



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA Y TECNOLOGIA EN

COMPUTACIÓN

“DISEÑO E IMPLEMENTACIÓN DE UN LABORATORIO DE SOFTWARE Y REDES
MEDIANTE EL USO DE UN SERVIDOR DE TERMINALES PARA LA ESCUELA DE
INGENIERÍA ELECTRÓNICA”

TESIS DE GRADO

Previa a la obtención del título de

INGENIERO EN ELECTRÓNICA Y COMPUTACIÓN

Presentado por:

FRANKLIN EFRAÍN AGUIRRE LÓPEZ

RIOBAMBA – ECUADOR

2009

Agradezco al Ing. Daniel Haro por apoyarme constantemente para la realización del presente trabajo, a mi hermana Claudia Aguirre y su esposo Segundo Bonilla que aportaron con un granito de arena.

Dedico mi esfuerzo empleado en este trabajo, a mi familia, a mi padre Efraín Aguirre Chamba, mi madre Leonila Micaela López Vivanco, quienes con su sacrificio y constante apoyo me han permitido culminar mi carrera. A mi hermana Isabel Aguirre. A mis sobrinos: Kevin y Geomara Bonilla, Jefferson García y Evelyn Aguirre que siempre han estado cerca de mí a lo largo de mi carrera estudiantil. A mi hija Ambar Kamila Aguirre Gallegos que desde el momento en que llegó a mi vida ha sido mi inspiración y a mi familia quienes han sido un pilar fundamental con su apoyo incondicional en todo momento.

NOMBRE	FIRMA	FECHA
Dr. Romeo Rodríguez	<hr/>	<hr/>
DECANO FACULTAD DE INFORMATICA Y ELECTRÓNICA		
Ing. Paul Romero	<hr/>	<hr/>
DIRECTOR DE ESCUELA INGENIERIA ELECTRÓNICA		
Ing. Daniel Haro	<hr/>	<hr/>
DIRECTOR DE TESIS		
Ing. Alberto Arellano	<hr/>	<hr/>
MIEMBRO DEL TRIBUNAL		
Lcdo. Carlos Rodríguez	<hr/>	<hr/>
DIR. DPTO. DOCUMENTACIÓN		
NOTA DE LA TESIS	<hr/>	

"Yo, Franklin Efraín Aguirre López, soy responsable de las ideas, doctrinas y resultados expuestos en esta Tesis de Grado; y, el patrimonio intelectual de la Tesis de Grado pertenece a la ESCUELA SUPERIOR POLITECNICA DEL CHIMBORAZO".

Franklin Efraín Aguirre López

ÍNDICE DE ABREVIATURAS

ACL	Access Control List (Listas de Control de Acceso)
API	Application Programming Interface (Interfaz de Programación de Aplicaciones)
BCP	Block Control Process (Bloque de Control de Proceso)
BIOS	Basic Input/Output System (Sistema Básico de Entrada/Salida)
BOOTP	Bootstrap Protocol (Protocolo de Arranque)
DHCP	Dinamic Host Configuration Protocol (Protocolo de Configuración Dinámica de Host)
DC	Diskless Computer (Computadora sin Disco)
DNS	Domain Name System (Sistema de Nombres de Dominio)
E/S	Entrada/Salida
FTP	File Transfer Protocol (Protocolo de Transferencia de Archivos)
GUI	Graphical User Interface (Interfaz Gráfica de Usuario)
GNU	Gnu is Not Unix (Gnu No es Unix)
IEEE	Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos)
LTSP	Linux Terminal Server Project (Proyecto de Servidor de Terminales Linux)
MBA	Managed Boot Agent (Agente Administrador de Arranque)
MS-DOS	Microsoft Disk Operating System (Sistema Operativo de Disco de Microsoft)
NFS	Network File System (Sistema de Archivos de Red)
NT	New Technology (Nueva Tecnología)
PC	Personal Computer (Computadora Personal)
POSIX	Portable Operating System Interface for X (Interfaz de Sistema

	Operativo Portable para X)
PXE	Pre-Boot eXecution Environment (Entorno de ejecución previo al arranque)
RAM	Random Access Memory (Memoria de Acceso Aleatorio)
ROM	Read Only Memory (Memoria sólo de lectura)
TCP/IP	Transmision Control Protocol/Internet Protocol. (Protocolo de Control de Transmisión/Protocolo de Internet)
UDP	User Datagram Protocol (Protocolo de Datagrama de Usuario)
SO	Sistema Operativo
SOR	Sistema Operativo de Red
TFTP	Trivial File Transfer Protocol (Protocolo de Transferencia de Archivos Trivial)
XDMCP	X Display Manager Control Protocol (Protocolo de Control de Administración de Visualización X)

ÍNDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

ÍNDICE DE ABREVIATURAS

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

CAPITULO I: INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

1.1	¿QUÉ ES UN SISTEMA OPERATIVO?	16
1.1.1	Máquina Desnuda	16
1.1.2	Funciones del Sistema Operativo	17
1.1.3	Concepto de usuario y de grupo de usuarios.....	22
1.2	ARRANQUE DE LA COMPUTADORA	23
1.2.1	Arranque hardware	24
1.2.2	Ubicación del SO.....	26
1.2.3	Arranque del SO	26
1.3	COMPONENTES Y ESTRUCTURA DEL SO	28
1.3.1	Componentes del SO	28
1.3.2	Estructura del SO.....	30
1.4	GESTIÓN DE PROCESOS	34
1.4.1	Servicios de Procesos	36
1.5	GESTIÓN DE MEMORIA	37
1.5.1	Servicios.....	38
1.6	COMUNICACIÓN Y SINCRONIZACIÓN ENTRE PROCESOS	39
1.6.1	Servicios de Comunicación y Sincronización	40
1.7	GESTIÓN DE E/S	42
1.7.1	Servicios.....	43

1.8	GESTIÓN DE ARCHIVOS Y DIRECTORIOS	43
1.8.1	Servicio de Archivos	44
1.8.2	Servicio de Directorios	49
1.8.3	Sistema de Archivos	51
1.9	SEGURIDAD Y PROTECCIÓN	52

CAPITULO II: DISKLESS BOOTING Y SISTEMAS OPERATIVOS DE RED

2.1	DISKLESS BOOTING	54
2.1.1	¿Cómo Funciona el Diskless Booting?.....	56
2.2	¿QUÉ SON LOS SISTEMAS OPERATIVOS DE RED?	60
2.3	WINDOWS SERVER 2003	61
2.3.1	Características	61
2.3.1.1	Sistema de Recuperación Automática.....	62
2.3.1.2	Volume Shadow Copy	62
2.3.1.3	Sistema de archivos encriptados	63
2.3.1.4	Driver Rollback	63
2.3.1.5	Directorio Activo (Active Directory).....	63
2.3.1.6	Políticas de Seguridad	63
2.3.1.7	Seguimiento de Errores "Event Tracker"	64
2.3.1.8	Versiones	64
2.3.1.9	Requerimientos	67
2.3.1.10	Funcionalidades.....	67
2.3.1.11	Controlador de Dominio (Directorio Activo)	68
2.3.1.12	Servidor de Archivos	68
2.3.1.13	Servidor de Impresión.....	69
2.3.1.14	Servidor DNS y DHCP	69
2.3.1.15	Servicio de terminal server	69
2.3.1.16	La herramienta Manage Your Server	69
2.4	GNU/LINUX	70
2.4.1	Introducción	70
2.4.2	Características de GNU/Linux	71
2.4.3	Estructura de los archivos en GNU/Linux	74
2.4.3.1	Directorios de Programas	76
2.4.3.2	Directorios y Archivos de Configuración.....	76

CAPÍTULO III: ESTUDIO COMPARATIVO ENTRE WINDOWS SERVER 2003 Y GNU/LINUX

3.1	INTRODUCCIÓN	86
3.2	CRITERIOS DE COMPARACIÓN	88
3.3	CUADROS COMPARATIVOS	90
3.3.1	Facilidad de Instalación.....	92
3.3.2	Disponibilidad de Software	95
3.3.3	Estabilidad	98
3.3.4	Utilización de Recursos y Rendimiento	101
3.3.5	Configurabilidad vs. Facilidad de uso	105
3.3.6	Código abierto vs. Código propietario	107
3.3.7	Sistema de Archivos	110
3.3.8	Seguridad y Precio	112
3.4	ANÁLISIS DE RESULTADOS DEL ESTUDIO COMPARATIVO ..	115
3.4.1	Comparativo General	115
3.4.1.1	Semejanzas y Diferencias.....	115
3.4.1.2	Ventajas y Desventajas.....	119
3.4.1.3	Resultados	121
3.4.2	Conclusiones	124
3.4.3	Recomendaciones	126

CAPÍTULO IV: CONFIGURACIÓN E INSTALACIÓN DE LTSP (LINUX TERMINAL SERVER PROJECT)

4.1	¿QUÉ ES LTSP?	127
4.2	TEORÍA DE OPERACIÓN	129
4.2.1	Pasos a Seguir por el Cliente LTSP	130
4.2.2	Carga del Kernel en la Memoria	135
4.2.2.1	Boot ROM.....	135
4.2.2.2	Medios Locales.....	136
4.3	INSTALANDO LTSP EN EL SERVIDOR	138
4.3.1	Instalar las Utilidades de LTSP (LTSP Utilities)	138
4.3.1.1	Instalando el Paquete RPM	139
4.3.1.2	Instalando el Paquete TGZ	139
4.3.2	Instalar los Paquetes Cliente de LTSP	139
4.3.3	Configurar los Servicios Necesarios de LTSP	146

4.3.4 Configuración Específica para los Clientes.....	152
4.3.4.1 /etc/dhcpd.conf.....	152
4.3.4.2 /etc/hosts	154
4.3.4.3 /opt/ltsp/i386/etc/lts.conf	154
4.3.5 Desplegando las Configuraciones Concurrentes	157
4.4 CONFIGURANDO LOS CLIENTES LTSP	158
4.4.1 Creando el Disquete de Inicio	158
4.5 EJECUTANDO NUESTRA DC.....	160
4.6 SCREEN SCRIPTS.....	161

CONCLUSIONES

RECOMENDACIONES

RESUMEN

SUMMARY

GLOSARIO

ANEXOS

BIBLIOGRAFÍA

ÍNDICE DE FIGURAS

Figura I.1	Niveles del Sistema Operativo.....	18
Figura I.2	Operaciones realizadas en el arranque de la computadora	23
Figura I.3	Una parte del mapa de memoria está construido con memoria	25
Figura I.4	El SO se encuentra almacenado en una unidad de disco	27
Figura I.5	Componentes del SO	29
Figura I.6	Estructura por capas de un Sistema Operativo	32
Figura I.7	Estructura cliente-servidor en un Sistema Operativo.....	33
Figura I.8	Elementos que constituyen un proceso	35
Figura I.9	Comunicación entre procesos locales y remotos.....	40
Figura I.10	Comunicación síncrona entre procesos.....	42
Figura I.11	Visión lógica y física del sistema de archivos.....	44
Figura I.12	Visión lógica de un archivo	45
Figura I.13	Visión física de un archivo	46
Figura I.14	Esquema jerárquico de directorios	50
Figura II.15	Nuevas características de Windows Server 2003	61
Figura II.16	Shadow copias.....	62
Figura II.17	Seguimiento de Errores.....	64
Figura II.18	Funcionalidades de Windows Server 2003	68
Figura III.19	Facilidad de instalación	94
Figura III.20	Disponibilidad de Software.....	97
Figura III.21	Estabilidad de las plataformas.....	100
Figura III.22	Utilización de recursos y rendimiento.....	104
Figura III.23	Configurabilidad vs. Facilidad de uso.....	106
Figura III.24	Código abierto vs. Código propietario	109
Figura III.25	Sistema de Archivos	111
Figura III.26	Seguridad y Precio	114
Figura III.27	Windows Server 2003 vs. GNU/Linux.....	122
Figura III.28	Porcentaje Final Windows Server 2003 - GNU/Linux	123
Figura IV.29	Instalador LTSP – Pantalla principal.....	140
Figura IV.30	Instalador LTSP – Pantalla de Configuración	141
Figura IV.31	Instalador LTSP – Lista de Componentes.....	143
Figura IV.32	Instalador LTSP – Ventana de Ayuda.....	144
Figura IV.33	Instalador LTSP – Lista de paquetes	145
Figura IV.34	Instalador LTSP – Instalación de paquetes	146
Figura IV.35	Itspcfg – Pantalla Inicial.....	147
Figura IV.36	Itspcfg – Pantalla inicial para configurar.....	148
Figura IV.37	Archivo /etc/dhcpd.conf.....	153
Figura IV.38	Archivo lts.conf	155
Figura IV.39	Itspcfg – Estado actual de los servicios.....	157
Figura IV.40	Pantalla del Terminal.....	160

ÍNDICE DE TABLAS

Tabla II-1	Versiones de Windows Server 2003.....	65
Tabla II-2	Requerimientos de Windows Server 2003	67
Tabla II-3	Directorios del Sistema	75
Tabla II-4	Archivos de Configuración	78
Tabla II-5	Subdirectorios de usr.....	79
Tabla II-6	Subdirectorios de /var.....	81
Tabla II-7	Subdirectorios y archivos de /proc	82
Tabla II-8	Prefijos de Nombres de Dispositivo	84
Tabla III-9	Facilidad de instalación entre WS2003 y GNU/Linux.....	92
Tabla III-10	Disponibilidad de Software entre WS2003 y GNU/Linux.....	95
Tabla III-11	Estabilidad de las plataformas WS2003 y GNU/Linux	98
Tabla III-12	Utilización de recursos y rendimiento	101
Tabla III-13	Configurabilidad vs. Facilidad de uso	105
Tabla III-14	Código abierto vs. Código propietario.....	107
Tabla III-15	Sistemas de Archivos.....	110
Tabla III-16	Seguridad y Precio.....	112

INTRODUCCIÓN

En la actualidad, debido a la nueva era tecnológica y por la falta de presupuesto, muchas instituciones se ven obligadas a esperar para poder adquirir nuevos equipos computacionales; en las universidades pasa lo contrario, se adquieren nuevos equipos creando basura al desechar los equipos considerados antiguos. La tendencia del nuevo milenio nos exige la reutilización de material y en el caso de la tecnología computacional se habla de poder reutilizar computadores que se supone son: equipos obsoletos.

Al momento de dar una solución viable para evitar crear basura tecnológica, debe considerarse la implementación de un "Servidor de Terminales" que nos permitirá ejecutar un sistema operativo nuevo en un computador con poco hardware.

Los servidores de terminales siempre han existido en la plataforma Unix, los terminales se ejecutaban como terminales tontos (PC sin CPU) y la interfaz era muy rústica al solo disponer de texto y no de gráficos. En la última década ya no se habla de este tipo de servidores debido a la aparición de sistemas operativos de escritorio, visuales y multimedia como: Windows y Mac.

Podemos utilizar los conceptos básicos sobre dichos servidores que ejecutaban solo texto en sus terminales, para adaptarlos a las necesidades actuales. Un computador de características tecnológicas modernas puede hacer de servidor y computadores antiguos sin disco duro pueden hacer de terminales que podrán ejecutar una interfaz gráfica proporcionada por el servidor

El contenido de esta tesis está estructurado en 4 capítulos, el **Capítulo I** proporciona una introducción y nociones generales acerca de los Sistemas operativos, el **Capítulo II** describe las principales características de las plataformas en estudio, el **Capítulo III** muestra un estudio investigativo sobre 2 plataformas para implementar el proyecto de tesis, el **Capítulo IV** se encarga de la documentación sobre la implementación del "Servidor de Terminales" previo el estudio en el Capítulo III.

Como resultado final se obtuvo un porcentaje final de **91.18%** y **48.04%** equivalentes a Excelente y Bueno respectivamente, lo que permitió determinar a GNU/Linux como la plataforma más adecuada para la configuración e instalación de LTSP (Linux Terminal Server Project), que se detalla en el Capítulo IV.

GNU/Linux es más seguro y más estable que Windows Server 2003, convirtiéndolo en la mejor opción a ser tomada al momento de instalar un servidor dentro de una red de computadoras.

Se recomienda el uso de un servidor con GNU/Linux debido a su alto desempeño mostrado en las pruebas realizadas en cuanto a facilidad de instalación, rendimiento y configurabilidad, permitiendo diseñar e implementar un laboratorio de software y redes con un costo muy reducido reutilizando computadores antiguos, a la vez que evitamos crear basura tecnológica y centralizamos la instalación de programas y la información para almacenarla sólo en nuestro servidor.

CAPITULO I

INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

1.1 ¿QUÉ ES UN SISTEMA OPERATIVO?

En esta sección se plantea en primer lugar el concepto de maquina desnuda para pasar acto seguido a introducir el concepto de sistema operativo y sus principales funciones.

1.1.1 Máquina Desnuda

El término de máquina desnuda se aplica a una computadora carente de sistema operativo. El término es interesante porque resalta el hecho de que una computadora en sí misma no hace nada. Una computadora solamente es capaz de repetir a alta velocidad la secuencia de: lectura de instrucción máquina, incremento del PC y ejecución de la instrucción leída.

Para que la computadora realice un función determinada ha de tener en memoria principal un programa máquina específico para realizar dicha función y ha de conseguirse que el registro PC contenga la dirección de comienzo del programa.

1.1.2 Funciones del Sistema Operativo

Un SO (sistema operativo) es un programa que tiene encomendadas una serie de funciones diferentes cuyo objetivo es simplificar el manejo y la utilización de la computadora, haciéndolo seguro y eficiente. Históricamente se han ido completando las misiones encomendadas al sistema operativo, por lo que los productos comerciales actuales incluyen una gran cantidad de funciones, como son interfaces gráficas, protocolos de comunicación, etc.

Las funciones clásicas, de un SO se pueden agrupar en las tres categorías siguientes:

- Gestión de los recursos de la computadora.
- Ejecución de servicios para los programas.
- Ejecución de los mandatos de los usuarios.

Como se muestra en la Figura I.1, el SO está formado conceptualmente por tres capas principales. La capa más cercana al hardware se denomina núcleo (kernel) y es la que gestiona los recursos hardware del sistema y la que suministra la funcionalidad básica del sistema operativo. Esta capa se ha de ejecutar en nivel núcleo, mientras que las otras se pueden ejecutar en niveles menos permisivos.

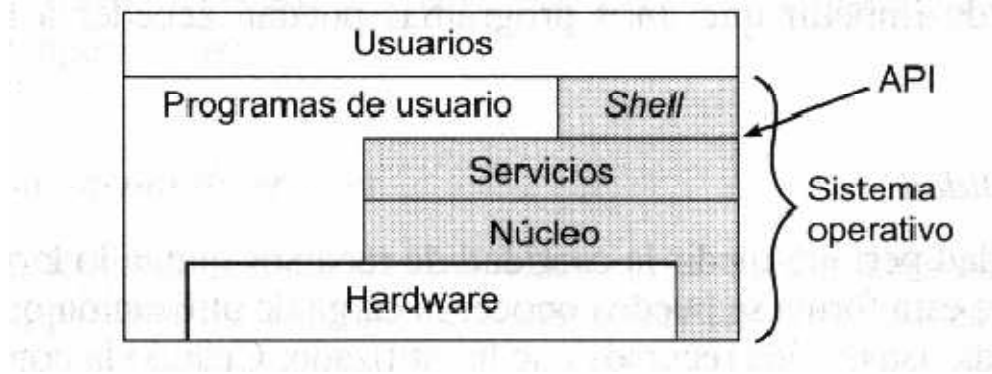


Figura I.1 Niveles del Sistema Operativo

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

La capa de servicios o **llamadas al sistema** ofrece a los programas unos servicios en forma de una interfaz de programación o API (Application Programming Interface).

Desde el punto de vista de los programas, esta capa extiende la funcionalidad de la computadora, por lo que se suele decir que el sistema operativo ofrece una máquina virtual extendida a los programas. De esta forma se facilita la elaboración de éstos, puesto que se apoyan en las funciones que le suministra el SO.

La capa de **intérprete de mandatos o Shell** suministra una interfaz a través de la cual el usuario puede dialogar de forma interactiva con la computadora. El Shell recibe los mandatos u órdenes del usuario, los interpreta y, si puede, los ejecuta.

Dado que el Shell se suele ejecutar en nivel de usuario, se puede considerar que no forma parte del SO. Seguidamente, se analizarán cada una de estas tres facetas del SO.

El SO como gestor de recursos

En una computadora actual suelen coexistir varios programas, del mismo o de varios usuarios, ejecutándose simultáneamente. Estos programas compiten por los recursos de la computadora, siendo el SO el encargado de arbitrar su asignación y uso. Como complemento a la gestión de recursos, el sistema operativo ha de garantizar la protección de unos programas frente a otros y ha de suministrar información sobre el uso que se hace de los recursos.

a) Asignación de recursos

El SO se encarga de asignar los recursos a los programas en ejecución. Para ello, ha de mantener unas estructuras que le permitan saber que recursos están libres y cuáles están asignados a cada programa. La asignación de recursos se realiza según la disponibilidad de los mismos y la prioridad de los programas, debiéndose resolver los conflictos que aparecen por las peticiones simultáneas.

Especial mención reviste la recuperación de los recursos cuando los programas ya no los necesitan. Una mala recuperación de recursos puede hacer que el sistema operativo considere, por ejemplo, que ya no le queda memoria disponible cuando, en realidad, sí la tiene. La recuperación se puede hacer porque el programa que tiene asignado el recurso le comunica al sistema operativo que ya no lo necesita o bien porque el programa haya terminado.

Los recursos manejados por el sistema operativo son físicos y lógicos. Entre los físicos se encuentra el procesador, la memoria principal y los periféricos. Entre los lógicos se pueden citar los archivos y los puertos de comunicación.

b) Protección

El SO ha de garantizar la protección entre los usuarios del sistema. Ha de asegurar la confidencialidad de la información y que unos trabajos no interfieran con otros.

Para conseguir este objetivo ha de impedir que unos programas puedan acceder a los recursos asignados a otros programas.

c) Contabilidad

La contabilidad permite medir la cantidad de recursos que, a lo largo de su ejecución, utiliza cada programa. De esta forma se puede conocer la carga de utilización que tiene cada recurso y se puede imputar a cada usuario los recursos que ha utilizado. Cuando la contabilidad se emplea meramente para conocer la carga de los componentes del sistema se suele denominar monitorización.

El SO como gestor de recursos

El SO ofrece a los programas un conjunto de servicios, o **llamadas al sistema**, que pueden solicitar cuando lo necesiten, proporcionando a los programas una visión de máquina extendida. El modelo de programación que ofrece el hardware se complementa con estos servicios software, que permiten ejecutar de forma cómoda y protegida ciertas operaciones. La alternativa consistiría en complicar los programas de usuario y en no tener protección frente a otros usuarios.

Los servicios se pueden agrupar en las cuatro clases siguientes: ejecución de programas, operaciones de E/S, operaciones sobre archivos y detección y tratamiento de errores.

a) Ejecución de programas

El SO incluye servicios para lanzar la ejecución de un programa, así como para pararlo o abortarlo. También existen servicios para conocer y modificar las condiciones de ejecución de los programas, para comunicar y sincronizar unos programas con otros.

La ejecución de programas da lugar al concepto de **proceso**. Un proceso se puede definir como un programa en ejecución. El proceso es un concepto fundamental en los sistemas operativos, puesto que el objetivo último de éstos es crear, ejecutar y destruir procesos, de acuerdo a las órdenes de los usuarios.

Para que un programa pueda convertirse en un proceso ha de estar traducido a código máquina y almacenado en un dispositivo de almacenamiento como el disco. Bajo la petición de un usuario, el sistema operativo creará un proceso para ejecutar el programa. Observe que varios procesos pueden estar ejecutando el mismo programa. Por ejemplo, varios usuarios pueden haber pedido al sistema operativo la ejecución del mismo programa editor.

b) Órdenes de E/S

Los servicios de E/S ofrecen una gran comodidad y protección al proveer a los programas de operaciones de lectura, escritura y modificación del estado de los periféricos. En efecto, la programación de las operaciones de E/S es muy compleja y dependiente del hardware específico de cada periférico. Los servicios del SO ofrecen un alto nivel de abstracción de forma que el programador de aplicaciones no tenga que preocuparse de esos detalles.

c) Operaciones sobre archivos

Los archivos ofrecen un nivel de abstracción mayor que el de las órdenes de E/S, permitiendo operaciones tales como creación, borrado, renombrado, apertura, escritura y lectura de archivos.

d) Detección y tratamiento de errores

Además de analizar detalladamente todas las ordenes que reciba, para comprobar que se pueden realizar, el SO se encarga de tratar todas las condiciones de error que detecte el hardware.

Entre las condiciones de error que pueden aparecer destacaremos las siguientes: errores en las operaciones de E/S, errores de paridad en los accesos a memoria o en los buses y errores de ejecución en los programas, como desbordamientos, violaciones de memoria, códigos de instrucción prohibidos, etc.

El SO como interfaz de usuario

El módulo del SO que permite que los usuarios dialoguen de forma interactiva con el sistema es el intérprete de mandatos o Shell. El Shell se comporta como un bucle infinito que está repitiendo constantemente la siguiente secuencia:

- Espera una orden del usuario. En el caso de interfaz textual, el Shell está pendiente de lo que escribe el usuario en la línea de mandatos. En las interfaces gráficas está pendiente de los eventos del apuntador (ratón) que manipula el usuario, además, de los del teclado.
- Analiza la orden y, en caso de ser correcta, la ejecuta, para lo cual emplea los servicios del SO.
- Concluida la orden vuelve a la espera.

El dialogo mediante interfaz textual exige que el usuario memorice la sintaxis de los mandatos, con la agravante de que son distintos para cada SO.

1.1.3 Concepto de usuario y de grupo de usuarios

Un usuario es una persona autorizada para utilizar un sistema informático. El usuario se autentica mediante su nombre de cuenta y su contraseña o password.

En realidad el SO no asocia el concepto de usuario con el de persona física sino con un nombre de cuenta. Una persona puede tener más de una cuenta y una cuenta

puede ser utilizada por más de una persona. Internamente, el SO asigna a cada usuario (cuenta) un identificador «UID» (User Identifier) y un perfil.

El sistema de seguridad de los sistemas operativos está basado en la entidad usuario. Cada usuario tiene asociados unos derechos, que definen las operaciones que le son permitidas.

Existe un usuario privilegiado, denominado **súperusuario** o **administrador**, que no tiene ninguna restricción, es decir, que puede hacer todas las operaciones sin ninguna traba.

1.2 ARRANQUE DE LA COMPUTADORA

El arranque de una computadora actual tiene dos fases: la fase de arranque hardware y la fase arranque del SO. La Figura I.2 resume las actividades más importantes que se realizan en el arranque de la computadora.

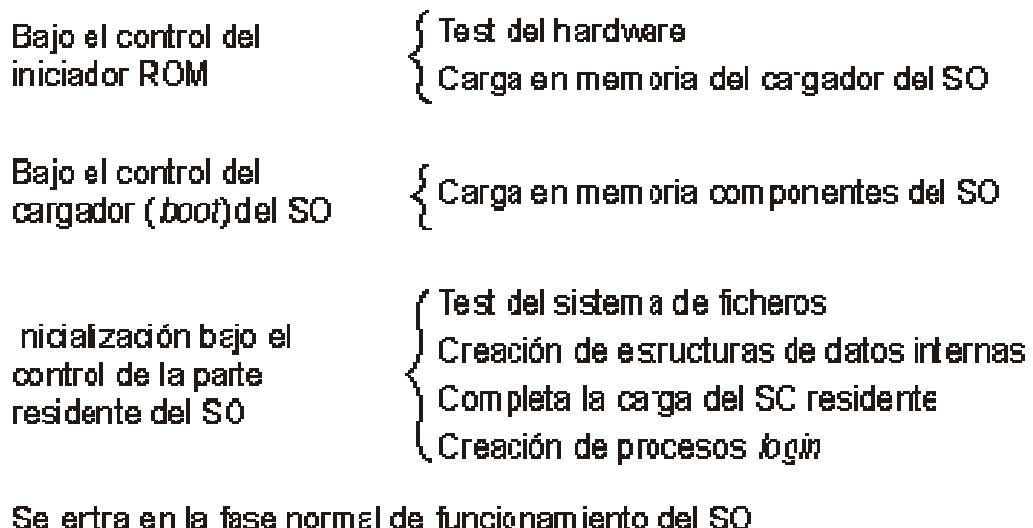


Figura I.2 Operaciones realizadas en el arranque de la computadora

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

1.2.1 Arranque hardware

Como se ha indicado con anterioridad, la computadora solamente es capaz de realizar actividades útiles si cuenta con el correspondiente programa cargado en memoria principal. Ahora bien, la memoria principal de las computadoras es volátil, lo que significa que, cuando se enciende la máquina, no contiene ninguna información válida. Por tanto, al arrancar la computadora no es capaz de realizar nada.

Para resolver esta situación, las computadoras antiguas tenían una serie de conmutadores que permitían introducir una a una las palabras en la memoria principal y en los registros. El usuario debía introducir a mano, y en binario, un primer programa que permitiese cargar otros programas almacenados en algún soporte, como la cinta de papel.

En la actualidad, la solución empleada es mucho más cómoda puesto que se basa en un programa permanente grabado en una memoria ROM. En efecto, como muestra la Figura I.3, una parte del mapa de memoria está construido con memoria ROM no volátil. En esta memoria ROM se encuentra a un programa de arranque, que está siempre disponible, puesto que la ROM no pierde su contenido. Llamemos iniciador ROM a este programa.

Cuando se arranca la computadora, o cuando se pulsa el botón de RESET, se genera una señal eléctrica que carga unos valores predefinidos en los registros. En especial, esta señal carga en el contador de programa la dirección de comienzo del iniciador ROM. De esta forma se cumplen todas las condiciones para que la computadora ejecute un programa y realice funciones útiles.

El iniciador ROM realiza tres funciones, primero hace una comprobación del sistema, que sirve para detectar sus características (por ejemplo; la cantidad de memoria principal disponible o los periféricos instalados) y comprobar si el conjunto funciona correctamente. Una vez pasada la comprobación, entra en la fase de lectura y almacenamiento en memoria del programa cargador¹ del SO.

Finalmente da control a este programa, bifurcando a la dirección de memoria en la que se ha almacenado, Para tener una mayor flexibilidad se hace que el programa iniciador ROM sea independiente del SO.

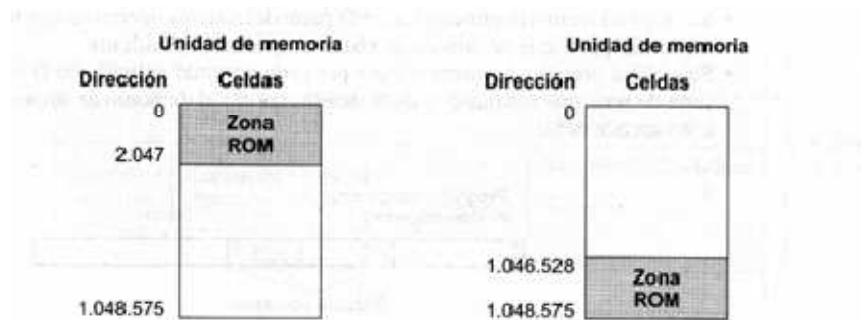


Figura I.3 Una parte del mapa de memoria está construido con memoria

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

En el caso de una computadora de tipo PC, la memoria ROM contiene, además del programa iniciador, software de E/S denominado BIOS (Basic Input-Output System). La BIOS de una computadora la proporciona el fabricante y suele contener procedimientos para leer caracteres del teclado y escribir en la pantalla.

¹ Se denomina carga a la operación combinada de leer un programa ubicado en un periférico y de almacenarlo en memoria principal. El programa que realiza la carga se llama cargador.

1.2.2 Ubicación del SO

El SO se encuentra almacenado en una unidad de disco tal y como muestra la Figura I.4. Hay una parte del mismo en la que nos interesamos en este momento, se trata del programa cargador del SO o boot del SO.

Este programa está almacenado en una zona predefinida del disco (por ejemplo: los cuatro primeros sectores del disco) y tiene un tamaño prefijado.

Como se indicó anteriormente, el iniciador ROM trae a memoria principal el programa cargado del SO. El programa iniciador ROM y el SO tienen un convenio sobre la ubicación, dirección de arranque y tamaño del cargador del SO. Obsérvese que el iniciador ROM es independiente del SO siempre que este cumpla con el convenio anterior, por lo que una máquina podrá soportar diversos sistemas operativos.

Para una mayor seguridad, el programa cargador del SO incluye, en una posición prefijada por el iniciador ROM, una contraseña. De esta forma, el iniciador ROM puede verificar que la información contenida en la zona prefijada contiene efectivamente el programa cargador de un SO.

1.2.3 Arranque del SO

El programa cargador del SO tiene por misión traer a memoria principal algunos de los componentes del SO. Una vez cargados estos componentes, se pasa a la fase de iniciación, que incluye las siguientes operaciones:

- Comprobación del sistema. Se completan las pruebas del hardware realizadas por dar ROM y se comprueba que el sistema de archivos tiene un

estado coherente. Esta operación exige revisar todos los directorios, lo que supone un largo tiempo de procesamiento.

- Se establecen las estructuras de información propias del SO, tales como tabla de procesos, las tablas de memoria y las de E/S.
- Se carga en memoria principal aquella parte del SO que ha de estar siempre en memoria, parte que se denomina **SO residente**.
- Se crea un proceso de inicio o login por cada terminal definido en el sistema, así como una serie de procesos auxiliares y de demonios (por ejemplo: el demonio de impresión o el demonio comunicaciones).

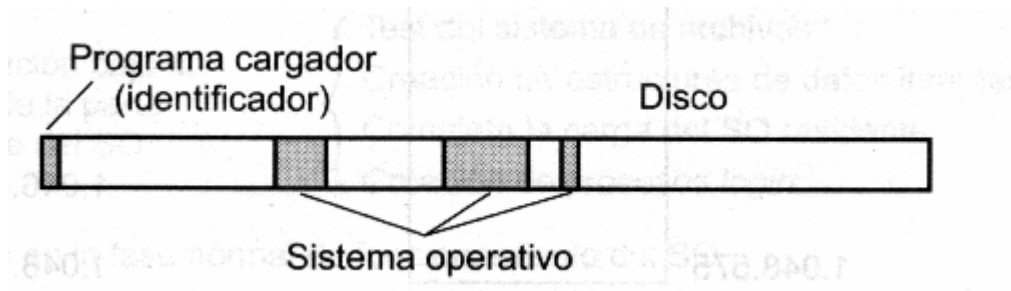


Figura I.4 El SO se encuentra almacenado en una unidad de disco

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

Los procesos de inicio presentan en su terminal el mensaje de bienvenida y se quedan a la espera de que un usuario inicie una sesión, para lo cual ha de teclear el nombre de su cuenta y su contraseña o password. El proceso de inicio **autentica** al usuario, comprobando que los datos introducidos son correctos y lanza un proceso Shell². El proceso Shell primero ejecuta uno o varios archivos de mandatos, como es el «autoexec.bat» en MS-DOS o los «.login» y «.cshrc» en UNIX. A continuación, el Shell se queda esperando órdenes de los usuarios, ya sean textuales o como

² En algunos de los sistemas operativos utilizados en las computadoras personales no hay fase de login, creándose directamente el proceso Shell para atender al usuario.

acciones sobre un menú o un icono. Para llevar a cabo las operaciones solicitadas por el usuario, el Shell genera uno o varios procesos.

1.3 COMPONENTES Y ESTRUCTURA DEL SO

El SO está formado por una serie de componentes especializados en determinadas funciones. Cada SO estructura estos componentes de forma distinta. En esta sección se describen en primer lugar los distintos componentes que conforman un SO, para pasar a continuación a ver las distintas formas que tienen los sistemas operativos de estructurar estos componentes.

1.3.1 Componentes del SO

Como se comentó previamente y se muestra en la Figura I.5, se suele considerar que un SO está formado por tres capas: el núcleo, los servicios y el intérprete de mandatos o Shell.

El núcleo es la parte del SO que interacciona directamente con el hardware de la máquina. Las funciones del núcleo se centran en la gestión de recursos, como el procesador, tratamiento de interrupciones y las funciones básicas de manipulación de memoria.

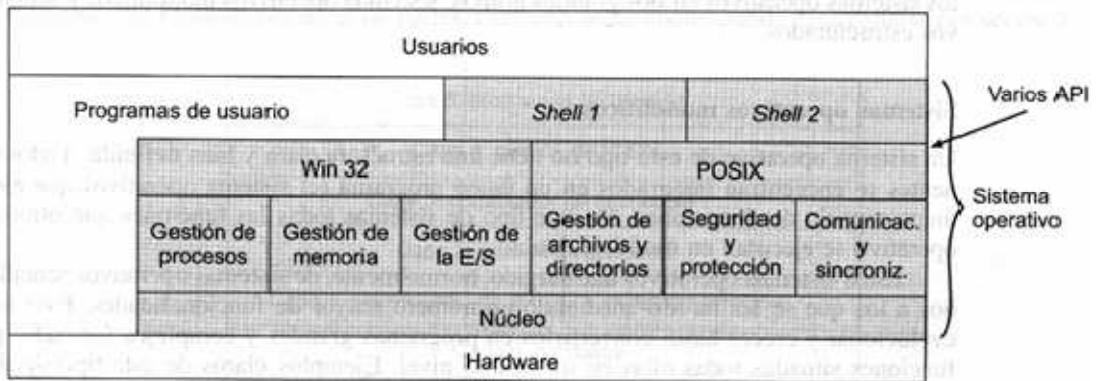


Figura I.5 Componentes del SO

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

Los servicios se suelen agruparse según su funcionalidad en varios componentes, cada uno de cuales se ocupa de las siguientes funciones:

- Gestión de procesos. Encargada de la creación, planificación y destrucción de procesos.
- Gestión de memoria. Componente encargada de saber qué partes de memoria están libres y cuáles ocupadas, así como de la asignación y liberación de memoria según la necesiten los procesos.
- Gestión de la E/S. Se ocupa de facilitar el manejo de los dispositivos periféricos.
- Gestión de archivos y directorios. Se encarga del manejo de archivos y directorios y de la administración del almacenamiento secundario.
- Comunicación y sincronización en los procesos. Encargada de ofrecer mecanismos para que los procesos puedan comunicarse y sincronizarse.
- Seguridad y protección. Este componente debe encargarse de garantizar seguridad a los usuarios y de definir lo que pueden hacer cada uno de ellos con los recursos del sistema.

