INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA

UNIDAD CULHUACAN

SEMINARIO DE TITULACIÓN

"SEGURIDAD DE LA INFORMACIÓN"

TESINA

"IMPLEMENTACIÓN DE UN SERVIDOR DE FILTRADO DE CONTENIDO WEB EN UNA ESCUELA PÚBLICA DE NIVEL BÁSICO"

QUE PRESENTAN PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

NICOLÁS ORENDA LÓPEZ

GABRIEL ARTURO MAGALLANES MENDOZA

ASESOR:

DR. GABRIEL SÁNCHEZ PÉREZ

VIGENCIA: DES/ESIME-CUL-2008/23/2/10

MÉXICO, D.F. OCTUBRE 2010

IPN
ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD CULHUACAN

TESINA

POR LA OPCIÓN DE TITULACIÓN SEMINARIO EN SEGURIDAD DE LA INFORMACIÓN

QUE PARA OBTENER EL TÍTULO DE INGENIERO EN COMPUTACIÓN

DEBERÁN DESARROLLAR:

ORENDA LÓPEZ NICOLÁS
MAGALLANES MENDOZA GABRIEL ARTURO

"IMPLEMENTACIÓN DE UN SERVIDOR DE FILTRADO DE CONTENIDO WEB EN UNA ESCUELA PÚBLICA DE NIVEL BÁSICO"

INTRODUCCION

CON EL AVANCE DE LA TECNOLOGIA, LAS INSTITUCIONES EDUCATIVAS HAN INCROPORADO EN SUS AULAS COMPUTADORAS Y ACCESO A LA INTERNET, ESTO CON LA FINALIDAD DE OFRECER UNA MEJOR EDUCACIÓN, SIN EMBARGO, LA INTERNET ES UNA HERRAMIENTA MUY PODEROSA Y MUCHAS VECES SU CONTENIDO NO ES EL MÁS APROPIADO PARA LA EDUCACION DE LOS ALUMNOS DE ESTAS INSTITUCIONES EDUCATIVAS. POR TAL MOTIVO, ES INDISPENSABLE CONTAR CON UNA SOLUCIÓN QUE PERMITA CONTROLAR Y RESTRINGIR EL ACCESO A CIERTOS MATERIALES DE LA WEB, COMO PORNOGRAFÍA, VIOLENCIA; TANTO VISUAL COMO LÉXICA, ETC. ÉSTA IMPLEMENTACIÓN CONSISTE EN INTEGRAR UNA SERIE DE DE HERRAMIENTAS DE SOFTWARE LIBRE QUE DÉ COMO RESULTADO UN ROBUSTO Y EFICAZ SISTEMA DE FILTRADO DE CONTENIDO WEB.

CAPITULADO

I. HARDWARE
II. SOFTWARE
III. DESARROLLO
IV. RESULTADOS

México D.F., Octubre de 2010

VIGENCIA: DES/ESIME-CUL-2008/23/2/10

ING. ARTURO DE LA CRUZ TELLEZ
Instructor del seminario

DR. GABRIEL SÁNCHEZ PÉREZ
Asesor

M. EN C. LUIS CARLOS CASTRO MADRID
Jefe de la carrera de I.C.

DEDICATORIA

Este trabajo está dedicado con mucho cariño y amor a la memoria de Josefina López Sánchez quien me heredó el tesoro más valioso que puede darse a un hijo: el estudio. Ya que, sin escatimar esfuerzo alguno, sacrificó gran parte de su vida para formarme y educarme. A pesar de que no estás aquí siempre vivirás en mi corazón.

Quiero dar las gracias a Nicolás Orenda Ramírez, porque es un padre ejemplar que su única ilusión en su vida ha sido convertirme en una persona de provecho.

Gracias papá por darme un carrera para mi futuro y por creer en mí. Aunque hemos pasado momentos difíciles te agradezco de todo corazón que estés a mi lado.

De igual manera pero no menos importante quiero agradecer al Ing. Alejandro Edgar Zacatenco Santos por todo su tiempo y sus conocimientos que me brindo para realizar este trabajo. Gracias por haber tenido la paciencia necesaria y por tu consejo que nunca olvidaré: ¡¿para qué duermes?!.

Atentamente

Nicolás Orenda López

INDICE GENERAL

RESUMEN 1

ABSTRACT 2

INTRODUCCIÓN 3

OBJETIVOS 4

JUSTIFICACIÓN 5

CAPÍTULO I HARDWARE 7

1.1 Red 8

1.1.1. Topología de redes 9

1.1.2. La red en cuestión 13

1.2. El servidor integral 14

CAPÍTULO II *SOFTWARE* 15

2.1. Filtrado de contenido 16

2.2. Sistema operativo *GNU/Linux* 16

2.2.1. Distribuciones 20

2.2.2. Sistema operativo *Ubuntu server* 10.04 22

2.3. *DHCP* 23

2.4. *Firewall* 24

2.4.1. Aplicaciones de *firewall* 25

2.4.2. *Netfilter/iptables* 26

2.5. *Proxy* 29

2.5.1. *Squid* 30

2.5.2. *SquidGuard* 33

2.5.3. *Sarg* 34

2.6. Apache 35

CAPÍTULO III DESARROLLO 37

3.1. Instalación del sistema operativo *Ubuntu server* 10.04 38

3.2. Cambio a usuario *root* y configuración de interfaz de red. 53

3.3. Instalación de servidor *DHCP* 58

3.3.1. Configuración del archivo dhcpd.conf 58

3.4. Instalación de *squid* 60

3.4.1. Configuración del archivo squid.conf 60

3.5. Implementación de reglas de firewall y bit de reenvío 63

3.6. Instalación de *squidGuard* 64

3.6.1. Descarga del archivo *blacklists* 64

3.6.2. Configuración del archivo squidGuard.conf 67

3.7. Instalación de sarg 70

3.7.1. Configuración del archivo sarg.conf 71

3.8. Instalación de apache2 72

3.8.1. Configuración del archivo default 73

CAPÍTULO IV RESULTADOS 75

4.1 Pruebas 76

CONCLUSIONES 78

BIBLIOGRAFÍA 79

INIDICE DE FIGURAS

CAPÍTULO I

Figura 1.1 Topología de malla 9

Figura 1.2 Topología de estrella 10

Figura 1.3 Topología de árbol 11

Figura 1.4 Topología de anillo 12

Figura 1.5 Topología de bus 12

Figura 1.6 Red en cuestión 13

CAPÍTULO II

Figura 2.1 Cronología de sistema operativo *UNIX* 19

CAPÍTULO III

Figura 3.1. Página para la descarga de *Ubuntu server* 10.04 38

Figura 3.2. Descarga de la imagen *ISO* 38

Figura 3.3 Configuración de la *BIOS* para arrancar desde el CD-ROM 39

Figura 3.4 Guardar cambios en la *BIOS* y salir 39

Figura 3.5 Selección del idioma de instalación 40

Figura 3.6 Iniciar la instalación de *Ubuntu Server* 10.04 40

Figura 3.7 Configuración regional 41

Figura 3.8 Configuración del teclado 41

Figura 3.9 Prueba del teclado 42

Figura 3.10 Modelo del teclado 42

Figura 3.11 Configuración de interfaces de red..... 43

Figura 3.12 Barra del proceso de detección de hardware..... 43

Figura 3.13. Nombre del servidor..... 44

Figura 3.14 Configuración de la zona horaria..... 44

Figura 3.15 Particionado de discos. 45

Figura 3.16 Seleccionar disco duro..... 45

Figura 3.17 Formateo del disco..... 46

Figura 3.18 Barra de proceso de instalación del sistema base 46

Figura 3.19 Nombre completo del usuario 47

Figura 3.20 Creación de cuenta 47

Figura 3.21 Contraseña para el usuario 48

Figura 3.22 Cifrado de la carpeta personal 48

Figura 3.23 Configuración de gestor de paquetes..... 49

Figura 3.24 Barra de proceso de instalación. 49

Figura 3.25 Actualizaciones automaticas 50

Figura 3.26 Programas adicionales..... 50

Figura 3.27 Barra de proceso de instalación de programas básicos..... 51

Figura 3.28 Configuración *GRUB*..... 51

Figura 3.29 Instalación completa..... 52

Figura 3.30 Pantalla para entrar al sistema operativo 52

Figura 3.31 Información del sistema 53

Figura 3.32 Cambio de usuario 54

Figura 3.33 Interfaces de red 54

Figura 3.34 Configuración de eth1 55

Figura 3.35 Configuraciones de red con alias 55

Figura 3.36 Verificación de las interfaces de red configuradas 56

Figura 3.37 Acceder al archivo interfaces 56

Figura 3.38 Configuración del archivo interfaces 57

Figura 3.39 Pantalla de inicio con la interfaz de red configurada 57

Figura 3.40 Instalación del servidor *DHCP* 58

Figura 3.41 Acceder al archivo de configuración dhcpd.conf 58

Figura 3.42 Archivo de configuración dhcpd.conf 59

Figura 3.43 Funcionando el servidor *DHCP* 59

Figura 3.44 Instalación de *squid* 60

Figura 3.45 Acceder al archivo de configuración squid.conf 60

Figura 3.46 Configuración de la *acl* 61

Figura 3.47 Configuración del puerto de *squid* 61

Figura 3.48 Configuración de la memoria cache 62

Figura 3.49 Configuración del disco cache 62

Figura 3.50 Funcionando *squid* 63

Figura 3.51 Reglas para redireccionar al puerto 3128 63

Figura 3.52 Bit de reenvío. 64

Figura 3.53 Instalación de *squidGuard* 64

Figura 3.54 Descarga del archivo blacklists.tar.gz 65

Figura 3.55 Descompresión del archivo *blacklists*.tar.gz 66

Figura 3.56 Conversión de categorías en formato de base de datos 66

Figura 3.57 Acceder al archivo de configuración *squidGuard*.conf 67

Figura 3.58 Configuración del horario *squidGuard*.conf 67

Figura 3.59 Rango de direcciones a restringir el acceso a páginas web 68

Figura 3.60 Categorías por dominio y *url* 68

Figura 3.61 Denegando accesos por subred 69

Figura 3.62 Creación de archivos en formato de base de datos 69

Figura 3.63 Comprobación de los archivos de base de datos 70

Figura 3.64 Instalación de *sarg* 70

Figura 3.65 Acceso al archivo *sarg*.conf 71

Figura 3.66 Archivo de configuración *sarg*.conf 71

Figura 3.67 Archivo *crontab* 72

Figura 3.68 Instalación del servidor apache2 73

Figura 3.69 Configuración de archivo default de apache2 73

Figura 3.70 Archivo index.html 74

Figura 3.71 Página de error 74

INDICE DE TABLAS

CAPÍTULO II

Tabla 2.1 Significado de los dígitos de la versión del *kernel* 19

Tabla 2.2 Directorios *Ubuntu Server* 10.04 23

Tabla 2.3 Correspondencia de tablas y cadenas de *iptables* 27

Tabla 2.4 Modificadores más comunes de *iptables* 28

Tabla 2.5 Parámetros de *iptables* 28

Tabla 2.6 Acciones de *iptables* 29

RESUMEN

La implementación de un servidor de filtrado de contenido web en una escuela pública de nivel básico, es una alternativa con el fin de restringir el acceso a páginas web no aptas para los alumnos de estas instituciones, permitiendo así una navegación segura y con calidad informativa.

Consiste en un servidor con sistema operativo *Linux* que asigna direcciones *IP* dinámicas en una red de área local con el servicio *DHCP*. Todas las peticiones de los clientes de la red local que son dirigidas a la internet son redireccionadas al servidor *proxy* de *Squid*. Con el programa *SquidGuard* que contiene diferentes categorías con archivos en formato de base de datos, con millones de direcciones web para filtrar, evalúa la petición del cliente, si su petición no se encuentra en algunas de las categorías que están prohibidas entonces permite la conexión, en caso contrario, con el servicio de Apache despliega una página en formato *HTML* informando que el acceso a esa página está bloqueado.

ABSTRACT

The implementation of a server web content filtering in a public school of basic level, it is an alternative in order to restrict access to not suitable web pages for students of these institutions, allowing this way a sure navigation and with informative quality.

It consists in a server with *Linux* operating system that assigns dynamic IP addresses on a local area network with DHCP service. All the requests from the local network clients that are for the Internet are redirected to the Squid proxy server. With the help of the program Squid Guard different categories containing database files with millions of web addresses to prohibit, evaluates the request client, if your request is not in the categories then allows the connection otherwise the Apache service displays an HTML page sent informing that access to that page is blocked.

INTRODUCCIÓN

Con el avance de la tecnología las instituciones educativas han incorporado en sus aulas computadoras y acceso hacia la internet, esto con la finalidad de ofrecer una mejor educación, sin embargo, la internet es una herramienta muy poderosa y muchas veces su contenido no es el más apropiado para la educación de los alumnos de estas instituciones.

Las estadísticas muestran que 9 de cada 10 niños con una edad que oscila entre 6 y 17 años han visto pornografía en la internet, en la mayoría de los casos accediendo en forma accidental mientras realizaban sus tareas o trabajos de investigación.

Por tal razón es indispensable contar con una solución que permita controlar y restringir el acceso a ciertos materiales de la web como pornografía violencia etc. Esta implementación consiste e integrar una de serie de herramientas de *software* libre quede como resultado un robusto y eficaz sistema de filtrado de contenido web, además cuenta con una cache de páginas web para acelerar la navegación y un reporte de estadísticas web para visualizar horarios, fechas, sitios web accedidos con mayor frecuencia etc.

La implementación de un servidor de filtrado de contenido web tiene la finalidad de garantizar la protección del menor con contenidos inapropiados para su edad.

OBJETIVO GENERAL

Diseñar e implementar un filtrado de contenido web para restringir el acceso a contenido web no apto para niños en una escuela pública de nivel básico mediante el uso e implementación de un servidor basado en *Linux*.

OBJETIVOS PARTICULARES

- Instalar el sistema operativo *Ubuntu server* 10.04
- Instalar y configurar un servidor *DHCP* multisegmento
- Implementar reglas simples de Firewall mediante *Netfilter*
- Instalar y configurar un servidor *proxy* basado en *squid*
- Instalar y configurar una herramienta de filtrado de contenido *squidguard*
- Instalar y configurar un servidor de contenido *web* basado en *apache*
- Instalar y configurar un intérprete de bitácoras de *squid* denominado *sarg*

JUSTIFICACIÓN

Actualmente el uso de la internet para mejorar el desarrollo del aprendizaje está adquiriendo día a día una mayor relevancia y presencia en el sector educativo. El secretario de educación pública Alonso Lujambio en una entrevista para el periódico Excélsior, señaló que el avance de la tecnología obliga a introducir en las escuelas nuevas herramientas para que los alumnos de primaria y secundaria tengan acceso a la internet y reciban clases por medio de una computadora para que obtengan un mejor rendimiento escolar[1]. Según datos de la *SEP* (Secretaría de Educación Pública) en el estudio realizado principales cifras ciclo escolar 2008-2009, la mayoría de las escuelas de educación básica en México cuentan con laboratorios o aulas de cómputo, equipadas con computadoras e internet[2].

El estudio realizado por la AMIPIC (Asociación Mexicana de Internet) en el año de 2009[3] muestra que uno de los servicios más populares que posee la internet es la web, en donde se pueden escuchar y visualizar páginas web que contienen texto, imágenes, videos u otros contenidos de multimedios que abundan sin control en la red.

