

Linux Ubuntu Server

Una visión práctica

1era Edición

Linux Ubuntu Server

Una visión práctica

1era Edición

Eduardo De la Hoz Correa

Emiro De la hoz Franco



Educosta – Editorial Corporación Universitaria de la Costa
Barranquilla – Atlántico – Colombia

2009.

AUTORES

Eduardo De la Hoz Correa

Ingeniero de Sistemas.

Especialista en Redes de Computadores

M.Sc. (C) en Ingeniería de Sistemas y Computación.

Docente Tiempo Completo - Investigador. Corporación Universitaria de la Costa – CUC.

edelahez6@cuc.edu.co

Emiro De la Hoz Franco

Ingeniero de Sistemas.

Especialista en Estudios Pedagógicos.

Especialista en Informática y Telemática.

M.Sc. (C) en Ingeniería de Sistemas y Computación.

Director de Programa Ingeniería de Sistemas. Corporación Universitaria de la Costa – CUC.

edelahez@cuc.edu.co

ISBN: 978-958-8511-55-9

Derechos Reservados: Esta obra es propiedad intelectual de sus autores y los derechos de publicación han sido legalmente transferidos al editor. Queda prohibida su reproducción parcial o total por cualquier medio sin permiso por escrito del propietario de los derechos del Copyright.

Nota Importante: La información contenida en este libro es producto del desarrollo de la investigación “Análisis de servicios de red bajo software libre utilizando el sistema operativo Linux UBUNTU”, que obedece a la línea de investigación “Redes Convergentes” y ha sido gestado a partir de la dinámica el grupo de investigación de “Ingeniería de Software y Redes de Computadores”, registrado en Colciencias con el código “COL0077064” y adscrito al Programa de Ingeniería de Sistemas de la Corporación Universitaria de la Costa – CUC. Se recalca que esta obra puede ser empleada en el ámbito académico, como texto de referencia en asignaturas relacionadas con los sistemas operativos y redes de computadores, y también en el ámbito profesional y productivo, como guía de referencia en la implementación de servicios de redes en el sector empresarial. Educosta no será jurídicamente responsable por: errores u omisiones; daños y perjuicios que se pudieran atribuir al uso de la información contenida en este libro, ni a la utilización indebida que pudiera dársele.

Edición autorizada para venta en todo el territorio colombiano.

Impreso en Barranquilla – Atlántico - Colombia - 2009.

Dedico esta obra a Dios, a mi esposa y compañera Dayana y mi pequeño
Eduardo, así como a mis padres a quienes debo todo lo terreno
Eduardo.

Dedico esta obra a Dios todopoderoso, luz que ilumina mi camino,
a mis padres (Rosalba y Emiro) grandes forjadores de mi futuro
y a mis amados hijos (Juan y Alejandro) promotores de mis más
grandes alegrías
Emiro.

PRÓLOGO

Existen en la actualidad gran variedad de distribuciones libres del Sistema Operativo Linux, el cual ha ganado un espacio preponderante por sus características de multiusuario, multitarea, estabilidad, seguridad, conectividad, escalabilidad y compatibilidad con gran variedad de aplicaciones. Una de las distribuciones más usadas en diferentes ámbitos, entre ellos el científico, académico, industrial y comercial, es la distribución UBUNTU, ésta ha sido patrocinada por la empresa Canonical Ltda, organización británica propiedad del sudafricano Mark Shuttleworth.

UBUNTU posee múltiples herramientas de configuración de servicios tales como DHCP (Dynamic Host Configuration Protocol), DNS (Domain Name System), LDAP y SAMBA, PROXY y el servidor WEB APACHE, entre otros. Por ello su funcionalidad es bastante amplia en lo referente a procesos de configuración de servicios para estaciones de trabajo y servidores.

La presente obra contiene una descripción conceptual de las generalidades de los sistemas operativos de libre distribución, el proceso de instalación y configuración base del sistema operativo Linux UBUNTU y la descripción conceptual y aplicada de los diferentes servicios anteriormente mencionados. El libro es una muy buena referencia a nivel de procedimientos a seguir para la configuración de servicios, debido a que de una forma didáctica y suficientemente ilustrativa proporciona la información necesaria para que el usuario administrador de red pueda configurar, montar y desmontar servicios.

El texto va dirigido a estudiantes de carreras afines a las Ciencias Computaciones e Ingeniería de Sistemas, sin embargo puede ser un libro guía de referencia para cualquier estudiante y profesional de otras disciplinas, que desee profundizar en la configuración de servicios sobre el Sistema Operativo Linux UBUNTU. Se ha percibido que en él se presentan suficientes definiciones a nivel conceptual de los servicios y aspectos necesarios para el abordaje de los procesos prácticos, de todas formas, se aprecia que su propósito es la implementación práctica de conceptos, por ello se recomienda como una obra que posibilita en el estudiante la apropiación de diferentes temáticas desde un enfoque práctico y aplicado.

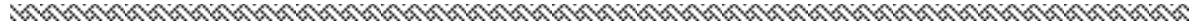
M.Sc. Ing. Jesús Estrada.

AGRADECIMIENTOS

Este libro pudo desarrollarse gracias a la ayuda incondicional de todo un equipo de trabajo que involucra a personal directivo, administrativo y académico de la Corporación Universitaria de la Costa – CUC. Entre ellos se destaca al consejo de fundadores de la institución, conformado por Eduardo Crissien Samper, Rubén Maury Pertuz (q.e.p.d.), Nulvia Borrero Barraza, María Ardila De Maury, Ramiro Moreno Noriega, Rodrigo Niebles De la Cruz (q.e.p.d.) y Miguel Antequera Stand. A los rectores Tito José Crissien Borrero, rector general y Mario Maury Ardila, rector ejecutivo, a Raquelina Villa Mendoza, vicerrectora académica, a Henry Maury Ardila, vicerrector de investigación, a Nadia Olaya Coronado, decana de la facultad de Ingenierías, a Perla Blanco Miranda coordinadora de producción intelectual, a Rodolfo José Cañas Cervantes, Ingeniero de Sistemas egresado de la Institución, el cual hizo significativos aportes de índole técnico que contribuyeron en gran medida a la realización de esta obra. Por último, de manera muy afectuosa agradecemos al Ingeniero Jesús Estrada, evaluador del libro, el cual hizo considerables aportaciones que invitaron a la reflexión, afinamiento y fortalecimiento de éste libro.

Eduardo y Emiro.

CONTENIDO



	Pág
Capítulo No 1. GENERALIDADES.....	3
1.1 DEFINICIÓN DE SISTEMA OPERATIVO.....	3
1.2 SISTEMAS OPERATIVOS LIBRES.....	4
1.2.1. Unix.....	5
1.2.2. BSD.....	6
1.2.3. GNU/LINUX.....	9
1.3 EL SOFTWARE LIBRE.....	10
1.4 FREEWARE Y SHAREWARE.....	12
1.4.1. Freeware.....	12
1.4.2. Shareware.....	13
1.5 LINUX.....	14
1.5.1. El proyecto.....	14
1.5.2. El Núcleo o Kernel.....	15
1.5.3. Distribuciones de Linux.....	17
1.5.4. Versiones LIVE.....	23
Capítulo No 2. INSTALACIÓN DE UBUNTU SERVER.....	27
Capítulo No 3. EL SERVICIO DHCP (Dynamic Host Configuration Protocol).....	50
3.1 FUNCIONAMIENTO DHCP.....	50
3.1.1. Asignación automática.....	51

3.1.2. Asignación dinámica	51
3.1.3. Asignación manual.....	51
3.1.4. Formato del mensaje DHCP	52
3.2 INSTALACIÓN Y CONFIGURACIÓN	55
3.2.1. Declaraciones	60
3.2.2. Parámetros	63
3.2.3. Opciones.....	67
3.2.4. Ejemplo de dhcpd.conf.....	68
Capítulo No 4. EL SERVICIO DNS (Domain Name System).....	76
4.1 FUNCIONAMIENTO DNS.....	77
4.2 CONCEPTOS	79
4.2.1. Clientes	79
4.2.2. Servidores DNS	79
4.3 INSTALACIÓN.....	80
4.3.1. Activando los repositorios	80
4.3.2. Instalación de bind	81
4.3.3. Configuración de la Red.....	81
4.3.4. Servidor Ubuntu	81
4.4 CLIENTES WINDOWS	84
4.5 ESTRUCTURA DE BIND	85
4.5.1. named.conf	85
4.5.2. named.conf.options.....	86
4.5.3. db.root.....	86
4.6 ZONAS	86

4.7 TIPOS DE REGISTROS DNS.....	87
4.7.1. SOA	87
4.7.2. A (Address).....	89
4.7.3. CNAME (Canonical Name)	89
4.7.4. NS (Name Server)	90
4.7.5. MX (Mail Exchange).....	90
4.8 CREACIÓN DE UN DOMINIO LOCAL.....	91
Capítulo No 5. LDAP y SAMBA	99
5.1 SAMBA	99
5.1.1. Instalación de Samba	99
Capítulo No 6. EL SERVICIO PROXY	110
6.1 PROXY WEB (Proxy cache de web).....	112
6.2 PROXIES TRANSPARENTES.....	114
6.3 REVERSE PROXY	115
6.4 PROXY NAT (NETWORK ADDRESS TRANSLATION)	116
6.4.1. NAT Estático	118
6.4.2. NAT Dinámico.....	119
6.4.3. NAT (Network Address Port Translation).....	120
6.5 PROXY ABIERTO.....	122
6.6 SERVIDOR PROXY SQUID	122
6.7 INSTALACIÓN Y CONFIGURACIÓN PROXY SQUID.....	124
6.8 PARÁMETROS BÁSICOS	126
6.8.1. Http_port	126
6.8.2. Cache_mem.....	127

6.8.3. Cache_dir.....	128
6.8.4. Control de acceso	129
6.8.5. Listas de Control de Acceso - ACL	129
6.8.5.1. Funcionamiento de las ACLs	132
6.8.5.2. Clasificación de las ACLs.....	133
6.8.5.2.1. Filtrado de paquetes de acuerdo a la Dirección Origen	133
6.8.5.2.2. Filtrado de paquetes de acuerdo al Direccionamiento, Protocolo o Puerto.....	135
6.8.5.2.3. ACLs Enumeradas	136
6.8.5.2.4. ACLs con denominación	137
6.8.5.3. Gestión de Comandos sobre Listas de Control de Acceso	137
6.8.5.3.1. Sintaxis completa para la creación de ACLs.....	137
6.8.5.3.2. Mostrar ACLs existentes	138
6.8.5.3.3. Eliminación de una ACL.....	138
6.8.6. Reglas de Control de Acceso.....	138
6.8.6.1. Aplicando Listas y Reglas de control de acceso	139
6.8.6.2. Restricciones de acceso a sitios web.....	142
Capítulo No 7. EL SERVIDOR WEB APACHE	147
7.1 VENTAJAS DE APACHE.....	148
7.2 SERVICIOS QUE OFRECE APACHE	149
7.3 DIRECTORIOS BÁSICOS	149
7.4 DESCARGA.....	150
7.5 CONFIGURACIÓN	152

LISTA DE FIGURAS



	Pág
Figura No 1. Logotipo del Sistema Operativo FreeBSD.....	5
Figura No 2. Logotipo Sistemas Operativo Linux (TUX)	5
Figura No 3. Logotipo del Proyecto GNU.....	9
Figura No 4. Mascota de Linux (TUX).....	15
Figura No 5. Logotipo Distribución RED HAT	19
Figura No 6. Logotipo Distribución Centos.....	20
Figura No 7. Logotipo Distribución Fedora.....	21
Figura No 8. Logotipo Distribución Debian.....	22
Figura No 9. Logotipo Distribución OpenSUSE	22
Figura No 10. Logotipo Distribución Ubuntu	23
Figura No 11. Proceso de Instalación Selección de idioma	27
Figura No 12. Opciones de inicio de instalación	29
Figura No 13. Carga del núcleo o Kernel de Linux	29
Figura No 14. Selección del idioma del sistema	30
Figura No 15. Proceso de Instalación - Carga de componentes adicionales	30
Figura No 16. Proceso de Instalación - Configuración de red.....	31
Figura No 17. Proceso de Instalación – Asignación del nombre de la máquina	31
Figura No 18. Proceso de Instalación - Particionamiento de discos	32
Figura No 19. Proceso de Instalación - Discos disponibles	33
Figura No 20. Proceso de Instalación – Particionamiento de Discos 1	34
Figura No 21. Proceso de Instalación – Particionamiento de Discos 2.....	34
Figura No 22. Proceso de Instalación – Particionamiento de Discos 3.....	35

Figura No 23. Proceso de Instalación – Particionamiento de Discos 4.....	36
Figura No 24. Proceso de Instalación – Particionamiento de Discos 5.....	37
Figura No 25. Proceso de Instalación – Particionamiento de Discos 6.....	37
Figura No 26. Proceso de Instalación – Particionamiento de Discos 7.....	37
Figura No 27. Proceso de Instalación – Particionamiento de Discos 8.....	38
Figura No 28. Proceso de Instalación – Particionamiento de Discos 9.....	38
Figura No 29. Proceso de Instalación – Particionamiento de Discos 10.....	39
Figura No 30. Proceso de Instalación – Particionamiento de Discos 11.....	40
Figura No 31. Proceso de Instalación – Particionamiento de Discos 12.....	40
Figura No 32. Proceso de Instalación – Particionamiento de Discos 13.....	40
Figura No 33. Proceso de Instalación – Particionamiento de Discos 14.....	41
Figura No 34. Proceso de Instalación – Particionamiento de Discos 15.....	41
Figura No 35. Proceso de Instalación – Particionamiento de Discos 16.....	42
Figura No 36. Proceso de Instalación – Formateo de Particiones.....	42
Figura No 37. Proceso de Instalación – Particionamiento de Discos 17.....	43
Figura No 38. Proceso de Instalación – Configuración usuarios y contraseñas 1.....	43
Figura No 39. Proceso de Instalación – Configuración usuarios y contraseñas 2.....	44
Figura No 40. Proceso de Instalación – Configuración usuarios y contraseñas 3.....	44
Figura No 41. Proceso de Instalación – Configuración del gestor de paquetes.....	45
Figura No 42. Proceso de Instalación – Selección de Servicios.....	46
Figura No 43. Proceso de Instalación – Selección de Servicios.....	46
Figura No 44. Proceso de Instalación – Culminación 1.....	47
Figura No 45. Proceso de Instalación – Culminación 2.....	47
Figura No 46. Formato del mensaje DHCP.....	52
Figura No 47. Autenticación por consola.....	55
Figura No 48. Actualización de la base de datos de paquetes.....	56
Figura No 49. Instalación del servidor DHCP.....	57
Figura No 50. Descarga del paquete de instalación DHCP.....	58
Figura No 51. Configuración de la tarjeta de Red para asignación de IPs.....	59
Figura No 52. Creación de la copia del archivo de configuración DHCP.....	68

Figura No 53. Configuración del archivo dhcpd.conf 1	69
Figura No 54. Configuración del archivo dhcpd.conf 2	70
Figura No 55. Reinicio de los servicios	71
Figura No 56. Configuración del PC para detección de IP dinámica	72
Figura No 57. Verificación por consola de direccionamiento IP dinámico	72
Figura No 58. Prueba de conectividad entre PCs	73
Figura No 59. Descarga del Instalador DNS	82
Figura No 60. Inicialización del servicio	82
Figura No 61. Edición del archivo Interfaces	83
Figura No 62. Edición del archivo resolv.conf	83
Figura No 63. Asignación del Servidor DNS preferido	84
Figura No 64. Archivo de configuración de Zonas named.conf	87
Figura No 65. Edición del archivo named.conf	92
Figura No 66. Agregando las Zonas en el archivo named.conf	92
Figura No 67. Creación del archivo db.labredes.com	94
Figura No 68. Configuración del archivo db.labredes .com	95
Figura No 69. Comando para reiniciar el bind	95
Figura No 70. Verificando la existencia del servidor	96
Figura No 71. Topología de la Red para el servicio DNS	¡Error! Marcador no definido.
Figura No 72. Flujo de documentos sin Proxy instalado	110
Figura No 73. Figura No 72. Flujo de documentos con Proxy instalado	111
Figura No 74. Proceso de Instalación del SQUID	124
Figura No 75. Creación de una copia del archivo de configuración squid.conf	124
Figura No 76. Edición del archivo squid.conf	125
Figura No 77. Descarga del Apache	150
Figura No 78. Culminación de la descarga	151
Figura No 79. Configuración del archivo apache2.conf	151

Capítulo No

GENERALIDADES



Capítulo No 1. GENERALIDADES

1.1 DEFINICIÓN DE SISTEMA OPERATIVO

La definición exacta de lo que es un Sistema Operativo, aún se encuentra en continuo desarrollo, el avance vertiginoso de los sistemas de cómputo dejan de paso las que surgen con el día a día, pero lo que sí es claro es que cada autor lo define según su punto de vista en su área de estudio, profundidad y sobre todo teniendo presente el cambiante mundo de la tecnología. A continuación se presentan cuatro posibles definiciones:

Un Sistema Operativo es un conjunto de herramientas lógicas, instaladas en el hardware, haciendo posible la utilización del componente físico de los equipos de cómputo, siendo estos últimos los que proporciona la capacidad bruta de procesamiento; los sistemas operativos ponen dicha capacidad al alcance de los usuarios y administran cuidadosamente el hardware para lograr un buen rendimiento.

Un sistema Operativo es un administrador de recursos; que posibilita la gestión del hardware del equipo o dispositivo que le contiene, tales como: procesadores, medios de almacenamiento, dispositivos de entrada/salida de datos y de comunicación.

Un Sistema Operativo es un programa que actúa como intermediario entre el usuario y el hardware del computador y su propósito es proporcionar el entorno en el cual el primero pueda ejecutar programas. Teniendo como objetivo principal, facilitar la usabilidad del sistema computacional por parte del usuario,

proporcionando las herramientas que hagan posible la eficiente interacción con el sistema.

Un Sistema Operativo es el conjunto de programas que controla la ejecución otros programas o aplicaciones y actúa como una interfaz entre el usuario y el hardware de una computadora, lo que indica que, un Sistema Operativo explota y administra los recursos de hardware de la computadora con el objeto de proporcionar un conjunto de servicios a los usuarios del sistema.

En resumen, se podría decir que los Sistemas Operativos son un conjunto de programas que proporcionan la interfaz del hardware con el usuario, y que tiene dos funciones primordiales, que son: la gestión del hardware refiriéndose al hecho de administrar de una forma más eficiente los recursos de la máquina y la de facilitar el trabajo al usuario permitiéndole una comunicación con los dispositivos de la máquina.

1.2 SISTEMAS OPERATIVOS LIBRES

Los sistemas operativos, en cuanto a su procedencia se pueden clasificar en libres o en propietarios, estos últimos son desarrollos producto de una empresa que tiene como principal fin la explotación del sistema operativo para la consecución de ingresos económicos por efectos de su comercialización o distribución de licencias de uso.

En cuanto a los sistemas operativos libres, también denominados de libre distribución, de código abierto (open source), éstos han sido desarrollados por comunidades académicas que plantean como filosofía fundamental la distribución tanto del sistema, como del código del programa que le constituye, para fines de uso y modificación, sin beneficio económico por efectos de su distribución. Existe

gran variedad de sistemas operativos en este tipo de denominación y éstos serán abordados en mayor detalle a continuación.



Figura No 1. Logotipo del Sistema Operativo FreeBSD¹



Figura No 2. Logotipo Sistemas Operativo Linux (TUX)²

1.2.1. Unix

Unix es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy. Durante fines de la década de 1970 y principios de la década de 1980, la influencia de Unix en ambientes académicos propició su adopción en masa, en especial la variante BSD, que había surgido en la Universidad de California, Berkeley, en las compañías de aquel entonces, siendo la más destacada Sun Microsystems. Hoy en día, junto a los sistemas Unix certificados, también se pueden encontrar sistemas similares a Unix, como Linux y los derivados de BSD.

Unix posee las siguientes características:

- Es un sistema operativo multiusuario, con capacidad de simular multiprocesamiento y procesamiento no interactivo.
- Está escrito en un lenguaje de alto nivel: C.
- Dispone de un lenguaje de control programable llamado SHELL.

¹ Disponible en Internet: <<http://www.freebsd.org/es/>>

² Disponible en Internet: <<http://www.home.unix-ag.org/simon/penguin/>>

- Ofrece facilidades para la creación de programas y sistemas y el ambiente adecuado para las tareas de diseños de software.
- Emplea manejo dinámico de memoria por intercambio o paginación.
- Tiene capacidad de interconexión de procesos.
- Permite comunicación entre procesos.
- Emplea un sistema jerárquico de archivos, con facilidades de protección de archivos, cuentas y procesos.
- Tiene facilidad para redireccionamiento de Entradas/Salidas.
- Garantiza un alto grado de portabilidad.

El sistema se basa en un núcleo llamado Kernel, que reside de forma permanente en 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.

1.2.2. BSD

BSD son las siglas de “Berkeley Software Distribution”. Así se llamó a las distribuciones de código fuente que se hicieron en la Universidad de Berkeley en California y que en origen eran extensiones del sistema operativo UNIX de AT&T Research³.

El sistema operativo completo incluye:

- El Kernel BSD, que se encarga de la programación del tiempo de ejecución de los procesos, la gestión de memoria, el multiproceso simétrico (SMP), los controladores de dispositivos, entre otros.
- La biblioteca C, la API base del sistema.
- La biblioteca C de BSD está basada en código procedente de Berkeley no del proyecto GNU.

³ Disponible en Internet: <<http://www.freebsd.org/doc/es/articles/explaining-bsd/>>

- Aplicaciones como las distintas shells, aplicaciones de gestión de ficheros, compiladores y enlazadores.
- Algunas de las aplicaciones derivan del proyecto GNU, otras no.
- El sistema X Window, que gestiona el entorno gráfico.
- El sistema X Window que se usa en la mayoría de versiones de BSD es producto de un proyecto aparte, el Proyecto XFree86⁴. Se usa el mismo código que en Linux. BSD por lo general no predetermina un “gestor de ventanas” como KDE o GNOME, aunque éstos y otros muchos estén disponibles.
- Muchos otros programas y utilidades.

A diferencia de las numerosas distribuciones de Linux tan sólo hay tres BSD libres. Cada proyecto BSD mantiene su propio árbol de fuentes y su propio Kernel. En la práctica, sin embargo, las diferencias en el entorno de usuario (“userland”) entre los distintos BSD son menores que las que hay en Linux.

Es difícil enumerar los objetivos de cada proyecto puesto que las diferencias son muy subjetivas. En general se encuentra lo siguiente:

- FreeBSD es “un avanzado sistema operativo para arquitecturas x86 compatibles (como Pentium® y Athlon™), amd64 compatibles (como Opteron™, Athlon™64 EM64T), UltraSPARC®, IA-64, PC-98 y ARM. FreeBSD es un derivado de BSD, la versión de UNIX® desarrollada en la Universidad de California, Berkeley”⁵. Tiene como meta ofrecer alto rendimiento y facilidad de uso al usuario final y es uno de los favoritos entre proveedores de contenidos web. Funciona en PC y en procesadores Alpha de Compaq. El proyecto FreeBSD cuenta con un número de usuarios significativamente mayor que los otros proyectos.

⁴ Disponible en Internet:< <http://www.xfree86.org/>>

⁵ Disponible en Internet:<<http://www.freebsd.org/es/>>

- NetBSD es un sistema operativo tipo Unix, libre, seguro y altamente portable, disponible para multitud de plataformas desde Opterons a 64-bits y sistemas de escritorio hasta dispositivos de mano y empotrados. Su buen diseño y sus características avanzadas lo hacen excelente para entornos de producción e investigación, además de tener el soporte de los usuarios con el código fuente completo⁶. Tiene como meta la Portabilidad: No en vano su lema es “of course it runs NetBSD” (que podría traducirse como “claro que funciona con NetBSD”).
- OpenBSD es un sistema operativo libre multi-plataforma basado en 4.4BSD que reúne los esfuerzos en la portabilidad, estandarización, seguridad proactiva y criptografía integrada, este sistema soporta emulación binaria de la mayoría de los programas para (Solaris), FreeBSD, Linux, BSD/OS, SunOS y HP-UX y tiene como meta la seguridad y la integridad del código combinando el concepto de código abierto y una revisión rigurosa del código que da como fruto un sistema muy correcto, elegido por instituciones preocupadas por la seguridad como bancos, entidades de cambio y departamentos gubernamentales de los EEUU.

Existen dos sistemas operativos BSD más que no son de código abierto, BSD/OS y el MacOS X de Apple:

- BSD/OS es el derivado más antiguo de 4.4BSD. No es código abierto pero es posible conseguir licencias de su código fuente a un precio relativamente bajo.
- Mac OS X es la última versión del sistema operativo para la gama Macintosh de Apple Computer Inc. El núcleo BSD Unix de éste sistema operativo, Darwin, está libremente disponible como sistema operativo de fuente abierto totalmente funcional para arquitecturas x86 y PPC. El

⁶ Disponible en Internet: <<http://www.netbsd.org/>>

sistema gráfico Aqua/Quartz y la mayoría de las demás aspectos característicos de Mac OS X son código cerrado.

BSD es uno de los sistemas operativos que ha realizado grandes contribuciones en el campo de los sistemas operativos en general, entre ellas tenemos:

- El manejo de memoria virtual paginado por demanda
- El control de trabajos
- El Fast FileSystem
- El protocolo TCP/IP
- El editor de texto vi

1.2.3. GNU/LINUX



Figura No 3. Logotipo del Proyecto GNU⁷

El proyecto GNU tuvo sus inicios en el año de 1984 por una iniciativa de la persona más relevante del movimiento del software libre en la actualidad, Richard Stallman.

El proyecto GNU fue diseñado con el objetivo de crear un sistema operativo completamente libre así como también para ser totalmente compatible con UNIX. La combinación de GNU y Linux es el sistema operativo GNU/Linux el cual actualmente se usa en millones de ordenadores.

Para asegurar que el software GNU permaneciera libre para que todos los usuarios pudieran "ejecutarlo, copiarlo, modificarlo y distribuirlo", nace la idea de la

⁷ Disponible en Internet: <<http://www.gnu.org/home.es.html>>

Licencia General Pública de GNU (GPL), la cual se diseñada para garantizar los derechos antes mencionados al tiempo que se crearan restricciones posteriores.

En 1985, Richard Stallman creó la Free Software Foundation para proveer soportes logísticos, legales y financieros al proyecto GNU. Programadores La Free Software Foundation también contrató para contribuir a GNU, aunque una porción sustancial del desarrollo fue (y continúa siendo) producida por voluntarios. A medida que GNU ganaba renombre, negocios interesados comenzaron a contribuir al desarrollo o comercialización de productos GNU y el correspondiente soporte técnico.

1.3 EL SOFTWARE LIBRE

El software libre se refiere a la libertad de código y a la capacidad de distribución que este código pueda tener; no es un fin del software libre el usufructo económico por efectos de la comercialización. Para comprender este concepto se debería pensar en la palabra libre como el concepto de “Libertad de expresión”, más no el de un producto que regalan por la compra de otro.

Se habla de “Libre” o libertad de software cuando los usuarios pueden ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Más precisamente, significa que los usuarios de programas tienen las cuatro libertades esenciales.

- La libertad de ejecutar el programa, para cualquier propósito (libertad 0).
- La libertad de estudiar y cambiar el programa para saber cómo y para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).

- La libertad de mejorar el programa y publicar sus mejoras, y versiones modificadas en general, para que se beneficie toda la comunidad (libertad 3). El acceso al código fuente es una condición necesaria.

Teniendo en cuenta los ítems anteriormente mencionados podemos decir que un programa que cumpla con ellos es software libre. Entonces, debería ser libre de redistribuir copias, con o sin modificaciones, ya sea gratis o cobrando una tarifa por distribución.

También debería tener la libertad de hacer modificaciones y usarlas como uso privativo en su trabajo o donde el desee, sin siquiera mencionar que existen. Si publica sus cambios, no debería estar obligado a notificarlo a alguien en particular, o de alguna forma en particular.

La libertad de ejecutar el programa significa que cualquier persona u organización puede usarlo en cualquier tipo de sistema de computación, para cualquier trabajo o propósito, sin estar obligado a comunicarlo a su programador, o alguna otra entidad específica.

La libertad de redistribuir copias hace referencia a la de incluir las formas binarias o ejecutables del programa, así como el código fuente; tanto para las versiones modificadas como para las que no lo están. Para que las libertades para realizar cambios y publicar versiones mejoradas, tengan sentido, debe tener acceso al código fuente del programa. Por consiguiente, el acceso al código fuente es una condición necesaria para el software libre.

Estas libertades serán reales siempre y cuando sean irrevocables y no se cometa error alguno, si el programador del software tiene la potestad de invalidar la licencia, o de cambiar retroactivamente sus términos, sin que el autor original se haya equivocado para justificarlo, el software no es libre.

Existen reglas para proteger la distribución de software como lo es el Copyleft, regla que propende que al redistribuir un programa, no se puede agregar restricciones para denegar a las demás personas las libertades principales. Esta y otras reglas son válidas siempre y cuando no entren en conflicto con las libertades principales.

Por otra parte hay que tener en cuenta que software libre no significa que no sea comercial. Un programa libre debe estar disponible tanto para su uso comercial, la programación comercial y la distribución comercial. Se da en muchos casos la posibilidad de pagar dinero para obtener copias de software libre, o haber obtenido copias sin costo. Pero lo que se debe tener claro es que no importa la procedencia de las copias, siempre tiene la libertad de copiar y modificar el software, incluso de vender copias.

Resumiendo, al hablar de software libre, se deben evitar los términos “regalar o gratuito”, porque dichos términos hacen referencia al precio, no la libertad. Algunos términos comunes como piratería implican opiniones con las que se espera no concuerde. Finalmente, hay que tener en cuenta que los criterios establecidos en este libro de software libre, requieren pensar con cuidado su interpretación. Para decidir si una licencia de software específica es una licencia de software libre, la juzgamos en base a estos criterios para determinar si su esencia concuerda conjuntamente con la terminología precisa.

1.4 FREEWARE Y SHAREWARE

1.4.1. Freeware

Según el Merriam-Webster, el término Freeware se define como: “tipo de software de computadora que se distribuye sin costo, disponible para su uso y por tiempo

ilimitado”⁸, el cual en ocasiones se distribuye junto a su código fuente aunque no siempre se hace. Este tipo de software se considera una variante gratuita del shareware, donde se tiene como fin que el usuario pruebe el producto durante un tiempo en un modo limitado, y si le satisface, pague por él, habilitando toda su funcionalidad. Suele incluir una licencia de uso, que permite su redistribución pero con algunas restricciones, como no modificar la aplicación en sí, ni venderla, y dar cuenta de su autor. El freeware suele incluir una licencia de uso, que permite su redistribución pero con algunas restricciones, como no modificar la aplicación en sí, ni venderla, y dar cuenta de su autor.

1.4.2. Shareware

Se denomina Shareware a una “modalidad de distribución de software con capacidad limitada o documentación incompleta, que está disponible para uso de prueba con poco o ningún costo, pero que puede ser actualizado mediante el pago de una cuota a su autor”⁹, aunque también existe el llamado "shareware de precio cero", pero esta modalidad es poco común.

No debe confundirse el shareware con el sistema freeware que indica que un software es totalmente gratuito, si bien es cierto que el primero se inspira y tiene sus raíces en el segundo. Tampoco debe confundirse el hecho de que un software sea Shareware o freeware con el hecho de que sea de código abierto, ya que esto último depende de la disponibilidad o no del código fuente.

Aunque el shareware se inspira en el freeware, no hay que olvidar que este último es un software totalmente gratuito, mientras que el primero en la mayoría de las veces se debe cancelar una mínima cuota. De igual forma hay que tener presente siempre la diferencia entre estos dos con el código abierto, ya que este último

⁸ Disponible en Internet: <<http://www.merriam-webster.com/dictionary/freeware>>

⁹ Disponible en Internet: <<http://www.merriam-webster.com/dictionary/Shareware>>

como se explicó en el apartado (1.2) es “libre”, mientras que los primeros siempre tienen alguna forma o sistema de comercialización.

1.5 LINUX

1.5.1. El proyecto

Linux es un sistema operativo que fue creado inicialmente como hobby por un joven estudiante, Linus Torvalds, en la Universidad de Helsinki en Finlandia. Linus tenía un interés en Minix, un pequeño sistema UNIX, y decidió desarrollar un sistema más robusto que Minix. Comenzó su trabajo en 1991 cuando se lanzó la versión 0.02 y trabajó constantemente hasta 1994, cuando se lanzó la versión 1.0 del Kernel de Linux. El núcleo o Kernel¹⁰, en el corazón de todos los sistemas Linux, es desarrollado y liberado bajo la GNU General Public License y su código fuente está disponible libremente para todos. Este es el núcleo que constituye la base sobre el cual se desarrolla un sistema operativo Linux. La actual versión completa es de 2,6 (publicada en diciembre de 2003) y su desarrollo continúa.

Aparte del hecho de que es de libre distribución, la funcionalidad de Linux, la adaptabilidad y robustez, se ha convertido en la principal alternativa para sistemas tanto Unix como Microsoft. IBM, Hewlett-Packard y otros gigantes del mundo de la informática han adoptado a Linux como su mano derecha y han decidido apoyar su desarrollo en menor o mayor escala. Es bien sabido hoy día el auge de Linux en sistemas Servidores y aunque apenas comienza a darle una fuerte lucha a los sistemas de escritorio existentes como Windows o MAC, ha sido adoptado como un hijo más en los fabricantes de hardware y por todos los fanáticos de los sistemas de alto desempeño en todo el mundo.

¹⁰ Si desea tener más información acerca del núcleo o Kernel del sistema Linux puede visitar su página oficial en internet: <http://www.linuxhq.com/>

A lo largo de la década de los 1990, lo calificaron como un proyecto de equipo aficionado, no apto para lo que la población en general necesitaba, pero gracias a los esfuerzos de los desarrolladores de sistemas de gestión de escritorio tales como KDE¹¹(K Desktop Environment o Ambiente de escritorio K) y GNOME¹² (GNU Network Object Model Environment o Ambiente de modelos de objetos de red), la suite de oficina del proyecto de OpenOffice.org y el navegador web del proyecto Mozilla, por nombrar sólo unos pocos, ahora hay una amplia gama de aplicaciones que se ejecutan en Linux y pueden ser utilizadas por cualquier persona, independientemente de sus conocimientos informáticos.

Linux tiene una mascota oficial, “Tux”, el pingüino de Linux, que fue seleccionado por Linus Torvalds para representar la imagen que se asocia con el sistema operativo. Tux fue creado por Larry Ewing, este último lo donó generosamente a la comunidad para ser libremente utilizado para promover Linux.

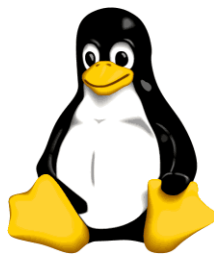


Figura No 4. Mascota de Linux (TUX)¹³

1.5.2. El Núcleo o Kernel

En el mundo de la computación, el núcleo o Kernel es la esencia y parte primordial de un sistema operativo. Teóricamente se le conoce como el corazón del sistema

¹¹ Disponible en Internet: <<http://www.kde.org/whatiskde/>>

¹² Disponible en Internet: <<http://www.gnome.org/about/>>

¹³ Disponible en Internet: <<http://www.home.unix-ag.org/simon/penguin/>>

operativo. Su función es la de gestionar recursos, a través de servicios de llamada al sistema y de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo, lo que se conoce como multiplexado.

Actualmente Linux es un núcleo monolítico híbrido. A diferencia de los núcleos monolíticos tradicionales, los controladores de dispositivos y las extensiones al sistema operativo se pueden cargar y descargar fácilmente como módulos, mientras el sistema continúa funcionando sin interrupciones, de igual forma los controladores pueden ser detenidos momentáneamente por actividades más importante bajo ciertas condiciones. Esta habilidad fue agregada para gestionar correctamente interrupciones de hardware, y para mejorar el soporte de Multiprocesamiento Simétrico.

Anterior al desarrollo de la serie 2.6 del núcleo, existieron dos tipos de versiones:

- **Versión de producción:** Se catalogaba como la versión estable que existía hasta la fecha y era el resultado final de las versiones de desarrollo o experimentales.

Los desarrolladores del núcleo al momento de tener una versión estable, lanzaban una nueva versión, considerada de producción ó estable. Desde ese momento esa versión era la que se debía utilizar para uso normal del sistema, ya que era la que se consideraba más estable y libre de fallos.

- **Versión de desarrollo:** Esta versión era de carácter experimental y la utilizaban los desarrolladores para, comprobar, programar y verificar nuevas características, correcciones, etc., antes de dar a conocer una versión de producción. Los núcleos usados para estas versiones, eran inestables y no eran recomendables para su uso en ambientes de producción.

Estas dos versiones del Kernel cumplían con una nomenclatura numérica especial. Se enumeraban con 3 números, de la siguiente forma: AA.BB.CC, donde:

- *AA*: Representa la serie o versión principal del núcleo. El máximo valor numérico posible en este primer campo es 1 o 2 ya que esos son los valores de la versión principal del núcleo. Este número solo cambiaba si el Kernel sufría un cambio muy importante.
- *BB*: En este campo se reconocía si la versión era de desarrollo ó de producción. Los valores impares, significaban que era de desarrollo, los pares, de producción.
- *CC*: Era el campo dedicado a indicarnos si existían nuevas revisiones a los fallos de programación dentro de una versión.

La serie **2.6** del núcleo trajo las siguientes modificaciones al sistema de numeración y al modelo de desarrollo: las versiones se enumeran con 4 dígitos de la siguiente forma: AA.BB.CC.DD y los campos de las versiones de producción y desarrollo han desaparecido.

- *AA*: Serie o versión principal del núcleo.
- *BB*: Revisión principal del núcleo. Pueden existir números pares e impares en este campo y se tratan de la misma manera.
- *CC*: Son las nuevas revisiones menores del núcleo. Este campo solo cambia si existen nuevos drivers y características soportadas.
- *DD*: Destinado a representar aquellas soluciones de programación o fallos de seguridad dentro de una revisión.

1.5.3. Distribuciones de Linux

Cuando Linus Torvalds desarrolló por primera vez Linux en agosto de 1991, el sistema operativo, básicamente, consistía en su núcleo y algunas herramientas GNU. Con la ayuda de otros, Linus añadió más y mejores herramientas y aplicaciones a su sistema original dando como resultado el soplo de vida a lo que hoy se conoce como el Sistema Operativo "LINUX".

Con el tiempo, personas, estudiantes universitarios y empresas del sector empezaron a crear gracias a su tipo de licencia, distribuciones de Linux con su propia selección de paquetes del ya existente núcleo de Linus, es de aquí donde nació el concepto de "distribución".

Hoy en día, la creación y venta de las distribuciones de Linux genera un multimillonario mercado de dólares. Existen distribuciones que están soportadas comercialmente, como Fedora (Red Hat), OpenSUSE (Novel), Ubuntu (Canonical Ltda.), Mandriva, y distribuciones mantenidas por la comunidad como Debian y Gentoo. Aunque hay otras distribuciones que no están relacionadas con alguna empresa o comunidad, como es el caso de Slackware. Linux también se puede descargar desde cualquier número de empresas y particulares, así mismo existen distribuciones para todos los tipos y para prácticamente cualquier tipo de sistema de cómputo.

Linux se refiere estrictamente al núcleo Linux, pero es comúnmente utilizado para describir un sistema operativo tipo Unix, que utiliza primordialmente filosofía y metodologías libres (también conocido como GNU/Linux) y que está formado mediante la combinación del núcleo Linux con las bibliotecas y herramientas del proyecto GNU. Además del núcleo Linux, las distribuciones incluyen habitualmente estas bibliotecas y herramientas y el sistema de ventanas X Window System, es en este momento cuando se le empieza a hablar de una distribución GNU/Linux.

Algunas de las distribuciones más populares son:

- **Red Hat** es una empresa dedicada al software libre, también conocida en el mundo Linux como “Sombrero Rojo” es una de los más grandes e importante proveedora, distribuidora y promotora de Linux. Esta empresa es famosa por los diferentes esfuerzos orientados a apoyar el movimiento del software libre y no sólo trabajan en el desarrollo de una de las distribuciones más populares de Linux, sino también en la comercialización de diferentes productos y servicios basados en software de código abierto. La empresa fue fundada en el año de 1995 y su sede principal queda en Raleigh, Carolina del Norte con oficinas en todo el mundo. La compañía tiene como producto bandera el sistema operativo Red Hat Enterprise Linux (RHEL), teniendo variantes como lo son: Mandriva Linux, Yellow Dog Linux (sólo para PowerPC), y CentOS (compilada a partir de las fuentes de Red Hat). En la Figura 5 se puede apreciar el logo que representa a la mencionada distribución.



Figura No 5. Logotipo Distribución RED HAT¹⁴

- **CentOS** es una distribución tipo empresarial de Linux¹⁵, la cual basa su código fuente en Red Hat, una de las distribuciones más antiguas de Linux que existe. CentOS es compilada por voluntarios alrededor el mundo los cuales usan su código fuente para generar un producto “Libre” para el público en general, con la salvedad que no es asistido ni mantenido por la Empresa Red Hat. Como la gran mayoría de las derivaciones del sombrero

¹⁴ Disponible en Internet: <<http://www.redhat.com/about/>>

¹⁵ Disponible en Internet: <<http://www.centos.org/>>

rojo, CentOS utiliza el gestor de paquetes YUM tanto para instalar como para actualizar su núcleo o Kernel. En la Figura 6 se puede apreciar el logo que representa a la mencionada distribución.