Todos estos componentes ofrecen una serie de servicios a través de una interfaz de llamadas al sistema. Como se muestra en la Figura 1.5, un SO puede incluir más de una interfaz de servicios (en la figura se han considerado las interfaces Win32 y POSIX). En este caso, los programas podrán elegir sobre qué interfaz quieren ejecutar, pero no podrán mezclar servicios de varias interfaces. Se dice, en este caso, que el SO presenta al usuario varias máquinas virtuales.

De igual forma, el sistema operativo puede incluir varios intérpretes de mandatos, unos textuales y otros gráficos, pudiendo el usuario elegir el que más le interese. Sin embargo, hay que observar que no se podrán mezclar mandatos de varios intérpretes.

1.3.2 Estructura del SO

Un SO es un programa grande y complejo que está compuesto, como se ha visto en la sección anterior, por una serie de componentes con funciones bien definidas.

Cada SO estructura estos componentes de distinta forma. En función de esta estructura se pueden agrupar los sistemas operativos en dos grandes grupos: sistemas operativos monolíticos y sistemas operativos estructurados.

Sistemas Operativos Monolíticos

Un SO de este tipo no tiene una estructura clara y bien definida. Todos sus componentes se encuentran integrados en un único programa (el SO) que se ejecuta en un único espacio de direcciones. En este tipo de sistemas todas las funciones que ofrece el SO se ejecuta en modo núcleo.

Estos sistemas operativos han surgido, normalmente, de sistemas operativos sencillos y pequeños a los que se les ha ido añadiendo un número mayor de funcionalidades. Esto les ha hecho evolucionar y crece hasta convertirlos en programas grandes y complejos formados por muchas funciones situadas todas ellas en un mismo nivel. Ejemplos claros de este tipo de sistemas son MS-DOS y UNIX. Ambos comenzaron siendo pequeños sistemas operativos, que fueron haciéndose cada vez más grandes debido a la gran popularidad que adquirieron.

El problema que se tiene en este tipo de sistemas radica en lo complicado que es modificar el SO para añadir nuevas funcionalidades y servicios.

En efecto, añadir una nueva característica al SO implica la modificación de un gran programa, compuesto por miles de líneas de código fuente y funciones, cada una de las cuales puede invocar a otras cuando así lo requiera. Además, en este tipo de sistemas no se sigue el principio de ocultación de la información. Para solucionar este problema es necesario dotar de cierta estructura al SO.

Sistemas Operativos Estructurados

Cuando se quiere dotar de estructura a un SO, normalmente se recurre a dos tipos de soluciones: sistemas por capas y sistemas cliente-servidor.

a) Sistemas por capa

En un sistema por capas, el SO se organiza como una jerarquía de capas, donde cada capa ofrece una interfaz clara y bien definida a la capa superior y solamente utiliza los servicios que le ofrece la capa inferior.

La principal ventaja que ofrece este tipo de estructuras es la modularidad y la ocultación de la información. Una capa no necesita conocer como se ha implementado la capa sobre la que se construye, únicamente necesita conocer la interfaz que ofrece.

Esto facilita enormemente la depuración y verificación del sistema, puesto que las capas se pueden ir construyendo y depurando por separado.

Un ejemplo de SO diseñado por capas es el OS/2, descendiente de MS-DOS. Tal como se muestra en la Figura I.6.

Capa 5: Programas de usuario
Capa 4: Gestión de la E/S
Capa 3: Controlador de la consola
Capa 2: Gestión de memoria
Capa 1: Planificación de la CPU y multiprogramación
Capa 0: hardware

Figura I.6 Estructura por capas de un Sistema Operativo

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

b) Modelo cliente-servidor

En este tipo de modelo, el enfoque consiste en implementar la mayor parte de los servicios y funciones del SO en procesos de usuario, dejando solo una pequeña parte del SO ejecutando en modo núcleo. A esta parte se le denomina *micronúcleo* y a los procesos que ejecutan el resto de funciones se les denomina *servidores*. La Figura I.7 presenta la estructura de un SO con estructura cliente-servidor, Como puede apreciarse en la figura, el SO está formado por diversas partes, cada una de las cuales puede desarrollarse por separado.

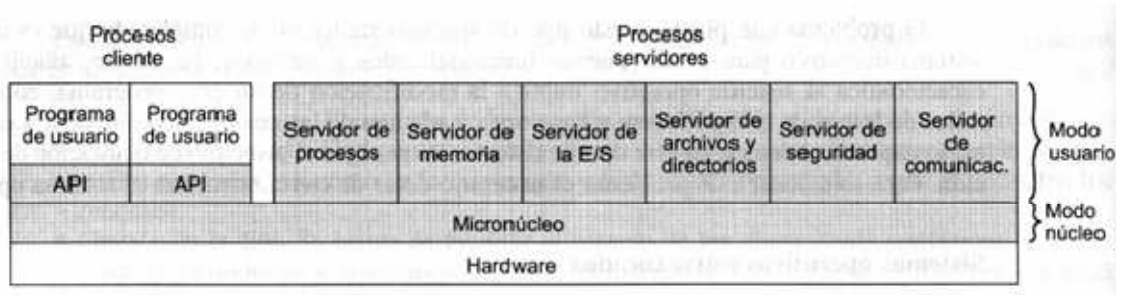


Figura I.7 Estructura cliente-servidor en un Sistema Operativo

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

No hay una definición clara de las funciones que debe llevar a cabo un micronúcleo. La mayoría incluyen la gestión de interrupciones, gestión básica de procesos y de memoria y servicios básicos de comunicación entre procesos. Para solicitar un servicio en este tipo de sistemas, como por ejemplo crear un proceso, el proceso de usuario (proceso denominado cliente) solicita el servicio al servidor del SO correspondiente, en este caso al servidor de procesos. A su vez, el proceso servidor puede requerir los servicios de otros servidores, como es el caso del servidor de memoria. En este caso, el servidor de procesos se convierte en cliente del servidor de memoria.

La ventaja de este modelo es la gran flexibilidad que presenta. Cada proceso servidor sólo ocupa de una funcionalidad concreta, lo que hace que cada parte pueda ser pequeña y esto a su vez facilita el desarrollo y depuración de cada uno de los procesos servidores.

En cuanto a las desventajas, cabe citar que estos sistemas presentan una mayor sobrecarga en el tratamiento de los servicios que los sistemas monolíticos. Esto se debe a que los distintos componentes de un SO de este tipo se ejecutan en espacios de direcciones distintos, lo que hace que su activación requiera más tiempo.

1.4 GESTIÓN DE PROCESOS

El componente principal de un SO es el que se encarga de la gestión de procesos. El proceso es un elemento central en los sistemas operativos, puesto que su función consiste en generar y gestionar los procesos y en atender a sus peticiones.

Como se indicó anteriormente, el proceso se puede definir como un **programa en ejecución**. De forma un poco más precisa, se puede definir el proceso como la unidad de procesamiento gestionada por el SO. No hay que confundir el concepto de programa con el concepto de proceso. Un programa no es más que un conjunto de instrucciones máquina, mientras que el proceso surge cuando un programa se pone en ejecución. Esto hace que varios procesos puedan ejecutar el mismo programa a la vez (por ejemplo: que varios usuarios estén ejecutando el mismo editor textos).

Para que un programa pueda ser ejecutado, ha de residir con sus datos en memoria principal, tal y como muestra la Figura I.8. Al contenido de los segmentos de memoria en los que reside el código y los datos del proceso se le denomina **imagen de memoria**. Observe que, durante su ejecución, el proceso va modificando los registros del modelo de programación de la computadora, de acuerdo a las instrucciones de máquinas involucradas. El contenido de los registros del modelo de programación es lo que se conoce como **estado del procesador**.

El SO mantiene por cada proceso una serie de estructuras de información que permite identificar las características de éste así como los recursos que tiene asignados. Una parte muy importante de esta estructura es el bloque de control del proceso (**BCP**) que incluye, entre otra información, el estado de los registros del proceso, cuando éste no se está ejecutando.

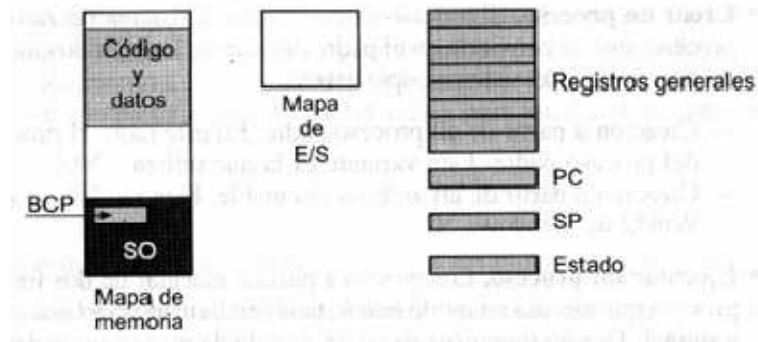


Figura I.8 Elementos que constituyen un proceso

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

El SO debe encargarse también de ofrecer una serie de servicios para la gestión de procesos y de gestionar los posibles interbloqueos que surgen cuando los procesos acceden a diferentes recursos. Dependiendo del número de procesos y de usuarios que puedan ejecutar simultáneamente, un SO puede ser:

- Monotarea, también llamado monoproceso. Este tipo de sistemas operativos sólo permite que exista un proceso en cada instante.
- Multitarea o multiproceso. Permite que coexistan varios procesos activos a la vez. El SO se encarga de ir repartiendo el tiempo del procesador entre estos procesos.
- Monousuario. Está previsto para soportar a un solo usuario.
- Multiusuario. Soporta varios usuarios trabajando simultáneamente desde varios terminales. A su vez cada usuario puede tener activo más de un proceso, por lo que el sistema, obligatoriamente, ha de ser multitarea. Los sistemas multiusuario reciben también el nombre de tiempo compartido, porque el SO ha de repartir el tiempo de la computadora entre los usuarios para que las tareas de todos ellos avancen de forma razonable.

1.4.1 Servicios de Procesos

El SO ofrece una serie de servicios que permiten definir la vida de un proceso. Esta vida está constituida por las siguientes fases: creación, ejecución y muerte del proceso.

En general, los sistemas operativos ofrecen los siguientes servicios para la gestión de procesos:

- **Crear un proceso.** El proceso es creado por el SO cuando así lo solicita otro proceso, que se convierte en el padre del nuevo. Existen dos modalidades básicas para crear un proceso en los sistemas operativos:
 - Creación a partir de un proceso padre. En este caso, el proceso hijo es una copia exacta del proceso padre. Esta variante es la que utiliza UNIX.
 - Creación a partir de un archivo ejecutable. Esta modalidad es la que se define en el API Win32 de Windows NT.
- **Ejecutar un proceso.** Los procesos se pueden ejecutar de dos formas: batch e interactiva. Un proceso que se ejecuta en modo batch, también llamado background, no está asociado a ningún terminal. Deberá tomar sus datos de entrada de un archivo y deberá depositar sus resultados en otro archivo. Un ejemplo típico de un proceso batch es un proceso de nóminas, que parte del archivo de empleados y del archivo de los partes de trabajo y genera un archivo de órdenes básicas para pagar las nóminas.

Por el contrario, un proceso que se ejecuta en modo interactivo está asociado a un terminal, por el que recibe la información del usuario y por el que contesta con los resultados. Un ejemplo típico de un proceso interactivo es un proceso de edición.

- **Terminar la ejecución de un proceso.** Un proceso puede finalizar su ejecución por varias causas, entre las que se encuentran las siguientes:
 - Ha terminado de ejecutar el programa.
 - Se produce una condición de error en su ejecución (por ejemplo: división por 0 o violación de memoria).
 - Otro proceso o el usuario deciden que ha de terminar.
- **Cambiar el programa de un proceso.** Algunos sistemas operativos incluyen, además de los anteriores, un servicio que cambia el programa que está ejecutando un proceso por otro programa almacenado en disco, observe que esta operación no consiste en crear un nuevo proceso que ejecuta ese nuevo programa. Se trata de eliminar el programa que está ejecutando el proceso y de incluir un nuevo programa que se trae del disco.

1.5 GESTIÓN DE MEMORIA

El gestor de memoria es uno de los componentes principales del SO. Su actividad se centra fundamentalmente en la categoría de gestión de recursos, puesto que tiene por objetivo casi exclusivo la gestión del recurso memoria. En este sentido se encarga de:

- Asignar memoria a los procesos para crear su imagen de memoria.
- Proporcionar memoria a los procesos cuando la soliciten y liberarla cuando así lo requieran.
- Tratar los posibles errores de acceso a memoria, evitando que unos procesos interfieran en la memoria de otros.
- Permitir que los procesos puedan compartir memoria entre ellos. De esta forma los procesos podrán comunicarse entre ellos.

- Gestionar la jerarquía de memoria y tratar los fallos de página en los sistemas con memoria virtual.

Además de estas funciones, el gestor de memoria, en la categoría de servicios a los programas, suministra los siguientes servicios: el de solicitar memoria, el de liberarla y el de permitir que los procesos compartan memoria. Los dos primeros servicios son necesarios para los programas que requieren asignación dinámica de memoria, puesto que, en este caso, la imagen de memoria ha de crecer o decrecer de acuerdo a las necesidades de ejecución. El tercer servicio es necesario cuando los procesos desean compartir segmentos de memoria para intercambiar datos entre sí.

1.5.1 Servicios

El gestor de memoria ofrece una serie de servicios a los procesos. Estos son:

- **Solicitar memoria.** Este servicio aumenta el espacio de datos de la imagen de memoria del proceso. El SO atenderá la petición siempre y cuando cuente con los recursos necesarios para ello. En general, el SO devuelve un apuntador con la dirección de la nueva memoria. El programa utilizará este nuevo espacio a través del mencionado apuntador, mediante direccionamientos relativos al mismo.
- **Liberar memoria.** Este servicio sirve para devolver trozos de la memoria del proceso. El SO recupera el recurso liberado y lo añade a sus listas de recursos libres, para su posterior reutilización.
- **Compartir memoria.** Dentro de esta categoría, el gestor de memoria se encarga de ofrecer servicios que permiten que los procesos puedan

comunicarse utilizando un segmento de memoria compartida. Para ello se permite que los procesos creen y liberen este tipo de segmentos.

1.6 COMUNICACIÓN Y SINCRONIZACIÓN ENTRE PROCESOS

Los procesos son entes independientes y aislados, puesto que, por razones de seguridad, no deben interferir unos con otros. Sin embargo, cuando se divide un trabajo complejo en varios procesos que cooperan entre sí para realizar ese trabajo, es necesario que se comuniquen para transmitirse datos y órdenes y se sincronicen en la ejecución de sus acciones. Por tanto, el SO debe incluir servicios de comunicación y sincronización entre procesos que, sin romper los esquemas de seguridad, han de permitir la cooperación entre ellos.

El SO ofrece una serie de mecanismos básicos de comunicación que permiten transferir cadenas de bytes, pero han de ser los procesos que se comunican los que han de interpretar la cadena de bytes transferida. En este sentido, se han de poner de acuerdo en la longitud de la información y en los tipos de datos utilizados.

Dependiendo del servicio utilizado, la comunicación se limita a los procesos de una máquina (procesos locales) o puede involucrar a procesos de máquinas distintas (procesos remotos). La Figura I.9 muestra ambas situaciones.

El SO ofrece también mecanismos que permiten que los procesos esperen (se bloqueen) y se despierten (continúen su ejecución) dependiendo de determinados eventos.

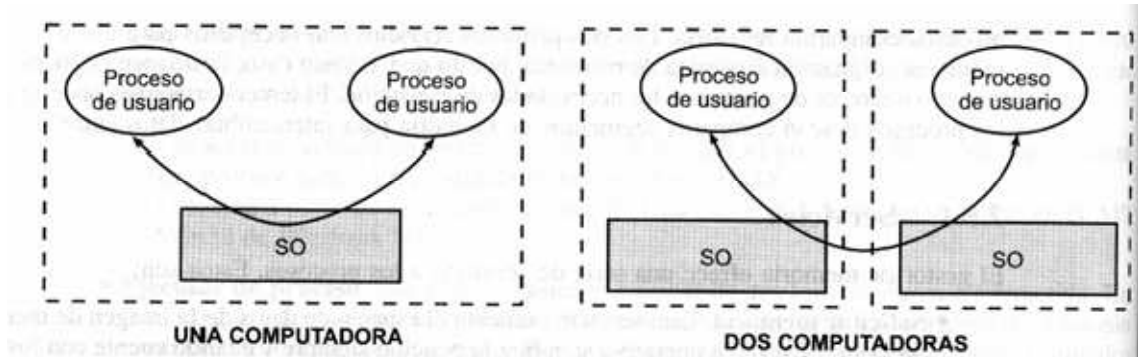


Figura I.9 Comunicación entre procesos locales y remotos

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

1.6.1 Servicios de Comunicación y Sincronización

Como se ha visto anteriormente, existen distintos mecanismos de comunicación y sincronización, cada uno de los cuales se puede utilizar a través de un conjunto de servicios propios. Estos mecanismos son entidades vivas, cuya vida presenta las siguientes fases:

- Creación del mecanismo.
- Utilización del mecanismo.
- Destrucción del mecanismo.

De acuerdo con esto, los servicios básicos de comunicación, que incluyen todos los mecanismos de comunicación, son los siguientes:

- **Crear.** Permite que el proceso solicite la creación del mecanismo.
- **Enviar o escribir.** Permite que el proceso emisor envíe información a otro.
- **Recibir o leer.** Permite que el proceso receptor reciba información de otro,
- **Destruir.** Permite que el proceso solicite la creación o destrucción del mecanismo.

Por otro lado, la comunicación puede ser síncrona o asíncrona. En la comunicación síncrona los dos procesos han de ejecutar los servicios de comunicación al mismo tiempo, es decir, el emisor ha de estar en el servicio de enviar y el receptor ha de estar en el servicio de recibir, Normalmente, para que esto ocurra, uno de ellos ha de esperar a que el otro llegue a la ejecución del correspondiente servicio (Figura I.10).

En la comunicación asíncrona el emisor no tiene que esperar a que el receptor solicite el servicio recibir, hace el envío y sigue con la ejecución. Esto obliga a que el SO establezca un almacenamiento intermedio para guardar la información enviada hasta que el receptor la solicite.

En cuanto a los mecanismos de sincronización, los mecanismos suelen incluir los siguientes servicios:

- **Crear.** Permite que el proceso solicite la creación del mecanismo.
- **Bloquear.** Permite que el proceso se bloquee hasta que ocurra un determinado evento.
- **Despertar.** Permite despertar a un proceso bloqueado.
- **Destruir.** Permite que el proceso solicite la destrucción del mecanismo.

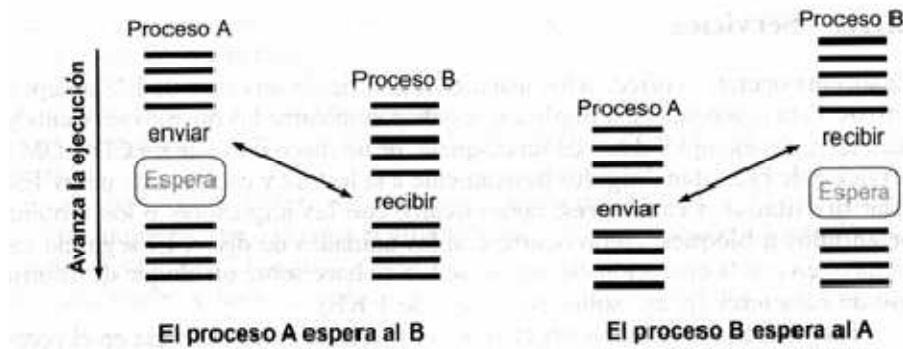


Figura I.10 Comunicación síncrona entre procesos

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

1.7 GESTIÓN DE E/S

Una de las principales funciones de un SO es la gestión de los recursos de la computadora y, en concreto, de los dispositivos periféricos. El gestor de E/S debe controlar el funcionamiento de todos los dispositivos de E/S para alcanzar los siguientes objetivos:

- Facilitar el manejo de los dispositivos periféricos. Para ello debe ofrecer una interfaz sencilla, uniforme y fácil de utilizar entre los dispositivos, y gestionar los errores que se pueden producir en el acceso a los mismos.
- Ofrecer mecanismos de protección que impidan a los usuarios acceder sin control a los dispositivos periféricos.

Dentro de la gestión de E/S, el SO debe encargarse de gestionar los distintos dispositivos de E/S; relojes, terminales, dispositivos de almacenamiento secundario y terciario, teclado, etc.

1.7.1 Servicios

El SO ofrece a los usuarios una serie de servicios de E/S independiente de los dispositivos. Esta independencia implica que deben emplearse los mismos servicios y operaciones de E/S para leer, por ejemplo, datos de un disquete, de un disco duro, de un CD-ROM o de un teclado. Los servicios de E/S están dirigidos básicamente a la lectura y escritura de datos.

Estos servicios pueden estar **orientados a caracteres**, como ocurre con las impresoras o los terminales, o pueden estar **orientados a bloques**, como ocurre con las unidades de disco. El segundo caso se diferencia del primero en que la operación elemental de E/S se hace sobre un bloque de información de un número fijo de caracteres (por ejemplo: sobre un bloque de 1 KB).

En general, los sistemas operativos consiguen la independencia en el acceso a los dispositivos modelándolos como archivos especiales.

1.8 GESTIÓN DE ARCHIVOS Y DIRECTORIOS

El servidor de archivos es la parte del SO que cubre una de las cuatro clases de funciones que tiene éste en su faceta de máquina extendida. Los objetivos fundamentales del servidor de archivos son los dos siguientes:

- Facilitar el manejo de los dispositivos periféricos. Para ello ofrece una visión lógica simplificada de los mismos en forma de chivos.
- Proteger a los usuarios, poniendo limitaciones a los chivos que es capaz de manipular cada usuario.

Los servicios que se engloban en el servidor de archivos son de dos tipos: los servicios dirigidos al manejo de datos, o archivos, y los dirigidos al manejo de los nombres, o directorios.

El servidor de archivos ofrece al usuario (Figura I.11) una **visión lógica** compuesta por una serie de objetos (archivos y directorios) identificables por un nombre lógico sobre los que puede realiza una serie de operaciones.

La **visión física** ha de incluir los detalles de cómo están almacenados estos objetos en los periféricos correspondientes (por ejemplo: en los discos).

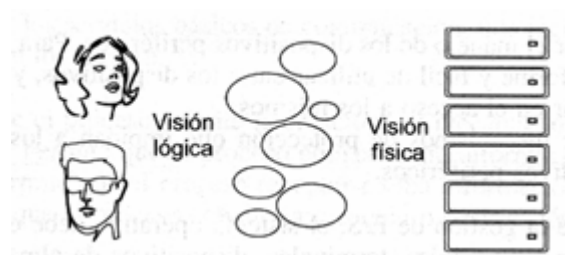


Figura I.11 Visión lógica y física del sistema de archivos

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

1.8.1 Servicio de Archivos

Un archivo es una unidad de almacenamiento lógico no volátil que agrupa un conjunto de información relacionada entre sí bajo un mismo nombre. Cada archivo tiene una información asociada que utilizan tanto los usuarios como el propio servidor de archivos. Entre las informaciones más usuales se pueden destacar las siguientes:

- Tipo de archivo (por ejemplo: archivo de datos, ejecutables, etc.).

- Propietario del archivo (identificador de usuario que creó el archivo y del grupo de dicho usuario).
- Tamaño del archivo. Este tamaño suele ser menor que el espacio de disco asignado al archivo, puesto que es muy raro que el último bloque se llene completamente. Por término medio queda sin usarse 1/2 bloque.
- Instantes (fecha y hora) importantes de la vida del archivo, como son los siguientes:
 - Instante en que se creó.
 - Instante de la última modificación.
 - Instante del último acceso.
- Derechos de acceso al archivo (sólo lectura, lectura-escritura, sólo escritura, ejecución, etc.).

Las operaciones sobre archivos que ofrece el servidor de chivos están referidas a la visión lógica de los archivos. La solución más común es que el archivo se visualice como un vector de bytes o caracteres, tal y como indica la Figura I.12. Algunos sistemas de archivos ofrecen visiones lógicas más elaboradas, orientadas a que pueden ser de longitud fija o variable. La ventaja de la sencilla visión de vector de caracteres es su flexibilidad, puesto que no presupone ninguna estructura específica interna en el archivo.

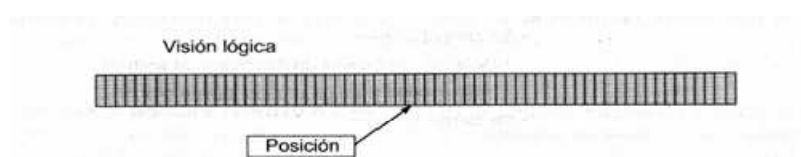


Figura I.12 Visión lógica de un archivo

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

La visión lógica del archivo incluye normalmente un **puntero de posición**. Este puntero permite hacer operaciones de lectura y escritura consecutivas sin tener que indicar la posición de la operación. Inicialmente el puntero indica la posición 0, pero después de hacer, por ejemplo, una operación de lectura de 7.845 bytes señalará a la posición 7.845. Otra lectura posterior se referirá a los bytes 7.845, 7.846, etc.

La visión física está formada por los elementos físicos del periférico que soportan al archivo. En el caso más usual de tratarse de discos, la visión física consiste en la enumeración ordenada de los bloques de disco en los que reside el archivo. La Figura I.13 muestra un ejemplo en el que el archivo A está formado, en ese orden, por los bloques 12, 20, 1, 8, 3, 16 y 19.

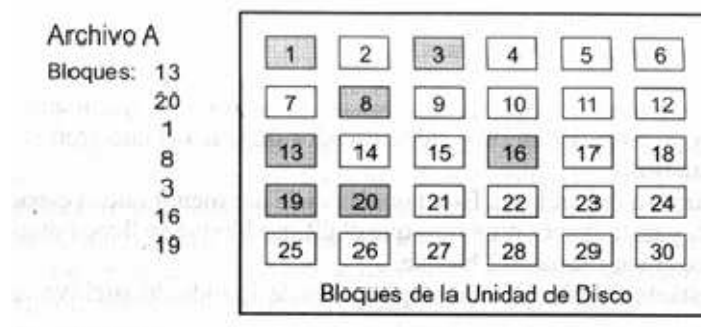


Figura I.13 Visión física de un archivo

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

Observe que debe existir una estructura de información que recoja la composición física de cada archivo, que se denominará de forma genérica "*descripción física del archivo*". Esta estructura es FAT en MS-DOS y nodo-i en UNIX, es de destacar que estas estructuras de información han de residir en el propio periférico (por ejemplo: disco), para que éste sea auto contenido y se pueda transportar de un sistema a otro.

El servidor de archivos es capaz de encontrar e interpretar estas estructuras de información liberando a los programas de usuario de esta tediosa labor.

Servicios de archivos

Un archivo es una entidad viva, que va evolucionando de acuerdo a los servicios que se solicitan del SO.

Los servicios que ofrece el servidor de archivos son los siguientes:

- **Crear un archivo.** Este servicio crea un archivo vacío. La creación de un archivo exige una interpretación del nombre, puesto que el servidor de archivos ha de comprobar que el nombre es correcto y que el usuario puede hacer la operación solicitada. La creación de un archivo deja abierto a éste devolviendo al usuario un identificador, descriptor o manejador de archivo de carácter temporal para su manipulación.
- **Abrir un archivo.** Un archivo debe ser abierto antes de ser utilizado. Este servicio comprueba que el archivo existe, que el usuario tiene derechos de acceso y trae a memoria información del objeto para optimizar el acceso al mismo. Además devuelve al usuario un identificador, descriptor o manejador de archivo de carácter temporal para su manipulación. Normalmente, todos los sistemas operativos tienen un límite máximo para el número de archivos que puede tener abierto un usuario.
 - Se crea el archivo.
 - Se abre: se genera un descriptor de archivo.
 - Se escribe y lee (el archivo puede crecer).
 - Se cierra y se borra.

- **Escribir y leer.** Estos servicios se realizan utilizando el identificador, descriptor o manejador de archivo (devuelto en las operaciones de creación y apertura), en vez del nombre lógico del mismo. Una operación de lectura permite traer datos del archivo a memoria. Para ello se especifica el identificador, descriptor o manejador obtenido en la apertura, la posición de memoria y la cantidad de información a leer. Normalmente, se lee a partir de la posición que indica el puntero de posición del archivo. Las operaciones de escritura permiten llevar datos situados en memoria al archivo. Para ello, y al igual que en las operaciones de lectura, se debe especificar el identificador obtenido en la creación o apertura, la posición de memoria y la cantidad de información a escribir. Normalmente se escribe a partir de la posición que indica el puntero de posición del archivo. Si está en medio, se sobrescribirán los datos, Si está al final del archivo, su tamaño crece. En este caso, el SO se encarga de hacer crecer el espacio físico del archivo añadiendo bloques libres. En algunos sistemas operativos se puede especificar la posición del archivo en la que se realizará la siguiente lectura o escritura.
- **Cerrar un archivo.** Terminada la utilización del archivo se debe cerrar, con lo que se elimina el identificador temporal obtenido en la apertura o creación y se liberan los recursos de memoria que ocupa el archivo.
- **Borrar un archivo.** El archivo se puede borrar, lo que supone que se borra su nombre del correspondiente directorio y que el sistema de archivos ha de recuperar los bloques de datos y el espacio de metainformación³ que tenía asignado.

³ La metainformación de un archivo se refiere a toda la información auxiliar que es necesario mantener para ofrecer la visión lógica de un archivo. Esta información incluye entre otros los bloques que ocupa el archivo en disco.

1.8.2 Servicio de Directorios

Un directorio es un objeto que relaciona de forma unívoca un nombre con un archivo. El servicio de directorios sirve para identificar a los archivos (objetos), por tanto ha de garantizar que la relación [nombre → archivo] sea unívoca. Es decir, un mismo nombre no puede identificar a dos archivos. Observe, por el contrario, que el que varios nombres se refieran al mismo archivo no presenta ningún problema, son simples sinónimos.

El servicio de directorios también presenta una **visión lógica** y una **visión física**. La visión lógica consiste en el bien conocido esquema jerárquico de nombres mostrado en la Figura I.14.

Se denomina **directorio raíz** al primer directorio de la jerarquía, recibiendo los demás el nombre de subdirectorios. El directorio raíz se representa por el carácter ("/" o "~", dependiendo del SO. En la Figura I.14, el directorio raíz incluye los siguientes nombres de subdirectorios: Textos, Div11 y Div2.

Se diferencia el **nombre relativo** o **local**, que es el nombre asignado al archivo dentro del subdirectorio en el que está el archivo, del nombre o camino absoluto, que incluye todos los nombres de todos los subdirectorios que hay que recorrer desde el directorio raíz hasta el objeto considerado, concatenados por el símbolo "/" o "~". Un ejemplo del nombre relativo es "Apl1", mientras que su nombre absoluto es "/Textos/Tipo/Sec1/Apl1".

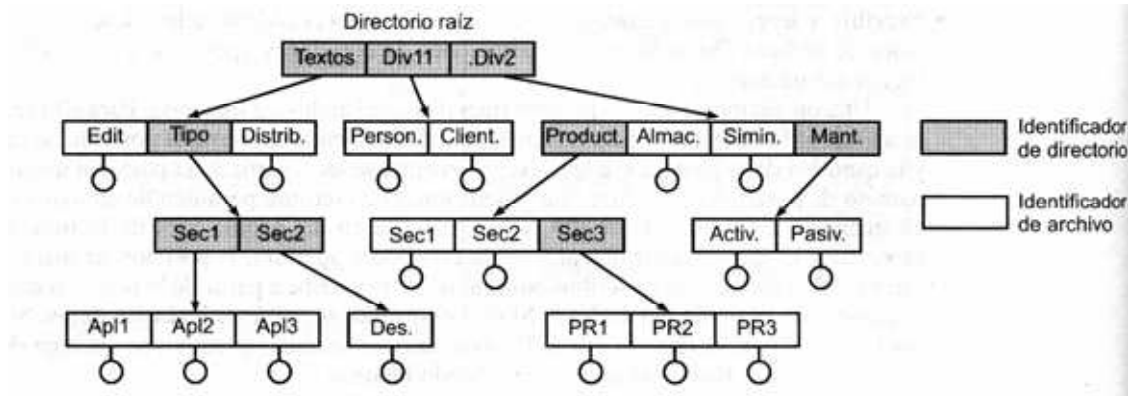


Figura I.14 Esquema jerárquico de directorios

Fuente: McGraw Hill - Sistemas Operativos, una visión aplicada

La ventaja del esquema jerárquico es que permite una gestión distribuida de los nombres, garantizando de forma sencilla que no existan nombres repetidos. En efecto, hasta con que los nombres relativos de cada subdirectorio sean distintos, aunque los nombres relativos de subdirectorios distintos sean iguales, para que no exista duplicación de nombres, puesto que quedarán diferenciados por el camino hasta llegar al correspondiente subdirectorio.

La **visión física** del sistema de directorios consiste en unas estructuras de información que permiten relacionar cada nombre lógico con la descripción física del correspondiente archivo. En esencia, se trata de una tabla NOMBRE-IDENTIFICADOR por cada subdirectorio. El NOMBRE no es más que el nombre relativo del archivo, mientras que el IDENTIFICADOR es una información que permite localizar la descripción física del archivo.

Un objeto directorio es básicamente un conjunto de entradas que relacionan nombres y archivos. El servidor de archivos incluye una serie de servicios que permiten manipular directorios.

Estos son:

- **Crear un directorio.** Crea un objeto directorio y lo sitúa en el árbol de directorios donde se especifique el nombre absoluto o relativo del nuevo directorio.
- **Borrar un directorio.** Elimina un objeto directorio de forma que nunca más pueda ser accesible y borra su entrada del árbol de directorios. Normalmente, sólo se pueden borrar directorios vacíos, es decir, un directorio sin entradas.
- **Abrir un directorio.** Abre un directorio para leer los datos del mismo. Al igual que un archivo, un directorio debe ser abierto para poder acceder a su contenido. Esta operación vuelve al usuario un identificador, descriptor o manejador de directorio de carácter temporal que permite su manipulación.
- **Leer un directorio.** Extrae la siguiente entrada de un directorio abierto previamente. Devuelve una estructura de datos como la que define la entrada de directorios.
- **Cerrar un directorio.** Cierra un directorio, liberando el identificador devuelto en la operación de apertura así como los recursos de memoria y del SO relativo al mismo.

1.8.3 Sistema de Archivos

Se denomina sistema de archivos al conjunto de archivos incluidos en una unidad de disco. El sistema de archivos está compuesto por los datos de los archivos, así como por toda la información auxiliar que se requiere.

Se denomina metainformación a toda la información auxiliar que es necesario mantener en un volumen. Resumiendo y completando lo visto en las secciones anteriores, la metainformación está compuesta por los siguientes elementos:

- Estructura física de los archivos (nodos-i de UNIX o FAT de MS-DOS).
- Directorios (archivos que contienen las tablas nombre-puntero).
- Estructura física del sistema de archivos (superbloque en UNIX).
- Estructura de información de bloques y nodos-i libres (mapas de bits).

Cada SO organiza las particiones de disco de una determinada forma, repartiendo el espacio disponible entre: el programa de carga (boot) del SO, la metainformación y los datos. Normalmente las tablas de los subdirectorios se almacenan como archivos por lo que compiten por los bloques de datos con los archivos de datos.

1.9 SEGURIDAD Y PROTECCIÓN

La seguridad reviste dos aspectos, uno es garantizar la identidad de los usuarios y el otro es definir lo que puede hacer cada uno de ellos. El primer aspecto se trata bajo el término de autenticación, mientras que el segundo se hace mediante los privilegios. La seguridad es una de las funciones del SO que, para llevarla a cabo, se ha de basar en los mecanismos de protección que le proporciona el hardware.

Autenticación

El objetivo de la autenticación es determinar que un usuario (persona, servicio o computadora) es quien dice ser. El SO dispone de un módulo de autenticación que se encarga de decidir la identidad de los usuarios. En la actualidad, las contraseñas (passwords) son el método más utilizado como mecanismo de autenticación.

Privilegios

Los privilegios especifican los recursos a los que puede acceder cada usuario. Para simplificar la información de privilegios es corriente organizar a los usuarios en grupos, asignando determinados privilegios a cada grupo.

La información de los privilegios se puede asociar a los recursos o a los usuarios:

- **Información por recurso.** En este caso se asocia la denominada lista de control de acceso (**ACL**, Access Control List) a cada recurso. Esta lista especifica los grupos y usuarios que pueden acceder al recurso.
- **Información por usuario.** Se asocia a cada usuario o grupo la lista de recursos que puede acceder, lista que se llama de capacidades (capabilities).