Según datos proporcionados por la INEGI (Instituto Nacional de Estadística Geográfica e Informática), en la encuesta nacional sobre la disponibilidad y uso de

[1] Hernández L, L.(2010 mayo). *Se asoma la nueva tecnología para las aulas.* Excélsior [en línea], Histórico. Recuperado el 27 de Octubre de 2010 de http://www.excelsior.com.mx/index.php?m=nota&buscado=1&id_nota=4597

[2] Lujambio I., A. (2009, octubre) *Principales cifras ciclo escolar 2008-2009 (Primera edición),[en línea]*.México D.F.: Secretaría de Educación Pública. Recuperado el 27 de Octubre 2010, de http://www.sep.gob.mx/work/models/sep1/Resource/890/1/images/PrincipalesCIF2008-2009.pdf

[3] Hábitos de internet (2009) *Principales actividades de entretenimiento de los internautas* Asociación Mexicana de Internet [en línea]. Recuperado el 27 de Octubre de 2010 de http://estudios.amipci.org.mx:8080/mashboard/main.jsp

las tecnologías de la información en los hogares, muestra que los principales usuarios de computadora tienen una edad de 12 a 17 años con un 29.6% mientras que otros usuarios de computadora tiene una edad de 6 a 11 años[4], principalmente con estos dos grupos de edades es evidente que los niños no tienen problemas para utilizar una computadora y mucho menos para la navegación en las páginas web, ya que es un hecho que éste constituye una fuente de información para la elaboración de trabajos escolares, tareas e investigaciones.

Sin embargo, del mismo modo que el mundo real, el mundo de la internet también puede ser peligroso, como lo explica Jorge Christian Duran Lara en su artículo de los niños del internet "*los niños y adolecentes utilizan el internet, en su mayoría, para complementar sus estudios, quienes podrían estar en riesgo de experimentar exposición inadvertida de contenido no apropiado en forma de imágenes, videos o contenido*"[5] es tal el cumulo de información, que uno de los grandes inconvenientes que tiene la web, como recurso pedagógico, estriba precisamente en que no todo el contenido de la web es algo educativo, puesto que si en un buscador web, por ejemplo, se busca la palabra "zorra", se obtienen diversos significados de los cuales se puede tener acceso a miles de páginas web con contenido no apto para niños, muchas de ellas con material pornográfico o agresivo. Por tal motivo, la implementación de un servidor de filtrado de contenido web, en sitios tales como una escuela pública de nivel básico, restringirá el acceso a páginas web cuyo contenido no sea apto para niños y con esto se garantizará que la navegación de calidad sea basta en su riqueza informativa y que cumpla su fin pedagógico con los alumnos de estas instituciones.

[4] *Encuesta nacional sobre la disponibilidad y uso de la información en los hogares* (2005). Instituto Nacional de Estadística Geográfica e Informática [en línea] Usuarios de computadora. Recuperado el 27 de Octubre de 2010 de http://www.canieti.org/assets/files/458/17%20de%20Mayo%20D%C3%ADa%20Mundial%20de%20Internet.pdf

[5] Durán J., C. (2010 mayo).*Los niños del internet* [en línea] Universidad Nacional Autónoma de México: Punto de seguridad. Recuperado el 27 de Octubre de 2010 de http://revista.seguridad.unam.mx/rs_unam_06/006_03/art_03.html

CAPÍTULO I

HARDWARE

1.1 Red

Es un conjunto de computadoras o dispositivos conectados entre sí, por medio de un enlace físico, que permiten establecer comunicación a través de ellas[6] . Algunas de las ventajas primordiales son:

- Compartir programas: Algunas empresas brindan la posibilidad de adquirir licencias de uso en red de *software* especializado o popular, con lo que se puede ahorrar una considerable suma de dinero, si se comparan con el hecho de comprar licencias individuales, ya que de acuerdo a esta modalidad, se compra un solo programa con el número de licencias necesario para el número de usuarios que van a utilizar la aplicación.

- Compartir recursos: Entre los periféricos que se pueden compartir en la red se encuentran impresoras dispositivos de almacenamiento masivo tales como discos duros y unidades de CD ROM, etc., los cuales pueden ser configurados para que estén disponibles para cualquier usuario en la red local, con la finalidad de permitir optimizar el uso de los recursos, puesto que los usuarios, desde sus maquinas o estaciones de trabajo, pueden llevar a cabo sus tareas de manera remota, , tales como la impresión de archivos, acceso a archivos compartidos, etc.

- Compartir servicios: Entre los servicios más comunes son e-mail, chat accesos a la internet, juegos, etc.

[6] Behrouz A. ,F. (2002). *Transmisión de datos y redes de comunicaciones*. Madrid, España: Mc Graw Hill, p. 4

1.1.1. Topología de redes

Es la forma en que está diseñada la red, ya sea física o lógicamente, la topología física es la representación geométrica; bajo la cual se describe como están distribuidos, organizados o interconectados; todo el conjunto de computadoras o dispositivos que constituyen una red, la topología lógica es la forma en cómo fluye la información a través de la red[8]. Para fines de este proyecto sólo se explicaran brevemente las topologías físicas.

Existen cinco posibles topologías físicas básicas:

- ***Topología de Malla***: Todos los dispositivos están interconectados entre sí. De esta manera es posible que la información se transmita de un dispositivo a otro por diferentes caminos. Si una conexión es terminada o interrumpida, otra conexión puede ser elegida para transferir la información[9]. La topología de malla se puede ver en la figura 1.1.

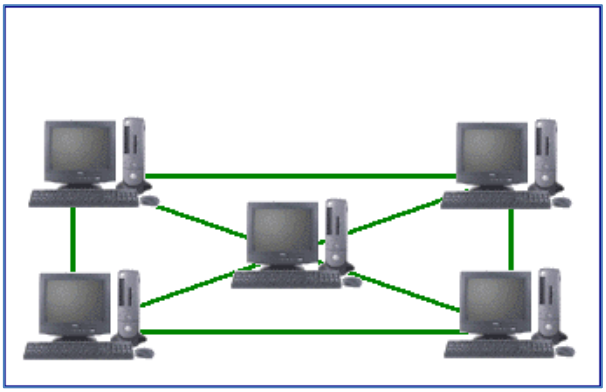

Figura 1.1 Topología de malla.

[8] Behrouz A. ,F. (2002). *Transmisión de datos y redes de comunicaciones*. Madrid, España: Mc Graw Hill, p. 22
[9] Ibidem, p. 23

- ***Topología de estrella***: Todos los dispositivos están conectados directamente a un punto central (router, switch o hub) y todas las comunicaciones se hacen a través de él[10]. Si un dispositivo quiere enviar información a otro, envía la información al punto central y este los retransmite al dispositivo final. Si el punto central falla quedará toda la red interrumpida, si es un dispositivo de los extremos, sólo este quedará aislado. La topología en estrella se puede ver en la figura 1.2.

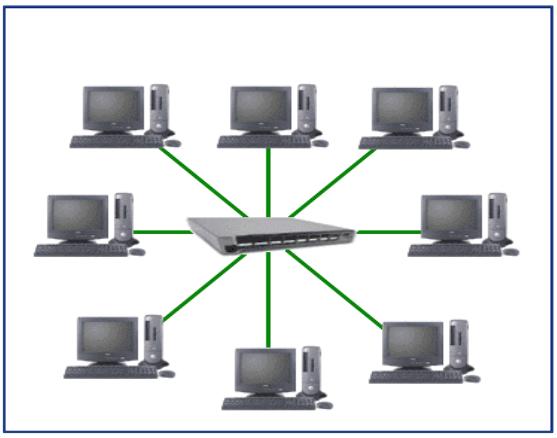

Figura 1.2 Topología de estrella.

- ***Topología de árbol:*** Es una variante de la topología de estrella. En la topología de árbol los dispositivos se conectan en una estructura jerárquica, es decir, la mayoría de los dispositivos se conectan a un punto central secundario y éste a su vez se conecta a un punto central general. Si falla un punto central secundario deja incomunicados todos los dispositivos que se conectan a él[11]. La topología de árbol se puede ver en la figura 1.3, en donde no todos los dispositivos se conectan directamente al concentrador central. La mayoría de los dispositivos se

[10] Behrouz A. ,F. (2002). *Transmisión de datos y redes de comunicaciones*. Madrid, España: Mc Graw Hill, p. 25
[11] Idem

conectan a un concentrador secundario que a su vez se conecta al concentrador central.

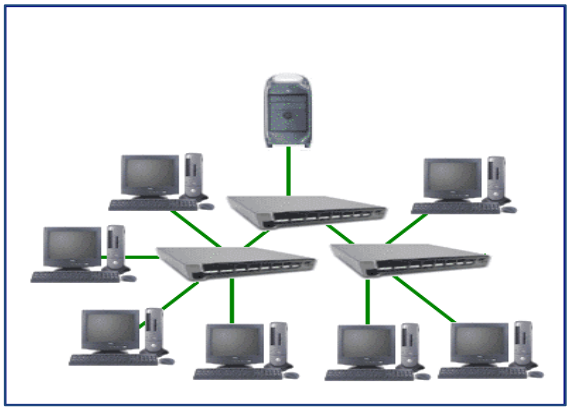

Figura 1.3 Topología de árbol.

- ***Topología de anillo:*** Todos los dispositivos están conectados a una única vía con los dos dispositivos que están a sus lados, la señal pasa a lo largo del anillo en una dirección, o de dispositivo a dispositivo, hasta que alcanza su destino. Cada dispositivo del anillo incorpora un receptor y un transmisor que hace la función de repetidor, cuando un dispositivo recibe una señal para otro dispositivo, su repetidor regenera los bits y los retransmite pasando la señal al siguiente dispositivo. Si un dispositivo falla la red deja de funcionar completamente[12]. La topología de anillo se puede ver en la figura 1.4.

[12]Behrouz A. ,F. (2002). *Transmisión de datos y redes de comunicaciones*. Madrid, España: Mc Graw Hill, p. 27

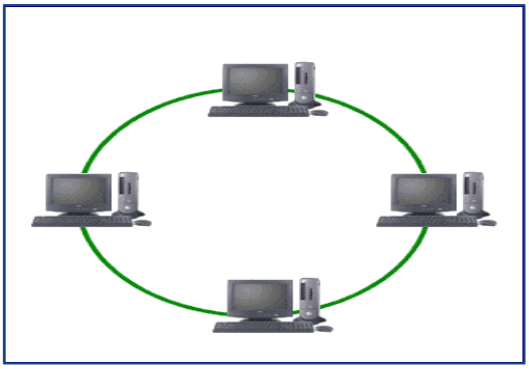

Figura 1.4 Topología de anillo

- ***Topología de bus:*** Todos los dispositivos están conectados directamente a un único canal de comunicación (denominado bus, troncal o *backbune*) y no tienen ninguna otra conexión entre ellos. Físicamente cada dispositivo está conectado a un cable común, por los que se pueden comunicar directamente entre ellos. Requiere un dispositivo llamado terminal, el cual se encuentra al final del canal de comunicación, la ruptura del cable hace que los dispositivos queden desconectados y la red falle total o parcialmente, en función del lugar en que se produzca la ruptura, en cambio, si un dispositivo falla, simplemente deja de comunicarse[13]. La topología de bus se puede ver en la figura 1.5.

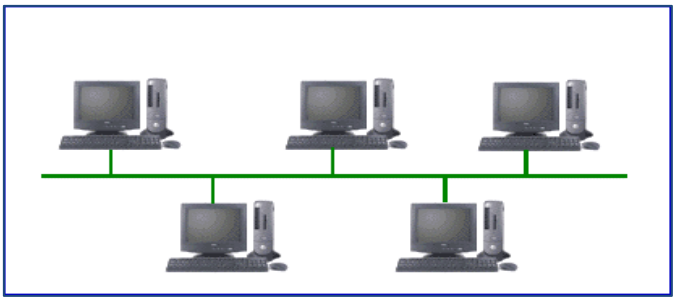

Figura 1.5 Topología de bus.

[13] Behrouz A. ,F. (2002). *Transmisión de datos y redes de comunicaciones*. Madrid, España: Mc Graw Hill, p. 26

1.1.2. La red en cuestión

La red en la que se implementará el servidor integral, cuya función principal será el filtrado de contenido web, está integrado por tres subredes, por motivos de seguridad[14], ya que se dispone sólo de una salida a la internet y se requiere aislar acceso a los recurso compartidos de la dirección, por tal, la primera subred es para la dirección y el área administrativa, la segunda subred es para la sala de profesores, la cual también dispondrá de recursos compartidos, y la tercera subred es para la aula de computo y puntos de acceso que usarán los alumnos para conectarse a la internet. Las tres subredes recibirán los servicios del servidor principal y estarán conectadas físicamente a la misma infraestructura de red, la división en subredes se realiza mediante el *DHCP,* el cual contendrá una pequeña lista de acceso que contiene la información necesaria para que el servidor *DHCP* reconozca a cada uno de los equipos y les asigne la ubicación en la subred correspondiente. El esquema de la red se puede ver en la figura 1.6.

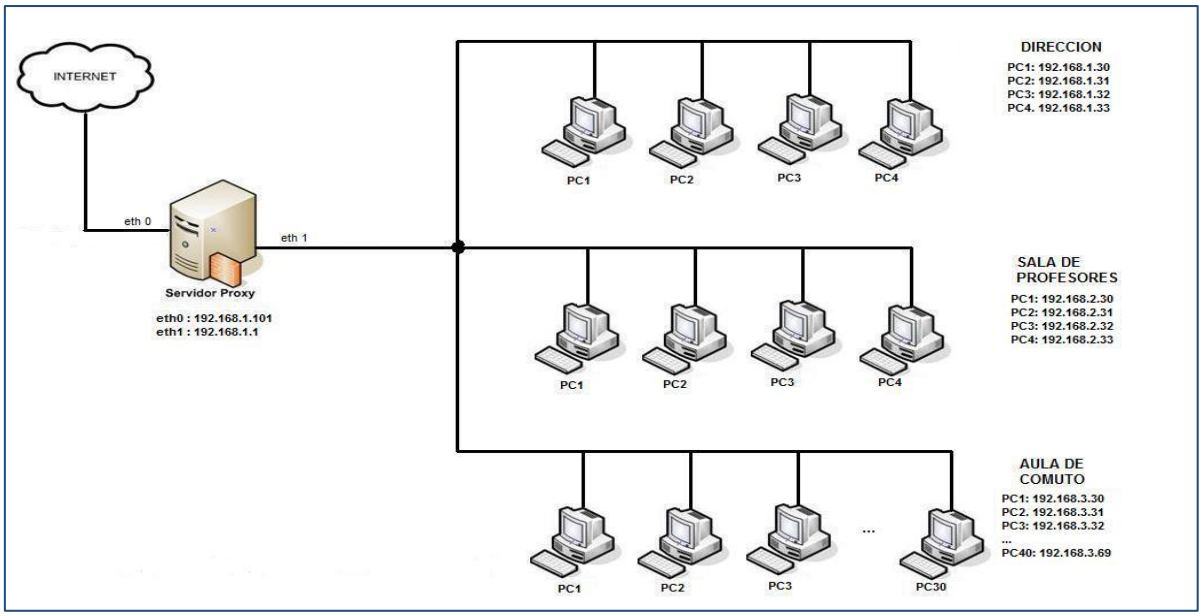

Figura 1.6 Red en cuestión.

[14] Denegación de servicio (DoS): Cualquier acción realizada por una persona u otra causa que inhabilite el software, hardware o ambos en un dispositivo dentro de una red.

1.2 El servidor integral

Para efectos de este proyecto, se usará un equipo, cuyas características se mencionan abajo, el cual será habilitado como servidor integral y que dará servicio a las tres subredes antes mencionadas.