Figura No 6. Logotipo Distribución Centos¹⁶

- El Proyecto **Fedora** es una asociación global de miembros de la comunidad del software libre el cual no solo busca incluir software libre y de código abierto, sino “ser el líder en ese ámbito tecnológico”¹⁷. El Proyecto Fedora está patrocinado por Red Hat, empresa que invierte tanto infraestructura como recursos para fomentar la colaboración e incubar nuevas e innovadoras tecnologías. El fin de este proyecto y de dicha colaboración es el de integrar en los productos Red Hat todas las tecnologías que resulten del continuo avance de la distribución. Dichos avances son desarrollados en Fedora y producidos bajo una licencia libre y de código abierto, desde su inicio, por lo que otras comunidades de software libre tienen la posibilidad de estudiar, adoptar y modificar.

Recientemente, la comunidad Fedora ha prosperado, y la distribución tiene la reputación de ser una distribución completamente abierta, enfocada en la innovación y orientada al trabajo en grupo con las comunidades Linux en general. En la siguiente figura se puede apreciar el logo que representa a la mencionada distribución.

¹⁶ Disponible en Internet: <<http://www.centos.org>>

¹⁷ Disponible en versión HTML en Internet: <<http://interviews.slashdot.org/article.pl?sid=06/08/17/177220>>

Figura No 7. Logotipo Distribución Fedora¹⁸

- El Proyecto **Debian** es “una asociación de personas que han hecho causa común para crear un sistema operativo (SO) libre”¹⁹. El sistema se le denomina Debian GNU/Linux, o simplemente Debian.

Los sistemas Debian actualmente usan el núcleo de Linux, sin embargo, se está trabajando para ofrecer Debian con otros núcleos, en especial con el Hurd. El Hurd es primordialmente “una colección de protocolos que formalizan cómo los diferentes componentes de un computador pueden interactuar”²⁰. Hurd especifica cómo los protocolos están diseñados para reducir los requisitos de confianza mutua que deben tener los actores de un sistema de cómputo y permitir así un sistema más extensible. Estos incluyen definiciones de interfaz para manipular los archivos y directorios y para resolver la ruta de los nombres, lo que permite cualquier proceso en la implementación de un sistema de archivos. El único requisito es que se tenga acceso al sistema de almacenamiento de respaldo y que el principal esté conectado al nodo donde se encuentran los archivos.

El Hurd es también un conjunto de servidores que ejecutan estos protocolos y son precisamente los primeros los encargados de los sistemas de ficheros, protocolos de red y autenticación. Estos servidores se ejecutan sobre el microkernel Mach²¹ y el uso de mecanismo de Mach IPC para la transferencia de información.

¹⁸ Disponible en Internet: < <http://fedoraproject.org> >

¹⁹ Disponible en Internet: <<http://www.debian.org/intro/about>>

²⁰ Disponible en versión HTML en Internet: <http://www.gnu.org/software/hurd/hurd/what_is_the_gnu_hurd.html>

²¹ J. BRADLEY CHEN y BRIAN N. BERSHAD. The impact of operating system structure on memory system performance, ACM Symposium on Operating Systems Principles - Proceedings of the fourteenth. Asheville, North Carolina, United States, 1994. p. 120 – 133.

Debian no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuir comercialmente este software mientras se respete su licencia. En la Figura 8 se puede apreciar el logo que representa a la mencionada distribución.



Figura No 8. Logotipo Distribución Debian²²

- El proyecto **OpenSUSE** es “un programa comunitario a nivel internacional patrocinado por Novell que promueve el uso de Linux en todas partes”²³. El objetivo de dicho proyecto es el de crear y distribuir la versión de Linux más utilizable del mundo. OpenSUSE es una distribución y proyecto libre auspiciado por Novell y AMD para el desarrollo y mantenimiento de un sistema operativo basado en Linux. En el año 2004 Novell adquiere SUSE Linux y lanza SUSE Linux Professional como un proyecto completamente de código abierto, involucrando a la comunidad en el proceso de desarrollo. La primera versión fue la beta de SUSE Linux 10.0 y la última versión estable es la OpenSUSE 11.2 liberada en el mes de noviembre de 2009. o "yo soy porque nosotros somos".



Figura No 9. Logotipo Distribución OpenSUSE²⁴

²² Disponible en Internet: <<http://www.debian.org>>

²³ Disponible en versión HTML en Internet: <http://es.opensuse.org/Bienvenidos_a_openSUSE.org>

²⁴ Disponible en Internet: <http://en.opensuse.org/Welcome_to_openSUSE.org>

- **Ubuntu** es una ideología étnica Sud-Africana que se enfoca en la gente las alianzas y las relaciones con los demás. La palabra proviene de los idiomas Zulu y Xhosa. Ubuntu es visto como un concepto tradicional africano, es considerado como uno de los principios fundamentales de la nueva República de Sudáfrica y está conectado con la idea del renacimiento africano.

Una traducción aproximada del principio de Ubuntu es "humanidad hacia otros" o "yo soy porque nosotros somos". Otra traducción podría ser: "la creencia en un enlace universal de compartir que conecta a toda la humanidad".

Como plataforma basada en software libre, el sistema operativo Ubuntu trae el espíritu de Ubuntu al mundo del software y se considera una distribución Linux que ofrece un sistema operativo enfocado a computadoras de escritorio y servidores. Se le considera una de las más importantes distribuciones de GNU/Linux a nivel mundial. Basada en Debian GNU/Linux, Ubuntu concentra su objetivo en la facilidad y la libertad de uso, la facilidad de instalación y los lanzamientos regulares (cada 6 meses). Su patrocinador oficial es Canonical Ltd., empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth.



Figura No 10. Logotipo Distribución Ubuntu²⁵

1.5.4. Versiones LIVE

²⁵ Disponible en Internet: <<http://www.ubuntu.com>>

Un "LiveCD", CD vivo o CD autónomo no es más que una distribución de Linux que funciona sin necesidad de ser instalada en un computador, para lo cual usa la memoria RAM como disco duro virtual y el propio medio como sistema de archivos. Este tipo de distribuciones usan sistema operativo basado en el núcleo Linux, BeOS, FreeBSD, Minix, Solaris, OS/2 o incluso Microsoft Windows aunque es ilegal distribuir un LiveCD de este último sistema operativo.

El primer Live CD Linux fue Yggdrasil Linux en 1995, seguido de DemoLinux en el año 2000. El apogeo de los LiveCD se da en el año 2003 con la distribución alemana Knoppix, basada en Debían.

Estas distribuciones son solo para demostraciones y pruebas, ya que al trabajar principalmente con la memoria RAM una vez apagado el computador, todo lo trabajado se pierde siendo esta su principal desventaja.

Para usar un Live CD solo es necesita bajar de internet la distribución deseada y configurar la computadora para que arranque desde la unidad de CD o DVD, después de reiniciar el equipo con el Live CD dentro de ella, este se iniciará automáticamente. Aunque no todas las distribuciones "LiveCD" vienen también con la opción de instalación, algunas nos dejan instalarla una vez después de probada.

Si se desea probar alguna de las distribuciones "LiveCD" disponibles se puede visitar la página: <http://www.livecdlist.com/> donde se encuentran todas las posibles hasta la fecha.

Capítulo No 2

INSTALACIÓN DE UBUNTU SERVER



Capítulo No 2. INSTALACIÓN DE UBUNTU SERVER

A continuación se presentará la instalación de Ubuntu Server en la versión 8.04, esta versión se escoge por ser LTS o de larga duración para soporte por Canonical.

Lo primero que se debe hacer es configurar la BIOS del equipo para que arranque desde el CD ROM donde tenemos el CD de Ubuntu que deseamos instalar, paso después de este es el de reiniciar el sistema con lo que nos mostrara una ventana como esta:

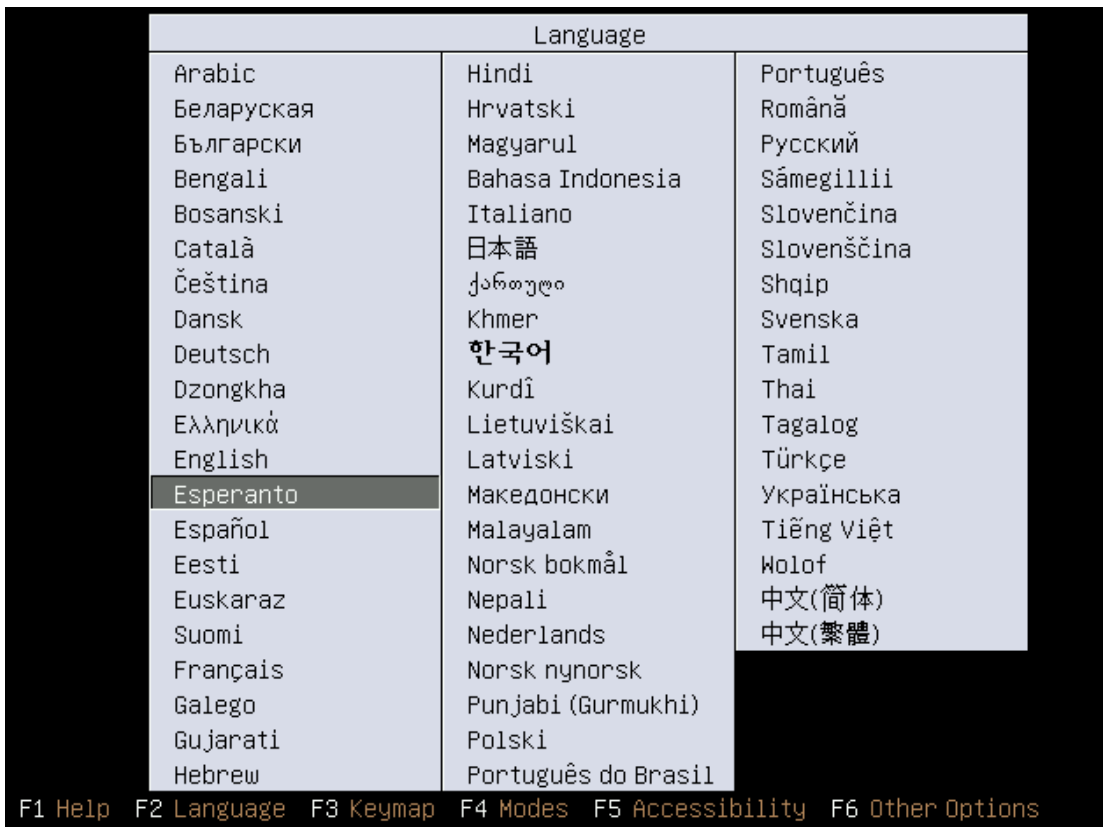


Figura No 11. Proceso de Instalación Selección de idioma

Por defecto está el idioma Esperanto por lo que escogemos el idioma desea que en nuestro caso es Español y presionamos Enter.

En estos momentos se muestra el menú de instalación con las siguientes opciones:

- Instalar Ubuntu: sirve para iniciar el proceso de instalación
- Verificar el CD en busca de defectos: Herramienta para la detección de anomalías en el CD de instalación de Ubuntu que repercutan en una instalación abortada antes del 100%.
- Recuperar un sistema dañado: Con esta función podremos recuperar una instalación con errores causados por usuarios u otro factor externo.
- Análisis de memoria: Herramienta bastante útil que nos da la posibilidad de analizar la memoria RAM del equipo para encontrar imperfecciones en las memorias si las hubiese y evitar pérdida de datos en momentos de ejecución del sistema.
- Arrancar desde el primer disco duro: En este punto el menú de opciones nos deja la vía libre para arrancar nuestro sistema por el disco duro local que tenga un sistema operativo instalado.



Figura No 12. Opciones de inicio de instalación

Para seguir con el proceso de instalación le damos Enter a la primera opción lo que carga el núcleo o Kernel de Linux como se muestra en la figura. Después de cargado el núcleo seguimos con la instalación escogiendo el lenguaje a utilizar en el sistema tomando como referencia el país donde nos encontremos, en nuestro caso escogemos Colombia como se muestra en la figura.

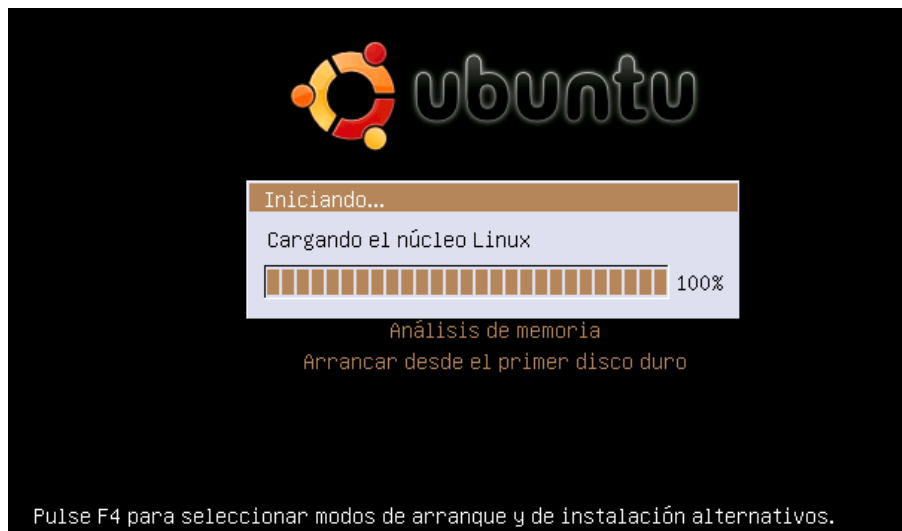


Figura No 13. Carga del núcleo o Kernel de Linux

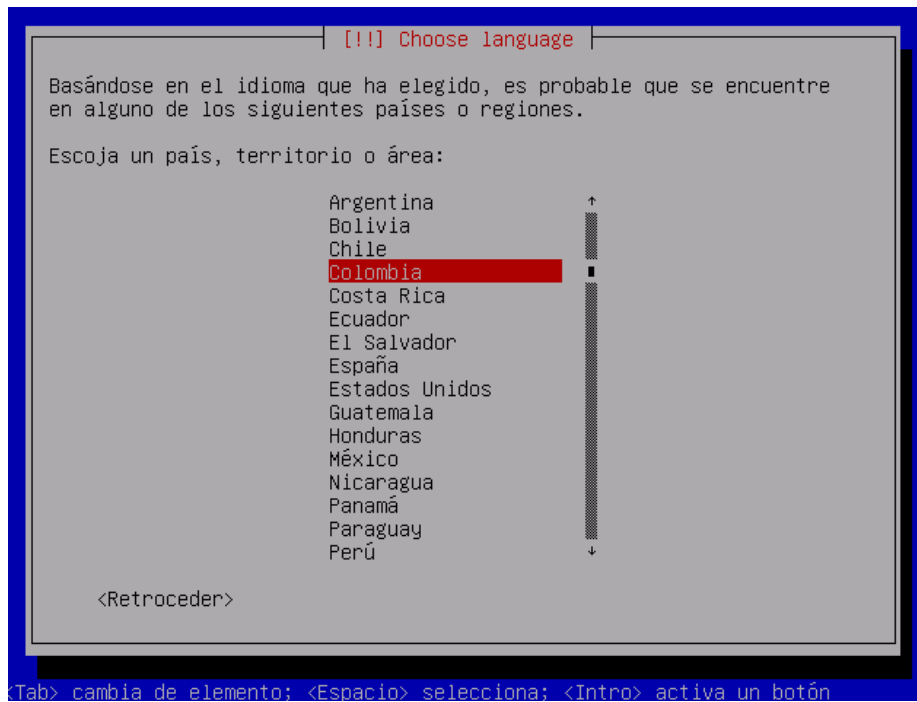


Figura No 14. Selección del idioma del sistema

Seguidamente el programa de instalación carga los componentes necesarios para seguir con el proceso y nos da la opción de configurar la red a lo cual decimos que no se hará en este momento ya que mas adelante se explicara este proceso con detalle.

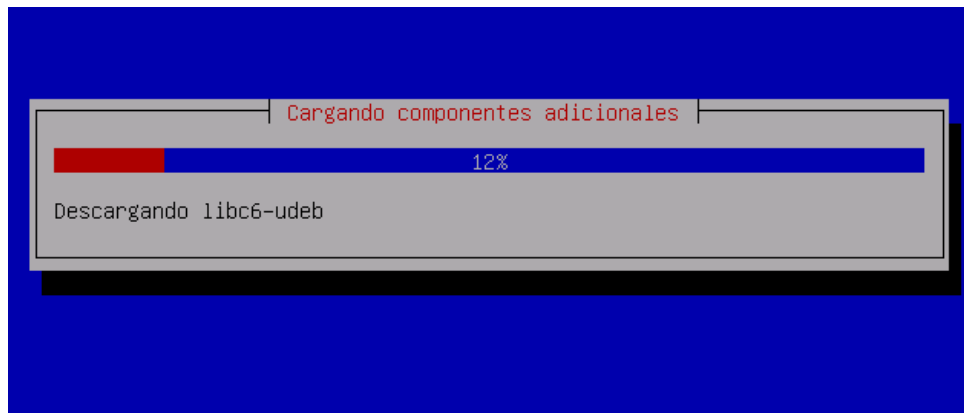


Figura No 15. Proceso de Instalación - Carga de componentes adicionales

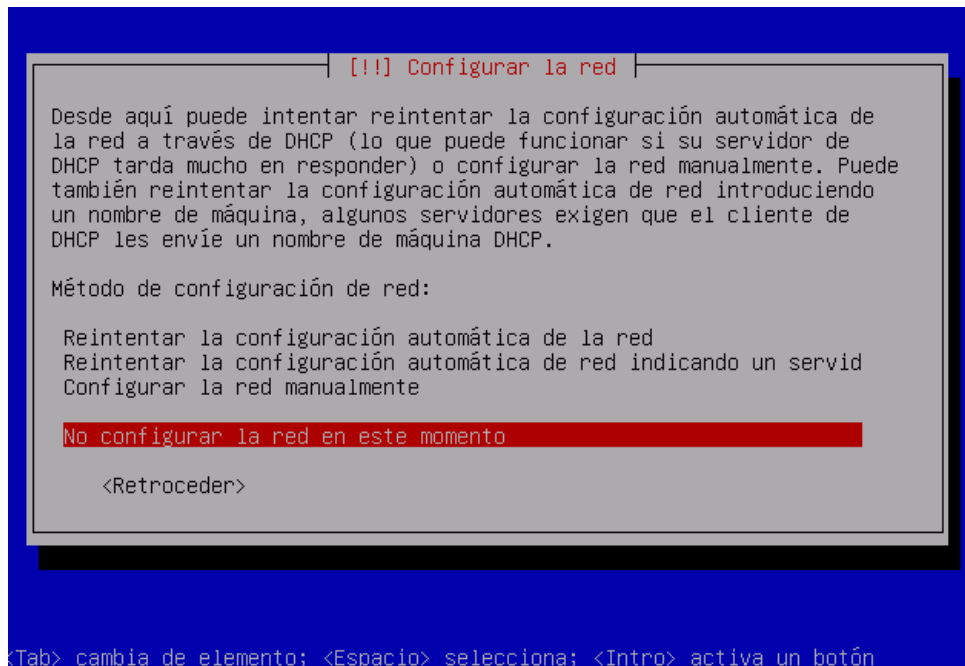


Figura No 16. Proceso de Instalación - Configuración de red

En el siguiente paso establecemos el nombre de la máquina como queremos que sea reconocida en una red, Ubuntu en nuestro caso, pero puede ser cualquier nombre que se le quiera dar.

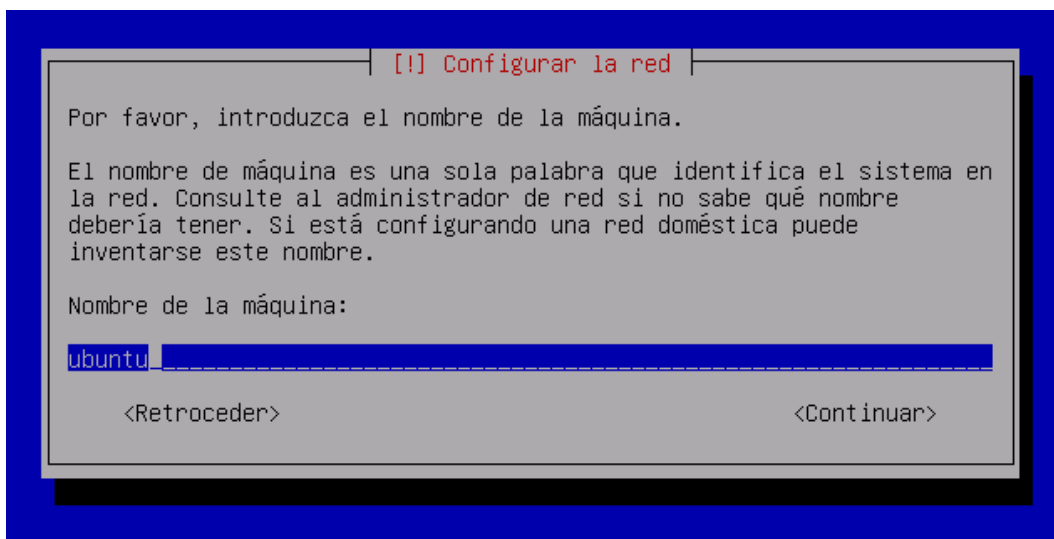


Figura No 17. Proceso de Instalación – Asignación del nombre de la máquina

Acto seguido empezamos el proceso de particionamiento de discos en el que escogeremos la opción “manual” para personalizar nuestro particionado en los discos locales.

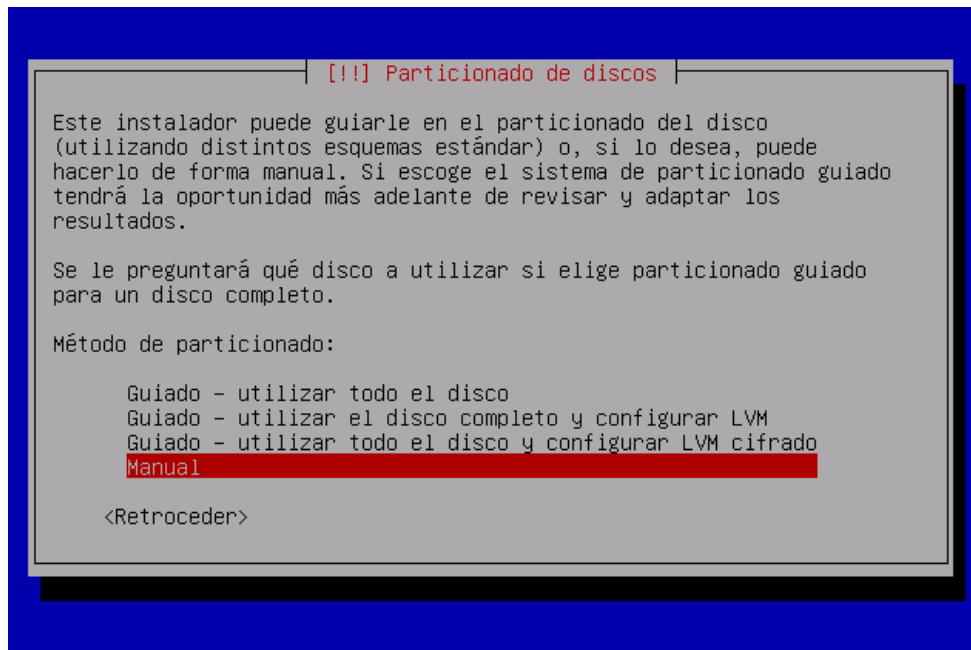


Figura No 18. Proceso de Instalación - Particionamiento de discos

Dependiendo de la cantidad de discos que tengamos en nuestro equipo variaran las opciones de particionado. En este libro se muestra como sería la instalación en una máquina que tenga un solo disco duro. Las opciones que muestra la instalación son:

- Particionado guiado: Opción que sirve para darle la oportunidad al programa de instalación que el mismo realice las particiones necesarias en los discos locales según características predeterminadas.
- Ayuda del particionado: Con esta opción podremos ver ayuda sobre el particionado en caso de necesitarla.

Después de estas opciones nos aparecerá el disco o los discos locales de la maquina disponibles para realizar las particiones por el usuario. Aparecen como opciones también: deshacer los cambios realizados a las particiones, lo que borra todos los cambios realizados si ese fuera el caso. De igual forma encontramos la opción de finalizar el particionado y escribir los cambios en el disco, lo que hace esta opción es afirmar los cambios realizados por el usuario en el disco o los discos disponibles.

A continuación y teniendo en cuenta las opciones presentadas anteriormente escogeremos la que nos muestra el disco disponible y daremos Enter sobre ella.

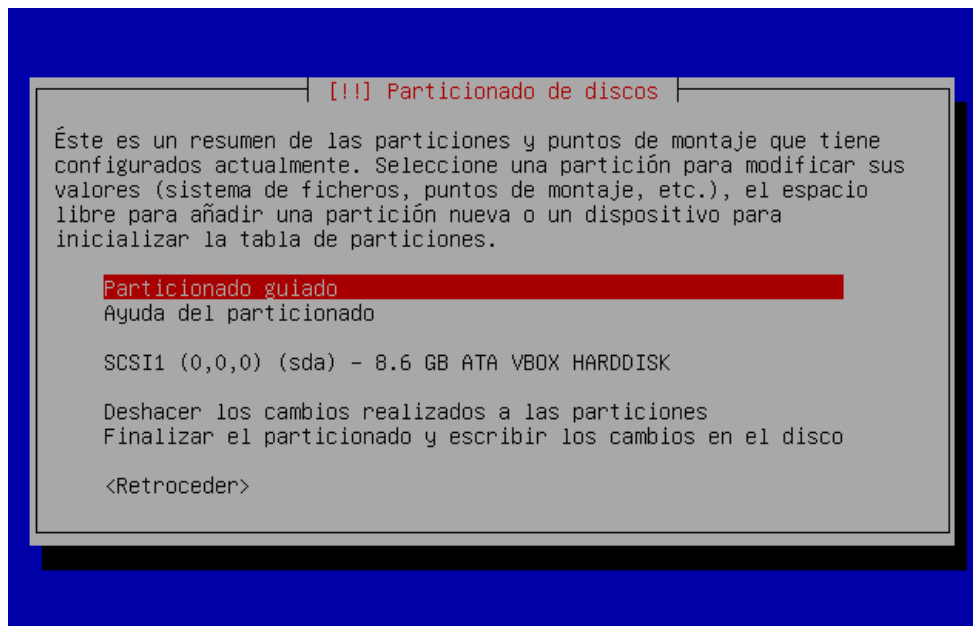


Figura No 19. Proceso de Instalación - Discos disponibles

Inmediatamente se nos muestra un aviso que nos dice que hemos decidido utilizar el dispositivo completo y que si queremos seguir debemos tener en cuenta que se borrarán todas las particiones en dicho disco, para esto decimos que SI deseamos crear un tabla de particiones nueva en el dispositivo de almacenamiento. Se nos

deja claro que podremos deshacer estos cambios más adelante si así lo deseamos.

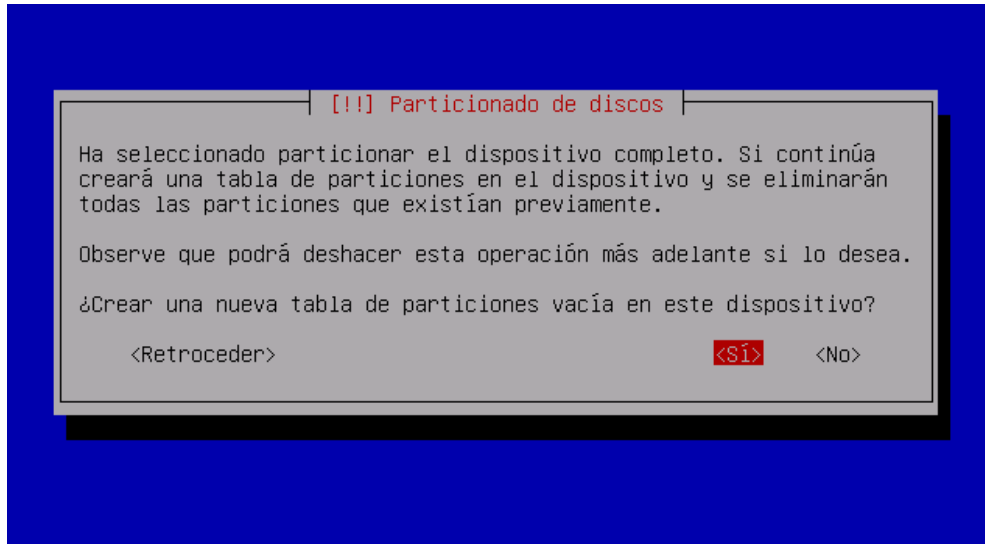


Figura No 20. Proceso de Instalación – Particionamiento de Discos 1

Al darle Enter a la opción SI, el programa de instalación nos muestra el espacio en el disco del tamaño de su capacidad total y nos dice que está libre y listo para ser usado como se muestra en la figura.

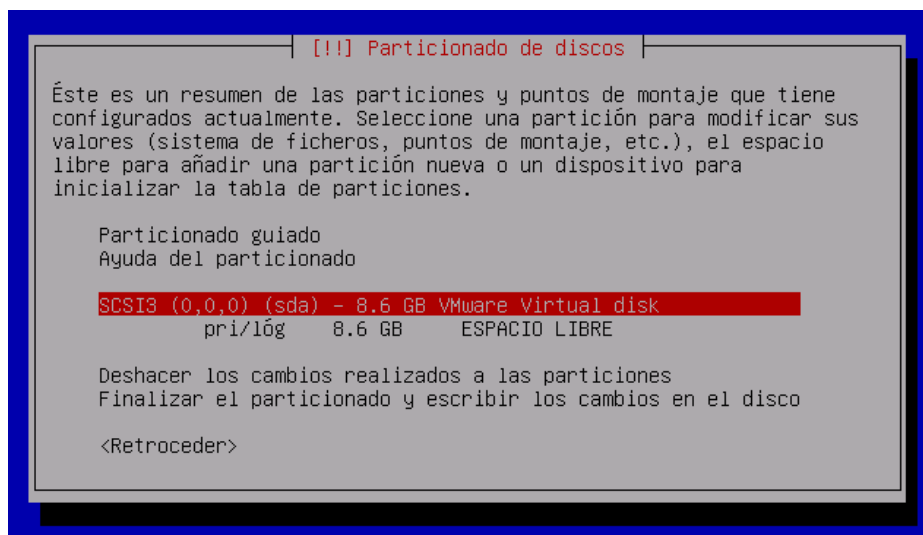


Figura No 21. Proceso de Instalación – Particionamiento de Discos 2

Escogemos ese espacio libre y damos Enter. Se nos da la oportunidad de usar el espacio libre de las siguientes maneras:

- Crear una partición nueva: Sirve para hacer particiones nuevas en el espacio libre de disco disponible, en nuestro caso, todo el disco.
- Particionar de forma automática el espacio libre: Como su nombre lo indica realiza un particionado automático del espacio libre que tiene el disco.
- Mostrar información de Cilindros/Cabezas/Sectores: Describe de donde a donde empieza y termina el espacio libre disponible en el disco.

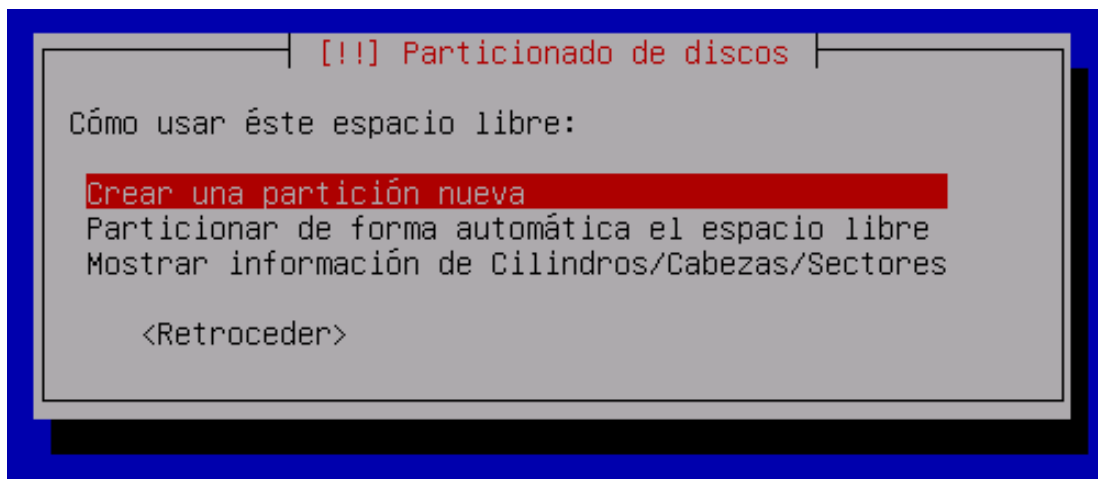


Figura No 22. Proceso de Instalación – Particionamiento de Discos 3

De estas opciones escogeremos la opción de Crear partición nueva, lo que nos muestra automáticamente el tamaño máximo a utilizar en el disco y el tamaño de la nueva partición.

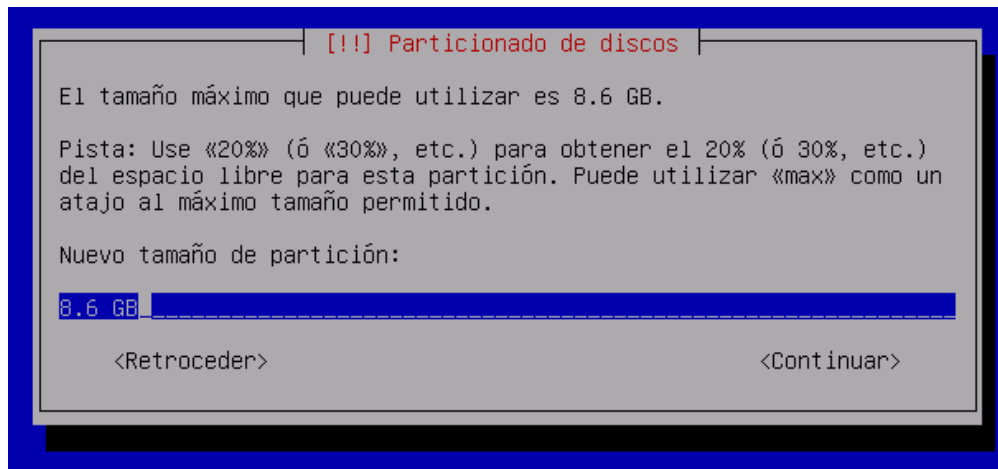


Figura No 23. Proceso de Instalación – Particionamiento de Discos 4

En estos momentos debemos tener en cuenta que para poder hacer satisfactoriamente un particionamiento manual se deben hacer como mínimo 2 particiones en el disco, la SWAP y la principal o “/” que es la que tiene el directorio raíz de Linux. Avanzaremos haciendo primero la SWAP recomendable del doble de la memoria RAM si tenemos menos de 1GB de RAM o de 1GB si tenemos más de este en nuestro equipo. La partición SAWP es para intercambio de en disco físico como soporte a la RAM mientras que la partición “/” es donde se creara el sistema de directorio de Linux.

Le damos entonces el tamaño a la nueva partición, 1 GB y le damos continuar.

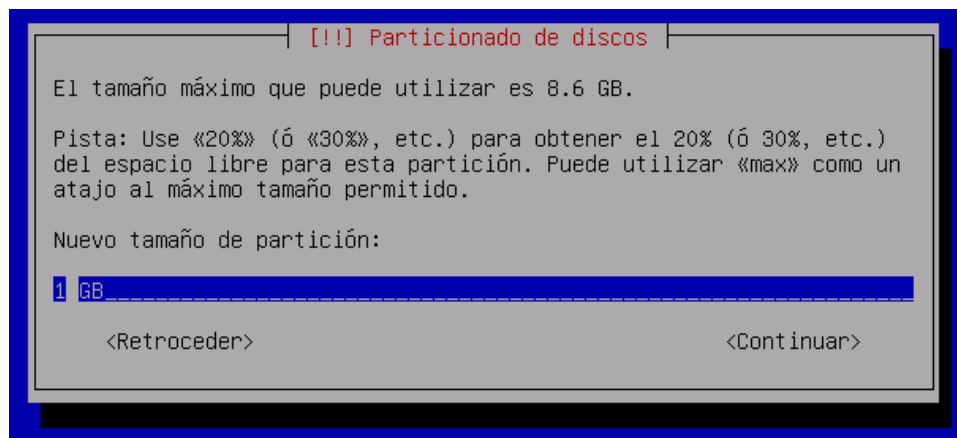


Figura No 24. Proceso de Instalación – Particionamiento de Discos 5

Escogemos que esta nueva partición sea primaria y damos Enter,

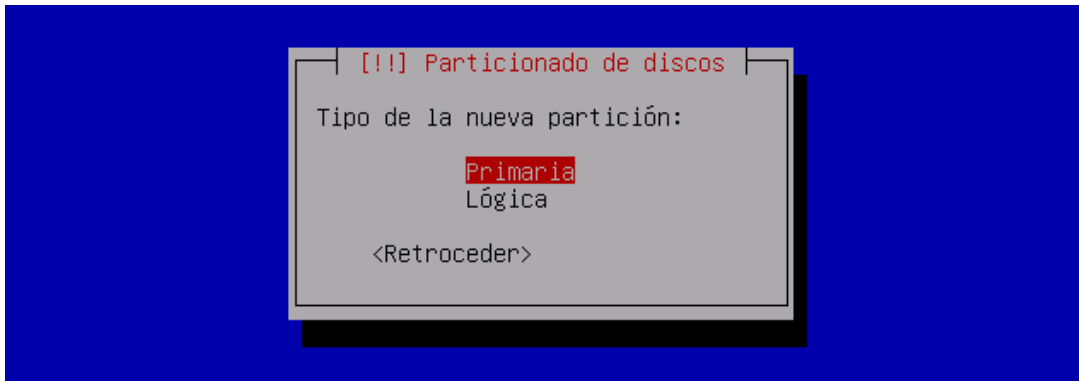


Figura No 25. Proceso de Instalación – Particionamiento de Discos 6

Luego que se ubique al principio del disco escogido para la instalación,

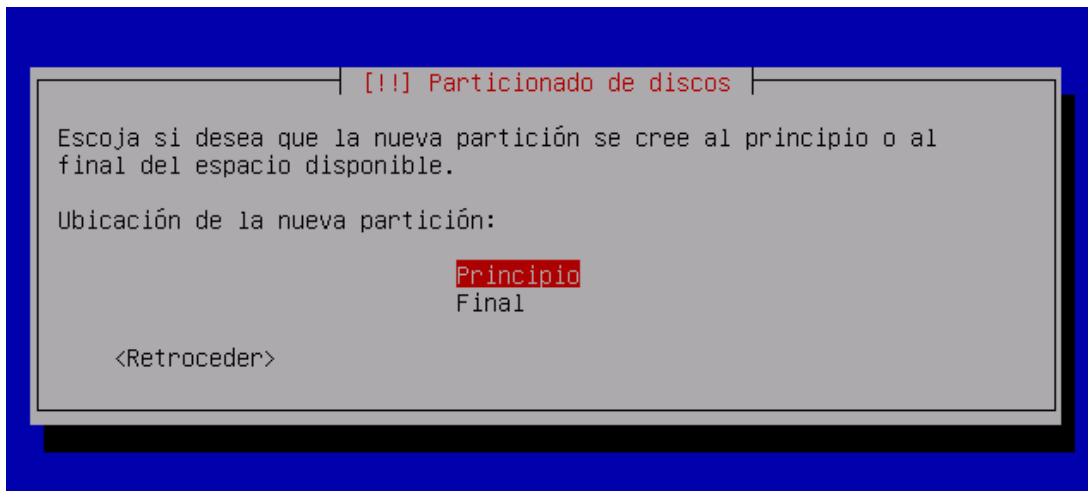


Figura No 26. Proceso de Instalación – Particionamiento de Discos 7

Al darle Enter crea la partición y nos muestra las opciones que tenemos para aplicar a esta partición. En esta nueva ventana escogemos la opción utilizar como y le damos Enter, lo que nos despliega todas las opciones de sistemas de archivos disponibles.