Dado que hay muchas formas de utilizar un recurso, la lista de control de acceso o la de capacidades, han de incluir el modo en que se puede utilizar el recurso.

Ejemplos de modos de utilización son los siguientes: leer, escribir, ejecutar, eliminar, test, control y administrar.

Los servicios relacionados con la seguridad y la protección se centran en la capacidad para asignar atributos de seguridad a los usuarios y a los recursos.

CAPITULO II

DISKLESS BOOTING Y SISTEMAS OPERATIVOS DE RED

2.1 DISKLESS BOOTING

Es un tipo de sistema que nos permite servir a otros computadores lo necesario para que arranquen un SO sin tener ni siquiera disco duro. Sino tan solo con acceso a la red donde está el servidor y la posibilidad de arrancar desde la red en el cliente. Existen varias razones por las cuales se hace esto. Una de ellas es la de reducir los costes de mantenimiento del software en una gran cantidad de máquinas.

Mediante un arranque por red los ficheros son mantenidos por un servidor central, con la ventaja de poder ser actualizados exclusivamente sobre esta máquina. Otra razón es la de usar ordenadores en lugares donde los discos duros no son suficientemente resistentes, como podría ser en la planta de una fabrica en la que éstos pueden ser relativamente frágiles. Finalmente, una nueva posibilidad de un

sistema como el anteriormente descrito permitiría conmutar entre diferentes sistemas operativos sin tener que cargar el software de nuevo.

El arranque por red a menudo coexiste con el arranque típico (por disco). Por ejemplo, un sistema podría estar funcionando bajo Windows en disco, y tener también la posibilidad de arrancar GNU/Linux por red. Hay varias aplicaciones interesantes de esta técnica; se puede utilizar para reinstalar Windows por red.

Para explicarnos mejor: cuando una instalación de Windows se corrompe, algo que "puede" suceder aunque nos parezca imposible, el administrador del sistema tiene la posibilidad de volver a cargar una instalación de Windows fresquita usando el arranque remoto de GNU/Linux por red, dejando la tediosa tarea de formatear el disco y copiar la susodicha instalación en disco, a un 'script' automático.

Como es sabido, cada computadora conectada a una red TCP/IP tiene los siguientes datos:

- Su dirección IP.
- Su máscara de red.
- La dirección IP de un router.
- La dirección IP de un servidor de nombres.

Todo esto se guarda en los ficheros de configuración de nuestro sistema operativo.

2.1.1 ¿Cómo Funciona el Diskless Booting?

Para poder arrancar por red, el computador debe obtener:

- Una identidad.
- La imagen de un sistema operativo.
- Normalmente, un sistema de ficheros con el que trabajar.

El **DISKLESS BOOTING** requiere:

- Un servidor de DHCP para determinar de una MAC suministrada, su ubicación en la red y darle una dirección IP fija. Además se requiere una placa base que permita BOOTEEO PXE mediante la red.
- Un servidor TFTP para poder transmitir el ROOM de arranque PXE y posteriormente pasar por el mismo servidor el kernel del sistema (el núcleo del mismo) el cual se configurara según los argumentos especificados en la configuración del PXE, para que después de esto el kernel sepa dónde buscar los ficheros de sistema en el servidor. Los cuales se transmitirán por NFS.

Supongamos un computador sin disco (DC - Diskless Computer) que tiene una ROM para el arranque por red. Podría ser uno entre varios DCs idénticos. ¿Cómo podemos distinguir este computador del resto? Existe una información que es única para ese ordenador (en realidad para su adaptador de red) y que se trata de su dirección física.

Cualquier adaptador de Ethernet en el mundo posee una dirección física de 48 bits exclusiva para él, porque a todo fabricante de hardware Ethernet, le han sido asignados bloques de direcciones. Por convención estos bloques de direcciones se

representan en forma de dígitos hexadecimales separados por dos puntos en grupos de dos; por ejemplo: 00:60:08:C7:A3:D8.

Los protocolos usados para obtener una dirección IP, dada una dirección física, son:

- **BOOTP (Bootstrap Protocol)** es un protocolo cliente/servidor diseñado para proporcionar los cuatro tipos de información mencionados anteriormente, a un computador sin disco.
- **DHCP (Dynamic Host Configuration Protocol)** es una extensión de BOOTP, es decir, lo mejora.

En nuestro caso, si no se especifica lo contrario, cualquier cosa que se diga del BOOTP, también es aplicable para el DHCP. (En realidad es una mentira decir que BOOTP y DHCP sólo se encargan de traducir direcciones físicas. En previsión, los diseñadores de BOOTP y DHCP les dejaron la capacidad de poder trabajar con cualquier tipo de direcciones hardware. Sin embargo, Ethernet es lo que la mayoría de la gente usa).

Un ejemplo de intercambio de BOOTP sería el siguiente:

- **DC:** Hola, mi dirección hardware es 00:60:08:C7:A3:D8, por favor, dame mi dirección IP.
- **Servidor BOOTP:** (busca la dirección en su base de datos) Tu nombre es *ws003*, tu dirección IP es 192.168.0.3, tu servidor es *192.168.0.2*, el fichero del que se supone que debes arrancar es "PATH donde se encuentra el archivo que contiene el kernel" (y alguna información adicional).

Nos podemos preguntar cómo fue capaz el DC de encontrar la dirección del servidor de BOOTP en la primera fase. La respuesta es que no lo hizo. La petición de BOOTP se realizó en forma de multidifusión (broadcast) dentro de la red local, de forma que cualquier servidor de BOOTP que pudiera responder a la petición, lo haría.

Después de obtener la dirección IP, el DC debe conseguir la imagen del sistema operativo y lanzarlo a ejecución. En esta fase, se usa otro protocolo de Internet, conocido como Trivial File Transfer Protocol (TFTP). Éste es una versión reducida del famoso FTP: no contempla autenticación, trabaja a través del User Datagram Protocol (UDP) en vez de Transmisión Control Protocol (TCP). Se prefirió usar UDP a TCP por simplicidad. La implementación de UDP en el DC puede ser suficientemente reducida como para caber en una ROM. Ya que UDP es un protocolo orientado a la transmisión por bloques, en oposición a la transmisión por cadenas, la transferencia se realiza bloque a bloque, de la siguiente forma:

- **DC:** Dame el bloque 1 de "PATH donde se encuentra el archivo que contiene el kernel".
- **TFTP servidor:** Aquí lo tienes.
- **DC:** Dame el bloque 2.

Y así en adelante, hasta que se transfiere el fichero completo. El funcionamiento consiste básicamente en el reconocimiento de cada bloque, y la pérdida de paquetes se soluciona mediante su retransmisión al cabo de un tiempo establecido. Cuando todos los bloques han sido recibidos, la ROM de arranque de la red pasa el control a la imagen del SO.

Finalmente, para poner en funcionamiento un SO, se le debe proporcionar un sistema de ficheros raíz. El protocolo utilizado por GNU/Linux y otros sistemas Unix es normalmente el NFS (Network File System), aunque no es el único. En este caso el código no necesita estar grabado en la ROM, sino que forma parte del SO que acabamos de cargar. Sin embargo, el SO debe ser capaz de ejecutarse con un sistema de ficheros raíz NFS, en vez de un disco real.

2.2 ¿QUÉ SON LOS SISTEMAS OPERATIVOS DE RED?

Un SOR (Sistema Operativo de Red) es un componente software de una computadora que tiene como objetivo coordinar y manejar las actividades de los recursos del ordenador en una red de equipos. Consiste en un software que posibilita la comunicación de un sistema informático con otros equipos en el ámbito de una red.

Dependiendo del fabricante del SOR, tenemos que el software de red para un equipo personal se puede añadir al propio SO del equipo o integrarse con él. Netware de Novell es el ejemplo más familiar y famoso de SOR donde el software de red del equipo cliente se incorpora en el SO del equipo. El equipo personal necesita ambos sistemas operativos para gestionar conjuntamente las funciones de red y las funciones individuales.

El software del SOR se integra en un número importante de sistemas operativos conocidos, tales como:

- Windows 2003 Server
- Windows 2000 Server/Professional
- Apple Talk
- GNU/Linux
- BSD family
- NetBIOS

Por motivos de estudio de esta tesis, usaremos dos SOR para estudiarlos y evaluarlos según las prestaciones que ofrecen cada uno de ellos. De ahora en adelante se hablarán de plataformas en lugar de referirnos a sistemas operativos. Las plataformas a estudiarse serán: Windows Server 2003 y GNU/Linux.

2.3 WINDOWS SERVER 2003

Windows Server 2003 es un SO de la familia Windows de la marca Microsoft para servidores que salió al mercado en el año 2003. Está basada en tecnología NT y su versión del núcleo NT es la misma que la de Windows XP usado en Workstations (Estaciones de Trabajo).

En términos generales, Windows Server 2003 se podría considerar como un Windows XP modificado, no con menos funciones, sino que estas están deshabilitadas por defecto para obtener un mejor rendimiento y para centrar el uso del procesador en las características de servidor.

Las nuevas características de Windows Server 2003 hacen que sea, hasta el momento, el SO más estable, robusto, y escalable de Microsoft. En esta sección se hace una introducción a las nuevas características y funcionalidades de la familia de Servidores Windows Server 2003.

2.3.1 Características

Entre las nuevas características que tiene Windows Server 2003, podemos detallar las siguientes:



Figura II.15 Nuevas características de Windows Server 2003

Fuente: <http://www.scribd.com/doc/8189371/CAPITULO-1-Introduccion-a-Windows-Server-2003>

2.3.1.1 Sistema de Recuperación Automática

Esta nueva herramienta permite recuperar el SO a su estado anterior mediante un diskette con información de la configuración y un backup (respaldo) previamente preparado mas el CDROM de instalación de Windows Server 2003.

2.3.1.2 Volume Shadow Copy

Este nuevo servicio ayuda a recuperar archivos perdidos erróneamente. Para ello el servicio Shadow Copy guarda versiones anteriores de archivos para su posterior recuperación, eliminando la necesidad de recurrir al Restore de backup. Utiliza un cache en disco para el almacenamiento de versiones de archivos, que luego se pueden recuperar cuando sea necesario desde esa copia. Ver la Figura II.16.

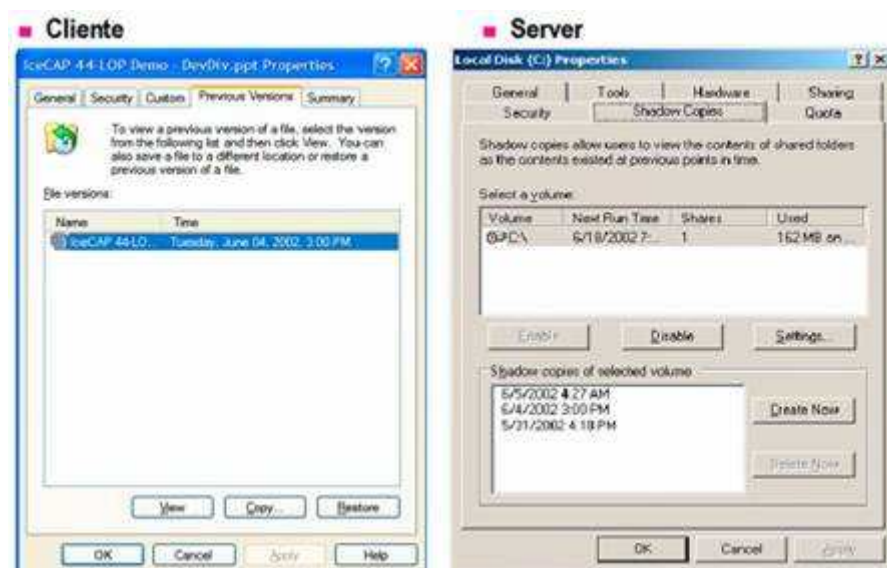


Figura II.16 Shadow copias

Fuente: <http://www.scribd.com/doc/8189371/CAPITULO-1-Introduccion-a-Windows-Server-2003>

2.3.1.3 Sistema de archivos encriptados (Encrypted File System - EFS)

Esta funcionalidad permite realizar una encriptación del sistema de archivos en forma segura y también que otros usuarios tengan acceso a esos archivos. Esta funcionalidad es muy importante puesto que si bien en ocasiones es necesario darle seguridad a ciertos archivos, también es importante poder compartirlos entre usuarios. El sistema de encriptación que utiliza EFS es una combinación de dos métodos, encriptación Asimétrica e infraestructura de clave pública (PKI).

2.3.1.4 Driver Rollback

Esta es una nueva utilidad para el manejo de versiones en drivers de dispositivos y permite volver a la versión anterior del driver si este ocasiona problemas.

2.3.1.5 Directorio Activo (Active Directory)

Es un centro de administración de una red, el mismo que está formado por varios objetos como dominios, sitios, unidades organizativas, computadoras, usuarios, grupos, carpetas y más.

2.3.1.6 Políticas de Seguridad

Las distintas configuraciones de seguridad, empleo de software, configuraciones del SO y comportamiento del software se puede agrupar en lo que se denomina las políticas de seguridad que dan lugar a los objetos de políticas de seguridad GPO (Group Policy Object – Objeto de Política de Grupo). Estos objetos facilitan la vinculación a los dominios, sitios o unidades organizativas.

2.3.1.7 Seguimiento de Errores "Event Tracker"

El Event Tracker es una nueva herramienta que permite recolectar para futuros análisis, los motivos por los cuales un Server se reinicia, se apaga, o fue apagado por falta de energía. En este caso la herramienta le preguntará en el primer Logon (Pantalla de Cierre de Sesión), el motivo del desperfecto para almacenarlo. Ver la Figura II.17.



Figura II.17 Seguimiento de Errores

Fuente: <http://www.scribd.com/doc/8189371/CAPITULO-1-Introduccion-a-Windows-Server-2003>

2.3.1.8 Versiones

En la Tabla II-1 se detalla cada versión de Windows Server 2003 y sus características más importantes.

Tabla II-1 Versiones de Windows Server 2003





Versión	Características
	<p>Esta versión provee una plataforma para el desarrollo y la instalación rápida de servicios y aplicaciones web, para servicios web y hosting.</p> <p>No puede ser un controlador de dominio pero puede ser miembro.</p> <p>No soporta Internet Connection Firewall (ICF) and Internet Connection Sharing (ICS).</p> <p>No soporta (DHCP), servidor de fax, Microsoft SQL Server, ni Terminal server.</p> <p>No soporta aplicaciones no Web.</p> <p>No soporta balance de carga (NLB), ni ASP.NET.</p> <p>Soporta escritorio remoto para administración.</p> <p>Solo se vende con equipos nuevos OEM.</p>
	<p>Para servicios de administración de Redes, ideal para servidor de archivos e impresoras y redes de grupo de trabajo de pequeña empresa:</p> <p>Soporta Directorio Activo, GPO, IIS 6, DHCP, DNS, WINS, Ruteo TCP/IP mediante: Routing and Remote Access service (RRAS), Network Address Translation (NAT), Internet Authentication Service (IAS), Routing Information Protocol (RIP), y Open Shortest Path First (OSPF).</p> <p>Terminal Server.</p>

Tabla II-1 Versiones de Windows Server 2003 (Continuación)

Versión	Características
	<p>Contiene todas las características de Windows Server 2003 Standard y provee escalabilidad y disponibilidad incrementada. Esta versión es ideal para servers utilizados en grandes redes y para bases de datos de uso intensivo.</p> <p>Soporta servicios de directorios unificados o metadirectorio denominado Microsoft Metadirectory Services (MMS) y Microsoft Consulting Service (MCS).</p> <p>Soporta clustering hasta de 8 nodos mediante el uso de todos los procesadores y un almacenamiento común.</p> <p>Soporta adición de memoria sin necesidad de apagar el equipo.</p> <p>Permite una mejor administración de recursos mediante el Windows System Resource Manager (WSRM).</p>
	<p>Contiene todas las características de Windows Server 2003 Enterprise Edition y, además, soporte para más memoria y más procesadores por computadora. Esta versión es ideal para uso de datawarehouses de gran tamaño, procesamiento online, y proyectos de consolidación de servidores.</p>

Fuente: <http://www.scribd.com/doc/8189371/CAPITULO-1-Introduccion-a-Windows-Server-2003>

2.3.1.9 Requerimientos

A continuación se detallarán los requerimientos mínimos que debe tener un Servidor para cada versión de Windows Server 2003. Ver Tabla II-2.

Tabla II-2 Requerimientos de Windows Server 2003

Característica/ versión	Standard	Web Server	Entrepise Server	Datacenter
Vel. de Procesador	133/550 MHz	133/550 MHz	133/733 MHz	400733 MHz
RAM mínima	128 MB	128 MB	128 MB	512 MB
RAM máxima	2 GB	4 GB	32 GB (x86) 64 GB (64-Bit)	64 GB (x86) 512 GB (64-Bit)
Nº max. procesadores	2	4	8	32
Espacio en disco	1.5 GB	1.5 GB	1.5 GB	1.5 GB

Fuente: <http://www.scribd.com/doc/8189371/CAPITULO-1-Introduccion-a-Windows-Server-2003>

2.3.1.10 Funcionalidades

Los servidores desempeñan muchos papeles en el ambiente cliente/servidor de una red, ver Figura II.18. Algunos servidores se configuran para proporcionar la autenticación y otros se configuran para funcionar con otros usos. Asimismo, muchos proporcionan los servicios de red que permiten a usuarios comunicar o encontrar otros servidores y recursos en la red. Los servicios que se pueden configurar en un servidor son:



Figura II.18 Funcionalidades de Windows Server 2003

Fuente: <http://www.scribd.com/doc/8189371/CAPITULO-1-Introduccion-a-Windows-Server-2003>

2.3.1.11 Controlador de Dominio (Directorio Activo)

Los Controladores de dominio almacenan datos del directorio y manejan la comunicación entre los usuarios y los dominios, incluyendo procesos de conexión del usuario, autenticación y búsquedas del directorio. En una red con Windows Server 2003, todos los servidores en el dominio que no sean controladores de dominio se llaman servidores miembros. Los servidores no asociados a un dominio se llaman servidores de grupo de trabajo.

2.3.1.12 Servidor de Archivos

Un servidor de archivos proporciona una localización central en la red donde se puede almacenar y compartir archivos.

2.3.1.13 Servidor de Impresión

Un servidor de impresión proporciona una localización central, donde los usuarios pueden imprimir. Un servidor de impresión provee a los clientes los drivers actualizados de las impresoras conectadas y maneja la cola de impresión.

2.3.1.14 Servidor DNS y DHCP

El Domain Name System (DNS) es un servicio estándar de Internet y de TCP/IP. El servicio de DNS permite a las computadoras cliente resolver nombres de dominio DNS. DHCP es un servidor que se encarga de proporcionar direcciones IP dinámicamente a los clientes.

2.3.1.15 Servicio de terminal server

Con un servidor de terminal Server, se instala una aplicación en un solo servidor los usuarios tendrán acceso a la aplicación sin la instalación de la misma en sus computadoras.

2.3.1.16 La herramienta Manage Your Server

Cuando Windows Server 2003 es instalado y un administrador realiza el login por primera vez, la herramienta Manage Your Server corre automáticamente. Esta herramienta se usa para agregar o para quitar Roles a Servers. Cuando agregue un Rol de Server a una computadora, la herramienta Manage Your Server agregará ese rol de la lista de roles disponibles. Después que el Server Role se agregue a la lista, se podrán utilizar varios wizards (asistentes) que le ayudarán a administrar roles específicos del Server.

2.4 GNU/LINUX

GNU/Linux es, a simple vista, un SO. Es una implementación de libre distribución UNIX para computadoras personales (PC), servidores, y estaciones de trabajo. Fue desarrollado para procesadores i386 y ahora soporta los procesadores de última generación de 64 bits tanto de AMD como de Intel.

2.4.1 Introducción

Este SO está diseñado para poder dar servicio a muchos usuarios al mismo tiempo, así como para ofrecer una interfaz entre los usuarios y el ordenador con su medio de almacenamiento, como discos duros y cintas. Los usuarios tienen sus propias interfaces a través de las cuales interactúan con el SO, como los intérpretes de comandos batch o administradores de ventanas GNOME o KDE para realizar tareas como añadir nuevos usuarios, dispositivos como tarjetas de red o escáneres, o incluso sistemas de archivos. Tales operaciones entran dentro de lo que se conoce como administración del sistema. La persona que lleva a cabo estas acciones se conoce como administrador del sistema o súper usuario. En este sentido, en GNU/Linux hay dos tipos de interacción: la interacción normal de los usuarios y el súper usuario, que es quién se encarga de realizar todas las tareas de administración del sistema.

La mayoría de estas tareas sólo es necesario realizarlas ocasionalmente, como son los casos de la adición de una nueva impresora y el montaje de un sistema de archivos. Pero hay otras tareas, en cambio, como agregar o quitar usuarios, que se han de llevar a cabo con mayor frecuencia. La administración básica del sistema abarca temas tales como el acceso al sistema por los súper usuarios. La selección de un nivel de ejecución al inicio, los archivos de configuración del sistema y el control del rendimiento.

Gracias a su filosofía de código libre, ha tenido mucha aceptación. Encontrándose muchas versiones comerciales como Red Hat y sus derivados, OpenSuSe, Mandriva, Debian, etc., incluso auspiciada por gobiernos como EEUU, Alemania, Suiza, etc.

2.4.2 Características de GNU/Linux

GNU/Linux es un SO multiusuario y multitarea de propósito general. Tiene las mismas características que UNIX. Pero también aporta elementos propios.

GNU/Linux soporta el modo protegido de los procesadores de Intel 80x86. Pero también las prestaciones más avanzadas de todos los procesadores que soporta su núcleo: PowerPC, SPARC, MIPS, sistemas IBM 390, PDA's, etc.

Además GNU/Linux soporta la paginación de memoria a disco duro. Esto supone ampliar los límites de la memoria RAM del sistema, utilizando un espacio reservado en el disco duro para alojar páginas de memoria no usadas en un cierto tiempo o que ya no caben en la memoria principal. Esto es lo que se conoce como área de intercambio o partición swap. Otras características importantes del sistema son:

- Todo el SO es de código abierto (Open Source).
- Soporta numerosos shell: csh (C Shell), sh (Bourne Shell), pero, quizás, el más extendido entre los usuarios e implementaciones de GNU/Linux es el bash (Bourne Again Shell). Además los intérpretes de GNU/Linux tienen capacidades que permiten escribir secuencias de programación (shell scripts) que simplifican la automatización de muchas tareas sin necesidad de escribir un programa específico.
- Proporciona una interfaz POSIX.
- Se puede trabajar con varias consolas de texto y graficas.

- Proporciona soporte para multiprocesadores utilizando un esquema de multiprocesos simétricos. Para aprovechar al máximo el paralelismo del hardware, se ha ido modificando progresivamente el núcleo con el objetivo de aumentar su concurrencia interna.
- En cuanto a la estructura de GNU/Linux, como se comentó previamente, tiene una organización monolítica al igual que ocurre con la mayoría de las implementaciones de UNIX. A pesar de este carácter monolítico, el núcleo no es algo estático y cerrado sino que se puede añadir y quitar módulos de código en tiempo de ejecución. Se trata de un mecanismo similar al de las bibliotecas dinámicas pero aplicando al propio sistema operativo. Se puede añadir módulos que correspondan con nuevos tipos de sistemas de archivos, nuevos manejadores de dispositivos o gestores de nuevos formatos de ejecutables.
- Un sistema GNU/Linux completo no solo está formado por el núcleo monolítico sino también incluyen programas del sistema (por ejemplo: demonios) y bibliotecas del sistema.
- Proporciona soporte para una gran variedad de tipos de sistemas de archivos (filesystems), entre ellos los utilizados en Windows, pero también tiene su sistema propio, llamado *extn*, que aporta numerosas prestaciones en cuanto a seguridad y capacidad de recuperación de los datos.
- Acerca de la gestión de memoria GNU/Linux, con la reciente aparición de la versión 2.6.xx del kernel puede llegar a gestionar hasta 64 GB de memoria RAM, múltiples procesadores y sistema de archivos (filessystems) de 64 bits, así como DVD y USB.
- También es capaz de manejar distintos formatos de archivos ejecutables.
- En cuanto a servicios de red, GNU/Linux incluye un completo soporte TCP/IP con soporte para la mayor parte de tarjetas de red actuales usando servicios como DHCP, DNS, etc. Así mismo incluye soporte SLIP/PPP para el acceso a

redes IP mediante puertos serie, tanto RS232, como ISDN, etc. También incluye soporte PLIP, para comunicación a través de puertos paralelos.

- Están, también, completamente soportados servicios NFS, FTP, TELNET, NNTP, SMTP, POP, IMAP, NIS y LDAP, así como servidores HTTP y HTTP seguro.
- Utiliza los editores de texto tradicionales como vi y numerosos editores que aportan características WYSIWYG. Son programas, unos bajo licencia GPL y otros comerciales. Entre los que están distribuidos bajo licencia de uso libre están la suite KOffice, OpenOffice (que pertenece a Sun Microsystems), AbiWord, etc.
- La gestión de procesos en GNU/Linux es básicamente igual que en cualquier otra variedad de UNIX. Utiliza para la creación el clásico fork en el que comparten con el padre su mapa de memoria, sus rutinas de manejo de señales y sus descriptores de archivos. Aunque Linux no implementa threads en el núcleo, se pueden construir bibliotecas de threads usando este nuevo servicio.
- GNU/Linux soporta tres clases de planificación: un algoritmo de tiempo compartido y dos algoritmos de planificación de tiempo real que se corresponde con los definidos por POSIX. El servicio sched _ setcheduler permite definir la clase de planificación del proceso que la invoca. Cada proceso de tiempo real tiene asociada una prioridad y un tipo de planificación que puede ser FIFO o Round_Robin. El planificador selecciona en cada momento el proceso listo para ejecutar que tenga mayor prioridad. Si el proceso es tipo FIFO, seguirá ejecutando hasta que se bloquee. Si es de tipo Round- Robin, cuando termine su rodaja, el proceso pasará al final de la cola de procesos listos para ejecutar de su misma prioridad.
- Utiliza un modelo de memoria que utiliza un esquema de paginación con tres niveles.

- Utiliza una cache común para todos los dispositivos de bloques. El tamaño de la cache es dinámico y crece de acuerdo a las necesidades de memoria del resto del sistema.
- En cuanto al acceso de los discos, se utiliza el algoritmo del ascensor con un único sentido de servicio.
- Siguiendo el modelo de UNIX, en GNU/Linux los usuarios ven los dispositivos como archivo y utilizan los servicios destinados a trabajar con archivos para acceder a los dispositivos.
- Está escrito en un lenguaje de alto nivel C.
- El sistema se basa en un Núcleo llamado Linux, que reside permanentemente en la memoria, y que atiende a todas las llamadas del sistema, administra el acceso a los archivos y el inicio o la suspensión de las tareas de los usuarios.

2.4.3 Estructura de los archivos en GNU/Linux

GNU/Linux hace uso de una serie de directorios con nombres específicos para distintas tareas de administración especializadas. Todos esos directorios se sitúan en el nivel superior del sistema de archivos principal de Linux, el directorio raíz del sistema de archivos, representado por una barra diagonal "/". Por ejemplo: el directorio */dev* contiene los archivos de dispositivos, y el directorio */home* contiene todos los directorios principales de los usuarios y todos los archivos de éstos. Para administrar el sistema, debemos familiarizarnos con la estructura de directorios del SO.

Tabla II-3 Directorios del Sistema

Directorios	Descripción
/bin	Programas relacionados con el sistema.
/sbin	Programa del sistema para tareas especializadas.
/lib	Bibliotecas del sistema.
/etc	Archivos de configuración para aplicaciones y servicios de red y del sistema.
/home	Ubicación de los directorios principales de usuarios y directorios de datos del servidor, como archivos de sitios de Web y FTP.
/mnt	Ubicación donde se montan los archivos de disquete y CD-ROM.
/var	Ubicación de sistemas de directorios del sistema cuyos archivos están en constante cambio, como registros, archivos de cola de impresión y archivos de bloqueo.
/usr	Programas y archivos relacionados con usuarios. Incluye varios subdirectorios clave, como /usr/bin, /usr/x11 y /usr/doc.
/usr/bin	Programas para usuarios.
/dev	Archivos de dispositivos
/usr/x11	Archivo de configuración del Sistema X Window.
/usr/share	Archivos compartidos.
/usr/share/doc	Documentación de aplicaciones.
/tmp	Directorio para archivos temporales del sistema.

Fuente: <http://www.manualespdf.es/manual-linux/>

2.4.3.1 Directorios de Programas

Los directorios que incluyen el término "bin" en su nombre se utilizan para almacenar programas.

El directorio "/bin" contiene los programas de usuario básicos, como el inicio de sesión, shell (bash, tcsh y zsh) y comandos (cp, mv, rm, ln, etc.).

El directorio "/sbin" contiene programas del sistema especializados para tareas como la administración del sistema de archivos (fsck, fdisk, nkfs) y operaciones del sistema como el apagado y el inicio (init).

El directorio "/usr/bin" contiene archivos de programa relacionados con tareas de usuario.

El directorio "/usr/sbin" contiene operaciones del sistema relacionadas con los usuarios, como useradd, para agregar nuevos usuarios.

El directorio "/lib" contiene todas las bibliotecas utilizadas por el sistema, incluida la biblioteca principal de GNU/Linux, libc, y subdirectorios tales como modules, donde se guardan todos los módulos del kernel.

2.4.3.2 Directorios y Archivos de Configuración

Al configurar diferentes elementos del sistema, como usuarios, aplicaciones, servidores o conexiones de red, se utilizan archivos de configuración que se guardan en determinados directorios del sistema.

Los archivos de configuración se guardan en el directorio `"/etc"`, mientras que en el directorio `"/etc/sysconfig"` contiene configuraciones de servicios y dispositivos más específicos.

`/etc`

El directorio `"/etc"` contiene los archivos de configuración del sistema, redes, servidores y aplicaciones. Ahí se encontrará el archivo `fstab` que contiene un listado de los archivos del sistema; el archivo `hosts` con las direcciones IP de los host del sistema, y `grub.conf` para los sistemas de arranque soportados por el cargador de arranque GRUB. Este directorio incluye varios subdirectorios, como `"/httpd"` para los archivos de configuración del servidor web Apache, y `X11` para los archivos de configuración del gestor de ventanas y el Sistema X Window. Puede configurar muchas aplicaciones y servicios editando directamente sus correspondientes archivos de configuración, aunque lo mejor es siempre utilizar la herramienta de configuración apropiada para cada caso, como las que ofrece CentOS 5.2. En la Tabla II-4 se puede ver algunos archivos de configuración utilizados habitualmente, que están ubicados en el directorio `"/etc"`.

Tabla II-4 Archivos de Configuración

Archivo	Descripción
/etc/inittab	Configura el estado predeterminado, así como las conexiones de terminal.
/etc/passwd	Contiene las configuraciones de contraseña e inicio de sesión de los usuarios.
/etc/shadow	Contiene contraseñas de usuario cifradas.
/etc/group	Contiene una lista de grupos con la configuración de cada uno de ellos.
/etc/fstab	Monta automáticamente sistemas de archivos al iniciar el sistema.
/etc/grub.conf	Archivos de configuración GRUB para el cargador de arranque GRUB.
/etc/modules.conf	Módulos del sistema que deben automáticamente.
/etc/termcap	Contiene una lista de especificaciones de tipo terminal para los terminales que pueden conectarse al sistema.
/etc/services	Servicio que se ejecuta en el sistema y puerto que utiliza cada uno de ellos.
/etc/profile	Archivo de configuración de shell predeterminado para los usuarios.
/etc/shells	Contiene los shell instalados en el sistema que pueden utilizar los usuarios.
/etc/mod	Mensaje del día del administrador del sistema.
/etc/inittab	Configura el estado predeterminado, así como las conexiones de terminal.

Fuente: <http://www.manualespdf.es/manual-linux/>

/usr

El directorio “/usr” contiene un gran número de subdirectorios importantes que se utilizan para dar soporte a impresoras, facilitar aplicaciones, bibliotecas y documentación. En la Tabla II-5 se puede ver todos los subdirectorios del directorio “/usr”.

Tabla II-5 Subdirectorios de usr

Archivo	Descripción
usr/bin	Contiene la mayoría de los comandos y programas de utilidad.
/usr/sbin	Contiene aplicaciones administrativas.
/usr/lib	Contiene bibliotecas para aplicaciones, lenguajes de programación, escritorios, etc.
usr/games	Contiene juegos y programas educativos.
/usr/include	Contiene los archivos de cabecera del lenguaje de programación C (.h).
/usr/doc	Contiene documentación de GNU/Linux.
/usr/local	Incluye software instalado localmente.
/usr/share	Contiene datos independientes de la arquitectura, como la documentación.
/usr/src	Contiene código fuente, incluido el código fuente de kernel.
/usr/X11R6	Contiene aplicaciones y bibliotecas basadas en el Sistema X Window.

Fuente: <http://www.manualespdf.es/manual-linux/>

/mnt

El directorio “/mnt” se utiliza normalmente como punto de montaje para unidades CD—ROM, DVD, disquetes, discos Zip y otros sistemas de archivos montados como particiones de Sistemas Windows.

Se trata de sistemas de archivos que se deben cambiar con frecuencia, a diferencia de las particiones que se realizan en los discos duros. CentOS 5.2 instala por omisión un subdirectorios para montar unidades de disquete y CD-ROM, `"/media/floppy"` y `"/media/cdrom"`, respectivamente. Si hay unidades adicionales, se agrega un número a su nombre, como `"/media/cdrom1"`, para una segunda unidad de CD-ROM. También puede crear directorios para las particiones que desee montar, como `"/media/Windows"`, en el caso de una partición Windows.

`/home`

El directorio `/home` contiene los directorios principales de usuario. Cuando se configura una cuenta de usuario, se crea aquí el directorio principal para dicha cuenta que suele tener el mismo nombre que el usuario. Como administrador del sistema se puede acceder al directorio principal de cualquier usuario, lo que nos otorga un control total sobre sus archivos.

`/var`

El directorio `"/var"` contiene subdirectorios para tareas cuyos archivos deben cambiar con frecuencia, como archivos de bloqueo, archivos de registro, archivos de servidor web o archivos de cola de impresión. Por ejemplo: el directorio `"/var"` contiene directorios de datos de servidor como `"/var/www"` para los archivos del sitio web del servidor web Apache, `"/var/ftp"` para los archivos del servidor FTP y `"/var/named"` para el servidor DNS.

Los directorios `"/var"` están diseñados para almacenar datos que cambian con el funcionamiento normal del sistema GNU/Linux. En la Tabla II-6 se puede ver los subdirectorios del directorio `"/var"`.

Tabla II-6 Subdirectorios de /var

Archivo	Descripción
/var/acount	Procesa registros de cuentas.
/var/cache	Contiene datos de caché para páginas man, datos del Proxy web, fuente o datos específicos de aplicaciones.
/var/crash	Contiene volcados de los fallos del sistema.
/var/games	Contiene datos variables de juegos.
/var/lib	Contiene información de estado para aplicaciones correctas.
/var/local	Se utiliza para datos cambiantes de programas instalados en /usr/local.
/var/lock	Contiene archivos de bloqueo, que indican cuándo están en uso un programa o archivo determinado.
/var/log	Contiene archivos de registros de todo lo que suceda con el sistema.
/var/mail	Contiene archivos de la bandeja de entrada de usuarios.
/var/opt	Contiene datos variables de aplicaciones instaladas en /opt.
/var/run	Contiene información sobre los procesos del sistema que se encuentran en ejecución.
/var/spool	Contiene datos de colas, como los correspondientes al correo, las noticias o las colas de impresión, así como los trabajos cron y at.
/var/tmp	Contiene archivos temporales cuyo contenido debe mantenerse entre reinicios del sistema.
/var/yp	Contiene archivos de datos de Network Information Service (NIS).
/var/www	Contiene archivos del sitio web del servidor web.
/var/ftp	Contiene archivos FTP del servidor FTP.
/var/named	Contiene archivos de configuración de dominios del servidor DNS.

Fuente: <http://www.manualespdf.es/manual-linux/>

/proc

El sistema de archivos "/proc" es un sistema de archivos especial que se genera en la memoria del sistema. No tiene una existencia real en el disco. "/proc" contiene archivos que proporcionan información importante acerca del estado del sistema.

Por ejemplo: "/proc/cpuinfo" contiene información sobre el procesador del equipo. "/proc/devices" contiene los dispositivos que están configurados actualmente para ejecutarse con el kernel. "/proc/filesystems" contiene un listado de los sistemas de archivos. Los archivos de "/proc" son, en realidad, interfaces para el kernel a través de las cuales se obtiene información del kernel relacionada con el sistema.

Tabla II-7 Subdirectorios y archivos de /proc

Archivo	Descripción
/proc/núm	Hay un directorio por cada proceso, etiquetado con el número de éste "/proc/1" será el directorio correspondiente al proceso 1.
/proc/cpuinfo	Contiene información sobre la CPU, como su tipo, versión, modelo y rendimiento.
/proc/devices	Contiene un listado de los controladores de dispositivos configurados para el kernel que se encuentra actualmente en ejecución.
/proc/dma	Muestra los canales DMA que se encuentra en uso actual.
/proc/filesystems	Contiene un listado de los sistemas de archivos configurados en el kernel.
/proc/interrupts	Muestra las interrupciones del uso.
/proc/ioports	Muestra los puertos de E/S en uso.
/proc/kcore	Contiene una imagen de la memoria física del sistema.
/proc/kmsg	Contiene mensajes generados por el kernel.