- Procesador *Intel Pentium* 4 a 1.8 GHz
- Memoria RAM de 1 GB DDR
- Disco duro de 40 GB IDE de 7200 RPM
- 2 Tarjetas de red base 10/100 (eth0 y eth1)
- Lectora de CD ROM

CAPÍTULO II

SOFTWARE

2.1 Filtrado de contenido

Es la acción que realiza un servidor mediante el uso de un *software* que permite restringir el acceso a ciertos contenidos de la web. El filtrado de contenido web determina a qué contenido podrá acceder computadora o una red completa. El *filtrado de contenido web* es usado regularmente para evitar el acceso a los contenidos nocivos o distractores, que se encuentran en la internet, los cuales pueden ser: pornografía, violencia, racismo, sectas, horóscopos, etc. Si algún usuario dentro de la red protegida por un filtro de contenido intentase acceder a alguna página de la red, que haya sido previamente etiquetada como "no permitida", el servidor con filtrado de contenido web redireccionará la petición a otro sitio o a alguna página previamente diseñada que contenga alguna advertencia.

2.2 Sistema operativo *GNU/Linux*

En los inicios de la informática los ordenadores eran máquinas pesadas y caras que sólo se podían encontrar en las universidades y centros de investigación. En los años setenta el sistema de referencia era *UNIX* propiedad de los laboratorios *Bell* de la compañía *American Telephone & Telegraph* (*AT&T*)[15], Tradicionalmente las versiones de *UNIX* fueron numeradas según la edición[16]. En 1981, *AT&T*, alteró el *UNIX*, haciendo algunas modificaciones particulares y lanzo *System* III. En 1983,

[15] Lima, J. *Historia y evolución de Linux/Unix desde 1991.* [en línea]. Recuperado el 28 de octubre de 2010 de http://www.unixsup.com/unixlinux/historiaunixcuxs.html

[16] *Bell Labs Early Contributions to Computer Science.* [en línea]. Recuperado el 28 de Octubre de 2010 de http://www.bell-labs.com/history/unix/blcontributions.html

después de una serie de cambios, fue lanzado el conocido *UNIX System* V[17], cuyo código era distribuido libremente a empresas y universidades por un precio simbólico. Como el código era distribuido libremente pronto empezaron a aparecer variantes mejoradas del sistema. Unas de las más importantes fue la desarrollada en la Universidad de California en *Berkeley*. Esta versión se conoció por sus siglas *BSD* (*Berkele Software Distribution*), lamentablemente estas mejoras introducidas por *BSD* se comercializaron por un bajo costo[18].Esto dio pie a que *AT&T* requiriera el pago de grandes cantidades de dinero por las nuevas versiones de su sistema operativo y se produjo la mayor división en el mundo *UNIX*. Esta división dio lugar a las dos principales variantes de *UNIX* que son las basadas en *BSD* y las basadas en System V[19]

Las compañías de *software* comercial vieron la posibilidad de hacer negocio y lanzaron sus propias versiones del sistema *UNIX* basadas en *BSD* o System V. Así nacieron las diferentes variantes de *UNIX* que son: *AIX* de *IBM*, *HP/UX* de *Hewlett.-Packard*, *IRIX* de *Silicon Graphiics*, *Sun OS*, *Xenix* de *Microsoft* etc.[20]

En 1984 un nuevo mercado comenzaba a tomar forma: la informática doméstica. Los ordenadores se abarataron, se hicieron más ligeros y comenzaron a invadir los hogares. Las empresas obligaron a sus programadores a firmar acuerdos de no revelación, por los que se comprometían a cerrar el código, y los programas comenzaron a venderse sin facilitar su código fuente[21].

Esto generó una reacción de rechazo que se hizo patente cuando Richard Matthew Stallman decidió iniciar el proyecto de crear un sistema operativo similar a *UNIX*,

[17] *Bell Labs Early Contributions to Computer Science.* [en línea]. Recuperado el 28 de Octubre de 2010 de http://www.bell-labs.com/history/unix/blcontributions.html
[18] *Berkeley Unix and the Birth of Open-Source Software* [en línea]. Recuperado el 28 de Octubre de 2010 de http://coe.berkeley.edu/labnotes/history_unix.html
[19] Shah, S. y Soyinka W. (2007). *Manual de administración de Linux.*México: Mc Graw Hill. p 6
[20] Lima, J. *Historia y evolución de Linux/Unix desde 1991.* [en línea]. Recuperado el 28 de octubre de 2010 de http://www.unixsup.com/unixlinux/historiaunixcuxs.html
[21] Idem

pero con una licencia que permitiese el acceso al código fuente, además de la libre distribución y copia.

Llamó al proyecto *GNU*, acrónimo recursivo que significa "*GNU*[22] *is Not UNIX*". Para proteger al nuevo sistema se creó la licencia *GNU/GPL* (Licencia Pública General *GNU*) y el copyleft (opuesto al copyright), que garantiza la libertad de uso, copia y modificación, y obliga a distribuir el código fuente junto con los binarios.
El proyecto *GNU* tuvo una gran acogida. Cientos de programadores de todo el mundo se identificaron con su manifiesto fundacional y comenzaron a colaborar y producir componentes del futuro sistema operativo libre. En 1985 Stallman creó la *Free Software Fundation* (FSF) para dar cobertura legal al proyecto y canalizar las ayudas económicas[23].

EN 1990 el sistema *GNU* estaba casi completo pero faltaba un componente esencial: el núcleo o *kernel*. El *kernel* es la parte esencial de un sistema operativo que provee los servicios más básicos del sistema. Se encarga de gestionar el hardware (microprocesador, memoria *RAM*, etc.) de la computadora y los periféricos conectados a él, a demás de proveer al resto de los programas acceso al hardware y de gestionar la ejecución de todos los programas. Sin núcleo no puede haber sistema operativo y aunque se había trabajado en *GNU* el *Hurd,* no se habían conseguido resultados efectivos.[24]

En 1991 Linus Torvalds, un estudiante finlandés de 21 años desarrolló un núcleo compatible con *UNIX* y lo denominó *Linux*. Todo comenzó como un proyecto fin de carrera: se trataba de programar un núcleo para sistema operativo inspirado en *Minix*, un pequeño *UNIX* desarrollado por el profesor Andrew Tanenbaum. En enero de 1992 se publicó la versión 0.02, y poco tiempo después, en marzo de 1994 se liberó la versión 1.0.0, ya lista para sistemas en producción. A partir de esta versión al combinar *Linux* con el sistema no completo *GNU* resultó un sistema operativo libre

[22] Ñú en inglés
[23] Shah, S. y Soyinka W. (2007). *Manual de administración de Linux.*México: Mc Graw Hill. p 6
[24] Sánchez, S. (1999). *Unix y Linux: Guía práctica. México:* Alfaomega. p 5

completo, cuyo nombre correcto es *GNU/Linux*[25]. En la figura 2.1 se puede observar la cronología de sistema operativo *UNIX*.

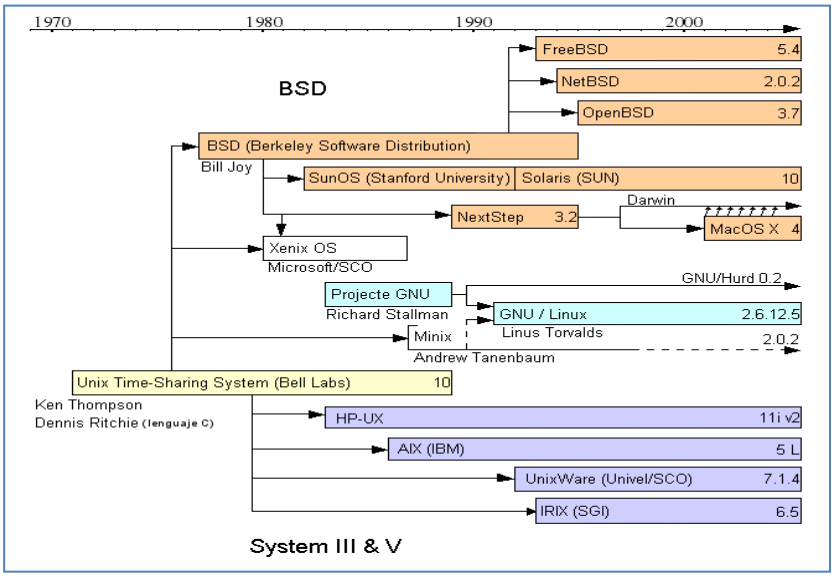

Figura 2.1 Cronología del sistema operativo *UNIX*[26]

En 1996 vería la luz el *kernel* 2.0.0 ya asociado con la mascota del pingüino (llamada Tux). Hoy en día las versiones del *kernel* por arriba de la 2.6 se enumeran con 4 dígitos (W,X.Y,Z)[27] en donde el significado de cada letra se muestra en la siguiente tabla:

Tabla2.1 Significado de los dígitos de la versión del *kernel*

Digito	Descripción
W	Indica la versión del *kernel*
X	Indica la revisión del *kernel*
Y	Indican las nuevas versiones del *kernel*. Estos números cambian cuando se incorporan nuevas características y drivers.
Z	Este dígito cambia cuando se corrigen fallos de programación o fallos de seguridad dentro de una versión

[25] Sánchez, S. (1999). *Unix y Linux: Guía práctica. México:* Alfaomega. p 6
[26] Martínez, R. (1998) *Kernel/Núcleo* [en línea]. Recuperado el 28 de Octubre de 2010 de http://www.linux-es.org/sobre_linux
[27] Martínez, R. (1998) *Kernel/Núcleo* [en línea]. Recuperado el 28 de Octubre de 2010 de http://www.linux-es.org/kernel

En la actualidad el *kernel Linux* va por la versión 2.6.36[28] (20 de Octubre 2010), está disponible para una gran variedad de arquitecturas y goza de un gran prestigio en la comunidad informática como fiable, sólido y seguro.

2.2.1. Distribuciones

Una distribución o *distro* es un sistema *GNU*/*Linux* que integra un *kernel*, librerías, un conjunto de aplicaciones de sistema y una colección de programas de usuario listo para instalar. Se encuentran empaquetadas de una determinada manera y con utilidades extras para facilitar la configuración[29].

Los programadores de las distintas distribuciones realizan un importante esfuerzo por recopilar lo mejor del *software* libre disponible, mejorar los procesos de instalación con el fin de facilitar la vida al usuario medio: recopilan el mejor *software* disponible, mejoran la detección de dispositivos y los entornos gráficos, implementan procesos de instalación automatizados, etc.

Algunas distribuciones son conocidas como mayoritarias, pues poseen un desarrollo sostenido e independiente, otras son basadas en las anteriores tomando de éstas una parte de sus características agradables y modificando otras[30]. Entre las distribuciones más conocidas y utilizadas pueden citarse a:

- *Redhat*: Creada por la compañía *RedHat*. Ofreció soporte hasta la versión 9 debido a que decidió concentrar sus esfuerzos en el desarrollo de la versión corporativa *RedHat Enerprise Linux* y delegó la versión común a *Fedora Core*.

[28] *The Linux kernel archive* [en línea].Recuperado el 28 de Octubre de 2010.de shttp://www.kernel.org/
[29] Pérez, C.M. y Pérez I.C. (1998).*Linux: Guía práctica para usuarios.* Madrid, España: Anaya multimedia. p 27
[30] Peterson, R. (2001). *Linux: Fundamentos de programación.* Bogotá, Colombia: Mc Graw Hill. p 25

- *Fedora Core*: Es una distribución enteramente libre desarrollada por la comunidad de *RedHat*. *Fedora* es generalista y está enfocada a una amplia comunidad de usuarios

- *Mandriva*: Antes conocida como *Mandrake* y rebautizada tras una fusión empresarial

- *Suse*: Es una de las principales distribuciones *GNU*/*Linux* existentes a nivel mundial, nacida en Alemania. Entre las principales virtudes de esta distribución se encuentra el que sea una de las mas sencillas de instalar y administrar, ya que cuenta con varios asistentes gráficos para completar diversas tareas.

- *Slackware*: Fue creada en 1993 y es de las más veterana de las distribuciones *GNU*/*Linux*. Su meta ha sido siempre su simplicidad y la estabilidad. La interface del programa de instalación es de texto, y necesita un mayor conocimiento de *Linux* que la mayoría de las otras distribuciones.

- *Debian*: Otra distribución con muy buena calidad. El proceso de instalación es quizás un poco más complicado, pero sin mayores problemas, y tiene una gran estabilidad

- *Ubuntu*: Distribución basada en *Debian*, por lo que ésta está enfocada en la gente común y esto conlleva a su facilidad de uso. Muy popular y con mucho soporte en la comunidad. El entorno de escritorio por defecto es GNOME, aunque existen versiones enfocadas a otros entornos de escritorio, como *kubuntu*.

2.2.2. Sistema operativo *Ubuntu Server* 10.04

Ubuntu está basado en *Debian*, *Ubuntu* pretende crear una distribución que proporcione un sistema *GNU*/*Linux* actualizado y coherente para la informática de escritorio y servidores. *Ubuntu* incluye una cuidadosa selección de los paquetes de *Debian*, y mantiene su poderoso sistema de gestión de paquetes que nos permite instalar y desinstalar programas de una forma fácil y limpia. A diferencia de la mayoría de las distribuciones, que vienen con una enorme cantidad de *software* que puden o no ser de utilidad, la lista de paquetes de Ubuntu se ha reducido para incluir sólo aplicaciones importantes y de alta calidad[31].

Con la mirada puesta en la calidad, *Ubuntu* proporciona un entorno robusto y funcional, adecuado tanto para uso doméstico como profesional y se publica una nueva versión cada seis meses. La numeración de las versiones de Ubuntu indica la fecha de lanzamiento de la distribución, más concretamente el año y el mes. El primer lanzamiento fue en Octubre del 2004, por lo tanto la versión fue la 4.10. La versión actual fue lanzada en Abril del 2010 por lo que su número de versión es 10.04. *Ubuntu* está disponible para las arquitecturas i386 (procesadores 386/486/*Pentium* (II/III/IV) y *Athlon*/*Duron*/*Sempron processors*), AMD64 (*Athlon64*, *Opteron* y los nuevos procesadores *Intel* de 64 bits), *PowerPC* (*iBook*/*Powerbook*, G4 y G5) y *ARM*[32].

Todos los elementos del sistema son tratados como ficheros desde nuestros archivos personales hasta los dispositivos *hardware* como la impresora, el ratón, los dispositivos de almacenamiento, etc. Estos ficheros están

[31] *Ubuntu* [en línea]. Recuperado el 28 de Octubre de 2010 de http://doc.ubuntu-es.org/Sobre_Ubuntu
[32] *Versiones de Ubuntu* [en línea]. Recuperado el 28 de Octubre de 2010 de http://www.guia-ubuntu.org/index.php?title=Versiones_de_Ubuntu

organizados en lo que se conoce como un sistema de ficheros. El sistema de fichero nativo de *Ubuntu server* 10.04 es ext4[33]

Ésta estructura se encuentra fuertemente jerarquizada para permitir una mayor familiaridad con el sistema, la mayoría de los directorios de Ubuntu se encuentran siempre en el mismo lugar que cualquiera de las distribuciones de *GNU*/*Linux*[34]. A continuación en la tabla 2.2 se mencionan algunos de los directorios para comprender la lógica del sistema.