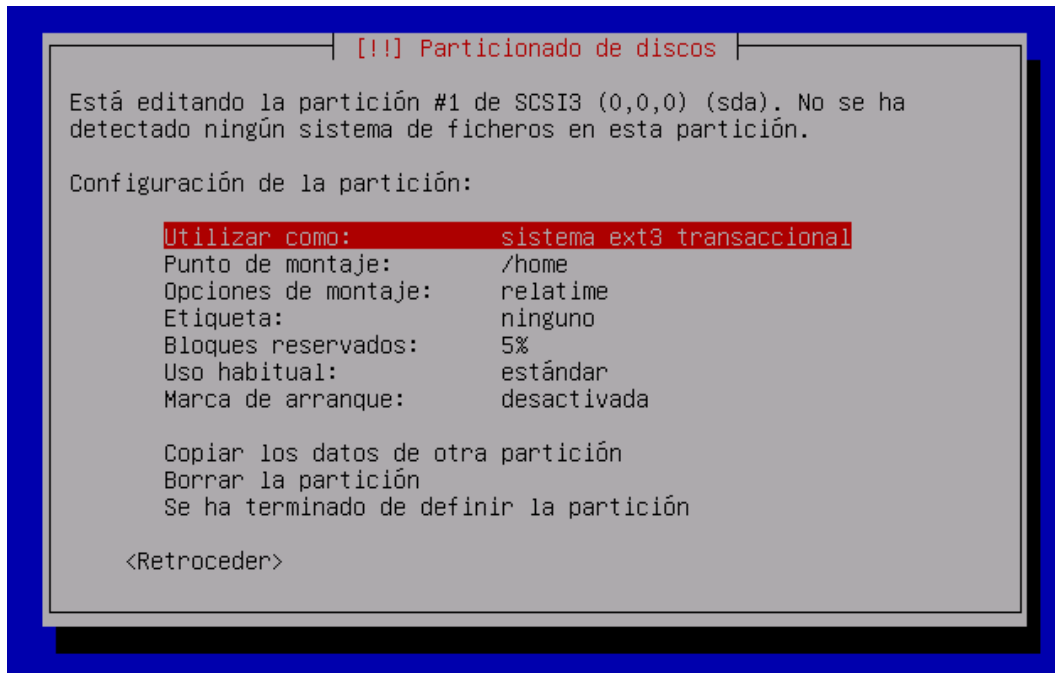


Figura No 27. Proceso de Instalación – Particionamiento de Discos 8

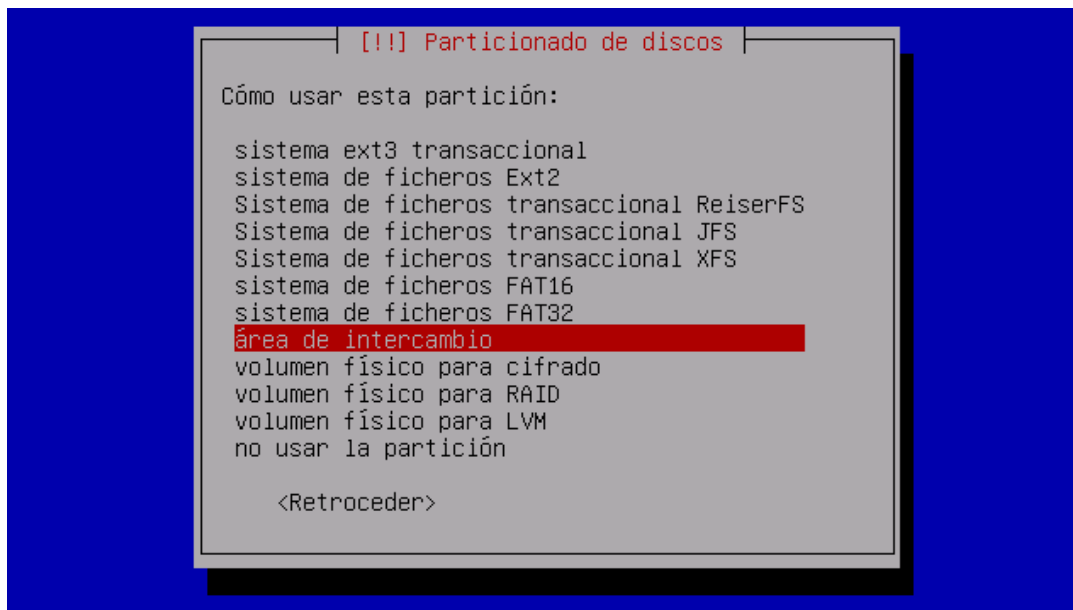
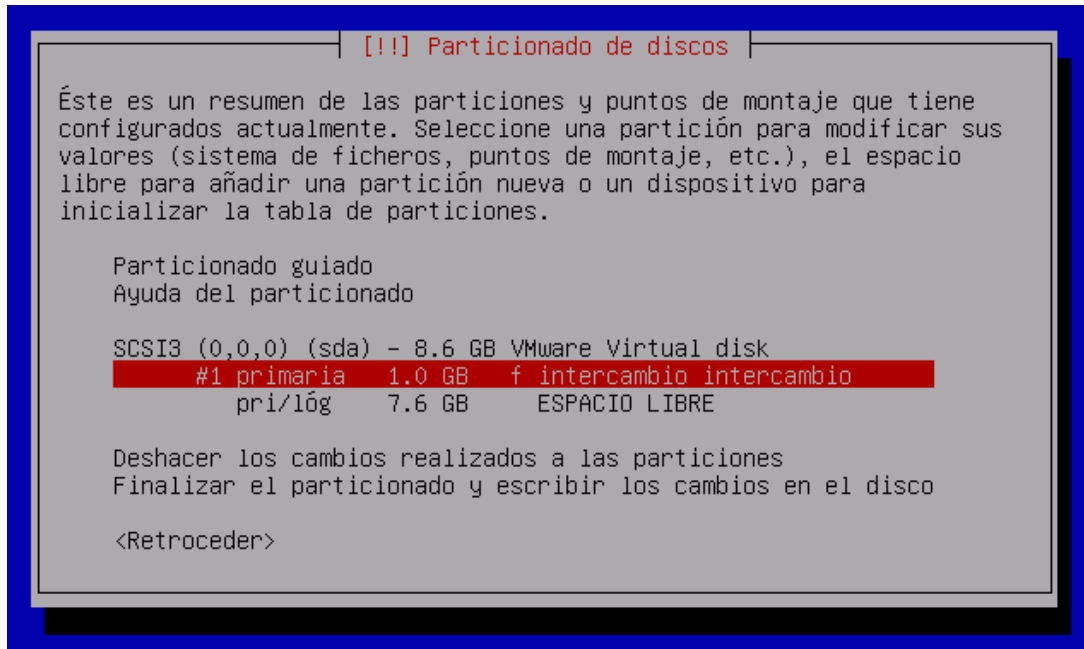


Figura No 28. Proceso de Instalación – Particionamiento de Discos 9

De todas ellas debemos tomar la que dice “área de intercambio” y dar Enter, de esta manera se crea la partición SWAP necesaria para el intercambio. Después debemos escoger la opción que dice que se ha terminado de definir la partición y de inmediato nos aparecerá que la partición fue creada con las especificaciones que se dieron.



```
[!!] Particionado de discos

Éste es un resumen de las particiones y puntos de montaje que tiene
configurados actualmente. Seleccione una partición para modificar sus
valores (sistema de ficheros, puntos de montaje, etc.), el espacio
libre para añadir una partición nueva o un dispositivo para
inicializar la tabla de particiones.

Particionado guiado
Ayuda del particionado

SCSI3 (0,0,0) (sda) - 8.6 GB VMware Virtual disk
#1 primaria 1.0 GB f intercambio intercambio
pri/lóg 7.6 GB ESPACIO LIBRE

Deshacer los cambios realizados a las particiones
Finalizar el particionado y escribir los cambios en el disco

<Retroceder>
```

Figura No 29. Proceso de Instalación – Particionamiento de Discos 10

Nótese que se le ha asignado una letra a la partición y que ya el sistema reconoce el tipo de archivos sobre la partición de disco creada.

Esa misma ventana nos muestra un espacio libre del tamaño del resto del disco el cual utilizaremos para crear la estructura de directorios en “/”. Para realizar esto simplemente escogemos ese espacio el cual está sin particionar dándole Enter y decimos que deseamos crear una partición nueva, del tamaño que se nos muestra el cual es el disponible del resto del disco que está libre y que sea primaria.

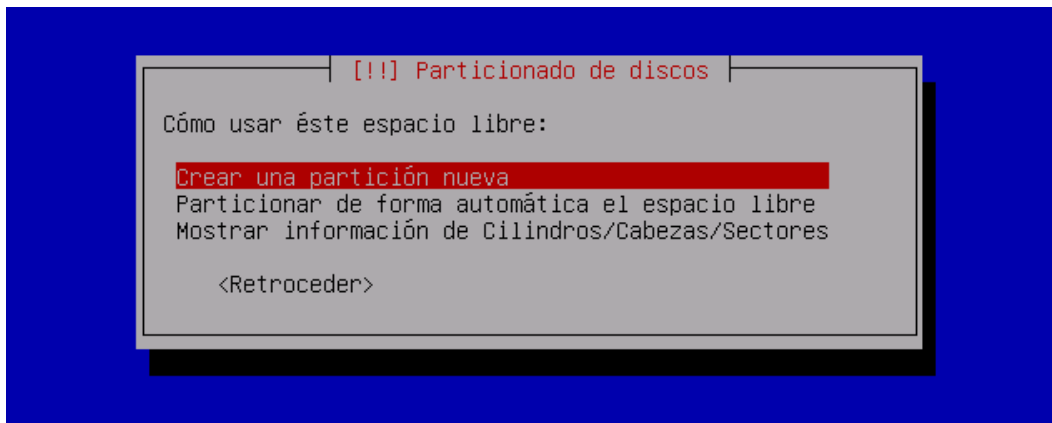


Figura No 30. Proceso de Instalación – Particionamiento de Discos 11

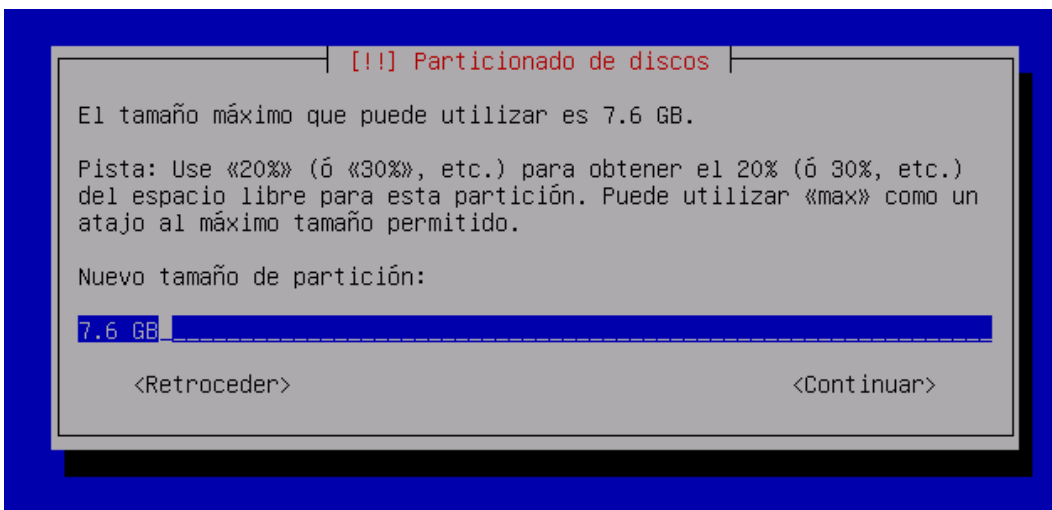


Figura No 31. Proceso de Instalación – Particionamiento de Discos 12

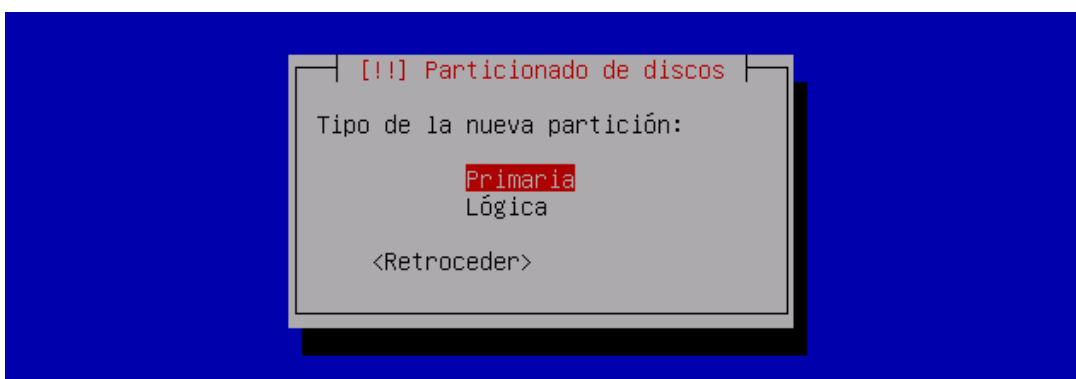


Figura No 32. Proceso de Instalación – Particionamiento de Discos 13

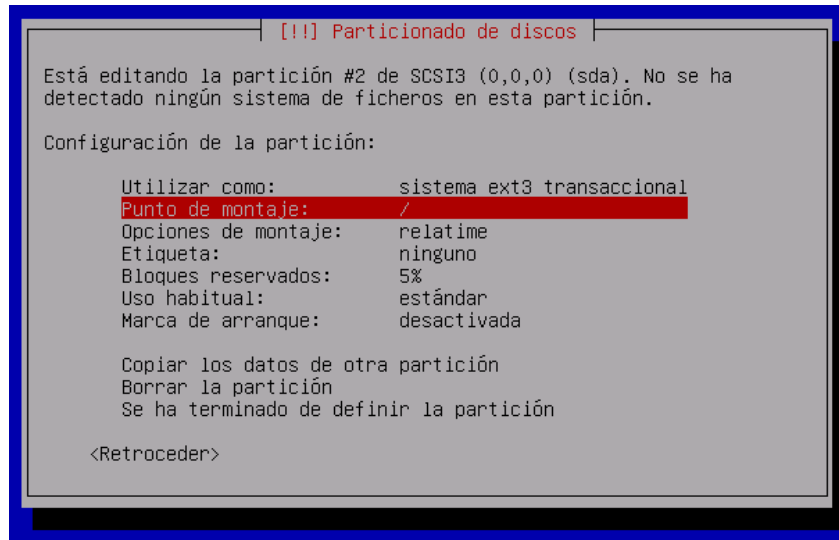


Figura No 33. Proceso de Instalación – Particionamiento de Discos 14

El punto de montaje será “/”, este se encuentra por defecto por lo que solo debemos escoger la opción de “se ha terminado de definir la partición”.

Así hemos terminado de crear las 2 particiones necesarias para poder instalar Linux en un equipo. Finalmente lo único que debemos hacer es decirle al programa de instalación que termine el particionado y escriba los cambios en el disco y en la siguiente ventana confirmamos los cambios hechos hasta ahora.

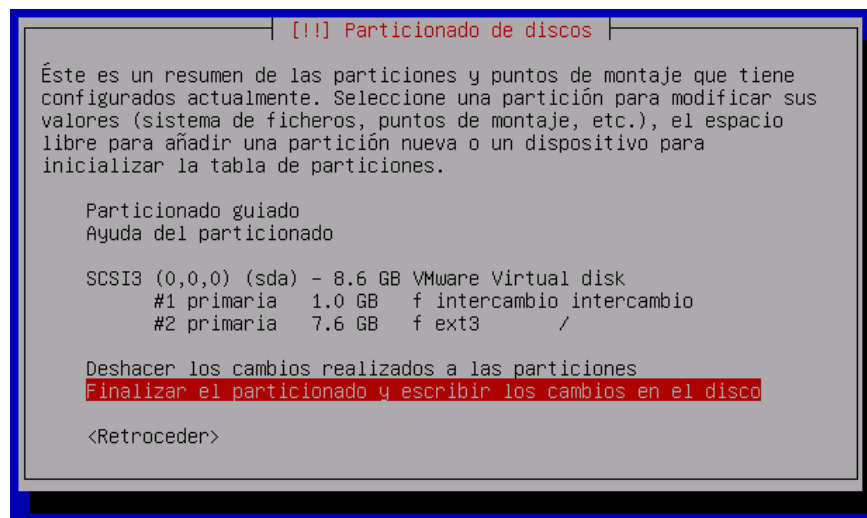


Figura No 34. Proceso de Instalación – Particionamiento de Discos 15

Se nos pide la confirmación de los cambios que se harán sobre el o los discos a lo que diremos que sí,

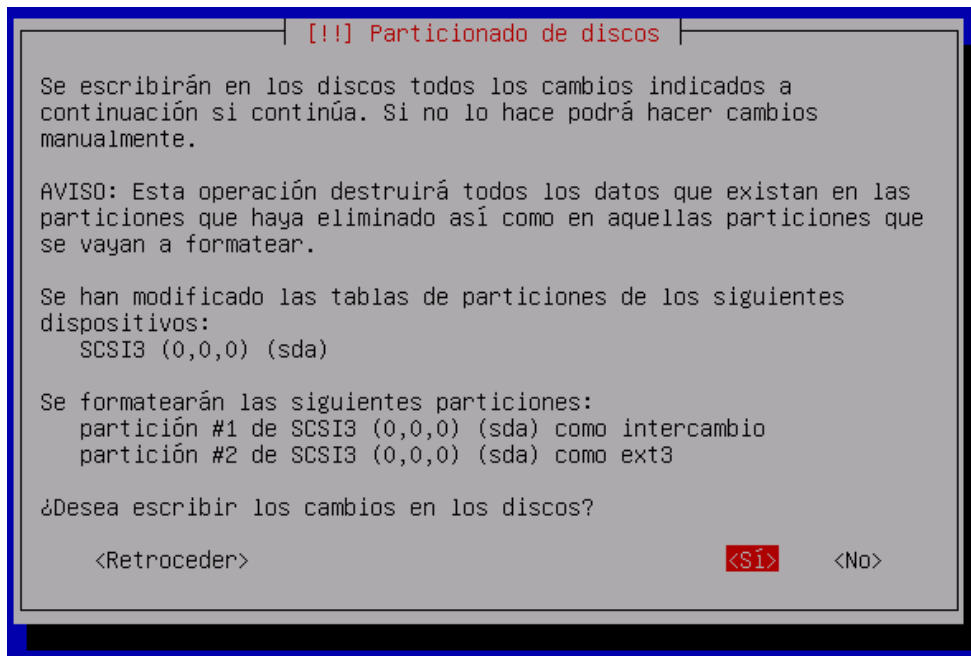


Figura No 35. Proceso de Instalación – Particionamiento de Discos 16

Automáticamente el programa formatea las particiones hechas y empieza el programa de instalación a copiar archivos.

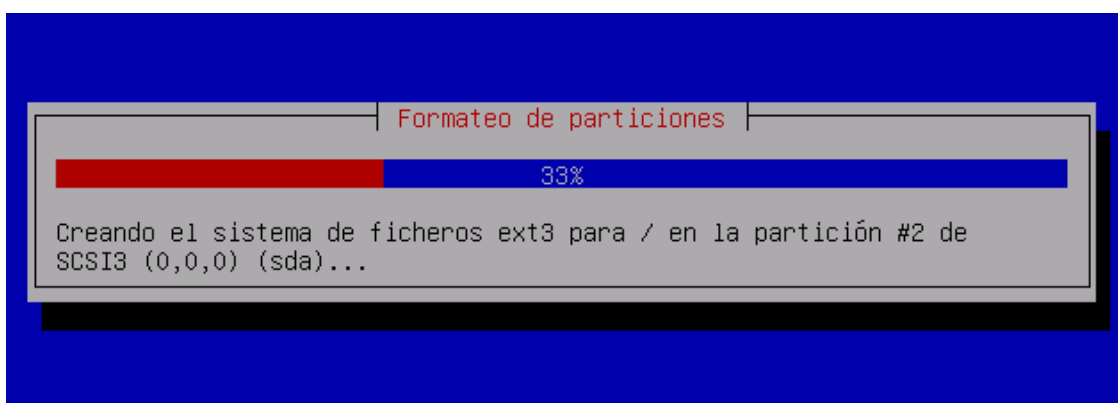


Figura No 36. Proceso de Instalación – Formateo de Particiones



Figura No 37. Proceso de Instalación – Particionamiento de Discos 17

Después de copiar los archivos y hacer la instalación de sistema, el programa de instalación pide introducir un nuevo usuario para tener acceso a la máquina, primero pide el nombre completo de dicho usuario y después el nik para entrar al sistema.

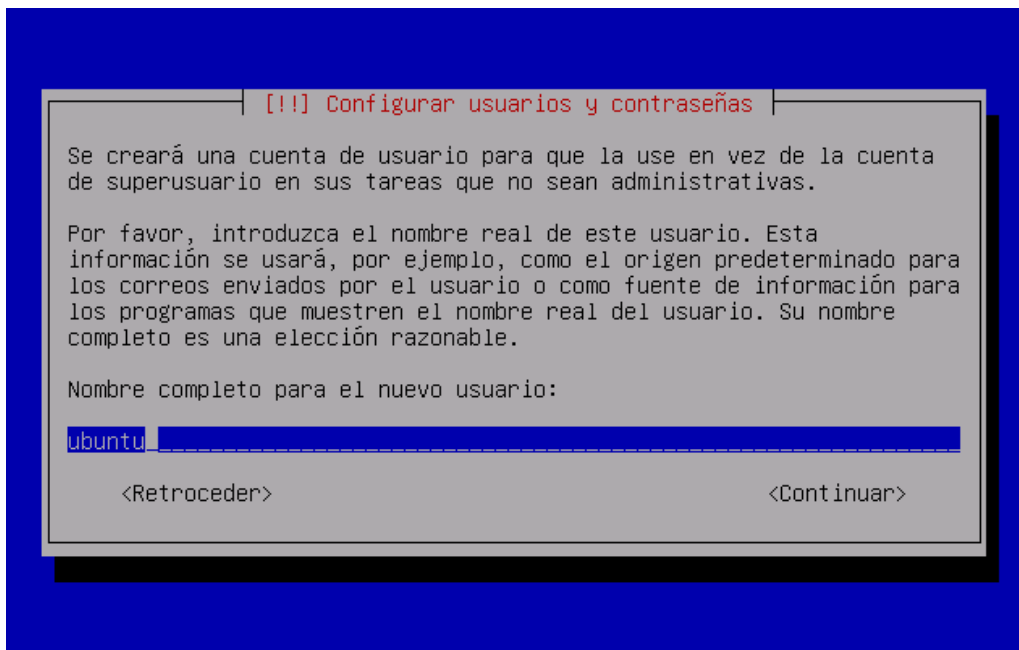


Figura No 38. Proceso de Instalación – Configuración usuarios y contraseñas 1

Seguidamente se pide la contraseña para el usuario que se dio, con los parámetros que se describen (Números, Letras, Mayúsculas y minúsculas) y en la siguiente ventana nos pide que repitamos la clave.

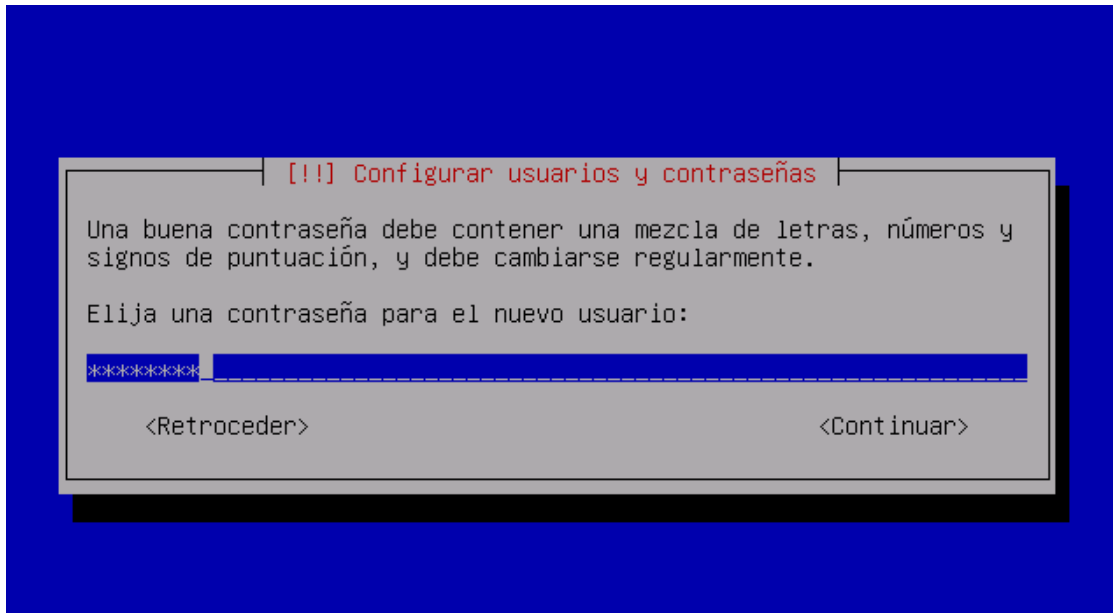


Figura No 39. Proceso de Instalación – Configuración usuarios y contraseñas 2

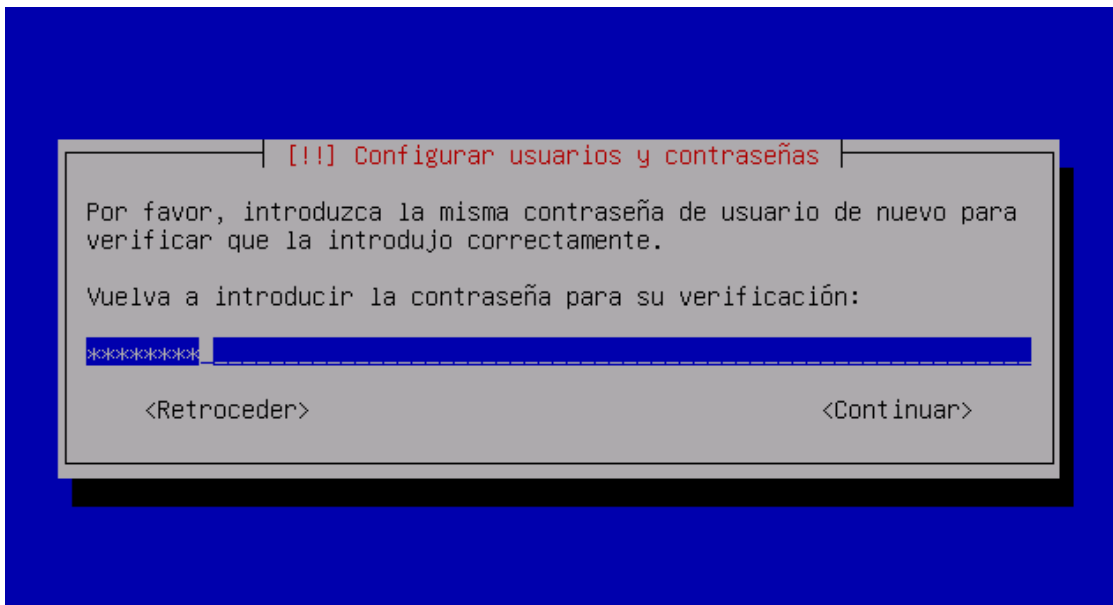


Figura No 40. Proceso de Instalación – Configuración usuarios y contraseñas 3

La ventana que sigue es la que permite configurar un servidor proxy para la conexión de la maquina a internet, en este punto le damos continuar y dejamos esa configuración para después que nuestro servidor este funcionando.

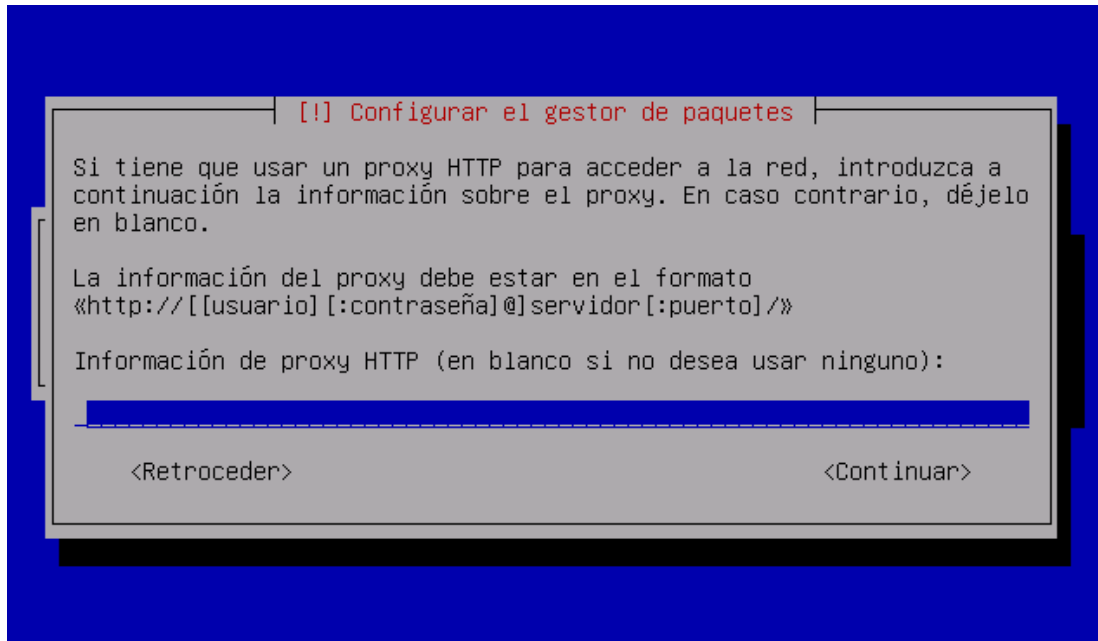


Figura No 41. Proceso de Instalación – Configuración del gestor de paquetes

El siguiente paso es instalar los servidores por defecto que la instalación de Ubuntu Server trae: DNS Server, LAMP Server, Mail Server, OpenSSH Server, PostgreSQL Server, Print Server, Samba File Server. Para esta opción de instalación vamos a dejar en blanco todas las casillas ya que vamos a instalar un sistema limpio y más adelante se instalaran los servicios concernientes a este libro por separado explicando su funcionamiento. Al darle Enter se instalan los paquetes necesarios para el funcionamiento del servidor con las opciones más básicas.

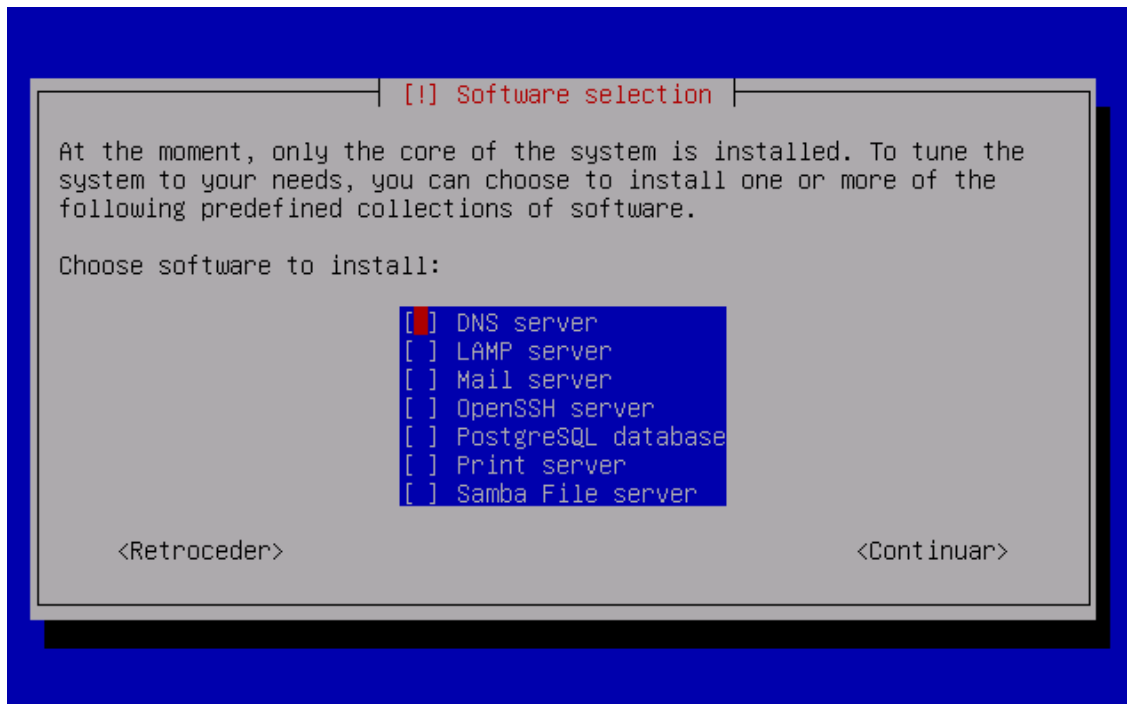


Figura No 42. Proceso de Instalación – Selección de Servicios

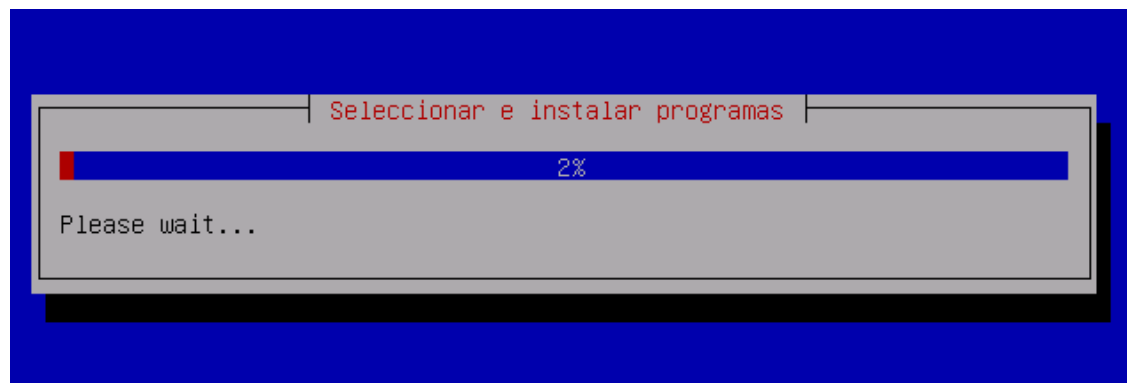


Figura No 43. Proceso de Instalación – Selección de Servicios

Después de la instalación nuestro sistema está listo para ser usado, solo le damos continuar y el sistema reinicia para su funcionamiento.

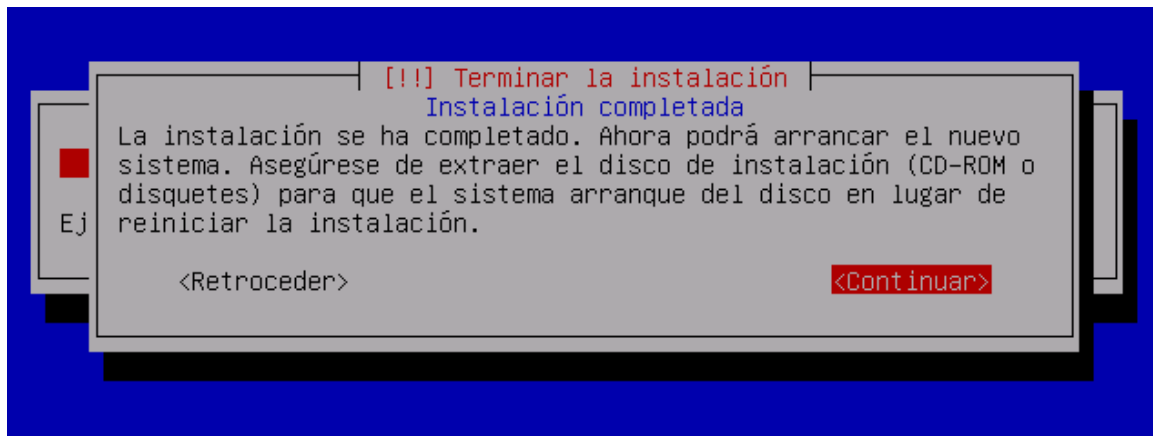


Figura No 44. Proceso de Instalación – Culminación 1

```
* Activating swap... [ OK ]
* Checking root file system...
fsck 1.40.8 (13-Mar-2008)
/dev/sda2: clean, 17437/463296 files, 136232/1851491 blocks [ OK ]

* Checking file systems...
fsck 1.40.8 (13-Mar-2008) [ OK ]

* Mounting local filesystems... [ OK ]
* Activating swapfile swap... [ OK ]
$Mounting securityfs on /sys/kernel/security: done.
Loading AppArmor profiles : done.
* Checking minimum space in /tmp... [ OK ]
* Skipping firewall: ufw (not enabled)... [ OK ]
* Configuring network interfaces... [ OK ]
* Setting up console font and keymap... [ OK ]
* Starting system log daemon... [ OK ]
* Starting kernel log daemon... [ OK ]
* Starting deferred execution scheduler atd [ OK ]
* Starting periodic command scheduler crond [ OK ]
* Running local boot scripts (/etc/rc.local) [ OK ]

Ubuntu 8.04.3 LTS ubuntu tty1
ubuntu login: _
```

Figura No 45. Proceso de Instalación – Culminación 2

Capítulo No 3

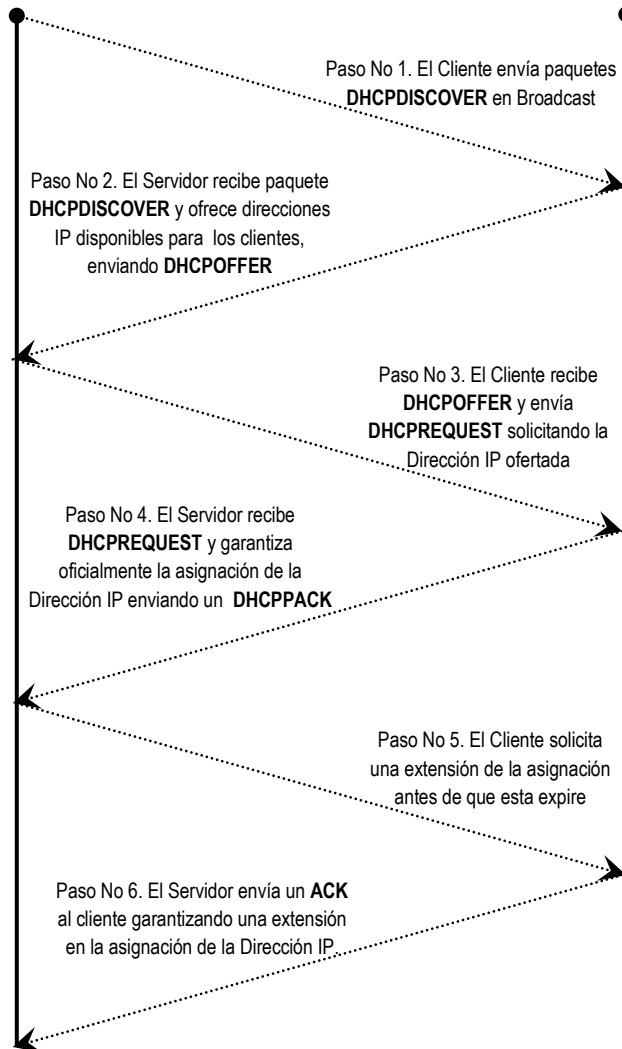
EL SERVICIO DHCP

(Dynamic Host Configuration Protocol)

Cliente DHCP



Servidor DHCP



Línea de Tiempo

DHCP

Dynamic Host Configuration Protocol

Capítulo No 3. EL SERVICIO DHCP (Dynamic Host Configuration Protocol)

El protocolo DHCP (Dynamic Host Configuration Protocol) fue desarrollado en los años 90 por la IETF (Internet Engineering Task Force) teniendo como objetivo principal superar las limitaciones de BOOTP. Este nuevo protocolo permitió nuevas formas de asignación de direcciones y la posibilidad de entregar a los clientes toda la información de red necesaria para su conectividad.

El protocolo DHCP fue diseñado para manejar rangos de direcciones IP de forma dinámica y automatizada basándose en el modelo Cliente-Servidor. DHCP utiliza un protocolo de comunicaciones basado en UDP sobre IP. Aquellos equipos de la red que necesiten direccionamiento IP, se les “presta” una dirección de un servidor que puede o no ser local. Cuando un PC de la red se enciende, pide una dirección IP o una renovación de la que le fue prestada. Este PC recibe la dirección IP con algunos parámetros adicionales, como su Gateway por defecto, servidor WINS, servidor DNS, etc. Debido a lo anteriormente expuesto se empieza a vislumbrar el objetivo general del protocolo DHCP; asignación y liberación de direcciones IP para una red de manera dinámica y automática.

3.1 FUNCIONAMIENTO DHCP

El fundamento del protocolo DHCP está en la asignación dinámica de direcciones IP a los computadores que hacen parte de una red. Teniendo en cuenta esta premisa, todo equipo que desee usar dicho protocolo para interactuar con los demás equipos de la red, deben ser configurados como clientes DHCP. Esto implicaría el envío de una petición de difusión DHCP por la red en busca de los parámetros necesarios para lograr su conectividad.

Cuando un servidor DHCP en la misma red escucha la petición, comprueba su base de datos local y envía una respuesta que incluye la dirección IP a conceder al cliente y que además según la configuración del servidor puede incluir otra información útil como los son los servidores de nombre, la máscara de red y Gateway por defecto.

Todas las direcciones y demás parámetros que un servidor DHCP puede conceder, tiene un contrato asociado que establece por cuánto tiempo puede ser usada por el cliente antes de que esta sea bloqueada y deba conectarse con el servidor para renovarla.