Tabla II-7 Subdirectorios y archivos de /proc (Continuación)

Archivo	Descripción
/proc/ksyms	Contiene la tabla de símbolos por el kernel.
/proc/loadavg	Muestra el promedio de carga del sistema.
/proc/meminfo	Muestra el uso de la memoria.
/proc/modules	Contiene un listado de los módulos de kernel que están cargados actualmente.
/proc/net	Contiene un listado con información de estado sobre los protocolos de red.
/proc/stat	Contiene estadísticas de funcionamiento del sistema, como las apariciones de fallo de página.
/proc/uptime	Muestra el tiempo que ha estado archivo el sistema.
/proc/versión	Muestra la versión del kernel.

Fuente: <http://www.manualespdf.es/manual-linux/>

/dev

Para montar un sistema de archivos, ha de especificar su nombre de dispositivo. Las interfaces de dispositivo que pueden estar asociadas al sistema las proporcionan archivos especiales conocidos como archivos de dispositivos. Los nombres de estos archivos de dispositivo se corresponden con los nombres de los dispositivos. Los archivos de dispositivo están ubicados en directorio “/dev” y normalmente consisten en nombres abreviados que terminan con el número del dispositivo. Por ejemplo: fd0 puede hacer referencia a la unidad de disquetes asociada al sistema.

Tabla II-8 Prefijos de Nombres de Dispositivo

Nombre de Dispositivo	Descripción
hd	Discos duros IDE; del 1 al 4 representan las particiones primarias; del 5 en adelante representan participaciones lógicas, a, b, c representan discos duros.
sd	Discos duros SCSI
scd	Unidades de CD-ROM SCSI.
fd	Unidades de disquete.
st	Unidades de cinta SCSI.
nst	Unidades de cinta SCSI sin rebobinado.
ht	Unidades de cinta IDE.
tty	Terminales.
lp	Puertos de impresora.
pty	Pseudoterminales (usado para inicios de sesión remotos).
js	Palancas de juegos analógicas.
midi	Puertos midi.
ttys	Puertos serie.
md	Dispositivos RAID.
cdrom	Vínculo para el archivo del dispositivo CD-ROM.
cdwriter	Vínculo para el archivo del dispositivo CD-R o CD-RW.
modem	Vínculo para el archivo del dispositivo módem.
floppy	Vínculo para el archivo del dispositivo de disquete.
tape	Vínculo para el archivo del dispositivo de cinta.
scanner	Vínculo para el archivo del dispositivo escáner.

Fuente: <http://www.manualespdf.es/manual-linux/>

Los discos duros IDE usan el prefijo `hd`, mientras que los discos duros SCSI usan el prefijo `sd`. Por otro lado, los dispositivos RAID utilizan el prefijo `md`. El prefijo de un disco duro va seguido de una letra que sirve de etiqueta al disco y de un número, correspondiente a la partición. Por ejemplo: `hda2` haría referencia a la segunda partición del primer disco duro IDE, mientras que la primera unidad de disco duro estaría representada por la letra `a`, como en `hda`. El dispositivo `sdb3` se refiere a la tercera partición del segundo disco duro SCSI (`sdb`). Sin embargo, los dispositivos RAID están numerados a partir del 0, al igual que sucede con los disquetes. El dispositivo `md0` se refiere al primer dispositivo RAID, y `md1` haría referencia al segundo. GNU/Linux soporta hasta cuatro particiones primarias para discos duros IDE numeradas del 1 al 4. No existe límite para el número de particiones lógicas. Para conocer el nombre de un dispositivo, puede usar `df` para mostrar las particiones del disco duro o bien examinar el archivo `"/etc/fstab"`.

Registro del sistema /var/log y syslogd

En el directorio `"/var/log"` se guardan diferentes registros del sistema correspondientes a otras tantas tareas realizadas en el mismo. Ahí puede hallar registros para correo, noticias y todas las demás operaciones del sistema como registros de servidor web. El archivo `"/var/log/messages"` contiene un registro de todas las tareas del sistema no cubiertas por los demás registros. Entre éstas figuran normalmente tareas de inicio, como la carga de controladores y el montaje de sistemas de archivos. Si se produce un fallo al cargarse un controlador de una tarjeta al inicio, encontrará ahí el correspondiente mensaje de error. En este archivo se registran también los inicios de sesión, con información de quién ha intentado entrar y en qué cuenta. El archivo `"/var/log/maillog"` registra las transmisiones de mensajes de correo y noticias.

CAPÍTULO III

ESTUDIO COMPARATIVO ENTRE WINDOWS SERVER 2003 Y GNU/LINUX

3.1 INTRODUCCIÓN

La selección de la plataforma a utilizar es una de las decisiones más importantes que el administrador de un laboratorio de software y redes debe tomar antes de empezar con la implementación de un Servidor de Terminales, esta decisión debe estar sustentada en un análisis minucioso basado en parámetros y/o criterios de comparación previamente establecidos, además de estar enmarcada en el cumplimiento de las metas y objetivos institucionales, referentes con la tecnología.

El establecimiento de los criterios para la realización del estudio comparativo entre Windows Server 2003 (WS2003) y GNU/Linux se basa en las características principales que nos ofrece cada una de las plataformas escogidas, por cada criterio se analizan los aspectos más relevantes que permitan la toma de una decisión.

Este capítulo se enfoca en el análisis y comparación de las plataformas WS2003 y GNU/Linux con el fin de ofrecer una guía que permita la selección de una de ellas para su inmediata implementación de nuestro Servidor de Terminales acorde a las necesidades, presupuesto y/o requerimientos de la Escuela de Ingeniería Electrónica de la ESPOCH, los resultados obtenidos con el presente estudio se exponen en forma de conclusiones y recomendaciones.

3.2 CRITERIOS DE COMPARACIÓN

Para la comparación de dos plataformas se pueden considerar una serie de parámetros siendo unos más determinantes que otros, pero para propósito de este estudio se ha considerado como más relevantes a los siguientes criterios de comparación:

- **Facilidad de instalación:** Este es uno de los principales aspectos al momento de elegir una plataforma a utilizar, para la instalación y configuración de nuestro servidor.
- **Disponibilidad de software:** Hablaremos sobre los paquetes de software adicionales que tendremos disponibles para su posterior instalación en caso de necesitarlos.
- **Estabilidad:** La estabilidad de un SO implica que éste no tiene que quedar bloqueado por ningún proceso que esté haciendo uso de algún recurso del sistema.
- **Utilización de recursos y rendimiento:** Esto implica la rapidez con la que ejecuta sus aplicaciones un SO y la manera de gestionar los recursos del mismo.
- **Configurabilidad vs. Facilidad de uso:** Un sistema operativo debe brindar la posibilidad de poder modificar la configuración que trae por defecto a la vez que debe ser fácil su utilización.
- **Código abierto vs. Código propietario:** Programas que pueden usarse en cualquiera de las dos plataformas a compararse y que deben demostrar estabilidad en su ejecución.
- **Sistema de archivos:** El sistema de archivos permite organizar la información dentro de los dispositivos de almacenamiento secundario en un formato inteligible para el SO.

- **Seguridad y Precio:** El costo de licencias por el uso de un SO y la seguridad es una parte fundamental para la elección de la plataforma a utilizar ya que tenemos que tomar en cuenta que el servidor estará conectado en red y puede recibir ataques externos.

3.3 CUADROS COMPARATIVOS

En esta sección se muestra el estudio de las plataformas Windows Server 2003 (WS2003) y GNU/Linux a manera de cuadros comparativos seguidos de una interpretación y calificación del criterio por parte del autor, dichos cuadros comparativos se encuentran clasificados de acuerdo a los criterios de comparación definidos anteriormente.

Para obtener resultados cuantitativos y cualitativos que permitan una selección mas sustentada de una tecnología, la calificación de cada criterio de comparación está basada en la siguiente tabla:

Regular	Bueno	Muy Bueno	Excelente
< 20%	>=20% y < 50%	>= 50% y < 80%	>= 80%

Cada uno de los ítems de la interpretación incluye la siguiente nomenclatura **(x,y)/z**, en donde:

- x:** Representa el puntaje que obtiene la plataforma Windows Server 2003.
- y:** Representa el puntaje que obtiene la plataforma GNU/Linux.
- z:** Representa la base sobre la cual se está calificando cada ítem.

La calificación definitiva de la plataforma en base a cada criterio se obtiene sumando los puntajes de todos los ítems de la interpretación, basándose en las siguientes fórmulas:

$$Pws = \Sigma(x), Pgnu = \Sigma(y), Pc = \Sigma(z)$$

Calificación de Windows Server 2003 (**Cc-WS**) = $(P_{ws} / PC) * 100\%$

Calificación de GNU/Linux (**Cc-GNU**) = $(P_{gnu} / PC) * 100\%$

En donde:

Pws: Puntaje acumulado de la plataforma Windows Server 2003 (WS2003).

Pgnu: Puntaje acumulado de la plataforma GNU/Linux.

Pc: Puntaje sobre el cual se califica el criterio.

Cc-WS: Porcentaje de la calificación total que obtuvo Windows Server 2003 (WS2003) en el criterio.

Cc-GNU: Porcentaje de la calificación total que obtuvo GNU/Linux en el criterio.

3.3.1 Facilidad de Instalación

La facilidad de instalación que ofrece una plataforma la podemos analizar según algunas de sus características y la manera como realiza ciertas tareas, tal como se muestra en la siguiente tabla:

Tabla III-9 Facilidad de instalación entre WS2003 y GNU/Linux

	WS2003	GNU/Linux
Interfaz gráfica	Este sistema cuenta con una interfaz gráfica intuitiva para su instalación.	Las nuevas distribuciones de este sistema ya cuentan con una interfaz gráfica intuitiva.
Personalizar paquetes a instalar	No nos permite personalizar el sistema a la hora de empezar la instalación.	Incluye la opción de escoger: <ul style="list-style-type: none">• Instalación personal.• Instalar todo.
Gestor de arranque	Trae el gestor de arranque Windows Boot Manager por defecto y no puede ser cambiado.	Podemos escoger entre dos gestores de arranque: <ul style="list-style-type: none">• LILO.• GRUB.
Drivers	A menudo los drivers deben ser instalados separadamente ya que no se instalan cuando se instala el SO. Cada fabricante tiene su propio driver.	La mayoría de dispositivos son reconocidos y se instalan automáticamente al momento de instalar el SO. No hace falta instalar drivers adicionales.

Tabla III-1 Facilidad de instalación entre WS2003 y GNU/Linux (Continuación)

	WS2003	GNU/Linux
Tiempo de instalación	El tiempo varía dependiendo de la versión a instalarse, configuración de hardware, si es actualización o es una instalación limpia. Por lo general el tiempo es de unos 20 minutos.	Las gamas son de 6 minutos a una hora, dependiendo de la distribución y de los paquetes requeridos. El tiempo oscila los 30 minutos para distribuciones de uso general.

Fuente: Análisis del autor

Interpretación:

- La interfaz gráfica de Windows Server 2003 es muy buena pero la interfaz gráfica de GNU/Linux ha evolucionado tanto porque en un principio la interfaz era muy rústica. **(1,1)/1**
- La personalización de paquetes en GNU/Linux es mucho mejor porque podemos hacer dos tipos de instalación y en Windows Server 2003 no podemos personalizar los paquetes que se desean instalar. **(0,2)/2**
- Los drivers para Windows Server 2003 son proporcionados en un CD para ser instalados posteriormente, mientras que para GNU/Linux los drivers vienen incluidos en la mayoría de las distribuciones existentes y no hace falta hacer ninguna instalación adicional. **(0.5,1)/1**
- El gestor de arranque es un factor importante a la hora de arrancar más de un SO. GNU/Linux nos da la posibilidad de instalar uno de los 2 gestores de

arranque que trae en el instalador y Windows Server 2003 no tiene esa posibilidad porque solo tiene un gestor de arranque por defecto. **(1,2)/2**

- El "tiempo de la instalación" puede ser medido dependiendo de qué acciones tomemos como: instalar el sistema operativo, drivers de los dispositivos, actualización del SO e instalación de aplicaciones. **(1,0)/1**

Calificación:

Pc = $\Sigma(z) = 1 + 2 + 1 + 2 + 1 = 7$

Pws = $\Sigma(x) = 1 + 0 + 0.5 + 1 + 1 = 3.5$ **Cc-Pws:** $Pws / Pc = (3.5/7)*100\% = 50.00\%$

Pgnu = $\Sigma(y) = 1 + 2 + 1 + 2 + 0.5 = 6$ **Cc-GNU:** $Pgnu / Pc = (6/7)*100\% = 85.72\%$

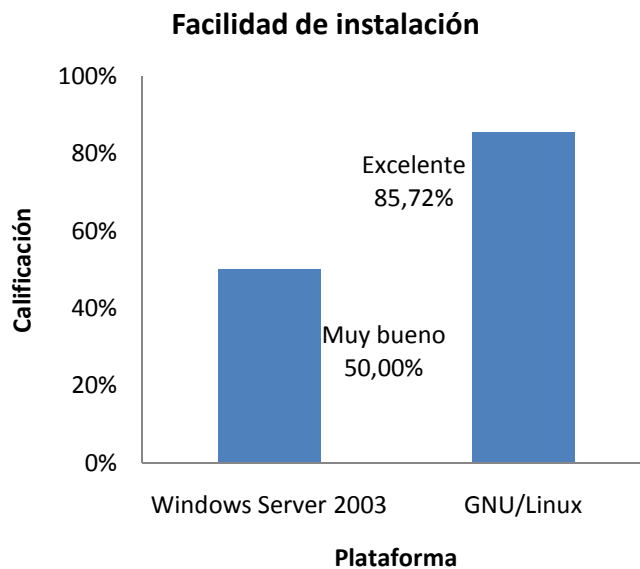


Figura III.19 Facilidad de instalación

Fuente: Análisis del autor

3.3.2 Disponibilidad de Software

Ambas plataformas cuentan con muchísimo software de todos los tipos. Se analizarán los paquetes que más se utilizan y que estén disponibles para cada plataforma. A continuación se detalla:

Tabla III-10 Disponibilidad de Software entre WS2003 y GNU/Linux

	WS2003	GNU/Linux
Internet	<p>Este sistema cuenta con un navegador propio pero un poco inseguro, se pueden instalar otros como:</p> <ul style="list-style-type: none">• Mozilla Firefox.• Opera.• Safari.	<p>Este sistema también cuenta con un navegador propio llamado Mozilla Firefox y que nos brinda mucha seguridad al navegar en la web. También se pueden instalar más navegadores disponibles para esta plataforma.</p>
Oficina	<p>Se puede instalar el paquete Microsoft Office que tiene muchas características para usuarios avanzados. Es una suite muy buena para la oficina.</p>	<p>Cuenta con el paquete OpenOffice que por el momento no tiene el poder de Microsoft Office pero para los usuarios que necesitan lo básico, este paquete es más que suficiente.</p>

Tabla III-2 Disponibilidad de Software entre WS2003 y GNU/Linux (Continuación)

	WS2003	GNU/Linux
Grabación de CDs – DVDs	Existen muchos paquetes para grabación de CDs – DVDs como: <ul style="list-style-type: none"> • Nero. • Roxio Easy CD/DVD Creator. 	Podemos encontrar algunos paquetes como: <ul style="list-style-type: none"> • K3B. • Toaster. • Nero (nuevo).
Desarrollo Java	Podemos encontrar los mismos paquetes que son propios de GNU/Linux pero para la plataforma Windows.	Podemos encontrar algunos paquetes tales como: <ul style="list-style-type: none"> • NetBeans. • Eclipse. • dBVisualizer.

Fuente: Análisis del autor

Interpretación:

- Windows Server 2003 tiene un navegador web propio pero se pueden instalar muchos otros como Mozilla Firefox que es propio de GNU/Linux. En GNU/Linux viene instalado Mozilla Firefox que es muy bueno. **(3,2)/3**
- OpenOffice es una suite de oficina que todavía está madurando debido a los muchos programadores que ayudan en la creación de nuevas características, no se puede saber en qué tiempo podrá igualar o mejorar a Microsoft Office. **(1,0.5)/1**
- Existen más paquetes para la grabación de CDs – DVDs para la plataforma GNU/Linux que para la plataforma Windows. **(2,3)/3**

- Para el Desarrollo Java existen los mismos paquetes para ambas plataformas pero debido a que dichos paquetes son propios de GNU/Linux, no se puede ejecutar en máquinas antiguas bajo Windows Server 2003.

(3,4)/4

Calificación:

Pc = $\Sigma(z)$ = 3 + 2 + 3 + 4 = **11**

Pws = $\Sigma(x)$ = 3 + 1 + 2 + 3 = **9**

Cc-Pws: $Pws / Pc = (9/11)*100\% = \mathbf{81.81\%}$

Pgnu = $\Sigma(y)$ = 2 + 0.5 + 3 + 4 = **9.5**

Cc-GNU: $Pgnu / Pc = (9.5/11)*100\% = \mathbf{86.4\%}$

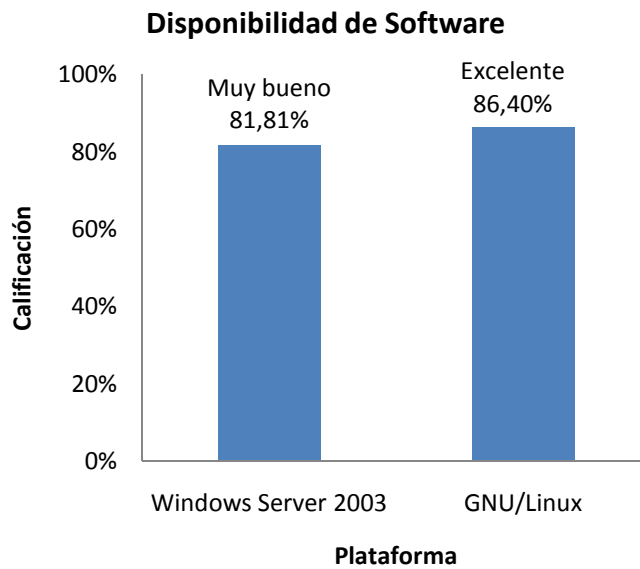


Figura III.20 Disponibilidad de Software

Fuente: Análisis del autor

3.3.3 Estabilidad

Es sin lugar a dudas uno de los argumentos que más a menudo se utiliza para demostrar la superioridad de cualquier SO que sirva para gestionar y administrar una red.

Tabla III-11 Estabilidad de las plataformas WS2003 y GNU/Linux

	WS2003	GNU/Linux
Aplicaciones	En caso de tener una aplicación que no responda, la podemos cerrar pero muchas veces no se cierran las aplicaciones lo cual nos obliga a reiniciar el servidor.	Este sistema tiene la posibilidad de matar una aplicación con el comando "kill" lo cual es una gran ventaja porque se basa en eliminar los procesos que no responden según el número del proceso.
Pantalla azul	Es muy común tener una pantalla azul en WS2003 debido a incompatibilidad de hardware o por mal funcionamiento de algún paquete.	Estas pantallas no son problema en GNU/Linux porque es compatible con mucho hardware y tiene menos probabilidades de que falle algún paquete.
Entornos gráficos	<ul style="list-style-type: none">• GUI integrado al kernel.• Si el entorno gráfico no responde, tendremos un cuelgue seguro.	<ul style="list-style-type: none">• GUI independiente del kernel.• Se puede matar el entorno gráfico si éste no responde.

Tabla III-3 Estabilidad de las plataformas WS2003 y GNU/Linux (Continuación)

	WS2003	GNU/Linux
Reinicios forzados	Reiniciar es requerido obligatoriamente después de que se actualiza, o se instala algún driver, como el de la tarjeta de video.	GNU/Linux sólo necesita reiniciarse para actualizar su kernel.

Fuente: Análisis del autor

Interpretación:

- Debemos contar con la posibilidad de poder matar un proceso en el caso que una aplicación determinada no responda, sin necesidad de un reinicio forzado de nuestro servidor. **(0,1)/1**
- No deben existir las pantallas azules cuando hay una incompatibilidad de hardware o por mal funcionamiento del software porque no podemos tener un servidor en nuestra red que se reinicie. **(1,2)/2**
- Los entornos gráficos deben ser eficientes al momento de ejecutarse e independientes del kernel para poder reiniciarlos en caso de no responder. **(1,2)/2**
- Nuestro servidor no puede reiniciarse forzosamente por la instalación de algún nuevo controlador o por la instalación de alguna nueva actualización de nuestro sistema. **(0,1)/1**

Calificación:

Pc = $\Sigma(z) = 1 + 2 + 2 + 1 = 6$

Pws = $\Sigma(x) = 0 + 1 + 1 + 0 = 2$

Pgnu = $\Sigma(y) = 1 + 2 + 2 + 1 = 6$

Cc-Pws: $Pws / Pc = (2/6)*100\% = 33.33\%$

Cc-GNU: $Pgnu / Pc = (6/6)*100\% = 100\%$

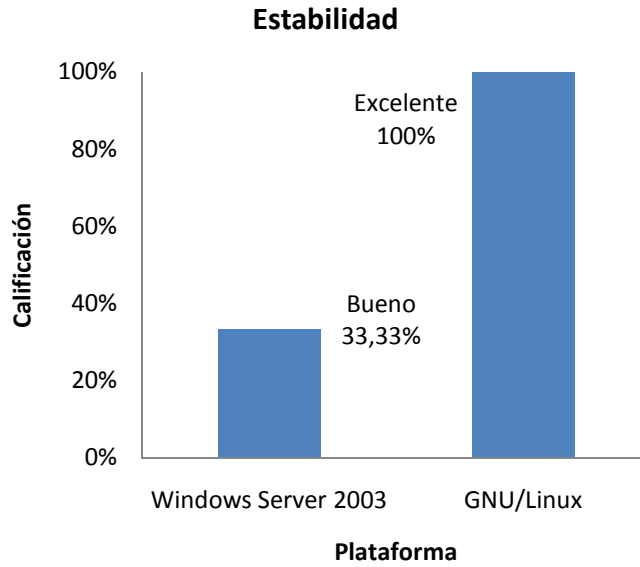


Figura III.21 Estabilidad de las plataformas

Fuente: Análisis del autor

3.3.4 Utilización de Recursos y Rendimiento

La utilización de recursos por parte de un servidor tiene que ser menor para aprovechar el rendimiento del hardware, con esto tenemos un servidor más rápido y más eficiente.

Tabla III-12 Utilización de recursos y rendimiento

	WS2003	GNU/Linux
Consumo de RAM	Si contamos con un computador antiguo el consumo de memoria es alto y no se puede gestionar debido a los requerimientos del SO.	En caso de poseer un computador antiguo, GNU/Linux consume menos memoria y la gestiona mejor. Por lo tanto lee menos de disco duro.
Fragmentación del disco	Es muy común tener una fragmentación del disco en WS2003 por lo que el sistema se pondrá un poco lento para acceder a la información.	En GNU/Linux no existe la fragmentación del disco duro por lo que el sistema no se ve afectado al momento de acceder a la información.
Usuarios del Sistema	Es común tener varios usuarios con privilegios de administrador y un error de un usuario afecta a todos los usuarios, lo cual ralentiza el sistema.	No existe disminución en el rendimiento cuando varios usuarios usan el sistema. Solo existe un usuario con privilegios.

Tabla III-4 Utilización de recursos y rendimiento (Continuación)

	WS2003	GNU/Linux
Paquetes de Actualización	Tenemos que guardar los archivos necesarios antes de instalar las actualizaciones para poder desinstalarlas en caso de tener algún problema con la instalación.	No hace falta guardar los archivos porque cada paquete a actualizar es independiente del sistema y en caso de tener algún problema podemos desinstalar ese paquete e instalar la versión anterior.
Instalación/Desinstalación de Programas	Existe un Registro del sistema para almacenar la información de todos los programas que se instalan con el fin de poder desinstalarlos. Pero con el paso del tiempo el registro se vuelve pesado repercutiendo en su rendimiento.	No existe ningún registro de sistema por lo que al momento de desinstalar paquetes y al pasar el tiempo, no hay problema con el rendimiento de nuestro sistema.

Fuente: Análisis del autor

Interpretación:

- Es típico toparnos con equipos antiguos como servidores por lo que es indispensable poder contar con un sistema que pueda gestionar la memoria para aumentar el rendimiento de los mismos. **(1,2)/2**

- No debe existir una fragmentación del disco duro para que el rendimiento del sistema no se vea afectado en ningún momento. **(0,1)/1**
- La instalación de programas no debe estar sujeto a un registro de sistema porque con el paso del tiempo dicho registro se puede sobrecargar haciendo que el sistema se ralentice. **(1,2)/2**
- En nuestro servidor solo tiene que existir un solo usuario con privilegios de administrador para no tener problemas de configuraciones o instalaciones de programas con errores. **(0,1)/1**
- Debemos poder instalar las actualizaciones sin tener ningún error y sin tener que volver a instalar algún paquete que se actualizo con algún problema. **(1,1)/2**

Calificación:

Pc = $\Sigma(z) = 2 + 1 + 2 + 1 + 2 = 8$

Pws = $\Sigma(x) = 1 + 0 + 1 + 0 + 1 = 3$ **Cc-Pws:** $Pws / Pc = (3/8)*100\% = 37.50\%$

Pgnu = $\Sigma(y) = 2 + 1 + 2 + 1 + 1 = 7$ **Cc-GNU:** $Pgnu / Pc = (7/8)*100\% = 87.50\%$

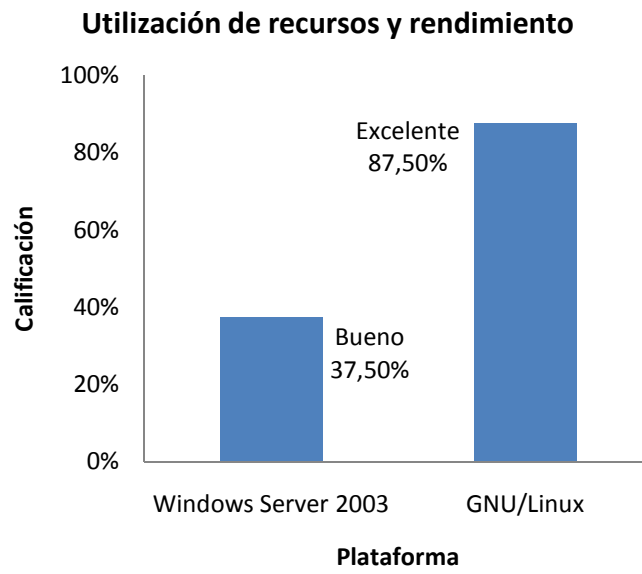


Figura III.22 Utilización de recursos y rendimiento

Fuente: Análisis del autor

3.3.5 Configurabilidad vs. Facilidad de uso

Deben existir posibilidades de configuración útiles sólo para el administrador de la red y personalizar el sistema a su gusto.

Tabla III-13 Configurabilidad vs. Facilidad de uso

	WS2003	GNU/Linux
Compilación del kernel	WS2003 no trae la posibilidad de poder compilar nuevamente el kernel. Esto resulta útil para añadir nuevas características a nuestro sistema.	GNU/Linux trae la opción de compilar el kernel sin necesidad de tener que volver a instalar el sistema operativo. Solo basta con descargar el kernel y activar los módulos necesarios.
Configuración de la red	Tenemos una GUI muy buena para la configuración de nuestra red.	Tenemos dos maneras de configurar nuestra red: <ul style="list-style-type: none">• Mediante la GUI.• Mediante línea de comandos.
Configuración de Servicios	Podemos configurar nuestros servicios mediante una interfaz gráfica y todo se realiza gráficamente.	Podemos configurar nuestros servicios, iniciarlos, detenerlos mediante la línea de comandos lo cual nos da un mayor control.

Fuente: Análisis del autor

Interpretación:

- Muchas veces necesitamos configurar un nuevo módulo y es indispensable poder compilar el kernel de nuestro sistema operativo para cumplir con nuestro objetivo. **(0,1)/1**
- Debemos tener la posibilidad de configurar nuestra red ya sea de manera visual o por línea de comandos, pero la segunda opción es la más recomendada para los usuarios avanzados o administradores. **(1,2)/2**
- La configuración de los servicios deben ser netamente en la línea de comandos y la ejecución puede ser de manera visual o por comandos. **(1,1.5)/2**

Calificación:

Pc = $\Sigma(z) = 1 + 2 + 2 = 5$

Pws = $\Sigma(x) = 0 + 1 + 1 = 2$

Cc-Pws: $Pws / Pc = (2/5)*100\% = 40.00\%$

Pgnu = $\Sigma(y) = 1 + 2 + 1.5 = 4.5$

Cc-GNU: $Pgnu / Pc = (4.5/5)*100\% = 90.00\%$

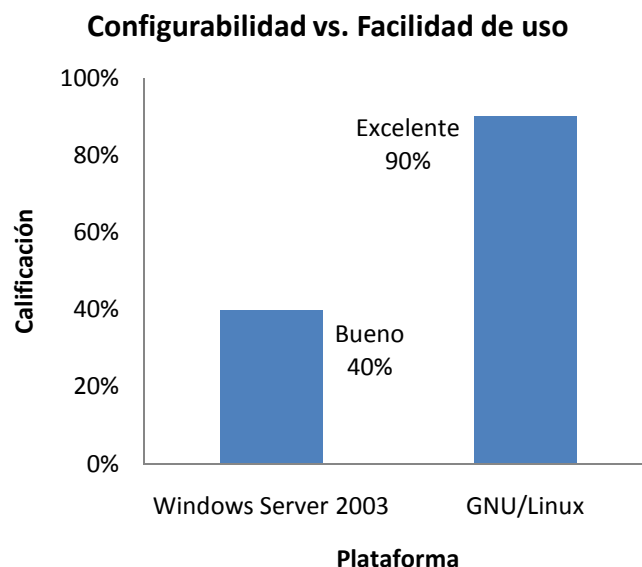


Figura III.23 Configurabilidad vs. Facilidad de uso

Fuente: Análisis del autor

3.3.6 Código abierto vs. Código propietario

En el desarrollo de código abierto no hay presiones comerciales, ni fechas límites ni contratos que cumplir. Los productos salen adelante cuando se considera que están listos. En el desarrollo de código propietario pasa lo contrario debido a que tienen fechas específicas para lanzar un nuevo programa.

Tabla III-14 Código abierto vs. Código propietario

	WS2003	GNU/Linux
Programas propios de la plataforma	WS2003 no trae muchos programas y si queremos instalar programas para una tarea específica, tenemos que comprarlos porque son programas de código propietario.	GNU/Linux trae sólo programas de código abierto por el hecho de ser un sistema operativo que también es código abierto, pocas veces tenemos que descargar algún programa adicional.
Facilidad para conseguir programas	Para WS2003 existen muchos programas de código propietario y pocos programas de código abierto.	Existen muchos programas de código abierto para GNU/Linux, los mismos que son fáciles de conseguir, y muy pocos programas de código propietario.

Tabla III-6 Código abierto vs. Código propietario (Continuación)

	WS2003	GNU/Linux
Estabilidad y eficacia en la ejecución	Los programas nativos que existen para esta plataforma se ejecutan de una manera estable pero muchas veces tienden a colapsar.	Los programas para esta plataforma son netamente código abierto y son 100% compatibles así que su ejecución es excelente y nunca colapsan.
Actualización de los programas	Los programas de este sistema operativo son comerciales y por lo tanto tenemos que esperar hasta que las empresas desarrolladoras lancen alguna actualización o parche de seguridad en caso de haber algún bug.	Los programas de código abierto no tienen fecha de lanzamiento y todos tenemos acceso al código de dichos programas por lo que las actualizaciones se pueden dar de un momento a otro.

Fuente: Análisis del autor

Interpretación:

- Windows Server 2003 tiene programas propios que en su mayoría son comerciales mientras que GNU/Linux tiene programas propios que en su mayoría son de libre distribución. Deberíamos poder contar con programas que sean código libre para ambas plataformas. **(0.5,0.5)/1**

- Debemos tener la posibilidad de poder conseguir o descargar programas desde algún almacén (repositorio) de programas para cada plataforma. **(0,1)/1**
- Los programas de código abierto han demostrado que se pueden ejecutar sin problemas y de manera excelente bajo Windows Server 2003 a pesar de no ser propios de la plataforma Windows. En GNU/Linux dichos programas se ejecutan siempre sin dificultades **(0,5,1)/1**
- En Windows Server 2003 siempre hay que esperar hasta que las empresas comerciales lancen alguna actualización de algún programa mientras que en GNU/Linux no hace falta actualizar los programas muy pronto. **(0,1)/1**

Calificación:

Pc = $\Sigma(z) = 1 + 1 + 1 + 1 = 4$

Pws = $\Sigma(x) = 0.5 + 0 + 0.5 + 0 = 1$ **Cc-Pws:** $Pws / Pc = (1/4)*100\% = 25.00\%$

Pgnu = $\Sigma(y) = 0.5 + 1 + 1 + 1 = 3.5$ **Cc-GNU:** $Pgnu / Pc = (3.5/4)*100\% = 87.50\%$

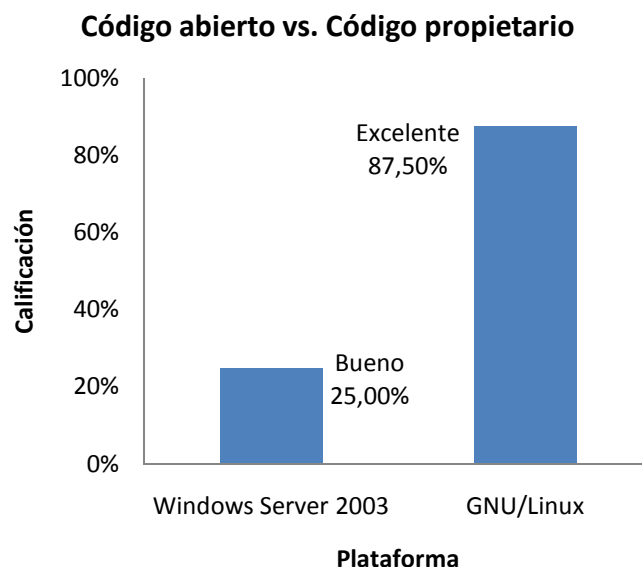


Figura III.24 Código abierto vs. Código propietario

Fuente: Análisis del autor

3.3.7 Sistema de Archivos

Todo SO tiene que contar con un sistema de ficheros tolerante a fallos en el cual la integridad de los datos está asegurada.

Tabla III-15 Sistemas de Archivos

	WS2003	GNU/Linux
Soporte de sistemas de archivos	WS2003 soporta sistemas de archivos propios de Microsoft como FAT, FAT32 y NTFS (en todas sus versiones).	GNU/Linux tiene soporte para muchos sistemas de ficheros como FAT, FAT32, NTFS (solo lectura), EXT2, EXT3, EXT4, ReiserFS, NFS, HPFS, JFS, etc.
Sistema de archivos nativo	<p>NTFS:</p> <ul style="list-style-type: none"> • Tamaño máximo de archivo de 16TB. • Número máximo de archivos igual a 2^{28}. • Tamaño máximo de volumen de 32TB. 	<p>EXT3:</p> <ul style="list-style-type: none"> • Tamaño máximo de archivo de 16TB. • Número máximo de archivos igual a 2^{32}. • Tamaño máximo de volumen de 256TB.
Extensión de archivos	WS2003 asigna extensiones para cada tipo de archivos lo cual puede causar problemas al momento de cambiar la extensión por error.	GNU/Linux no existen las extensiones porque se basa en el contenido de los archivos y si el archivo tiene extensión, ésta forma parte del nombre.

Fuente: Análisis del autor

Interpretación:

- Windows Server 2003 sólo tiene soporte para sistemas de archivos propios de Microsoft. GNU/Linux tiene soporte para su propio sistema de ficheros y de otros sistemas operativos. **(1,2)/2**
- Windows Server 2003 soporta NTFS, el cual tiene ciertas limitantes en relación a Ext3 de GNU/Linux en cuanto al tamaño máximo de volumen y número máximo de archivos **(1,3)/3**
- Un SO tiene que basarse en el contenido de un archivo y no en la extensión del mismo ya que si cambiamos la extensión por error podríamos tener conflictos con nuestro archivo. **(0,1)/1**

Calificación:

Pc = $\Sigma(z) = 2 + 3 + 1 = 6$

Pws = $\Sigma(x) = 1 + 1 + 0 = 2$

Cc-Pws: $Pws / Pc = (2/6)*100\% = 33.00\%$

Pgnu = $\Sigma(y) = 2 + 3 + 1 = 6$

Cc-GNU: $Pgnu / Pc = (6/6)*100\% = 100\%$

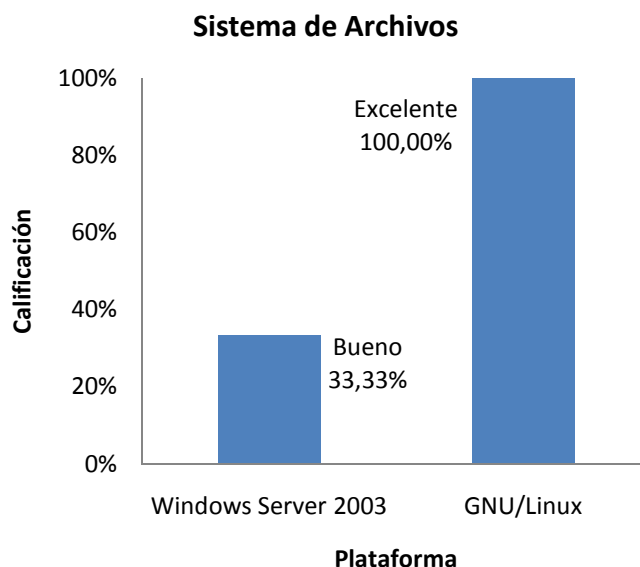


Figura III.25 Sistema de Archivos

Fuente: Análisis del autor

3.3.8 Seguridad y Precio

La seguridad y el precio a la hora de instalar un SO en un servidor es algo a tomar en cuenta si queremos que nuestro servidor este protegido contra ataques y mal uso de usuarios inexpertos.