Tabla 2.2 Directorios Ubuntu Server 10.04

Directorio	Descripción
/bin	Contiene los ejecutables (binarios) esenciales para el sistema. Si se observa el contenido se encuentran los comandos más básicos
/boot	Contiene los archivos usados por el sistema durante el arranque.
/dev	Almacena los controladores para el acceso a los dispositivos físicos del disco, ratón, tarjetas, *scanner*, etc.
/var	Contiene información variable, tanto generada por el propio sistema como por los usuarios.
/var/log	Se almacenan los registros detallados de toda la actividad desarrollada en el transcurso de una sesión de trabajo.
/lib	Contiene las librerías usadas por diferentes aplicaciones, evitando que cada programa incluya las suyas propias con la consiguiente redundancia de ficheros
/etc	Es el directorio destinado para almacenar todos los archivos de configuración
/home	Contiene el árbol de directorios propios de cada usuario del sistema.
/tmp	Es un directorio temporal usado generalmente por las aplicaciones para almacenar algunos ficheros en tiempo de ejecución.
/media	Cuando se monta un *CDROM*, una memoria *USB* o un disquete se crea aquí automáticamente un subdirectorio

2.3. *DHCP*

Protocolo de configuración dinámica de host (*DHCP*, *Dynamic Host* Configuration *Protocol*) es un protocolo que permite a los dispositivos de una red local obtener su

[33] *Tipos de particiones y sistema de archivos* [en línea]. Recuperado el 29 de Octubre de http://www.guia-ubuntu.org/index.php?title=Particionar_el_disco_duro
[34] *Idem*

propia información de configuración de red (dirección IP, máscara de sub-red, puerta de enlace, etc.), es decir, supervisa y distribuye las direcciones IP de una red de área local asignando una dirección IP dinámica a cada dispositivo que se une a la red[35]. Cuando un cliente es requiere obtener su dirección de red, solicita una dirección con el formato de una solicitud *DHCP*. Un servidor *DHCP* escucha las solicitudes de los clientes. Cuando se recibe una solicitud, revisa su base de datos local y emite la respuesta apropiada. La respuesta siempre incluye la dirección y puede incluir servidores de nombre, mascaras de red y una puerta de enlace predeterminada. El cliente recibe la respuesta del servidor y configura sus parámetros locales con los datos recibidos[36].

El servidor *DHCP* mantiene una lista de direcciones que puede emitir. Cada dirección se emite por un periodo de tiempo, durante el cual el cliente tiene autorización para utilizar la dirección asignada. Al término de dicho periodo de tiempo, es de esperarse que el cliente ya no utilice la dirección. En consecuencia, el servidor *DHCP* supone que la dirección vuelve a estar disponible y la regresa a su acervo de direcciones.

2.4. *Firewall*

Es un dispositivo de *hardware* o *software* sobre un sistema operativo, que aísla a una red de área local con la internet permitiendo filtrar el tráfico de red para decidir si un paquete pasa, se modifica, se descarta o se convierte. Generalmente está constituido por un conjunto de reglas en las que examina el origen y destino de los paquetes para decidir si la conexión puede establecerse o no[37].

[35] Kurose, J.F. y Ross, K.W.(2004). *Red de computadores: Un enfoque descendente basado en Internet.*México:Addison Wesley. p 331
[36] Shah, S. y Soyinka W. (2007). *Manual de administración de Linux.*México: Mc Graw Hill. p 574
[37] Kurose, J.F. y Ross, K.W.(2004). *Red de computadores: Un enfoque descendente basado en Internet.*México:Addison Wesley. p 635

Existen dos tipos de *firewalls*: *firewall* de filtrado de paquetes que opera en la capa de red y firewall de pasarelas de aplicación que operan en la capa aplicación.

Firewall de filtrado de paquetes: Utiliza un *router* para analizar las cabeceras de los paquetes y aplica un conjunto de reglas para si el paquete es rechazado o se deja pasar[38]. Las decisiones de filtrado se basan en:

- Dirección IP origen y destino
- Protocolo origen y destino
- Numero de puerto
- Contenido

Firewall de pasarelas de aplicación: Es un servidor de aplicación específico capaz de filtrar conexiones a servicios (*HTTP*, *Telnet*, correo, *FTP*, etc.), es decir, reenvía o bloquea las conexiones a servicios[39].

2.4.1 Aplicaciones de firewalls

Existen infinidad de *firewalls* en *GNU*/*Linux* algunos de los más importantes son los siguientes:

IPCOP: Es para pequeñas oficinas y usuarios domésticos, es una distribución *Linux* servidor de seguridad, que requiere una PC dedicada, la cual puede ser de baja potencia, para ejecutar el *software*. Se pueden configurar las reglas del *firewall* desde una interfaz web amigable, este *firewall* está basado en *Netfilter*.

[38] La guía definitiva para proteger de hackers a sus servidores Linux.(2000).*Linux máxima seguridad*. Madrid, España:Prentice Hall. p 520.
[39] Ibidem, p 521

Shorewall: Es una herramienta para la configuración de *firewalls*. Sólo necesita que se le proporcionen algunos datos en algunos ficheros de texto simple y éste creará las reglas de firewall correspondientes a través de *iptables*, por tal, está basado también en *Netfilter*.

UFW: Es un programa de línea de comandos que ayuda a manejar las *iptables*, provee algunos comandos sencillos para administrar las *iptables*. *Gufw* es una interface gráfica para el *UFW* es muy intuitivo y facilita en extremo el uso de *Netfilter*.

2.4.2. *Netfilter liptables*

Netfilter es un subsistema que se incluye en el *kernel* de *GNU/Linux* para interceptar y manipular paquetes de red, está compuesto por una serie de módulos y herramientas libres para construir firewalls. Para hacer que la configuración sea más fácil *Netfilter* suministra una herramienta llamada *iptables*[40].

La infraestructura de *Netfilter* está compuesta por tres tipos de operaciones: *Nat*, *Filter* y *Mangle*. Cada operación tiene su propia tabla. En cada tabla existe una serie de cadenas por las que pasa un paquete. Una cadena es sólo una lista reglas que actúan sobre un paquete que fluye por el sistema. En *Netfilter* existe 5 cadenas predefinidas: *PREROUTING*, *FORWARD*, *POSTROUTING*, *INPUT* Y *OUTPUT*[41]. En la tabla 2.3 se describen cada una de las tablas y la relación con las cadenas.

[40] Shah, S. y Soyinka W. (2007). *Manual de administración de Linux.* México: Mc Graw Hill. p 302
[41] *Ibidem*, p 306

Tabla 2.3 Correspondencia de tablas y cadenas de iptables

Tabla	Función de la tabla	Cadena	Función de la cadena
FILTER	Filtrado de paquetes	*INPUT*	Filtrado de paquetes que llegan al firewall o estación de trabajo.
		OUTPUT	Filtrado de paquetes que salen del propio equipo
		FORWARD	Permite o niega el paso de paquetes dirigidos a otro equipo que esté detrás del firewall
NAT	Enrutamiento de direcciones de red	*PREROUTING*	Los paquetes entrantes pasan a través de esta cadena antes de que se consulte la tabla de ruteo local, principalmente para *DNAT*
		POSROUTING	Los paquetes salientes pasan por esta cadena después de haberse tomado la decisión del ruteo, principalmente para *SNAT*
MANGLE	Modificación de paquetes	*PREROUTING* *POSTROUTING* *INPUT* *OUTPUT* *FORWARD*	Permite la modificación de los paquetes

Iptables es la herramienta responsable de proporcionar una interfaz para crear y administrar las reglas de *Netfilter*, es decir, en base a las reglas de configuración de *firewall* filtra paquetes de red para aceptarlos rechazarlos o modificarlos

Iptables tiene, a su vez, parámetros y comandos que permiten el comportamiento de una o varias reglas, es decir, agregar, modificar o eliminar una regla a una cadena. En la tabla 2.4 se muestran los modificadores más comunes.

Tabla 2.4 Modificadores más comunes de *iptables*

Comando	Descripción
-A	Agrega nueva regla a la cadena especificada.
-I	Inserta nueva regla antes de la regla numero_regla en la cadena especificada de acuerdo a los parámetros sometida.
-R	Reemplaza la regla numero_regla en la cadena especificada.
-E	Modifica el nombre de la cadena [nombre_anterior-cadena-nombre_nueva_cadena]
-L	Listado de las reglas de la cadena especificada. Si no se determina una cadena en particular, listará todas las cadenas existentes.
-N	Crea una nueva cadena asociada al nombre.
-P	Modifica la acción por defecto de la cadena preseleccionada.
-D	Eliminar la regla numero_regla en la cadena seleccionada.
-Z	Pone los contadores de paquetes y bytes a cero en la cadena seleccionada, de no seleccionar una cadena, pondrá a cero todos los contadores de todas las reglas en todas las cadenas.

Todas las reglas de *iptables* tienen definida su condición por los parámetros, que constituyen su parte primordial. En la tabla 2.5 se muestran los parámetros y su función de *iptables*

Tabla 2.5 Parámetros de *iptables*

Parametro	Descripción
-i	Interfaz de entrada (eth0, eth1, eth2…)
-o	Interfaz de salida (eth0, eth1, eth2…)
--sport	Puerto de origen.
--dport	Puerto destino.
-p	El protocolo del paquete a comprobar, tcp, udp, icmp ó all. Por defecto es all.
-j	Esto especifica el objetivo de la cadena de reglas, es decir; ejecuta una acción
--lines_numbers	Cuando listamos las reglas, agrega el número que ocupa cada regla dentro de la cadena.

Y finalmente las acciones que estarán siempre al final de cada regla que determinará que hacer con los paquetes afectados. Si no se especifica

ninguna acción, se ejecutará la opción por defecto que cada regla tiene asignada, las acciones de *iptables* se muestran en la tabla 2.6.

Tabla 2.6 Acciones de *iptables*

Acción	Descripción
ACCEPT	Paquete aceptado
REJECT	Paquete rechazado. Se envía notificación a través del protocolo *ICMP* a quien envió originalmente
DROP	Paquete rechazado. Sin notificación
MASCARADE	Enmascaramiento de la dirección IP origen de forma dinámica. Esta acción es sólo válida en la tabla *NAT* en la cadena de *POSTROUTING*
DNAT	Permite que la dirección (y opcionalmente el puerto) de destino del paquete sean reescritos para la *NAT*. Mediante la opción '—to-destination' debe indicarse el destino a usar.
SNAT	Permite que la dirección (y opcionalmente el puerto) de origen del paquete sean reescritos para la *NAT*. Mediante la opción '—to-destination' debe indicarse el origen a usar.

La estructura de una regla de *iptables* básicamente es como se muestra a continuación:

Iptables→-t→tabla→comando→cadena→regla con parámetros→acción

2.5. *Proxy*

Es un *software* o *hardware* que actúa como intermediario entre una red de área local y la internet, permitiendo a los clientes realizar conexiones a la internet a través de él. Cuando un usuario se conecta a la internet con una aplicación cliente, configurada para utilizar un servidor *proxy*, la aplicación primero se conectará con el servidor *proxy* y le dará la solicitud. El *proxy* se conectará entonces al servidor remoto, al que contiene la aplicación a la que el cliente desea conectarse y le envía la solicitud.

Después, el servidor remoto le envía la respuesta al servidor proxy, el cual a su vez la envía a la aplicación del cliente.[42]

Una aplicación muy común del servidor *proxy* es la *cache web*, que almacena la información que contienen las páginas que los usuarios de la red de área local visitan con mayor frecuencia, por un determinado periodo de tiempo, la finalidad del *cache web* es disminuir el uso de ancho de banda en la internet y aumentar la velocidad de acceso a los documentos de los usuarios[43].

Por otra parte, se pueden crear registros de actividad para guardar las peticiones de los usuarios cuando solicitan conexiones a la internet. Las conexiones de Internet pueden filtrarse para analizar tanto las solicitudes del cliente como las del servidor. El filtrado se realiza comparando la solicitud del cliente con una lista de solicitudes autorizadas o una lista de sitios prohibidos.

2.5.1. *Squid*

Es un *software* libre que implementa un servidor *proxy* y un demonio para el manejo eficiente de la memoria cache de páginas *web*, publicado bajo la licencia *GLP*[44]. Tiene una amplia variedad de utilidades:

- *Proxy* y *cache*: Proporciona un servidor *proxy* que soporta peticiones *HTTP*, *HTTPS*, *FTP* a equipos que necesitan acceder a internet y a su vez provee la funcionalidad de cache especializado en el cual almacena de forma local las páginas consultadas recientemente por los usuarios.

[42] Honeycutt,J, (1998). *La biblia de internet.* Madrid, España: Anaya Multimedia. p 279
[43] Kurose, J.F. y Ross, K.W.(2004). *Red de computadores: Un enfoque descendente basado en Internet.* México:Addison Wesley. p 152
[44] *Squid: Optimising web delivery.* [en linea]. Recuperado el 29 de Octubre de 2010 de http://www.squid-cache.org/

- Proxy para SSL: También es compatible con SSL (*Secure Socket Layer*) con lo que también acelera las tracciones cifradas y es capaz de configurarlo con amplios controles de acceso sobre las peticiones de los usuarios.

- Cache transparente. Puede ser configurado para ser uso como *proxy* transparente de manera que las conexiones son ruteadas dentro del *proxy* sin configuración por parte del cliente, y habitualmente sin que el propio cliente conozca de su existencia.

- *WCCP*: Permite interceptar y redirigir el trafico que recibe de un *router* hacia uno o más *proxys* cache, haciendo control de la conectividad de los mismo.

- Control de acceso. Ofrece la posibilidad de establecer reglas de control de acceso Esto permite establecer políticas de acceso en forma centralizada, simplificando la administración de la red

- Aceleración de servidores *HTTP*: Un usuario hace una petición hacia un objeto de internet, este es almacenado en el cache, si otro usuario hace petición hacia el mismo objeto, y este no ha sufrido modificación alguna desde que lo accedió el usuario anterior, *Squid* mostrará el que ya se encuentra en cache en lugar de volver a descargarlo desde internet.

El archivo principal de configuración de *squid* se llama squid.conf el cual contiene varias opciones de configuración:

- Opciones de autentificación
- Controles de acceso
- Opciones de reenvío

- Opciones SSL
- Opciones de red
- Opciones que afectan al algoritmo de selección de vecino
- Opciones de memoria *cache*
- Opciones de disco *cache*
- Opciones del archivo *log*
- Opciones para *FTP*
- Opciones para programas de apoyo externo
- Opciones para la reescritura de direcciones *URL*
- Opciones para el ajuste de *cache*
- Opciones *HTTP*
- Tiempos de espera
- Parámetros de administración
- Opciones para el servicio de registro de *cache*
- Opciones de acelerador *HTTPD*
- Retraso de parámetros
- Opciones de configuración *WCCPv1* y *WCCPv2*
- Manejo de conexiones persistentes
- Opciones de compendio de *cache*
- Opciones *SNMP*
- Opciones *ICP*
- Opciones *ICP* multicast
- Opciones internas ICONO
- Opciones de errores de página
- Opciones que influyen en envió de solicitud
- Opciones avanzadas de red
- Opciones *DNS*
- Opciones misceláneas

2.5.2. *SquidGuard*

Es un complemento de la licencia libre (GPL) para *Squid* que funciona como filtro, redireccionador y controlador de acceso flexible y rápido. Permite definir varias reglas de acceso con diferentes restricciones para diferentes grupos de usuarios de una cache de *Squid*. Squ*i*dGuard utiliza la interfaz de redirección estándar de *Squid* y una base de datos con millones direcciones web clasificadas en grupos pornografía, violencia, publicidad, etc.[45] Permite hacer lo siguiente:

- Limitar el acceso web para algunos usuarios a una lista de servidores web o direcciones *URL* aceptados o conocidos
- Bloquear el acceso a algunos servidores web o direcciones *URL* de una lista para algunos usuarios
- Bloquear el acceso a las direcciones *URL* que coincidan con una lista de expresiones regulares o palabras para algunos usuarios
- Redirigir las direcciones *URL* a una página de información inteligente basada en *CGI*
- Redirigir a los usuarios no registrados a un formulario de registro
- Redirigir los anuncios a un *GIF* vacio
- Utilizar diferentes reglas de acceso basadas en la hora del día, el día de la semana, la fecha, etc.
- Utilizar reglas diferentes para distintos grupos de usuarios.