3.1.1. Asignación automática

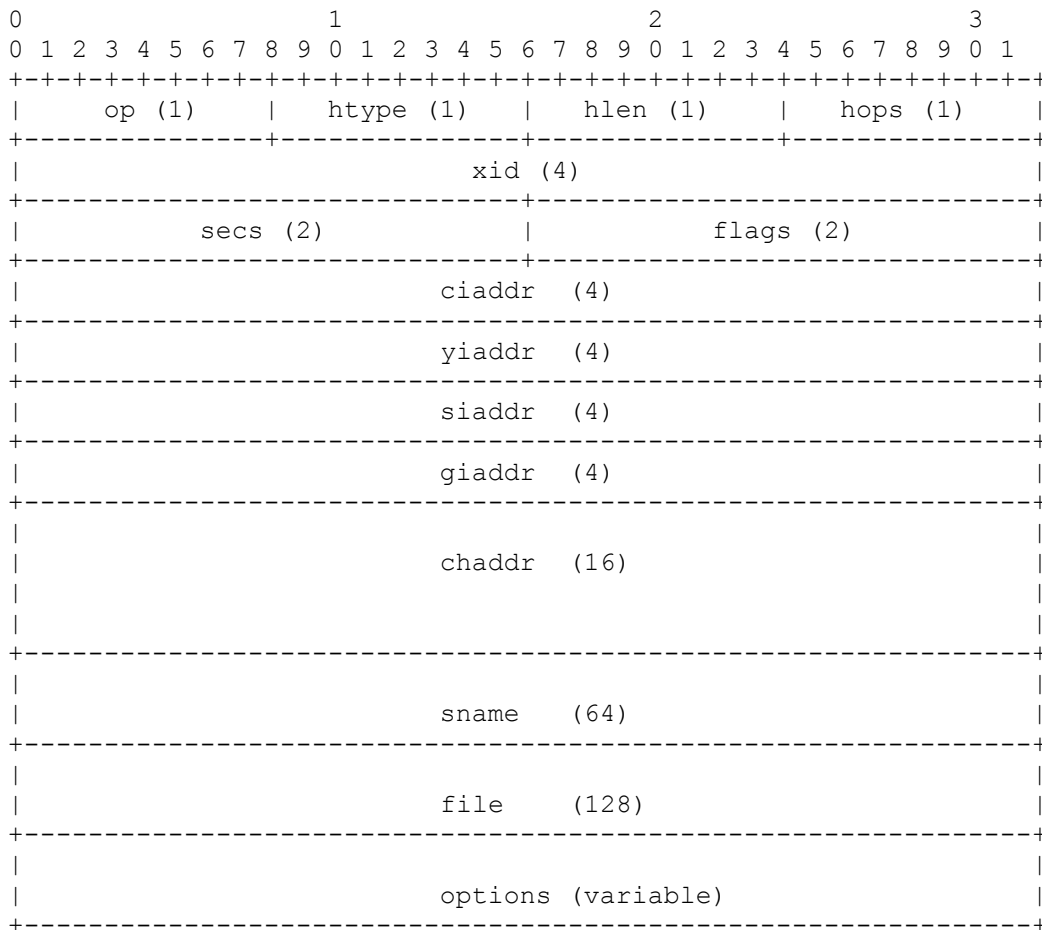
Este tipo de asignación DHCP facilita mucho la tarea del administrador ya que se utiliza en redes donde el número de clientes no varía demasiado, y la asignación de una dirección IP permanente resulta más eficiente.

3.1.2. Asignación dinámica

La asignación de direcciones IP dinámicas utilizando el protocolo DHCP, acuerda la utilización de una dirección por un período de tiempo limitado con un cliente. Más conocida como alquiler o arrendamiento de direcciones este mecanismo de asignación permite la reutilización automática de direcciones IP que ya no son necesitadas por los hosts a los que estaban asignadas logrando una eficiencia en la utilización y asignación de los rangos de direccionamiento disponibles.

3.1.3. Asignación manual

En este tipo de asignación de direccionamiento IP, las direcciones son asignadas por el administrador de la red a cada cliente.

Figura No 46. Formato del mensaje DHCP²⁶

3.1.4. Formato del mensaje DHCP

A continuación se hace una pormenorizada descripción de cada uno de los campos del formato del mensaje DHCP.

Op: Indica solicitud o respuesta

- 1 Request
- 2 Reply

²⁶ Disponible en Internet: <<http://www.ietf.org/rfc/rfc2131.txt>>

Htype: El tipo de hardware, por ejemplo:

- 1 Ethernet
- 6 IEEE 802 Networks

Hlength: Longitud en bytes de la dirección hardware. Ethernet y las redes en anillo usan 6 por ejemplo.

Hops: El cliente lo pone a 0. Cada "router" que retransmite la solicitud a otro servidor lo incrementa, con el fin de detectar bucles. El RFC 951²⁷ sugiere que un valor de 3 indica un bucle.

X ID: Número aleatorio usado para comparar la solicitud con la respuesta que genera.

Seconds: Fijado por el cliente. Es el tiempo transcurrido en segundos desde que el cliente inició el proceso de arranque.

Flags Field: El bit más significativo de este campo se usa como flag de difusión. Todos los demás bits deben estar en 0; están reservados para usos futuros. Normalmente, los servidores DHCP tratan de entregar los mensajes DHCPREPLY directamente al cliente usando unidifusión. La dirección de destino en la cabecera IP se pone al valor de la *dirección IP* fijada por el servidor DHCP, y la dirección MAC a la *dirección hardware* del cliente DHCP. Si un host no puede recibir un datagrama IP en unidifusión hasta saber su propia dirección IP, el bit de difusión se debe poner a 1 para indicar al servidor que el mensaje DHCPREPLY se debe enviar como una difusión en IP y MAC. De otro modo, este bit debe ponerse a cero.

Client IP adress: Fijada por el cliente. O bien es su dirección IP real, o 0.0.0.0.

²⁷ Disponible en versión HTML en Internet: <<http://www.ietf.org/rfc/rfc951.txt>>

Your IP Address: Fijada por el servidor si el valor del campo anterior es 0.0.0.0

Server IP address: Fijada por el servidor.

Router IP address: Fijada por el "router" retransmisor si se usa retransmisión *BOOTP*.

Client hardware address: Fijada por el cliente y usada por el servidor para identificar cuál de los clientes registrados está arrancando.

Server host name: Nombre opcional del host servidor acabado en X'00'.

Boot file name: El cliente o bien deja este campo vacío o especifica un nombre genérico, como "router" indicando el tipo de archivo de arranque a usar. En la solicitud de DHCPDISCOVER se pone al valor nulo. El servidor devuelve la ruta de acceso completa del archivo en una respuesta DHCPOFFER. El valor termina en X'00'.

Options: Los primeros cuatro bytes del campo de opciones del mensaje DHCP contienen el cookie(99.130.83.99). El resto del campo de opciones consiste en parámetros marcados llamados opciones especificadas con detalle en la norma RFC 1533.

Algunas implementaciones de DHCP pueden actualizar el DNS asociado con los servidores para reflejar las nuevas direcciones IP mediante el protocolo de actualización de DNS establecido en RFC 2136²⁸.

²⁸ Disponible en versión HTML en Internet: <<http://www.ietf.org/rfc/rfc2136.txt>>

El DHCP es una alternativa a otros protocolos de gestión de direcciones IP de red, como el BOOTP (*Bootstrap Protocol*), siendo el primero más avanzado, pero ambos son los usados normalmente.

En Windows 98 o posterior, cuando el DHCP es incapaz de asignar una dirección IP, se utiliza un proceso llamado "Automatic Private Internet Protocol Addressing".

3.2 INSTALACIÓN Y CONFIGURACIÓN

El primer paso que se debe realizar para hacer cualquier tipo de instalaciones en un servidor Linux es usar un usuario con privilegios (root) para tales acciones, por lo que en nuestro Ubuntu Server lo primero que haremos será una autenticación por consola que nos de tales privilegios. Para lograr esto digitamos el comando `$sudo su`, y damos <Enter>, digitamos la clave que se nos pidió en la instalación y listo. En estos momentos estamos utilizando el usuario "root".

```
cuc@us804:~$ sudo su
[sudo] password for cuc:
root@us804:/home/cuc# _
```

Figura No 47. Autenticación por consola

En este momento podemos utilizar cualquier comando de administración y el sistema lo ejecutara por los privilegios del usuario que estamos utilizando.

Para hacer la instalación de todos los servicios descritos en este libro debemos hacer una actualización del sistema por completo para tener las últimas versiones de los paquetes que se encuentren disponibles para descarga. Utilizando el comando `#sudo apt-get update`, logramos este objetivo.

```
[sudo] password for cuc:
root@us804:/home/cuc# apt-get update
Obj http://co.archive.ubuntu.com hardy Release.gpg
Des:1 http://security.ubuntu.com hardy-security Release.gpg [189B]
Ign http://security.ubuntu.com hardy-security/main Translation-es
Ign http://security.ubuntu.com hardy-security/restricted Translation-es
Obj http://co.archive.ubuntu.com hardy/main Translation-es
Ign http://security.ubuntu.com hardy-security/universe Translation-es
Ign http://security.ubuntu.com hardy-security/multiverse Translation-es
Des:2 http://security.ubuntu.com hardy-security Release [58,5kB]
Obj http://co.archive.ubuntu.com hardy/restricted Translation-es
Des:3 http://security.ubuntu.com hardy-security/main Packages [214kB]
Obj http://co.archive.ubuntu.com hardy/universe Translation-es
Obj http://co.archive.ubuntu.com hardy/multiverse Translation-es
Des:4 http://co.archive.ubuntu.com hardy-updates Release.gpg [189B]
Des:5 http://security.ubuntu.com hardy-security/restricted Packages [9942B]
Ign http://co.archive.ubuntu.com hardy-updates/main Translation-es
Des:6 http://security.ubuntu.com hardy-security/main Sources [33,9kB]
Ign http://co.archive.ubuntu.com hardy-updates/restricted Translation-es
Ign http://co.archive.ubuntu.com hardy-updates/universe Translation-es
Ign http://co.archive.ubuntu.com hardy-updates/multiverse Translation-es
Des:7 http://security.ubuntu.com hardy-security/restricted Sources [946B]
Obj http://co.archive.ubuntu.com hardy Release
Des:8 http://security.ubuntu.com hardy-security/universe Packages [106kB]
75% [Esperando las cabeceras] [8 Packages 2419/106kB 2%] 32,4kB/s 3s_
```

Figura No 48. Actualización de la base de datos de paquetes

Después de realizar esta acción tenemos actualizada nuestra base de datos de paquetes disponibles para descargar e instalar en nuestro servidor. Comenzamos entonces con el paquete que convertirá nuestra maquina en un servidor DHCP, `#apt-get install dhcp3-server`, al presionar `<Enter>`, el sistema descargara el paquete para su instalación, preguntándonos si deseamos bajarlo e instalarlo a lo cual decimos que sí.

```
root@us804:/etc/apt# apt-get install dhcp3-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  dhcp3-client dhcp3-common
Paquetes sugeridos:
  avahi-autoipd resolvconf
Se instalarán los siguientes paquetes NUEVOS:
  dhcp3-server
Se actualizarán los siguientes paquetes:
  dhcp3-client dhcp3-common
2 actualizados, 1 se instalarán, 0 para eliminar y 26 no actualizados.
Necesito descargar 821kB de archivos.
After this operation, 774kB of additional disk space will be used.
¿Desea continuar [S/n]? S_
```

Figura No 49. Instalación del servidor DHCP

Automáticamente comienza la descarga del paquete y su instalación. Al finalizar estas dos acciones y al querer activar el servicio sale un error que indica que este no puede ser activado, es normal dicho error ya que aún no se ha configurado el archivo que controla el servicio DHCP.

```
buntu9.1 [281kB]
Des:3 http://co.archive.ubuntu.com hardy-updates/main dhcp3-server 3.0.6.dfsg-1u
buntu9.1 [319kB]
Descargados 821kB en 11s (72,4kB/s)
Preconfigurando paquetes ...
(Leyendo la base de datos ...
15312 ficheros y directorios instalados actualmente.)
Preparando para reemplazar dhcp3-client 3.0.6.dfsg-1ubuntu9 (usando .../dhcp3-cl
ient_3.0.6.dfsg-1ubuntu9.1_i386.deb) ...
Desempaquetando el reemplazo de dhcp3-client ...
Preparando para reemplazar dhcp3-common 3.0.6.dfsg-1ubuntu9 (usando .../dhcp3-co
mmon_3.0.6.dfsg-1ubuntu9.1_i386.deb) ...
Desempaquetando el reemplazo de dhcp3-common ...
Seleccionando el paquete dhcp3-server previamente no seleccionado.
Desempaquetando dhcp3-server (de .../dhcp3-server_3.0.6.dfsg-1ubuntu9.1_i386.deb
) ...
Configurando dhcp3-common (3.0.6.dfsg-1ubuntu9.1) ...
Configurando dhcp3-client (3.0.6.dfsg-1ubuntu9.1) ...


Configurando dhcp3-server (3.0.6.dfsg-1ubuntu9.1) ...
Generating /etc/default/dhcp3-server...
* Starting DHCP server dhcpd3                                [fail]
invoke-rc.d: initscript dhcp3-server, action "start" failed.

root@us804:/etc/apt# _
```

Figura No 50. Descarga del paquete de instalación DHCP

En este momento ya tenemos instalado el paquete para hacer funcionar el servidor DHCP, ahora tenemos que configurarlo mediante su archivo de configuración. Antes de entrar a configurar este archivo debemos configurar la interfaz de red por la que el servidor recién instalado entregará dirección a los equipos de su red, para esto configuramos el archivo “dhcp3-server” ubicado en “/etc/default/”, diciéndole que en nuestro caso debe tomar la interfaz de red eth0 para hacer la entrega de direcciones.

Para esto damos el comando “#nano /etc/default/dhcp3-server” y sustituimos la línea INTERFACES="" por INTERFACES="eth0", con esto se logra decirle al servidor que utilice la tarjeta de red eth0 para entregar direcciones IP a los pc's que se la soliciten.



```
GNU nano 2.0.7 File: /etc/default/dhcp3-server
# Defaults for dhcp initscript
# sourced by /etc/init.d/dhcp
# installed at /etc/default/dhcp3-server by the maintainer scripts
#
# This is a POSIX shell fragment
#
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="eth0"
[ Read 11 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figura No 51. Configuración de la tarjeta de Red para asignación de IPs

Guardamos los cambios y se cierra el archivo.

El archivo de configuración del servidor DHCP que acabamos de instalar, y para las distribuciones que lo tienen instalado, es por defecto **/etc/dhcp3/dhcpd.conf** y en él encontramos un conjunto de declaraciones y parámetros que le darán a la configuración del servidor su capacidad y robustez. Este archivo está en texto ASCII y se puede modificar utilizando un editor de texto. Es necesario crear este archivo desde la primera vez que se pretenda utilizar DHCP ya que la instalación no nos proporciona un archivo de configuración por defecto o de ejemplo como para otros servicios. La estructura general de este archivo es la siguiente:

Parámetros globales;

Declaracion1{

[parámetros relacionados con la declaración]

[subdeclaración anidada]


```
}  
Declaracion2{  
  [parámetros relacionados con la declaración]  
  [subdeclaración anidada]  
}...
```

Los bloques de declaraciones se utilizan comúnmente para agrupar varias características comunes a un conjunto de máquinas clientes que tengan una o varias características en particular.

3.2.1. Declaraciones

Las declaraciones describen la topología de la red, los clientes que la conforman, de igual forma, proveen las direcciones de red para los clientes o aplica un grupo de parámetros a un grupo de clientes.

Group: Se puede usar para aplicar a parámetros globales o a un grupo de declaraciones. Esta declaración es usada para aplicar parámetros a un conjunto de clientes, redes compartidas o subredes que se deseen agrupar. La sintaxis de **group** es la siguiente:

```
group etiqueta {  
  [parámetros]  
  [subdeclaraciones]  
}
```

La **etiqueta** es el nombre definido por el usuario para identificar el grupo. El bloque *parámetros* es una lista de parámetros que se aplican a este grupo específico, mientras que el de **subdeclaraciones** se utiliza para describir a los

clientes que perteneciendo a la declaración actual necesitan ser especificados mediante una nueva declaración.

Host: Esta declaración aplica un conjunto de parámetros y declaraciones a una máquina específica, asimismo los parámetros indicados al grupo. Es usada en la mayoría de los casos para fijar la dirección de arranque, o para clientes BOOTP (DHCP también tiene la capacidad de responder a este tipo de peticiones).

La sintaxis de **host** es la siguiente:

```
Host etiqueta {  
  [parámetros]  
  [subdeclaraciones]  
}
```

La **etiqueta** es el nombre definido por el usuario para identificar a la máquina dentro de un grupo. **Parámetros** y **subdeclaraciones** hacen referencia al mismo tipo de declaraciones de **group**.

Shared-network: Esta declaración agrupa a un conjunto de direcciones de clientes de una misma red física. Las declaraciones tipo **shared-network** deben nombrarse de manera representativa de la red a la cual pertenezca. Su sintaxis es:

```
shared-network etiqueta {  
  [parámetros]  
  [subdeclaraciones]  
}
```

Donde la **etiqueta** es el nombre definido para la red compartida. **parámetros** y **subdeclaraciones** son como se especificó anteriormente en **group**.

Subnet: Se utiliza para aplicar un conjunto de declaraciones y parámetros a un conjunto de direcciones que se encuentran dentro de la especificada por esta declaración. Su sintaxis es la siguiente:

```
subnet numero-subred netmask máscara {  
    [parámetros]  
    [subdeclaraciones]  
}
```

El **numero-subred** como su nombre lo indica claramente hace referencia a la red a la cual queremos especificar las nuevas características, la **máscara** nos indica la máscara de red. **Parámetros** y **subdeclaraciones** son como se especificó anteriormente.

Para este tipo de declaraciones se deben tener en cuenta los siguientes puntos: se debe incluir una declaración **subnet** por cada subred que se quiera configurar en la red; si esto no se especifica, el protocolo DHCP fallará al iniciar. De igual forma los parámetros dentro de un **shared-network**, pero fuera de una declaración **subnet** son considerados parámetros globales.

Range: Es utilizado en DHCP para especificar el rango de direcciones válidas para los clientes. La sintaxis es la siguiente:

```
range [dynamic-bootp] dirección-inicial [dirección-final];
```

La palabra clave ***dynamic-bootp*** es utilizada para que el servidor obtenga el rango de direcciones para el protocolo BOOTP. Los campos ***dirección-inicial*** y el ***dirección-final*** siendo este último de carácter opcional, son las direcciones reales de los bloques de inicio y final de las direcciones IP a ser asignadas.

3.2.2. Parámetros

Señala la manera como ejecutar una tarea o que opciones de configuración de red se envían al cliente. Es precisamente con la modificación de estos parámetros que se logra modificar el comportamiento del servidor para grupos de clientes. Aquellos parámetros que se encuentren declarados antes de una sección encerrada en corchetes ({ }) son considerados parámetros globales y aplican a todas las secciones debajo de éste.

Always-reply-rfc1048. Su sintaxis es la siguiente:

```
always-reply-rfc1048;
```

Es utilizado para clientes BOOTP, se necesita para algunos clientes que necesitan esta respuesta del servidor para ser completamente compatibles con BOOTP RFC.

Authoritative. Su sintaxis es la siguiente:

```
authoritative;
```

```
no authoritative;
```

Este parámetro se utiliza para designar a una red en particular como “autorizada”, este es su valor por defecto. Cuando una red es no autorizada, el servidor enviará un DHCPNAK al cliente. El cliente debería entonces reintentar la petición.

Default-lease-time. La sintaxis de este parámetro es la siguiente:

default-lease-time segundos;

El valor en segundos es el tiempo que dura el contrato asignado a la dirección IP si el cliente no pide cualquier duración.

Dynamic-bootp-lease-cutoff. La sintaxis de este parámetro es la siguiente:

dynamic-bootp-lease-cutoff fecha;

Se utiliza cuando el servidor necesite limitar la entrega de direcciones a clientes BOOTP los cuales por defecto reciben direcciones que nunca caducan. La fecha se especifica en la forma SAAAA/MM/DD HH:MM:SS, donde S es el día de la semana en formato de cron (0 = domingo, 6 = sábado); AAAA es el año; MM es el mes (01 = enero, 12 = diciembre); DD es el día en formato de dos dígitos; HH es la hora en el formato de 24 horas (0 = Medianoche, 23 = 11 de la noche), MM representa a su vez los minutos y SS representa los segundos.

Dynamic-bootp-lease-length. La sintaxis de este parámetro es la siguiente:

dynamic-bootp-lease-length segundos;

Se utiliza para caducar una dirección entregada a un cliente BOOTP luego de que pase el tiempo especificado.

Filename. La sintaxis de este parámetro es la siguiente:

filename nombre-archivo;

Se utiliza cuando se necesita conocer el nombre de un archivo para usarlo en el arranque. Con frecuencia se utiliza con **next-server** para recuperar un archivo remoto con el fin de configurar la instalación o arrancar un cliente sin disco.

Fixed-address. Con esta opción se logra asignar una o más direcciones IP a un cliente. Si se presenta el caso de entregarle a un cliente más de una dirección IP, cuando el cliente inicie, le será asignada la dirección a la que corresponda la red en la cual está iniciando. La sintaxis de este parámetro es la siguiente:

```
fixed-address dirección [, dirección];
```

Este parámetro aparece sólo bajo la declaración "host". Especifica el conjunto de direcciones asignables al cliente.

Get-lease-hostname. La sintaxis de este parámetro es la siguiente:

```
get-lease-hostname [true = false];
```

Si se configura a true, el servidor resuelve todas las direcciones en el ámbito de la declaración y la usará para la declaración.

Hostname. hardware. Sirve para asignar direcciones IP a un cliente, basadas en la dirección MAC de la tarjeta de interfaz de red. Este parámetro se utiliza dentro de una declaración host. La sintaxis de este parámetro es la siguiente:

```
hardware [ ethernet - token-ring ] dirección-hardware
```

Se utiliza para que el servidor pueda identificar a una máquina específica. La dirección hardware no es más que la dirección MAC es decir la dirección física de la interfaz, especificada mediante un conjunto de octetos decimales delimitados por símbolos de dos puntos. Este parámetro se utiliza para fijar las direcciones de los clientes DHCP y se requiere para los clientes BOOTP.

Max-lease-time. La sintaxis de este parámetro es la siguiente:

```
max-lease-time segundos;
```

Cuando un cliente pide la duración de su contrato, esta duración se concede

siempre y cuando no exceda el número de segundos especificado por esta opción. De lo contrario, se concede un contrato al máximo de segundos especificado aquí.

Server-identifier. La sintaxis de este parámetro es la siguiente:

```
server-identifier nombre-máquina;
```

Se utiliza para enviar la IP del servidor a la interfaz apropiada de la máquina cliente.

Server-name. La sintaxis de este parámetro es la siguiente:

```
server-name nombre;
```

El nombre es el nombre de la máquina servidor que arranca el cliente remoto. Este parámetro se usa por los clientes remotos o aplicaciones de instalación de red.

Use-host-decl-names. La sintaxis de este parámetro es la siguiente:

```
use-host-decl-names [ true - false ];
```

Este parámetro se usa en el mismo ámbito que las otras declaraciones de máquina. Añadirá la opción *host-name* a la declaración *host*, usando el nombre de máquina en la declaración de la opción *host*.

Use-lease-addr-for-default-route. La sintaxis de este parámetro es la siguiente:

```
use-lease-addr-for-default-route [ true - false ];
```

Se utiliza cuando se quiere utilizar *arp* (*Address Resolution Protocol* ; Protocolo de Resolución de Direcciones) para encontrar todas las direcciones remotas.

3.2.3. Opciones

Algunos parámetros deben iniciar con la palabra clave option. Option configura las opciones de DHCP, mientras que los parámetros configuran los valores que no son opcionales o controlados como el comportamiento del servidor DHCP. Aunque son muchas las opciones soportadas por el servidor DHCP aquí veremos sólo las de uso más común. La sintaxis general de una opción es la siguiente:

option nombre-opcion [modificadores]

Broadcast-Address: Especifica una dirección de una subred de cliente como dirección de difusión.

Domain-name: Hace referencia al nombre de dominio que debería usar el cliente como nombre de dominio local cuando se realizan bucles locales.

Domain-name-servers: La lista de direcciones IP de servidores DNS que usa el cliente para resolver nombres de máquinas.

Host-name: Cadena de caracteres usada para identificar el nombre del cliente.

Nis-domain: El nombre del dominio NIS (*Network Information Service*).

Nis-server: La lista de servidores NIS disponible para unirse a ellos.

Routers: Especifica la lista de direcciones IP para enrutadores en la subred del cliente. Deben ser listados en orden de preferencia.

Subnet-mask: Especifica la máscara de subred. Si ésta no es especificada, se utiliza la de la declaración de subnet, si se especifica se sobrescribe la de subnet.

Option netbios-name-servers: Lista de los servidores NBNS en orden de preferencia. Este se utiliza para clientes WINS.

Option netbios-node-type: Especifica el tipo de nodo que permite NetBios sobre clientes TCP/IP. Los valores son: 1(Tipo B), 2(Tipo P), 4(Tipo M), 8(Tipo H).

3.2.4. Ejemplo de dhcpd.conf

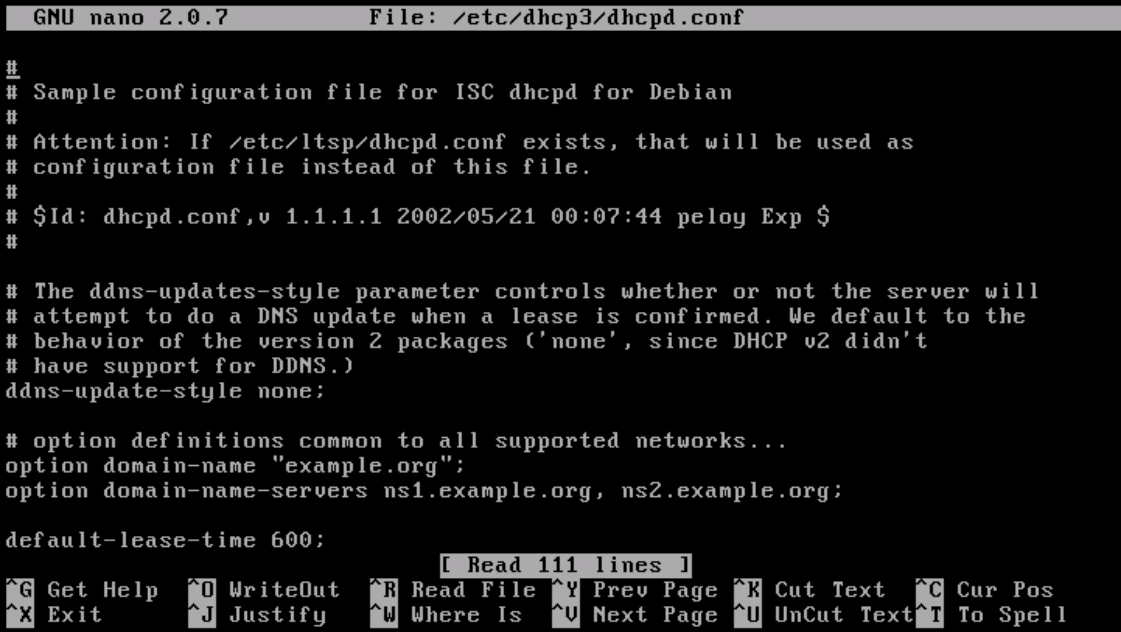
Antes de hacer cualquier modificación sobre este archivo vamos a crear una copia exacta para poder modificarlo sin temor a dañarlo, esta es una buena práctica para todos aquellos archivos que se encargan de la administración de servidores.

Ejecutamos entonces el comando `#sudo cp /etc/dhcp3/dhcpd.conf /etc/dhcp3/dhcpd.conf_Copia`, y logramos crear una copia exacta del archivo original para efectos de restauración si así fuese el caso.

```
root@us804:/# cp /etc/dhcp3/dhcpd.conf /etc/dhcp3/dhcpd.conf_Copia
root@us804:/# cd /etc/dhcp3/
root@us804:/etc/dhcp3# ls
dhclient.conf          dhclient-enter-hooks.d  dhcpd.conf
dhclient.conf.dpkg-dist dhclient-exit-hooks.d  dhcpd.conf_Copia
root@us804:/etc/dhcp3# _
```

Figura No 52. Creación de la copia del archivo de configuración DHCP

En este momento podremos ejecutar el comando `#nano /etc/dhcp3/dhcpd.conf`, y configurar dicho archivo para nuestras necesidades.



```
GNU nano 2.0.7 File: /etc/dhcp3/dhcpd.conf
#
# Sample configuration file for ISC dhcpd for Debian
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#
# $Id: dhcpd.conf,v 1.1.1.1 2002/05/21 00:07:44 peloy Exp $
#
# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;

[ Read 111 lines ]
^G Get Help ^O WriteOut ^R Read File ^V Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell
```

Figura No 53. Configuración del archivo dhcpd.conf 1

Este archivo tiene toda la configuración del servidor DHCP. En él podemos ver comentarios (aquellas líneas que empiezan con el símbolo #) y sentencias ejecutables aquellas que carecen de este símbolo. El desplazamiento sobre el archivo nos llevara hasta aquellas sentencias que deseamos modificar o a aquellos lugares donde deseamos ubicar nuevas que cumplan tareas específicas.

A continuación procedemos a modificar el contenido del archivo, comentando o añadiendo solo aquellas sentencias que sean necesarias para nuestros fines. En el ejemplo que se muestra a continuación se borrara todo el contenido del archivo y se plantea la existencia de una red con las siguientes características: Red: 192.168.2.0, Gateway: 192.168.2.1 y DNS: 10.2.0.11 primario y 11.2.0.15 secundario.

Siguiendo con los lineamientos anteriores nuestro archivo de configuración DHCP “#/etc/dhcp3/dhcpd.conf” quedara como sigue:

```
GNU nano 2.0.7      File: /etc/dhcp3/dhcpd.conf
subnet 192.168.2.0 netmask 255.255.255.0 {
  range 192.168.2.2 192.168.2.10;
  option domain-name-servers 10.2.0.11, 10.2.0.15;
  option domain-name "labredes.com";
  option routers 192.168.2.1;
  option broadcast-address 192.168.2.255;
  default-lease-time 600;
  max-lease-time 7200;
}

[ Read 9 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^U Next Page  ^U UnCut Text ^T To Spell
```

Figura No 54. Configuración del archivo dhcpd.conf 2

A continuación procederemos a reiniciar los servicios con el comando “#/etc/init.d/dhcp3-server restart” en el servidor y nos debe aparecer que los servicios se han reiniciado con éxito. En este punto ya los computadores que se encuentren disponibles para recibir dirección de este servidor podrán adquirirla.

```
root@us804:/# /etc/init.d/dhcp3-server restart
* Stopping DHCP server dhcpd3          [ OK ]
* Starting DHCP server dhcpd3          [ OK ]
root@us804:/# _
```

Figura No 55. Reinicio de los servicios

Para lograr que los equipos de la red logren adquirir una dirección del servidor que se acaba de configurar se deben seguir los siguientes pasos en los computadores de la red.

En las conexiones de red del equipo vamos a la tarjeta de red que queremos que obtenga la dirección IP dinámica y le decimos que nos muestre sus propiedades del protocolo TCP/IP y debemos asegurarnos que este configurado para que obtenga una dirección de forma automática y sus DNS de igual forma.

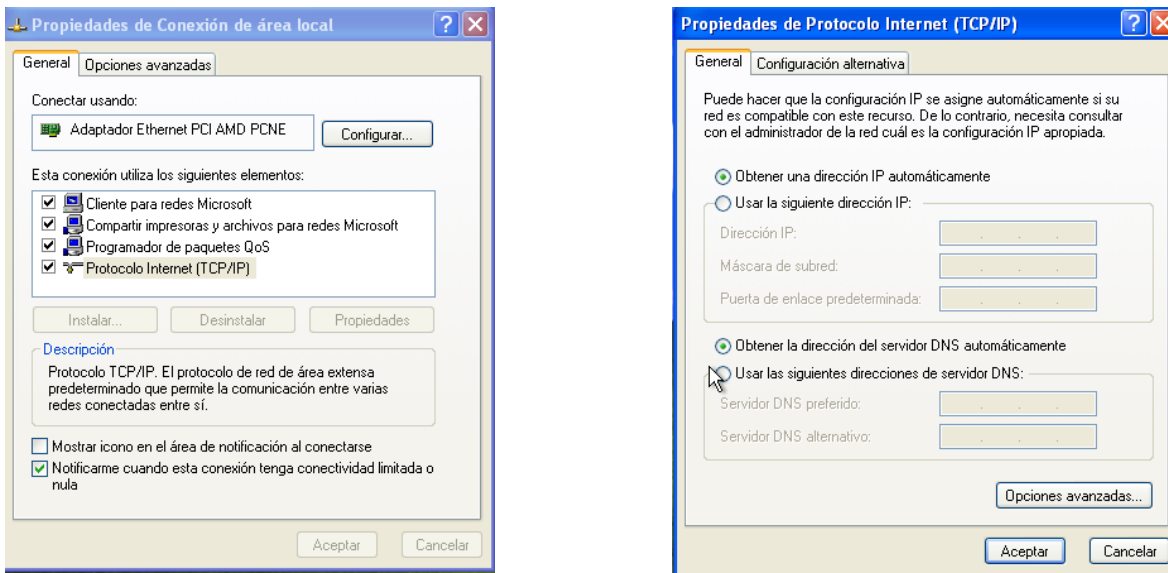


Figura No 56. Configuración del PC para detección de IP dinámica

Inmediatamente después de hacer estas configuraciones nos vamos a una consola de DOS y revisamos la dirección IP asignada:

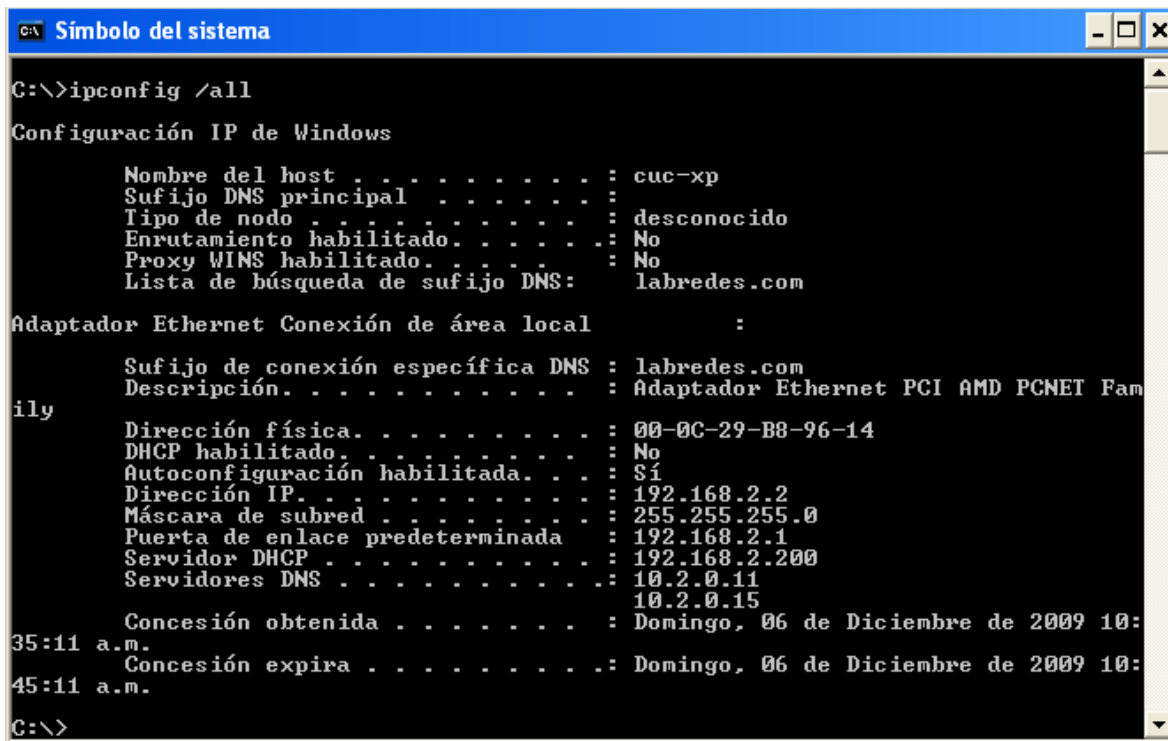
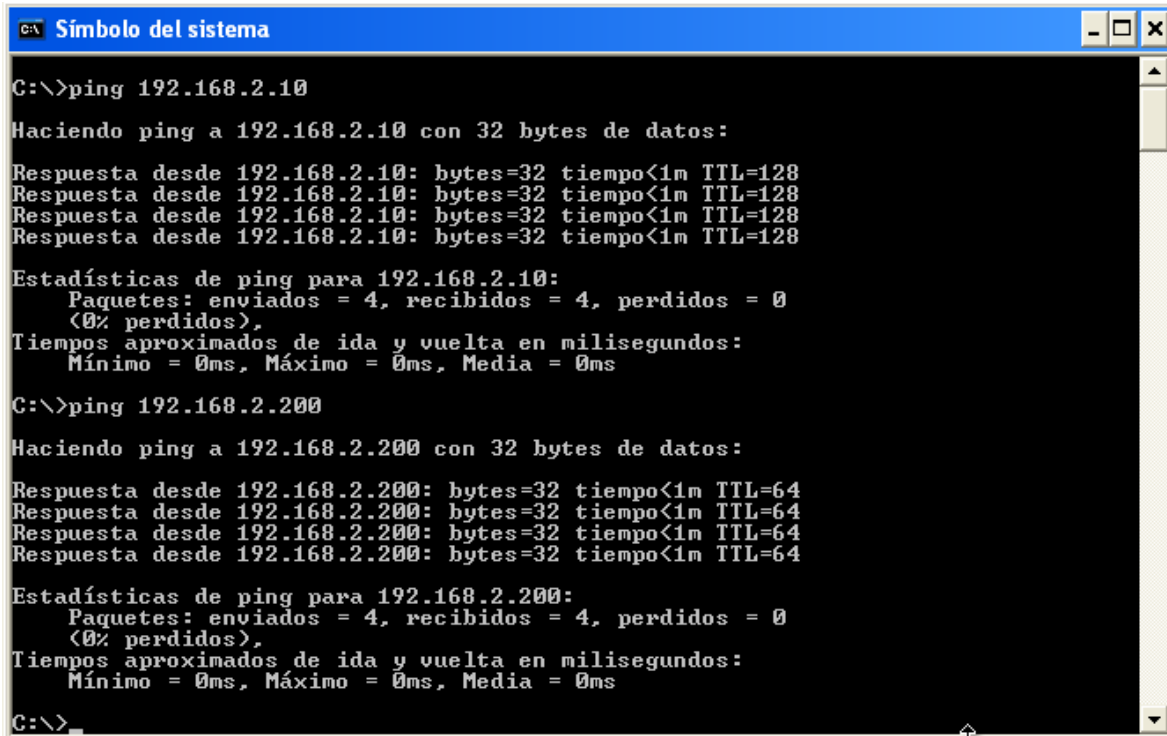


Figura No 57. Verificación por consola de direccionamiento IP dinámico

Y a continuación probamos conectividad con los demás equipos de la red incluyendo el servidor:



```
C:\>ping 192.168.2.10

Haciendo ping a 192.168.2.10 con 32 bytes de datos:

Respuesta desde 192.168.2.10: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.2.10: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.2.10: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.2.10: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.2.10:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\>ping 192.168.2.200

Haciendo ping a 192.168.2.200 con 32 bytes de datos:

Respuesta desde 192.168.2.200: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.2.200: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.2.200: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.2.200: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.2.200:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\>
```

Figura No 58. Prueba de conectividad entre PCs

En este momento hemos terminado de configurar nuestro servidor DHCP para que entregue direccionamiento IP a una red de computadoras.

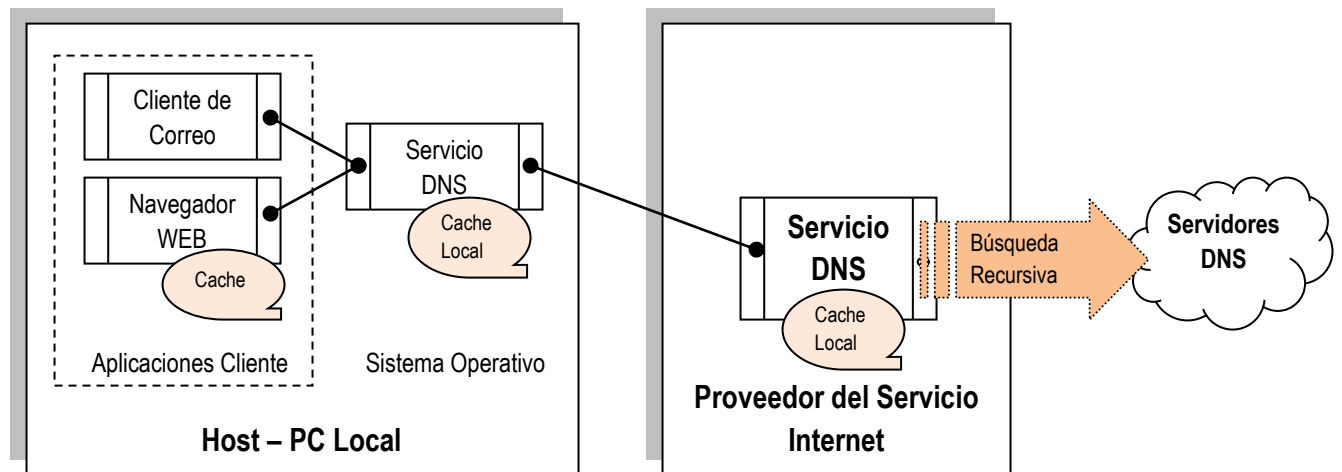
Capítulo No

EL SERVICIO DNS

(Domain Name System)

DNS

Domain Name System



Capítulo No 4. EL SERVICIO DNS (Domain Name System)

El sistema de nombres de dominio (DNS) es un sistema para asignar nombres a equipos y servicios de red que se organiza en una jerarquía de dominios. Desde la creación de TCP/IP, se ha buscado la forma de hacer que el mapeado de las direcciones IP sea más amigable para el usuario, debido a esta necesidad surgen los famosos servidores DNS los cuales nos ayudan a buscar equipos y servicios mediante nombres descriptivos, dejando en el olvido el mantenimiento del archivo HOSTS.TXT que se distribuía a todas las máquinas de Internet a través de FTP. Por esta razón se creó un sistema distribuido en el cual, cada sitio mantenía información de sus propias máquinas, como direcciones IPs con sus respectivos nombres, es a este servicio al que hoy se le conoce como DNS.

Buscando la facilidad en los recursos de red, este tipo de sistemas proporcionan un método para asignar el nombre descriptivo de un equipo o servicio a otros datos asociados a dicho nombre, como una dirección IP. Es claro para todos que un nombre es más fácil de aprender que cualquier otra forma de conocida de asociación a algo o a alguien, como es el caso en redes de datos las ya famosas direcciones numéricas que los equipos usan para comunicarse a través de una red. En la mayoría de los casos los administradores de red prefieren usar un nombre descriptivo (por ejemplo, cuc.edu.co) para referirse a un servidor de correo electrónico o servidor web en una red en lugar de una dirección IP, como 10.2.0.11. Cuando un usuario escribe un nombre DNS descriptivo en una aplicación, los servicios DNS convierten el nombre en su dirección numérica.

Los servidores **DNS** se utilizan para múltiples propósitos. Los más comunes son:

- **Resolución de nombres:** Dado el nombre completo de un *host* (por ejemplo *cuc.edu.co*), obtener su *dirección IP* (en este caso, *10.2.0.11*).
- **Resolución inversa de direcciones:** Es el mecanismo inverso al anterior. Consiste en, dada una *dirección IP*, obtener el nombre de la misma.
- **Resolución de servidores de correo:** Dado un *nombre de dominio* (por ejemplo *cuc.edu.com*) obtener el servidor a través del cual debe realizarse la entrega del correo electrónico (en este caso, *correo.cuc.edu.co*).

Es necesario aclarar algunos términos que son utilizados al momento de hablar de DNS, esto con el fin de evitar ambigüedades o confusiones. El **Host Name** o "*nombre de un host*", es un solo término la cual está formado por letras, números y guiones. Ejemplos de nombres de host son "cuc", "heraldo" y "rcn", entre otros. El término **Fully Qualified Host Name (FQHN)**, es el "*nombre completo*" de un host y se forma con la unión del *hostname*, seguido de un punto y su correspondiente nombre de dominio. Por ejemplo, "cuc.edu.co". El **Domain Name** o *nombre de dominio* es la concatenación sucesiva de nombres separados por puntos. Algunos ejemplos son "correo.cuc.edu.co", "com.co" y "co". Los dominios de nivel superior o **Top Level Domains (TLD)** son aquellos que no pertenecen a otro dominio. Ejemplos de este tipo son "com", "org", "co" y "es".

4.1 FUNCIONAMIENTO DNS

Un servidor DNS proporciona resolución de nombres para redes basadas en TCP/IP. Por ende los equipos clientes que hagan parte de estas redes, utilizarán nombres en lugar de direcciones IP numéricas para identificar hosts remotos.

A continuación se detalla un ejemplo que representa dicha afirmación, si la maquina X identificada con dirección IP=10.0.0.1, desea comunicarse con una

maquina Y siendo esta la dirección 200.0.0.1, el proceso será el siguiente: La máquina X pregunta quién es el “acreditado” de todos los nombres de máquina del sitio donde se encuentra Y. Dicho acreditado lo llamaremos “Servidor DNS”. La máquina X recibe una respuesta parecida a “Nombre del servidor DNS de Y”. Seguidamente, la máquina X pregunta al Servidor DNS de Y, ¿Cuál es la dirección IP de la máquina Y?, al cual el servidor DNS de Y le contesta con la dirección IP de la máquina Y. Con esta dirección IP en su poder, el cliente A puede comenzar la comunicación con la máquina Y.

Este proceso puede tener lugar de forma recursiva hasta que el equipo cliente reciba las direcciones IP o hasta que se establezca que el nombre consultado no pertenece a ningún host del espacio de nombres DNS especificado.

Inicialmente, el DNS nació de la necesidad de recordar fácilmente los nombres de todos los servidores conectados a Internet. En un inicio, SRI (ahora SRI International) alojaba un archivo llamado *HOSTS* que contenía todos los nombres de dominio conocidos (técnicamente, este archivo aún existe - la mayoría de los sistemas operativos actuales todavía pueden ser configurados para revisar su archivo hosts). El crecimiento explosivo de la red causó que el sistema de nombres centralizado en el archivo HOSTS no resultara práctico y en 1983, Paul Mockapetris publicó los RFCs 882 y 883 definiendo lo que hoy en día ha evolucionado el DNS moderno. (Estos RFCs han quedado obsoletos por la publicación en 1987 de los RFCs 1034 y 1035).

4.2 CONCEPTOS

4.2.1. Clientes

Un cliente es una aplicación que envía consultas a un Servidor DNS para obtener información de un determinado dominio. Un cliente pueden ser Navegadores (Internet Explorer y FireFox), Clientes de Correo (Outlook, Thunderbird), Mensajeros (Yahoo Messenger, MSN Mesenger), etc.

4.2.2. Servidores DNS

Son los encargados de responder las peticiones de los clientes. Un servidor DNS puede funcionar de diversas formas, como:

- **Caché:** Estos guardan las consultas ejecutadas por un determinado tiempo, así reducen el tiempo que lleva el proceso de resolución de dominios.
- **Forwarding:** Reenvía todas las consultas recibidas a otro servidor DNS esto disminuye la carga al servidor.
- **Maestro:** Se encarga de administrar uno o varios dominios, teniendo en su configuración los subdominios de cada dominio.
- **Root Servers:** Servidores Raíz son servidores fijos que proporcionan la información de los Top Level (.com, .net, .edu, .org, etc.) o dominios de nivel superior, en la actualidad solo existen 13 servidores Raíz distribuidos por todo el mundo.
- **Top Level:** Dominio de nivel superior (.com, .net, .edu, .org, etc.) proporciona el servidor con autoridad de un dominio.

4.3 INSTALACIÓN

Una de las distribuciones que nos facilitan la administración y configuración de nuestro Linux, es la distribución Ubuntu, ya que está basada en debian y al ser una derivación de debian, también hereda la famosa herramienta apt-get para administrar el software que nos facilita la instalación y desinstalación de aplicaciones para Ubuntu sin necesidad de compilar nada.

4.3.1. Activando los repositorios

Para instalar bind necesitamos activar los repositorio multiverse y universe, si estamos usando la versión *Desktop* de Ubuntu bastará con ir al menú *Sistema - Administración - Orígenes de Software* y activar Multiverse y Universe, después cerramos y enseguida nos aparecerá un diálogo donde tenemos que hacer click en *Recargar* para que baje las listas de las aplicaciones de internet. Si utilizamos la versión *Server* tenemos que editar el siguiente archivo *sources.list*.

Ejecutamos el siguiente comando para editar el archivo con el editor de texto **nano** que es para consola.

```
sudo nano /etc/apt/sources.list
```

Quitamos las almohadillas (#) de las líneas donde aparezca *multiverse* y *universe*, nótese que después de estas almohadillas sigue la palabra **deb** o **deb-src**. Guardamos el archivo presionando <Ctrl> y la tecla X y después escribimos aceptamos con una S los cambios hechos. Por último hacemos una actualización de las listas ejecutando el siguiente comando: **sudo apt-get update**.

4.3.2. Instalación de bind

Para tener nuestro servicio DNS utilizaremos bind9 que es un Servidor DNS muy usado, ejecutamos el siguiente comando para instalarlo. **sudo apt-get install bind9.**

4.3.3. Configuración de la Red

Existen diversas formas de implementar un servidor DNS pero a nosotros solamente nos interesan dos formas:

- Como Servidor para nuestra LAN, el servidor DNS recibe todas las consultas de las computadoras clientes y estas las resuelve.
- Como administrador de un dominio, que se encargue de administrar un dominio por ejemplo "pitufos.com", cuando un cliente haga una petición del dominio pitufos.com nuestro bind reconocerá que ese dominio le pertenece y hará una resolución local devolviendo la dirección especificada por nosotros. No tendrá la necesidad de conectarse con los Root Servers o Servidores Raíz para obtener la información de ese dominio.

4.3.4. Servidor Ubuntu

Como se hizo en la instalación de DHCP, después de actualizar la lista de paquetes debemos descargar el paquete que convertirá nuestra máquina en un servidor DNS, `#apt-get install bind`, al presionar <Enter>, el sistema descargará el paquete para su instalación, preguntándonos si deseamos bajarlo e instalarlo a lo cual decimos que sí.

```
root@us804:/# apt-get install bind9
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  bind9-host dnsutils libbind9-30 libdns35 libisc35 libisccc30 libiscfg30
  liblwres30
Paquetes sugeridos:
  bind9-doc resolvconf rblcheck
Se instalarán los siguientes paquetes NUEVOS:
  bind9
Se actualizarán los siguientes paquetes:
  bind9-host dnsutils libbind9-30 libdns35 libisc35 libisccc30 libiscfg30
  liblwres30
8 actualizados, 1 se instalarán, 0 para eliminar y 18 no actualizados.
Necesito descargar 1213kB de archivos.
After this operation, 762kB of additional disk space will be used.
¿Desea continuar [S/n]? S_
```

Figura No 59. Descarga del Instalador DNS

Después de descargar los datos referentes al paquete de DNS, el sistema inicializa el servicio y podremos empezar a editar el archivo de configuración del servicio para nuestras necesidades.

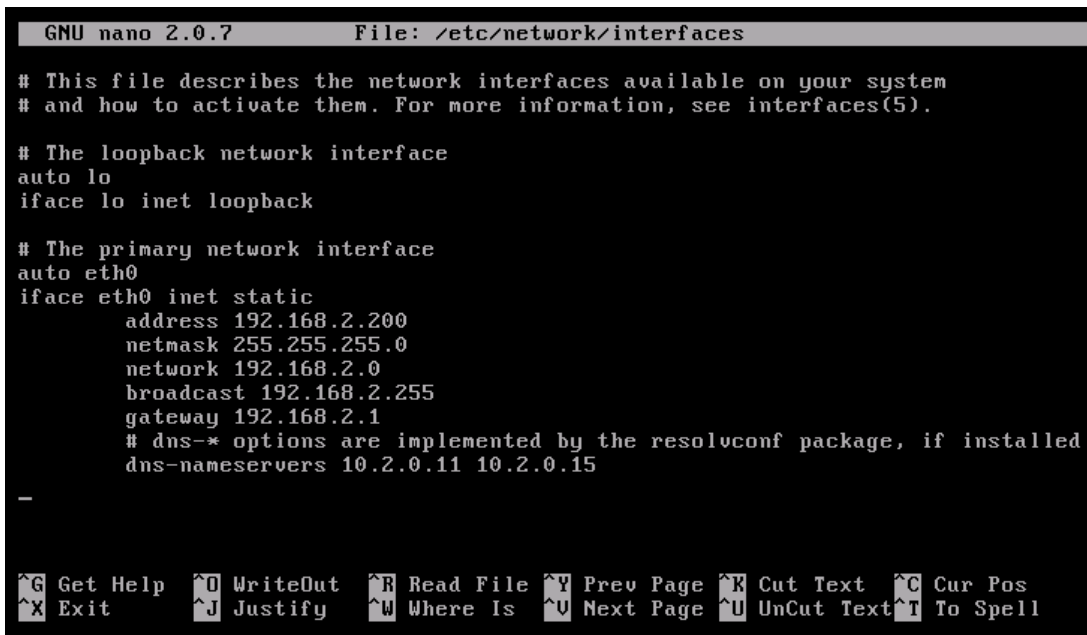
```
Configurando libisccc30 (1:9.4.2.dfsg.P2-Zubuntu0.2) ...
Configurando libiscfg30 (1:9.4.2.dfsg.P2-Zubuntu0.2) ...
Configurando libbind9-30 (1:9.4.2.dfsg.P2-Zubuntu0.2) ...
Configurando liblwres30 (1:9.4.2.dfsg.P2-Zubuntu0.2) ...
Configurando bind9-host (1:9.4.2.dfsg.P2-Zubuntu0.2) ...
Configurando dnsutils (1:9.4.2.dfsg.P2-Zubuntu0.2) ...

Configurando bind9 (1:9.4.2.dfsg.P2-Zubuntu0.2) ...
Adding group `bind' (GID 115) ...
Done.
Adding system user `bind' (UID 105) ...
Adding new user `bind' (UID 105) with group `bind' ...
Not creating home directory `/var/cache/bind'.
wrote key file "/etc/bind/rndc.key"
Reloading AppArmor profiles : done.
* Starting domain name service... bind [ OK ]

Processing triggers for libc6 ...
ldconfig deferred processing now taking place
root@us804:/home/cuc# _
```

Figura No 60. Inicialización del servicio

Este archivo se encuentra en “#/etc/network/” y se llama “interfaces”, la configuración de este será la siguiente: (digitamos el comando “#nano /etc/network/interfaces”).



```
GNU nano 2.0.7 File: /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

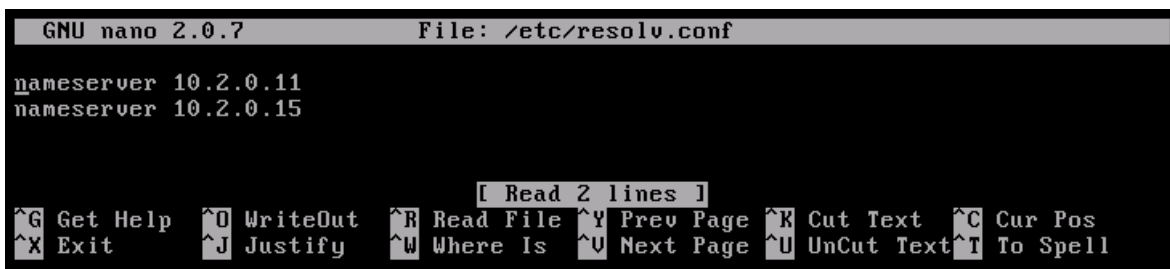
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.2.200
    netmask 255.255.255.0
    network 192.168.2.0
    broadcast 192.168.2.255
    gateway 192.168.2.1
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 10.2.0.11 10.2.0.15

-
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page  ^U UnCut Text ^T To Spell
```

Figura No 61. Edición del archivo Interfaces

Inmediatamente debemos configurar el archivo “resolv.conf” que se encuentra en “#/etc”, para eso digitamos el comando “#nano /etc/resolv.conf”, verificamos que el contenido del archivo contenga los siguientes servidores de nombres:



```
GNU nano 2.0.7 File: /etc/resolv.conf
nameserver 10.2.0.11
nameserver 10.2.0.15

[ Read 2 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page  ^U UnCut Text ^T To Spell
```

Figura No 62. Edición del archivo resolv.conf

Guardamos y cerramos el archivo. Con esto nos aseguramos que todos los pc's de nuestra red obtengan respuesta de resolución de nombres de dichos servidores.

4.4 CLIENTES WINDOWS

Para que las PCS con Windows de nuestra red LAN puedan usar nuestro servidor DNS montado en Ubuntu, tenemos que configurar con los siguientes parámetros.

Ir al *Panel de control - Conexiones de red* y seleccionar la *Conexión* de nuestra red y hacer clic derecho e ir a *Propiedades*. Seleccionamos *Protocolo Internet (TCP/IP)* y clic en *Propiedades* y establecemos la IP (192.168.1.200) de nuestro servidor en apartado *Servidor DNS preferido*.

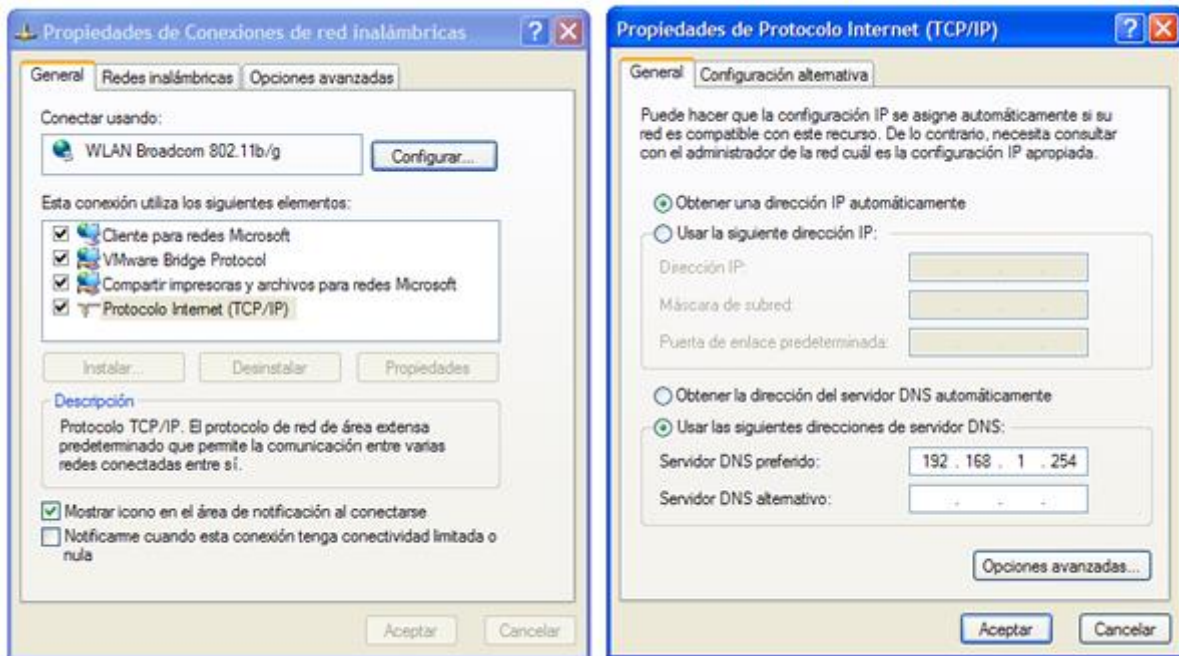


Figura No 63. Asignación del Servidor DNS preferido

Para probar nuestro DNS solamente hay que abrir el FireFox o IE y ver si no deja acceder a una página. Otra forma para comprobar es abrir el *Símbolo del sistema* y ejecutamos el comando *nslookup* y después escribir un dominio para obtener la dirección IP. Nuestro PC Windows enviara la consulta a nuestro servidor Ubuntu.

Hasta aquí no hemos configuramos nada en nuestro servidor, simplemente nada mas lo instalamos para que sea el DNS de nuestra LAN. Más adelante veremos la estructura de bind para así poder configurar nuestro propio dominio local.

4.5 ESTRUCTURA DE BIND

Bind está compuesto por varios archivos en donde podemos agregar y modificar algunos parámetros para cambiar el funcionamiento de nuestro servidor DNS. A continuación una descripción detallada de cada uno de estos.

4.5.1. named.conf

Archivo principal donde se especifican las zonas y la ubicación de sus archivos.

Contenido del archivo named.conf

```
1. include "/etc/bind/named.conf.options";
2.
3. // prime the server with knowledge of the root servers
4. zone "." {
5.     type hint;
6.     file "/etc/bind/db.root";
7. };
8.
9.
10. zone "localhost" {
11.     type master;
12.     file "/etc/bind/db.local";
13. };
14.
15. zone "127.in-addr.arpa" {
```

```
16. type master;
17. file "/etc/bind/db.127";
18.};
19.
20.zone "0.in-addr.arpa" {
21. type master;
22. file "/etc/bind/db.0";
23.};
24.
25.zone "255.in-addr.arpa" {
26. type master;
27. file "/etc/bind/db.255";
28.};
29.
30.// zone "com" { type delegation-only; };
31.// zone "net" { type delegation-only; };
32.include "/etc/bind/named.conf.local";
```

4.5.2. named.conf.options

Nos sirve para establecer la mayoría de las opciones de nuestro servidor. Aquí podemos definir el comportamiento de nuestro servidor como por ejemplo las redes que pueden utilizar el servidor, los derechos de transferencia, etc.

db.0 db.127 db.local db.255

Archivos de zonas predefinidas para nuestra red local e internet.

4.5.3. db.root

Archivo que contiene la información de todos los Root Servers

4.6 ZONAS

Las zonas no son más que archivos en donde se establece la información de un dominio, para crear una nueva zona primero tenemos que especificar la ruta del archivo de zona, esto se hace en el interior del archivo `/etc/bind/named.conf` si lo

abrimos podremos observar que ya hay algunas zonas predeterminadas como se muestra en la figura:



```
GNU nano 2.0.7 File: /etc/bind/named.conf
zone "localhost" {
    type master;
    file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
    type master;
    file "/etc/bind/db.255";
};

[ Read 51 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figura No 64. Archivo de configuración de Zonas named.conf

4.7 TIPOS DE REGISTROS DNS

Para crear un dominio antes tenemos que saber los tipos de registros que existen, estos registros los usaremos a la hora de crear una nueva zona. Los registros DNS son: SOA, A, CNAME, NS y MX, PTR. De ellos se realiza a continuación una breve descripción.

4.7.1. SOA

Es el Registro principal en donde se definen las características de la zona y también contiene información del dominio. Dentro de este registro podemos encontrar las siguientes directrices.

host.midominio.com. Indica el nombre del host del DNS al cual pertenece la zona.

root.midominio.com. Cuenta de correo electrónico del administrador.

SOA está constituido además por los parámetros Serial, Refresh, Retry, Expire y TTL.

Serial: Es un número incremental arbitrario que indica la última actualización de la zona. Es recomendable que cuando modifiquemos algo pongamos la fecha con el siguiente formato dd-mm-aa.

Refresh: Tiempo en segundos en que el servidor secundario sincronizará la información consultando al servidor primario.

Retry: Tiempo en segundos en que esperara el servidor secundario para reintentar la consulta si antes hubo un error.

Expire: Tiempo máximo en segundos en intentar comunicarse el servidor secundario, en caso de no poder contactar con el primario.

TTL: Tiempo máximo en segundos en que los servidores de cache guardarán la información acerca del dominio.

Ejemplo:

1. \$TTL 604800
2. @ IN SOA dns.chickeneitor.com. root.chickeneitor.com. (
3. 27042007 ;Serial
4. 604800 ;Refresh
5. 86400 ;Retry
6. 2419200 ;Expire

7. 604800) ;Negative Cache TTL

4.7.2. A (Address)

Traducción de un dominio a una dirección IP. Se usa para establecer una dirección IP a un subdominio, y es el más usado. En pocas palabras se usa para crear la mayor parte de subdominios, el formato es el siguiente.

Host-Subdominio IN A Dirección IP

Ejemplo:

Dns	IN	A	165.23.1.201
www	IN	A	165.23.1.100
Mail	IN	A	165.23.1.200
ftp	IN	A	165.23.1.45

4.7.3. CNAME (Canonical Name)

Se usa para crear alias de un subdominio. Un subdominio puede tener varios alias o mejor dicho podemos tener varios subdominios apuntando a una sola dirección IP. Veamos un ejemplo.

Formato:

Host-Subdominio IN CNAME SUBDOMINIO.

Ejemplo:

www	IN	A	165.23.1.201
Website	IN	CNAME	www.chickeneitor.com.
Mail	IN	CNAME	www.chickeneitor.com.

Definimos primero un host con el registro **A**, después le asignamos dos alias, obsérvese que cuando escribimos un subdominio siempre tiene que terminar con un punto (.). El resultado final sería que `website.chickeneitor.com` y `mail.chickeneitor.com` estarían asociados a la misma dirección IP `165.23.1.201` por ser alias de `www.chickeneitor.com`.

4.7.4. NS (Name Server)

Especifica los servidores DNS de un dominio. Ya mencionamos que por cada Zona se configura un dominio, cada dominio debe tener asociado sus servidores DNS.

Ejemplo:

```
@      IN  NS  dns1.chickeneitor.com.
@      IN  NS  dns2.chickeneitor.com.
dns1   IN  A   165.23.1.128
dns2   IN  A   165.23.1.127
```

En las dos primeras líneas establecimos los dos DNS de nuestro dominio, después especificamos la dirección IP para cada subdominio.

4.7.5. MX (Mail Exchange)

Define los dominios en donde se alojará el correo, un dominio puede tener varios Host de correo en donde cada uno tendrá diferente prioridad.

Ejemplo:

```
@      IN  MX  0   mail.chickeneitor.com.
Mail   IN  A    165.23.1.136
```

Primero creamos el nuevo registro MX con el nombre del host o subdominio después establecemos una dirección IP con el registro A.

4.7.6. PTR (Pointer)

Inverso del registro A, es decir traduce de IPs a Nombres de dominios.

Ejemplo.

```
136.1.23.165      IN    PTR    mail.chickeneitor.com.
```

Cuando indicamos la IP del host esta tiene que estar inversa.

4.8 CREACIÓN DE UN DOMINIO LOCAL

Ahora vamos a crear un dominio local llamado "labredes.com" para nuestra LAN. Para empezar tenemos que modificar el archivo named.conf, ya había mencionado que aquí se definen las zonas. Editemos el archivo named.conf con el siguiente comando "#nano /etc/bind/named.conf".


```
GNU nano 2.0.7 File: /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local
include "/etc/bind/named.conf.options";

// prime the server with knowledge of the root servers
zone "." {
    type hint;
    file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912
zone "localhost" {
    [ Read 40 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell
```

Figura No 65. Edición del archivo named.conf

Después agregamos las siguientes líneas al final de las zonas predefinidas,

```
GNU nano 2.0.7 File: /etc/bind/named.conf
    type master;
    file "/etc/bind/db.0";
};
zone "255.in-addr.arpa" {
    type master;
    file "/etc/bind/db.255";
};
zone "labredes.com"{
    type master;
    file "/etc/bind/db.labredes.com";
};
include "/etc/bind/named.conf.local";
-
```

Figura No 66. Agregando las Zonas en el archivo named.conf

En donde **zone** es el nombre de la zona, en nuestro caso sería *"labredes.com"*, **type** se refiere si es maestro o esclavo, como es un servidor primario tendrá el valor de *master* y en **file** se indica el archivo de nuestra zona, para mantener un orden lo llame *db.labredes.com*, porque en realidad no es una regla que lleve el prefijo *db* podemos llamarle como nosotros queramos.

Archivo named.conf con la nueva zona.

```
1. include "/etc/bind/named.conf.options";
2.
3. zone "." {
4.     type hint;
5.     file "/etc/bind/db.root";
6. };
7.
8. zone "localhost" {
9.     type master;
10.    file "/etc/bind/db.local";
11. };
12.
13. zone "127.in-addr.arpa" {
14.    type master;
15.    file "/etc/bind/db.127";
16. };
17.
18. zone "0.in-addr.arpa" {
19.    type master;
20.    file "/etc/bind/db.0";
21. };
22.
23. zone "255.in-addr.arpa" {
24.    type master;
25.    file "/etc/bind/db.255";
26. };
27.
28. zone "chickeneitor.com" {
29.    type master;
30.    file "/etc/bind/db.labredes.com";
31. };
32.
33. // zone "com" { type delegation-only; };
34. // zone "net" { type delegation-only; };
```

- 35.
36. include "/etc/bind/named.conf.local";

Hasta aquí ya le indicamos a bind una nueva zona ahora para agregar la información tendremos que crear el archivo *db.labredes.com*. Este proceso se realiza creando una copia de uno de los ya existentes con el siguiente comando: “#cp /etc/bind/db.127 /etc/bind/db.labredes.com”.

```
root@us804:/# cp /etc/bind/db.127 /etc/bind/db.labredes.com
root@us804:/# ls /etc/bind
db.0      db.empty      db.root        named.conf.options
db.127    db.labredes.com  named.conf      rndc.key
db.255    db.local       named.conf.local zones.rfc1918
root@us804:/# _
```

Figura No 67. Creación del archivo *db.labredes.com*

Editamos el archivo con “#nano /etc/bind/db.labredes.com” y agregamos lo siguiente:

Archivo db.labredes.com

Código

1. \$TTL 604800
2. @ IN SOA dns.labredes.com. root.labredes.com. (
3. 27042007 ; Serial
4. 604800 ; Refresh
5. 86400 ; Retry
6. 2419200 ; Expire
7. 604800) ; Negative Cache TTL
8. ;
9. @ IN NS dns.labredes.com.
10. @ IN A 192.168.1.201
11. @ IN MX 0 mail.labredes.com.
12. dns IN A 192.168.1.200
13. www IN A 192.168.1.201

```

14. mail    IN  A    192.168.1.203
15. website IN  CNAME www.labredes.com.

```

```

GNU nano 2.0.7      File: /etc/bind/db.labredes.com      Modified
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      dns.labredes.com. root.labredes.com. (
                        27042707      ; Serial
                        604800        ; Refresh
                        86400         ; Retry
                        2419200       ; Expire
                        604800 )      ; Negative Cache TTL
;
@         IN      NS       dns.labredes.com.
@         IN      A        192.168.2.201
@         IN      MX 0     mail.labredes.com
dns       IN      A        192.168.2.200
www       IN      A        192.168.2.201
mail      IN      A        192.168.2.203
website  IN      CNAME    www.labredes.com_
@         IN      NS       localhost.
1.0.0     IN      PTR     localhost.

^G Get Help   ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos
^X Exit       ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell

```

Figura No 68. Configuración del archivo db.labredes .com

Luego de guardar los cambios y salir a la consola solo falta reiniciar bind para que los cambios surtan efecto, esto se logra con el comando: “#/etc/init.d/bind9 restart”.

```

root@us804:/# /etc/init.d/bind9 restart
* Stopping domain name service... bind      [ OK ]
* Starting domain name service... bind      [ OK ]
root@us804:/# _

```

Figura No 69. Comando para reiniciar el bind

Al comprobar en un pc la veracidad del servidor nos damos cuenta de su existencia:

```
C:\>nslookup
DNS request timed out.
    timeout was 2 seconds.
*** No se puede encontrar el nombre de servidor para la dirección 192.168.2.200:
    Timed out
*** Los servidores predeterminados no están disponibles
Servidor predeterminado: UnKnown
Address: 192.168.2.200

> www.labredes.com
Servidor: UnKnown
Address: 192.168.2.200

Nombre: www.labredes.com
Address: 192.168.2.201

> _
```

Figura No 70. Verificando la existencia del servidor

Capítulo No 5

LDAP y SAMBA



Capítulo No 5. LDAP y SAMBA

5.1 SAMBA

5.1.1. Instalación de Samba

Para hacer cualquier instalación de algún servicio como se describió en ocasiones anteriores, debemos hacer una actualización del sistema por completo para tener las últimas versiones de los paquetes que se encuentren disponibles para descarga. Utilizando el comando `#sudo apt-get update`, logramos este objetivo. Después de esto instalamos SAMBA con el siguiente comando: `#sudo aptitude install samba samba-client smbfs smbclient`, lo que realiza la descarga de los paquetes necesarios para ejecutar SAMBA en nuestro servidor.

```
cuc@us804:~$ sudo aptitude install samba samba-client smbfs smbclient
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Reading extended state information
Initializing package states... 0%
```

Después de que la máquina realiza la respectiva descarga, instalación e inicialización del servicio, SAMBA queda listo para ser configurado y suplir con las necesidades de cada administrador de servidores como se muestra en la siguiente figura.


```
account_policy_get: tdb_fetch_uint32 failed for field 8 (bad lockout attempt), r
returning 0
account_policy_get: tdb_fetch_uint32 failed for field 9 (disconnect time), retur
ning 0
account_policy_get: tdb_fetch_uint32 failed for field 10 (refuse machine passwor
d change), returning 0
Importing account for nobody...ok
Importing account for cuc...ok
* Starting Samba daemons [ OK ]

Configurando smbclient (3.0.28a-1ubuntu4.10) ...
Configurando smbfs (3.0.28a-1ubuntu4.10) ...
Processing triggers for libc6 ...
ldconfig deferred processing now taking place
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Reading extended state information
Initializing package states... Hecho
Writing extended state information... Hecho
Building tag database... Hecho
cuc@ubuntu:~$ * Reloading /etc/samba/smb.conf smbd only

cuc@ubuntu:~$
cuc@ubuntu:~$ _
```

Estos momentos el sistema instalado está listo para utilizar SAMBA. Lo primero que debemos hacer es configurar SAMBA editando el archivo que se encuentra en la ruta: `/etc/samba/smb.conf`, en este archivo se pueden cambiar las configuraciones por defecto o añadir algunas nuevas dependiendo de las necesidades de cada red.

Antes de editar el archivo de configuración, como se realizó en capítulos anteriores, se recomienda hacer una copia del original y protegerla contra escritura para tener las configuraciones originales como referencia y reutilizarlas si fuese necesario. Para tal efecto, se hace una copia de seguridad del archivo como se muestra en la figura a continuación con el siguiente comando: `# sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.copia`, si fuese necesario se debe especificar la clave de administrador para poder hacer la respectiva copia. Al final se muestra el archivo generado y aquellos que ya existían.

```
cuc@ubuntu:~$ cd /etc/samba/
cuc@ubuntu:/etc/samba$ ls
dhcp.conf  gdbcommands  smb.conf
cuc@ubuntu:/etc/samba$ sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.copia
[sudo] password for cuc:
cuc@ubuntu:/etc/samba$ ls
dhcp.conf  gdbcommands  smb.conf  smb.conf.copia
cuc@ubuntu:/etc/samba$ _
```

Ahora bien SAMBA tiene la principal característica de darnos la oportunidad de interactuar con el ambiente WINDOWS, por lo que veremos cómo hacer para compartir archivos con el servicio ya instalado.

Para esto debemos editar el archivo `smb.conf`, para esto utilizamos el editor de texto `nano` como sigue a continuación:

```
cuc@ubuntu:/etc/samba$ sudo nano smb.conf
[sudo] password for cuc: _
```

Si es necesario se debe dar la clave de administrador y luego se empieza la edición. En el siguiente ejemplo vamos a compartir una carpeta con el ánimo de poner a disposición de los usuarios de la red archivos de cualquier tipo.

```
GNU nano 2.0.7 File: /etc/samba/smb.conf
#
# Sample configuration file for the Samba suite for Debian GNU/Linux.
#
#
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options most of which
# are not shown in this example
#
# Any line which starts with a ; (semi-colon) or a # (hash)
# is a comment and is ignored. In this example we will use a #
# for commentary and a ; for parts of the config file that you
# may wish to enable
#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.
#
#=====Global Settings=====
[ Read 323 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell
```

Este archivo tiene secciones que 'direccionarán' la compartición de unidades e impresoras de Linux con máquinas Windows, el fichero smb.conf mostrado en esta sección es lo más simple posible, para lograr el objetivo planteado.

Cada sección del fichero empieza con una cabecera como [global], [impresoras], etc.

La sección [global] define unas pocas variables que Samba usará para definir la compartición de todos los recursos.

La sección [homes] permite a los usuarios remotos acceder a sus respectivos directorios principales en la máquina Linux local (cada uno al suyo nada más). Esto es, si un usuario de Windows intenta conectar a este recurso desde su máquina Windows, será conectado a su directorio personal, sin olvidar que para hacer esto, el usuario tiene que tener una cuenta en la máquina Linux.

El fichero smb.conf que vamos al que se le haran las respectivas modificaciones, es el original, la copia hecha con anterioridad sera guardada para efectos de daño irreparable en este archivo.

Lo que se propone hacer con el archivo es modificarlo de tal manera que todos los usuarios de las maquinas Windows puedan acceder a una carpeta compartida con el animo de acceder a cualquier tipo de recurso que el ella se encuentra alojado, para ello un usuario de Windows debe estar en la red local del servidor que tenga montado el SAMBA.

Con el archivo smb.conf abierto buscamos las líneas que contengan lo siguiente:

```
[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
printable = yes
guest ok = no
```

Estas lineas pueden son las encargadas de dar al demonio SAMBA las instrucciones de compartición de los recursos que se desean compartir. El anterior es un ejemplo de cómo compartir una impresora y lo utilizaremos de base para hacer la compartición de la carpeta que nosotros queremos hacer visible a todos los usuarios de nuestra red.