Tabla III-16 Seguridad y Precio

	WS2003	GNU/Linux
Precio de licencias	WS2003 tiene precio de licencia dependiendo de la versión que queramos adquirir. Una licencia cuesta entre \$400 y \$3000.	GNU/Linux no tiene precio de licencia por lo que resulta barata su implementación.
Aplicaciones y administración de usuarios	En WS2003 tanto los usuarios como las aplicaciones por lo regular operan con privilegios de Administrador.	En GNU/Linux, es todo lo contrario, el usuario root sólo se utiliza para realizar configuraciones del SO.
Acceso remoto	WS2003 usar RPC para conectarse remotamente, el cual es inseguro y muchas veces fue atacado por virus como el Blaster, Sasser y Slammer.	GNU/Linux no depende del servicio RPC ya que cuenta con una herramienta opensource llamada ssh (Security Shell - Shell seguro) que hasta el momento no ha sido atacado.

Tabla III-8 Seguridad y Precio (Continuación)

	WS2003	GNU/Linux
Parches de seguridad	WS2003 lanza mensualmente parches de seguridad porque cada vez hay nuevos virus interesados en demostrar que Windows tiene muchos agujeros de seguridad.	En GNU/Linux no es necesario aplicar parches de seguridad cada mes a menos que alguien encuentre algún agujero de seguridad y publique el parche (cosa que no pasa muy seguido).

Fuente: Análisis del autor

Interpretación:

- Windows Server 2003 tiene un costo de licencia muy elevado en relación a los requerimientos de los usuarios mientras que GNU/Linux no tiene precio y nos brinda muchas posibilidades al igual que su competidos. **(0,1)/1**
- No es necesario ejecutar las aplicaciones como administrador del sistema debido al riesgo que puede correr nuestro servidor. **(0,1)/1**
- Windows Server 2003 tiene su propio programa para comunicación remota y no es muy segura a menos que apliquemos algún parche proporcionado por Microsoft mientras que GNU/Linux tiene también su propio programa para conexión remota y no necesita de parches. **(0.5,1)/1**
- Windows Server 2003 es un SO que es muy atacado por hackers y por tal motivo Microsoft tiene que lanzar parches de seguridad. En GNU/Linux no hace falta aplicar parches de seguridad porque el SO no es atacado debido a su seguridad. **(0.5,1)/1**

Calificación:

Pc = $\Sigma(z) = 1 + 1 + 1 + 1 = 4$

Pws = $\Sigma(x) = 0 + 0 + 0.5 + 0.5 = 1$ **Cc-Pws:** $Pws / Pc = (1/4)*100\% = 25.00\%$

Pgnu = $\Sigma(y) = 1 + 1 + 1 + 1 = 4$ **Cc-GNU:** $Pgnu / Pc = (4/4)*100\% = 100\%$

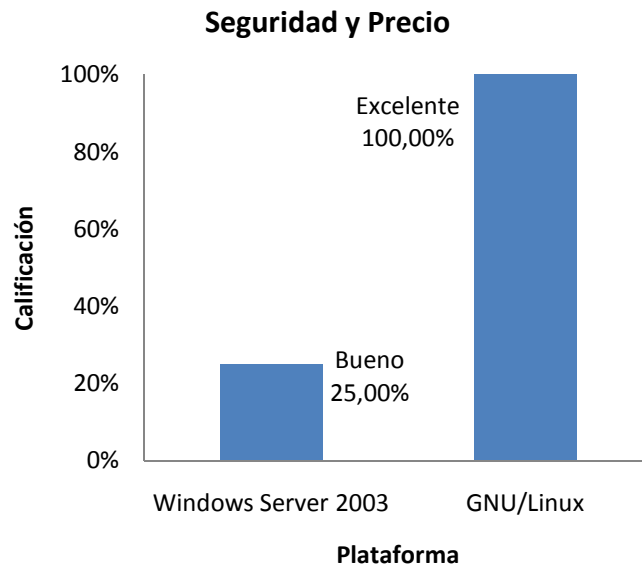


Figura III.26 Seguridad y Precio

Fuente: Análisis del autor

3.4 ANALISIS DE RESULTADOS DEL ESTUDIO COMPARATIVO

3.4.1 Comparativo General

El estudio de las dos plataformas siempre evidencia un conjunto de semejanzas y diferencias entre las mismas, las cuales deben ser consideradas al momento de inclinarse por una de ellas en particular.

El conjunto de semejanzas y diferencias obtenidas con el presente estudio se resumen en las siguientes secciones:

3.4.1.1 Semejanzas y Diferencias

Semejanzas

Las dos plataformas usadas para la comparativa del diseño e implementación de un laboratorio de software y redes tienen características comunes y propias de ellas tales como:

- Servidores Web, DNS, DHCP, entre otros.
- Configuración del sistema mediante GUI.
- Administración de Grupos y Usuarios del sistema.
- Se puede trabajar con varias consolas de texto a la vez.
- Utilidad de disco para escoger en que disco y en que partición vamos a instalar nuestro SO. También podremos crear una nueva partición.
- Ambas plataformas se pueden instalar en equipos diseñados para servidor o en equipos de escritorio.
- Ventana de Registro para iniciar sesión.
- Atajos de teclado similares como Alt+F4 para cerrar ventanas.
- Acceso Remoto para la administración

Diferencias

Las plataformas de estudio poseen muchas características que las diferencian entre sí, las más importantes se mencionan a continuación:

- **Completo acceso vs. No acceso.** Tener acceso al código fuente es probablemente la diferencia más significativa entre GNU/Linux y Windows Server 2003. El hecho de que GNU/Linux posee una licencia pública asegura que el usuario de cualquier manera pueda acceder y alterar el código. ¿Podemos revisar el código fuente de Windows? La respuesta es no. A menos que seamos miembro de un selecto grupo (elite), nuestros ojos nunca verán una pieza del código fuente de Windows Server 2003.
- **Libertad de licencias vs. Restricciones de licencias.** El tener acceso es la diferencia entre las licencias. Cualquier experto en tecnología e internet puede entender sobre licencias de computadoras. Pero veamos lo que es la clave en el aspecto de las licencias (sin inmiscuirnos en el ámbito legal). Con GNU/Linux que este con licencia GPL, somos libres de modificar el software, distribuirlo y venderlo. También con la licencia GPL tenemos libertad de descargar una copia de GNU/Linux e instalarlo en cuantas máquinas deseemos. Con las licencias de Microsoft no podemos hacer nada de lo ya mencionado. El límite esta en cuantas licencias compremos. Así que si compramos 10 licencias Windows Server 2003, legalmente podemos instalar ese SO en solo 10 máquinas.
- **Soportes de conexiones en línea vs. Soporte de escritorio bajo tarifa.** Este es un problema por el cual muchas empresas le dan la espalda a GNU/Linux. Con GNU/Linux tenemos soporte de una gran comunidad en foros, búsquedas en línea y muchas páginas web. Por supuesto, si sentimos necesidad podemos adquirir contratos de soporte de algunas grandes

empresas que manejan GNU/Linux (RedHat o Novell por ejemplo). Si tenemos algún problema, debemos exponerlo en algún servicio de ayuda y en pocos minutos estaremos recibiendo ayuda y sugerencias. La mayoría de los problemas en GNU/Linux son documentados así que es probable que encontremos una solución a nuestro problema. Podemos comprar soporte de la propia empresa Microsoft para Windows Server 2003. La mayoría de las empresas que contratan soporte de Microsoft se han quejado de la seguridad que esta carece, así que podemos comprar soporte de GNU/Linux y Windows.

- **Soporte de Hardware completo vs. Soporte de hardware en solo una parte.** Un problema que se está convirtiendo en un problema que poco a poco va desapareciendo es el del hardware. Años atrás si queríamos instalar algún tipo de distribución de GNU/Linux teníamos que estar completamente seguros que nuestro hardware se ajustara perfectamente a la distribución para que trabajara al 100%. Este no es el caso de ahora, hoy podemos tomar cualquier PC o laptop y tener varias distribuciones GNU/Linux para instalar y casi siempre funcionara al 100%. Claro existen algunas excepciones por ejemplo el hibernar/suspender algunas veces es un problema en las laptops. Con Windows sabemos que cada pieza siempre funcionará con ese SO. Sin embargo existen también casos en los que podemos encontrar algunos contratiempos.
- **La línea de comandos vs. No línea de comandos.** No importa lo lejos que el SO GNU/Linux haya llegado o lo bonito que luzca, la herramienta de línea de comandos siempre será una invaluable herramienta de administración. Podemos utilizar GNU/Linux por años y no utilizar la línea de comandos al igual que con Windows Server 2003. Algunos usuarios de Windows Server 2003 ni siquiera saben que el cmd existe.

- **Instalación de aplicaciones centralizadas vs. Las descentralizadas.**
Con GNU/Linux tenemos (en casi todas las distribuciones) un punto centralizado donde podemos buscar, agregar y borrar programas. Es un paquete de gestión de programas llamado Synaptic. Con Synaptic podemos abrir, buscar una aplicación, instalarla y todo eso, sin hacer ningún tipo de búsqueda. Windows Server 2003 no tiene nada comparado con esto. Con Windows Server 2003 tenemos que buscar el software/programa que queramos utilizar o insertar el CD/DVD del instalador en el equipo y correr el setup.exe o install.exe. Hasta ahora en GNU/Linux la instalación de programas es simple, centralizada y sin dolores de cabeza.
- **Rigidez vs. Flexibilidad.** Comparamos a GNU/Linux y a Windows Server 2003 con un cuarto donde el piso y el techo son movibles o no. Con GNU/Linux tenemos un cuarto donde el techo y el piso pueden ser movidos tan altos como queramos. Con Windows Server 2003 ese piso y ese techo son inmóviles. No podemos ir más allá de donde Microsoft haya dicho que podamos ir. Por ejemplo: los escritorios con GNU/Linux podemos hacerlos lucir como queramos que luzcan, existe la experiencia Compiz. Con Microsoft tenemos que manejar el escritorio que ellos han decidido que sea nuestro escritorio perfecto.
- **Automatizado vs. Contenido removible no automatizado.** Años atrás teníamos que montar un floppy para usarlo y sacarlo para dejarlo de usar, esos días llegaron a su fin. Un problema que altera a los nuevos usuarios de GNU/Linux es como se utilizan los contenidos que son removibles. La idea de montar un CD cada vez que lo vayamos a usar no queda muy claro para dichos usuarios que son nuevos en GNU/Linux.
- **Correr a nivel múltiple vs. Correr en un solo nivel.** GNU/Linux nos da la posibilidad de detenernos a distintos niveles. Con esto podemos trabajar

desde la línea de comandos (nivel 3) o desde la interfaz gráfica (nivel 5). Esto en realidad nos puede salvar cuando tenemos problemas con nuestro servidor X Window. En Windows Server 2003 seremos afortunados si podemos trabajar en la línea de comandos en Modo a prueba de fallos y puede que no tengamos las herramientas para arreglar el problema. En GNU/Linux aún en línea de comandos a nivel 3, podemos instalar utilidades (yum -y install NOMBRE). Trabajar en diferentes niveles es de alguna manera benéfico.

3.4.1.2 Ventajas y Desventajas

GNU/Linux

Ventajas:

- La gran mayoría de los ataques de hackers son dirigidos a servidores Windows Server 2003 al igual que los virus los cuales se enfocan principalmente a servidores con éste SO.
- La plataforma GNU/Linux es más robusta lo cual hace más difícil que algún intruso pueda violar el sistema de seguridad de éste SO.
- Al tener una plataforma más estable, esto favorece el desempeño de aplicaciones de todo tipo tales como: bases de datos, aplicaciones XML, multimedia, etc.
- La eficiencia de su código fuente hace que la velocidad de las aplicaciones GNU/Linux sean superiores a las que corren sobre cualquier otro SO, lo cual se traduce en velocidad del sistema.
- Un servidor GNU/Linux requiere menor mantenimiento y por lo tanto es más económico. En otras plataformas de servidores es más costoso debido a que es necesaria una frecuente atención y

monitoreo contra ataques de virus, hackers y errores de código, instalación y actualización de parches y Service Packs.

- El SO GNU/Linux y así como otras aplicaciones son Open Source y Gratuitos.
- No requieren supervisión tan estrecha ni pagos de pólizas de mantenimiento necesarias para obtener los Service Packs.
- Puede montar particiones de otras plataformas como: Unix, Mac, Windows, etc.

Desventajas:

- No tiene una página oficial que nos brinde soporte en línea.
- Algunas distribuciones tienen paquetes con su propia extensión.
- No está diseñado para multimedia.
- No existen equivalentes libres de programas comerciales tales como: AutoCAD o Adobe Illustrator.

Windows Server 2003

Ventajas:

- Al ser de mayor facilidad de uso Windows Server 2003 en este momento continúa siendo el SO más comercial lo cual se refleja en la disponibilidad de aplicaciones, facilidad de mantenimiento así como soporte en el desarrollo de nuevas aplicaciones. Puntos que pueden ser cruciales en la elección de servidores que corren aplicaciones web.
- Fruto de la inversión realizada por Microsoft y aunado a una comunidad de programadores cada vez más grande se ha logrado facilitar el desarrollo de aplicaciones y sistemas que corran sobre

servidores Windows lo cual se ve reflejado en tiempos de desarrollo menores.

- Está diseñado para aprovechar al máximo las aplicaciones multimedia como video conferencias.
- Los Servidores pueden instalarse y configurarse visualmente sin necesidad de tener que recurrir a la línea de comandos.

Desventajas:

- El precio de una licencia es muy elevada.
- Sólo puede montar particiones propias de su plataforma.
- Necesita aplicar parches de seguridad con mucha frecuencia.
- Las particiones sufren a menudo de fragmentaciones lo cual hace que el sistema se vuelva lento.

3.4.1.3 Resultados

El Puntaje Final y el % sobre el Total de cada plataforma se los obtiene de la siguiente manera:

Puntaje Total de la Evaluación: (PT) = Σ (Pc)

Puntaje Final de Windows Server 2003 (PFws) = Σ (Pws)

Puntaje Final de GNU/Linux (PFgnu) = Σ (Pgnu)

Porcentaje Final de Windows Server 2003 (%WS) = (PFws/PT) * 100%

Porcentaje Final de GNU/Linux (%GNU) = (PFgnu/PT) * 100%

PT = 7 + 11 + 6 + 8 + 5 + 4 + 6 + 4 = 51

PFws = 3.5 + 9 + 3 + 3 + 2 + 1 + 2 + 1 = 24.5

$$\text{PFgnu} = 6 + 9.5 + 6 + 7 + 4.5 + 3.5 + 6 + 4 = 46.5$$

$$\%WS = (24.5/51) * 100\% = 48.04\%$$

$$\%GNU = (46.5/51) * 100\% = 91.18\%$$

Un Resumen de los resultados obtenidos mediante este Estudio Comparativo se muestra a través de la siguiente figura:

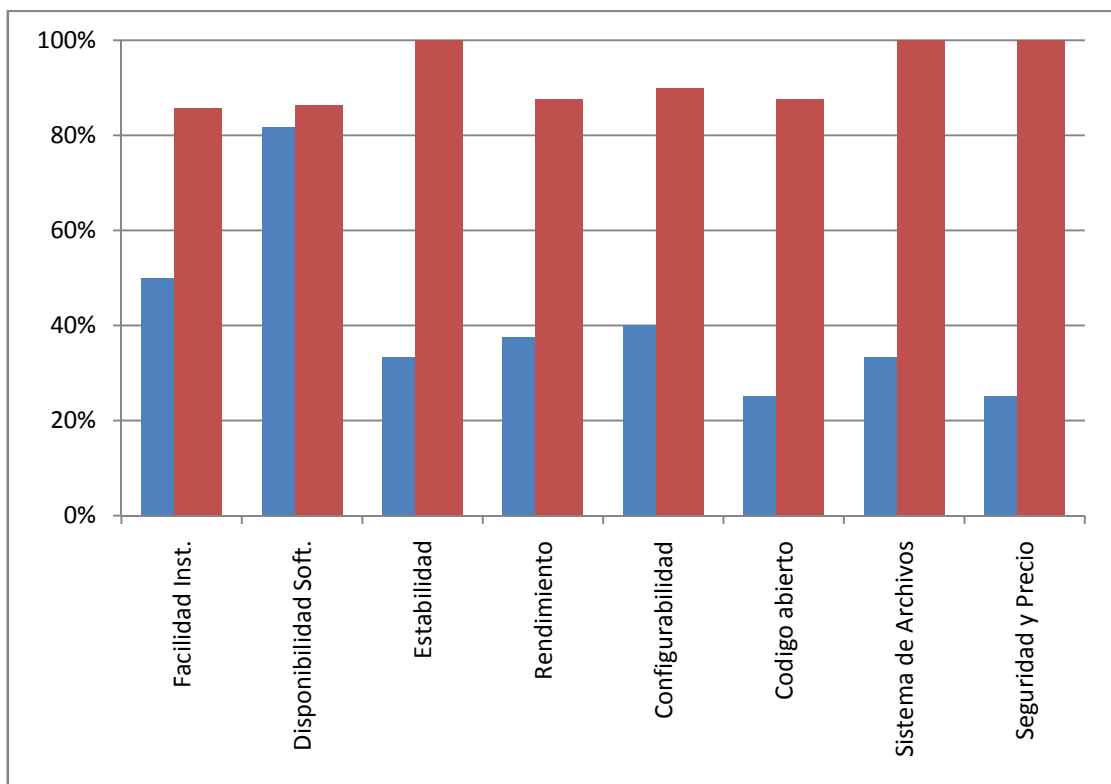


Figura III.27 Windows Server 2003 vs. GNU/Linux

Fuente: Análisis del autor

El Histograma anterior muestra claramente la superioridad de la plataforma GNU/Linux sobre su contendiente Windows Server 2003 en la mayoría de aspectos, siendo así que las plataformas obtienen un Porcentaje Final de **91.18%** y **48.04%** equivalentes a Excelente y Bueno respectivamente. Ver Figura 3.10.

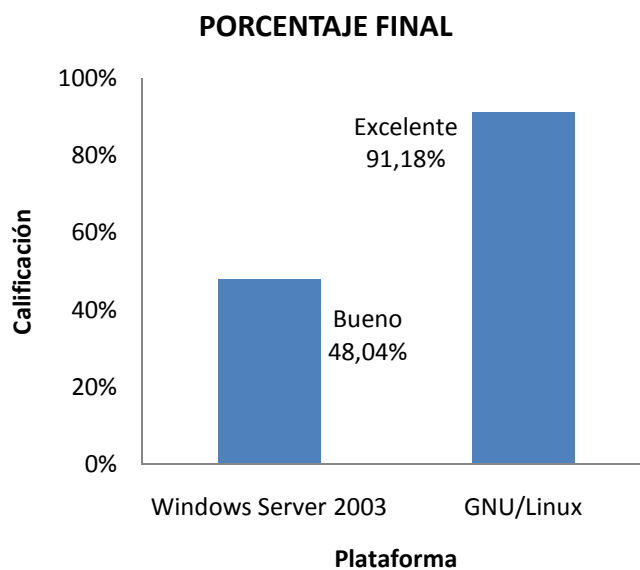


Figura III.28 Porcentaje Final Windows Server 2003 - GNU/Linux

Fuente: Análisis del autor

3.4.2 Conclusiones

1. La plataforma GNU/Linux nos permite escoger los paquetes (programas) que requeriremos justo en el momento de la instalación del SO o también podemos instalarlos posteriormente.
2. La interfaz gráfica de Windows Server 2003 es muy buena pero gracias a los aportes de muchos programadores en el mundo, la interfaz gráfica de GNU/Linux es igual o incluso mejor que la de su competidor.
3. La disponibilidad de software para las dos plataformas es muy extensa, y hoy en día se puede encontrar software para cualquier requerimiento que tengamos en nuestro servidor.
4. En un inicio GNU/Linux era un SO sólo para expertos debido a que se creía que solo se lo podía usar en servidores pero en la actualidad cualquier usuario puede usarlo sin necesidad de tener conocimientos avanzados.
5. En Windows Server 2003 tenemos que instalar los servidores que necesitemos después de su instalación, no podemos elegirlos al momento de instalar el SO.
6. Podemos tener un servidor seguro con Windows Server 2003 pero los costos son muy elevados mientras que en GNU/Linux tenemos un costo igual a 0.
7. En un SOR debemos contar con un repositorio (almacén o bodega) donde podamos tener todos los paquetes que necesitemos, para no depender de los CDs de instalación.
8. En GNU/Linux no hace falta que tengamos un antivirus mientras que en Windows Server 2003 es indispensable contar con un buen antivirus para proteger el servidor.
9. Los errores o problemas de seguridad en los programas para GNU/Linux se corrigen rápidamente debido a que son Open Source y cualquier persona puede acceder a su código fuente para corregir los fallos.

10. Ambas plataformas tienen sistemas de archivos journaling para impedir que los datos se corrompan pero en Windows Server 2003 tenemos fragmentaciones de disco con mucha frecuencia.
11. En GNU/Linux podemos visualizar desde una consola de comandos los procesos que se están ejecutando en el sistema y en caso de que algún proceso no responda, podemos terminarlo mediante su PID (Identificador de proceso).
12. Contamos con servidores propios de GNU/Linux que también tienen su equivalente para Windows Server 2003 y que han demostrado estabilidad en su ejecución.
13. Windows Server 2003 está sujeto a leyes de copyright, lo cual impide que se pueda hacer copias del SO para instalarlo más de un servidor. GNU/Linux está sujeto a leyes de copyleft que nos permiten hacer copias y redistribuirlo para ser instalado en más de un servidor.

3.4.3 Recomendaciones

1. Escoger únicamente los paquetes necesarios a ser instalados en nuestro servidor porque de lo contrario se puede llegar a ocupar demasiado espacio en disco.
2. Habilitar sólo los procesos necesarios en nuestro servidor y deshabilitar los procesos innecesarios para garantizar rapidez y mejorar el rendimiento de nuestro SOR.
3. En caso de necesitar paquetes específicos en nuestro servidor, es recomendable crear nuestro propio repositorio para instalarlos en cualquier momento.
4. En caso utilizar Windows Server 2003, se recomienda tener un antivirus instalado y actualizado. Además de herramientas anti espías para asegurar nuestro servidor al máximo.
5. Acceder remotamente a nuestro servidor de una manera segura para evitar que los hackers puedan obtener la contraseña del administrador de nuestro servidor.
6. Aprender todos los comandos usados para la administración de nuestro servidor ya que podríamos necesitar reiniciar nuestro servidor sin interfaz gráfica para solucionar problemas desde la línea de comandos.
7. Usar herramientas que nos permitan desfragmentar los discos duros del servidor para mejorar la velocidad de acceso a la información.
8. Respalidar la información al menos una vez por semana para poder recuperarla en caso de ser atacados por hackers.
9. Evitar que los usuarios puedan ejecutar programas con privilegios de administrador debido a que el servidor se vuelve vulnerable en ese preciso momento.

CAPÍTULO IV

CONFIGURACIÓN E INSTALACIÓN DE LTSP (LINUX TERMINAL SERVER PROJECT)

4.1 ¿QUÉ ES LTSP?

Linux Terminal Server Project (LTSP) nos brinda una manera simple de utilizar estaciones de trabajo de bajo costo tanto como terminales gráficas o bien como terminales de caracteres sobre un servidor GNU/Linux.

En una configuración de oficina tradicional hay PCs de bastante poder ubicadas en cada escritorio. Cada una con varios gigabytes de espacio en disco. Los usuarios almacenan su información en sus discos locales y las copias de respaldo se realizan raramente (si es que se realizan).

Afortunadamente, hay otro camino. Utilizando LTSP, se puede tomar cada PC de bajo rendimiento, removerle el disco duro, el floppy y la lectora de CDs y agregarle una placa de red con chip de inicio. Muchas tarjetas de red poseen sócalos para bootroms (memorias de inicio), que están esperando que se les inserte uno de estos chips.

Durante la fase de inicio, la estación de trabajo sin disco obtiene su dirección IP y un kernel desde el servidor, montando luego su sistema de archivos raíz desde el mismo servidor vía NFS.

La estación de trabajo puede ser configurada de tres maneras:

X Window Gráfico

Utilizando X Window, la estación de trabajo puede ser usada para acceder a cualquier aplicación gráfica en el servidor o en cualquier otro servidor dentro de la red.

Sesiones de Telnet basadas en caracteres

Cada estación de trabajo puede invocar múltiples sesiones de telnet al servidor. Cada una de estas sesiones aparecerá en una pantalla virtual separada. Presionando desde Alt+F1 hasta Alt+F9 se cambiará entre cada una de dichas sesiones.

Prompt de Shell

La estación de trabajo puede ser configurada para dejarnos justo dentro de una sesión de Bash en la consola. Esto es muy útil para depurar problemas con X Window o con NFS.

Lo que es realmente impresionante de todo esto es que se puede tener montones de estaciones de trabajo, todas servidas desde un único servidor GNU/Linux. ¿Hasta cuántas estaciones nos preguntamos? Pues bien, esto depende del tamaño del servidor y de las aplicaciones que se han de utilizar.

4.2 TEORÍA DE OPERACIÓN

Iniciar una DC involucra muchos pasos. Comprender qué está pasando a través del camino hará mucho más fácil resolver los problemas que pudiesen surgir.

Existen 4 servicios básicos requeridos para arrancar un cliente LTSP, que son:

- DHCP
- TFTP
- NFS
- XDMCP

LTSP es muy flexible. Cada uno de los servicios descritos anteriormente puede ser servido desde el mismo servidor o desde diferentes servidores. Se describirá una configuración simple que consta de un solo servidor.

4.2.1 Pasos a Seguir por el Cliente LTSP

1. Carga del kernel Linux en la memoria del cliente. Esto se lo puede realizar de muchas maneras:
 - Bootroom (Etherboot, PXE, MBA, Netboot).
 - Disquete.
 - Disco duro.
 - CD-ROM.
 - Dispositivo USB.

Cada uno de los métodos de carga del kernel será descrito más adelante para su mejor comprensión.

2. Una vez que el kernel es cargado en la memoria, se continuará con la ejecución del mismo.
3. El kernel inicializará el sistema entero y todos los dispositivos E/S que se encuentren conectados al cliente.
4. Durante el proceso de carga del kernel, una imagen de *ramdisk* (disco de memoria) también será cargada en la memoria del cliente y será montada temporalmente como el sistema de archivos raíz. Un argumento *root=/dev/ram0* en la línea de comandos del kernel le dice al mismo que debe montar la imagen como su directorio raíz.
5. Normalmente cuando el núcleo finaliza su inicialización, lanza al programa *init*. Pero en este caso, hemos instruido al kernel para que cargue un script shell. Hacemos esto mediante el argumento *init=/linuxrc* en la línea de comandos del kernel.
6. El script */linuxrc* comienza escaneando el bus PCI en busca de alguna tarjeta de red. Para cada dispositivo PCI que escanea, realiza una búsqueda en el archivo */etc/niclist* para verificar la existencia de dicho dispositivo. Una vez que un dispositivo es encontrado, el nombre del módulo de la interface de red es retornado para que sea posteriormente cargado en el kernel. Para las

placas ISA, el nombre del módulo debe estar especificado en la línea de comandos del kernel, junto con cualquier parámetro de IRQ o dirección de memoria que fuera requerido.

7. Una vez que la placa de red ha sido identificada, el script */linuxrc* cargará el módulo de kernel que soporta a dicha tarjeta.
8. Un pequeño cliente DHCP llamado *dhclient* correrá entonces, para realizar otra petición al servidor DHCP. Necesitamos hacer esta petición por separado porque necesitamos más información de la que la memoria ROM de inicio recibió con la primera consulta de DHCP.
9. Cuando *dhclient* obtenga una respuesta desde el servidor, correrá el archivo */etc/dhclient-script*, el cual tomará la información recogida y configurará la interface *eth0*.
10. Hasta este punto, el sistema de archivos raíz había sido un disco RAM. Ahora, el script */linuxrc* montará un nuevo sistema de archivos raíz a través de NFS. El directorio exportado por el servidor por lo general es */opt/ltsp/i386*. No puede tan sólo montar al nuevo sistema de archivos como */*. Debe primero montarlo en */mnt*. Luego, ejecutará *pivot_root* (pivotaje raíz). *pivot_root* realizará un intercambio del sistema de archivos corriente por uno nuevo. Cuando esto esté completado, el sistema de archivos NFS será montado como */*, y el viejo será montado en */oldroot*.
11. Una vez completado el montaje y el pivotaje del nuevo sistema de archivos raíz, se acaba el script */linuxrc* y necesitamos invocar un programa real de */sbin/init*.
12. *init* leerá el archivo */etc/inittab* y comenzará a configurar el ambiente de la DC.
13. Uno de los primeros puntos en el archivo de *inittab* es el comando *rc.sysinit* que se ejecutará mientras la estación de trabajo esté en el estado de "*sysinit*".

14. El script *rc.local* creará un disco RAM de 1 MB que contendrá todas las cosas que se necesitan ser escritas o modificadas de algún modo.
15. El disco RAM será montado en el directorio */tmp*. Cualquier archivo que necesite ser escrito o modificado existirá realmente en el directorio */tmp*, con algún enlace simbólico apuntando hacia él.
16. El sistema de archivos */proc* es montado.
17. El archivo *lts.conf* será descompuesto gramaticalmente y todos los parámetros con éste archivo que estén relacionados con las estaciones de trabajo serán asignados como variables de entorno para su uso en el script *rc.sysinit*.
18. Si la DC está configurada para realizar intercambio sobre NFS, el directorio */var/opt/lts/swapfiles* del servidor será montado en */tmp/swapfiles* en el cliente. Luego, si no hay ningún archivo de intercambio para esta estación de trabajo que haya sido creado con anterioridad, se creará uno automáticamente. El tamaño de este archivo de intercambio está configurado en el archivo *lts.conf*. El archivo de intercambio será activado, usando el comando *swapon*.
19. La interface de red *loopback* es configurada. Esta es una interface de red que tiene la dirección IP 127.0.0.1.
20. Si la opción de Aplicaciones Locales (Local Apps) está habilitada, entonces el directorio */home* será montado para que las aplicaciones puedan utilizar los directorios personales de los usuarios.
21. Varios directorios son creados en el sistema de archivos */tmp* para que contengan algunos de los ficheros transitorios que son necesarios mientras el sistema se está ejecutando. Estos son:
 - */tmp/compiled*
 - */tmp/var*
 - */tmp/var/run*

- /tmp/var/log
- /tmp/var/lock
- /tmp/var/lock/subsys

Todos estos serán creados si no existen en el servidor.

22. El archivo */tmp/syslog.conf* será creado. Este archivo contendrá información que le dirá al demonio *syslogd* hacia qué máquina de la red se le debe enviar la información de los *logs* (historiales). Esta máquina es especificada en el archivo *lts.conf*. Hay un enlace simbólico llamado */etc/syslog.conf* que apunta hacia */tmp/syslog.conf*.
23. El demonio *syslogd* es iniciado, usando el archivo de configuración creado en el paso anterior.
24. Una vez que el script *rc.sysinit* haya finalizado, el control regresa al programa */sbin/init* el cual cambia el *runlevel* (nivel de ejecución) a 5.
Eso causará que algunas de las entradas del archivo */etc/inittab* sean ejecutadas.
25. Por defecto, estas entradas en *inittab* ejecutarán el script */etc/screen_session* en *tty1*, *tty2* y *tty3*. Lo que quiere decir que podemos ejecutar 3 sesiones al mismo tiempo y el tipo de sesión es controlado por las entradas *SCREEN_01*, *SCREEN_02* y *SCREEN_03* en el archivo *lts.conf*.
Más entradas pueden configurarse en *inittab* para poder tener más sesiones, en caso de ser necesario.
26. Si *SCREEN_01* tiene asignado el valor *startx*, entonces el script */etc/screen.d/startx* será ejecutado e iniciará el sistema X Window dándole una interface gráfica al usuario.
En el archivo *lts.conf* existe un parámetro llamado *XSERVER*. Si dicho parámetro falta o tiene asignado el valor "auto", se intentará una detección automática de la tarjeta de video. Si la tarjeta es PCI o AGP tendremos que

conseguir el id del distribuidor y el tipo de dispositivo para verificar en el archivo */etc/vidlist*.

Si la tarjeta es soportada por *Xorg*, la rutina *pci_scan* devolverá el nombre del controlador del módulo. Si solo es soportada por *XFree86*, *pci_scan* devolverá el nombre del servidor X a ser usado. El script *startx* puede indicar la diferencia entre usar el servidor *XFree86* o *Xorg* dependiendo de qué tan vieja o nueva sea la tarjeta de video.

27. Si *Xorg* es usado, el script */etc/build_x4_cfg* será invocado para generar el archivo *XF86Config*. Si *XFree86* es usado, el script */etc/build_x3_cfg* será invocado para generar el archivo *XF86Config*. Estos archivos estarán en el directorio */tmp*, el cual es un ramdisk visto únicamente por la estación de trabajo.

El archivo *XF86Config* será creado en base a las entradas que contenga el archivo *lts.conf*.

28. Una vez que el archivo *XF86Config* ha sido creado, el script *startx* iniciará el servidor X con el nuevo archivo de configuración.

29. El servidor X enviará una petición *XDMCP* al servidor LTSP, el cual le ofrecerá una ventana de diálogo llamada *ventana de login* (registro de usuario).

30. En este punto el usuario puede entrar al sistema, consiguiendo una sesión desde el servidor.

La manera en que funciona LTSP confunde a mucha gente al principio porque están sentados en una estación de trabajo pero ejecutan una sesión en el servidor. Todos los comandos y programas serán ejecutados en el servidor pero la salida será desplegada en la DC.

4.2.2 Carga del Kernel en la Memoria

La obtención del kernel Linux para las DCs puede llevarse a cabo de las 2 siguientes maneras:

- Boot ROM.
- Medios Locales.

4.2.2.1 Boot ROM

- Etherboot

Etherboot es un proyecto Open Source de arranque muy popular. Contiene controladores para las tarjetas de red más conocidas y funciona muy bien con el proyecto LTSP.

El kernel Linux debe ser preparado con *mknbi-linux*, el cual permitirá que el kernel se cargue desde la red, anteponiendo al kernel algo de código adicional y añadiendo el *initrd* al final del mismo.

Los núcleos (kernel) que son usados con el proyecto LTSP ya vienen preparados y listos para ser cargados desde la red. Etherboot también puede grabarse en un disquete para permitir cargar el kernel desde la red.

- PXE

"Abreviatura de Pre-Boot eXecution Environment (Entorno de ejecución previo al arranque). Pronunciado como "pixie", PXE es uno de los componentes propios de Intel.

Un arranque PXE no puede cargar archivos de más de 32 KB pero el kernel Linux es más grande que eso. Por lo tanto nuestra configuración PXE nos permite cargar un segundo escenario del cargador de arranque llamado *pxelinux* que es lo suficientemente pequeño para ser cargado en memoria y permitirnos cargar archivos de mayor tamaño como el kernel Linux.

- MBA

Agente administrador de arranque (MBA, Managed Boot Agent), es un gestor de arranque que pertenece a una empresa llamado emBoot. emBoot solía ser parte de Landworks de 3Com. MBA en realidad es 4 gestores de arranque en uno que incluye: *PXE*, *TCP/IP*, *RPL* y *Netware*.

La implementación de MBA con PXE funciona muy bien pero también se puede usar *pxelinux* para cargar un kernel Linux.

El método *TCP/IP* puede usarse pero primero se debe preparar el kernel con una utilidad llamada *imggen*.

- Netboot

Netboot es un proyecto de software libre al igual que Etherboot, que provee imágenes de memorias de arranque para cargar un kernel. La diferencia es que Netboot es un controlador NDIS empaquetado o paquete de controladores de las tarjetas de red que son enviados con el producto por los fabricantes.

4.2.2.2 Medios Locales

- Disquete

Existen dos maneras de arrancar un cliente LTSP mediante un disquete. La primera es cargando Etherboot en el sector de arranque del disquete que actuará exactamente de la misma manera que una memoria de arranque. El código de arranque será ejecutado, la tarjeta de red inicializada y el kernel Linux será cargado desde el servidor.

También se puede grabar el kernel e *initrd* a un disquete lo cual resulta más rápido para cargar el kernel en una red.

- Disco duro

El Disco duro puede ser usado con LILO o GRUB para cargar el kernel Linux e *initrd*. O también se puede cargar la imagen Etherboot desde el disco duro y actuará como una memoria de arranque.

- CD-ROM

Un CD-ROM arrancable puede cargar o un kernel Linux o una imagen de arranque Etherboot.

- Dispositivo USB

Al igual que un CD-ROM, un disquete o un disco duro se puede usar un dispositivo de memoria USB para arrancar un módulo Etherboot, un kernel Linux completo o una imagen *initrd*.