[45] *SquidGuard* [en línea]. Recuperado el 29 de Octubre de 2010 de http://www.squidguard.org/

El archivo principal de configuración de *squidGuard* se llama squidGuard.conf

2.5.3 *Sarg*

Sarg (*Sq*uid *Analysis Report Generator*) es una herramienta de análisis de *logs* de Squid, genera reportes en *HTML*, con campos como: usuarios, direcciones *IP*, bytes transmitidos, sitios web y tiempos, permitiendo ver con detalle la actividad de todos los equipos y/o usuarios dentro de una red de área local[46].

Tiene soporte para generar reportes en diferentes idiomas, mediante los reporte de uso web se puede obtener la siguiente información:

- *Topten* de sitios más visitados
- Reportes diarios, semanales y mensuales
- Gráficas semanales y mensuales del consumo por usuario/host
- Detalles de todos los sitios a los que entro un usuario/host
- Descargas

Sarg puede ser configurado para generar reportes web de los accesos a internet de forma periódica, además de poder ejecutarlo manualmente para generar reportes de fechas usuarios o dominios en específico. El principal archivo de configuración de *sarg* se llama *sarg.conf*.

[46] *Squid Analysis Report Generator.* [en línea]. Recuperado el 29 de Octubre de 2010 de http://sarg.sourceforge.net/sarg.php

2.6 Apache

Es un servidor web de *HTTP* de código abierto para plataformas *UNIX*(*BSC*, *GNU*/*Linux*,etc), *Microsoft Windows*, *Macintosh* y otras. Un servidor web es programa que permite acceder a páginas web alojadas en una computadora. El protocolo más utilizado para ver páginas web es el *HTTP* (*Hyper Text Transfer Protocol*)[47].

El principal archivo de configuración de apache se llama apache2.conf, el cual tiene diferentes opciones de configuración:

- Directiva *VirtualHost*: Contiene una configuración predeterminada preparada para servidores virtuales, se puede modificar o dejarlo tal cual, si sólo se tiene un único sitio web, o usarlo como plantilla para servidores virtuales si se tienen varios sitios web

- Directiva *ServerAdmin*: Especifica la dirección de correo del administrador del servidor. El valor por default es webmaster@localhost.

- Directiva *Listen*: Especifica el puerto y ocasionalmente la dirección IP por la que escucha Apache2. El valor por default de la directiva *Listen* es 80.

- Directiva *ServerName*: es opcional y especifica con cual *FQDN*(Full *Qualified Domain Name*, Nombre de Dominio Totalmente Cualificado) responderá al sitio *web*. El servidor virtual predeterminado no especifica ninguna directiva *ServerName*.

[47] Shah, S. y Soyinka W. (2007). *Manual de administración de Linux.*México: Mc Graw Hill. p 420

- Directiva *DocumentRoot*: Especifica donde debe buscar Apache los archivos que conforman el sitio. El valor predeterminado es /var/www.

- Directiva *DirectIndex*: Es la página servida por default por el servidor cuando un usuario solicite el índice de un directorio añadiendo la barra de división (/) al final del nombre del directorio.

- Directiva *ErrorDocument*: Permite especificar un archivo que usará Apache2 para los eventos de error específicos. Por ejemplo, si un usuario solicita un recurso que no existe, se producirá un *error 404* que mostrará el archivo de configuración predeterminado.

- De forma predeterminada, el servidor escribe los registros de las transferencias en el archivo access.log.

CAPÍTULO III

DESARROLLO

3.1. Instalación del sistema operativo Ubuntu server 10.04

El sistema operativo *Ubuntu server* 10.04 se descarga de la siguiente página *http://www.ubuntu.com/server/get-ubuntu/download* como se muestra en la figura 3.1.

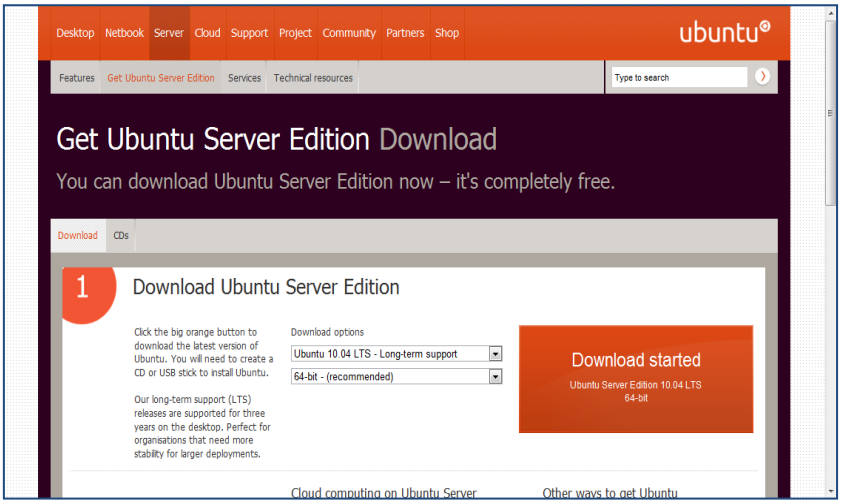

Figura 3.1. Página para la descarga de *Ubuntu server* 10.04

En la figura 3.2 muestra que el archivo de descarga, es una imagen *ISO*, por lo que es necesario guardar y *quemar* en un *CD*.

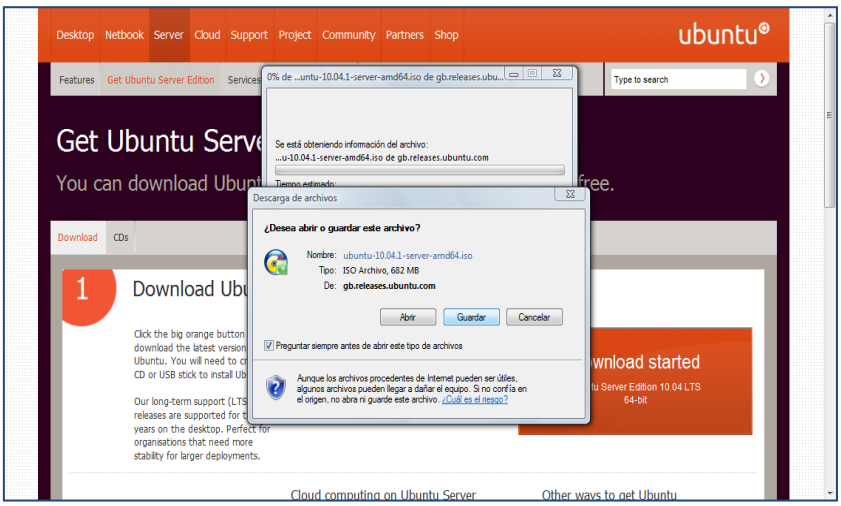

Figura 3.2. Descarga de la imagen *ISO*

Se accede al *BIOS* (Sistema Básico de Entrada/Salida) del servidor para configurarlo de manera que arranque desde la unidad lectora de *CD-ROM*, como se ilustra en la figura 3.3.

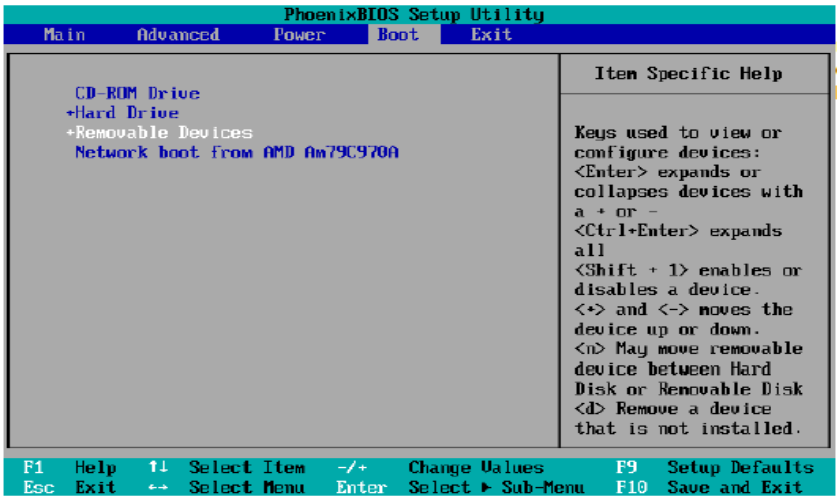

Figura 3.3 Configuración de la *BIOS* para arrancar desde el *CD-ROM*

Se guardan los cambios y se sale de la *BIOS* como se ilustra den la figura 3.4

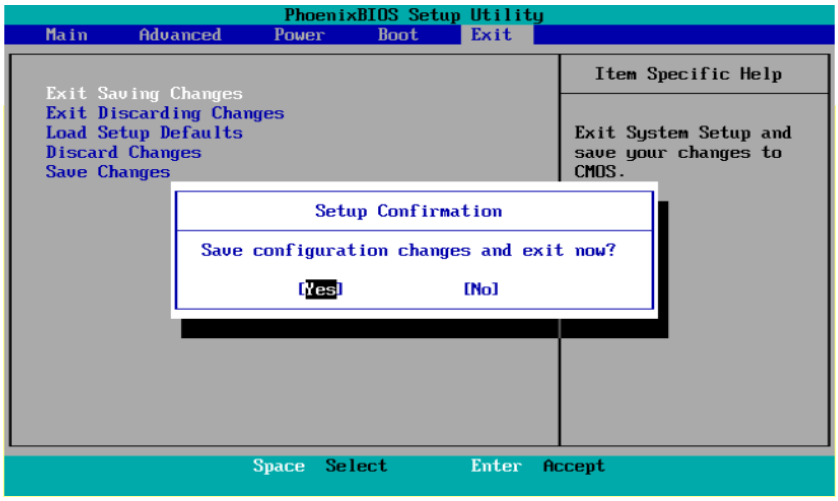

Figura 3.4 Guardar cambios en la *BIOS* y salir

Al iniciar la instalación se muestra la pantalla para seleccionar el lenguaje de instalación, se selecciona el lenguaje en español como muestra en la figura 3.4

Figura 3.5 Selección del idioma de instalación

A continuación aparece la pantalla de instalación *Ubuntu* con una serie de opciones en donde se selecciona la opción de Instalar *Ubuntu Server* como se muestra en la figura 3.6.

Figura 3.6 Iniciar la instalación de *Ubuntu Server* 10.04

Se inicia la configuración regional, donde se selecciona México como se muestra en la figura 3.7

Figura 3.7 Configuración regional

Se realiza la configuración del teclado, para probar que el modelo del teclado sea detectado se selecciona la opción Si y se presiona *enter* como se muestra en la figura 3.8

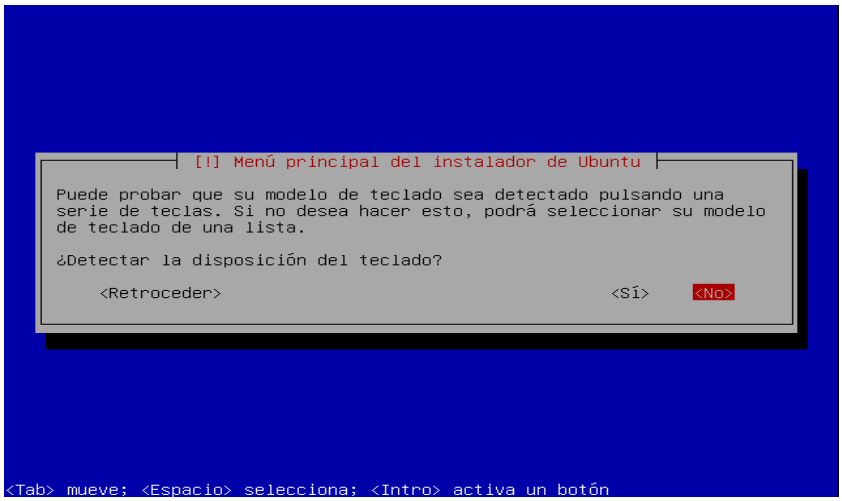

Figura 3.8 Configuración del teclado

Se inicia la prueba del teclado pulsando cada tecla que aparece en pantalla como se muestra en la figura 3.9

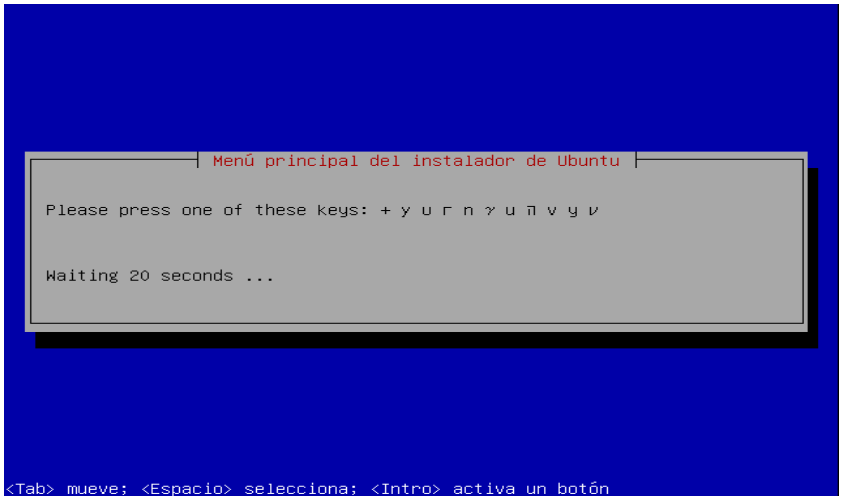

Figura 3.9 Prueba del teclado

Si se está de acuerdo con el modelo del teclado según las teclas pulsadas presionar *enter* para continuar .como se muestra en la figura 3.10

Figura 3.10 Modelo del teclado

Se configuran las interfaces de red, el sistema detectará dos interfaces de red: eth0 y eth1, en donde eth0 se utilizará como interfaz de red primaria ya que estará conectada a la internet, y eth1 se utilizará para la red de área local. Por lo tanto se presiona *enter* en la interfaz eth0 como se muestra en la figura 3.11.

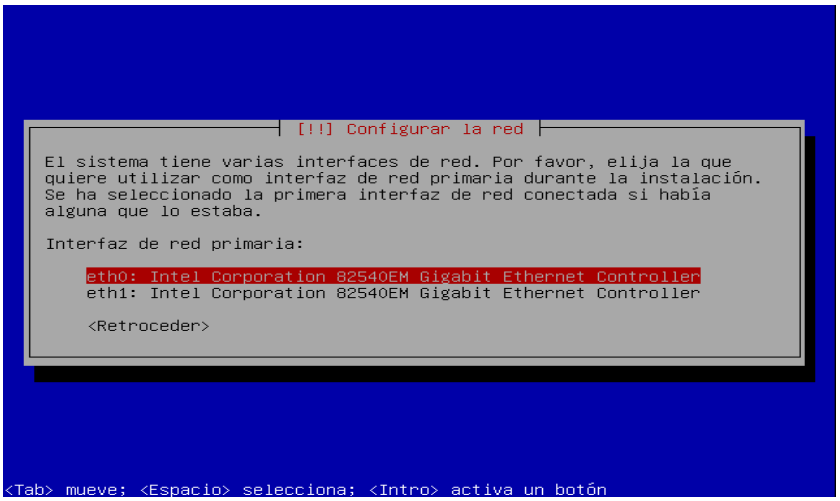

Figura 3.11 Configuración de interfaces de red.

En la figura 3.12 se muestra la barra del proceso de detección del hardware.

