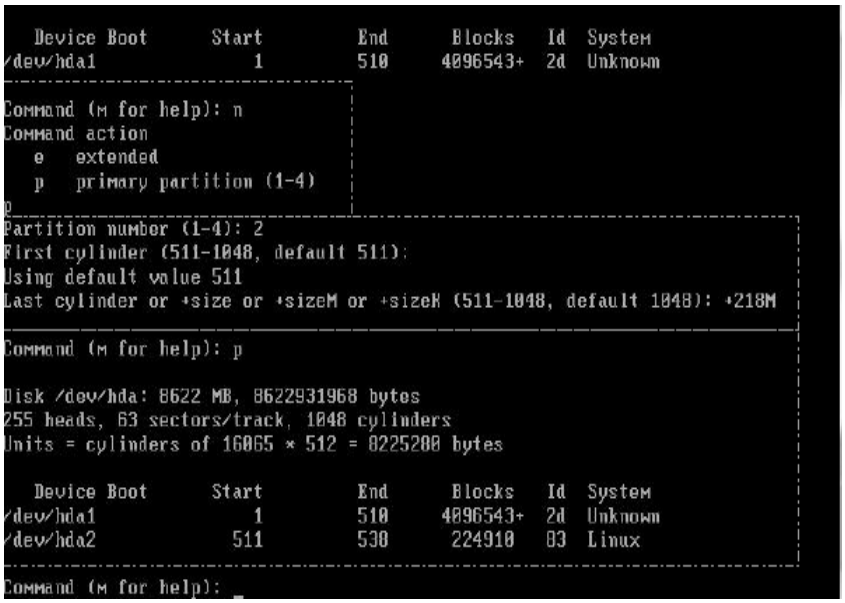
Pensando en una instalación básica se crearan sólo dos particiones para Linux, la swap y la nativa de Linux:

n(crea una nueva partición)

Podemos elegir entre primaria o extendida, aquí vamos a generar una partición primaria:

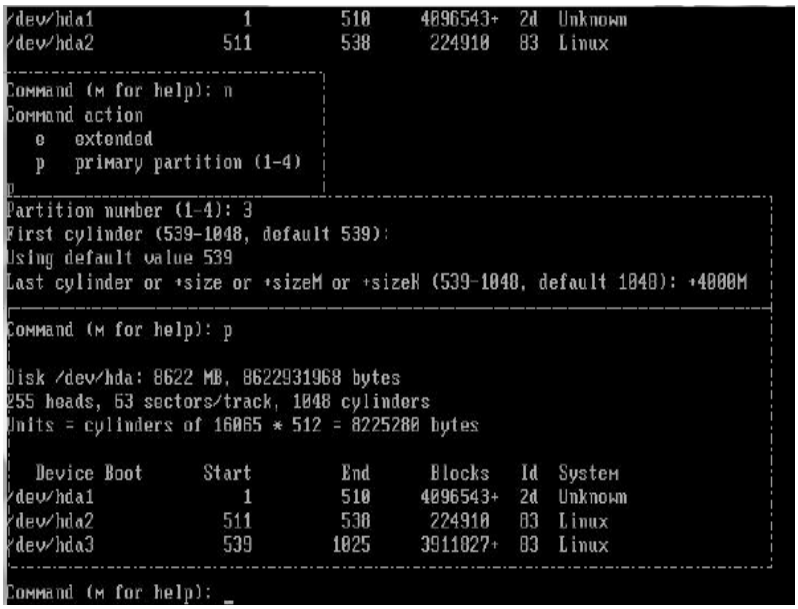


Ahora tenemos que definir el inicio y el final de la partición, por default nos ofrece el primer cilindro disponible del disco, para indicar el final de la partición utilizamos un método de suma de mega bites a partir del punto de inicio de la partición como se muestra en la imagen:



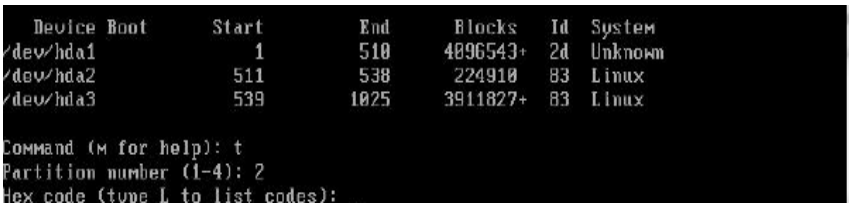
Esta primera partición será utilizada como swap

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



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

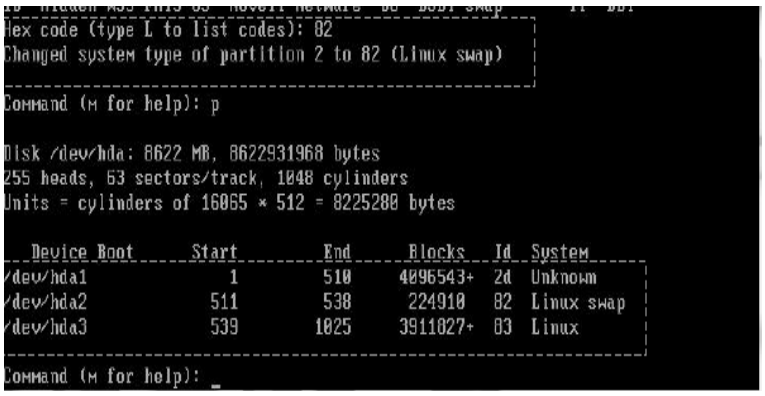
Para cambiar el tipo presionamos la letra t (de tipo) e indicamos al sistema que queremos modificar la partición numero 2.



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

Aquí se muestra el listado donde vemos que el código 82 es el de LinuxSwap

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

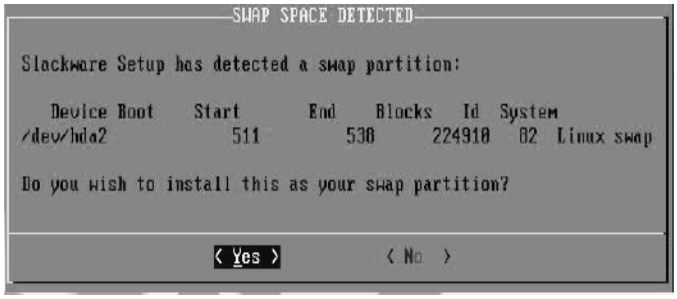


Ahora tenemos que guardar los cambios realizados en las particiones y comenzar la instalación con el comando setup.

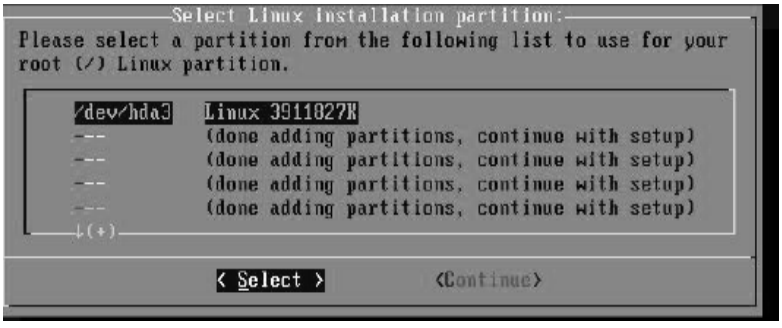
Se despliega el asistente de instalación ahora en formato gráfico en forma de menú:

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

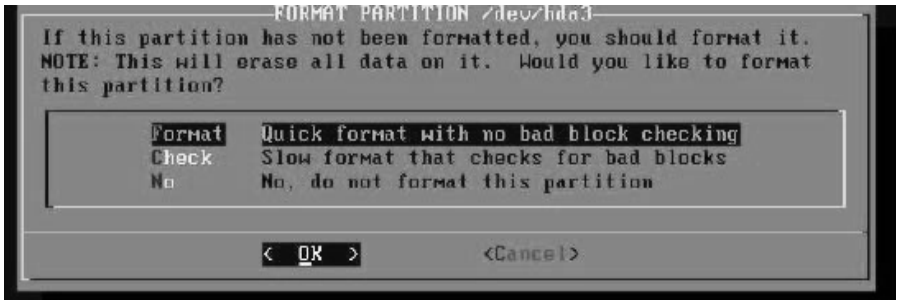
Una vez que se ha definido el tipo de teclado el sistema automáticamente detecta la partición de swap, nos solicita darle formato y la prepara para la instalación:



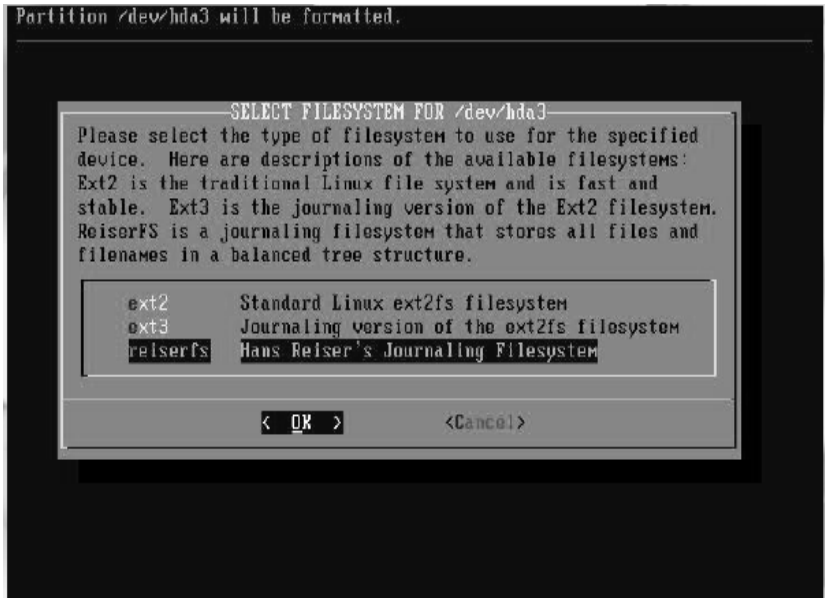
Una vez que ha preparado el área de swap el proceso de instalación reconoce las particiones de Linux, la partición que elijamos en este momento es donde se va a instalar el sistema:



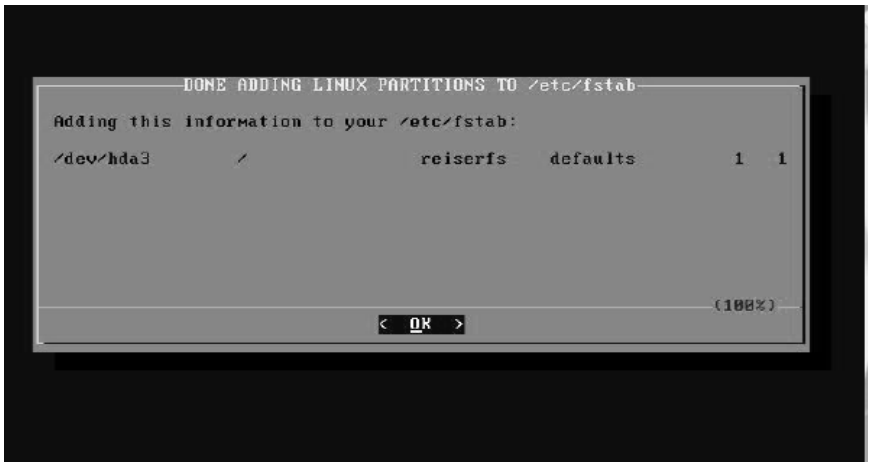
Ya que elegimos la partición, el instalador solicita darle formato rápido, formato con revisión de sectores dañados o montar la partición sin formatear:



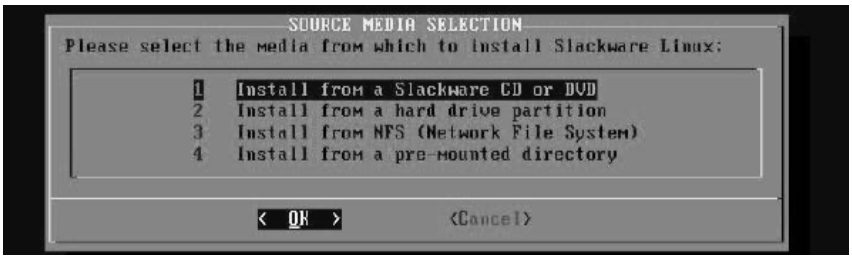
Elegimos la opción de dar formato, nos solicita el tipo de Sistema de Archivos con el cual va a ser formateada la partición.



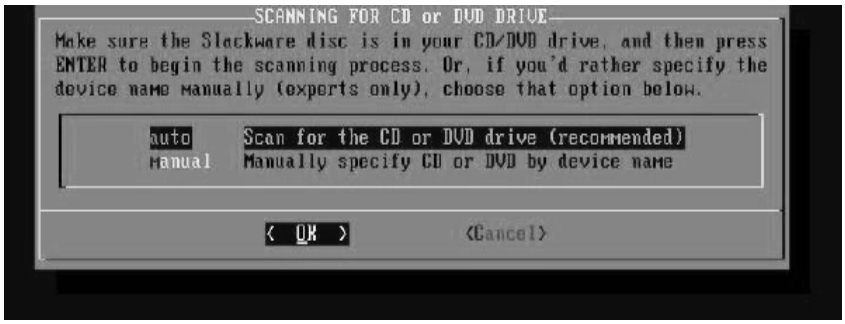
Ya terminado el formato de todas las particiones entrega un reporte con información de la(s) particiones.



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



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

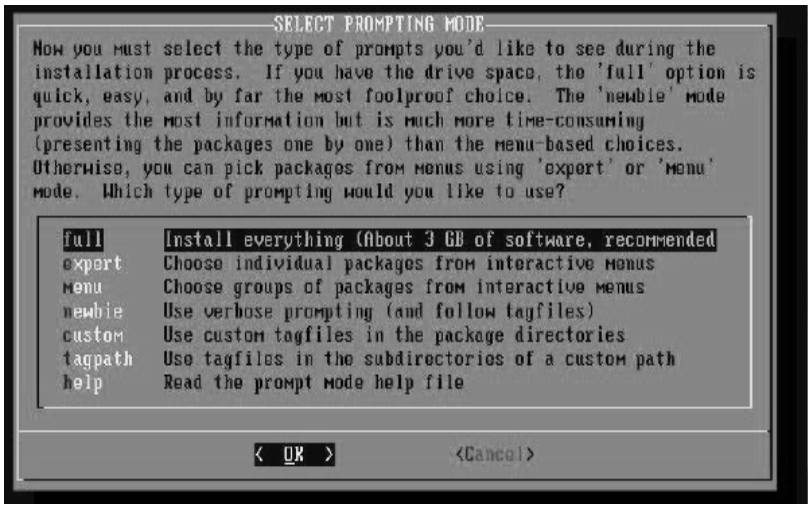


Lo más conveniente es decirle que aplique un montaje del CD en forma automática, ya que de otra forma necesitaríamos saber en que IDE se encuentra conectada la unidad de CD y si esta como maestro o como esclavo, información que sólo la podemos obtener abriendo el CPU o revisando la configuración en el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

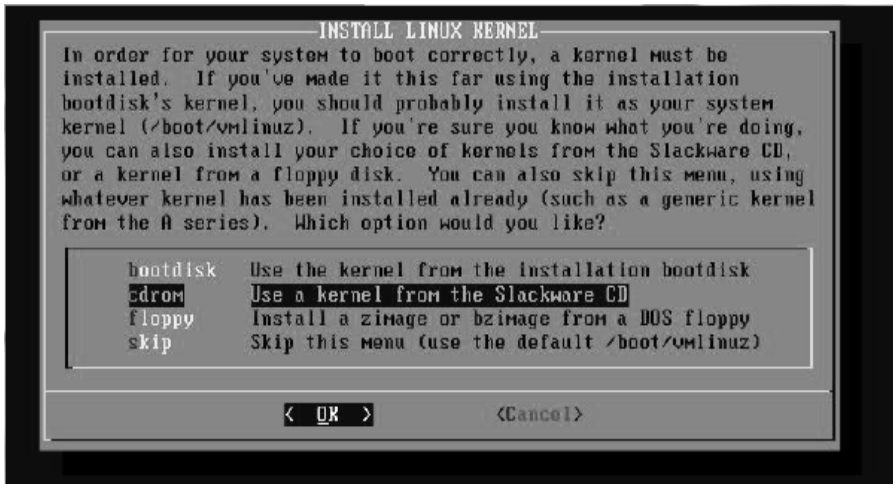
Ya que se ha detectado el CD el asistente nos pedirá elegir las series de paquetes que se van a instalar.

Con la barra espaciadora marcamos o desmarcamos las series de paquetes.

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

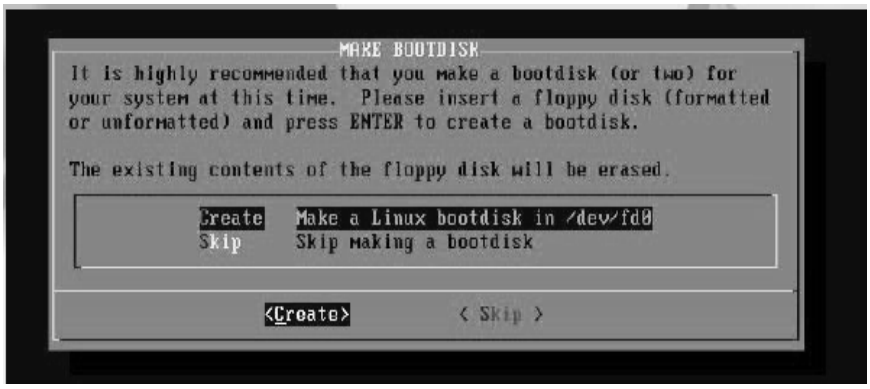


Después de haber terminado la instalación de paquetes continuamos con la configuración del sistema, el siguiente paso es indicarle desde que medio deseamos instalar el Kernel para el sistema, obviamente en este caso es desde CD.