Antes de realizar este paso se debe crear una carpeta “compartidos” en la ruta /var/www, quedando como ruta final /var/www/compartidos, como se aprecia en la figura, esto puede variar según las necesidades del usuario por lo que se deja a preferencia del administrador del servicio SAMBA la ruta donde él la quiera crear.

```
cuc@us804:~$ cd /var/www/  
cuc@us804:/var/www$ ls  
ccna1 ccna2 compartidos index.html  
cuc@us804:/var/www$ _
```

Siguiendo con la edición del archivo `smb.conf`, después de las líneas de la configuración de la impresora que se mencionó anteriormente, escribimos una sección igual pero con las siguientes características:

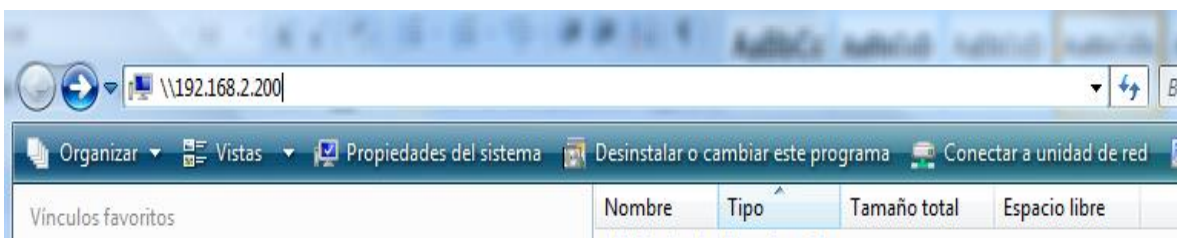
```
GNU nano 2.0.7 File: /etc/samba/smb.conf  
  
# Windows clients look for this share name as a source of downloadable  
# printer drivers  
[print$]  
  comment = Printer Drivers  
  path = /var/lib/samba/printers  
  browseable = yes  
  read only = yes  
  guest ok = no  
  
[ArchivosCompartidos]  
comment = Carpeta Publica  
path = /var/www/compartidos  
browseable = yes  
writable = yes  
guest ok = yes  
  
# Uncomment to allow remote administration of Windows print drivers.  
# Replace 'ntadmin' with the name of the group your admin users are  
# members of.  
; write list = root, @ntadmin  
  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

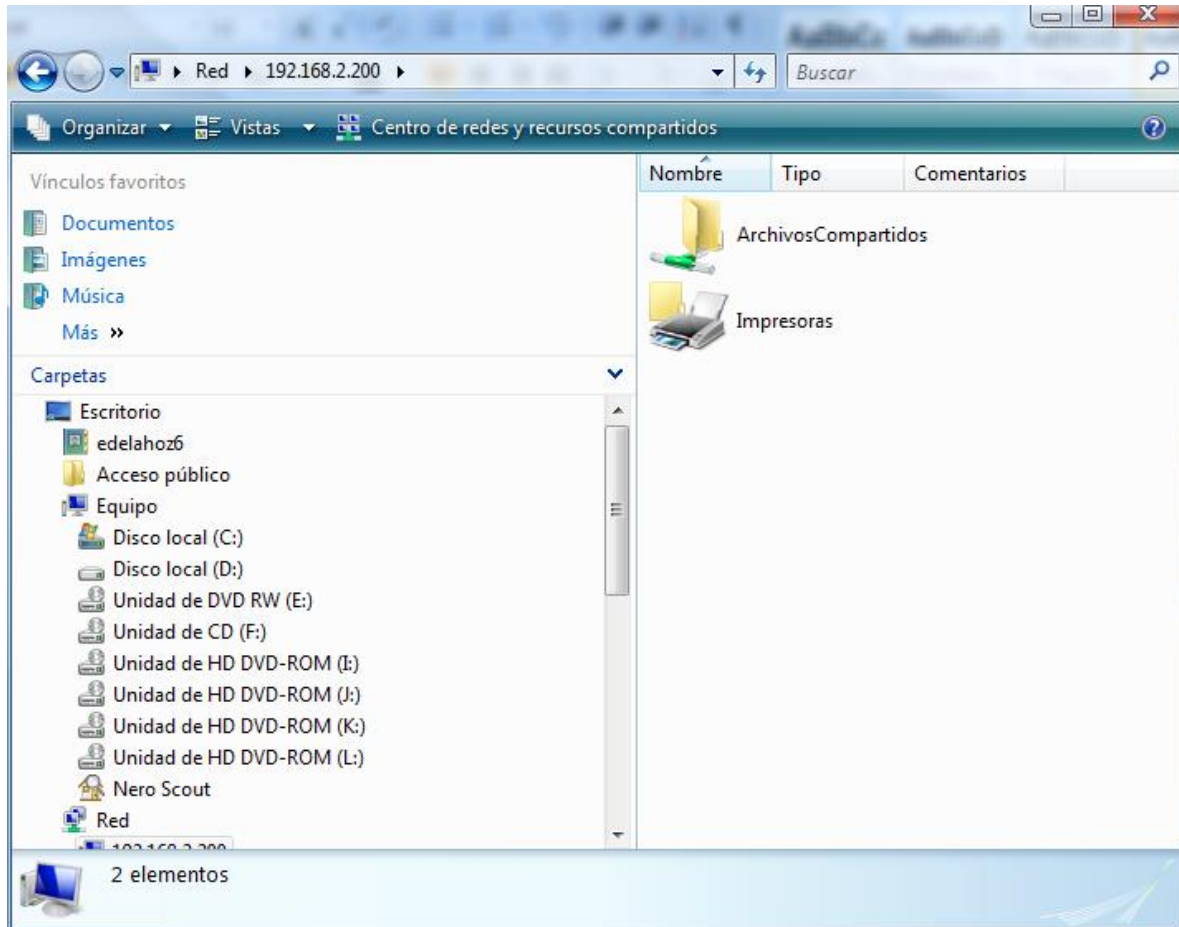
Dentro de la carpeta `compartidos` se pueden colocar tanto directorios como archivos con los permisos que se así se requieran como se muestra en la siguiente figura:

```
cuc@us804:/var/www/compartidos$ ls -l
total 24
drwxrwxrwx 5 cuc    cuc    4096 2010-03-18 06:03 accesoPublico
drwxr-xr-x 6 nobody nogroup 4096 2010-04-06 05:35 CCNA1
drwxr-xr-x 9 nobody nogroup 4096 2010-04-08 02:02 CCNA2
drwsrwsrwt 2 cuc    cuc    4096 2010-04-07 05:48 flash
drwxrwxrwx 4 root   root   4096 2010-04-06 14:33 LaboratorioDeRedes
drwxr-xr-x 3 nobody nogroup 4096 2010-03-19 05:43 Proyectos de I+D+i
cuc@us804:/var/www/compartidos$ _
```

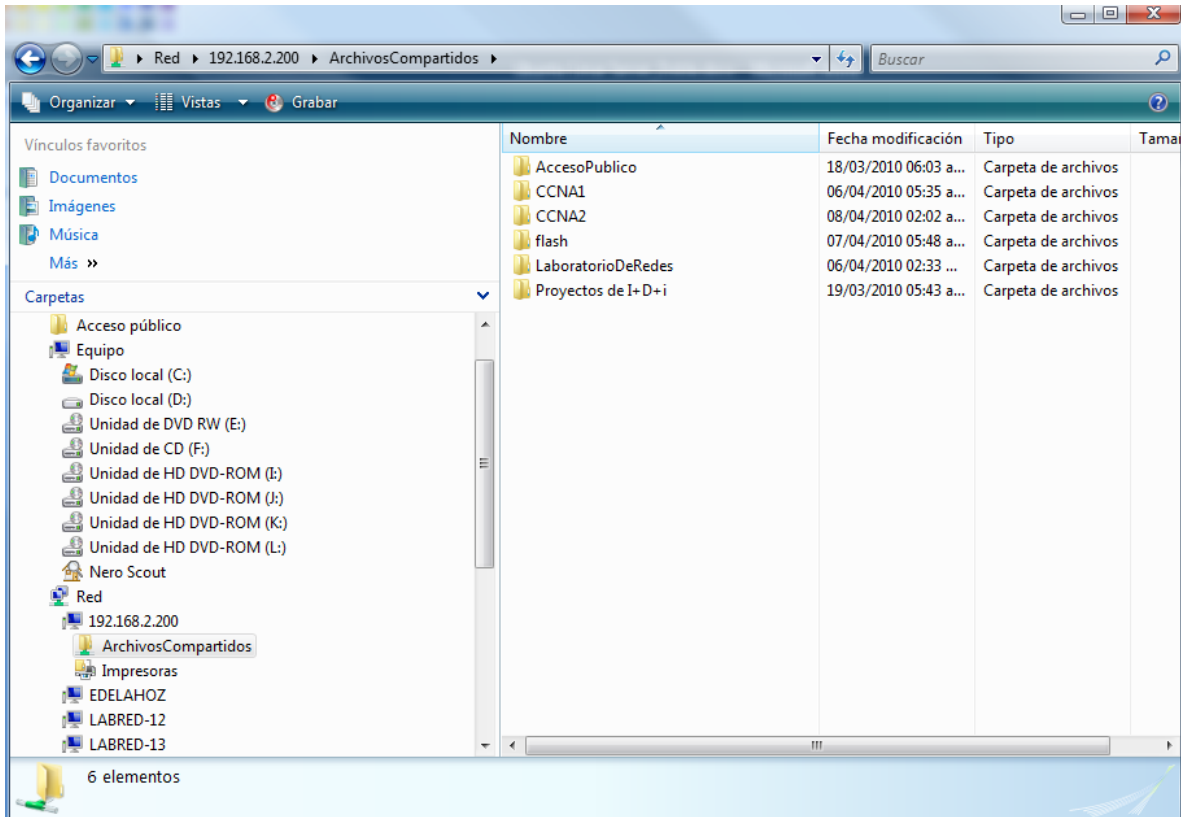
Después de realizar estas configuraciones previas podemos confirmar la compartición de los archivos desde un equipo con Windows para lo cual seguimos los siguientes pasos:

Primero se debe abrir una ventana de explorador de archivos y colocar la ruta de red: \\<dirección del servidor>, en nuestro caso: \\192.168.2.200 y aparecerá lo siguiente:





La carpeta “archivos compartidos” hace referencia al título de la sección que fue declarada en el archivo `smb.conf` del servicio SAMBA, dentro de ella estarán todas las carpetas que se mencionaron anteriormente.

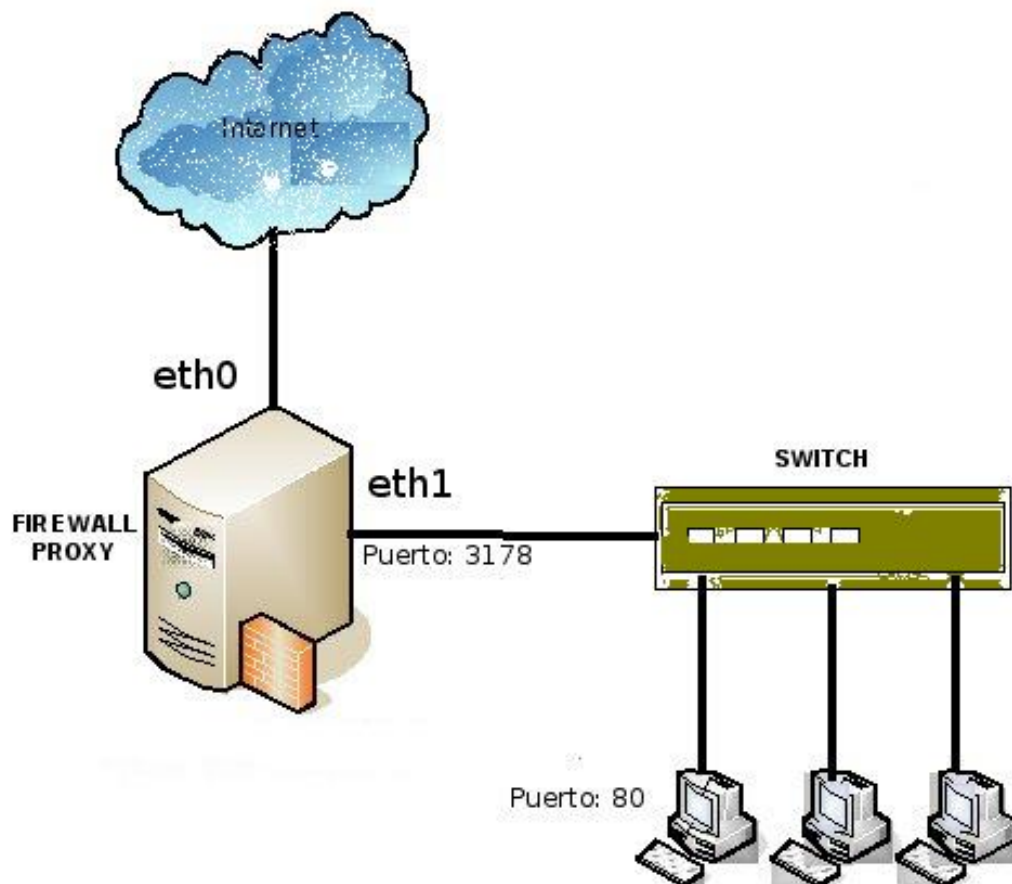


En estos momentos todos los equipos que pertenezcan a la red podrán acceder a esta carpeta a hacer uso de los archivos compartidos que en ella se encuentren.

Capítulo No 6

EL SERVICIO PROXY

PROXY



Capítulo No 6. EL SERVICIO PROXY

Cuando un equipo envía directamente una petición a un servidor http, sin la intermediación de un proxy y con el propósito de acceder a una página HTML, éste la busca en la Web y retorna la respuesta al equipo solicitante (cliente).



Figura No 71. Flujo de documentos sin Proxy instalado

Contrario a lo anterior, cuando se cuenta con un proxy, como intermediario de la comunicación entre un equipo cliente y un equipo servidor, el cliente envía una solicitud al servidor proxy, éste la busca en su memoria caché y si la encuentra, envía la respuesta al equipo cliente que la solicitó, en caso de no localizarla en la memoria cache, remite la petición al equipo servidor remoto, el cual a su vez efectúa una búsqueda en la web, retornando una respuesta al servidor proxy, una vez localizada la información. A su vez, el proxy almacena una copia de la información en su memoria cache para futuras búsquedas y reenvía dicha información al cliente.

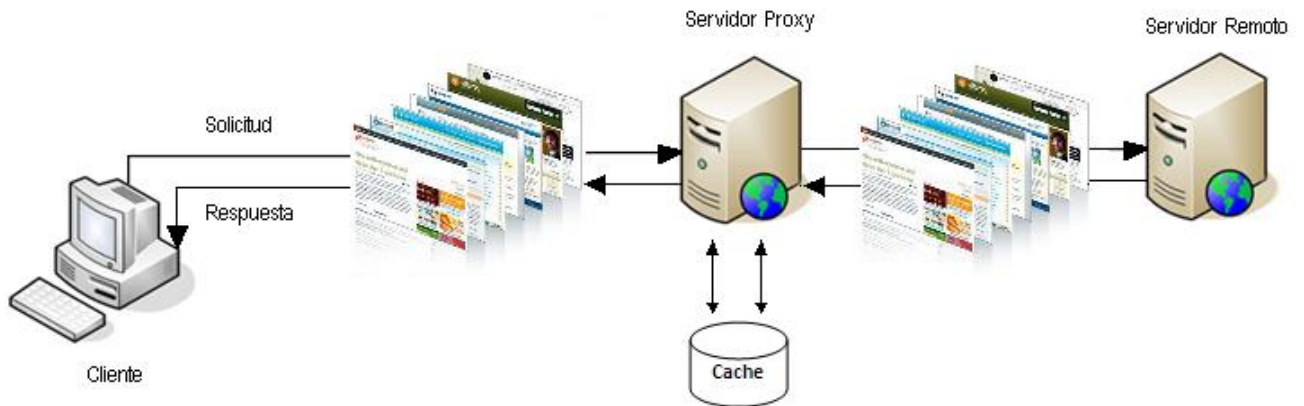


Figura No 72. Flujo de documentos con Proxy instalado

A partir de lo anteriormente enunciado, el proxy, que es referenciado comúnmente como un servidor, es un software que proporciona la funcionalidad necesaria para que varios computadores conectados a una Red de Área Local, obtengan acceso a Internet, gracias a la intermediación de un único computador. El uso más connotado que se le da al PROXY es el de intermediar entre la WEB y los clientes, propiciando la conexión directa a Internet, vía *WWW*, *HTTP*, conexión *Telnet*, *FTP* o cualquier tipo de protocolo empleado en Internet.

Los proxies presentan una serie de ventajas relacionadas con: el **control** de las peticiones de los clientes, el **ahorro** en la utilización de recursos de máquina, la **velocidad** en la resolución de peticiones gracias al uso de la memoria cache, el **filtrado** en el acceso a contenidos mediante la aplicación de listas de control de acceso, la **presentación** de información modificada mediante el uso de algoritmos diseñados para dicho propósito y el **anonimato** debido a que los clientes de una determinada red de área local acceden con un mismo identificador a la web.

Por otra parte presentan una serie de desventajas relacionadas con: el **abuso**, resolviendo peticiones no solicitadas, en el caso en que el sistema se encuentre sobrecargado, la **sobrecarga** debido a la multiplicidad de solicitudes emitidas por

los clientes en determinados momentos, la **intromisión**, situación que se presenta debido a que las peticiones deben pasar por el proxy y pueden existir usuarios que requieran confidencialidad en la información, la **incoherencia** mediante la presentación de información antigua, almacenada en cache no actualizada y la **irregularidad**, que implica la presentación de problemas de comunicaciones punto a punto, debido a que el proxy representa a más de un usuario cliente.

Existe una clasificación de los diferentes tipos de proxy, los cuales a su vez complementan sus funcionalidades en diferentes escenarios de aplicación, las implementaciones más comunes de proxy son: proxy WEB, proxy transparente, reverse proxy, proxy abierto y proxy NAT. Cada una de estas clasificaciones se describirá en detalle a continuación.

6.1 PROXY WEB (Proxy cache de web)

La funcionalidad de un proxy web radica en verificar el contenido de la navegación en internet, posibilitando la valoración de factores inherentes a la seguridad, rapidez, control de tráfico y anonimato. Como previamente se ha indicado, cuando se utiliza un software navegador o browser, las acciones realizadas en éste, primero pasan por el proxy, al cual se podrá acceder por medio de una dirección IP.

Un proxy WEB, adicional a la habitual utilidad de un proxy, suministra una cache para las páginas web visitadas y los recursos descargados, teniendo en cuenta que dichas páginas y recursos podrán ser posteriormente accedidas por los diferentes equipos que constituyen la red, liberando así, la carga de los enlaces por donde se transfieren los flujos de información y mejorando ostensiblemente los tiempos de acceso. En el gráfico siguiente se puede apreciar el funcionamiento del proxy web.

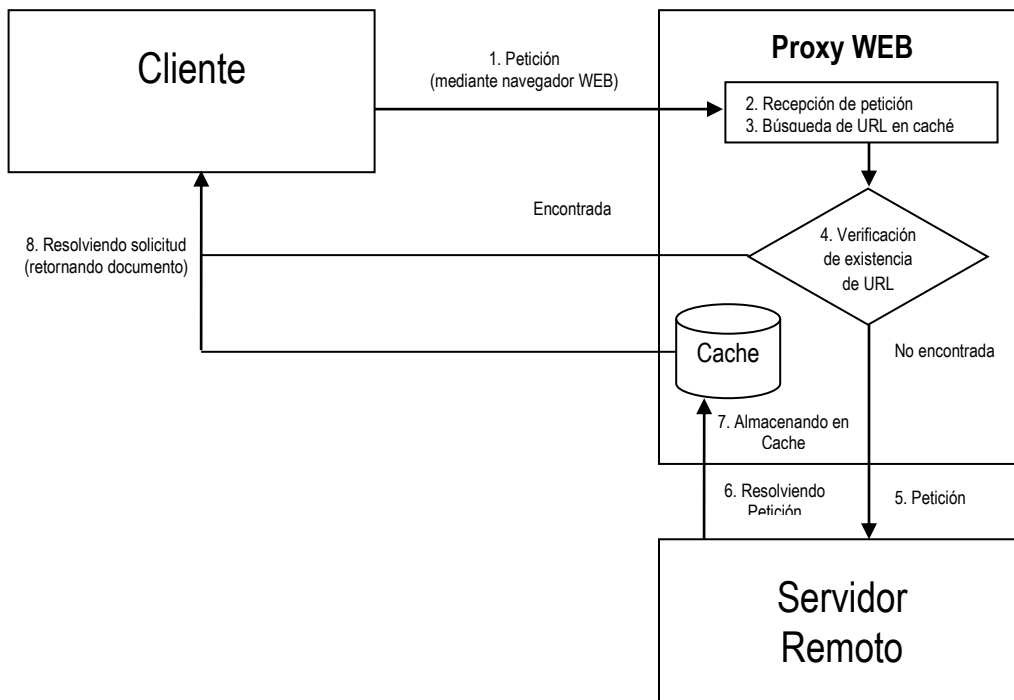


Figura No 73. Funcionamiento del Proxy WEB

Es importante resaltar, en referencia al gráfico anterior, que cuando el proxy identifica la inexistencia de la información buscada, en la caché, se envía la petición a un servidor remoto y este resuelve la petición, la cual es retornada al proxy y almacena en la memoria cache para referencias posteriores.

Teniendo en cuenta que la capacidad de almacenamiento de la memoria caché, es limitada, con el objeto de racionalizar el uso del recurso, se utiliza un algoritmo para identificar cuando las páginas almacenadas en ésta no están siendo usadas con regularidad, lo anterior, con el propósito de ser eliminadas de la caché, según su antigüedad, tamaño y frecuencia de acceso. Para ello se utilizando los algoritmos LRU (Least Recently Used) el usado menos recientemente y el LFU (Least Frequently Used) el usado menos frecuentemente.

Teniendo en cuenta que las páginas de uso frecuente tienen mayor probabilidad de uso y que de igual forma, es probable que las páginas que no hayan sido usadas durante mucho tiempo permanezcan sin ser usadas un prolongado período de tiempo, el algoritmo LRU utiliza el historial de acceso a las página para predecir lo que acontecerá respecto a su usabilidad.

Por otra parte el Algoritmo LFU lleva un contador de la cantidad de accesos o referencias que se hace de cada página, con el propósito de reemplazar la página con el menor valor de contador. Es importante resaltar que cuando una página inicialmente se usa con mucha regularidad y luego se deja de usar, presentará un alto valor en el contador de usos, tendiendo a permanecer durante un prolongado período de tiempo aunque ya no se requiera.

Tres importantes funcionalidades de los proxies web, son: el filtrado de contenidos, el cambio de formato de las páginas web y la protección contra virus. La primera consiste en bloquear el acceso a ciertas páginas web que contengan información que atente contra la seguridad de las organizaciones, la ética y los buenos principios; la segunda tiene como propósito proporcionar contenidos a audiencias específicas, mostrando páginas web en dispositivos móviles (Celulares y PDAs) y la finalidad de la tercera es brindar protección contra virus y diferentes contenidos que atenten contra los servicios de páginas web remotas.

6.2 PROXIES TRANSPARENTES

El propósito principal de un proxy transparente es combinar la funcionalidad de un servidor proxy con un NAT (Network Address Translation - Traducción de Dirección de Red), con el propósito de que las conexiones sean enrutadas en el proxy sin requerir configuración en el lado del cliente, incluso desconociendo el cliente la existencia del proxy. Los proxies transparentes son comúnmente usados

por los proveedores de servicios de internet (ISP) para prestar servicios a sus abonados (clientes).

Los proxies transparentes son utilizados por las empresas, en la promoción de políticas de uso de la red corporativa, proporcionando seguridad y mejorando los tiempos de respuesta en el acceso a las páginas web, mediante la implementación del almacenamiento caché. Además ofrecen múltiples ventajas, tales como: la configuración de cache inteligente, que implica el almacenamiento de imágenes y demás objetos que no requieren descargas continuas, el cacheo de contenido dinámico de páginas en formato php y asp, entre otras, la definición de restricciones en cuanto al tamaño de las descargas de recursos con considerable tamaño, la posibilidad de conexión con otros proxies y la facilidad de no requerir configuración adicional del lado del cliente.

6.3 REVERSE PROXY

Un Reverse Proxy, también conocido como proxy inverso, es un servidor proxy que, contrario a posibilitar el acceso a internet de los usuarios de la red corporativa, permite a usuarios de internet el acceso a servidores internos. Es usado como intermediador, por los usuarios de internet, que desean tener acceso a un sitio web alojado en un servidor interno de una determinada red corporativa.

Dos de sus mas connotadas funcionalidades son: el propender por la disminución de la carga de trabajo del servidor interno, haciendo uso de la memoria cache, razón por la cual es también denominado acelerador de servidor y brindar protección de ataques externos directos, fortaleciendo la red corporativa.

El Reverse Proxy utiliza algoritmos para el equilibrio de carga, distribuyendo la carga de trabajo mediante el redireccionamiento de las solicitudes efectuadas a otros servidores, éste proceso, se realiza con el propósito de hacer uso eficiente y

racional del recurso de comunicación. Para el equilibrio de carga se necesita efectuar traducciones de las URL externas a las URL internas correspondientes, identificando en qué servidor se encuentra la información solicitada.

Las principales razones para instalar un reverse proxy son la seguridad, la distribución o equilibrio de carga, el cache de contenido estático y el cifrado (aceleración SSL). El protocolo SSL (Security Sockets Layer) suministra servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente haciendo uso de un algoritmo de cifrado simétrico (RC4 o IDEA) para cifrar la clave de sesión, cifrando a su vez dicha clave mediante un algoritmo de cifrado de clave pública (Algoritmo RSA). La clave de sesión se usa para cifrar los datos que proceden o viajan hacia el servidor. El proceso consiste en generar una clave de sesión, distinta para cada transacción, posibilitando esto, que aunque la clave sea descifrada por un intruso, no pueda ser utilizada en transacciones futuras. El cifrado SSL es realizado por el reverse proxy, debido a que éste está equipado con un hardware de aceleración SSL, que cumple tal finalidad.

6.4 PROXY NAT (NETWORK ADDRESS TRANSLATION)

La funcionalidad de un Proxy NAT radica en mapear o “traducir” direcciones IP de una red privada a direcciones IP de una red pública o viceversa. Este proceso se realiza debido a que el número de direcciones públicas existentes para acceder a internet es insuficiente para suplir la demanda de equipos que requieren este tipo de conexión, por ello mediante el NAT se efectúa la asociación de una o varias direcciones IPs de una red corporativa a respectivas direcciones IPs externas. Este proceso es muy común en organizaciones con Redes privadas constituidas por gran cantidad de equipos de cómputo a los cuales se les han asignado direcciones IPs en un dominio privado y que desean acceder a servicios externos mediante un Servidor Proxy NAT, haciendo uso de un limitado número de

direcciones IPs públicas, por supuesto el proceso de traducción de las IPs privadas a las IPs públicas es totalmente transparente para el usuario.

Cada paquete de datos que viaja por la red posee una cabecera que posibilita su identificación. En dicha cabecera se almacenan ciertos campos como la dirección IP origen y la dirección IP destino del paquete, cuando el paquete de datos pasa por el proxy NAT, el campo dirección IP origen cambian, asignándole el valor correspondiente a la dirección IP pública y además se almacena en un tabla la traducción efectuada, para futuras referencias en caso de que se requiera enviar un paquete de respuesta desde el exterior hacia la red corporativa.

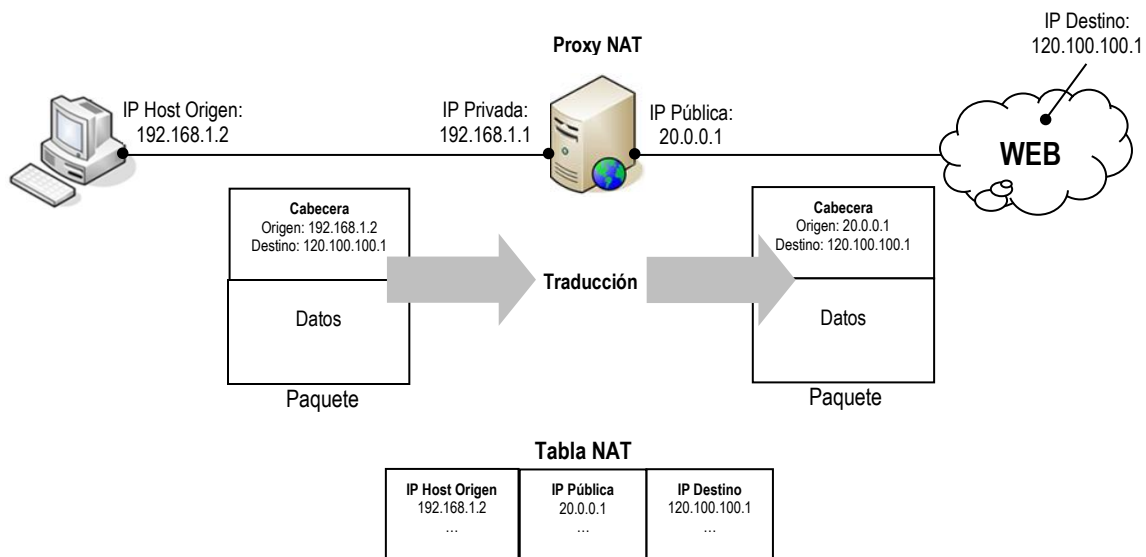


Figura No 74. Funcionamiento del Proxy NAT

De acuerdo a la forma como se asocian las direcciones públicas a las direcciones privadas, existe una clasificación de los NAT, en Estáticos, Dinámicos y NAPT (Network Address Port Translation), cada uno de ellos se describe a continuación, detallando un ejemplo de su proceso de configuración.

6.4.1. NAT Estático

Consiste en la asignación una a una de direcciones de red externas o públicas por cada dirección de red interna o privada. Este proceso de asignación de direcciones locales individuales, mapeadas a direcciones globales específicas, se emplea para garantizar el acceso hacia afuera de la red corporativa o para posibilitar el acceso al equipo local desde equipos externos a la red corporativa, mediante una dirección pública fija. Es importante anotar que si existen varios puntos de salida en la red, cada proxy NAT, ubicado en dichos puntos, deberá poseer una actualización sincronizada de la tabla de traducción.

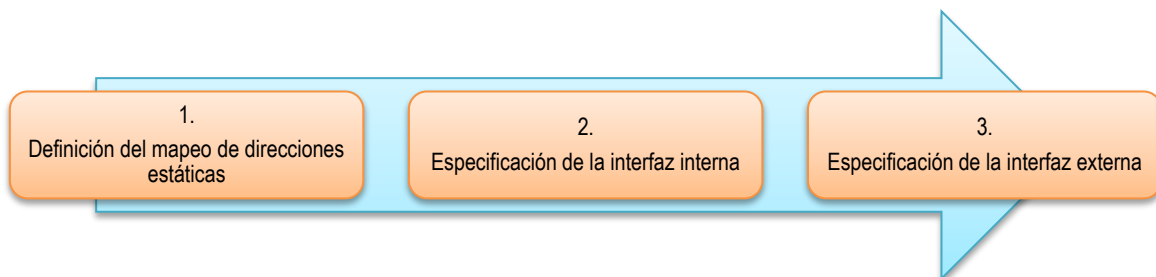


Figura No 75. Configuración de NAT Estático

En el esquema siguiente, se aprecia una topología de red en la cual se puede efectuar un proceso de configuración de NAT estático, obsérvese en él una Dirección IP privada, en el extremo de la red corporativa (192.168.1.1), asociada a una interfaz interna y una dirección IP pública, hacia la WEB, asociada a una interfaz externa.

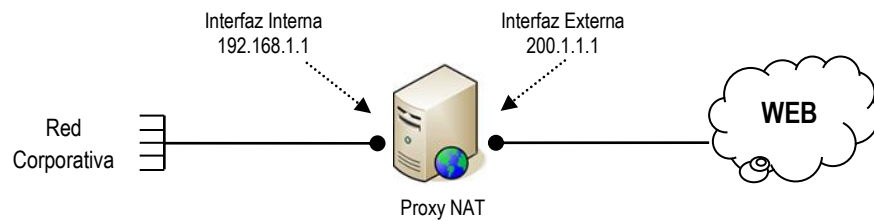


Figura No 76. Esquema de Ejemplo para Configuración de NAT Estático

6.4.2. NAT Dinámico

Consiste en configurar el mapeo de un pool o dominio de direcciones IP públicas que se asociarán a un conjunto de direcciones IP privadas, teniendo en cuenta que la asignación una a una de direcciones externas a direcciones internas se hace de forma dinámica de acuerdo a solicitud del servicio y cada vez que un par de direcciones (privada y pública) es liberado, puede ser reasignado otro par en el que estén involucrados diferentes direcciones (privada y pública).

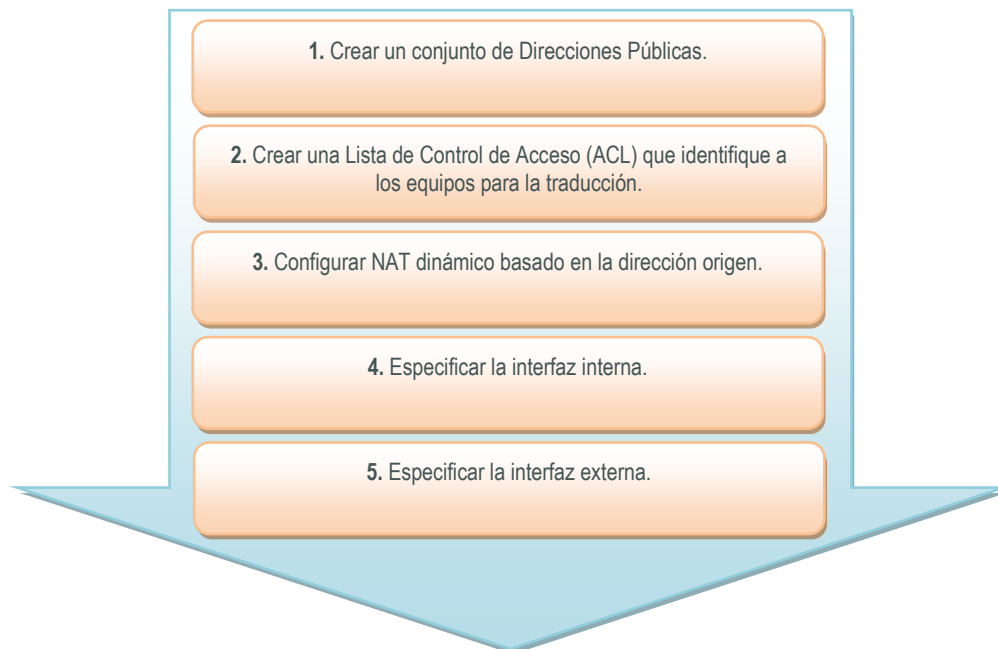


Figura No 77. Configuración de NAT Dinámico

En el esquema siguiente, se aprecia una topología de red en la cual se puede efectuar un proceso de configuración de NAT dinámico, obsérvese en él un conjunto de direcciones IP en el rango 192.168.1.0/24 y en el extremo de dicha red una interfaz interna y por otra parte, un conjunto (pool) de Direcciones IP públicas, hacia la WEB y una interfaz externa.

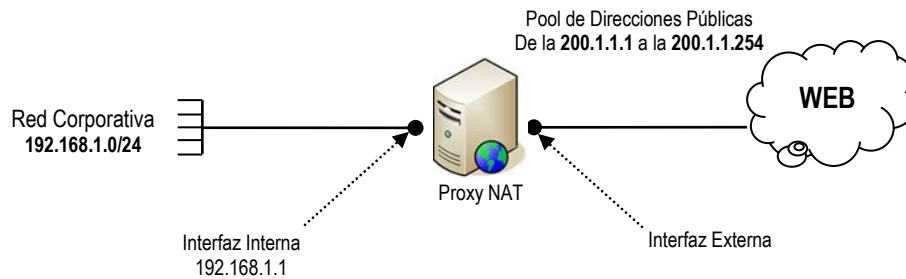


Figura No 78. Esquema de Ejemplo para Configuración de NAT Dinámico

6.4.3. NAPT (Network Address Port Translation)

La traducción de direcciones de red y puertos, consiste en permitir el acceso simultáneo a un número plural de equipos que constituyen una red interna, hacia el exterior de dicha red, haciendo uso de una única dirección IP pública, posibilitando la asociación de tuplas del tipo, dirección IP local - número de puerto, a tuplas del tipo, dirección IP pública - número de puerto. Este tipo de Proxy NAT es utilizado en organizaciones que poseen una red IP privada (LAN Red de Área Local) y una conexión WAN (Red de Área Extensa) a un Proveedor de Servicio Internet (IPS). La finalidad de usar este tipo de configuración NAT es emplear una sola dirección IP pública asignada por el IPS, para mapear del tráfico externo de toda una red corporativa.

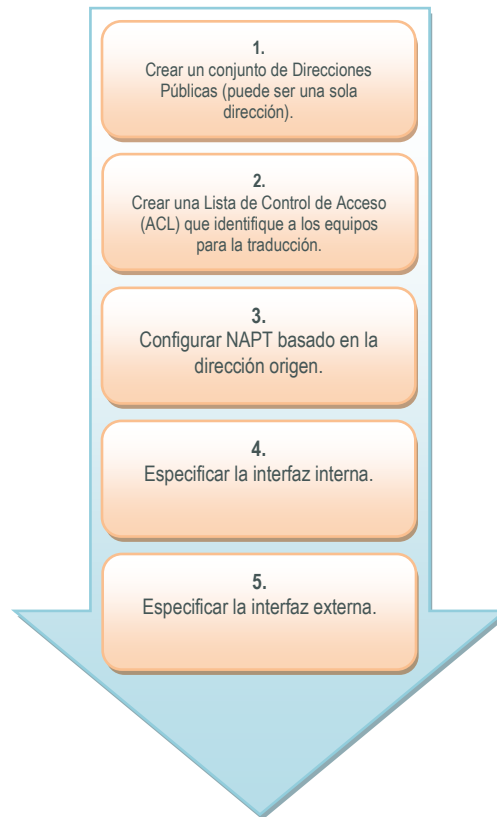


Figura No 79. Configuración de NAPT

En el esquema siguiente, se aprecia una topología de red en la cual se puede efectuar un proceso de configuración de NAPT, obsérvese en él un rango de direcciones IP 192.168.1.0/24 y en el extremo de dicha red una interfaz interna y por otra parte, un conjunto (pool) de veinte (20) Direcciones IP públicas, hacia la WEB y una interfaz externa.

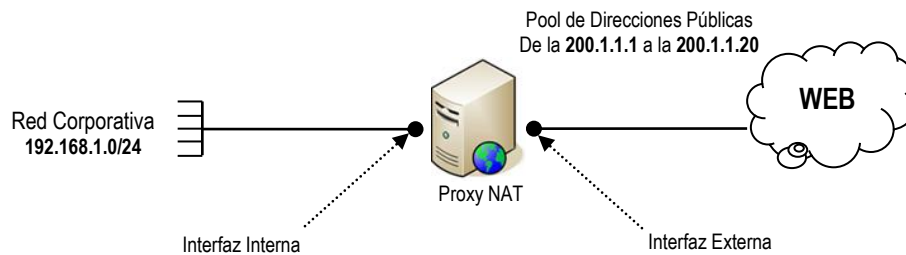


Figura No 80. Esquema de Ejemplo para Configuración de NAPT

6.5 PROXY ABIERTO

Un proxy abierto acepta peticiones desde cualquier ordenador, esté o no conectado a su red, gracias a su configuración el proxy ejecutará peticiones de los ordenadores conectados a él, como si fueran peticiones propias del proxy. Por lo anterior, este tipo de proxy es usado para el envío masivo de correos (spam), como pasarela. Un proxy se usa, normalmente, para almacenar y redirigir servicios, mediante el almacenamiento de peticiones en el servidor proxy, lo que mejora la velocidad general de los usuarios, convirtiéndolo en un servicio supremamente útil, sin embargo, con una configuración abierta a la WEB, puede usarse para fines indebidos. Por esta razón, la gran mayoría de servidores, como los de IRC (Internet Relay Chat – Servidores de Mensajería Instantánea), o servidores de correo electrónico, impiden el acceso a estos proxys a sus servicios, mediante el uso de las llamadas BlackList o listas negras.

6.6 SERVIDOR PROXY SQUID

Existen muchos servidores Proxy para Linux. Una solución muy popular es el módulo de PROXY de Apache, por otra parte, una implementación más completa y robusta de un Proxy *HTTP* es Squid, el servidor Proxy más popular y extendido entre los sistemas operativos basados en UNIX®. Es muy confiable y versátil. Al ser *software libre*, además de estar disponible el código fuente, está libre del pago de costosas licencias o con restricción a un uso con determinado número de usuarios. Squid será utilizado en este caso para instalar y configurar un servidor Proxy. El servidor Proxy cuenta con una caché, en la cual se guarda una copia de cada página web que se visita, de tal forma que si otra máquina solicita una página a la que ya ha entrado el servidor, el acceso será mucho más rápido, ya que se obtendrá directamente desde el Proxy, sin necesidad de conectarse al

servidor original. Esto permite un uso eficiente del ancho de banda y un menor tiempo de respuesta. Como efecto colateral, las máquinas clientes no están directamente conectadas al exterior, esta es una forma de incrementar la seguridad de la red interna. Un Proxy bien configurado puede ser tan efectivo como un buen firewall. El servidor Proxy puede ser configurado para servir de intermediario de varios protocolos utilizados por las aplicaciones de Internet, tales como *HTTP*, *FTP*, *Telnet*, *socks*, *irc*, *real audio*, etc. Es importante resaltar que no todos estos protocolos son soportados por Squid. En particular Squid soporta los siguientes protocolos:

- Hyper Text Transfer Protocol (HTTP), que es el protocolo de transferencia de hipertexto en el que está basado la WWW.
- File Transfer Protocol (FTP) o protocolo de transferencia de archivos.
- Gopher
- Wide Area Information (WAIS)
- Secure Socket Layer (SSL) - el cual es usado para transacciones en línea seguras.

La función de Proxy en Linux puede ser también reemplazada con un *Firewall*, pero los *Firewall* tienen la gran desventaja (en la mayoría de los casos) de no manejar caché en disco, por lo que cada página solicitada por los clientes en la red es traída desde Internet en todos los casos así haya sido cargada recientemente.

A continuación detallaremos el proceso paso a paso de configuración de un servidor Proxy Squid.

6.7 INSTALACIÓN Y CONFIGURACIÓN PROXY SQUID

Aptitude -r install squid

```
root@us804:~# aptitude -r install squid
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Initializing package states... Hecho
Writing extended state information... Hecho
Building tag database... Hecho
The following NEW packages will be automatically installed:
  openssl openssl-blacklist squid-common ssl-cert
The following packages have been kept back:
  libc6 libc6-i686 libcurl3-gnutls libdbus-1-3 libgnutls13
  libhtml-parser-perl libnewt0.52 libssl0.9.8 linux-image-2.6.24-24-server
  linux-image-server linux-server linux-ubuntu-modules-2.6.24-24-server
  python2.5 python2.5-minimal sudo tzdata wget whiptail
The following NEW packages will be installed:
  openssl openssl-blacklist squid squid-common ssl-cert
0 packages upgraded, 5 newly installed, 0 to remove and 18 not upgraded.
Need to get 7856kB of archives. After unpacking 19,7MB will be used.
Do you want to continue? [Y/n/?] Y_
```

Figura No 81. Proceso de Instalación del SQUID

Después de descargar el paquete, podemos empezar a configurar nuestro servidor proxy, pero primero debemos hacer una copia de respaldo de este archivo para evitar pérdida de información en caso de que lo dañemos el momento de la configuración. Utilizamos entonces el comando “#cp /etc/squid/squid.conf /etc/squid/squid.conf_copia” y realizamos la copia:

```
root@us804:~# cp /etc/squid/squid.conf /etc/squid/squid.conf_copia
root@us804:~# ls /etc/squid/
squid.conf  squid.conf_copia
root@us804:~# _
```

Figura No 82. Creación de una copia del archivo de configuración squid.conf

Después de esto podremos modificar nuestro archivo de configuración para nuestras necesidades. Squid utiliza el archivo de configuración localizado en “#/etc/squid/squid.conf”, y se puede editar con el siguiente comando, en consola (terminal) *nano /etc/squid/squid.conf*.

```
GNU nano 2.0.7 File: /etc/squid/squid.conf
#
# WELCOME TO SQUID 2.6.STABLE18
#
# -----
#
# This is the default Squid configuration file. You may wish
# to look at the Squid home page (http://www.squid-cache.org/)
# for the FAQ and other documentation.
#
# The default Squid config file shows what the defaults for
# various options happen to be. If you don't need to change the
# default, you shouldn't uncomment the line. Doing so may cause
# run-time problems. In some cases "none" refers to no default
# setting at all, while in other cases it refers to a valid
# option - the comments for that keyword indicate if this is the
# case.
#
#
# OPTIONS FOR AUTHENTICATION
# -----
#
[ Read 4529 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figura No 83. Edición del archivo squid.conf

Antes de comenzar con la configuración, hemos de prestar mucha **atención** para evitar dejar **espacios vacíos** en lugares indebidos. El siguiente es un ejemplo de cómo **NO** debes comentar un parámetro.

Opción **incorrectamente** descomentada

```
http_port 3128
```

El siguiente es un ejemplo de cómo **SÍ** debe descomentarse un parámetro.

Opción **correctamente** descomentada

```
http_port 3128
```

6.8 PARÁMETROS BÁSICOS

6.8.1. Http_port

Squid, por defecto, utilizará el puerto 3128 para atender peticiones de los clientes; sin embargo, se puede especificar que lo haga en cualquier otro puerto o bien que lo haga en varios puertos a la vez.