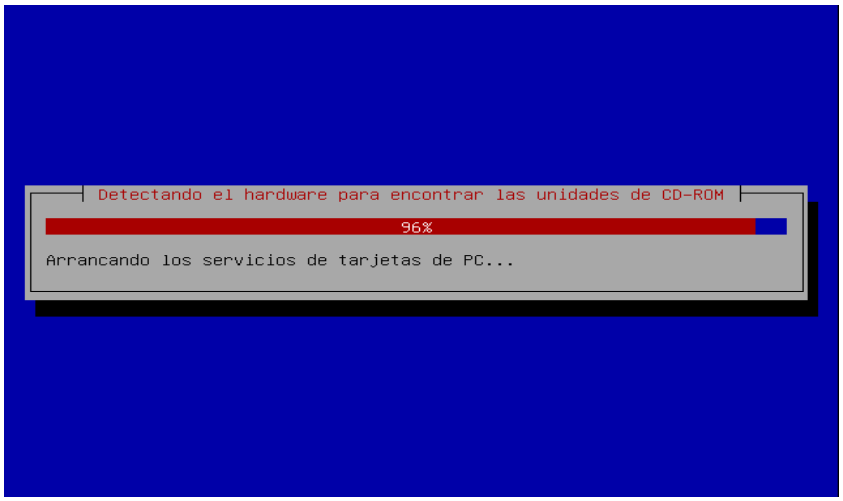

Figura 3.12 Barra del proceso de detección de hardware

En la siguiente pantalla de la instalación se introduce el nombre que tendrá el servidor para este proyecto se seleccionó filtrado tal y como se muestra en la figura 3.13.

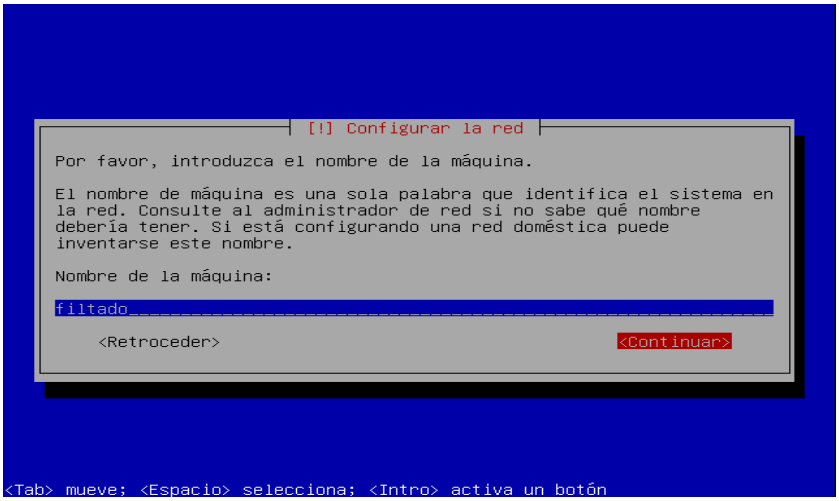

Figura 3.13. Nombre del servidor

Configuración de la zona horaria, si se está de acuerdo con la zona horaria que aparece en la pantalla seleccionar si y presionar *enter* como se muestra en la figura 3.14.

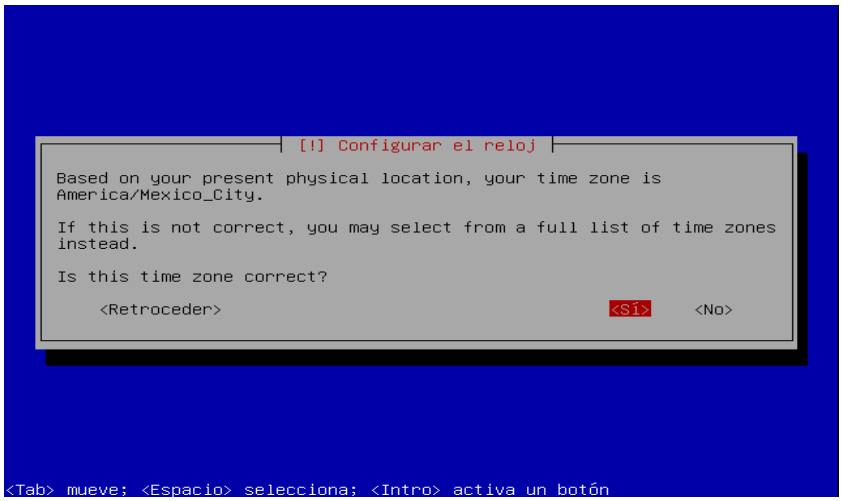

Figura 3.14 Configuración de la zona horaria

En la pantalla de particionado de discos seleccionar todo el disco y presionar *enter* como se observa en la figura 3.15

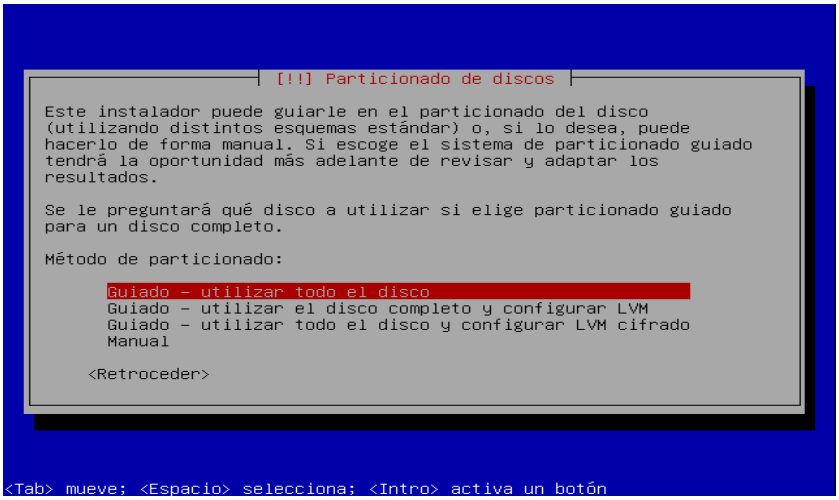

Figura 3.15 Particionado de discos.

En la siguiente pantalla seleccionamos el disco en donde se quiere instalar el sistema operativo Ubuntu Server 10.04 y presionar *enter*, en caso de que el sistema detecte más discos todos parecerán aquí como se muestra en la figura 3.16.

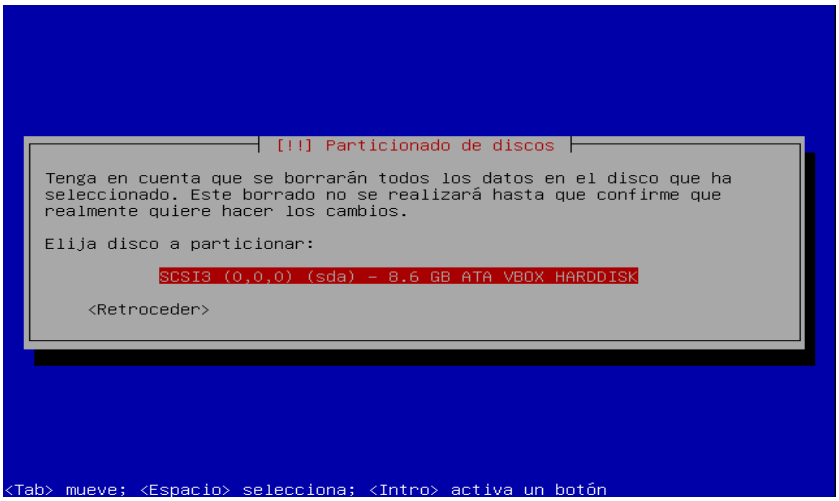

Figura 3.16 Seleccionar disco duro

En la siguiente pantalla seleccionar si y presionar *enter* para formatear el disco duro antes seleccionado tal y como se muestra en la figura 3.17

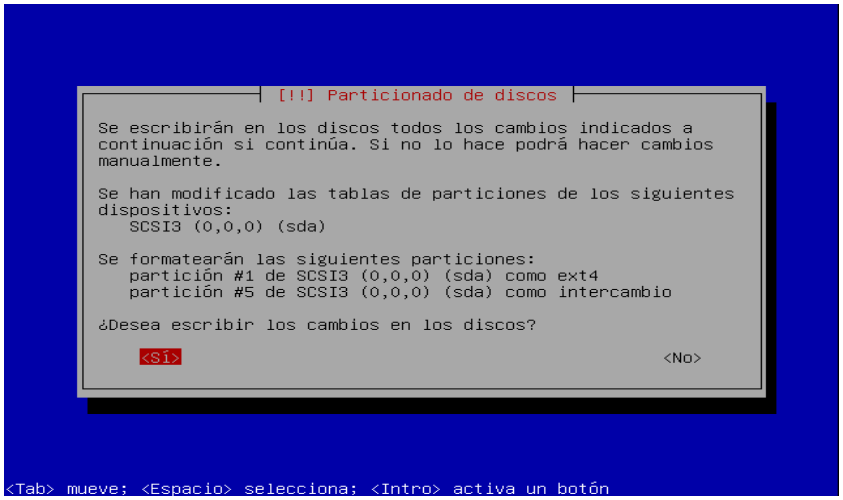

Figura 3.17 Formateo del disco

En la En la figura 3.18 aparece el progreso de la instalación del sistema base del sistema operativo Ubuntu server 10.04.

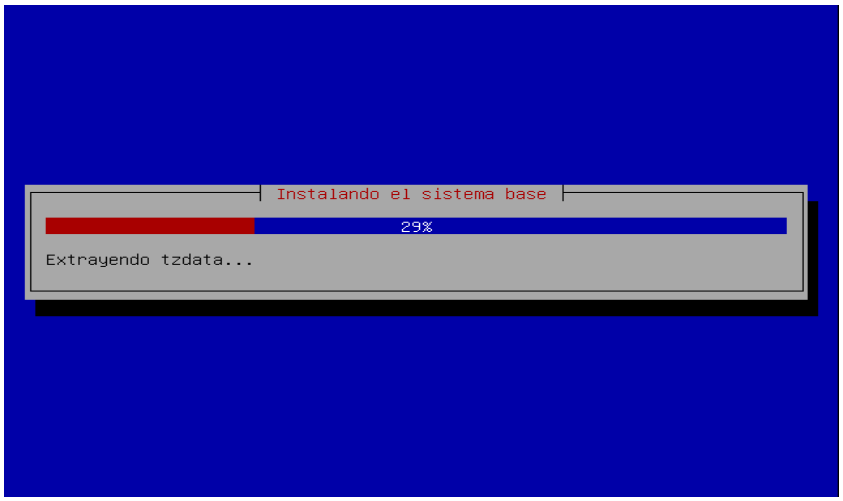

Figura 3.18 Barra de proceso de instalación del sistema base

Se crea la cuenta de usuario se introduce el nombre completo del usuario y presionar *enter* en la opción continuar como se muestra en la figura 3.19

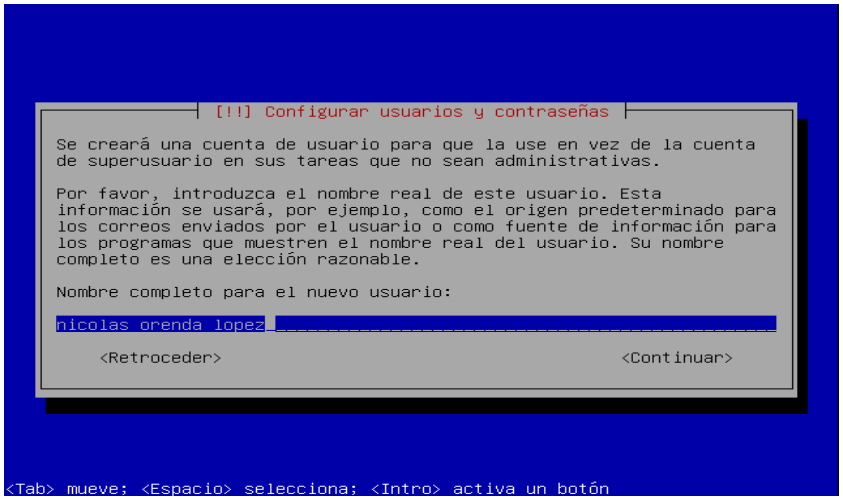

Figura .319 Nombre completo del usuario

A continuación aparece automáticamente el nombre del usuario que se introdujo anteriormente, si se esta de acuerdo con el nombre de la cuenta que se creará presionar *enter* para continuar como se muestra en la figura 3.20

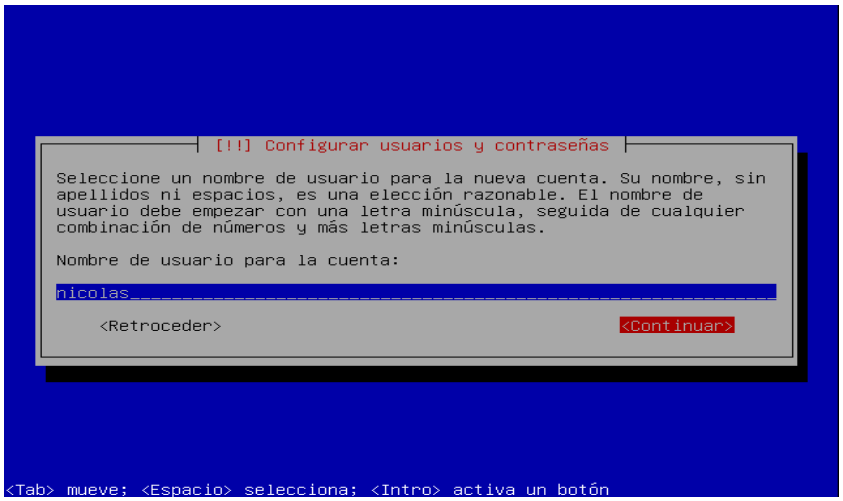

Figura 3.20 Creación de cuenta

Se introduce la contraseña para la cuenta antes creada teclear la contraseña y presionar *enter* para continuar, aparece otra pantalla para confirmar la contraseña como ilustra en la figura 3.21

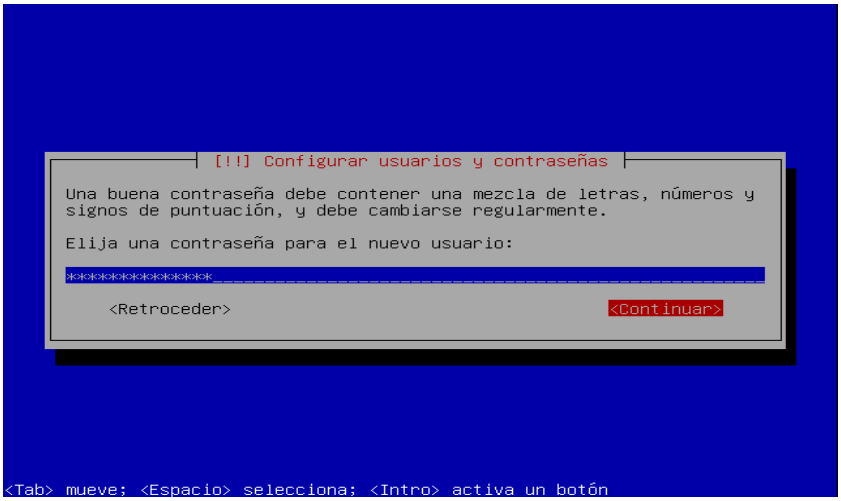

Figura 3.21 Contraseña para el usuario

En la opción de configuración de carpeta personal del usuario seleccionar no y presionar *enter* como se muestra en la figura 3.22.