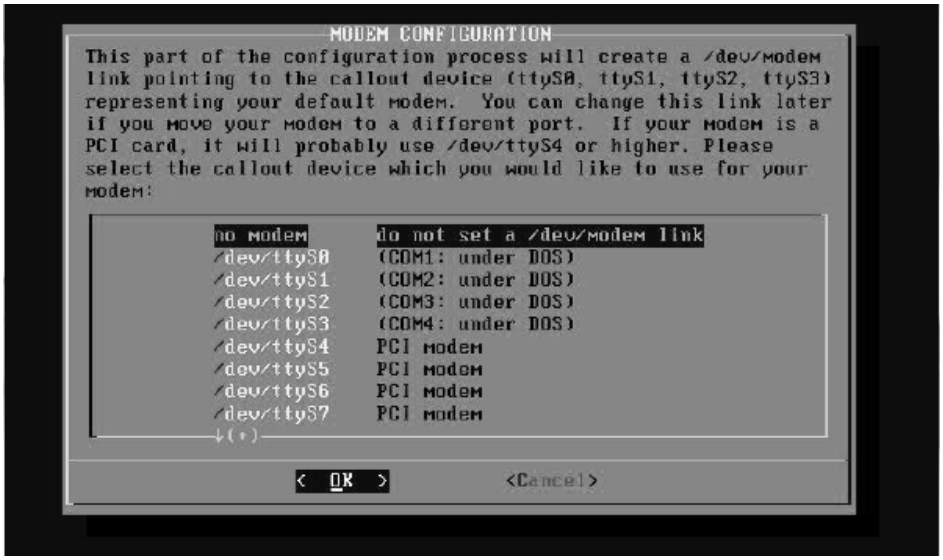


Ahora tenemos que elegir cual kernel es el que vamos a instalar, este depende de la arquitectura y de las características de la computadora donde se esta instalando. El kernel adecuado para las x86 es el bare.i.

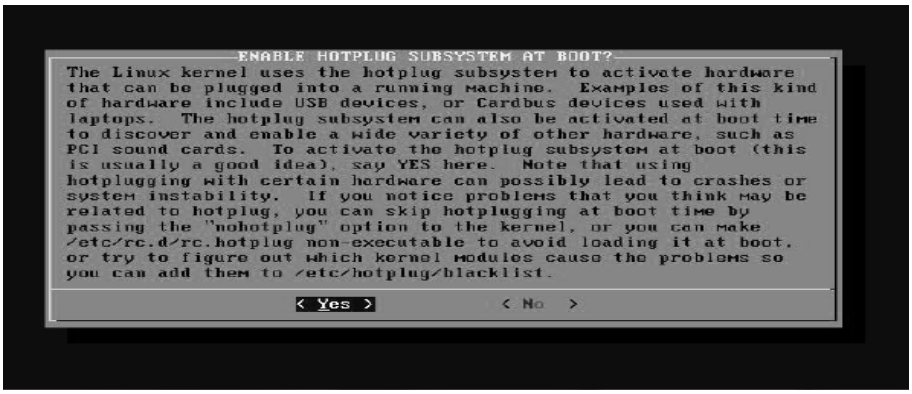
El asistente nos ofrece la posibilidad de hacer un disco de arranque para poder iniciar desde él el sistema, este disco puede servir como de recuperación o de emergencia para recuperar el sistema.



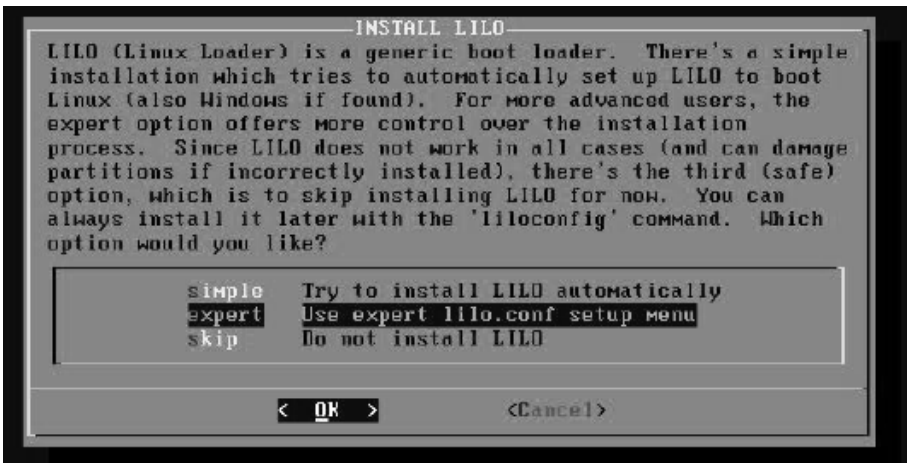
El asistente también nos permite configurar un módem, si es que contamos con uno externo o que se encuentre en una ranura de expansión podremos configurarlo fácilmente, si es un módem integrado a la tarjeta principal generalmente no es detectado ya que pertenece a la familia de los winmodem.



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

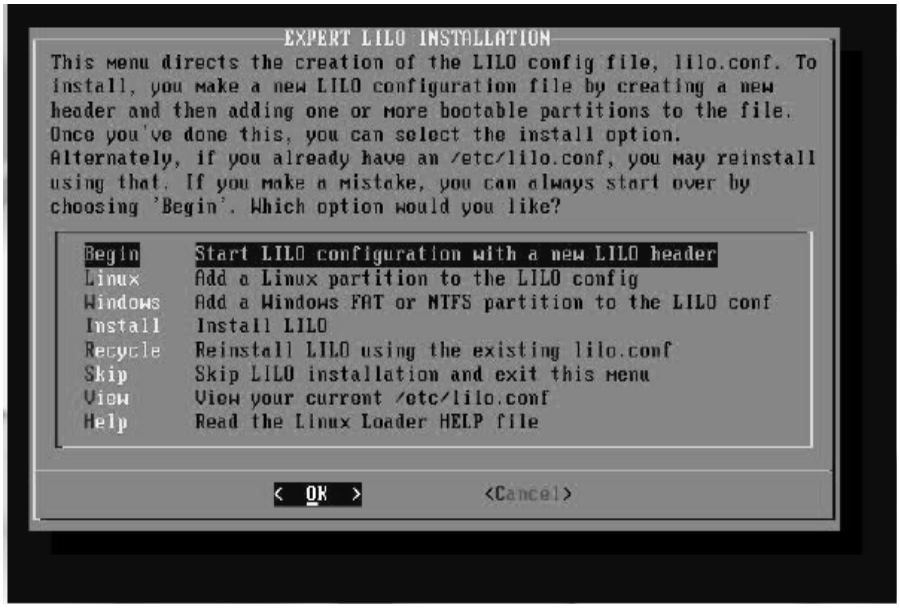


LiLo es el gestor de arranque el cual nos ayuda a generar un menú de inicio para equipos multisistemas, entre otras cosas, como la instalación esta contemplada para una máquina que tiene dos sistemas operativos se realizará la configuración, de este menú elegimos la opción en modo experto.

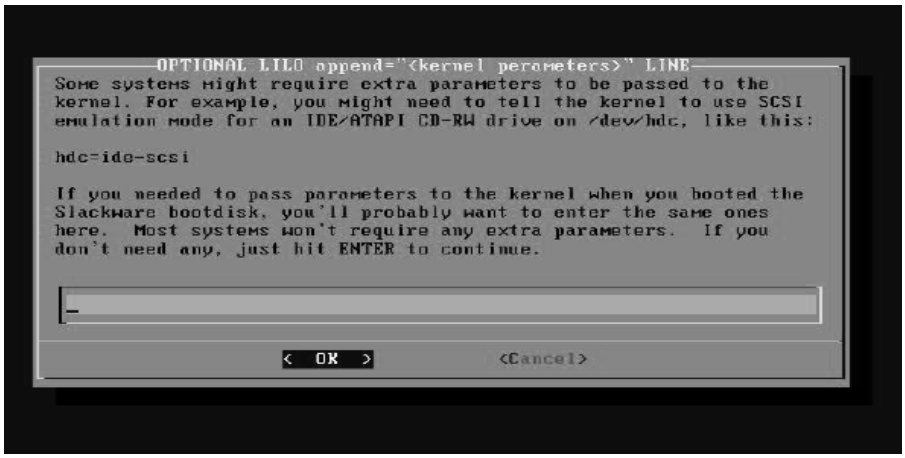


El cual nos lleva al siguiente menú:

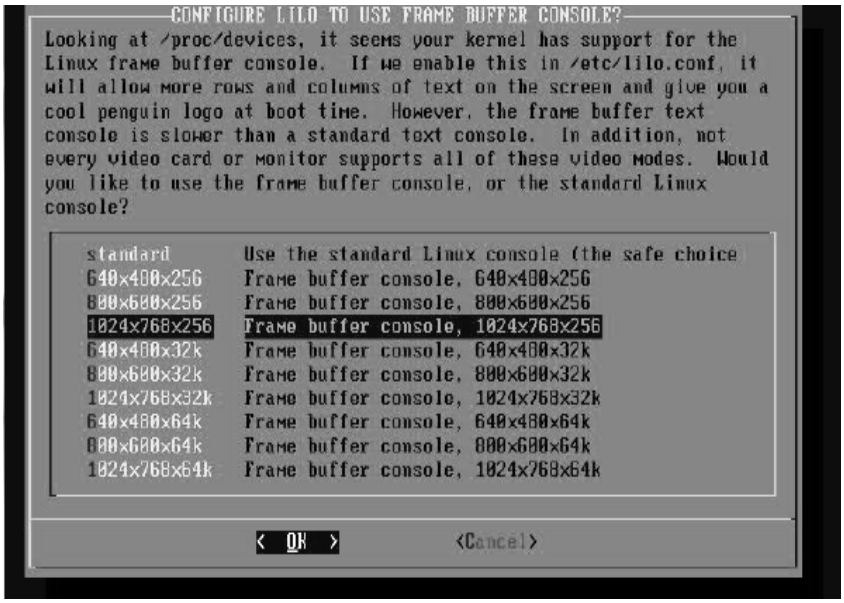
Este menú nos permite iniciar la configuración de LILO con las propiedades principales, para ello tenemos que iniciar con la opción Begin:



Podemos ingresar opciones extra si es que las requerimos, la mayoría de los sistemas no requieren opciones extra.



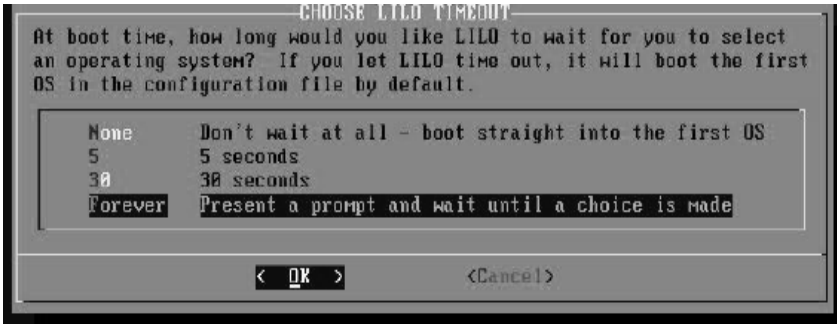
Enseguida podemos elegir del menú el tipo de resolución que deseamos para modo texto o consola, para ello necesitamos saber las características de la tarjeta de video y que tipos de resolución soporta, hoy en día casi todas las tarjetas soportan una resolución de 1024x768x256.



El siguiente paso es indicarle en donde deseamos instalar LILO, lo recomendable es que se instale en el MBR del disco duro.

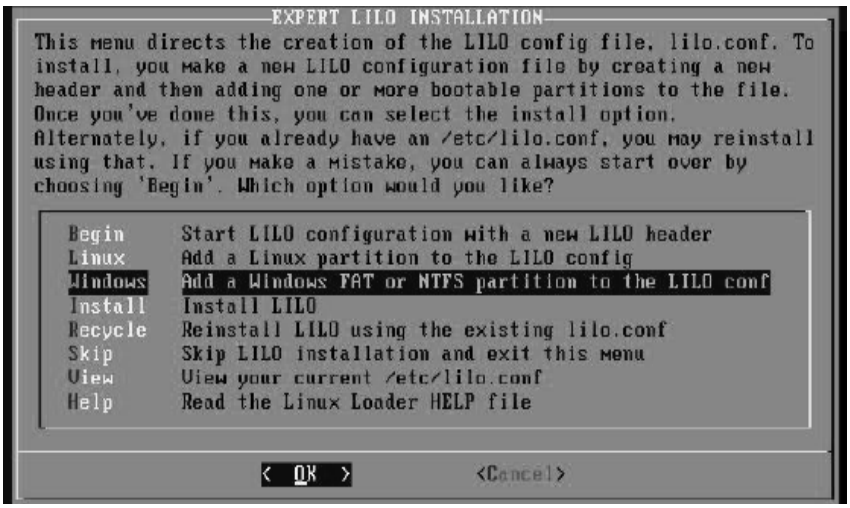
Para lo cual el asistente nos preguntara en que disco deseamos que se instale, esto por si contamos con más de un disco.

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

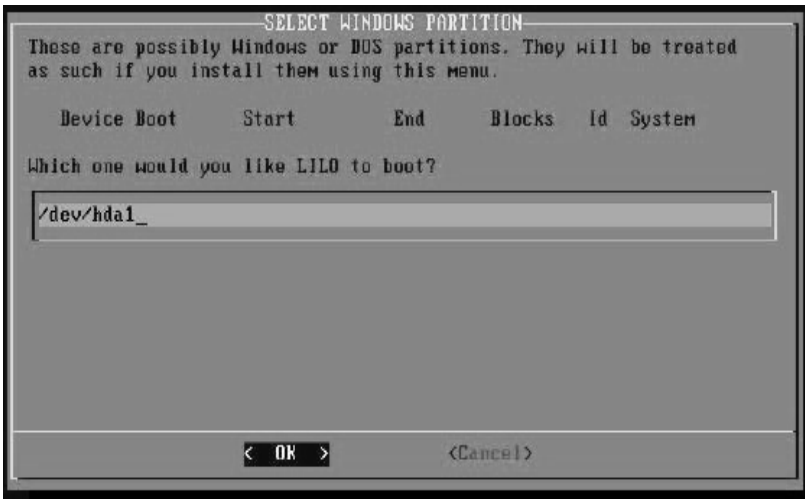


Hemos terminado de configurar la parte inicial de LILO, ahora hay que agregar los sistemas que deseamos sean visibles en el menú de booteo de los sistemas operativos.

Comenzaremos agregando un sistema Windows, para ello del menú elegimos esta opción:



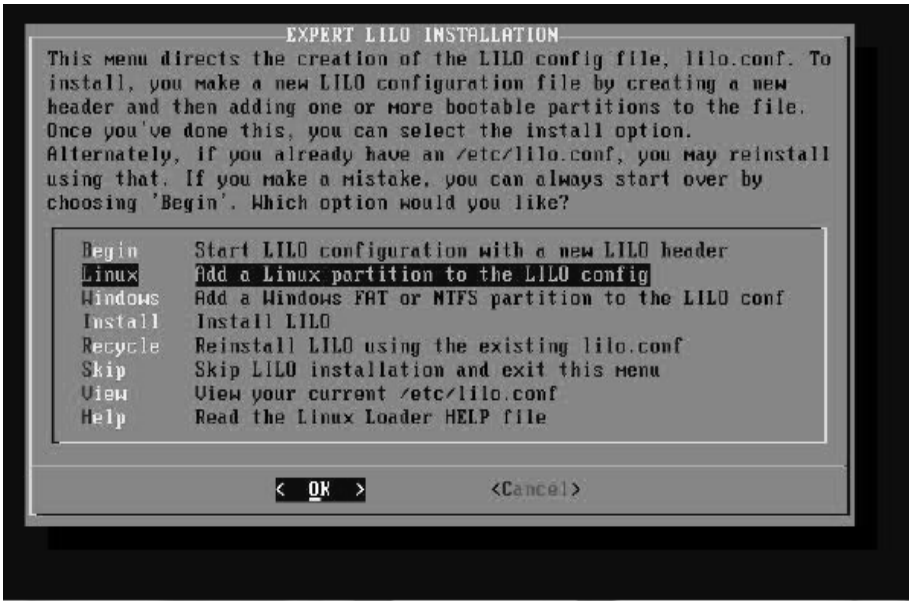
El asistente nos mostrará un listado de las particiones que son de sistemas tipo Windows, solicitándonos la elección de la partición apropiada.