4.3 INSTALANDO LTSP EN EL SERVIDOR

En nuestro servidor podemos instalar cualquier distribución GNU/Linux, tenemos que escoger la que mejor se acople a nuestros requerimientos. No existe ningún requisito sobre una distribución en particular, el único requisito es que nuestro servidor pueda soportar NFS (Network File System, Sistema de archivo de red).

Hay tres pasos a seguir para instalar y configurar un servidor LTSP, que son:

- Instalar las utilidades de LTSP (LTSP Utilities).
- Instalar los paquetes cliente de LTSP (LTSP client packages).
- Configurar los servicios necesarios de LTSP.

4.3.1 Instalar las Utilidades de LTSP (LTSP Utilities)

Empezamos con LTSP versión 4.2 update2 que tiene un paquete de utilidades para instalar y administrar los paquetes cliente LTSP (Todo lo que se ejecuta en las DCs), y para configurar los servicios en nuestro servidor.

La utilidad de administración se llama *ltspadmin* y la herramienta de configuración se llama *ltspcfg*. Estas dos herramientas son parte del paquete *ltsp-utils*.

El paquete *ltsp-utils* está disponible tanto en formato *RPM* como *TGZ*. Tenemos que escoger el formato de acuerdo a nuestra distribución y seguir las instrucciones apropiadas.

4.3.1.1 Instalando el Paquete RPM

Descargar la versión más reciente del paquete ltsp-utils en formato RPM e instalamos usando el siguiente comando:

```
# rpm -Uhv ltsp-utils-0.25-0.noarch.rpm
```

El comando anterior instalará las utilidades en el servidor.

4.3.1.2 Instalando el Paquete TGZ

Descargar la versión más reciente del paquete ltsp-utils en formato TGZ e instalamos usando los siguientes comandos:

```
# tar xzf ltsp-utils-0.25-0.tgz
# cd ltsp_utils
# ./install.sh
# cd ..
```

Los comandos anteriores instalarán las utilidades en el servidor. Esto es muy útil para distribuciones GNU/Linux que no soporten el formato RPM.

4.3.2 Instalar los Paquetes Cliente de LTSP (LTSP Client Packages)

Una vez que la instalación del paquete ltsp-utils se haya completado, podemos ejecutar el comando ltspadmin que es una utilidad usada para administrar los paquetes cliente de LTSP (LTSP Client packages).

Ejecutamos el comando ltspadmin y nos mostrara una pantalla. Ver Figura IV.29.



Figura IV.29 Instalador LTSP – Pantalla principal

Fuente: Análisis del autor

Desde esta pantalla podemos elegir "Install/Upgrade (Instalar/Actualizar)" y si es la primera vez que ejecutamos este comando, se nos desplegará la pantalla de configuración. Ver Figura IV.30.

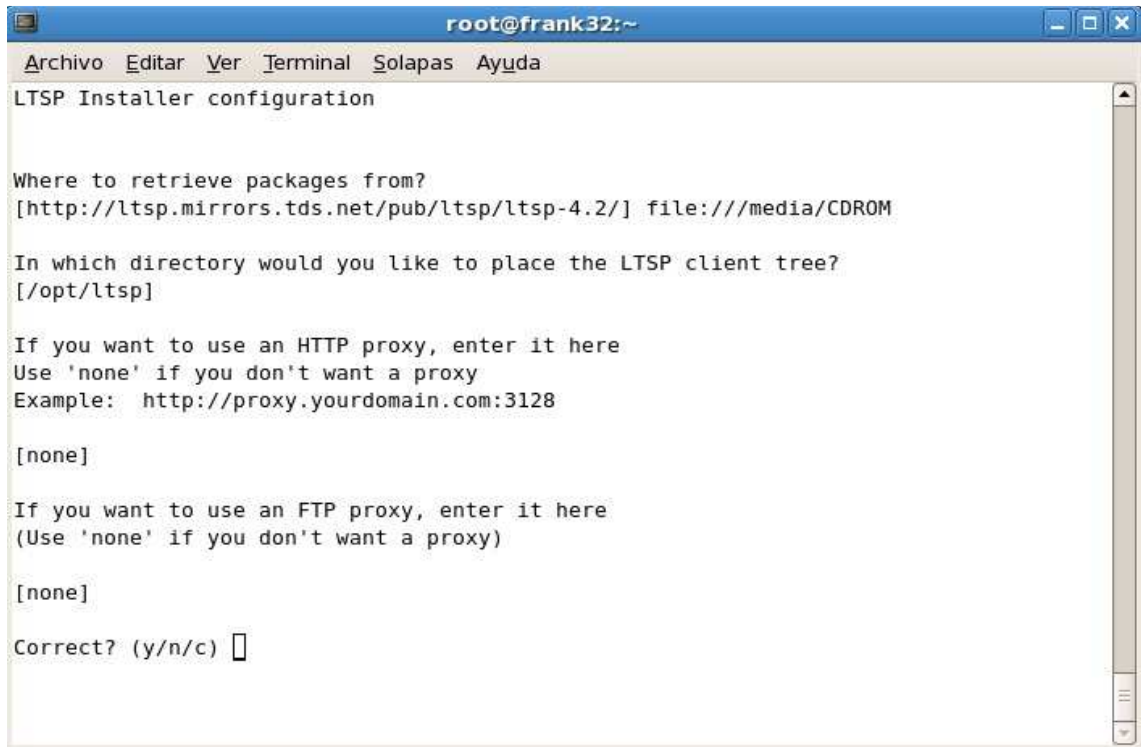


Figura IV.30 Instalador LTSP – Pantalla de Configuración

Fuente: Análisis del autor

En la pantalla de configuración se pueden asignar valores que serán utilizados por el instalador para descargar e instalar los paquetes LTSP. Estos valores son:

Desde dónde se copiarán los paquetes LTSP:

Esta ubicación tiene que estar en formato URL indicando donde se encuentra el repositorio de paquetes. Usualmente el instalador tiene la siguiente URL predeterminada que es: "http://ltsp.mirrors.tds.net/pub/ltsp/ltsp-4.2/", pero si queremos instalar los paquetes desde un sistema de archivos local tenemos que usar la sentencia "file://" en lugar de "http://". Si los paquetes están en un CD-ROM y éste está montado en "/media/CDROM", el valor que tenemos que poner es el siguiente: "file:///media/CDROM" (Notar que tiene 3 barras oblicuas).

En que directorio estará la estructura del cliente LTSP:

Este directorio estará en el servidor, en el lugar más adecuado donde penderá el árbol de directorios de los clientes LTSP. Por defecto viene configurada la siguiente ubicación: /opt/ltspl. El directorio es creado en caso de no existir.

Dentro de este directorio, los directorios raíz serán para cada arquitectura serán creados. Actualmente sólo estaciones de trabajo x86 están soportadas oficialmente por LTSP pero también se podrían trabajar en puertos de otras arquitecturas como PPC y Sparc.

HTTP Proxy:

Si el servidor está detrás de un firewall (cortafuegos) y todo acceso a la web debe ser a través de un servidor proxy, se puede configurar el programa de instalación para que use el proxy en esta sección. El valor debe contener la dirección URL del proxy incluyendo el protocolo y el puerto. Un ejemplo podría ser: *http://peietc.esepoch.edu.ec:8080*.

Si no necesitamos un servidor proxy para acceder a la web, podemos asignar el valor "none".

FTP proxy:

Esto es usado para paquetes localizados en un servidor FTP. Si necesitamos acceder a un servidor proxy FTP, en esta sección tenemos que poner la dirección. La sintaxis es similar a la opción del servidor proxy descrito en la parte de arriba.

Si no necesitamos un servidor proxy FTP, podemos asignar el valor "none".

Una vez que terminamos de asignar los valores en nuestra pantalla de configuración, el instalador hará una petición al repositorio de paquetes y obtendrá la lista de los componentes disponibles hasta la fecha. Ver Figura IV.31.

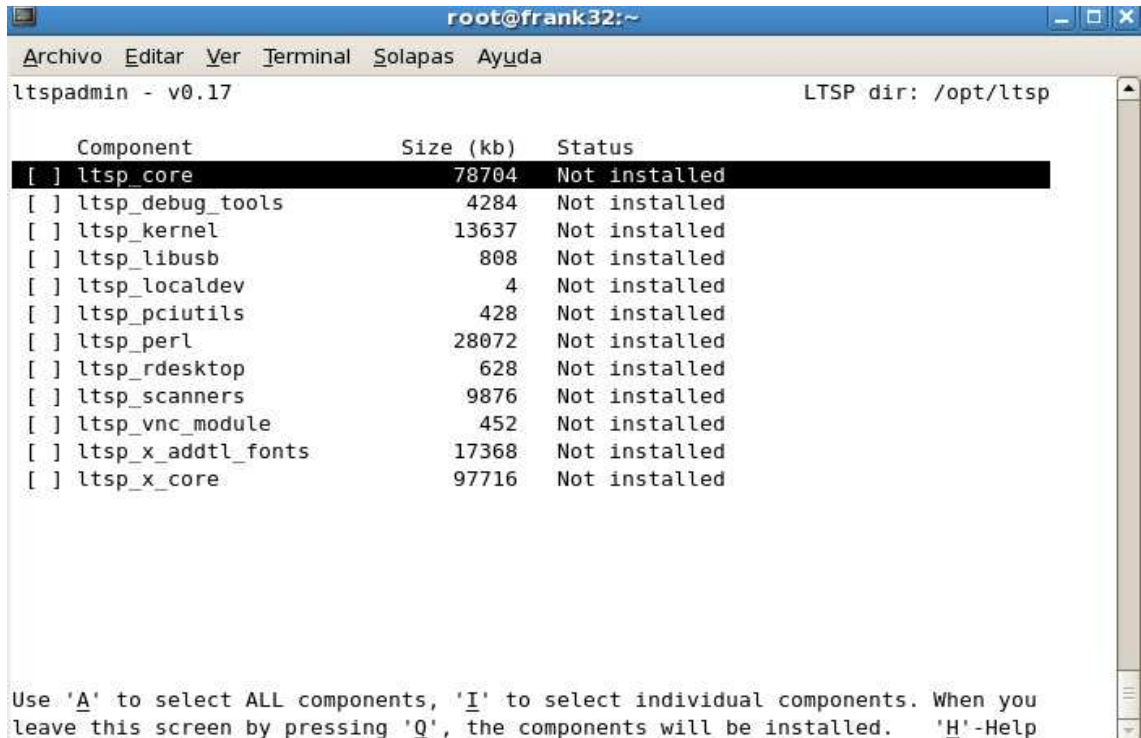


Figura IV.31 Instalador LTSP – Lista de Componentes

Fuente: Análisis del autor

Seleccionamos uno a uno los componentes que queremos instalar. Para seleccionar cada componente debemos movernos de arriba a abajo en cada ítem con fondo negro y presionar la tecla "I". Para seleccionar todos los componentes tenemos que presionar la tecla "A". Es aconsejable instalar todos los componentes para que funcione al 100% nuestro servidor LTSP.

Existen muchas teclas que nos sirven para navegar dentro de esta pantalla. Por ejemplo si queremos desplegar la ayuda presionamos la tecla "H". Ver Figura IV.32.

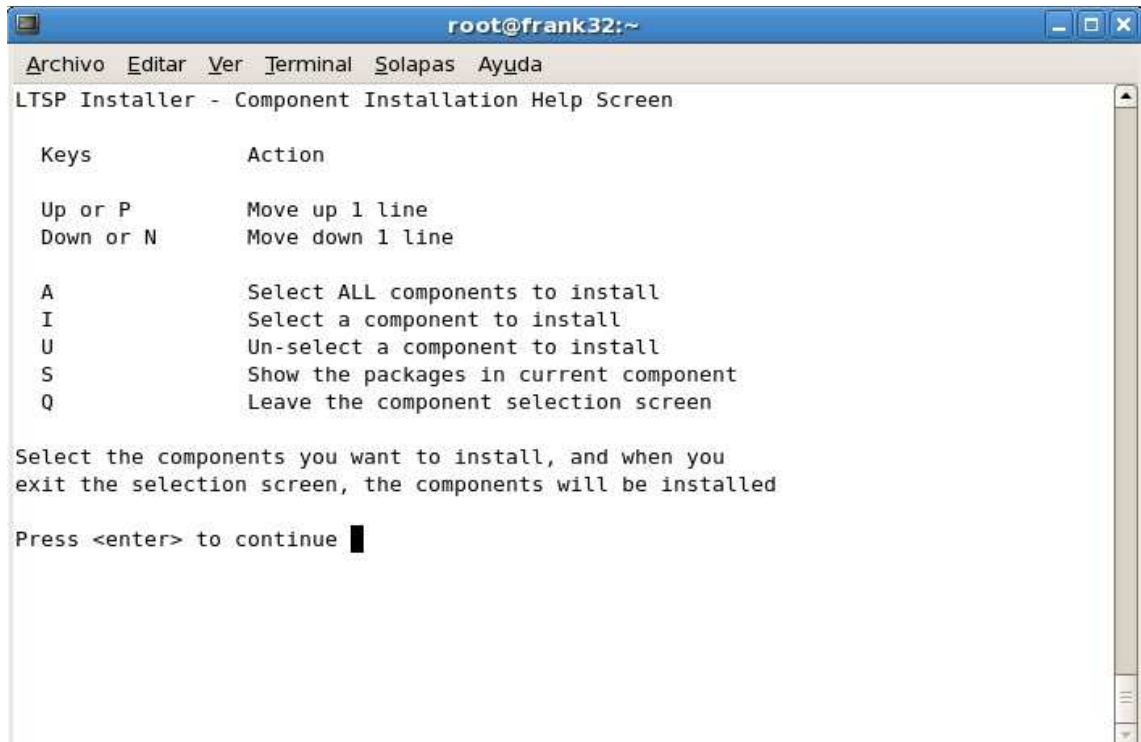


Figura IV.32 Instalador LTSP – Ventana de Ayuda

Fuente: Análisis del autor

Si queremos ver la lista de paquetes que tiene cada componente en particular, presionamos la tecla "S" y la lista de paquetes será desplegada. Al hacer esto nos mostrará la versión que está actualmente instalada y también la última versión disponible. Ver Figura IV.33.

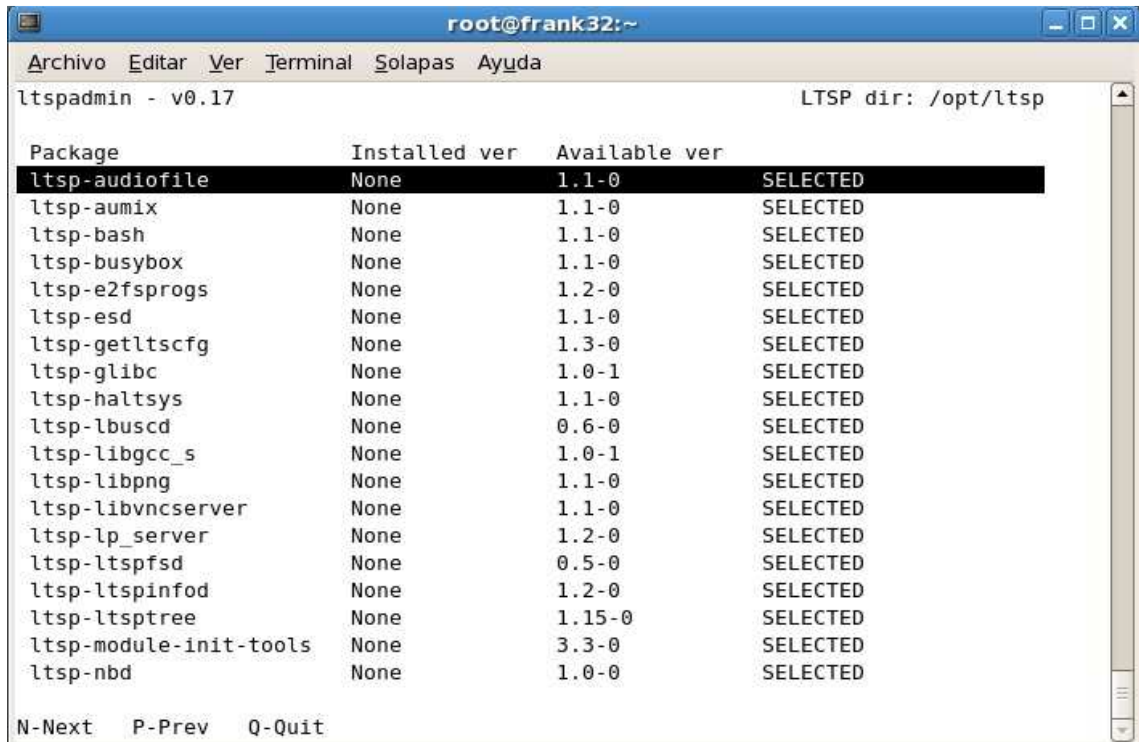


Figura IV.33 Instalador LTSP – Lista de paquetes

Fuente: Análisis del autor

Una vez que los componentes deseados se hayan instalado, podemos salir de la pantalla de selección de componentes. El instalador nos mostrará un pantalla para elegir si realmente queremos Instalar/Actualizar los paquetes seleccionados. Si la respuesta es "Y" (si), se descargarán e instalarán los paquetes seleccionados. Ver Figura IV.34

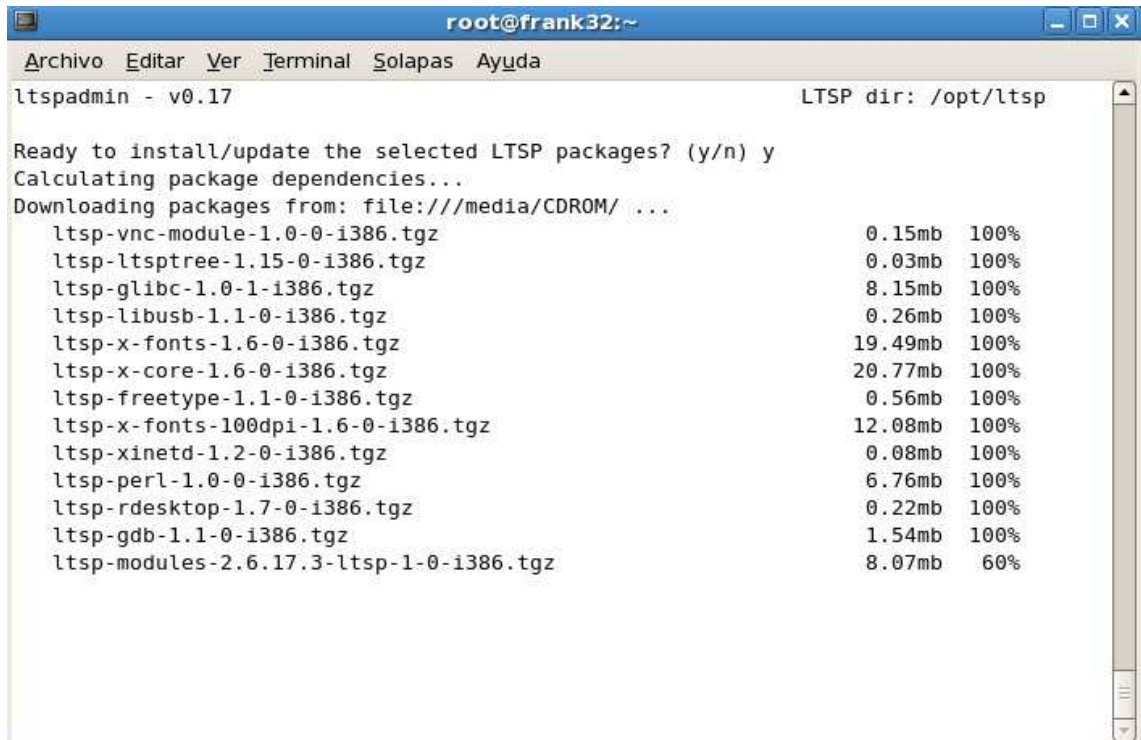


Figura IV.34 Instalador LTSP – Instalación de paquetes

Fuente: Análisis del autor

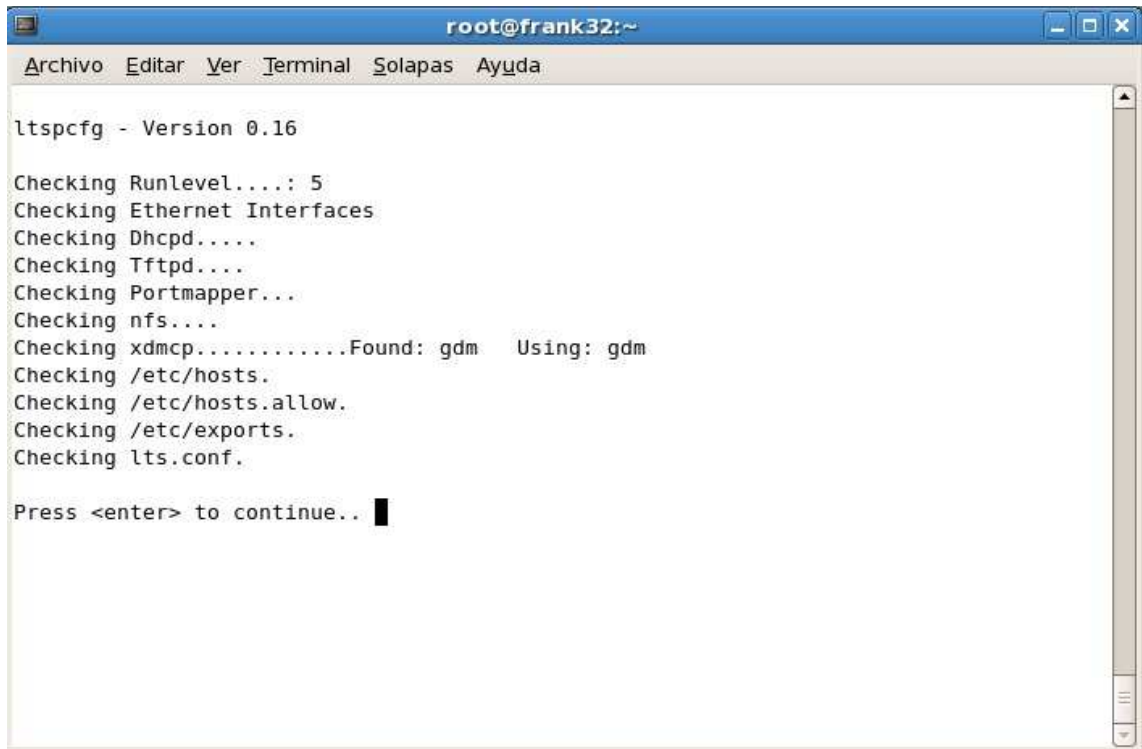
4.3.3 Configurar los Servicios Necesarios de LTSP

Existen 4 servicios básicos requeridos por una DC pueda arrancar y son:

- DHCP
- TFTP
- NFS
- XDCMP

ltspcfg puede ser utilizado para configurar todos estos servicios y muchas otras cosas relacionadas con LTSP.

Podemos acceder a `ltspcfg` desde `ltspadmin` o podemos escribir `ltspcfg` en la línea de comandos. Cuando ejecutamos la utilidad `ltspcfg`, se explorará el servidor para evaluar lo que está actualmente instalado y en ejecución. Ver Figura IV.35.



```
ltspcfg - Version 0.16

Checking Runlevel....: 5
Checking Ethernet Interfaces
Checking Dhcpd....
Checking Tftpd....
Checking Portmapper...
Checking nfs....
Checking xdmcp.....Found: gdm    Using: gdm
Checking /etc/hosts.
Checking /etc/hosts.allow.
Checking /etc/exports.
Checking lts.conf.

Press <enter> to continue.. █
```

Figura IV.35 `ltspcfg` – Pantalla Inicial

Fuente: Análisis del autor

Con esto mostramos todos los componentes que la utilidad haya buscado y encontrado.

Para configurar todos los componentes necesarios, presionamos la tecla “C” y el menú de configuración será desplegado. En el menú de configuración tenemos que configurar cada componente de uno en uno para asegurarnos que el servidor tiene los valores de configuración apropiados para servir a las estaciones de trabajo. Ver Figura IV.36.

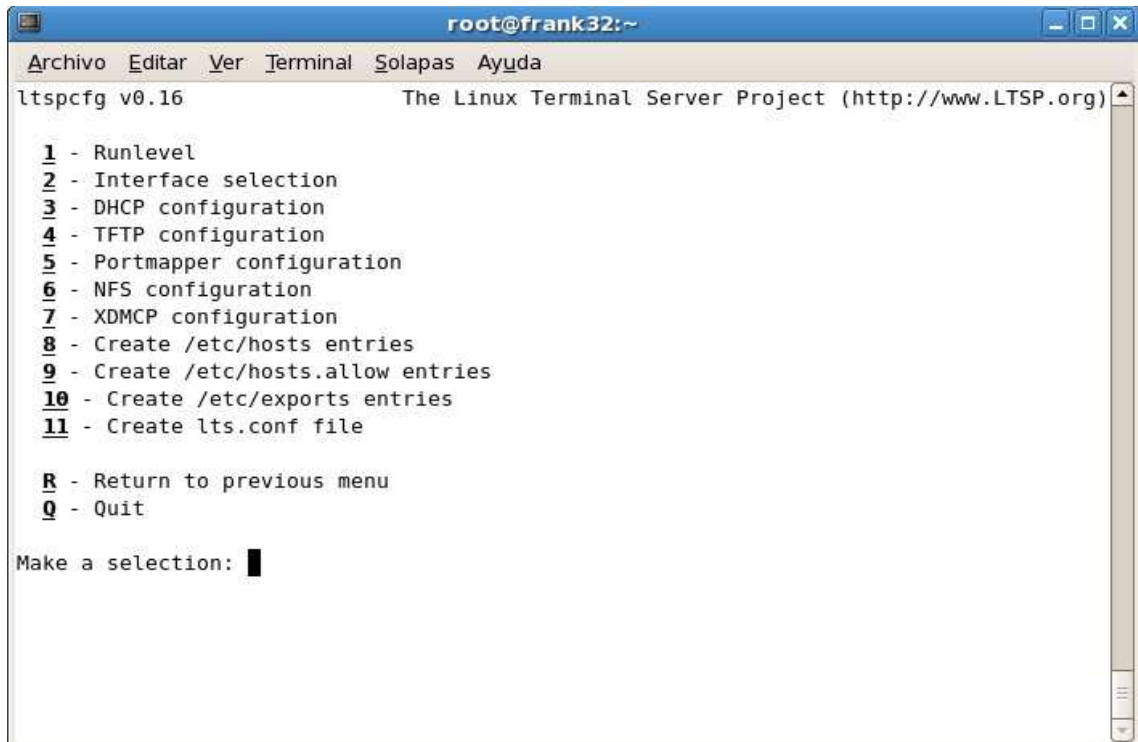


Figura IV.36 ltspcfg – Pantalla inicial para configurar

Fuente: Análisis del autor

1. Runlevel

La variable *Runlevel* es usada por el programa *init*. Con sistemas GNU/Linux y Unix, en cualquier momento se les puede asignar un Runlevel específico. El Runlevel 2 ó 3 es usado comúnmente cuando el servidor está en modo de texto. El Runlevel 5 generalmente es usado para indicar que el sistema está en modo gráfico en una red.

En un servidor LTSP se usa por defecto el Runlevel 5. La mayoría de los sistemas ya vienen configurados para soportar *NFS* y *XDMCP* cuando se ejecutan en un Runlevel 5. Para aquellos sistemas que no traen soporte para la configuración mencionada, LTSP se encargará de darles soporte.

2. Interface selection

Para sistemas que tienen múltiples interfaces de red, necesitaremos especificar en qué interface se conectarán las estaciones de trabajo.

Para especificar la interface, la herramienta de configuración nos permitirá elegir la interface que usará LTSP para conectar las estaciones de trabajo y también nos permitirá crear los archivos de configuración restantes como `dhcpd.conf`, `/etc/exports`, etc.

3. DHCP configuration

DHCP necesita ser configurado para proporcionar las peticiones del cliente. Entre estas peticiones están: kernel Linux, dirección IP, máscara de red, dirección de broadcast, y directorio raíz.

Al seleccionar este ítem del menú, se creará el archivo de configuración de DHCP llamado `dhcpd.conf`. DHCP debe habilitarse para que se ejecute cada vez que se inicie el servidor.

4. TFTP configuration

TFTP es usado por las DCs para descargarse el kernel Linux. El servicio `tftpd` tiene que habilitarse para que se ejecute en el servidor y pueda entregar el kernel.

5. Portmapper configuration

Portmapper es usado por los servicios *RPC*. Cada servicio *RPC* es como un servicio NFS.

6. NFS configuration

NFS es el servicio que permite que el árbol de directorio local sea montado desde máquinas remotas. Esto es requerido por LTSP debido a que los clientes montan su sistema de archivos raíz desde el servidor.

Este ítem del menú configura el servicio NFS para que se ejecute cada vez que se inicia el servidor. El archivo de configuración es `"/etc/exports"` y su creación será descrita más adelante.

7. XDMCP configuration

XDMCP "Protocolo de control de administración de visualización X (**X** Display **M**anager **C**ontrol **P**rotocol)". El servidor X envía una petición XDMCP al administrador de visualización del servidor LTSP para recibir una pantalla de inicio de sesión, el cual nos pide un usuario y un password.

Los administradores de visualización más comunes son *XDM*, *GDM* y *KDM*. Este ítem del menú nos indicará que administradores de pantalla se han encontrado en el servidor y nos pedirá configurar uno para que se ejecute por defecto.

Por propósitos de seguridad, el administrador de visualización está configurado por defecto para no permitir que los clientes remotos se conecten. Este es el motivo por el cual siempre se ve una infame pantalla gris y un cursor "X" grande. *Itspcfg* puede configurar el administrador de visualización para permitir que los clientes remotos se conecten y obtengan una pantalla de inicio de sesión.

8. Create /etc/hosts entries

Muchos servicios como NFS y el administrador de visualización tienen que mapear o rastrear la dirección IP y el nombre del host cliente. Para añadir los host cliente también se puede instalar Berkeley Internet Naming Daemon (BIND) que es una implementación del protocolo DNS (Domain Name Service, Sistema de Nombres de Dominio), asegurándonos de tener los archivos de resolución inversa configurados correctamente. El uso de BIND es probablemente la mejor manera de asignar nombres a los host cliente pero no se hablara de ello debido a que nos alejamos de nuestro propósito.

Una manera más simple de configurar el mapeo de direcciones IP y nombres de los host clientes, lo podemos hacer mediante el archivo */etc/hosts*.

9. Create */etc/hosts.allow* entries

Algunos servicios usan una capa de seguridad conocida como *tcpwrappers*, esto se puede realizar configurando el archivo */etc/hosts.allow*. Este ítem del menú lo podemos configurar nosotros mismos.

10. Create */etc/exports* entries

Este es el archivo que NFS usa para determinar qué directorios podrán ser montados por las máquinas remotas. Este ítem del menú creará dicho archivo.

11. Create *lts.conf* file

La configuración de cada estación de trabajo está dirigida por las entradas del archivo *lts.conf*. Para estaciones de trabajo que tengan al menos un puerto PCI, podría no requerir ninguna entrada adicional en el archivo *lts.conf*. Este ítem del menú nos creará un archivo *lts.conf* por defecto para nuestro servidor.

4.3.4 Configuración Específica para los Clientes

Ahora es el momento de decirle a nuestro servidor LTSP todos los detalles de las estaciones de trabajo. Hay tres archivos que contienen información acerca de las estaciones de trabajo, que son:

- /etc/dhcpd.conf
- /etc/hosts
- /opt/ltsp/i386/etc/lts.conf

4.3.4.1 /etc/dhcpd.conf

La estación de trabajo necesita una dirección IP y otra información que se puede obtener mediante el servidor DHCP, esta información es:

- Dirección IP.
- Nombre del host cliente.
- Dirección IP del servidor.
- Pasarela (Gateway) predeterminada.
- Ruta o ubicación del núcleo a cargar.
- Servidor y ubicación del directorio a ser montado como sistema de archivos raíz.

Para nuestro ambiente LTSP de ejemplo hemos escogido DHCP para controlar la asignación de direcciones IP de las estaciones de trabajo.

Durante el script *ltsp_initialize*, un archivo *dhcpd.conf* de muestra es instalado, llamado */etc/dhcpd.conf.example*. Podemos copiar el mismo en */etc/dhcpd.conf* para usarlo como base para nuestra configuración de DHCP.

Probablemente necesitaremos modificar las partes correspondientes a la estación de trabajo específica y a un ambiente en particular. Ver Figura IV.37.

```
ddns-update-style          none;

default-lease-time 21600;
max-lease-time 21600;

option subnet-mask         255.255.255.0;
option broadcast-address   192.168.0.255;
option routers             192.168.0.2;
option domain-name-servers 192.168.0.2;
option domain-name        "ltsp_frank32.org"; # You really should fix this
option option-128 code 128 = string;
option option-129 code 129 = text;

next-server                192.168.0.2;
option root-path           "192.168.0.2:/opt/ltsp/i386";

subnet 192.168.0.0 netmask 255.255.255.0 {
    use-host-decl-names on;
    option log-servers 192.168.0.2;

    host ws003 {
        hardware ethernet 00:E0:4C:0A:6E:40;
        fixed-address 192.168.0.3;
        filename "/lts/vmlinuz-2.6.17.3-ltsp-1";
    }

    host ws004 {
        hardware ethernet 00:E0:4C:13:53:87;
        fixed-address 192.168.0.4;
        filename "/lts/vmlinuz-2.6.17.3-ltsp-1";
    }
}
```

Figura IV.37 Archivo /etc/dhcpd.conf

Fuente: Análisis del autor

Como en LTSP versión 2.09pre2, no se necesita especificar un kernel en particular para cargar. El paquete estándar de núcleo soporta todas las placas de red que GNU/Linux soporta. Hay dos archivos de kernel incluidos en el paquete LTSP-kernel. Uno de los archivos tiene el Linux Progress Patch (parche que muestra una barra de progreso gráfica) y el otro no. Los nombres de estos núcleos son:

- vmlinuz-2.6.17.3-ltsp-1
- vmlinuz-2.6.17.3-ltsp-lpp-1

Podemos notar que el kernel está ubicado en el directorio `/tftpboot/lts`, pero en la entrada "filename" del archivo `/etc/dhcpd.conf` falta la ruta `/tftpboot` al inicio. Esto es así debido a que en versiones de RedHat superiores a 7.1, TFTP corre con la opción '-s', lo que causa que el demonio `tftpd` corra en modo seguro, Esto significa que realiza un `chroot` al directorio `/tftpboot` cuando arranca. Por lo tanto, todos los archivos que están disponibles para el demonio `tftpd` son relativos al directorio `/tftpboot`.

4.3.4.2 /etc/hosts

Mapeo de direcciones IP a nombres de host. Las computadoras generalmente se comunican mediante su dirección IP. Como humanos necesitamos ponerle nombres a las computadoras porque se nos hace más fácil recordar un nombre que una dirección IP. Es en este momento es donde entran en escena DNS o `/etc/hosts` para asignar direcciones IP y nombres a las estaciones de trabajo. Las direcciones IP asignadas por lo general no son requeridas, excepto en los ambientes LTSP porque sin la dirección IP, NFS nos mostrará mensajes de error de permisos cuando las estaciones de trabajo intenten montar el sistema de ficheros raíz.

Además de los problemas de NFS, si las estaciones de trabajo no están listadas en el archivo `/etc/hosts` también podríamos tener problemas con los administradores de visualización GDM o KDM.

4.3.4.3 /opt/ltsp/i386/etc/lts.conf

Hay un sinnúmero de entradas de configuración que pueden ser especificadas en el archivo `lts.conf`.

El archivo `lts.conf` tiene una sintaxis simple, que consta de múltiples secciones. Hay una sección por defecto llamada **[default]** y pueden haber secciones individuales para cada cliente LTSP cambiando el nombre `default` por el nombre del cliente. Los clientes son identificados por el nombre del host, la dirección IP o la dirección MAC. Se mostrará un archivo típico `lts.conf`. Ver Figura IV.38.

```
-----  
#  
# Config file for the Linux Terminal Server Project (www.ltsp.org)  
#  
[ws003]  
SERVER          = 192.168.0.2  
XSERVER         = auto  
X_MOUSE_PROTOCOL = "PS/2"  
X_MOUSE_DEVICE  = "/dev/psaux"  
X_MOUSE_RESOLUTION = 400  
X_MOUSE_BUTTONS = 3  
USE_XFS         = N  
SCREEN_01       = startx  
X_MODE_0        = 800x600  
X_COLOR_DEPTH   = 24  
XkbLayout       = es  
  
[ws004]  
SERVER          = 192.168.0.2  
XSERVER         = auto  
X_MOUSE_PROTOCOL = "PS/2"  
X_MOUSE_DEVICE  = "/dev/psaux"  
X_MOUSE_RESOLUTION = 400  
X_MOUSE_BUTTONS = 3  
USE_XFS         = N  
SCREEN_01       = startx  
X_MODE_0        = 800x600  
X_COLOR_DEPTH   = 24  
XkbLayout       = es
```

Figura IV.38 Archivo `lts.conf`

Fuente: Análisis del autor

A continuación se detallan las entradas más importantes del archivo `lts.conf`:

XSERVER

Si la tarjeta de video está conectada a la ranura PCI y está soportada por *Xorg*, sólo necesitamos el paquete `lts_x_core`. Que contiene todos los controladores para X4.

Hay algunos paquetes de *XFree86* disponibles para LTSP en caso de que alguna tarjeta de video no sea soportada por *Xorg*.