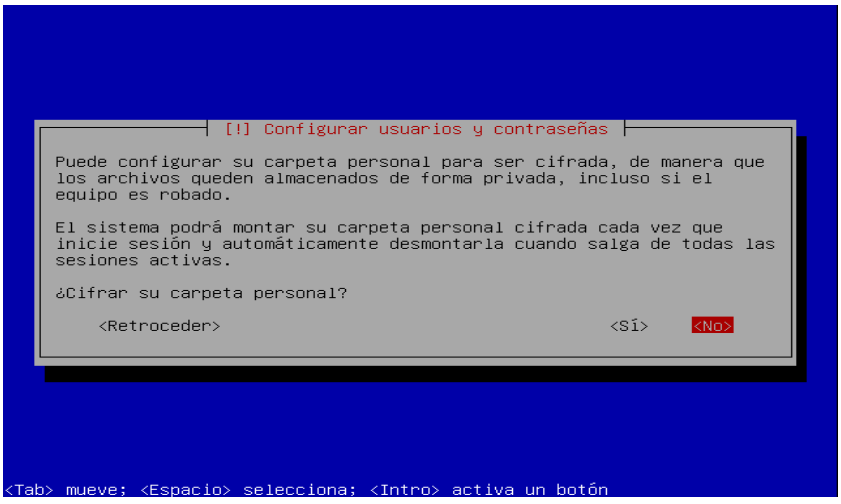

Figura 3.22 Cifrado de la carpeta personal

En la siguiente pantalla de la instalación seleccionar continuar y presionar *enter*

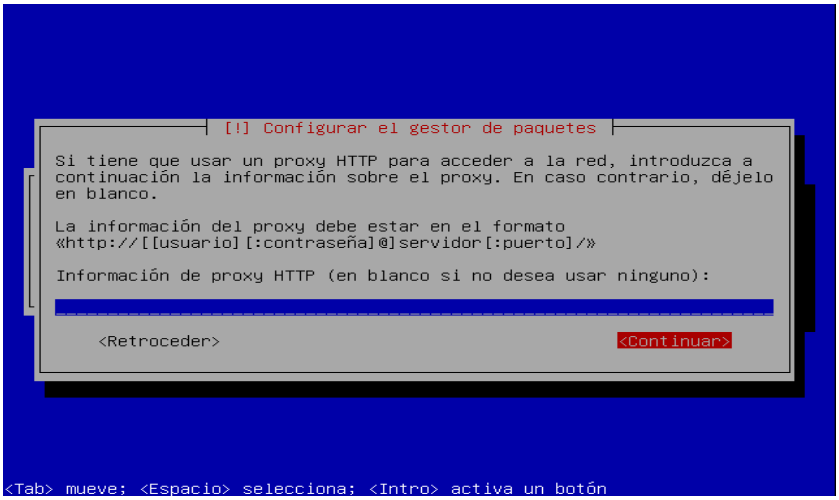

Figura 3.23 Configuración de gestor de paquetes

A Continuación aparece la barra del proceso de la instalación tal y como se muestra en la figura 3.24.

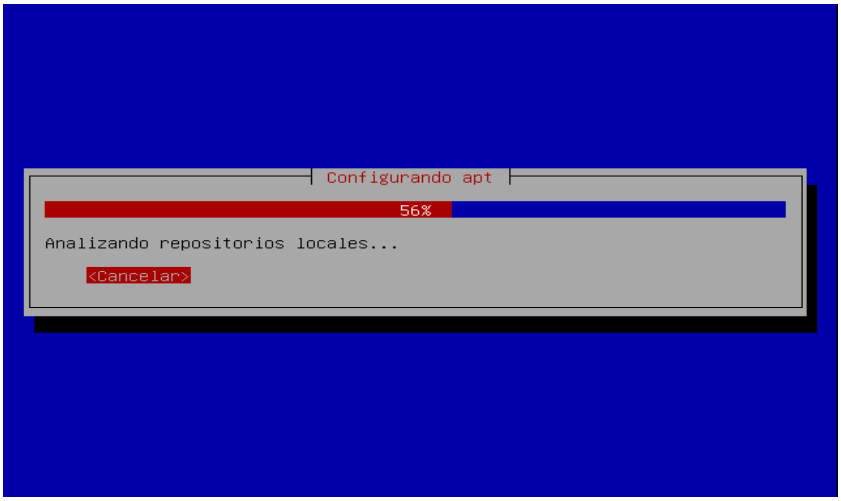

Figura 3.24 Barra de proceso de instalación.

En la siguiente pantalla seleccionar la segunda opción para instalar las actualizaciones automáticas y presionar *enter* como se muestra en la figura 3.25

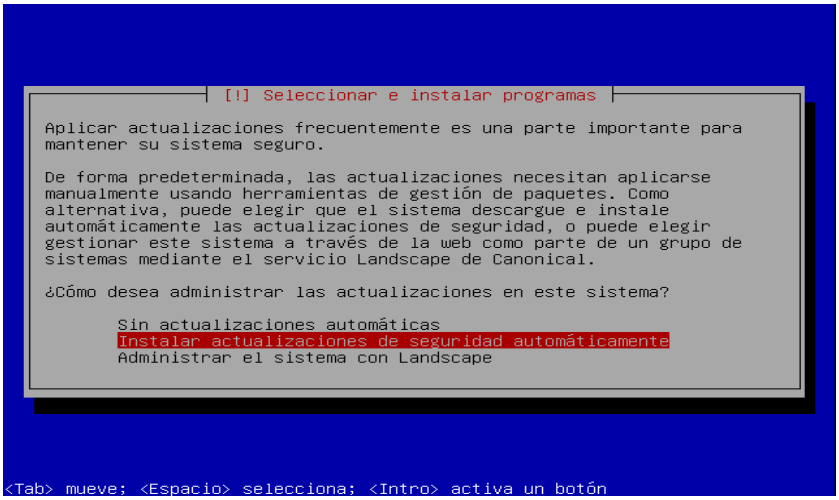

Figura 3.25 Actualizaciones automaticas

Si se requiere instalar algunos programas adicionales al sistema básico seleccionar alguno de la lista con la barra espaciadora, para este proyecto no seleccionar nada y presionar *enter* para continuar como se muestra en la figura 3.26.

Figura 3.26 Programas adicionales

En la figura 3.27 muestra la barra de proceso para la instalación de los programas básicos y los programas que fueron seleccionados

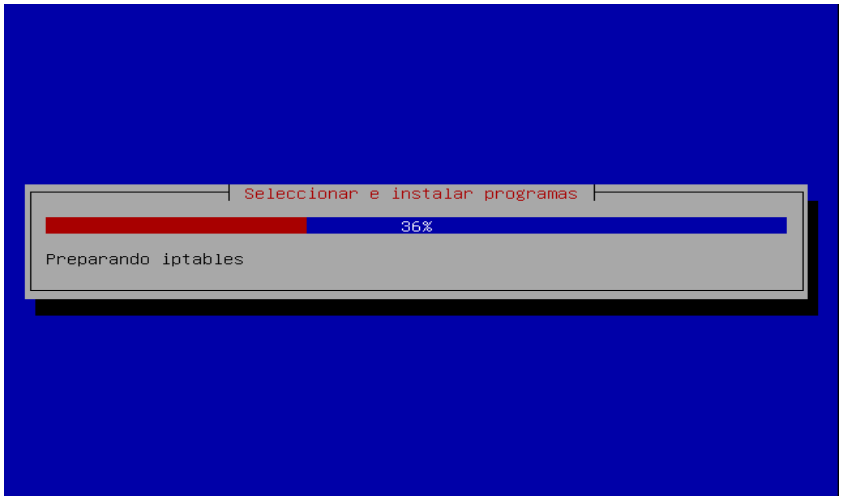

Figura 3.27 Barra de proceso de instalación de programas básicos

Configuración de *GRUB* (*Grand Unifier Bootloader*) gestor de arranque que permite tener diferentes sistemas operativos, presionar *enter* en la opción si para su configuración como se muestra en la figura 3.28.

Figura 3.28 Configuración GRUB

Por último aparece la pantalla que indica el final de la instalación, la unidad de CD-ROM expulsará el CD para que sea retirado y reinicie el servidor leyendo el disco

duro que contiene el sistema operativo Ubuntu Server 10.04, presionar *enter* en continuar como se muestra en la figura 3.29. A continuación se reiniciará el equipo

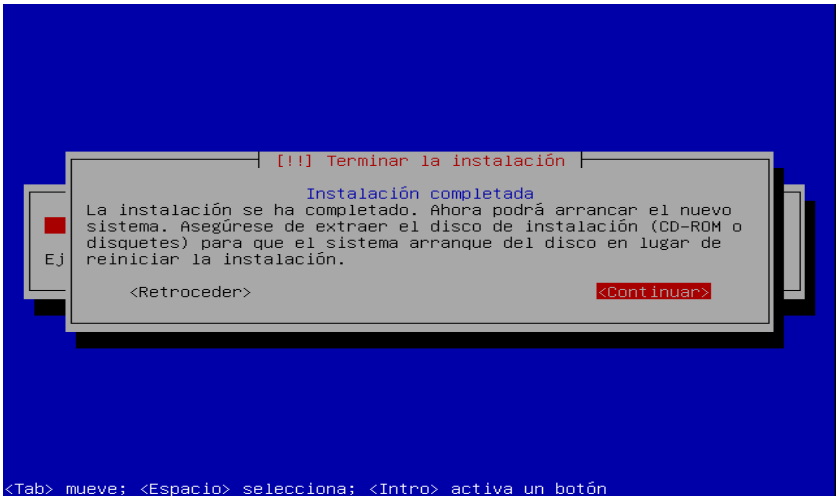

Figura 3.29 Instalación completa

Se muestra la pantalla de inicio de sesión del sistema operativo Ubuntu server 10.04 donde se introduce el usuario y la contraseña, las cuales se configuraron en los pasos anteriores como se muestra en la figura 3.30

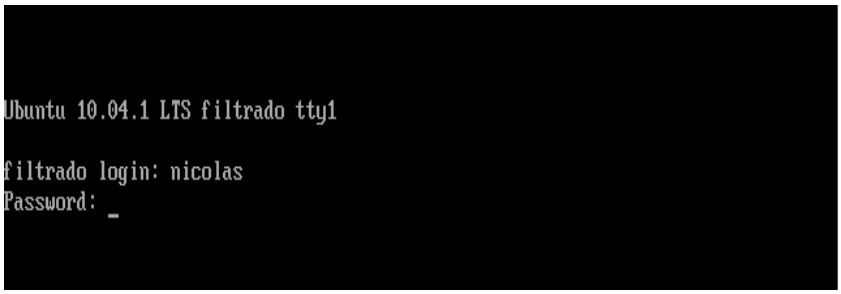

Figura 3.30 Pantalla para entrar al sistema operativo

A continuación en la pantalla de bienvenida se muestra la información del sistema como se muestra en la figura 3.31

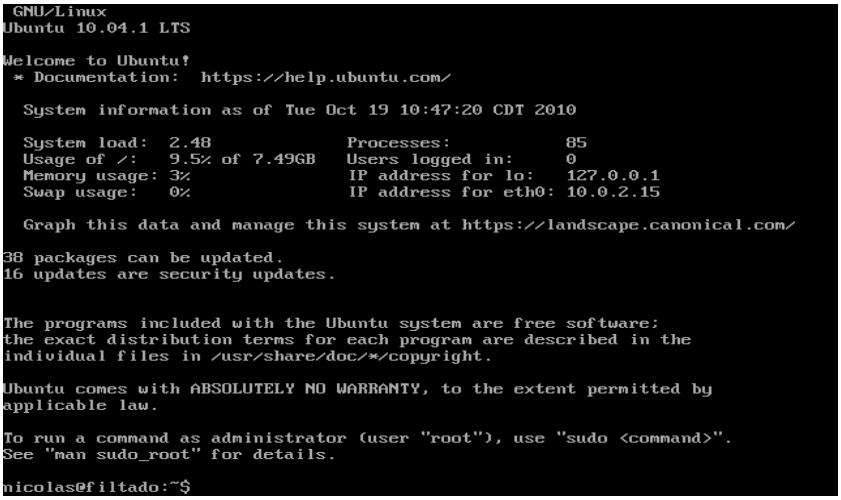

Figura 3.31 Información del sistema

3.2 Cambio de usuario *root* y configuración de interfaz de red

Antes de realizar cualquier instalación se debe verificar que se encuentra en modo superusuario, es decir, como usuario *root*, esto se puede ver si en la línea de comandos aparece el símbolo #, se hace para tener todos los permisos de usuario, ya que si se hacen las instalaciones como usuario normal símbolo $, se tendrá que ante poner en cada línea de comando el comando sudo e ingresar la contraseña cada vez que se requiera ejecutar algún programa o una instalación.

Para ingresar a modo *root* se hace con el comando *su passwd* y se ingresa una nueva contraseña para el usuario *root*, el sistema pide la confirmación de dicha contraseña. Una vez hecho esto se cambia a usuario *root* desde la línea de comandos y escribir el comando *su* y se ingresa la contraseña de *root*, si la contraseña es correcta el *prompt* cambiará al símbolo # .

Se recomienda realizar las instalaciones desde el la raíz como se muestra en la figura 3.32.

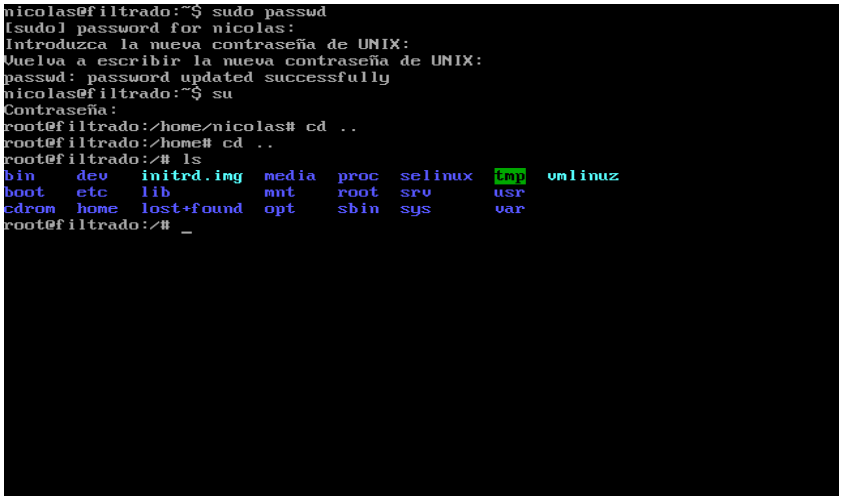

Figura 3.32 Cambio de usuario

Para verificar que las interfaces de red estén dadas de alta, se hace con el comando *ifconfig* como se muestra a continuación en la figura 3.33.

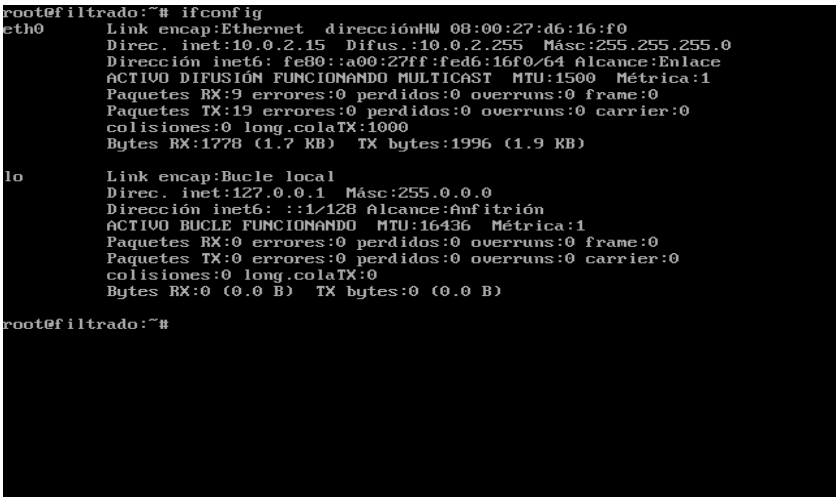

Figura 3.33 Interfaces de red

Para este proyecto se requieren dos interfaces de red eth0 que estará conectada a la internet y eth1 que estará conecta a las tres subredes: la dirección, sala de profesores y la aula de cómputo. Como se puede observar en la figura 3.33 la interfaz de red eth1 no aparece, para configurar la interfaz de red eth1 que estará

conectada para la subred de la dirección se hace con el comando *ifconfig eth1 192.168.1.1/24* tal y como se muestra en la figura 3.34.