En el caso de un *Proxy Transparente*, regularmente se utilizará el puerto 80 y se valdrá del re direccionamiento de peticiones de modo tal que no habrá necesidad alguna de modificar la configuración de los navegadores Web para utilizar dicho servidor proxy. Bastará con utilizar el proxy como puerta de enlace.

Es importante recordar que los servidores WEB, como Apache, también utilizan dicho puerto, por lo que será necesario reconfigurar el servidor HTTP que se encontrará instalado en la misma máquina para utilizar otro puerto disponible, o bien desinstalar o deshabilitar el servidor HTTP.

Hoy en día ya no es del todo práctico el utilizar un *Proxy Transparente*, a menos que se trate de un servicio de *Café Internet* u oficina pequeña, siendo uno de los principales problemas con los que lidian los administradores el mal uso y/o abuso del acceso a Internet por parte del personal. Es por esto que puede resultar más conveniente configurar un servidor proxy con restricciones por contraseña, lo cual no puede hacerse con un *Proxy Transparente*, debido a que se requiere un diálogo de nombre de usuario y contraseña.

Algunos programas utilizados comúnmente por los usuarios suelen traer por defecto el puerto 8080 -servicio de cacheo WWW- para utilizarse al configurar el servidor proxy. Si queremos aprovechar esto a nuestro favor y ahorrarnos el tener

que dar explicaciones innecesarias al usuario, podemos especificar que *SQUID* escuche peticiones en dicho puerto también. Para hacer esto, podríamos dejar así la sección correspondiente del archivo de configuración.

```
#  
# You may specify multiple socket addresses on multiple lines.  
#  
# Default: http_port 3128  
http_port 3128  
http_port 8080
```

Si se desea incrementar la seguridad, puede vincularse el servicio a una IP que sólo se pueda acceder desde la red local. Considerando que el servidor utilizado posee una IP 192.168.1.254, puede hacerse lo siguiente:

```
#  
# You may specify multiple socket addresses on multiple lines.  
#  
# Default: http_port 3128  
http_port 192.168.1.254:3128  
http_port 192.168.1.254:8080
```

6.8.2. Cache_mem

El parámetro `cache_mem` establece la cantidad de memoria principal a ser usada por:

- Objetos en tránsito.
- Objetos Hot.
- Objetos negativamente almacenados en el caché.

Los datos de estos objetos se almacenan en bloques de 4 Kb. El parámetro `cache_mem` especifica un límite máximo en el tamaño total de bloques

acomodados, donde los objetos en tránsito tienen mayor prioridad. Sin embargo los objetos *Hot* y aquellos negativamente almacenados en el caché podrán utilizar la memoria no utilizada hasta que esta sea requerida. De ser necesario, si un objeto en tránsito es mayor a la cantidad de memoria especificada, *SQUID* excederá lo que sea necesario para satisfacer la petición.

Por defecto se establecen 8 MB. Pudiendo especificarse una cantidad mayor si así se considera necesaria, dependiendo esto de los hábitos de los usuarios o necesidades establecidas por el administrador.

En un servidor con al menos 128 MB de RAM, se deberían establecer 16 MB como valor para este parámetro:

```
cache_mem 16 MB
```

6.8.3. Cache_dir

Este parámetro se utiliza para establecer qué tamaño se desea tenga el caché en el disco duro para Squid. Si encontramos la siguiente línea, estaremos reservando 100 MB de disco duro para caché.

```
cache_dir ufs /var/spool/squid 100 16 256
```

Se puede incrementar el tamaño del caché hasta donde lo desee el administrador. Cuanto más grande sea el caché, más objetos se almacenarán en éste y, por lo tanto, se utilizará menos ancho de banda de conexión a Internet.

La siguiente línea establece un caché de 700 MB:

```
cache_dir ufs /var/spool/squid 700 16 256
```

Los números 16 y 256 significan que el directorio del cache contendrá 16 subdirectorios con 256 niveles cada uno. Estos dos últimos números es mejor no modificarlos. Es muy importante considerar que si, se especifica un determinado tamaño de caché y éste excede al espacio real disponible en el disco duro *Squid* se bloqueará inevitablemente. Por lo tanto, hay que tener mucho cuidado a éste respecto.

6.8.4. Control de acceso

Es posible reconfigurar Squid para restringir el acceso al Proxy y así obtener un servidor más robusto. Estas restricciones se pueden lograr de múltiples maneras: restringir conjuntos de máquinas por direcciones *IP*, restringir máquinas individuales, restringir la carga de páginas específicas, restringir la carga de páginas con cierto texto en las *URLs*, restringir el acceso por horas y días, etc. Esto se hace posible gracias al uso de los ACL (Access Control List).

Al establecer *Listas de Control de Acceso* se puede definir una red o bien, ciertas máquinas en particular, teniendo en cuenta que a cada lista se le asignará una *Regla de Control de Acceso* que permitirá o denegará el acceso a Squid. Para mayor ilustración a continuación se desarrolla en detalle la temática correspondiente a las listas de control de acceso.

6.8.5. Listas de Control de Acceso - ACL

Las Listas de Control de Acceso – ACL, son utilizadas por los administradores de redes informáticas tanto para limitar en su totalidad el tráfico de la red, como para conceder acceso sólo a un tráfico específico. Posibilitando en gran forma el control de la seguridad de la red. Las ACLs son secuencias de instrucciones que aplicadas a direcciones o protocolos con el propósito de permitir o denegar servicios de red.

Las ACLs son utilizadas no solo para permitir o denegar el acceso de los equipos host a determinadas direcciones de red, su funcionalidad posibilita incluso controlar el tráfico de red de acuerdo al puerto TCP que utiliza dicho tráfico. TCP hace parte de la familia de protocolos TCP/IP los cuales trabajan de la siguiente forma: TCP es el encargado de establecer el proceso de comunicación entre la aplicación origen (host) y la aplicación destino (servidor), funciona dividiendo el conjunto de datos en paquetes, de la aplicación origen e integrando los paquetes recibidos, del lado de la aplicación destino. El protocolo IP, por su parte, es el encargado del proceso de comunicación entre el host y el servidor.

En cada paquete de datos que genera TCP, se puede identificar el servicio que se solicita, el cual puede ser: transferencia de archivos, conexión tipo telnet, Messenger, entre otros. Y cada uno de estos servicios tiene asociado un puerto, a continuación la identificación de los puertos más usados, con su respectiva clasificación.

Denominación	Rango	TCP	UDP	Comunes TCP-UDP
Puertos Comunes	0 – 1023	21 – FTP 23 – Telnet 25 – SMTP 80 – HTTP 110 – POP3 194 – IRC 443 – HTTPS	69 – TFTP 520 – RIP	53 – DNS 161 – SNMP 531 – AOL Instant Messenger, IRC
Puertos Registrados	1024 – 49151	1863 – MSN 8008 – HTTP Alternativo 8080 – HTTP Alternativo	1812 – Protocolo de autenticación RADIUS	1433 – MS SQL 2948 – WAP (MMS)
Puertos Dinámicos	49152 - 65535		5004 – RTP (Protocolo de transporte de voz y video) 5060 – SIP (VoIP)	

El control de acceso a la red es en gran medida regulado gracias al filtrado de paquetes, proceso que se da en la capa de red del modelo OSI o en la capa de Internet del modelo TCP/IP y que consiste en extraer de cada paquete, que pasa

por el enrutador o firewall, información que permita tomar la decisión respecto a la autorización o denegación del acceso del paquete a la red. Para conceder o denegar el acceso de los paquetes a la red, es necesario verificar que éstos cumplan con unas reglas definidas mediante Listas de Control de Acceso ACLs, donde dichas reglas evalúan el direccionamiento IP origen y destino, el número de puerto origen y destino (identificando el servicio requerido) y el protocolo que indica el paquete.

Los paquetes son valorados con el propósito de conceder o denegar el acceso de éstos a la red, utilizando como criterios de valoración los definidos en las reglas contenidas en las ACLs, éstos son:

- Dirección IP de origen
- Dirección IP de destino
- Tipo de mensaje ICMP
- Puerto TCP/UDP de origen
- Puerto TCP/UDP de destino

Por omisión el tráfico que ingresa a los enrutadores o firewalls no es filtrado, debido a que éstos dispositivos no tienen configuradas de manera predeterminada ninguna ACL. Después de haber sido definidas un conjunto de ACLs, en estos dispositivos, y cuando los paquetes intentan acceder a la red a través de una determinada interfaz, las ACLs son revisadas de arriba abajo, instrucción por instrucción, detectando si la información contenida en el paquete coincide con los patrones de las ACLs. Las ACLs son empleadas tanto para controlar el tráfico entrante y saliente de una red interna y una red externa, como para controlar este mismo tráfico entre dos segmentos de red interna.

Las ACLs pueden configurarse por protocolo, por dirección o por interfaz. En el primer caso, es importante definir, para cada protocolo de la interfaz, una ACL y de esa forma se podrá controlar el tráfico de dicha interfaz; en el segundo caso, se

deben crear dos ACLs para controlar el tráfico de una interfaz, es decir, una ACL para el tráfico entrante y una ACL para el tráfico saliente (de acuerdo a la dirección del tráfico); y en el tercer caso, se define una ACL por cada interfaz física.

El número de ACLs que deben definirse en el router o firewall se determina a partir de los tres criterios anteriormente mencionados, por ejemplo: si se tiene un dispositivo con tres interfaces, las cuales serán configuradas para IP e IPX. El total de ACLs a configurar es 12, valor que se deduce de multiplicar 3 interfaces por 2 direcciones (entrante y saliente) por dos protocolos (IP e IPX).

Las ACLs se pueden definir para controlar las siguientes tareas:

Tareas	Ejemplo
Acción: Limitar el tráfico Reacción: Mejorar el rendimiento	Se puede configurar la red de la empresa de tal forma que no se pueda enviar tráfico de video y de esta forma se disminuye significativamente el tráfico de la red y se mejora considerablemente el rendimiento de la misma.
Control de flujo de tráfico	En algunas redes, de acuerdo a sus requerimientos, en cuanto a actualizaciones de enrutamiento, se puede restringir el envío de las mismas, optimizando el uso del ancho de banda de la red.
Seguridad en el acceso a la red	Se puede definir mediante ACLs que ciertos usuarios accedan a un área específica de la red, por ejemplo la unidad de recursos financieros y que no tengan acceso a otros segmentos de la red.
Clasificación del tráfico a enviar o bloquear	Se pueden definir reglas mediante ACLs, en las que se permita el tráfico de correo electrónico y se bloquee el tráfico Telnet.

6.8.5.1. Funcionamiento de las ACLs

Las ACL se definen para evaluar los paquetes entrantes y salientes del dispositivo donde son configuradas, las ACLs no actúan sobre los paquetes que se generan en el dispositivo. De acuerdo a lo anterior, existe una clasificación de las ACL en

ACLs de entrada y ACLs de salida. Las primeras procesan los paquetes antes de ser éstos enrutados a la interfaz de salida, en el caso en que los paquetes sean autorizados, se procede con su enrutamiento, en caso contrario son descartados y cuando esto ocurre se almacena en el dispositivo la referencia de la búsqueda, para futuras evaluaciones; en las segundas los paquetes son procesados mediante ACLs luego de ser enrutados a la interfaz de salida.

Las ACLs son evaluadas una por una, de forma secuencial de arriba hacia abajo. El funcionamiento de las ACLs es el siguiente: en primera instancia se evalúa la cabecera del paquete confrontando la información contenida en éste con la ACL, si la información coincide no se evalúa el resto de las ACLs y el paquete es permitido o descartado dependiente de su evaluación, en caso de que la información contenida en el paquete no coincida con la primera ACL, se continua el proceso de verificación con las siguientes ACLs de la lista, una a una. Al final de la lista de ACLs se define una sentencia por omisión que descarta todos los paquetes cuya evaluación no ha coincidido con la lista de ACLs predecesoras, la sentencia por omisión es **deny all traffic**.

6.8.5.2. Clasificación de las ACLs

6.8.5.2.1. Filtrado de paquetes de acuerdo a la Dirección Origen

Este tipo de ACLs autorizan o descartan paquetes de acuerdo al direccionamiento IP origen de donde proceden los paquetes, sin tener en cuenta el direccionamiento IP destino del paquete, ni el puerto asociado al tráfico de datos del mismo. Para comprender de mejor forma esta definición, se analizará el siguiente ejemplo: se desea permitir el tráfico que procede de la red 192.168.1.0/24. La ACL a configurar sería.

```
access-list 1 permit 192.168.10.0 0.0.0.255
```

En el ejemplo anterior se aprecia: el comando para definir la lista de control de acceso (access-list), el número de identificación de la ACL (1, para este ejemplo), acción (permitir o denegar), el direccionamiento IP de la red de donde procede el tráfico de datos (192.168.10.0) y la WildCard (0.0.0.255).

La WildCard se determina a partir de la máscara de subred, que en este caso es 255.255.255.0, expresada en formato decimal, en binario sería 11111111.11111111.11111111.00000000. Para calcular la WildCard, se convierten los unos en cero y los ceros en uno en la máscara de subred, la cual quedaría de la siguiente forma: 00000000.00000000.00000000.11111111, expresada en formato binario, en formato decimal sería 0.0.0.255.

Otro ejemplo de configuración de ACLs filtrando paquetes de acuerdo a la Dirección Origen.

```
access-list 2 deny 192.168.15.1
access-list 2 permit 192.168.15.0 0.0.0.255
access-list 2 deny 192.168.0.0 0.0.255.255
access-list 2 permit 192.0.0.0 0.255.255.255
```

En el ejemplo anterior se aprecia que las sentencias configuradas para las ACLs van de lo específico a lo general. Valorando la información contenida en la cabecera de los paquetes entrantes a la red, en relación al direccionamiento, de la siguiente forma:

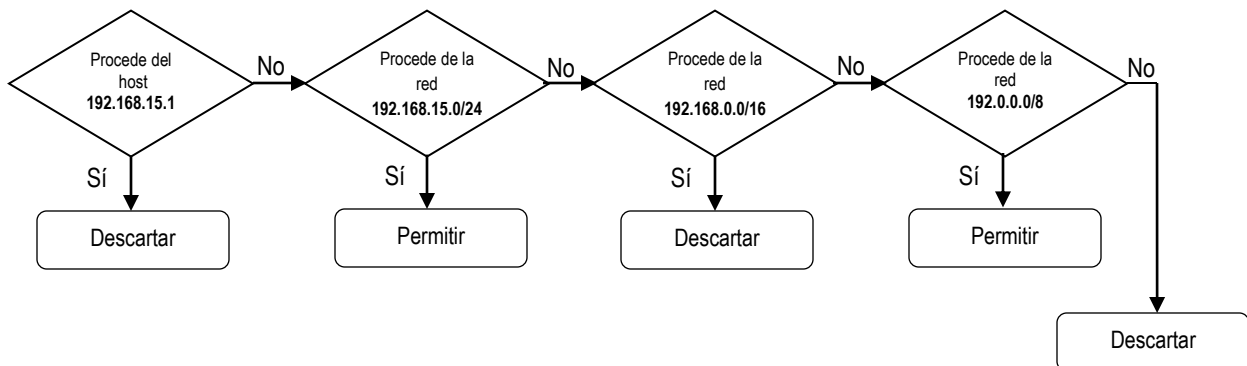


Diagrama de Evaluación de Listas de Control de Acceso

6.8.5.2.2. Filtrado de paquetes de acuerdo al Direccionamiento, Protocolo o Puerto

Es factible definir las ACLs en función de tres criterios: tipo de protocolo, direccionamiento IP de origen y direccionamiento IP de destino, puertos TCP o UDP de origen y puertos TCP o UDP de destino. Para comprender de mejor forma esta definición, se analizará el siguiente ejemplo: se desea permitir el tráfico que procede de la red 192.168.10.0/24, con el objeto de acceder al servicio HTTP. La ACL a configurar sería.

```
access-list 10 permit tcp 192.168.10.0 0.0.0.255 any eq 80
```

En el ejemplo anterior se aprecia: el comando para definir la lista de control de acceso (access-list), el número de identificación de la ACL (10, para este ejemplo), acción (permitir o denegar), el direccionamiento IP de la red de donde procede el tráfico de datos (192.168.10.0), la WildCar (0.0.0.255) y la especificación de que el protocolo o servicio al cual se accede es 80, es decir HTTP.

Para aplicar las ACLs con el fin de controlar el flujo de datos, se requiere efectuar dos acciones: Configurar la ACL, indicando un número o nombre de lista de

control de acceso y asociándole a éste las condiciones de accesibilidad. A continuación, aplicar en las interfaces las ACLs previamente configuradas.

Ejemplo: creación de una ACL para restringir el acceso a internet de tal forma que sólo se pueda efectuar navegación WEB. Mediante las instrucciones siguientes se define una ACL numerada como 200 permitiendo las solicitudes a los puertos 80 y 443 y además se crea una ACL enumerada como 201 gracias a la cual se admiten respuestas de los servicios HTTP y HTTPS.

```
access-list 200 permit tcp 192.168.15.0 0.0.0.255 any eq 80
```

```
access-list 200 permit tcp 192.168.15.0 0.0.0.255 any eq 443
```

```
access-list 201 permit tcp any 192.168.15.0 0.0.0.255 established
```

6.8.5.2.3. ACLs Enumeradas

Son usadas en redes relativamente pequeñas, con un tipo de tráfico similar. Es importante resaltar que el número le es asignado a la ACL de acuerdo al protocolo que se desea filtrar, de la siguiente forma:

Rango	Protocolo y Propósito de la ACL
De la 1 a la 99	ACL IP para Filtrado de paquetes de acuerdo a la Dirección Origen.
De la 100 a la 199	ACL IP para Filtrado de paquetes de acuerdo al Direccionamiento, Protocolo o Puerto.
De la 600 a la 699	ACL AppleTalk.
De la 800 a la 899	ACL IPX.
De la 1300 a la 1999	ACL IP para Filtrado de paquetes de acuerdo a la Dirección Origen.
De la 2000 a la 2699	ACL IP para Filtrado de paquetes de acuerdo al Direccionamiento, Protocolo o Puerto

6.8.5.2.4. ACLs con denominación

Son usadas en redes más extensas, en las que es importante identificar el propósito de cada ACL, de acuerdo a su nombre. La asignación del nombre de la ACL se hace en base a los siguientes criterios: pueden contener caracteres alfanuméricos, no deben contener espacios en blanco, ni caracteres espaciales, como signos de puntuación, es necesario que comiencen por una letra y es deseable que el nombre de la ACL se escriba en mayúscula.

6.8.5.3. Gestión de Comandos sobre Listas de Control de Acceso

6.8.5.3.1. Sintaxis completa para la creación de ACLs

access-list NúmeroListaAcceso deny|permit remark Dir.IP.Origen [wildcard origen] [log]

Parámetro	Explicación
NúmeroListaAcceso	Número que permite identificar la Lista de Control de Acceso, es un valor comprendido entre 1 y 99, de 1300a 1999 o de 2000 a 2699.
deny permit	Puede colocarse una de estos dos parámetros, el primero para denegar el acceso y el segundo para permitirlo, previa evaluación de las condiciones establecidas en la ACL.
Remark	Se utiliza para agregar un comentario a las entradas de las ACLs con el objeto de poder establecer un mayor análisis de la lista.
Dir.IP.Origen	Dirección IP del equipo host o de toda una red de origen (el formato de una Dirección IP es del tipo W. X.Y.Z, donde cada una de estas variables toma valores decimales entre 0 y 255).
wildcard origen	Son los bits de WildCard para aplicar a la dirección origen, para definirla se utilizan cuatro números decimales separados por puntos, donde cada número representado en binario equivale al complemento del respectivo número de la máscara de red. Este parámetro es opcional.
Log	Es utilizado para generar un mensaje informativo relacionado con el paquete que se evalúa. Este parámetro es opcional.

6.8.5.3.2. Mostrar ACLs existentes

Con el siguiente comando se puede apreciar el listado de ACLs existentes.

```
show access-list
```

6.8.5.3.3. Eliminación de una ACL

Mediante el comando siguiente se está eliminando la lista de control de acceso enumerada como 10.

```
no Access-list 10
```

6.8.6. Reglas de Control de Acceso

Estas definen si se permite o no el acceso a Squid. Se aplican a las *Listas de Control de Acceso*. Deben colocarse en la sección de reglas de control de acceso definidas por el administrador, es decir, a partir de donde se localiza el siguiente texto:

```
#  
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS  
#
```

La sintáxis básica es la siguiente:

```
http_access [deny o allow] [lista de control de acceso]
```

También se pueden definir reglas valiéndose de la expresión **!**, la cual significa *excepción*. Por ejemplo, se tienen dos listas de control de acceso, una denominada *lista1* y otra denominada *lista2*:

```
http_access allow lista1 !lista2
```

Aquí se permite el acceso a los pertenecientes a la *lista1* excepto a los de la *lista2*. Este tipo de reglas son útiles cuando se tiene un gran grupo de IP dentro de un rango de red al que se debe **permitir** acceso, y otro grupo dentro de la misma red al que se debe **denegar** el acceso.

6.8.6.1. Aplicando Listas y Reglas de control de acceso

Una vez comprendido el funcionamiento de la Listas y las Reglas de Control de Acceso, procederemos a ver varios casos de ejemplo.

Ejemplo 1:

Considerando, como ejemplo, que se dispone de una red 192.168.1.0/255.255.255.0, si se desea definir toda la red local, utilizaremos la siguiente línea en la sección de *Listas de Control de Acceso*:

```
acl totalared src 192.168.1.0/255.255.255.0
```

Habiendo hecho lo anterior, la sección de listas de control de acceso debe quedar más o menos del siguiente modo:


```
#  
# Recommended minimum configuration:  
acl all src 0.0.0.0/0.0.0.0  
acl manager proto cache_object  
acl localhost src 127.0.0.1/255.255.255.255  
acl todalared src 192.168.1.0/255.255.255.0
```

A continuación procedemos a aplicar la regla de control de acceso:

```
http_access allow todalared
```

De esta forma, la zona de reglas de control de acceso debería quedar más o menos de este modo:

```
#  
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR  
# CLIENTS  
#  
http_access allow localhost  
http_access allow todalared  
http_access deny all
```

La regla **http_access allow todalared** permite el acceso a *SQUID* a la *Lista de Control de Acceso* denominada *todalared*, la cual está conformada por 192.168.1.0/255.255.255.0. Esto significa que cualquier máquina desde 192.168.1.1 hasta 192.168.1.254 podrá acceder a *SQUID*.

Ejemplo 2:

Si sólo se desea permitir el acceso a *SQUID* a ciertas direcciones IP de la red local, deberemos crear un fichero que contenga dicha lista. Generemos el fichero */etc/squid/lista*, dentro del cual se incluirán solo aquellas direcciones IP que desea confirmen la Lista de Control de acceso. Ejemplo:

```
192.168.1.1
192.168.1.2
192.168.1.3
192.168.1.15
192.168.1.16
192.168.1.20
192.168.1.40
```

Denominaremos a esta lista de control de acceso como *redlocal*:

```
acl redlocal src "/etc/squid/lista"
```

Habiendo hecho lo anterior, la sección de listas de control de acceso debe quedar

más o menos del siguiente modo:

```
#
# Recommended minimum configuration:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl redlocal src "/etc/squid/lista"
```

A continuación procedemos a aplicar la regla de control de acceso:

```
http_access allow redlocal
```

Habiendo hecho lo anterior, la zona de reglas de control de acceso debería quedar

más o menos de este modo:

```
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR
# CLIENTS
#
http_access allow localhost
http_access allow redlocal
http_access deny all
```

La regla **http_access allow redlocal** permite el acceso a *SQUID* a la *Lista de Control de Acceso* denominada *redlocal*, la cual está conformada por las

direcciones IP especificadas en el archivo `/etc/squid/lista`. Esto significa que cualquier máquina no incluida en `/etc/squid/lista` no tendrá acceso a `SQUID`.

6.8.6.2. Restricciones de acceso a sitios web

Denegar el acceso a ciertos sitios Web permite hacer un uso más racional del ancho de banda con el que se dispone. El funcionamiento consiste en denegar el acceso a nombres de dominio o direcciones Web que contengan patrones en común. **Definiendo patrones comunes.** Primero se crea un archivo que contenga las direcciones Web y palabras usualmente utilizadas en nombres de ciertos dominios, por ejemplo `/etc/squid/sitiosdenegados`:

```
www.sitioporno.com
sitioindeseable.com
napster
sex
porn
mp3
xxx
adult
```

Se debe crear una *Lista de Control de Acceso* que a su vez defina al archivo `/etc/squid/sitiosdenegados`.

En este caso se denominará como *negados*. Entonces la línea correspondiente quedaría del siguiente modo:

```
acl negados url_regex "/etc/squid/sitios-denegados"
```

A continuación se especificará una regla de control de acceso para esta *ACL*:
`http_access deny negados` Note que ésta debe ir antes de cualquier otra regla que permita el acceso a cualquier otra lista. Ejemplo:

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR
CLIENTS
#
http_access deny negados
http_access allow localhost
http_access deny all
```

Si por ejemplo, al incluir una palabra en particular afecta el acceso a un sitio Web, puede crearse una lista de dominios o palabras que contengan un patrón, pero que consideraremos como apropiados. Por ejemplo en la lista anterior de sitios denegados está la palabra *sex*, con la cual se negaría acceso a cualquier nombre de dominio que incluya dicha cadena de caracteres, como *extremesex.com*. Sin embargo también estaría bloqueando a sitios como *sexualidadjoven.cl*, el cual no tiene que ver en lo absoluto con pornografía, sino orientación sexual para la juventud. Para solucionar este problema se puede añadir este nombre de dominio en un archivo que se puede denominar.

/etc/squid/sitios-inocentes.

Este archivo será definido en una Lista de Control de Acceso del mismo modo en que se hizo anteriormente con el archivo que contiene dominios y palabras denegadas.

```
acl inocentes url_regex "/etc/squid/sitios-inocentes"
```

Para hacer uso del archivo, solo basta utilizar la expresión **!** en la misma línea utilizada para la *Regla de Control de Acceso* establecida para denegar el mismo.

http_access deny negados !inocentes

La regla anterior especifica que se denegará el acceso a todo lo que encuentre en la *Lista de Control de Acceso* denominada *negados* **excepto** lo que haga parte de la *Lista de Control de Acceso* denominada *inocentes*. Es decir, se podrá acceder sin dificultad a www.sexualidadjoven.cl manteniendo la restricción para la cadena de caracteres *sex*.

Capítulo No 7

EL SERVIDOR WEB APACHE

Servidor WEB



Capítulo No 7. EL SERVIDOR WEB APACHE

El tráfico web es la cantidad de datos enviados y recibidos por los visitantes de un sitio web. Se considera que este tráfico ocupa la mayor porción de flujo de datos de Internet y es determinado por el número de visitantes y de páginas que se visitan. Este elemento de Internet, puede ser medido y esta medición arroja la popularidad de los sitios web aunque estos no estén asociados a un portal específico. Los posibles criterios de medición que son utilizados son: el número de visitas, el promedio de páginas visitadas por un usuario, promedio de duración de vista de una página en particular, las clases de direcciones IP requeridas para abrir una web y su contenido, las páginas más populares y las portadas más requeridas.

Apache es el servidor web más usado en sistemas Linux. Los servidores web son implementados para servir a contenedores web como Mozilla, Netscape, Google Chrome o Internet Explorer, entre otros, las páginas que son solicitadas por equipos clientes. Para que el funcionamiento se dé correctamente, el usuario que desea utilizar el servicio debe introducir un URL (Localizador de Recursos Uniforme - Uniform Resource Locator) en su respectiva barra de direcciones para direccionar a un servidor web por medio de la técnica llamada FQDN (Fully Qualified Domain Name - Nombre de Dominio Totalmente Cualificado). Por ejemplo, para ver la página web del sitio web de Ubuntu solo se debería introducir únicamente el FQDN.

Cuando hablamos de protocolos utilizados en servidores WEB, el más implementado para la visualización de páginas Web es el HTTP (Hyper Text Transfer Protocol – Protocolo de transferencia de Hipertexto), el cual puede ser

implementado con niveles de seguridad sobre HTTPS (Secure Sockets Layer) o con el fin de subir y descargar archivos utilizando el FTP (File Transfer Protocol).

Los servidores web Apache son utilizados en su gran mayoría en conjunto con el motor de bases de datos MySQL, y lenguajes de programación como PHP, Python y Perl. A esta configuración en el mundo del software libre se le denomina LAMP (Linux, Apache, MySQL y Perl/Python/PHP) y conforma una potente y robusta plataforma para el desarrollo y distribución de aplicaciones basadas en la web. En esta obra solo se abordara el tema del servidor WEB como herramienta para la publicación de páginas a clientes en general.

Entre los más grandes sitios de la red Internet que utilizan los servicios de Apache, podemos encontrar:

- Amazon.com
- Yahoo!
- W3 Consortium
- Financial Times
- Network solutions
- MP3.com

7.1 VENTAJAS DE APACHE

- Su licencia. Esta es de código abierto del tipo BSD que permite el uso comercial y no comercial de Apache.
- El soporte de una comunidad robusta de desarrolladores de código abierto.
- Arquitectura Modular. Con esta característica los usuarios de Apache pueden adicionar fácilmente funcionalidad a sus ambientes específicos.
- Portabilidad. Apache trabaja sobre todas las versiones recientes de UNIX y Linux, Windows, BeOs y mainframes.

7.2 SERVICIOS QUE OFRECE APACHE

- Filtrado WEB, el cual se puede hacerse por contenidos, tipos de contenidos MIME, por extensión de archivos a descargar, por dominios u otra preferencia del administrador del sistema.
- Cortafuegos. funciona en conjunción con el servicio de filtrado web y el servicio de antivirus
- Manejador de bases de datos para autenticación de un gran número de usuarios.
- Respuestas personalizadas a los errores y problemas.
- Hosts virtuales; permite al servidor distinguir entre diferentes peticiones hechas por distintas direcciones IP o nombres (mapeadas en la misma máquina).
- Registros (logs) confiables configurables; se puede configurar el Apache para generar estos logs personalizados.

7.3 DIRECTORIOS BÁSICOS

Apache cuenta con dos directorios básicos que son:

- El Directorio Raíz donde se encuentran los archivos de configuración, y los logs.
- El Directorio Web donde se guardan las paginas, sites, cgi que van a ser mostradas.

Para realizar las configuraciones de red, el servidor HTTP Apache debe iniciarse con permisos de root o administrador. Específicamente, necesita enlazarse al puerto 80 para escuchar peticiones y aceptar conexiones. Una vez hecho esto, Apache abandona todos sus derechos y se ejecuta como un usuario distinto de

root, como se especifica en sus archivos de configuración. Por defecto es con el usuario Apache.

A continuación se muestran los pasos necesarios para instalar y configurar el servidor HTTP APACHE.

7.4 DESCARGA

Si bien muchas de las distribuciones de Linux vienen con el servidor HTTP Apache se puede descargar siempre la última versión desde <http://www.apache.org/dist>.

En nuestro caso vamos a descargar el paquete como se ha realizado en los capítulos anteriores, utilizando el comando “#apt-get install apache2”.

```
root@us804:/# apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  apache2-mpm-worker apache2-utils apache2.2-common libapr1 libaprutil1
  libpcre3 libpq5
Paquetes sugeridos:
  apache2-doc
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-mpm-worker apache2-utils apache2.2-common libapr1
  libaprutil1 libpcre3 libpq5
0 actualizados, 8 se instalarán, 0 para eliminar y 18 no actualizados.
Necesito descargar 1866kB de archivos.
After this operation, 6205kB of additional disk space will be used.
¿Desea continuar [S/n]? S_
```

Figura No 84. Descarga del Apache

Después de descargar los datos necesarios para el óptimo funcionamiento de nuestro servidor se nos muestra nuevamente el símbolo del sistema,

```
Module env installed; run /etc/init.d/apache2 force-reload to enable.
Module mime installed; run /etc/init.d/apache2 force-reload to enable.
Module negotiation installed; run /etc/init.d/apache2 force-reload to enable.
Module setenvif installed; run /etc/init.d/apache2 force-reload to enable.
Module status installed; run /etc/init.d/apache2 force-reload to enable.
Module auth_basic installed; run /etc/init.d/apache2 force-reload to enable.
Module authz_default installed; run /etc/init.d/apache2 force-reload to enable.
Module authz_user installed; run /etc/init.d/apache2 force-reload to enable.
Module authz_groupfile installed; run /etc/init.d/apache2 force-reload to enable
.
Module authn_file installed; run /etc/init.d/apache2 force-reload to enable.
Module authz_host installed; run /etc/init.d/apache2 force-reload to enable.

Configurando libpcre3 (7.4-1ubuntu2.1) ...

Configurando apache2-mpm-worker (2.2.8-1ubuntu0.14) ...
 * Starting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name,
using 192.168.2.200 for ServerName
[ OK ]

Configurando apache2 (2.2.8-1ubuntu0.14) ...
Processing triggers for libc6 ...
ldconfig deferred processing now taking place
root@us804:/# _
```

Figura No 85. Culminación de la descarga

Ahora nos disponemos a configurar el archivo ubicado en “#/etc/apache2/” donde se encuentra todo lo referente a Apache, nuestro servidor Web, pero primero hacemos la respectiva copia de seguridad de dicho archivo:

```
root@us804:/# cp /etc/apache2/apache2.conf /etc/apache2/apache2.conf_copia
root@us804:/# ls /etc/apache2/
apache2.conf      conf.d      httpd.conf      mods-enabled    sites-available
apache2.conf_copia  envvars    mods-available  ports.conf      sites-enabled
root@us804:/# _
```

Figura No 86. Copia de seguridad del archivo apache2.conf

Seguidamente, digitamos el comando “#nano /etc/apache2/apache2.conf” lo que nos muestra el archivo de configuración de Apache como sigue:

```
GNU nano 2.0.7 File: /etc/apache2/apache2.conf
#
# Based upon the NCSA server configuration files originally by Rob McCool.
#
# This is the main Apache server configuration file. It contains the
# configuration directives that give the server its instructions.
# See http://httpd.apache.org/docs/2.2/ for detailed information about
# the directives.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# The configuration directives are grouped into three basic sections:
# 1. Directives that control the operation of the Apache server process as a
#    whole (the 'global environment').
# 2. Directives that define the parameters of the 'main' or 'default' server,
#    which responds to requests that aren't handled by a virtual host.
#    These directives also provide default values for the settings
#    of all virtual hosts.
# 3. Settings for virtual hosts, which allow Web requests to be sent to
#
# [ Read 298 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figura No 87. Configuración del archivo apache2.conf

7.5 CONFIGURACIÓN

Apache soporta un gran conjunto de opciones de configuración que son fáciles y sencillas de seguir. La configuración por defecto de Apache es bastante eficiente y con ella se puede crear documentos HTML de manera inmediata y publicarlos.

Apache se configura con una serie de directivas y estas directivas suelen estar distribuidas entre los archivos 'httpd.conf', 'srm.conf' y 'access.conf' en el directorio **etc/httpd/conf**.

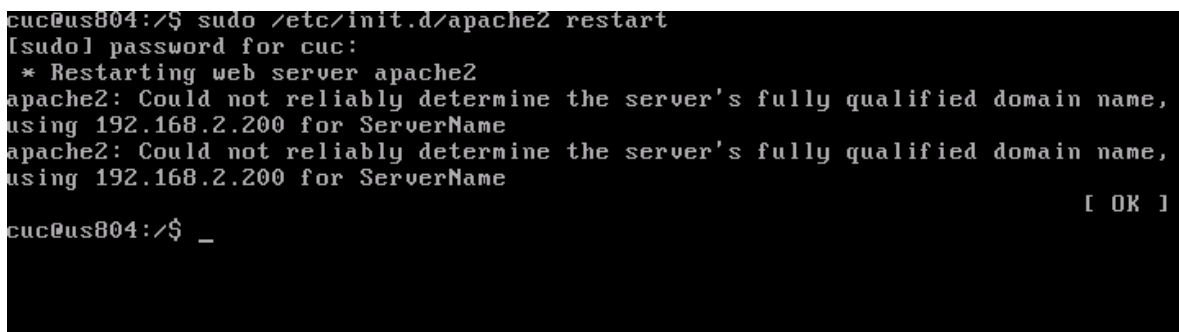
Estas directivas están agrupadas según su función en distintos módulos cargables. Por defecto se cargan casi todos quedando como opcionales los restantes.

- **ServerType:** Especifica el tipo de servidor que puede ser standalone o inetd.
Ej: *ServerType standalone*
- **ServerRoot:** Es la ruta donde está instalado Apache y dentro se encuentran los distintos subdirectorios como el conf, log y otros.
Ej: *ServerRoot "/etc/httpd"*
- **PidFile:** Identifica el lugar donde se encuentra el archivo del número identificador del proceso.
Ej: *PidFile /var/run/httpd.pid*
- **ScoreBoardFile:** Archivo que almacena información interna del proceso del servidor. Hay que tener en cuenta que cada llamada de Apache debe tener su propio archivo.
Ej: *ScoreBoardFile /var/run/httpd.scoreboard*
- **Timeout:** Número de segundos que espera antes de enviar un mensaje de tiempo de espera agotado.
Ej: *Timeout 300*
- **KeepAlive:** Permite o no que se haga más de una petición por conexión. Los valores son On y Off.
Ej: *KeepAlive On*
- **MaxKeepAliveRequests:** Indica el número máximo de peticiones por conexión. Si el valor es 0 indica que son ilimitadas.
Ej: *MaxKeepAliveRequests 100*

- **KeepAliveTimeout:** Tiempo de espera hasta la siguiente petición desde el mismo cliente y conexión.
Ej: *KeepAliveTimeout 15*
- **MaxClients:** Máximo número de clientes que pueden conectarse simultáneamente al servidor. 0 es ilimitado.
Ej: *MaxClients 150*
- **Listen:** Permite que Apache escuche por una dirección IP determinada o bien por un determinado puerto además de los definidos por defecto.
Ej: *Listen 11.1.0.11:80*
Listen 80
Listen 8080
- **AddLanguage:** Permite especificar qué lenguajes pueden estar sujetos a negociación para que en función de la configuración de idioma del cliente la información se le pueda presentar en su idioma. En el caso que nos ocupa es el español entre otros:
Ej: *AddLanguage es .es*
- **ServerName.** Nombre del servidor.
Ej: *ServerName diana.yadinet.org*
- **DocumentRoot:** Ruta donde se ubican las páginas de la web
Ej: *DocumentRoot "/var/www/html"*
- **ServerAdmin:** Indica la dirección de e-mail del administrador
Ej: *ServerAdmin webmaster@yadinet.org*
- **VirtualHost:** Define los parámetros de un host virtual

```
EJ: <VirtualHost direccion.ip.host.o.algun_dominio.com>
    ServerAdmin webmaster@host.some_domain.com
    DocumentRoot /www/docs/host.some_domain.com
    ServerName host.some_domain.com
    ErrorLog logs/host.some_domain.com-error_log
    CustomLog      logs/host.some_domain.com-access_log
    common
</VirtualHost>
```

Para saber que apache está corriendo correctamente debemos asegurarnos que el demonio asociado al servicio está activo en la máquina, para esto digitamos el comando: `#sudo /etc/init.d/apache2 restart`, lo cual detendrá el servicio y lo inicializara nuevamente o lo que se conoce como “reset”. Para saber que el servicio está activo debe aparecernos la palabra [OK] lo cual indica que existe funcionalidad al 100%. Si fuera necesario debemos digitar la password como administrador y esperar a que se levante el servicio.



```
cuc@us804:/$ sudo /etc/init.d/apache2 restart
[sudo] password for cuc:
* Restarting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name,
using 192.168.2.200 for ServerName
apache2: Could not reliably determine the server's fully qualified domain name,
using 192.168.2.200 for ServerName
[ OK ]
cuc@us804:/$ _
```

Figura No 88. Reinicio del servicio Apache

Para realizar la respectiva prueba si nuestro servidor Web está funcionando, debemos desplegar la pagina de prueba que se encuentra en el directorio `/var/www`, para esto nos dirigimos a este directorio y revisamos en esta ubicación la existencia de la página de prueba:


```
cuc@us804:/$ cd /var/www/  
cuc@us804:/var/www$ ls  
index.html  
cuc@us804:/var/www$ _
```

Figura No 89. Ubicación de la carpeta WWW

En esta carpeta se deben colocar todas las páginas que se deseen publicar en nuestro servidor web, en nuestro caso, revisaremos el contenido del archivo index.html y lo modificaremos para este ejemplo. Digitamos entonces el comando #nano index.html y presionamos <Enter> y se nos muestra el contenido del archivo y lo modificamos a disposición:

```
GNU nano 2.0.7          File: index.html          Modified  
  
<html><body><h1>Mi servidor Apache esta funcionando!</h1></body></html>_  
  
[ Error writing index.html: Permission denied ]  
^G Get Help    ^O WriteOut    ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos  
^X Exit        ^J Justify    ^W Where Is  ^U Next Page ^U UnCut Text ^T To Spell
```

Figura No 90. Contenido de archivo index.html

Cabe recordar que después de realizar cualquier cambio utilizando el programa “nano”, se debe guardar las modificaciones presionando las teclas “Ctrl” y “O” simultáneamente y colocándole el nombre que deseamos y finalizamos con la tecla <Enter> para guardar y “Ctrl” y “X” para salir del modo edición del programa “nano”.