Se pueden crear configuraciones individuales para cada cliente en el archivo *lts.conf* o se pueden crear configuraciones que pueden ser compartidas por los clientes LTSP.

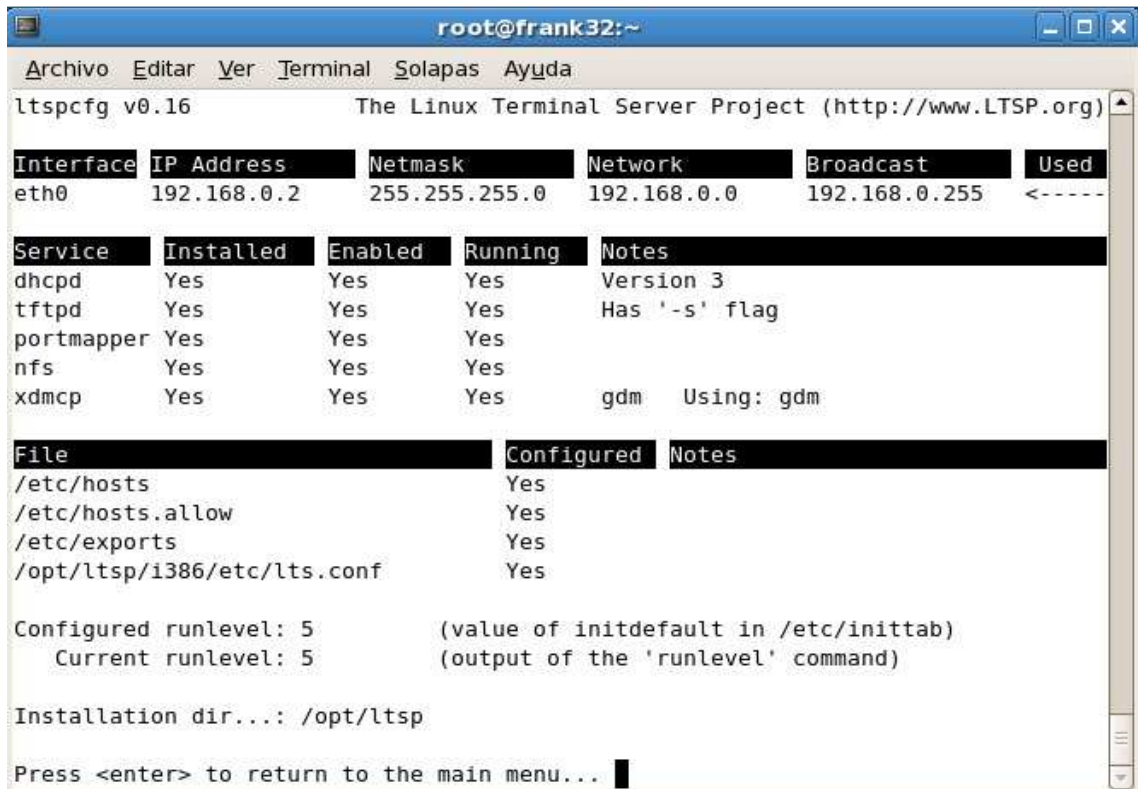
Los clientes usados en esta tesis tienen chip de video incluido que se auto detectan correctamente, así que no necesitamos ninguna configuración especial en la entrada XSERVER. Se puede escribir la sentencia "auto" en la entrada XSERVER para las tarjetas más comunes como en nuestro caso.

RUNLEVEL

Cuando queremos ejecutar las estaciones de trabajo en modo gráfico, tenemos que poner el Runlevel en "5" en nuestro archivo de configuración *lts.conf*, aunque en la versión 4.2update2 de LTSP no hace falta especificar el Runlevel porque automáticamente lo pone a "5".

4.3.5 Desplegando las Configuraciones Concurrentes

Con ltspcfg se puede desplegar el estado actual de todos los servicios requeridos por LTSP. Desde el menú ltspcfg presionamos la tecla "S" y podremos ver el estado de los servicios. Ver Figura IV.39.



```
root@frank32:~
Archivo Editar Ver Terminal Solapas Ayuda
ltspcfg v0.16 The Linux Terminal Server Project (http://www.LTSP.org)

Interface IP Address Netmask Network Broadcast Used
eth0 192.168.0.2 255.255.255.0 192.168.0.0 192.168.0.255 <-----

Service Installed Enabled Running Notes
dhcpd Yes Yes Yes Version 3
tftpd Yes Yes Yes Has '-s' flag
portmapper Yes Yes Yes
nfs Yes Yes Yes
xdmcp Yes Yes Yes gdm Using: gdm

File Configured Notes
/etc/hosts Yes
/etc/hosts.allow Yes
/etc/exports Yes
/opt/ltsp/i386/etc/lts.conf Yes

Configured runlevel: 5 (value of initdefault in /etc/inittab)
Current runlevel: 5 (output of the 'runlevel' command)

Installation dir...: /opt/ltsp

Press <enter> to return to the main menu... █
```

Figura IV.39 ltspcfg – Estado actual de los servicios

Fuente: Análisis del autor

4.4 CONFIGURANDO LOS CLIENTES LTSP

Una vez que el servidor ha sido configurado, es tiempo de concentrarnos en las estaciones de trabajo.

El proyecto LTSP trata acerca de qué es lo que sucede cuando el kernel está en memoria. Hay varias maneras de hacer que un kernel se cargue en memoria, incluyendo Etherboot, Netboot, PXE y disco floppy.

Para el desarrollo de esta tesis, usaremos un disquete con código del proyecto Etherboot para los computadores más viejitos y también arranque PXE para los computadores que tengan soporte de arranque desde la LAN.

4.4.1 Creando el Disquete de Inicio

Etherboot es un paquete de software para la creación de imágenes ROM que puedan descargar código a través de una red Ethernet y luego ejecutarlo en un computador x86. Muchos dispositivos de red poseen un sócalo donde un chip ROM puede ser instalado. Etherboot es el código que puede ser colocado en dicha ROM.

Podemos descargar el paquete Etherboot y configurarlo para el tipo de bootrom que necesitemos. Luego, podemos compilar el código fuente para producir la imagen que será escrita en el chip EPROM, o bien en un diskette para pruebas. Una alternativa mucho más simple es ir al sitio web de Marty Connor: www.Rom-O-Matic.net.

Marty ha realizado un excelente trabajo al ponerle un frente basado en web al proceso de configuración y compilación de las imágenes de bootrom con Etherboot.

En su sitio, podemos elegir qué tipo de tarjeta de red tenemos, y qué clase de imagen queremos. Luego, tenemos la oportunidad de modificar cualquiera de las opciones de configuración de Etherboot. Por último, presionando el botón 'Get ROM' y una imagen bootrom personalizada será generada para nuestra tarjeta de red.

Una vez que se ha guardado la imagen en el disco duro, necesitamos grabarla en un disquete. Insertamos un disquete y ejecutamos el siguiente comando:

```
# dd if=Imagen_Etherboot of=/dev/fd0
```

4.5 EJECUTANDO NUESTRA DC

Asumiendo que el servidor y la estación de trabajo han sido configurados correctamente, debería ser sólo una cuestión de insertar el disquete en la disquetera y encender la estación de trabajo.

El código Etherboot será leído desde el disquete y cargado en memoria, la tarjeta de red será encontrada e inicializada, la petición DHCP será enviada por la red y una respuesta será enviada desde el servidor y el kernel será descargado en la estación de trabajo. Una vez que el kernel ha inicializado el hardware de la estación de trabajo, X Window arrancará y una ventana de diálogo de inicio debería aparecer en la terminal. Ver Figura IV.40.

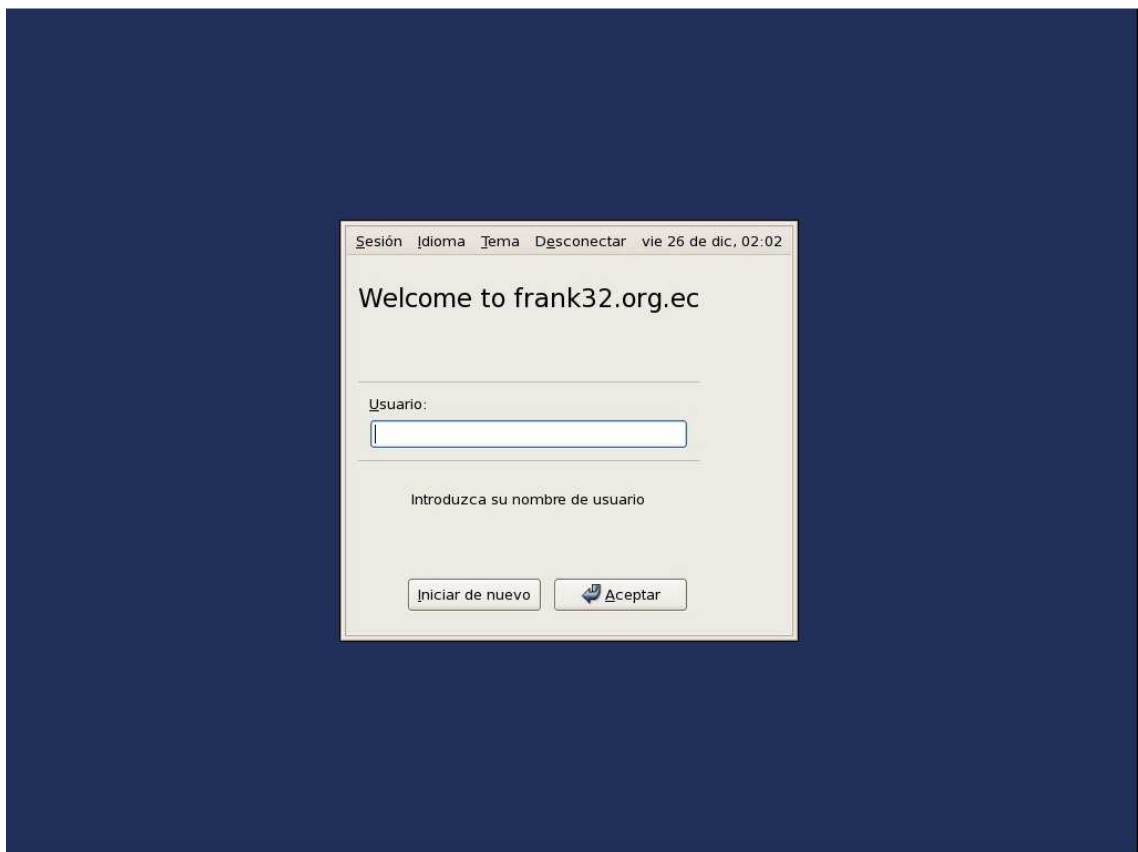


Figura IV.40 Pantalla del Terminal

4.6 SCREEN SCRIPTS

Una característica de LTSP que fue añadida desde la versión 4.0 es la posibilidad de poder ejecutar scripts para arrancar varios tipos de sesiones en la pantalla de nuestros DCs.

Podemos especificar múltiples screen scripts para una DC y obtener múltiples sesiones. Pueden ser diferentes tipos de sesiones o pueden ser el mismo tipo de sesiones. Por ejemplo podríamos especificar lo siguiente:

```
SCREEN_01 = startx
```

```
SCREEN_02 = shell
```

Esto permitirá arrancar el servidor X en la primera pantalla y un prompt Shell en la segunda pantalla. Podemos cambiarnos entre las pantallas presionando Ctrl–Alt–F1 para acceder a la primera pantalla y Ctrl–Alt–F2 para acceder a la segunda pantalla.

Podemos especificar hasta 12 pantallas para una DC, pero por lo general siempre se va a trabajar con una pantalla.

Los tipos de screen scripts disponibles son:

startx

Este script permite ejecutar el servidor X con una –opción de consulta, que envía una petición XDMCP al display manager para obtener una pantalla de registro que nos pedirá login y password en la pantalla.

shell

Este script ejecuta un prompt Shell en una DC. Este script es generalmente usado para reparar posibles problemas con nuestra DC. No es recomendado usar este script porque no podremos ejecutar aplicaciones en nuestro cliente.

telnet

Con este script podemos iniciar una sesión telnet que se conectará al servidor y proporcionará una pantalla basada en caracteres. Por defecto, telnet se conectará al servidor LTSP pero existe la posibilidad de especificar otro servidor diferente. Por ejemplo:

```
SCREEN_01 = telnet server2.midominio.com
```

rdesktop

Con este script podemos iniciar el programa rdesktop, el cual se conectará a un servidor de ventanas de Microsoft. Podemos especificar cualquier opción rdesktop que queramos después del nombre del screen script. Por ejemplo, si queremos especificar el servidor al que nos conectaremos, podemos hacer lo siguiente:

```
SCREEN_01 = rdesktop -f w2k3.midominio.com
```

El ejemplo anterior iniciará rdesktop en modo de pantalla completa. El usuario verá una ventana de registro y podrá entrar al sistema sólo una vez. Esto es muy útil cuando sólo se quiere conseguir una ventana de registro sin tener un tipo de registro como el de GNU/Linux. El usuario no sabrá ni siquiera que están trabajando sobre GNU/Linux. En el Anexo 1 se describe la instalación de CentOS 5 que fue la plataforma ganadora.

CONCLUSIONES

- 1.** El presente Estudio Comparativo permitió determinar como la mejor plataforma, para el diseño e implementación de un laboratorio de software y redes mediante el uso de un servidor de terminales, a GNU/Linux.
- 2.** Se debe tener en cuenta cuáles son nuestras necesidades al momento de elegir una plataforma para nuestro servidor de terminales debido a que las principales características serán el costo de su implementación y el beneficio que obtendremos.
- 3.** Windows Server 2003 trae muchas herramientas de configuración para nuestro servidor de manera visual pero es atacado por hackers con mucha frecuencia y la mayoría de los virus que se esparcen en internet, están destinados a perjudicar la integridad de ésta plataforma.
- 4.** Debido a que Windows Server 2003 no es Open Source, no existen proyectos derivados del mismo y por lo tanto no se puede configurar ningún servidor de terminales. En GNU/Linux encontramos muchos proyectos como LTSP, cuya finalidad es la reutilización de computadores que son cambiados cada dos años y producen basura tecnológica.

5. GNU/Linux ofrece su código fuente al momento de su adquisición por lo tanto si alguien desea aportar algún proyecto o corregir algún error, lo puede hacer sin necesidad de tener que comprar ninguna licencia. Lo contrario pasa con Windows Server 2003 porque aunque compremos su licencia, no podremos tener acceso a su código fuente.
6. Con Windows Server 2003 tendremos que instalar obligatoriamente un antivirus tanto en el servidor como en los clientes. Mientras que con GNU/Linux se instalará un antivirus sí y solo sí nuestro servidor tiene conectados clientes con Windows XP o cualquier otra versión, para centralizar la búsqueda de virus.
7. Windows Server 2003 sólo puede montar particiones con sistemas de archivos de Microsoft así que si queremos montar particiones que no sean de Microsoft, tendremos que instalar programas adicionales mientras que con GNU/Linux no es necesario instalar programas adicionales ya que tiene soporte para muchos tipos de sistemas de archivos.
8. El presente estudio determina un porcentaje final de **48.04%** (Bueno) y **91.18%** (Excelente) para Windows Server 2003 y GNU/Linux respectivamente, ya que éste último se muestra superior a Windows Server 2003 en casi todos los aspectos considerados para este estudio.

RECOMENDACIONES

- 1.** Revisar detenidamente el presente Estudio Comparativo y tener en cuenta que se han comparado aspectos referentes a redes, antes de determinar a una de las plataformas como la más idónea para el diseño e implementación de un servidor de terminales.
- 2.** Considerar la implementación de GNU/Linux para nuestro servidor de terminales ya que nos brinda muchas características como seguridad y reducción al máximo de costos tanto en licencias como en la compra de computadoras.
- 3.** Tratar de instalar programas necesarios dependiendo del uso que se le vaya a dar a nuestro laboratorio de software y redes, ya que nuestro servidor almacenará la información de los usuarios y toda la información de los programas instalados en su disco duro.
- 4.** Poner en práctica la configuración e instalación de la plataforma ganadora tanto en escuelas, colegios y universidades para ir aprendiendo el uso de programas que sean software libre, y porque el uso de GNU/Linux implica ahorro económico frente a Windows Server 2003.

- 5.** Estudiar la posibilidad de migrar los servidores que tengan Windows Server 2003 a GNU/Linux, ya que para todos los servidores que tiene Windows 2003 tienen su equivalente libre en GNU/Linux y en ocasiones han demostrado ser más estables y seguros.
- 6.** Continuar incrementando funcionalidad a nuestro servidor de terminales ya que con una buena configuración y administración podríamos hacer que los clientes puedan acceder a Internet y compartir recursos entre ellos.
- 7.** Difundir el uso de herramientas de software libre ya que nuestro país no se puede dar el lujo de comprar licencias para todos los servidores que necesita implementar, reduciendo enormemente recursos económicos, obteniendo los mismos beneficios que nos brinda el software privativo.

RESUMEN

El objetivo de esta tesis es seleccionar, mediante un Estudio Comparativo entre Windows Server 2003 y GNU/Linux, la plataforma más adecuada para el diseño e implementación de un laboratorio de software y redes para la Escuela de Ingeniería Electrónica y Tecnología en Computación de la Escuela Superior Politécnica de Chimborazo.

La investigación se basó en el Método Científico General, se utilizó como herramienta para la instalación de cada plataforma, el programa VMWare 5.5.9 con el que se puso en ejecución GNU/Linux y Windows Server 2003, y se evaluó cada uno de los servicios de red que posee.

Se obtuvo un porcentaje final del **91.18%** para GNU/Linux y **48.04%** para Windows Server 2003, esto determinó como la plataforma más adecuada a GNU/Linux para la instalación y configuración de un servidor de terminales, denominándolo Linux Terminal Server Project (LTSP). GNU/Linux es **100%** más seguro y **95%** más estable que Windows Server 2003, convirtiéndolo en la mejor opción a ser tomada al momento de instalar un servidor dentro de una red de computadoras.

Se recomienda el uso de un servidor con GNU/Linux debido a su alto desempeño mostrado en las pruebas realizadas, permitiendo diseñar e implementar un laboratorio de software y redes con un costo muy reducido reutilizando computadores antiguos, a la vez que evitamos crear basura tecnológica y centralizamos la instalación de programas y la información para almacenarla sólo en nuestro servidor.

SUMMARY

The objective of this thesis is to select, by means of a Comparative Study between Windows Server 2003 and GNU/Linux, the platform more adapted for the design and implementation of a software laboratory and networks for the Electronic Engineering and Technology in Computation School of the Escuela Superior Politecnica de Chimborazo.

The investigation was based on General the Scientific Method, was used like tool for the installation of each platform, the program VMWare 5.5.9 with which it was put in execution GNU/Linux and Windows Server 2003, and evaluated each one of the services of network that it has.

One obtained a final percentage of the **91,18%** for GNU/Linux and **48,04%** for Windows Server 2003, this determined like the most suitable platform to GNU/Linux for the installation and configuration of a terminal server, denominating it Linux Terminal Server Project (LTSP). GNU/Linux is **100 %** more safe and **95 %** more stable than Windows Server 2003, turning it into the best choice to be taken at the moment of installing a server inside a computer network.

The use of a servant with GNU/Linux due to its high performance shown in the made tests is recommended, allowing to design and to implement a laboratory of software and networks with a cost very reduced being reused old computers, simultaneously that we avoided to create technological sweepings and we centralized the installation of programs and the information to store it only in our servant.

GLOSARIO

API	Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación.
Batch	Se dice que un proceso es batch cuando se realiza de forma secuencial y automática por el computador.
BIOS	Pequeño programa que coordina las actividades de los distintos componentes de un ordenador y comprueba su estado.
DHCP	Protocolo que permite a un dispositivo de una red, conocido como servidor DHCP, asignar direcciones IP temporales a otros dispositivos de red, normalmente equipos.
DNS	Servidor automatizado utilizado en el internet cuya tarea es convertir nombres fáciles de entender (como www.ejemplo.com) a direcciones numéricas de IP.
FTP	Protocolo que nos permite transferir archivos en internet, a demás de copiarlos y poder distinguir entre dos tipos de archivos que pueden ser binarios o ASCII.
GUI	Sistema de interacción entre el ordenador y el usuario, caracterizado por la utilización de íconos y elementos gráficos en su concepción.

- Hacker** Persona que tiene un conocimiento profundo acerca del funcionamiento de redes de forma que puede advertir los errores y fallas de seguridad del mismo.
- Integridad** Se refiere a las medidas de salvaguarda que se incluyen en un sistema de información para evitar la pérdida accidental de los datos.
- Interface** Dispositivo que permite la conexión de dos elementos entre sí, de manera que se pueda producir un intercambio de información entre ambos.
- Memoria RAM** Contiene los programas y los datos con los que el computador está trabajando en un momento determinado.
- Memoria ROM** Contiene programas que son piezas fundamentales del sistema y que no pueden ser borrados ni por el usuario ni por la propia máquina.
- Plataforma** Es un término de carácter genérico que designa normalmente una arquitectura de hardware, aunque también se usa a veces para sistemas operativos o para el conjunto de ambos.
- Protocolo** Se denomina protocolo a un conjunto de normas y/o procedimientos para la transmisión de datos que han de ser observados por los dos extremos de un proceso comunicacional (emisor y receptor).

- Repositorio** Podría definirse como la base de datos fundamental para el diseño; no sólo guarda datos, sino también algoritmos de diseño y, en general, elementos software necesarios para el trabajo de programación.
- Shell** Es un tipo de utilidad cuya finalidad consiste en hacer más fácil el manejo del sistema operativo o de una aplicación por parte del usuario.
- X Window** Es un software desarrollado por el Instituto de Tecnología de Massachusetts (conocido como MIT) que se ha convertido en un estándar para Unix. Básicamente, ofrece la posibilidad de visualizar simultáneamente en pantalla varios procesos en ejecución.

ANEXOS

ANEXO 1

INSTALACIÓN EN MODO GRÁFICO DE CENTOS 5

Inserte el disco DVD de instalación de CentOS 5 y en cuanto aparezca el diálogo de inicio (boot), pulse la tecla ENTER o bien ingrese las opciones de instalación deseadas.



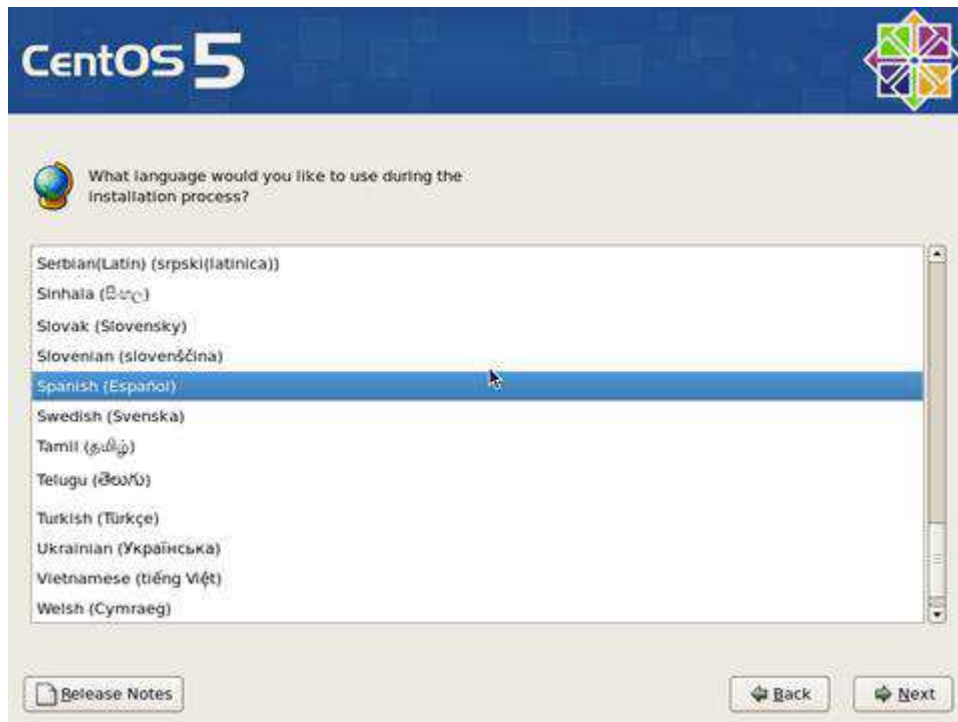
Si desea verificar la integridad del disco a partir del cual se realizará la instalación, seleccione «**OK**» y pulse la tecla ENTER, considere que esto puede demorar varios minutos. Si está seguro de que el disco o discos a partir de los cuales se realizará la instalación están en buen estado, seleccione «**Skip**» y pulse la tecla ENTER.



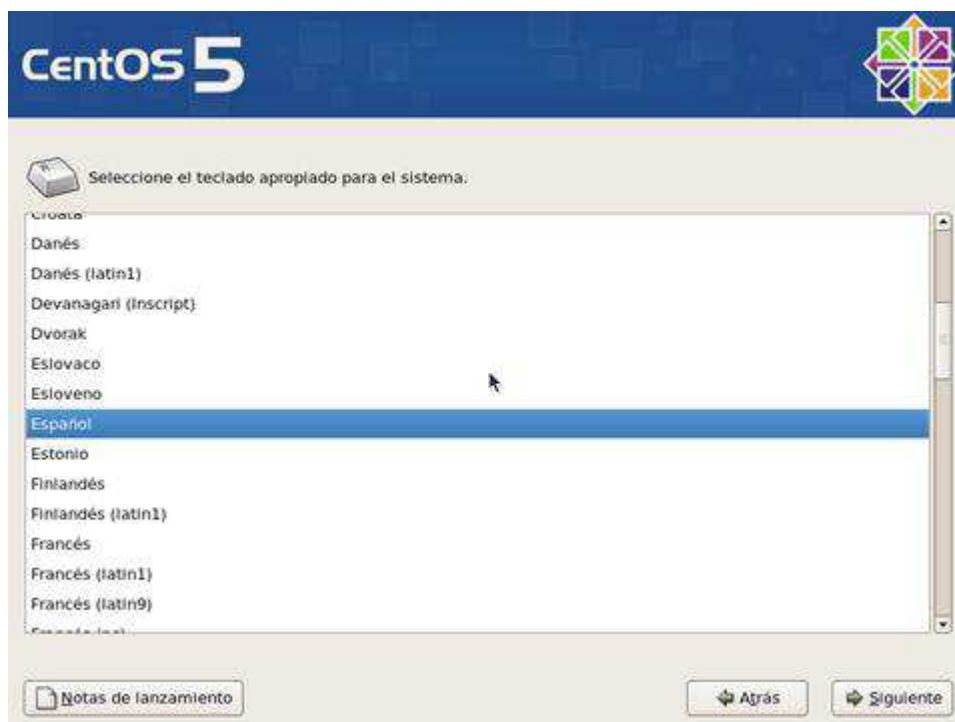
Haga clic sobre el botón «Next» en cuanto aparezca la pantalla de bienvenida de CentOS.



Seleccione «**Spanish**» como idioma para ser utilizado durante la instalación.



Seleccione el mapa de teclado que corresponda al dispositivo utilizado. El mapa «**Español**» o bien «**Latinoamericano**» de acuerdo a lo que corresponda. Al terminar, haga clic sobre el botón «**Siguiente**».

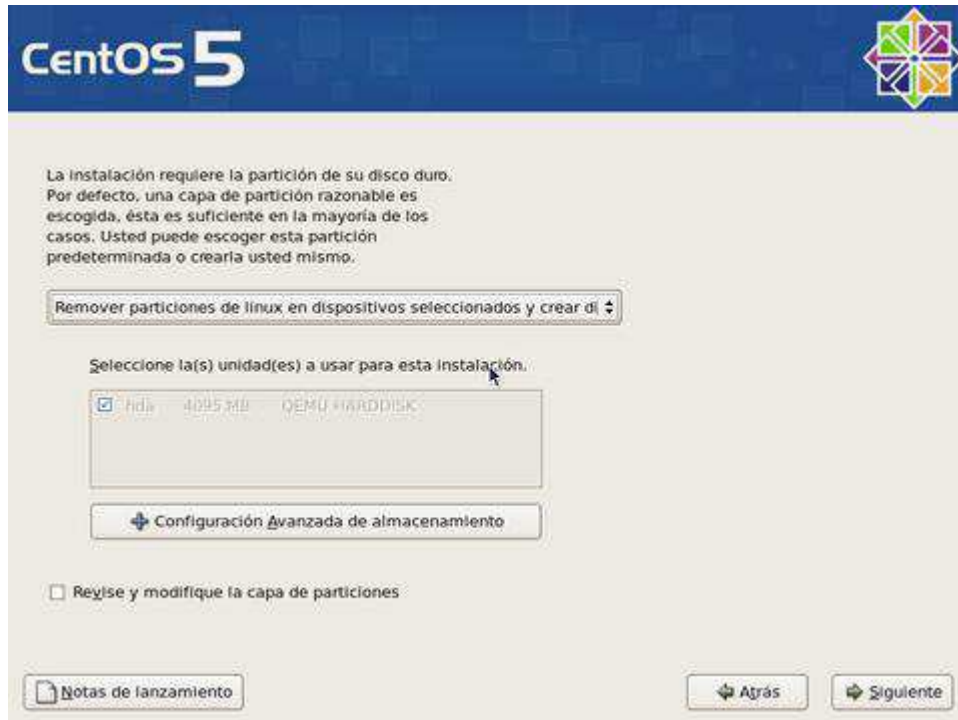


Salvo que exista una instalación previa que se desee actualizar (no recomendado), deje seleccionado «**Instalar CentOS**» y haga clic en el botón «**Siguiete**» a fin de realizar una instalación nueva.

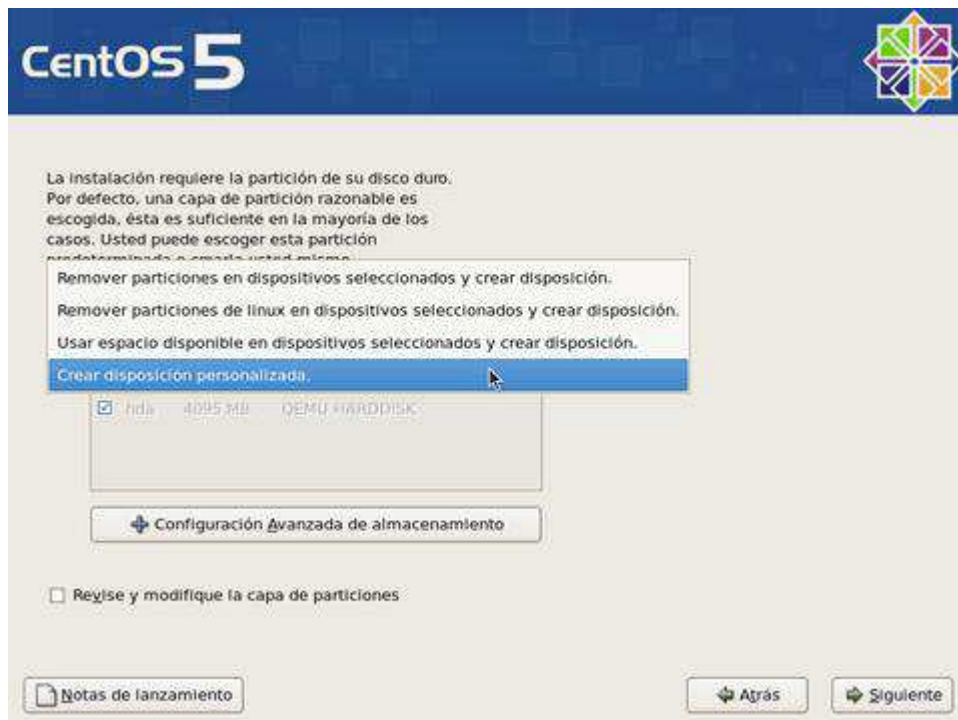


Para crear las particiones de forma automática, lo cual puede funcionar para la mayoría de los usuarios, puede seleccionar:

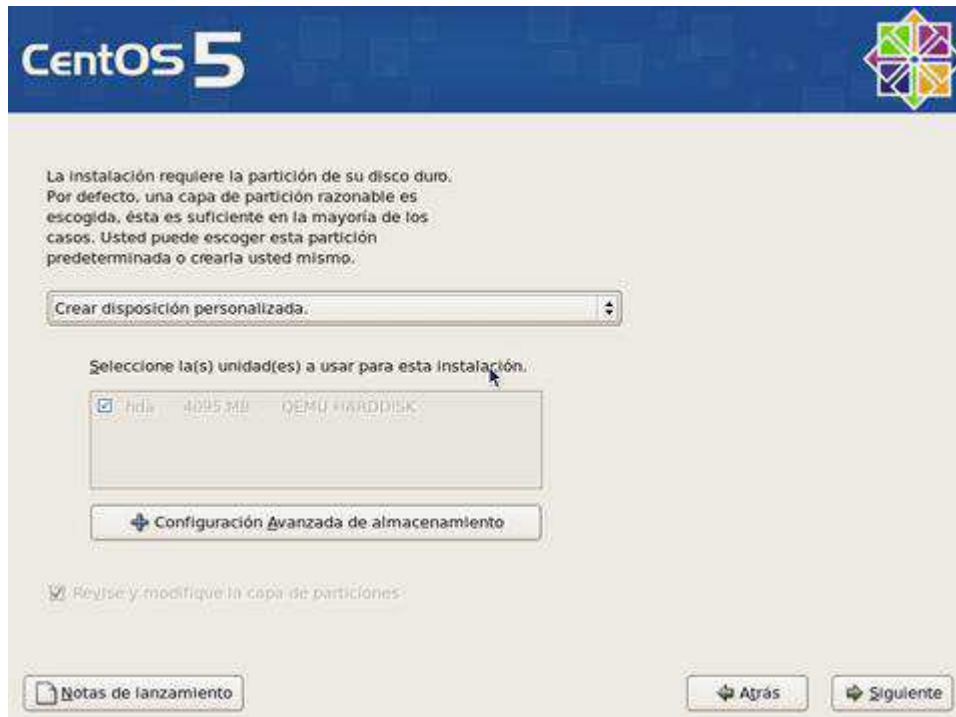
- **«Remover particiones en dispositivos seleccionados y crear disposición»**, lo cual eliminaría cualquier partición de cualquier otro sistema operativo presente, y creará de forma automática las particiones necesarias.
- **«Remover particiones de GNU/Linux en dispositivos seleccionados y crear disposición»**, lo cual eliminaría cualquier partición otra instalación de GNU/Linux presente, y creará de forma automática las particiones necesarias.
- **«Usar espacio disponible en dispositivos seleccionados y crear disposición»**, lo cual creará de forma automática las particiones necesarias en el espacio disponible.



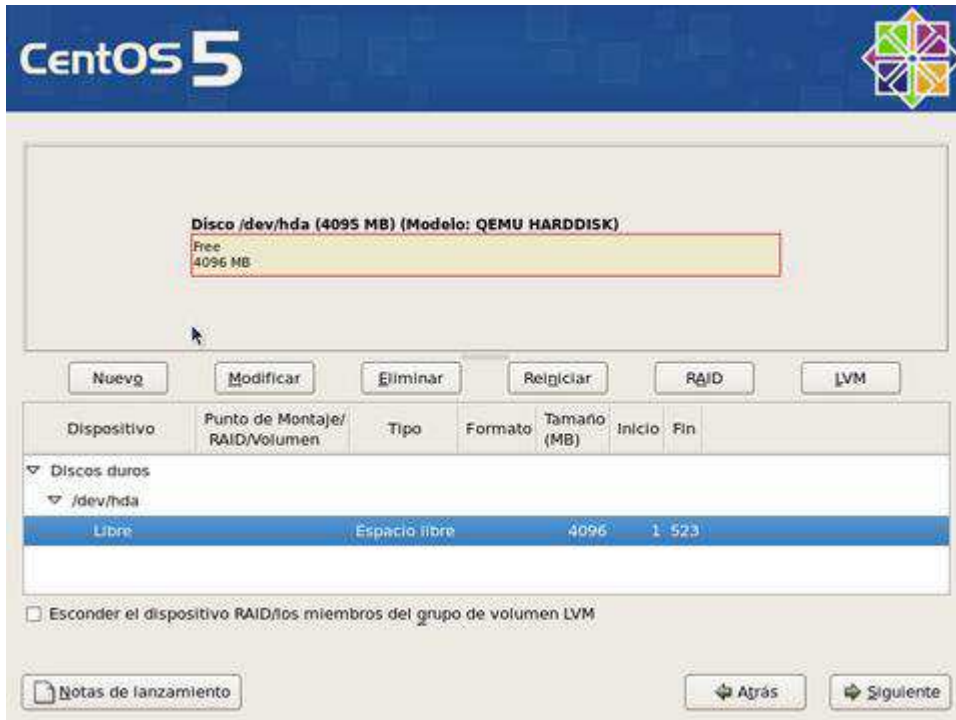
Conviene crear una disposición que permita un mayor control. Seleccione «**Crear disposición personalizada**».