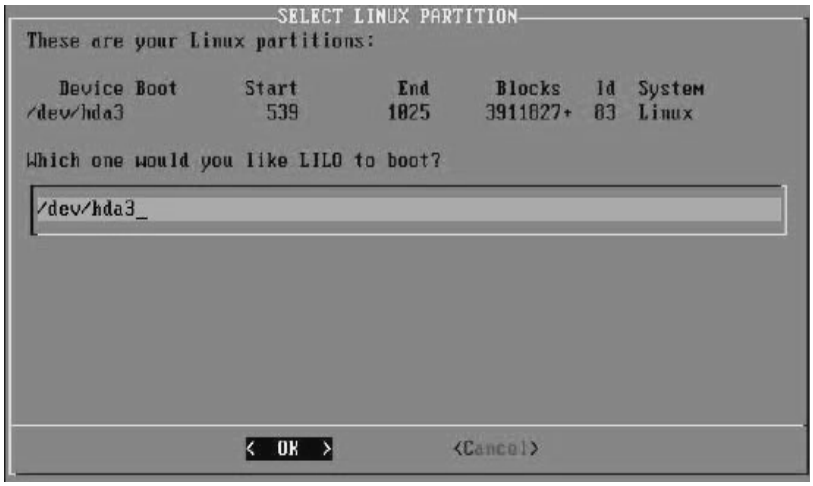
Ahora nos preguntará por la etiqueta con que queremos que se muestre este sistema en el menú de booteo.

Y de esta forma hemos agregado el sistema Windows, ahora tenemos que agregar Linux siguiendo este mismo procedimiento pero usando la opción de Linux del menú de configuración.

Ahora vamos a configurar la partición de Linux, para ello del menú elegimos dicha opción:



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

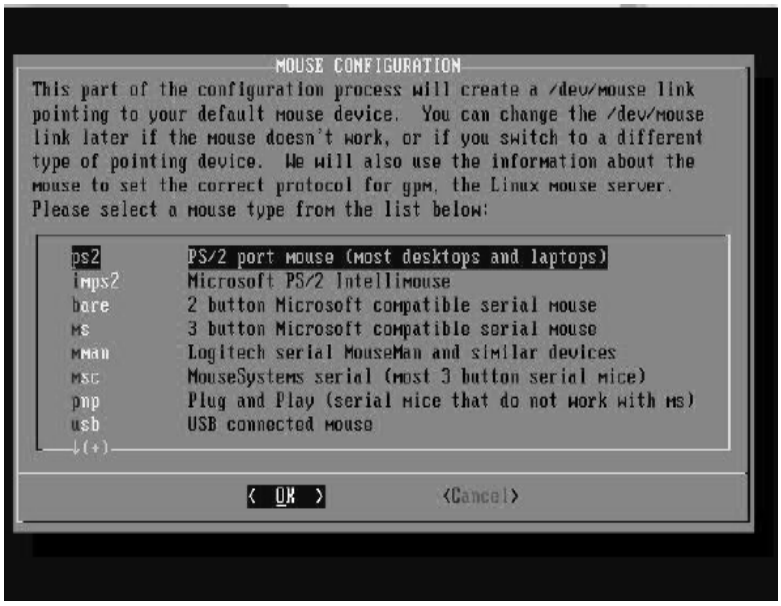


De igual forma tenemos que insertar la etiqueta como nombre que llevará en el menú de booteo.

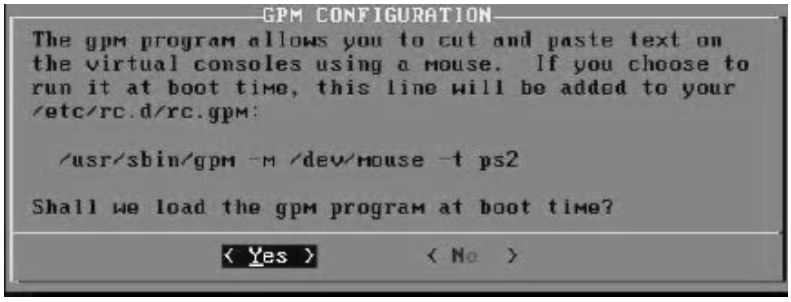
De esta forma hemos terminado de agregar los sistemas a nuestro gestor de arranque, si tuviéramos más sistemas el procedimiento seria el mismo.

El siguiente paso es indicarle al sistema que instale Lilo.

Ahora tenemos que indicarle que tipo de mouse tenemos conectado a la computadora, en este caso es un ps2.



La opción GPM sirve para habilitar el mouse en modo consola, es de gran ayuda para copiar y pegar en forma rápida.



Procedemos a configurar la red si es que nuestro equipo cuenta con una tarjeta que es soportada por esta distribución de Linux y es detectada automáticamente.

El asistente nos ofrece la posibilidad de indicarle los servicios que deseamos estén en funcionamiento, si no le indicamos al sistema que se inicien desde este menú lo podemos hacer posteriormente.

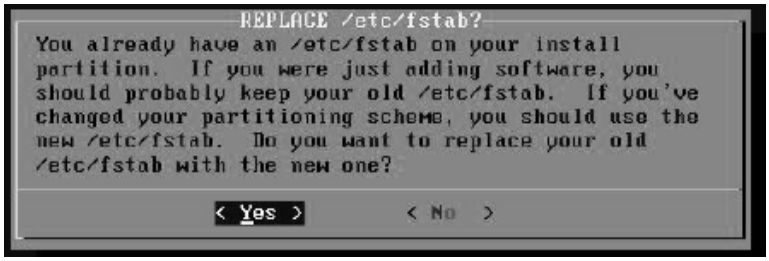
Una mala idea es configurar las fuentes para las consolas ya que no sabemos como puede variar la tipografía.

Si contamos con un reloj atómico que sincronice todas las computadoras, lo podemos configurar en este momento.

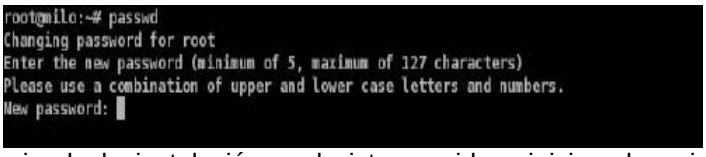
Es importante definir la zona horaria ya que nos ayuda a los cambios automáticos de horario por temporadas.

El entorno gráfico que elijamos en este momento es con la que va a iniciar la X por default.

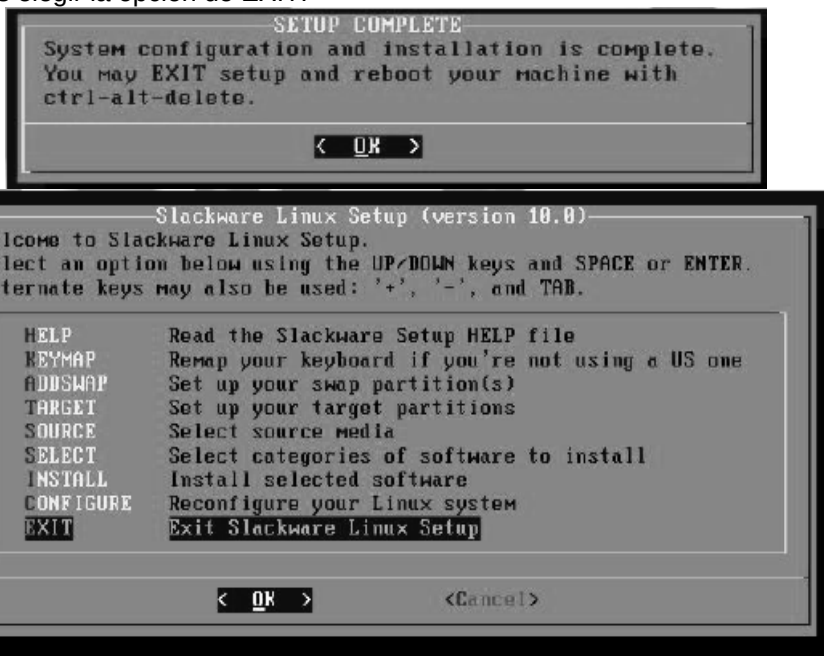
A continuación el sistema nos indica que ya se tiene un archivo /etc/fstam, y nos manda un aviso de que va a ser remplazado a lo cual le indicamos que si lo deseamos remplazar.



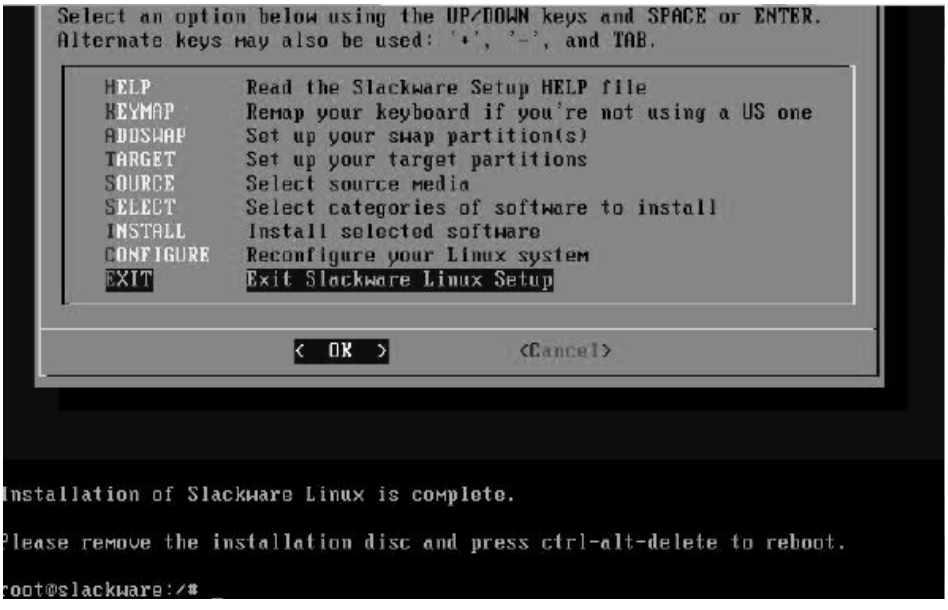
Ahora es momento de teclear una buena contraseña para el usuario administrador root.