La prueba final para saber si todo esta correcto es abrir una ventana de cualquier navegador y digitar la dirección IP del equipo donde se encuentra el servidor WEB instalado o en su defecto el nombre de dominio asociado a esta dirección. En nuestro caso como el servidor DNS aún no se ha instalado se especificara la dirección IP.

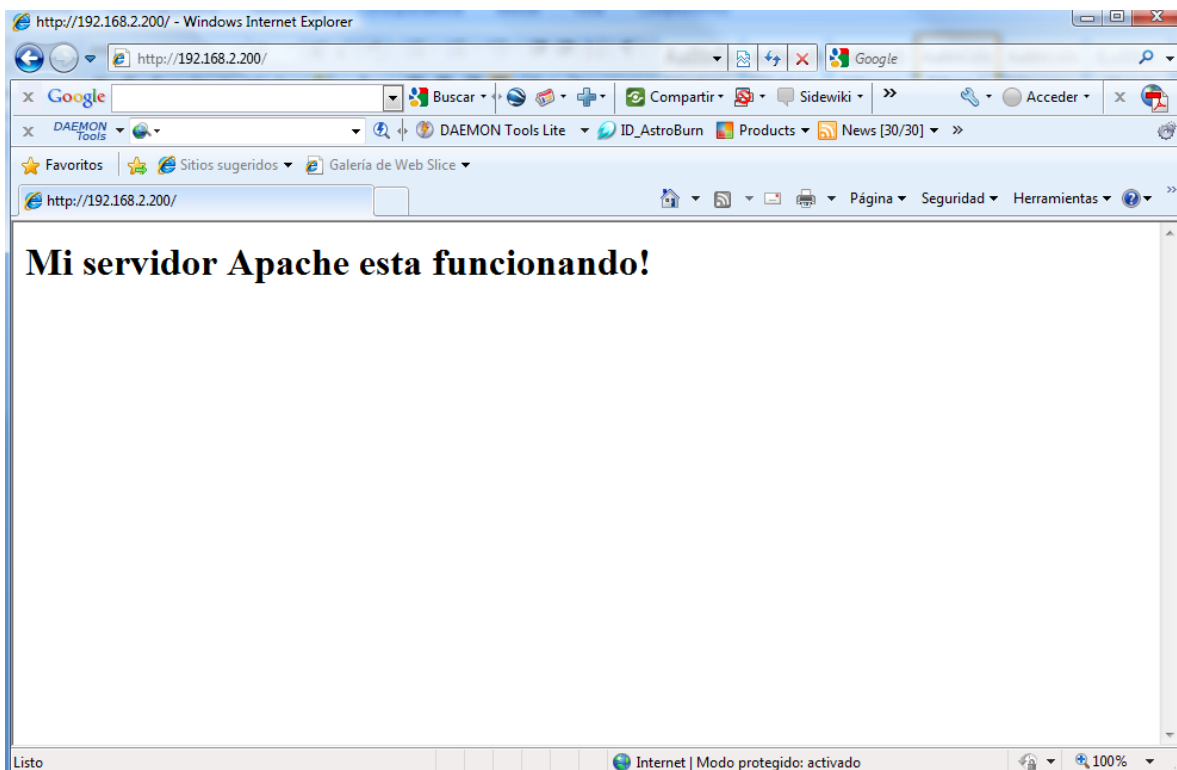


Figura No 91. Contenedor WEB con página de inicio de servidor APACHE

De esta misma forma, en la carpeta WWW se puede colocar cualquier página o carpeta que contenga la información a mostrar en nuestro servidor Web y este mismo se encargara de desplegarla según los parámetros que fueron entregados con anterioridad.

GLOSARIO



Alan Cox: Junto con Linus Torvalds, unos de los desarrolladores más activos del kernel.

Alpha: La computadora de arquitectura RISC (*Reduced Instruction Set Computer*) (Computadora con juego de instrucciones reducido) desarrollada por Digital Equipment Corporation.

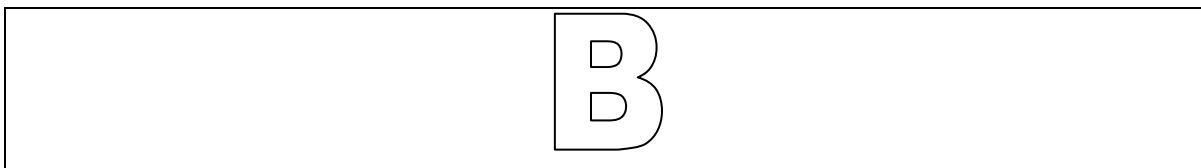
Aplicue: Normalmente se trata de un pequeño programa diseñado para ser ejecutado empotrado en otro programa anfitrión.

Applet: Inglés. «aplique».

Arranque dual: El acto de configurar un ordenador para que pueda arrancar más de un sistema operativo. El nombre es algo confuso, pues es posible arrancar más de dos sistemas operativos, a diferencia de lo que implica la palabra «dual». En inglés: «*Dual Boot*».

Arranque: Proceso por el cual un ordenador comienza a ejecutar un sistema operativo cuando se le aplica la energía de alimentación. En inglés: «*bootstrap*» o más comúnmente «*boot*».

ATAPI: Sigla de *AT Attachment Packet Interface* (interfaz de paquetes para conectar a AT). ATAPI es el protocolo mediante el cual las unidades de CD-ROM se comunican con la computadora sobre la interfaz IDE.



Biblioteca: Cuando se habla de ordenadores, se refiere al conjunto de rutinas que realizan las operaciones usualmente requeridas por los programas. Las bibliotecas pueden ser compartidas, lo que quiere decir que las rutinas de la biblioteca residen en un fichero distinto de los programas que las utilizan. Las rutinas de biblioteca pueden «enlazarse estáticamente» al programa, en cuyo caso se agregan físicamente las copias de las rutinas que el programa necesita. Estos binarios enlazados estáticamente no requieren de la existencia de ningún fichero de biblioteca para poder funcionar. Los programas enlazados con bibliotecas compartidas no funcionarán a menos que se instalen las bibliotecas necesarias. En inglés: «library».

Binario: Aunque se denomina binario al sistema de numeración en base dos que usan las computadoras, con frecuencia la palabra se refiere a la forma ejecutable de un programa. Lo contrario a «código fuente». En inglés: «*binary*».

Binary: Inglés, vea «binario».

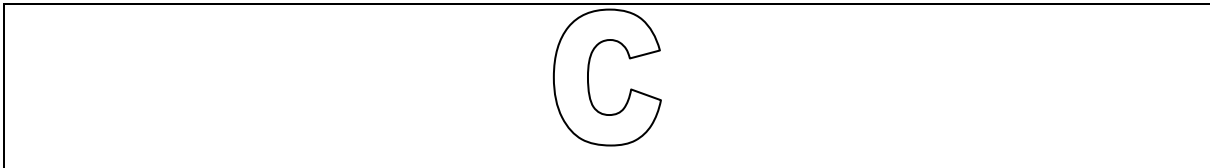
BIOS: Sigla de *Basic Input/Output System* (sistema de entrada/salida básico). En los sistemas compatibles con PC, el BIOS se utiliza para realizar todas las funciones necesarias para colocar en estado inicial el hardware del sistema cuando se lo conecta a la alimentación de energía. El BIOS controla el proceso de arranque, proporciona rutinas de entrada/salida de bajo nivel (de aquí su nombre)

y (usualmente) permite que el usuario modifique los detalles de la configuración del hardware del sistema.

Boot Diskette: Inglés, «disquete de arranque».

Boot: Abreviatura de «bootstrap». «arranque».

Bootstrap: Inglés, «arranque».



Cabeza: Cuando se refiere a unidades de disco, la cantidad de cabezas de una unidad de disco. En cada platina de una unidad de disco, hay dos cabezas -- una en cada superficie -- aunque una de las superficies no se utilice. En inglés: «*head*». Vea también «geometría».

Carga del sistema: Es una medida que nos indica la carga que están produciendo los procesos que están usando la CPU en un momento determinado. Si ejecutamos un solo proceso que consuma el 50% de CPU tendríamos una carga de 0.50. Si ejecutamos un solo proceso que requiere toda la potencia de CPU disponible (100%) tendríamos una carga de 1.00. Si ejecutamos dos procesos con las características del anterior, cada uno consumiría el 50% de la potencia de la CPU y la carga subiría a 2.00. y así sucesivamente. La carga del sistema se puede consultar con los comandos `top` y `w`.

Cilindro: Cuando se refiere a unidades de disco, corresponde a la cantidad de distintas posiciones que pueden ocupar las cabezas de lectura/escritura sobre la platina del disco. Cuando se mira desde arriba de las platinas, cada posición de

una cabeza describe un círculo imaginario con diferentes diámetros sobre la superficie de la platina, pero cuando se mira de costado, estos círculos pueden pensarse como una serie de cilindros anidados uno dentro de otro, y de allí el término. En inglés: «*cylinder*». Vea también «Geometría».

CISC: Sigla de *Complex Instruction Set Computer* (computadora con juego de instrucciones complejo). Se trata de una filosofía de diseño de ordenadores en la cual el procesador se diseña para ejecutar una cantidad relativamente grande de instrucciones diferentes, cada una de las cuales tarda distinto tiempo en ejecutarse (de acuerdo a la complejidad de la instrucción). Lo contrario de «RISC».

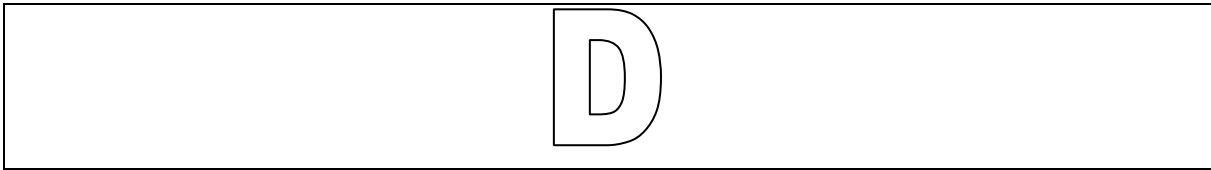
CMOS: En un principio la sigla de *Complementary Metal Oxide Semiconductor* (semiconductor de óxido metálico complementario) -- la tecnología de semiconductores que se utiliza en muchos circuitos integrados. Ahora con frecuencia designa al hardware de bajo nivel que contiene la configuración del BIOS y el reloj por hardware de la computadora.

Código fuente: El formato entendible por las personas de las instrucciones que conforman un programa. También se lo conoce como «fuentes». Sin los fuentes de un programa es muy difícil modificarlo. En inglés: «*source code*».

Consola virtual: Las consolas virtuales proporcionan múltiples «pantallas» en las cuales el usuario puede ingresar y ejecutar programas. El monitor del ordenador muestra una pantalla a la vez; hay una secuencia de teclas para alternar entre las distintas consolas virtuales. En inglés: «*virtual console*».

Controlador de dispositivo: Software que controla un dispositivo que está conectado a, o es parte de, una computadora. (En inglés: *device driver*).

Cylinder: Inglés, «cilindro».



Daemon: Inglés, «Demonio».

Demonio: Un demonio es un programa que funciona sin intervención humana, para cumplir una tarea determinada. Por ejemplo, *lpd* es un demonio que controla el flujo de los trabajos de impresión en una impresora.

Dependencias: Cuando se refiere a paquetes, las dependencias son requerimientos que existen entre paquetes. Por ejemplo, el paquete *foo* puede requerir ficheros que son instalados por el paquete *bar*. En este ejemplo, *bar* debe estar instalado, pues sino *foo* tendrá dependencias sin resolver. Normalmente, RPM no permitirá que se instalen paquetes con dependencias sin resolver.

Desmontaje: El acto de revocar el acceso a un sistema de ficheros. (Debe usted advertir que el programa que desmonta los sistemas de ficheros se denomina *umount*.) En inglés: «*umount*».

Device Driver: Inglés, «controlador de dispositivo».

Dirección IP: Las direcciones IP son el método mediante el cual se identifican los ordenadores individuales (o, en una interpretación más estricta, las interfaces de red de dichos ordenadores) dentro de un red TCP/IP. Todas las direcciones IP consisten en cuatro números separados por puntos, donde cada número está entre 0 y 255.

Disco duro: Un disco rígido contiene un medio magnético rotante (en forma de discos) que gira rápidamente. Hay pequeñas cabezas que flotan sobre la superficie de cada disco, y sirven para leer y escribir en el disco a medida que rota. En inglés: «*Hard Disk*».

Disk Drive: Inglés, «disco rígido».

Disk Druid: Disk Druid (Druida de disco) es un componente del programa de instalación de RHL que se utiliza para realizar las particiones de las unidades de disco durante el proceso de instalación.

Diskette: Inglés, «disquete».

Disquete de arranque: El disquete que se utiliza para arrancar distintas instalaciones RHL. En inglés: «*boot diskette*».

Disquete de rescate: Disquete que contiene un entorno de sistema rudimentario. Como el nombre sugiere, el disquete de rescate se utiliza normalmente en un intento de «rescatar» un sistema dañado para evitar la reinstalación total del sistema operativo. En inglés: «*Rescue Diskette*».

Disquete para soporte PCMCIA: El disquete necesario para las instalaciones de RHL que requieren el uso de un dispositivo PCMCIA durante la instalación. En inglés: «*PCMCIA Support Diskette*».

Disquete suplementario: Un disquete que se requiere en algunas clases de instalaciones de RHL. En inglés: «*Supplemental Diskette*».

Disquete: Dispositivo de almacenamiento masivo de pequeña capacidad, que viene en un cartucho intercambiable, con el propósito de leer y/o escribir en el mismo, mediante su uso en una unidad compatible. En inglés «*diskette*».

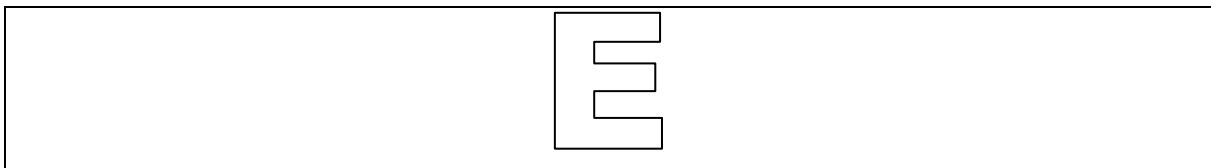
Distribución: Un sistema operativo (en general Linux), que se ha empaquetado para facilitar su instalación. En inglés: «*distribution*».

Distribution: Inglés, «distribución».

Domain name: Inglés, «Nombre de dominio».

Driver: Inglés, «Controlador de dispositivo».

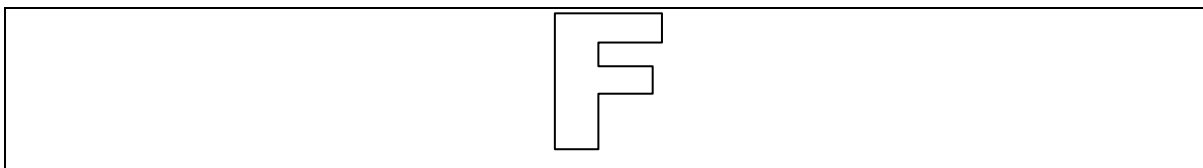
Dual Boot: Inglés, «arranque dual».



EIDE: Sigla de *Enhanced Integrated Drive Electronics* (electrónica de unidad integrada mejorada), y es la nueva versión de la interfaz estándar integrada y otra nomenclatura para una implementación particular de interfaces IDE. Con EIDE se pueden tener discos más rápidos y de mayor capacidad; la mayoría de los sistemas que se venden hoy día utilizan EIDE.

Errata: Errata es el original en latín de «¡auch!». Cuando se detectan errores en el software, se realiza la reparación de los errores y los cambios con frecuencia se entregan como errata. RHL no es una excepción a esta regla; disponemos de una página web para las Errata en <http://www.redhat.com/errata>.

Extended Partition: Inglés, «partición extendida».



FAQ: Sigla de *Frequently Asked Questions* (Preguntas Frecuentes). La información acerca de Linux se presenta generalmente en forma de listas de preguntas y sus respuestas, denominadas FAQs.

Fdisk: fdisk es un programa de utilidad que se usa para crear, borrar o modificar las particiones en una unidad de disco. Hay que tener mucho cuidado al usar este programa, ya que, un uso inapropiado del mismo puede hacernos perder nuestra información en el disco duro.

Filesystem: Inglés, «Sistema de ficheros».

Floppy: Término con connotaciones históricas para referirse a un disquete. Vea «disquete».

Formatear: Dar formato. El acto de escribir un sistema de ficheros en una unidad de disco.

Formatting: Inglés, «formatear».

FQDN: Sigla de *Fully Qualified Domain Name* (Dominio completamente expresado). Un FQDN es un nombre entendible por personas que incluye el nombre de la computadora y el nombre de dominio asociado a la misma. Por ejemplo, dada la computadora llamada «foo» y el nombre de dominio «bar.com», el FQDN será «foo.bar.com».

FTP: Sigla de *File Transfer Protocol* (Protocolo de transferencia de ficheros). También es el nombre del programa que, tal como su nombre indica, permite copiar ficheros desde un sistema a otro a través de la red.



Gateway: Inglés, «pasarela».

Geometría: Cuando se refiere a unidades de disco, las características físicas de su organización interna. Debe usted advertir que la unidad de disco puede informar una «geometría lógica» que es diferente de su «geometría física», normalmente para evitar las limitaciones impuestas por el BIOS. En inglés «*Geometry*». Vea además «cilindro», «cabeza» y «sector».

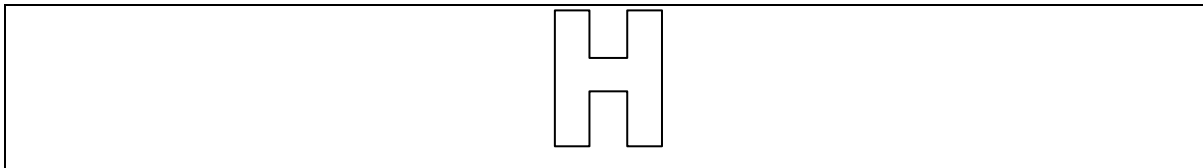
Geometry: Inglés, «geometría».

GID: Abreviatura de *Group ID* (Identificador de grupo). Por medio del GID se identifica la pertenencia de un usuario a un grupo. Los GIDs son números, aunque se almacenan nombres entendibles para personas en el fichero */etc/group*.

Group: Inglés, «grupo».

Grupo: El grupo es la manera de asignar derechos de acceso específicos a ciertas clases de usuarios. Por ejemplo, todos los usuarios que trabajan en el Proyecto X pueden agregarse al grupo *proyx*. Los recursos del sistema (como por ejemplo espacio en disco) que se dedican al Proyecto X se pueden configurar

entonces para permitir su acceso total sólo a los miembros de *proyx*. En inglés: «*group*».



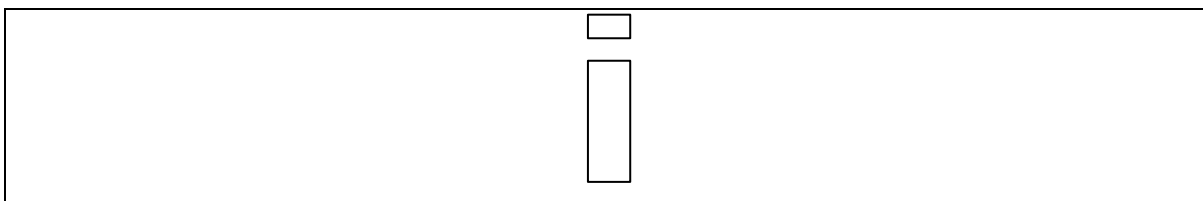
Hard Disk: Inglés, «disco rígido».

Head: Inglés, «cabeza».

Hostname: Inglés, «nombre de máquina».

I18n: Una «i» seguida de 18 letras, seguida de una «n», que corresponde a una abreviatura de la palabra inglesa «internationalization» (internacionalización). Vea «internacionalización».

IDE: Sigla de *Integrated Drive Electronics* (electrónica de unidad integrada), que denota la interfaz estándar usada para conectar fundamentalmente unidades de disco y CD-ROM a un ordenador. Vea también «EIDE» y «ATAPI».



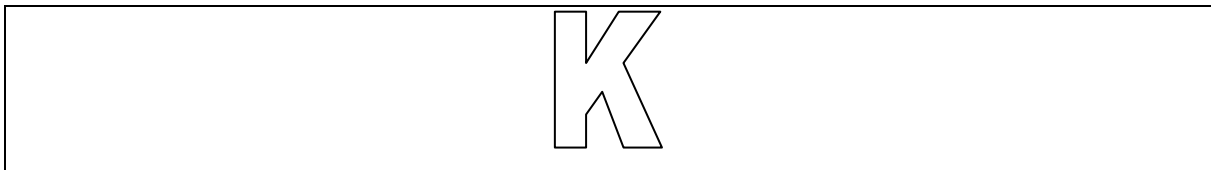
Intel: Compañía responsable de la producción de los microprocesadores más usuales en las computadoras personales compatibles con PC. Estos procesadores incluyen el 80386, 80486, Pentium, Pentium Pro, Pentium II y Pentium III.

Intercambio: También se lo conoce como «espacio de intercambio» («*swap space*»). Cuando un programa necesita más memoria de la que hay disponible físicamente en el ordenador, la información que no se está utilizando en ese momento se puede escribir en un búfer temporal en el disco, denominado «swap», y de esa manera se libera memoria. Algunos sistemas operativos admiten el intercambio contra un fichero específico, pero Linux normalmente realiza los intercambios contra una partición dedicada al intercambio. El término «swap» está mal elegido, pues en Linux se lo usa para denotar el intercambio de páginas según la demanda («*demand paging*»). En inglés: «Swap».

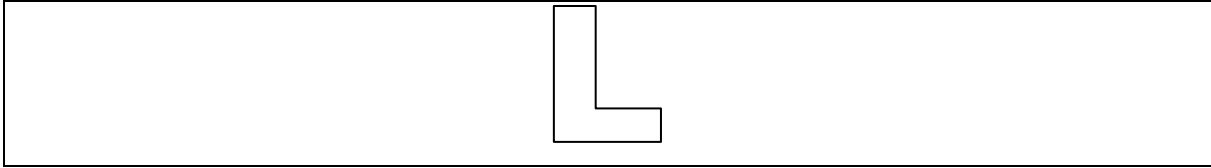
Internacionalización: La práctica de diseñar y escribir programas que pueden configurarse fácilmente para interactuar con el usuario en más de un idioma. Con frecuencia se lo denomina «i18n», a causa de la cantidad de letras entre la «i» de arranque y la «n» final. En inglés: «*Internationalization*».

Internationalization: Inglés, «internacionalización».

ISP: Inglés, siglas de *Internet Service Provider*. «psi».



Kernel: Inglés, «núcleo».



Library: Inglés, «biblioteca».

LILO: Cargador de arranque muy utilizado en sistemas Linux que se basan en procesadores compatibles con los de Intel.

Linus Torvalds: Creó Linux en 1991 mientras era estudiante universitario.

Linux: Sistema operativo completo, robusto, disponible libremente, que fue desarrollado originalmente por Linus Torvalds.

Linuxconf: Versátil programa de configuración del sistema escrito por Jacques Gelinas. Linuxconf proporciona un enfoque basado en menús para la configuración del sistema a través de distintas interfaces de usuario.

Llamada al sistema: Es una rutina que cumple una función a nivel del sistema en nombre de un proceso. En inglés: «System Call».

Load Average: Inglés, «carga del sistema».

Logical Partition: Inglés, «partición lógica».



Máscara de red: Una máscara de red es un conjunto de cuatro números separados por puntos. Cada número se representa normalmente como el equivalente decimal de un número binario de 8 bits, lo que significa que cada número puede tomar valores entre 0 (todos los bits en cero) y 255 (todos los bits en uno). Cada dirección IP consiste de dos partes (la dirección de red y el número de máquina). La máscara de red se usa para determinar el tamaño de cada una de estas partes. Las posiciones de los bits en uno de la máscara se consideran parte del espacio reservado para la dirección de red, mientras que los bits que están puestos a cero se consideran parte del espacio apartado para el número de máquina. En inglés: «netmask».

Master Boot Record: Inglés, «Registro de arranque maestro».

Memoria: Cuando se refiere a ordenadores, la memoria (en general) es cualquier hardware capaz de almacenar datos para recuperarlos posteriormente. En este contexto, la memoria en general se refiere específicamente a la RAM.

MILO: Cargador que se usa generalmente para sistemas Linux basados en el procesador Alpha.

Module: Inglés, «módulo».

Módulo: En Linux, un módulo es un conjunto de rutinas que realizan funciones a nivel de sistema, y que pueden cargarse y descargarse dinámicamente desde el núcleo cuando sea requerido. Los módulos con frecuencia contienen

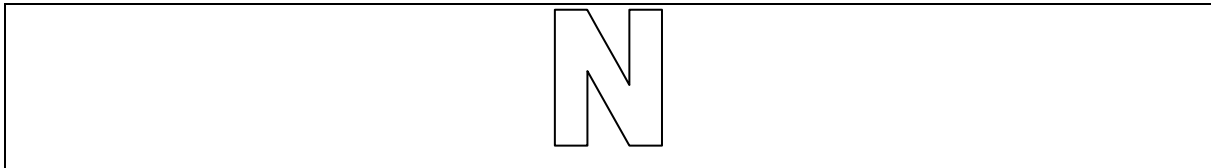
controladores de dispositivos, y están fuertemente ligados a la versión del núcleo; la mayoría de los módulos construidos con una versión dada de núcleo, no se cargarán de manera apropiada en un sistema que corra un núcleo con versión distinta. En inglés: «module».

Montaje: El acto por medio del cual los sistemas de fichero se hacen accesibles a los usuarios del sistema. En inglés: «mount».

Mount Point: Inglés, «punto de montaje».

Mount: Inglés, «montaje».

Mouse: Inglés: ratón. «ratón serie» y «ratón PS/2».



Nameserver: Inglés, «Servidor de nombres».

Netmask: Inglés, «máscara de red».

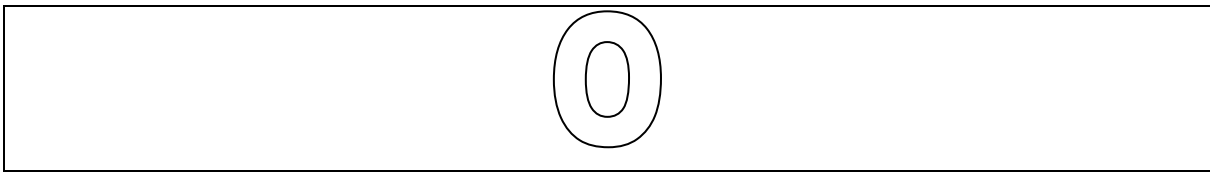
NFS: Sigla de *Network File System*, NFS es un método para lograr que un sistema de ficheros de una máquina remota sea accesible para el sistema local. Desde la perspectiva del usuario, un sistema de ficheros montado por NFS es indistinguible de un sistema de ficheros que reside en una unidad de disco directamente adosada a la máquina.

Nombre de dominio: El nombre de dominio se utiliza para expresar que las computadoras pertenecen a una determinada organización. Los nombres de

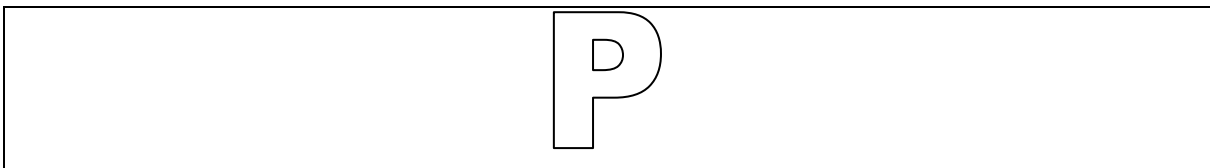
dominio son jerárquicos por naturaleza, y cada nivel de la jerarquía se separa de los otros niveles mediante un punto. Por ejemplo, el departamento de Finanzas de la Corporación Foo puede usar un dominio «finanzas.foo.com». En inglés: «*Domain Name*».

Nombre de máquina: Es una cadena de caracteres entendible para las personas que se usa para identificar una máquina computadora en particular. En inglés: «hostname».

Núcleo: La parte central de un sistema operativo, sobre la cual el resto del sistema se apoya. En inglés: «kernel».



Operating System: Inglés, «sistema operativo».



Packages: Inglés, «paquetes».

PAM: Sigla de *Pluggable Authentication Modules* (Módulos enchufables de autenticación). PAM es un sistema de autenticación que controla el acceso a RHL.

Paquetes: Ficheros que contienen software; están escritos en un cierto formato que permite la fácil instalación y borrado del software. En inglés: «packages».

Partición: El segmento del espacio de almacenamiento de una unidad de disco que puede accederse como si fuese un disco entero. En inglés: «partition».

Partición extendida: Segmento de una unidad de disco que contiene otras particiones. En inglés: *Extended Partition*. Vea también «partición».

Partición lógica: Partición que existe dentro de una partición extendida. Vea también: «partición» y «partición extendida». En inglés: «*Logical Partition*».

Partition Table: Inglés, «tabla de particiones».

Partition Type: Inglés, «tipo de partición».

Partition: Inglés, «partición».

Pasarela: En términos de redes, se refiere al dispositivo que conecta uno o más ordenadores de una red a otra red. El dispositivo puede ser un hardware especializado (como un *router*), o puede ser una computadora de propósito general que se configura para actuar de pasarela. En inglés: «*Gateway*».

PC Card: Inglés, «PCMCIA».

PCMCIA: Sigla de *Personal Computer Memory Card International Association* (Asociación Internacional Tarjetas de Memoria para Computadoras Personales). Esta organización produce una serie de estándares que definen las características físicas, eléctricas y de software para pequeños dispositivos del tamaño de tarjetas de crédito que pueden contener memoria, modems, adaptadores de red, etc. También se las conoce como «PC Cards» (tarjetas para PC), estos dispositivos se usan principalmente en computadoras portátiles (aunque también algunos sistemas de escritorio pueden utilizar tarjetas PCMCIA).

PCMCIA Support Diskette: Inglés, «Disquete para soporte PCMCIA».

Permisos: El conjunto de identificadores que controlan el acceso a los ficheros. Los permisos constan de tres campos: usuario, grupo y mundo. El campo de usuario controla el acceso del propietario del fichero, y el campo de grupo controla el acceso de cualquiera que concuerda con la especificación de grupo del fichero. Como el nombre implica, el campo mundo controla el acceso de cualquier otro usuario. Cada campo contiene el mismo conjunto de bits que especifican las operaciones que pueden o no realizarse, tales como lectura, escritura y ejecución.

PLIP: Sigla de *Parallel Line Internet Protocol* (Protocolo de Internet para líneas paralelas). PLIP es un protocolo que permite comunicaciones TCP/IP sobre el puerto paralelo de la computadora, mediante el uso de un cable especialmente diseñado.

POSIX: Sigla un tanto forzada de *Portable Operating System Interface* (Interfaz portable de sistema operativo). Conjunto de estándares que crecieron a partir del sistema operativo UNIX.

Proceso: Un proceso (en términos simplísticos en cierto modo) es una instancia de un programa en ejecución sobre un sistema Linux. En inglés: «process».

Process: Inglés, «proceso».

PSI: Siglas de Proveedor de Servicios Internet. Empresa u organización que ofrece acceso a Internet a usuarios finales y corporativos.

Punto de montaje: El directorio bajo el cual se puede acceder a un sistema de ficheros luego de su montaje. En inglés: «Mount Point».

R

Raíz: (N. del T.) Traducción de la palabra «root». En determinados contextos se usa en castellano (ej.: «el directorio *raíz*»), mientras que en otros su traducción es desaconsejada (ej.: «el usuario root», en cuyo caso podría ser sinónimo de «superusuario» o «administrador»). Esta distinción favorece a los lectores hispanos pues quita algunas de las ambigüedades del término inglés.

RAM: Sigla de *Random Access Memory* (Memoria de acceso directo). La RAM se usa para mantener los programas mientras se están ejecutando, y los datos mientras se los procesa. La RAM es volátil, lo que significa que la información escrita en la RAM desaparecerá cuando se apague la alimentación de energía del ordenador.

Ratón PS/2: El ratón PS/2 toma su nombre a partir de la computadora original donde se comenzó a utilizar, la IBM PS/2. El ratón PS/2 puede identificarse fácilmente por el pequeño conector redondo en el extremo del cable.

Ratón serie: Un ratón serie es uno diseñado para conectarse al puerto serie del ordenador. El ratón serie puede identificarse fácilmente por el conector de forma rectangular que posee en el extremo de su cable.

Rearrancar: Recomenzar el proceso de arranque. En inglés: «reboot». Vea también «arranque».

Reboot: Inglés, «rearraancar».

Red Hat Software: Compañía de software sita en North Carolina. Produce y pone en el mercado software para el sistema operativo Linux, lo que incluye a Red Hat Linux.

Registro de arranque maestro: En inglés: «Master Boot Record» o más conocido por su sigla «MBR», es una sección del espacio de almacenamiento de la unidad de disco que se pone aparte con el propósito de guardar la información necesaria para comenzar el arranque en un ordenador personal.

Rescue Diskette: Inglés, «disquete de rescate».

RISC: Sigla de *Reduced Instruction Set Computer* (Computadora con juego reducido de instrucciones). Filosofía de diseño de computadoras en la cual el procesador está optimizado para ejecutar un número relativamente pequeño de instrucciones diferentes en una cantidad de tiempo predeciblemente pequeña.

ROM: Sigla de *Read Only Memory* (Memoria de sólo lectura). La ROM se usa para mantener los programas y datos que deben sobrevivir cuando se apaga el ordenador. Como la ROM no es volátil, los datos en la misma permanecerán sin cambios hasta la próxima vez que se encienda la computadora. Como el nombre implica, los datos no pueden escribirse con facilidad en la ROM; dependiendo de la tecnología que se usó en la ROM, la escritura puede requerir de un hardware especial, o incluso ser imposible. El BIOS del ordenador se almacena en ROM.

root: (raíz) El nombre de la cuenta de ingreso que da acceso completo y total a todos los recursos del sistema. También se usa para describir el directorio denominado con «/», como en la expresión «el directorio raíz».

RPM: Sigla de *Red Hat Package Manager* (Gestionador de paquetes de Red Hat). rpm es también el nombre del programa que permite la instalación, actualización y eliminación de paquetes.



SCSI: Sigla de *Small Computer System Interface* (Interfaz de sistema para pequeñas computadoras), SCSI es una interfaz estándar para conectar una amplia variedad de dispositivos a la computadora. Los dispositivos SCSI más populares son las unidades de disco, aunque también es común encontrar unidades de cinta y «scanners».

Sector: Cuando se refiere a una unidad de disco, la cantidad de áreas de tamaño fijo (normalmente 512 bytes) que se pueden acceder mediante una cabeza de lectura/escritura, en una rotación del disco, sin que la cabeza cambie su posición. Vea también «geometría».

Servidor de nombres: En términos de redes TCP/IP, un servidor de nombres es un ordenador que traduce un nombre entendible por personas (como «foo.bar.com») en una dirección numérica (como «10.0.2.14»). En inglés: «nameserver».

setgid: Llamada al sistema que puede usarse para asignar el GID de un proceso. Los programas grabados con el atributo «setgid» pueden adoptar el GID del grupo al cual pertenece el fichero programa.

setuid: Llamada al sistema que se usa para asignar el UID de un proceso. Los programas grabados con «setuid» pueden adoptar el UID del usuario que es

dueño del fichero programa. Esto se considera un posible problema de seguridad si el fichero es «setuid root».

Shadow Password: Normalmente, la contraseña de cada usuario se almacena en forma cifrada en el fichero */etc/passwd*. Este fichero debe poderlo leer cualquier usuario para que ciertas funciones del sistema trabajen correctamente. Sin embargo, esto significa que cualquiera puede obtener copias de las contraseñas cifradas de todos los usuarios, con lo cual resulta sencillo ejecutar un programa que adivine las contraseñas de los usuarios. Las «shadow passwords», por otro lado, almacenan la contraseña cifrada en un fichero distinto altamente protegido, lo que hace mucho más difícil el «crackeo» de contraseñas.

SILO: Cargador que se usa generalmente para sistemas Linux basados en el procesador SPARC.

Sistema de ficheros: Es el método mediante el cual se almacena la información en las unidades de disco. Los distintos sistemas operativos normalmente usan diferentes sistemas de ficheros, lo que dificulta el compartir los contenidos de una unidad de disco entre ellos. Sin embargo, Linux admite múltiples sistemas de ficheros, lo cual hace posible la lectura/escritura de particiones dedicadas a MS-Windows, por ejemplo. En inglés: «*Filesystem*»

Sistema operativo: Conjunto de software que controla los distintos recursos del ordenador. En inglés: «operating system».

SLIP: Sigla de *Serial Line Internet Protocol* (Protocolo de Internet para líneas serie). SLIP es un protocolo que permite la comunicación TCP/IP sobre líneas serie (típicamente una conexión por módem a través de la red telefónica conmutada).

SMB: Sigla de *Server Message Block* (Bloque de mensajes de servidor), SMB es el protocolo de comunicación que usan los sistemas operativos basados en MS-Windows para permitir los recursos compartidos a través de la red.

Source code: Inglés, «código fuente».

SPARC: Arquitectura RISC desarrollada por Sun Microsystems.

Supplemental Diskette: Inglés, «disquete suplementario».

Swap: Inglés, «intercambio».

System Call: Inglés, «Llamada al sistema».

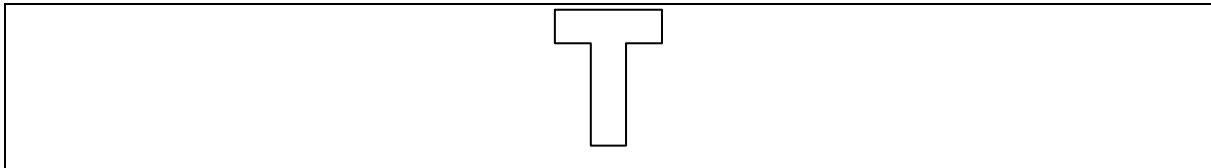


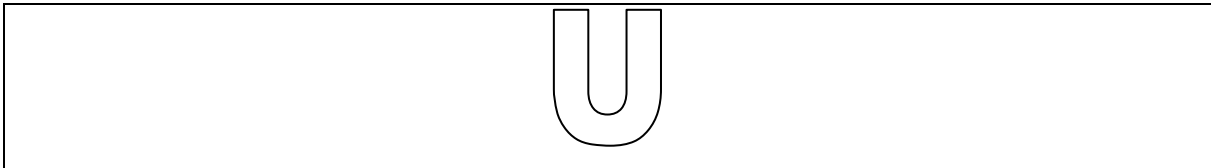
Tabla de particiones: La tabla de particiones es la sección del espacio de almacenamiento de la unidad de disco que se pone aparte para definir las particiones que existen en dicha unidad de disco. En inglés: «partition table».

TCP/IP: Sigla de *Transmission Control Protocol/Internet Protocol* (Protocolo de control de transmisión/Protocolo de Internet), TCP/IP es el nombre dado al estándar de redes de uso común en la actualidad en Internet.

Tipos de partición: Las particiones tienen un campo que se usa para determinar el tipo de sistema de ficheros que se espera que vaya a contener la partición. El tipo de partición es en realidad un número, aunque muchas veces nos referimos al tipo mediante un nombre. Por ejemplo, el tipo de partición denominado «Linux

Native» es el 82. Tenga en cuenta que este número es hexadecimal. En inglés: «Partition Type».

Torvalds, Linus: «Linus Torvalds».

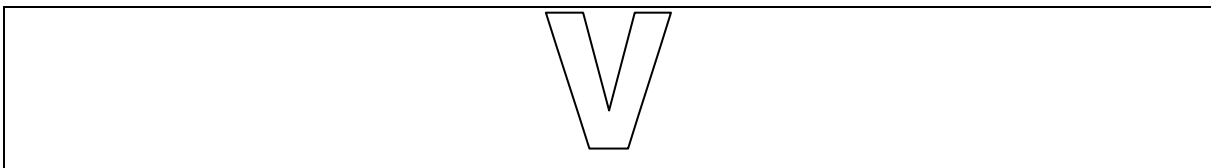


UID: Abreviatura de *User ID* (ID de usuario). Es el medio por el cual se identifica a un usuario en las distintas partes de un sistema RHL. Los UIDs son numéricos, aunque hay nombres en formato entendible por personas que se almacenan en el fichero */etc/passwd*.

Unidad de disco: «disco rígido».

UNIX: Conjunto de sistemas operativos del estilo de Linux que crecieron a partir de la versión original escrita por unos tipos de una compañía telefónica

Unmount: Inglés, «desmontaje».



Virtual Console: Inglés, «consola virtual».



Widget: Representación estandarizada en pantalla de un control que el usuario puede manipular. Ejemplos de «widgets» son las barras de desplazamiento, los botones y las cajas de texto.



X Window System: (Sistema de ventanas X) También denominado «X», esta interfaz gráfica de usuario proporciona la bien conocida metáfora de «ventanas sobre un escritorio», común a la mayoría de los sistemas hoy en día. Bajo X, los programas de aplicación actúan como clientes y acceden al servidor X que gestiona toda la actividad en pantalla. Además, las aplicaciones X pueden ejecutarse en un sistema distinto al del servidor X, lo que permite la visualización remota de las aplicaciones.

XFree86: Implementación libre del «X Window System».