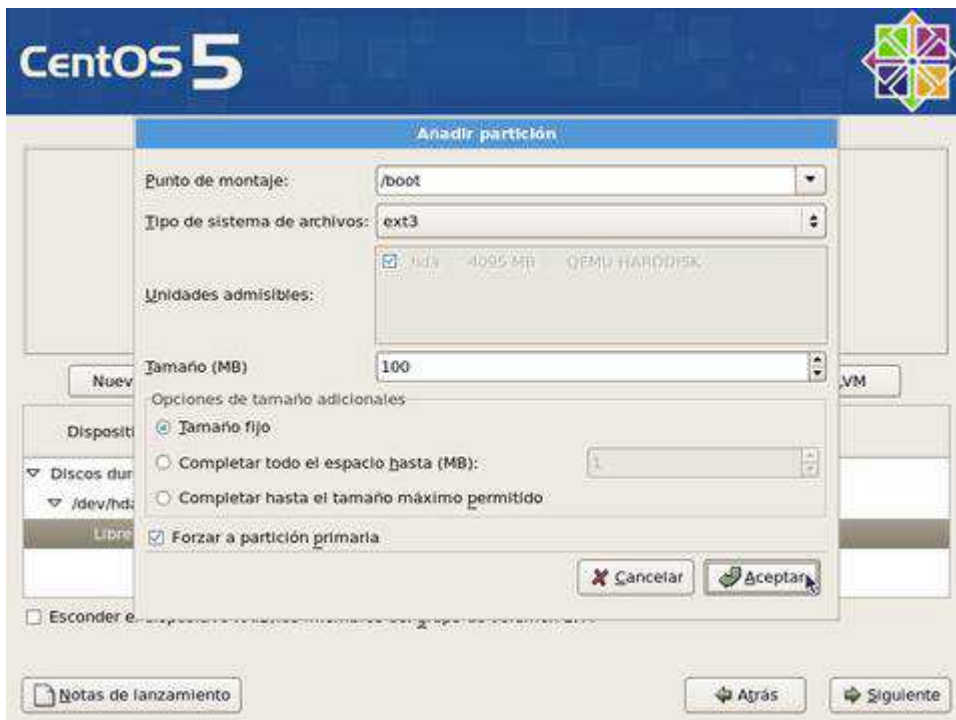
Una vez seleccionado «**Crear disposición personalizada**», haga clic sobre el botón «**Siguiente**».



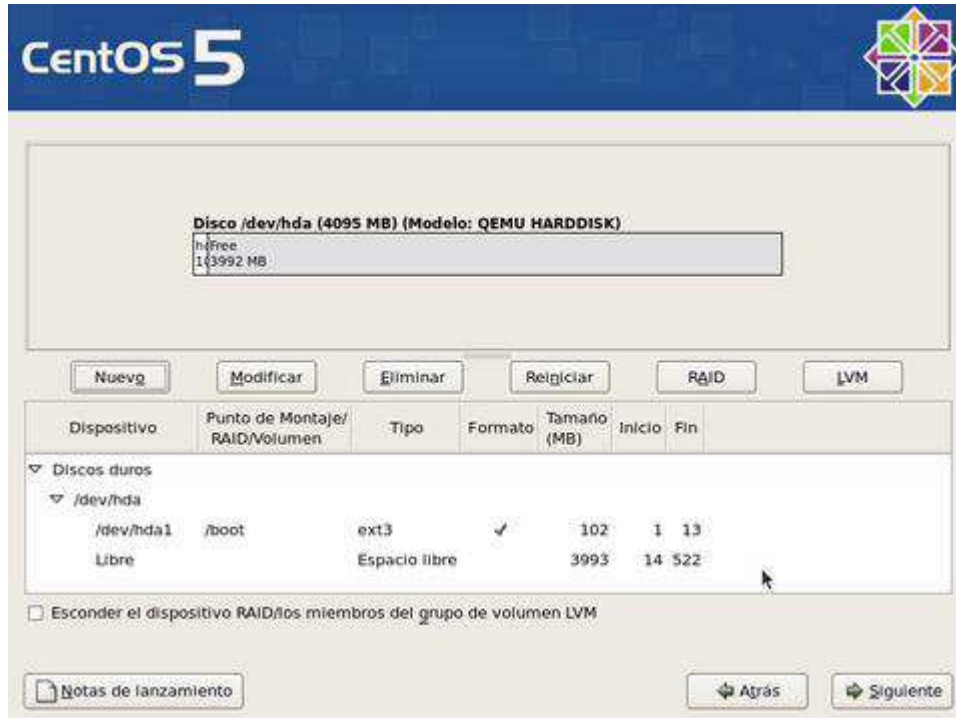
La herramienta de particiones mostrará el espacio disponible. Haga clic en el botón «**Nuevo**».



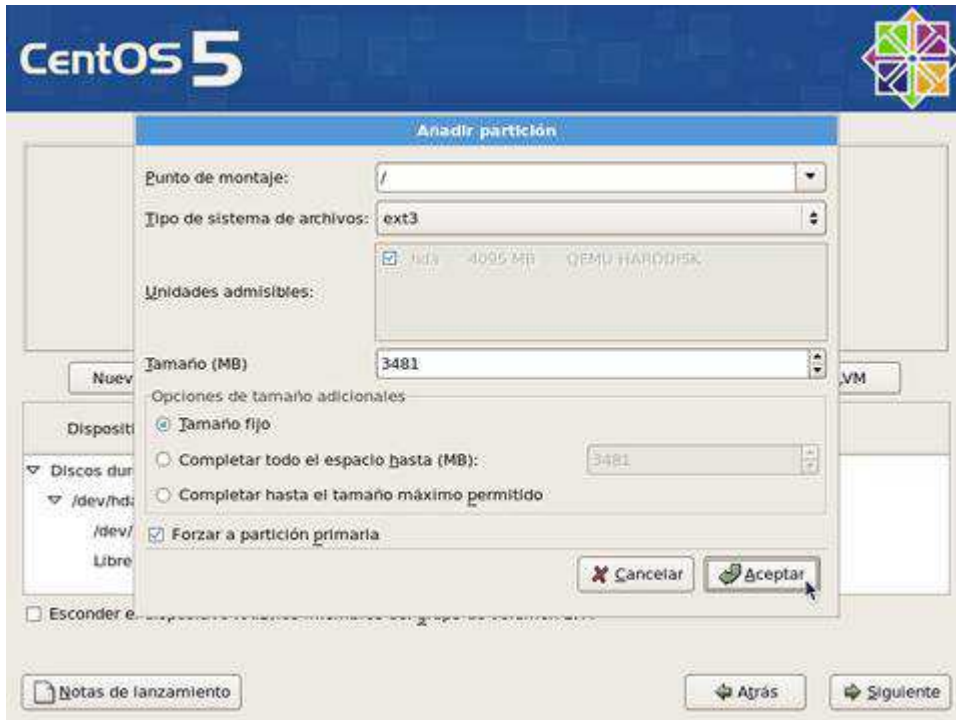
Asigne 100 MB a la partición /boot y defina ésta como partición primaria, siempre que la tabla de particiones lo permita.



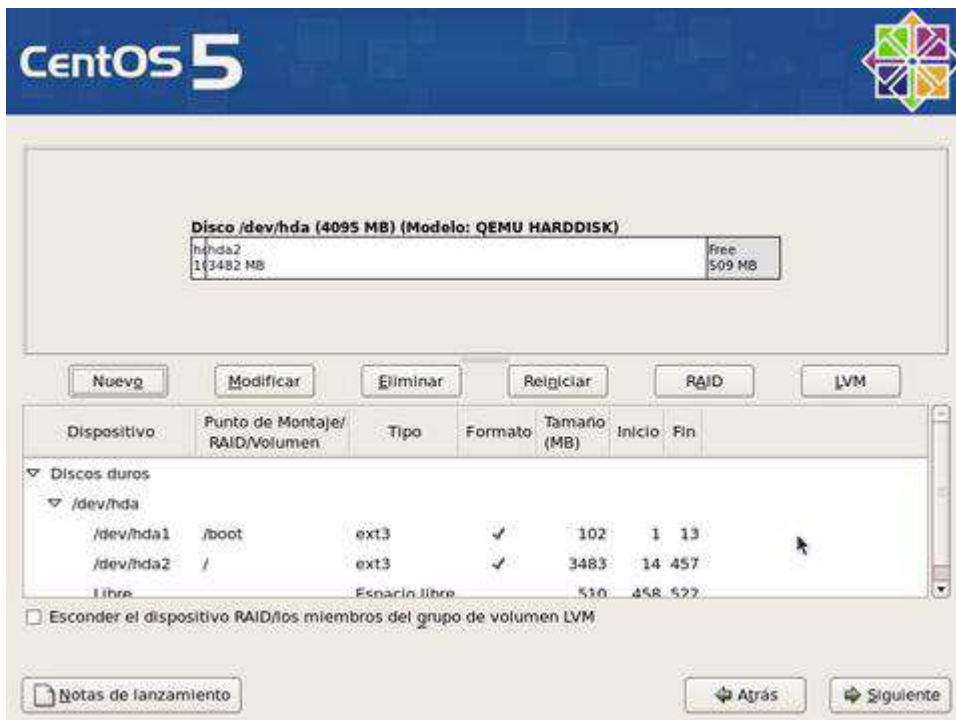
Si está conforme, haga clic otra vez en el botón «**Nuevo**» y proceda a crear la siguiente partición.



Asigne a la partición / el resto del espacio disponible menos lo que tenga calculado asignar para la partición de intercambio (200% de la memoria física, o cuanto baste para 2 GB). Se recomienda asignar / como partición primaria, siempre que la tabla de particiones lo permita.



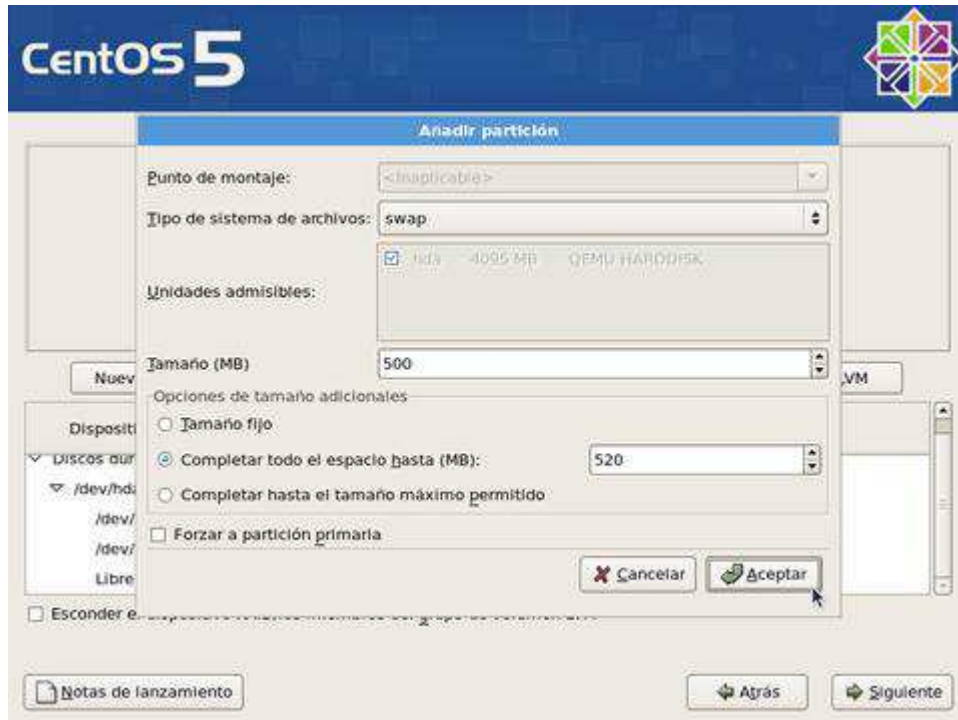
Si está conforme, haga clic otra vez en el botón «**Nuevo**» y proceda a crear la siguiente partición.



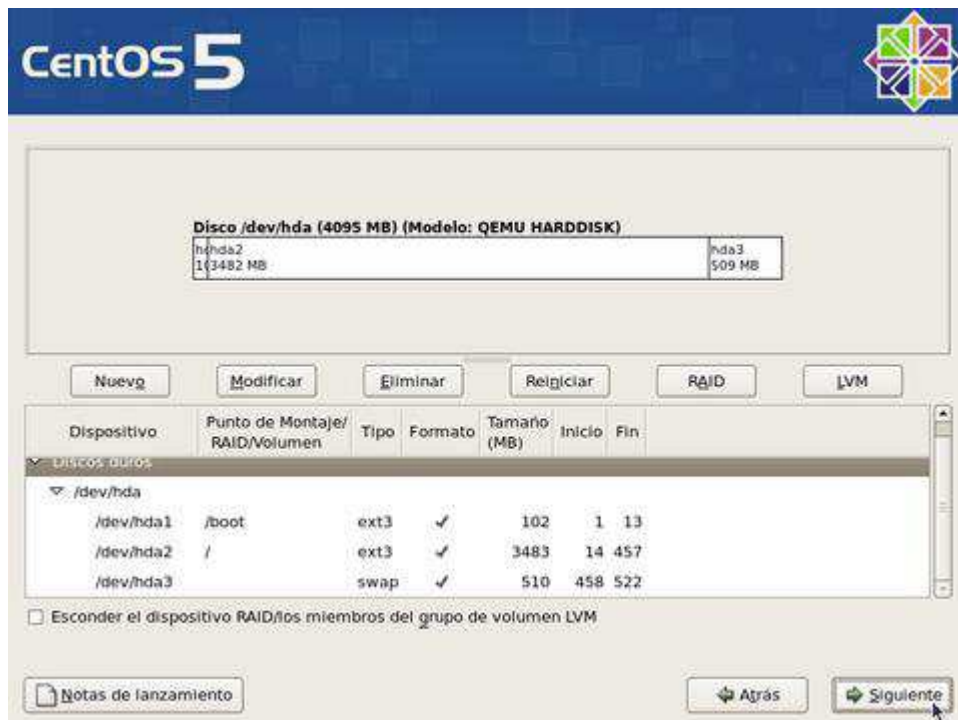
La partición para la memoria de intercambio no requiere punto de montaje. Seleccione en el campo de «**Tipo de sistema de archivos**» la opción «**swap**», asigne el 200% de la memoria física (2*RAM). Por tratarse de la última partición de la tabla, es buena idea asignarle el espacio por rango, especificando valores ligeramente por debajo y ligeramente por arriba de lo planeado.

Otras particiones que se recomienda asignar, si se dispone del espacio en disco duro suficiente, son:

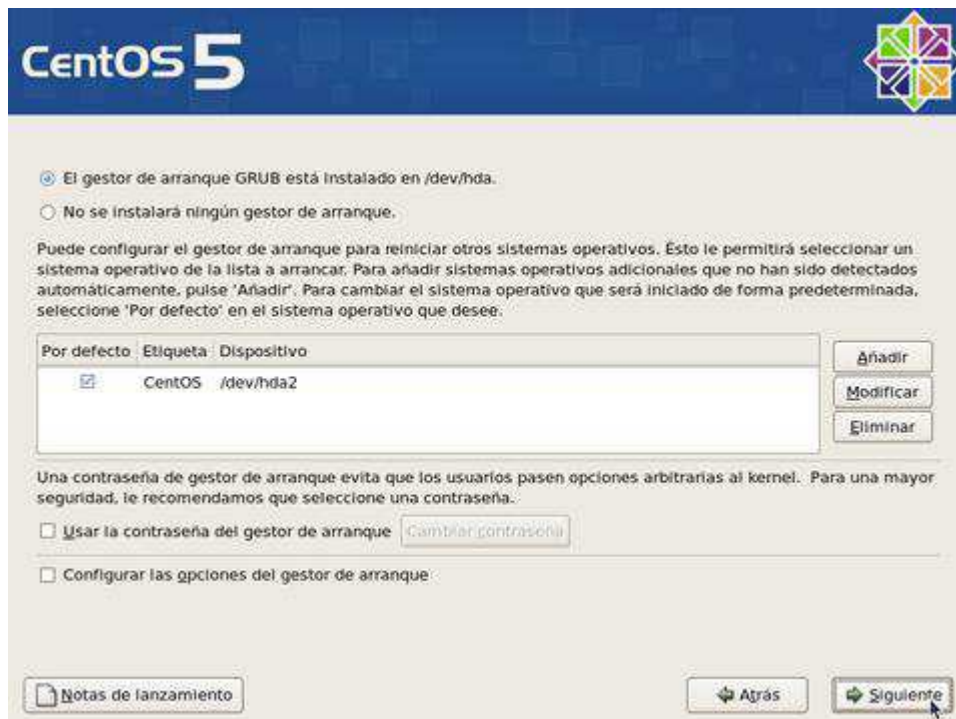
/usr	Requiere al menos 1.5 GB en instalaciones básicas. Debe considerarse el sustento lógico a utilizar a futuro. Para uso general, se recomiendan no menos de 5 GB y, de ser posible, considere un tamaño óptimo de hasta 8 GB en instalaciones promedio.
/tmp	Requiere al menos 350 MB y puede asignarse hasta 2 GB o más dependiendo de la carga de trabajo y tipo de aplicaciones. Si por ejemplo el sistema cuenta con un grabador de DVD, será necesario asignar a /tmp el espacio suficiente para almacenar una imagen de disco DVD, es decir, al menos 4.2 GB.
/var	Requiere al menos 512 MB en estaciones de trabajo sin servicios. En servidores regularmente se le asigna al menos la mitad del disco duro.
/home	En estaciones de trabajo se asigna al menos la mitad del disco duro a esta partición.



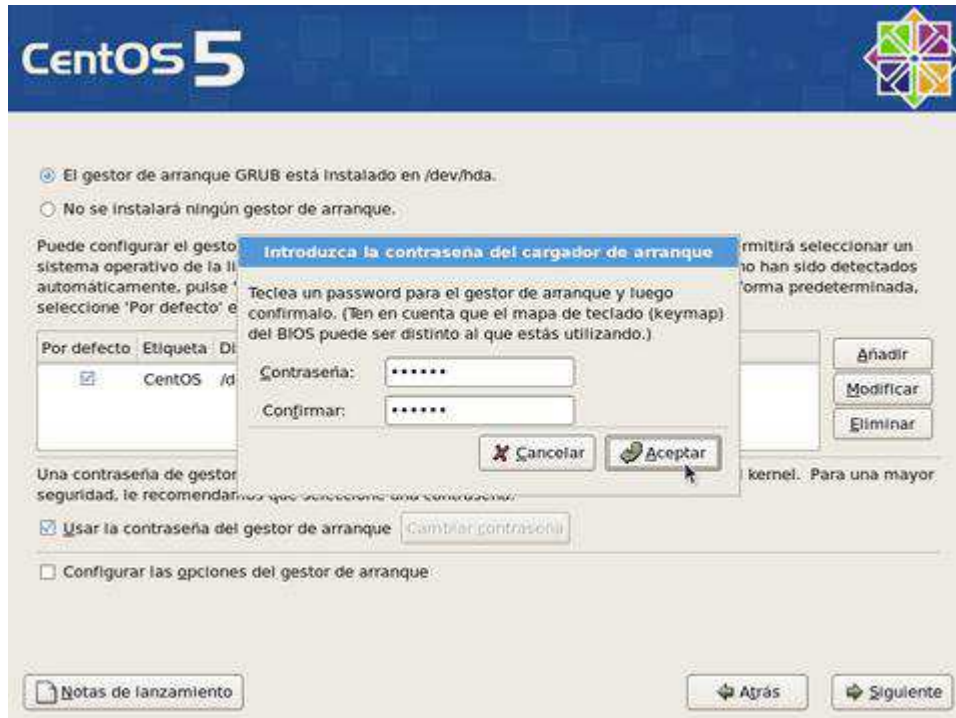
Si está conforme con la tabla de particiones creada, haga clic sobre el botón «siguiente» para pasar a la siguiente pantalla.



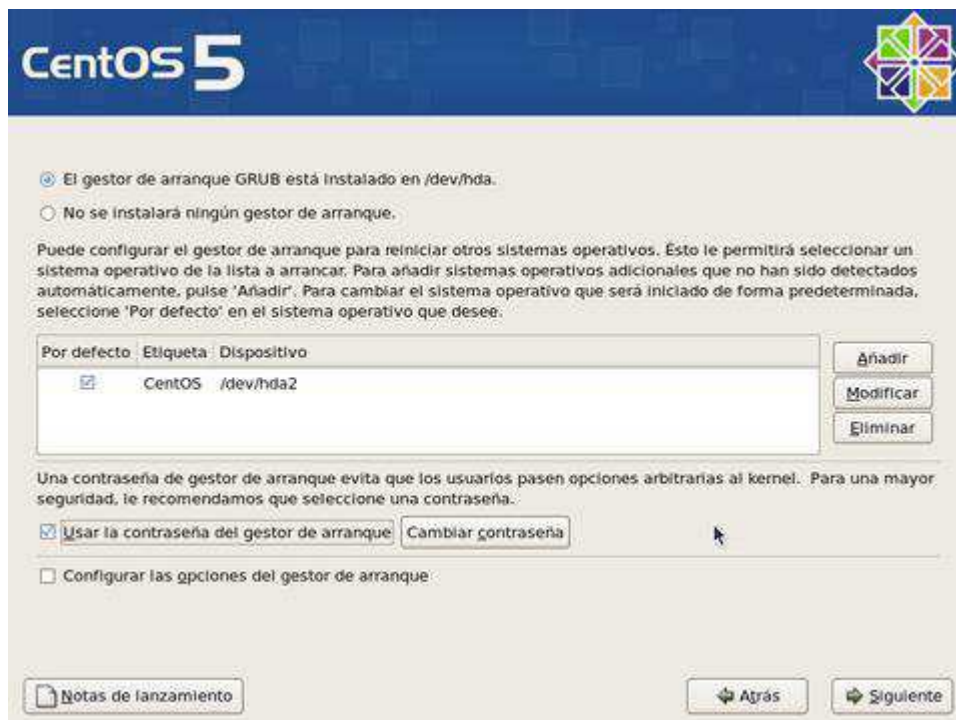
Ingresará a la configuración del gestor de arranque. Por motivos de seguridad, y principalmente con la finalidad de impedir que alguien sin autorización y con acceso físico al sistema pueda iniciar el sistema en nivel de ejecución 1, o cualquiera otro, haga clic en la casilla «**Usar la contraseña del gestor de arranque**».



Se abrirá una ventana emergente donde deberá ingresar, con confirmación, la clave de acceso exclusiva para el gestor de arranque. Al terminar, haga clic sobre el botón «**Aceptar**».



Al terminar, haga clic sobre el botón «**Siguiente**».



Para configurar los parámetros de red del sistema, haga clic sobre el botón «**Modificar**» para la interfaz eth0.



En la ventana emergente para modificar la interfaz eth0, desactive la casilla «**Configurar usando DHCP**» y especifique la dirección IP y máscara de subred que utilizará en adelante el sistema. Si no va a utilizar IPv6, también desactive la casilla. Confirme con el administrador de la red donde se localice que estos datos sean correctos antes de continuar. Al terminar, haga clic sobre el botón «**Aceptar**».



Asigne un nombre de anfitrión (HOSTNAME) para el sistema. Se recomienda que dicho nombre sea un FQDN (Fully Qualified Domain Name) resuelto al menos en un DNS local. Defina, además, en esta misma pantalla, la dirección IP de la puerta de enlace y las direcciones IP de los servidores DNS de los que disponga. Si desconoce que dato ingresar, defina éste como **localhost.localdomain**. Al terminar, haga clic sobre el botón «**Siguiente**».

CentOS 5

Dispositivos de red

Activar al inicio	Dispositivo	IPv4/Máscara de red	IPv6/Prefijo	Modificar
<input type="checkbox"/>	em0	10.0.0.50/8	Desactivado	

Nombre del Host

Configurar el nombre del host:

de forma automática a través de DHCP

manualmente (ej. "mipc.dominio.com.ar")

Configuración miscelánea

Puerta de enlace:

DNS Primario:

DNS Secundario:

Seleccione la casilla «**El sistema horario usará UTC**», que significa que el reloj del sistema utilizará UTC (Tiempo Universal Coordinado), que es el sucesor de GMT (Greenwich Mean Time, que significa Tiempo Promedio de Greenwich), y es la zona horaria de referencia respecto a la cual se calculan todas las otras zonas del mundo. Haga clic con el ratón sobre la región que corresponda en el mapa mundial o seleccione en el siguiente campo la zona horaria que corresponda a la región donde se hospedarán físicamente el sistema.



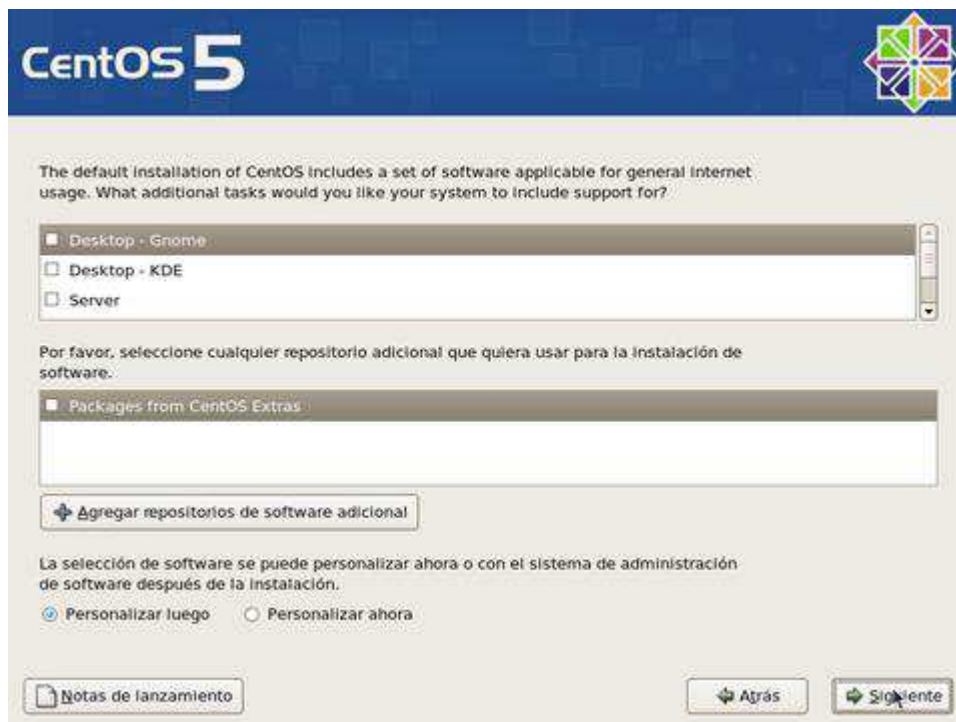
Asigne una clave de acceso al usuario **root**. Debe escribirla dos veces a fin de verificar que está coincide con lo que realmente se espera. Por razones de seguridad, se recomienda asignar una clave de acceso que evite utilizar palabras provenientes de cualquier diccionario, en cualquier idioma, así como cualquier combinación que tenga relación con datos personales.



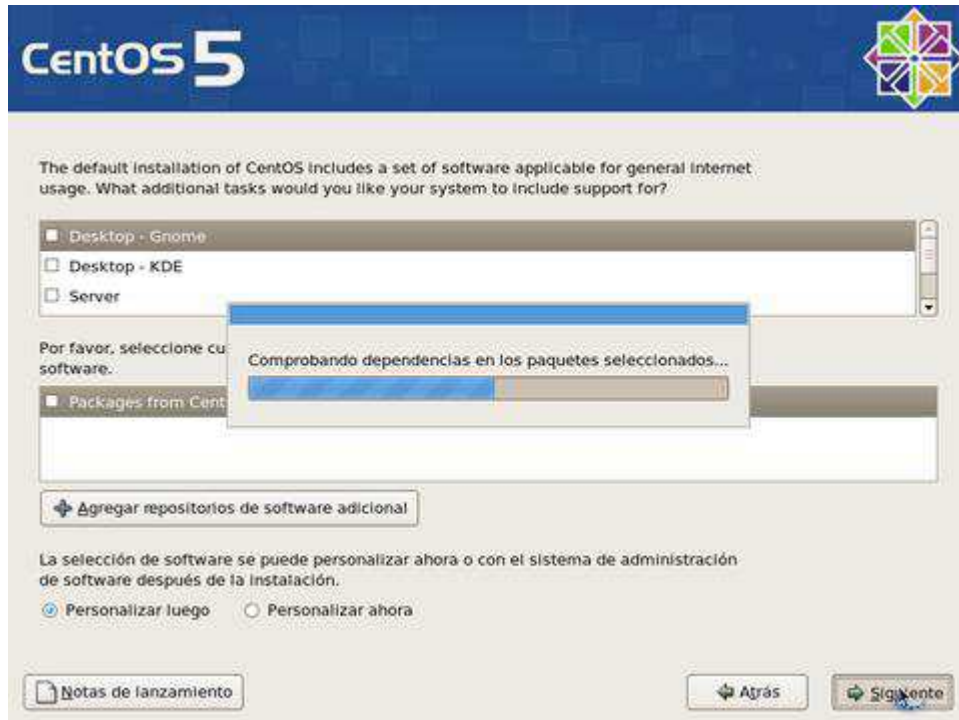
Al terminar, haga clic sobre el botón «**Siguiente**», y espere a que el sistema haga la lectura de información de los grupos de paquetes.



En la siguiente pantalla podrá seleccionar los grupos de paquetes que quiera instalar en el sistema. Añada o elimine a su conveniencia. Lo recomendado, sobre todo si se trata de un servidor, es realizar una instalación con el mínimo de paquetes, desactivando todas las casillas para todos los grupos de paquetes. El objeto de esto es solo instalar lo mínimo necesario para el funcionamiento del sistema operativo, y permitir instalar posteriormente solo aquello que realmente se requiera de acuerdo a la finalidad productiva que tendrá el sistema. Al terminar, haga clic sobre el botón «**Siguiente**».



Se realizará una comprobación de dependencias de los paquetes a instalar. Este proceso puede demorar algunos minutos.



Antes de iniciar la instalación sobre el disco duro, el sistema le informará respecto a que se guardará un registro del proceso en si en el fichero **/root/install.log**. Para continuar, haga clic sobre el botón **«Siguiente»**.



Si iniciará de forma automática el proceso de formato de las particiones que haya creado para instalar el sistema operativo. Dependiendo de la capacidad del disco duro, este proceso puede demorar algunos minutos.



Se realizará automáticamente una copia de la imagen del programa de instalación sobre el disco duro a fin de hacer más eficiente el proceso. Dependiendo de la capacidad del microprocesador y cantidad de memoria disponible en el sistema, este proceso puede demorar algunos minutos.



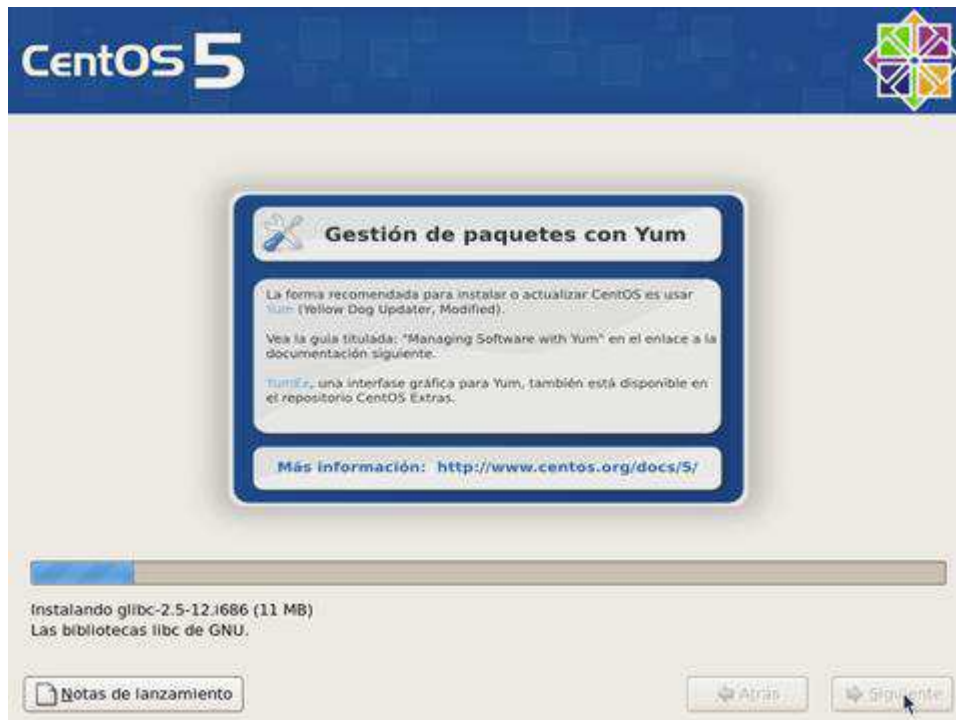
Espere a que se terminen los preparativos de inicio del proceso de instalación.



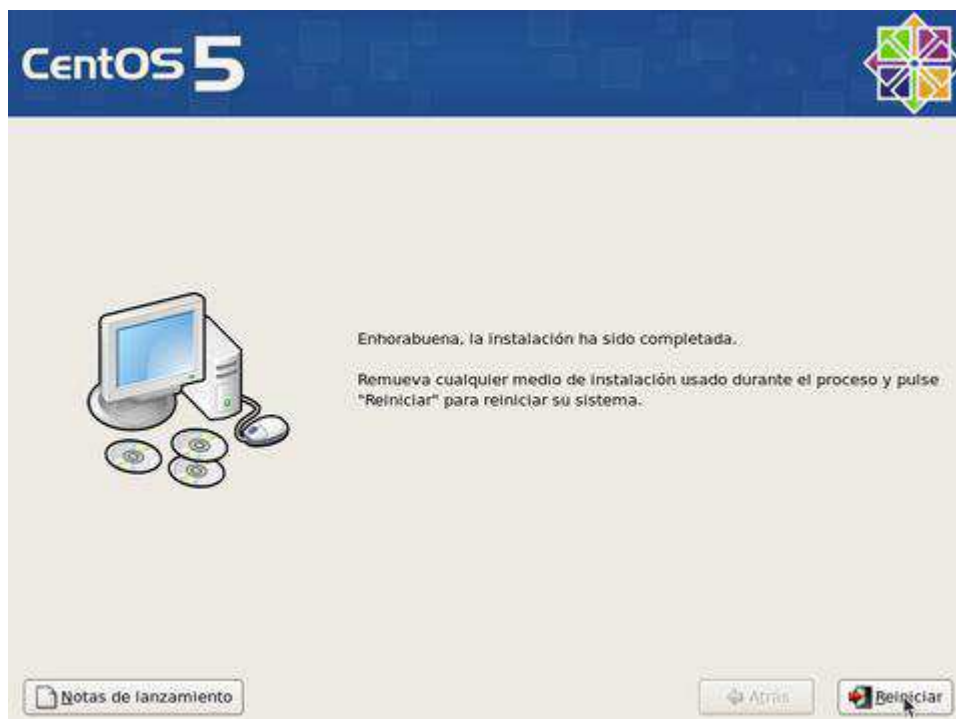
Se realizarán preparativos para realizar las transacciones de instalación de paquetes.



Iniciará la instalación de los paquetes necesarios para el funcionamiento del sistema operativo. Espere algunos minutos hasta que concluya el proceso.



Una vez concluida la instalación de los paquetes, haga clic sobre el botón **«Reiniciar»**.



BIBLIOGRAFÍA

BIBLIOGRAFÍA EN INTERNET

EBOOKS

ARANDA ALAMANSA, Joaquín, y Otros. Sistemas Operativos: Teoría y Problemas.

España: Sanz Y Torres, 2002. pp. 01 – 68

http://gqmfiv.bay.livefilestore.com/y1pCRzLKONV5C8dyeCzD0AIug0XuVQDx_vkb0mSB8XIHgiTfhCsyP-

c9kH_B4v69Waueq3dbfdyGU8/sistemas_operativos_-_teoria_y_problemas_-_j_aranda.pdf

(2009-01-13)

CARRETERO PÉREZ, Jesús, y Otros. Sistemas Operativos: Una visión aplicada.

México: McGraw Hill, 2001. pp. 33 – 74

http://www.4shared.com/get/11210612/b8172d3b/CARRETERO_Jesus_-_Sistemas_Operativos_PDF.html

(2008-11-13)

CURSOS Y MANUALES WINDOWS SERVER 2003

Curso, Manual, Tutorial - Windows 2003 Server

<http://d.scribd.com/docs/1yxdlgvi0f3283bmbns5.pdf>

(2009-02-03)

Windows Server 2003 Standard Edition

[http://www.aticana.com.ar/files/SO/Microsoft/Windows Server 2003 Standard Edition.pdf](http://www.aticana.com.ar/files/SO/Microsoft/Windows%20Server%202003%20Standard%20Edition.pdf)

(2009-02-03)

GNU/LINUX

http://www.lugro.org.ar/biblioteca/cursos/curso_intro/c178.html

(2008-11-13)

http://www.lugro.org.ar/biblioteca/cursos/curso_intro/bases.html

(2008-11-13)

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

(2008-09-22)

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

(2008/09/22)

<http://www.hispalinux.es/GNULinux>

(2008/09/22)

MANUALES DE CONFIGURACIÓN LTSP

<http://ltsp.mirrors.tds.net/pub/ltsp/docs/ltsp-4.1-en.pdf>

(2008-07-01)

<http://www.ltsp.org/~sbalneav/LTSPManual.pdf>

(2008-07-15)

http://comunidad.molinux.info/index.php/LTSP_en_la_escuela_con_Molinux

(2008-07-11)

WINDOWS SERVER 2003

http://es.wikipedia.org/wiki/Windows_Server_2003

(2008/12/12)

<http://www.microsoft.com/latam/windowsserver2003/evaluation/overview/enterprise.msp>

(2008/12/01)

<http://www.visualwin.com/>

(2008/09/22)

<http://www.scribd.com/doc/6509109/Caracteristicas-do-Windows-Server-2003>

(2008/09/22)