Se ha terminado la instalación y el sistema pide reiniciar el equipo presionando la combinación de teclas ctrl-alt-sup, si presionamos enter nos regresa al menú de instalación, donde podemos elegir la opción de EXIT.



Eligiendo la opción de EXIT el sistema extrae la charola de CD para retirar el disco de instalación, ahora podemos retirar el CD y reiniciar el equipo con la combinación de teclas ctrl-alt-sup.



Después de reiniciar el sistema debemos tener la pantalla del gestor de arranque LILO en la cual podemos elegir el sistema a arrancar, y de esta manera Linux Slackware está instalado.

ANEXO II CÓDIGO FUENTE DEL SISTEMA DESARROLLADO

A continuación se adjunta el código fuente de la aplicación desarrollada en el capitulo VI, se generará un archivo con extensión ".php" con el nombre del encabezado de cada sección de código.

----------administra.php----------

```
<?
session_start();
if($_POST){
require_once "documents/conexion.php";
$nick=$_POST['nick'];
$password=$_POST['password'];
function generasesion($nick,$password){
$conexion=conectarse();
$bd="Pixup";
mysql_select_db($bd,$conexion) or die ("Error al seleccionar la base de datos " . mysql_error($conexion));
$sqlconsulta="SELECT * FROM Admin";
$resultado = mysql_query($sqlconsulta,$conexion) or die("Error en la consulta " .
mysql_error($conexion));
    while( ($fila=mysql_fetch_array($resultado,MYSQL_ASSOC)) ) {
        if((strcasecmp($nick,$fila['Login'])==0) && (strcmp($password,$fila['Password'])==0)){
    return true;
    }
   }
return false;
}
if(!generasesion($nick,$password)){
echo "NO puedes Entrar<br>";
echo "<A href='administra.php'>Vuevle a intentarlo</A>";
}
else{
$_SESSION['usuario']=$nick;

?>
<html>
<head>
<title>ACCESO RESTRINGIDO</title>
</head>
<body  bgcolor="Yellow">
<h1 align="center">PÁGINA CON ACCESO RESTRINGIDO</h1><hr><br>
<h2 align="center">PARA LA ADMINISTRACION DEL INVENTARIO DE EQUIPO DE
COMPUTO</h2><br><br>
<h2  align="center">BRAINUP SYSTEMS</h2><br><br>
<h4 align="center"><a href="master.php">CONSULTAR</a></h4>
</body>
</html>
<?
}
}
else{
?>
<html>
<head>
<title>
INVENTARIO DE EQUIPOS DECOMPUTO VALIDA USUARIO</title>
</head>
<body bgcolor="Yellow">
<h1 align="center">INVENTARIO DE EQUIPOS DE COMPUTO<br> VALIDA USUARIO</h1><hr>
<center>
<form action="<? print $_SERVER['PHP_SELF']; ?>" method="post">
LOGIN:
<input type="text" name="nick" maxlength="30"><br><br>
PASSWORD:
<input type="password" name="password" maxlength="30"><br><br>
```

```
<input type="submit" value="Ingresar">
</form>
</center>
</body>
</html>
<?
}
?>
```

----------generasesion.php----------

```
<?
session_start();
if(!isset($_SESSION['usuario'])){
header("Location: denegado.php");
}
else{
?>
<html>
<head>
<title>INFORME</title>
</head>
<body  bgcolor="Silver">
<center>
<?
echo "<h2 align='center'>Estás en CATALOGO GENERO:</h2>"; $_SESSION['usuario'];
echo  $_SESSION['usuario'];
echo "<br><br>"
?>
<FORM action="destruyesesion.php">
<INPUT type="submit" value="SALIR">
</FORM>
<?
}
?>
<?
if($_GET){
$genero=$_GET['genero'];
require_once "documents/conexion.php";
$conexion=conectarse();
$bd="Pixup";
mysql_select_db($bd,$conexion) or die ("Error al seleccionar la base de datos " . mysql_error($conexion));
$sqlInserta="INSERT INTO Genero VALUES ('','$genero')";
$resultado=mysql_query($sqlInserta,$conexion) or die ("Error en la insercion " .mysql_error($conexion));
print "Número de registros afectados :" . mysql_affected_rows($conexion);
print "<br>Registro Insertado";
print "<br>genero agregado : <b>$genero</b><br><br>";
?>
<a href="<? print $_SERVER['HTTP_REFERER']; ?>">INSERTAR OTRO GENERO</a>
<?

}
else{

?>
<form action="<? print $_SERVER['PHP_SELF']; ?>" method="get">
INSERTAR UN NUEVO GENERO<br> <br>
<?
require_once "documents/conexion.php";
$conexion=conectarse();
$bd="Pixup";

mysql_select_db($bd,$conexion) or die ("Error al seleccionar la base de datos " . mysql_error($conexion));

$sqlConsulta="SELECT Genero.Nombre AS 'Genero'FROM Genero";
$resultado=mysql_query($sqlConsulta,$conexion) or die ("Error en la consulta " . mysql_error($conexion));
print "Numero de Generos :" . mysql_affected_rows($conexion) . "<br><br>";
```

```
$contador=0;
echo "<table border='2'>";
echo "<tr>";
echo "<td bgcolor='red'>"; echo "Genero";   echo "</td>";
while($fila=mysql_fetch_array($resultado)){
$Genero=$fila['Genero'];
echo "<tr><td>";echo $Genero;echo "</td></tr> ";
$contador++;
}
echo "</table><br><br>";

?>
Nombre
<input type="text" name="genero"> <br><br>
<input type="submit" value="Insertar Genero">
</form>
<?
}
?>
<?
?>
</body>
</html>
```

----------master.php----------

```
<?
session_start();
if(!isset($_SESSION['usuario']))
header("Location: denegado.php");
else{
?>
<FORM>
<head>
<title>INFORME</title>
</head>
<body  bgcolor="Yellow">
<center>
<?
echo "<h2 align='center'>Estás en informe de Inventario:</h2>"; $_SESSION['usuario'];
echo "Como usuario:  ";
echo  $_SESSION['usuario'];
echo "<br><br>"
?>
<?
}
?>
<?
//menu de master
if ($_SESSION['usuario']=="master"){
echo "<A href='administraequipo.php'>Administrar Equipos</A><br>";
}
>
<br><br>
<A href="destruyesesion.php">SALIR POR COMPLETO</A>
</body>
</html>
```

----------denegado.php----------

```
<html>
<head>
<title>NO PUEDE ENTRAR A ESTA PAGINA</title>
</head>
<body  bgcolor="Yellow">
<center>
USTED NO TIENE PERMISOS PARA VER LA INFORMACION DE ESTA PAGINA<br>
<IMG src="alto.gif" width="89" height="90" border="0">
```

```
</center>
<CENTER>
<a href="administra.php"><em>VOLVER A INTENTAR</em> </a>
</CENTER>
</body>
</html>
```

----------insertaequipo.php----------

```
<?
session_start();
if(!isset($_SESSION['usuario'])){
header("Location: denegado.php");
}
else{
?>
<html>
<head>
<title>EQUIPO</title>
</head>
<body  bgcolor="Silver">
<center>

<?
echo "<h2 align='center'>ESTAS EN CATALOGO DE EQUIPOS DE COMPUTO:</h2>";
$_SESSION['usuario'];
echo "Como usuario:  ";
echo  $_SESSION['usuario'];
echo "<br><br>";

}
?>
</center>
<html>
<head>
<title>AGREGAR EQUIPO</title>
</head>
<body bgcolor="blue">
<center><h1>CATALOGO DE EQUIPOS</h1><hr></center>

<?
if($_GET){

$marca=$_GET['marca'];
$modelo=$_GET['modelo'];
$NSCpu=$_GET['NSCpu'];
$NSMonitor=$_GET['NSMonitor'];
$NSMouse=$_GET['NSMouse'];
$NSTeclado=$_GET['NSTeclado'];
$MemoriaRam=$_GET['MemoriaRam'];
$SistOp=$_GET['SistOp'];
$DiscoDuro=$_GET['DiscoDuro'];
$Usuario=$_GET['Usuario'];

define("HOST","localhost");
define("USUARIO","root");
define("PASSWORD","");


$conexion = mysql_pconnect(HOST,USUARIO,PASSWORD) or die("Error al conectarse ". mysql_error());
$bd="Pixup";

mysql_select_db($bd,$conexion) or die ("Error al seleccionar la base de datos " . mysql_error($conexion));
$sqlInserta="INSERT INTO Equipo VALUES
('','$marca','$modelo','$NSCpu','$NSMonitor','$NSMouse$_GET','$NSTeclado','$MemoriaRam','$SistOp','$
DiscoDuro','$Usuario')";
$resultado=mysql_query($sqlInserta,$conexion) or die ("Error en la insercion " .mysql_error($conexion));
print "<center>";
```

```
print "Número de registros afectados :" . mysql_affected_rows($conexion);
print "<br>Registro Insertado";
print "<br>Usuario agregado : <b>$nombre</b><br><br>";
print "</center>";

?>
<a href="<? print $_SERVER['HTTP_REFERER']; ?>"><center>INSERTAR OTRO USUARIO</center>
</a> <br>
<?
}
else{
?>

<form  action="<? print $_SERVER['PHP_SELF']; ?>" method="get">

<strong>INSERTAR UN NUEVO EQUIPO</strong><br> <br>
MARCA
<input type="text" name="marca"> <br><br>
MODELO
<input type="text" name="modelo"><br><br>
NUMERO DE SERIE DEL CPU
<input type="text" name="NSCpu"><br><br>
NUMERO DE SERIE DEL MONITOR
<INPUT  type="text" name="NSMonitor"><br><br>
NUMERO DE SERIE DEL MOUSE
<input type="text" name="NSMouse"><br><br>
NUMERO DE SERIE DEL TECLADO
<input type="text" name="NSTeclado"><br><br>
MEMORIA RAM (MB)
<input type="text" name="MemoriaRam"><br><br>
SISTEMA OPERATIVO
<input type="text" name="SistOp"><br><br>

DISCO DURO (GB)
<input type="text" name="DiscoDuro"><br><br>

USUARIO ASIGNADO AL EQUIPO
<input type="text" name="Usuario"><br><br>

<center><input type="submit" value="Agregar usuario"></center>
</form>

<?

}
?>
<center><a href="administraequipo.php">REGRESAR</a></center>

</body>
</html>
```

----------actualizaequipo.php----------

```
<?
session_start();
if(!isset($_SESSION['usuario'])){
header("Location: denegado.php");
}
else{
?>

<html>
<head>
<title>ACTUALIZAR</title>
</head>
<body  bgcolor="Silver">
<center>
```

```
<?
echo "<h2 align='center'>ESTAS EN CATALOGO EQUIPO:</h2>"; $_SESSION['usuario'];
echo "Como usuario:  ";
echo  $_SESSION['usuario'];
echo "<br><br>";

}
?>

<html>
<head>
<title>ACTUALIZAR EQUIPO</title>
</head>
<body bgcolor="#9999cc">
<h1 align="center">ACTUALIZAR EQUIPO</h1><hr>
<center>
<?
define("HOST","localhost");
define("USUARIO","root");
define("PASSWORD","");

$conexion = mysql_pconnect(HOST,USUARIO,PASSWORD) or die("Error al conectarse ". mysql_error());
$bd="Pixup";
mysql_select_db($bd,$conexion) or die ("Error al seleccionar la base de datos " . mysql_error($conexion));

$sqlconsulta="SELECT * FROM  Equipo";
$resultado=mysql_query($sqlconsulta,$conexion) or die ("Error en la consulta " . mysql_error($conexion));
print "Número de Equipos :" . mysql_affected_rows($conexion) . "<br><br>";
print "Selecciona el Equipo a modificar<br><br>";



print "<form action='actualizaequipo0.php' method='get'>";

print "<select name='idEquipo'>";
while($fila=mysql_fetch_array($resultado,MYSQL_NUM)){
print "<option value='".$fila[0]."'>".$fila[2];
}
print "</select>";
?>
<br><br>
<input type="submit" value="Actualizar Equipo">
</form>
<br>
<A href='administraequipo.php'>Regresar</A><br>
</center>
</body>
</html>



<html>
<head>
<title>ACTUALIZAR E</title>
</head>
<body bgcolor="#9999cc">
<h1 align="center">ACTUALIZAR EQUIPO</h1><hr>
<center>
<?

define("HOST","localhost");
define("USUARIO","root");
define("PASSWORD","");

$conexion = mysql_pconnect(HOST,USUARIO,PASSWORD) or die("Error al conectarse ". mysql_error());
```

```
$bd="Pixup";
mysql_select_db($bd,$conexion) or die ("Error al seleccionar la base de datos " . mysql_error($conexion));
$idEquipo=$_GET['idEquipo'];
$sqlconsulta="SELECT * FROM Equipo WHERE IdEquipo=$idEquipo";
$resultado=mysql_query($sqlconsulta,$conexion) or die ("Error al listar el Equipo ".
mysql_error($conexion));


while($fila=mysql_fetch_array($resultado,MYSQL_ASSOC)){
?>
<form action='actualizaequipo1.php' method="get">
<input type="hidden" name="idEquipo" value="<? print $fila['IdEquipo']; ?>">

MARCA:
<input type="text" name="Marca" value="<? print $fila['Marca']; ?>"><br><br>
MODELO:
<input type="text" name="Modelo" value="<? print $fila['Modelo']; ?>"><br><br>
NUMERO DE SERIE DEL CPU
<input type="text" name="NSCpu" value="<? print $fila['NSCpu']; ?>"><br><br>
NUMERO DE SERIE DEL MONITOR:
<input type="text" name="NSMonitor" value="<? print $fila['NSMonitor']; ?>"><br><br>
NUMERO DE SERIE DEL MOUSE:
<input type="text" name="NSMouse" value="<? print $fila['NSMouse']; ?>"><br><br>
NUMERO DE SERIE DEL TECLADO
<input type="text" name="NSTeclado" value="<? print $fila['NSTeclado']; ?>"><br><br>
MEMORIA RAM (MB)
<input type="text" name="MemoriaRam" value="<? print $fila['MemoriaRam']; ?>"><br><br>
SISTEMA OPERATIVO
<input type="text" name="SistOp" value="<? print $fila['SistOp']; ?>"><br><br>
DISCO DURO (GB)
<input type="text" name="DiscoDuro" value="<? print $fila['DiscoDuro']; ?>"><br><br>
USUARIO ASIGNADO AL EQUIPO
<input type="text" name="Usuario" value="<? print $fila['Usuario']; ?>"><br><br>

<?
}

?>

<input type="submit" value="Actualizar Equipo">
</form>
</center>
</body>
</html>
```

----------actualizaequipo0.php----------

```
<html>
<head>
<title>ACTUALIZAR E</title>
</head>
<body bgcolor="#9999cc">
<h1 align="center">ACTUALIZAR EQUIPO</h1><hr>
<center>
<?

define("HOST","localhost");
define("USUARIO","root");
define("PASSWORD","");

$conexion = mysql_pconnect(HOST,USUARIO,PASSWORD) or die("Error al conectarse ". mysql_error());

$bd="Pixup";
mysql_select_db($bd,$conexion) or die ("Error al seleccionar la base de datos " . mysql_error($conexion));
$idEquipo=$_GET['idEquipo'];
$sqlconsulta="SELECT * FROM Equipo WHERE IdEquipo=$idEquipo";
```

```
$resultado=mysql_query($sqlconsulta,$conexion) or die ("Error al listar el Equipo ".
mysql_error($conexion));


while($fila=mysql_fetch_array($resultado,MYSQL_ASSOC)){
?>
<form action='actualizaequipo1.php' method="get">
<input type="hidden" name="idEquipo" value="<? print $fila['IdEquipo']; ?>">

MARCA:
<input type="text" name="Marca" value="<? print $fila['Marca']; ?>"><br><br>
MODELO:
<input type="text" name="Modelo" value="<? print $fila['Modelo']; ?>"><br><br>
NUMERO DE SERIE DEL CPU
<input type="text" name="NSCpu" value="<? print $fila['NSCpu']; ?>"><br><br>
NUMERO DE SERIE DEL MONITOR:
<input type="text" name="NSMonitor" value="<? print $fila['NSMonitor']; ?>"><br><br>
NUMERO DE SERIE DEL MOUSE:
<input type="text" name="NSMouse" value="<? print $fila['NSMouse']; ?>"><br><br>
NUMERO DE SERIE DEL TECLADO
<input type="text" name="NSTeclado" value="<? print $fila['NSTeclado']; ?>"><br><br>
MEMORIA RAM (MB)
<input type="text" name="MemoriaRam" value="<? print $fila['MemoriaRam']; ?>"><br><br>
SISTEMA OPERATIVO
<input type="text" name="SistOp" value="<? print $fila['SistOp']; ?>"><br><br>
DISCO DURO (GB)
<input type="text" name="DiscoDuro" value="<? print $fila['DiscoDuro']; ?>"><br><br>
USUARIO ASIGNADO AL EQUIPO
<input type="text" name="Usuario" value="<? print $fila['Usuario']; ?>"><br><br>
<?
}
?>
<input type="submit" value="Actualizar Equipo">
</form>
</center>
</body>
</html>
```

----------actualizaequipo1.php----------

```
<html>
<head>
<title>ACTUALIZAR EQUIPO</title>
</head>
<body bgcolor="#9999cc">
<h1 align="center">ACTUALIZAR EQUIPO</h1><hr>
<center>
<?
define("HOST","localhost");
define("USUARIO","root");
define("PASSWORD","");

$conexion = mysql_pconnect(HOST,USUARIO,PASSWORD) or die("Error al conectarse ". mysql_error());

$bd="Pixup";

mysql_select_db($bd,$conexion) or die ("Error al seleccionar la base de datos " . mysql_error($conexion));

$idEquipo=$_GET['idEquipo'];
$Marca=$_GET['Marca'];
$Modelo=$_GET['Modelo'];
$NSCpu=$_GET['NSCpu'];
$NSMonitor=$_GET['NSMonitor'];
$NSMouse=$_GET['NSMouse'];
$NSTeclado=$_GET['NSTeclado'];
$MemoriaRam=$_GET['MemoriaRam'];
$SistOp=$_GET['SistOp'];
$DiscoDuro=$_GET['DiscoDuro'];
$Usuario=$_GET['Usuario'];
```

```
$sqlactualiza="UPDATE Equipo SET Marca='$Marca', Modelo='$Modelo', NSCpu='$NSCpu',
NSMonitor='$NSMonitor', NSMouse='$NSMouse', NSTeclado='$NSTeclado',
MemoriaRam='$MemoriaRam', SistOp='$SistOp', DiscoDuro='$DiscoDuro', Usuario='$Usuario' WHERE
IdEquipo=$idEquipo";
$resultado=mysql_query($sqlactualiza,$conexion) or die ("Error al actualizar el Equipo ".
mysql_error($conexion));
print "Número de registros actualizados : " . mysql_affected_rows($conexion);
print "<br>Registro Actualizado<br><br>";
?>
<a href="actualizaequipo.php">Actualizar otro usuario</a> </center>
</body>
</html>
```

----------eliminaequipo.php----------

```
<?
session_start();
if(!isset($_SESSION['usuario'])){
header("Location: denegado.php");
}
else{
?>

<html>
<head>
<title>ELIMINAR EQUIPOS</title>
</head>
<body  bgcolor="Silver">
<center>

<?
echo "<h2 align='center'>ESTAS EN CATALOGO DE EQUIPOS:</h2>"; $_SESSION['usuario'];
echo "Como usuario:  ";
echo  $_SESSION['usuario'];
echo "<br><br>";

}
?>
<html>
<head>
<title>ELIMINAR EQUIPO</title>
</head>
<body bgcolor="#9999cc">
<h1 align="center">ELIMINAR EQUIPO</h1><hr>
<center>
<?
//require_once "lib/conexion.php";
//$conexion=conectarse();

define("HOST","localhost");
define("USUARIO","root");
define("PASSWORD","");

$conexion = mysql_pconnect(HOST,USUARIO,PASSWORD) or die("Error al conectarse ". mysql_error());
$bd="Pixup";

mysql_select_db($bd,$conexion) or die ("Error al seleccionar la base de datos" . mysql_error($conexion));
if($_GET){
$idEquipo=$_GET['idEquipo'];
$sqlactualiza="DELETE FROM Equipo WHERE IdEquipo=$idEquipo";
$resultado=mysql_query($sqlactualiza,$conexion) or die ("Error al eliminar los registros ".
mysql_error($conexion));
print "Número de registros eliminados : " . mysql_affected_rows($conexion) . "<br><br>";
}
$sqlconsulta="SELECT * FROM  Equipo";
$resultado=mysql_query($sqlconsulta,$conexion) or die ("Error en la consulta " . mysql_error($conexion));
print "Número de registros : " . mysql_affected_rows($conexion) . "<br><br>";
```

```
print "Selecciona el Equipo a eliminar<br><br>";
print "<form action='".$_SERVER['PHP_SELF']."' method='get'>";


print "<select name='idEquipo'>";
while($fila=mysql_fetch_array($resultado)){
print "<option value='".$fila['IdEquipo']."'>".$fila['Marca'];
}
print "</select>";
?>
<br><br>
<input type="submit" value="Eliminar Equipo">
</form><br><br>
<a href="administraequipo.php">REGRESAR</a>
</center>
</body>
</html>
```

----------consultaequipo.php----------

```
<?
session_start();
if(!isset($_SESSION['usuario'])){
header("Location: denegado.php");
}
else{
?>

<html>
<head>
<title>CATALOGO DE EQUIPOS</title>
</head>
<body  bgcolor="Silver">
<center>

<?
echo "<h2 align='center'>ESTAS EN CATALOGO DE EQUIPOS:</h2>"; $_SESSION['usuario'];
echo "Como usuario:  ";
echo  $_SESSION['usuario'];
echo "<br><br>";

}
?>
<html>
<head>
<title>LISTADO DE EQUIPOS</title>
</head>
<body bgcolor="blue">
<h1 align="center">CATALOGO DE  EQUIPOS</h1><hr>
<center>

<?
require_once "documents/conexion.php";
$conexion=conectarse();
$bd="Pixup";

mysql_select_db($bd,$conexion) or die ("Error al seleccionar la base de datos" . mysql_error($conexion));

$sqlconsulta="SELECT Equipo.Marca as 'MARCA', Equipo.Modelo as 'Modelo', Equipo.NSCpu as 'Num
de Serie del Cpu', Equipo.NSMonitor as 'Num de serie del Monitor', Equipo.NSMouse as 'Num de Serie
del Mouse', Equipo.NSTeclado as 'Num de Serie del Teclado', Equipo.MemoriaRam as 'Memoria Ram',
Equipo.SistOp as 'Sistema Operativo', Equipo.DiscoDuro as 'Disco Duro', Equipo.Usuario as 'Usuario'
FROM Equipo";
$resultado=mysql_query($sqlconsulta,$conexion) or die ("Error en la consulta ".mysql_error($conexion));
print "Número de registros afectados : " . mysql_affected_rows($conexion) . "<br><br>";


echo "<form >";
```

```
print "<table>";
print "<tr bgcolor='purple'><td>MARCA</td><td>MODELO</td><td>NUMERO DE SERIE DEL
CPU</td><td>NUMERO DE SERIE DEL MONITOR</td><td>NUMERO DE SERIE DEL
MOUSE</td><td>NUMERO DE SERIE DEL TECLADO</td><td>MEMORIA RAM (MB)</td><td>SISTEMA
OPERATIVO</td><td>DISCO DURO (GB)</td><td>USUARIO ASIGNADO AL EQUIPO</td></tr>";
$contador=0;
while($fila=mysql_fetch_array($resultado)){
$color="#dddddd";
if($contador%2==0)
$color="#888888";
print "<tr bgcolor='$color'>";
print "<td>" . $fila['MARCA'] . "</td>";
print "<td>" . $fila['Modelo'] . "</td>";
print "<td>" . $fila['Num de Serie del Cpu'] . "</td> ";
print "<td>" . $fila['Num de serie del Monitor'] . "</td> ";
print "<td>" . $fila['Num de Serie del Mouse'] . "</td> ";
print "<td>" . $fila['Num de Serie del Teclado'] . "</td> ";
print "<td>" . $fila['Memoria Ram'] . "</td> ";
print "<td>" . $fila['Sistema Operativo'] . "</td> ";
print "<td>" . $fila['Disco Duro'] . "</td> ";
print "<td>" . $fila['Usuario'] . "</td> ";

print "</tr>";
$contador++;
}
print "</table>";
?>
<br><br><A href="administraequipo.php">REGRESAR</A>
</BODY>
</HTML>
```

----------destruyesesion.php----------

```
<?
session_start();
if(!isset($_SESSION['usuario']))
header("Location: denegado.php");
else{
unset($_SESSION['usuario']);
header("Location: administra.php");
}
?>
```

----------conexion.php----------

```
<?
function conectarse(){
define("HOST","localhost");
define("USUARIO","root");
define("PASSWORD","");
$conexion = mysql_pconnect(HOST,USUARIO,PASSWORD) or die("Error al conectarse " .
mysql_error());
return $conexion;
}
?>
```

Finalmente copiar estos archivos al “Document root” de Apache ubicado en /Usr/local/Apache2/htdocs/