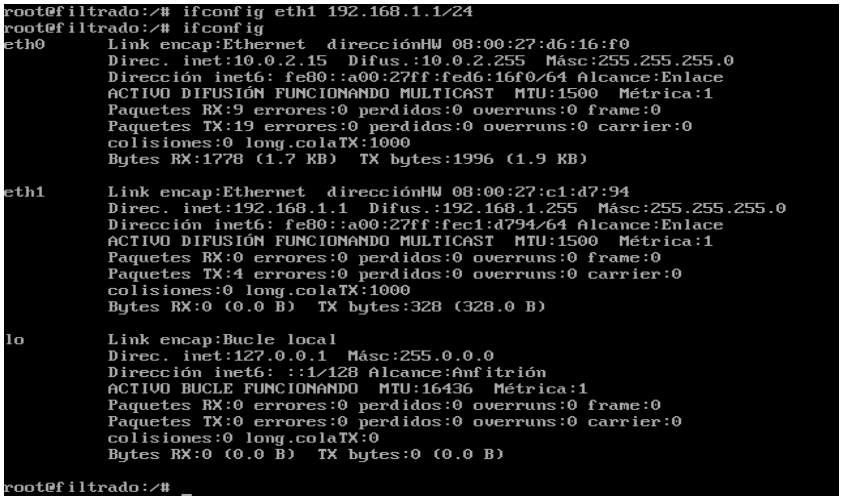

Figura 3.34 Configuración de eth1

Para las otras dos subredes: la sala de profesores y el aula de computo, se requiere de una alias para cada una de las subredes antes mencionadas, de la siguiente manera para la sub red de la sala de profesores es *ifconfig eth1:2 192.168.2.1/24* y para la sub red del aula de computo es *ifconfig eth1:3 192.168.3.1/24* como se muestra en la figura 3.35

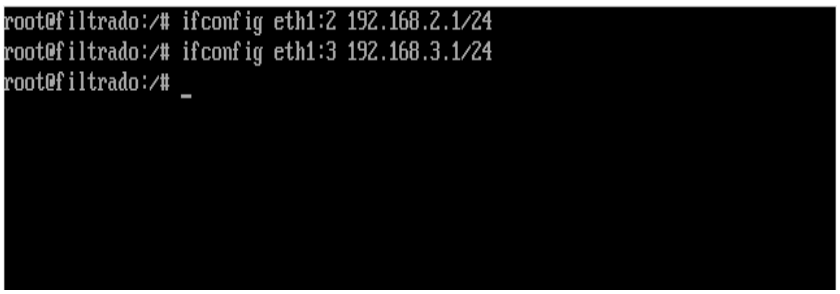

Figura 3.34 Configuraciones de red con alias

Se verifica la configuración de las interfaces de red en la línea de comandos con *ifconfig* como se muestra en la figura 3.35

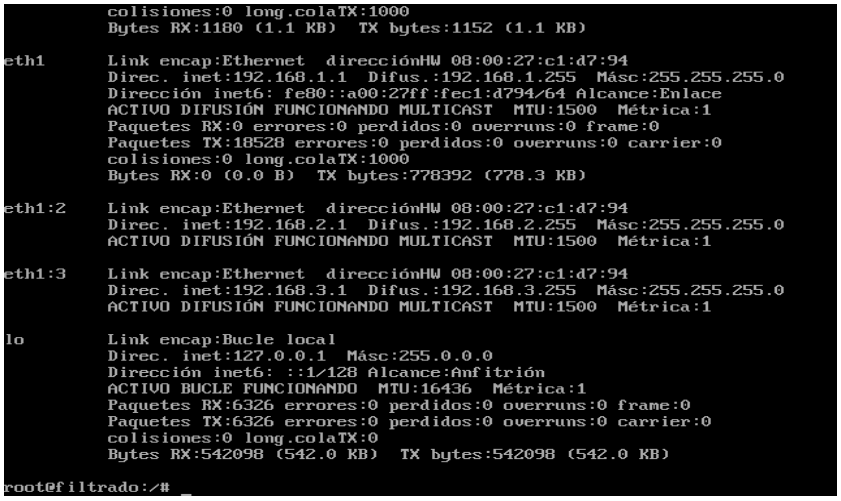

Figura 3.36 Verificación de las interfaces de red configuradas

Las configuraciones antes hechas para la interfaz de red eth1 se perderán si el servidor se reinicia. Para no perder estas configuraciones se edita el archivo *interfaces* que se encuentra en la siguiente dirección */etc/network,* acceder al archivo por medio de algún editor de texto por ejemplo vi como se muestra en la figura 3.37

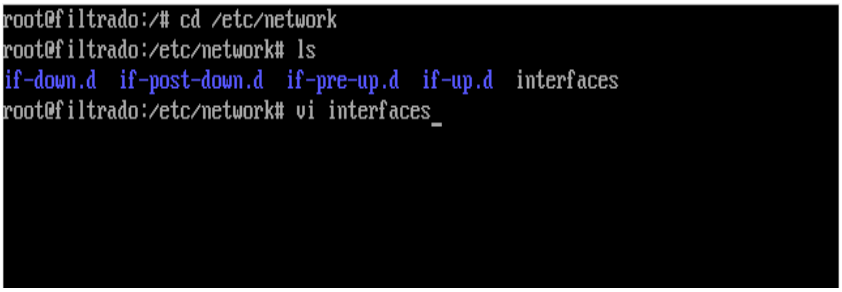

Figura 3.37 Acceder al archivo interfaces

En el archivo *interfaces* agregar las siguientes líneas para que automáticamente al iniciar el servidor se carguen la interfaz de red eth1 junto con sus alias para que no se pierda la configuración. Para editar el archivo presionar la tecla <*i*>, después de que se agregan las líneas de las configuraciones de las interfaces de red presionar <esc>y escribir: x para guardar los cambios y salir del archivo como se muestra en la figura 3.38.

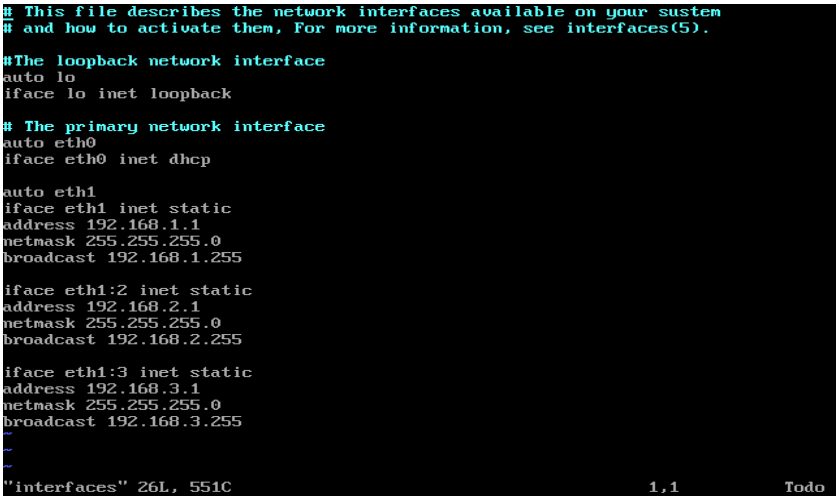

Figura 3.38 Configuración del archivo interfaces

Una vez hecho lo anterior, al reiniciar el servidor se mostraran los cambios realizados para cada una de las subredes: dirección, sala de profesores y aula de computo tal y como se aprecia en la figura 3.39.

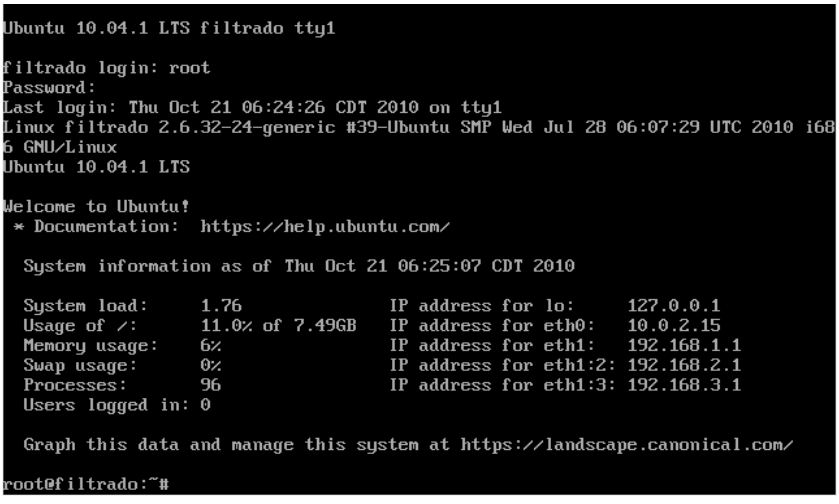

Figura 3.39 Pantalla de inicio con la interfaz de red configurada

3.3 Instalación del servidor *DHCP*

Para instalar el servidor *DHCP,* situarse el directorio raíz y escribir el comando *apt-get install dhcp3-sever* como se muestra en la figura 3.40

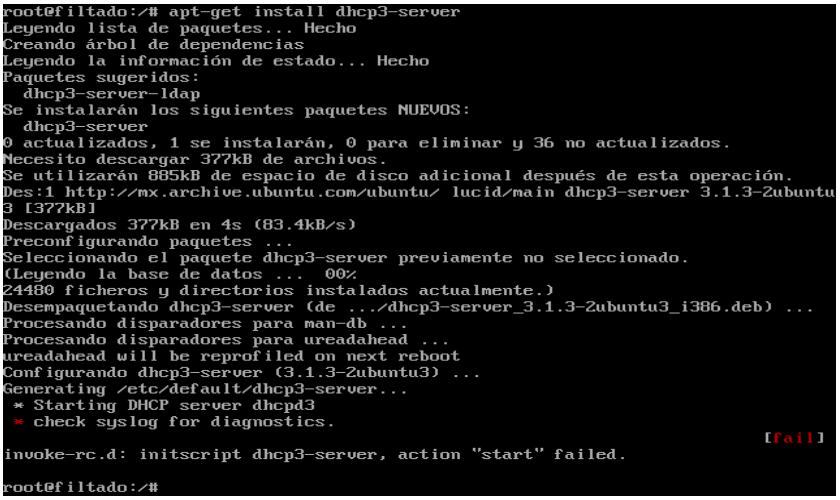

Figura 3.40 Instalación del servidor *DHCP*

3.3.1 Configuración del archivo *dhcpd*.conf

Para acceder al archivo de configuración *dhcpd.conf* situarse en la dirección */etc/dhcp3*. Acceder al archivo por medio de editor de texto vi como se muestra en la figura 3.41.

Figura 3.41 Acceder al archivo de configuración dhcpd.conf

En el archivo de configuración se agregan las siguientes líneas para que el servidor asigne el rango las direcciones IP a cada una de las subredes que se están estableciendo, como se muestra en la figura 3.42

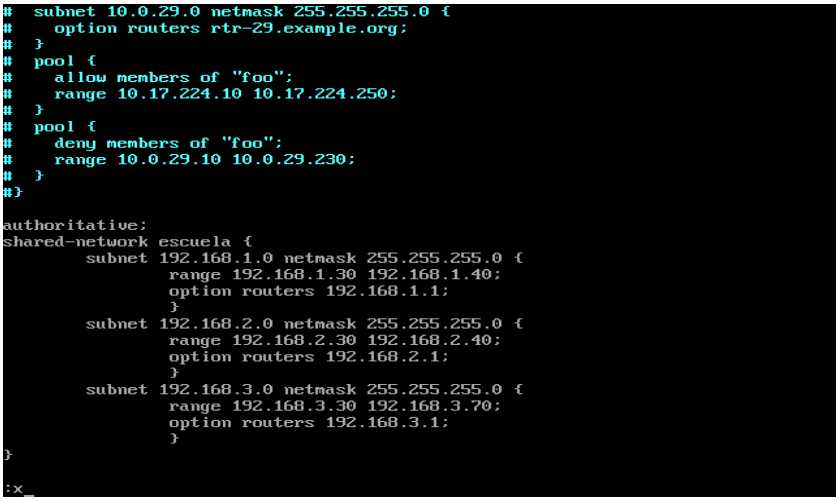

Figura 3.42 Archivo de configuración dhcpd.conf

Al terminar de configurar el archivo guardar los cambios y salir. Para confirmar que el servidor *DHCP* está funcionando correctamente se hace en dentro del directorio */etc/init.d* y escribir el siguiente comando *service dhcp3-server start* al ejecutar este comando y si está bien configurado mostrara un OK como se muestra en figura 3.43. En caso de que falle el servidor *DCHP* se tendrá que revisar las sintaxis que se realizó en el archivo de configuración dhcpd.conf.

Figura 3.43 Funcionando el servidor *DHCP*

3.4 Instalación de *squid*

Para instalar el servidor *proxy* de *Squid* situarse en el directorio raíz y escribir el siguiente comando *apt-get install squid* como se muestra en la figura 3.44.

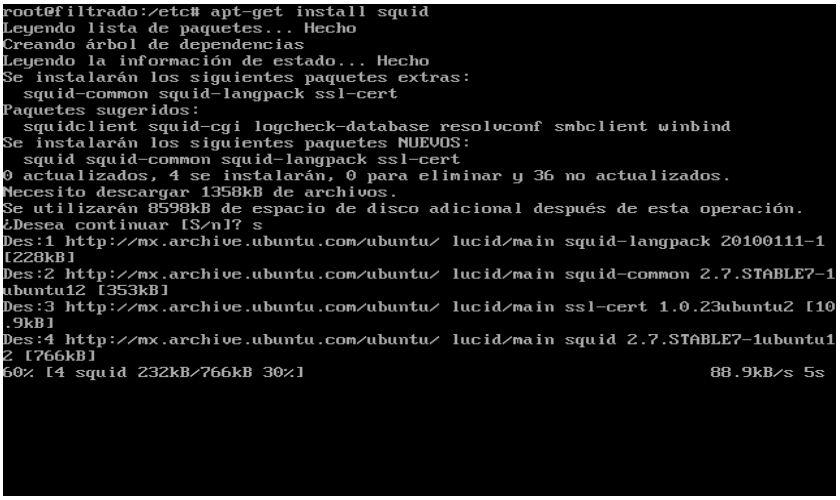

Figura 3.44 Instalación de Squid

3.4.1 Configuración del archivo *squid.conf*

Para acceder al archivo de configuración *squid*.conf situarse en la dirección */etc/squid.* Acceder al archivo por medio del editor de texto vi como se muestra en la figura 3.45.

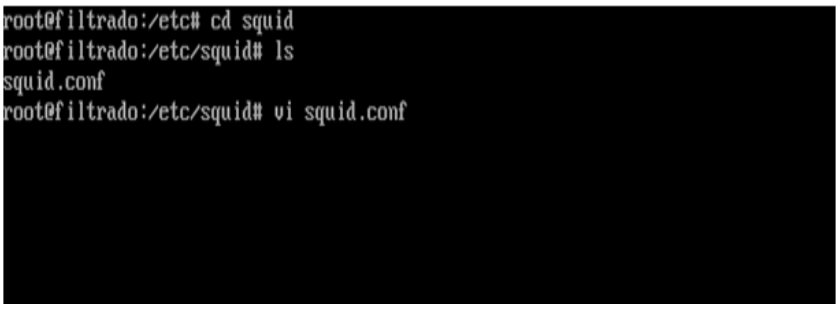

Figura 3.45 Acceder al archivo de configuración squid.conf

En el archivo de configuración squid.conf se configurará las *ACL* (listas de acceso) permitiendo solo dar acceso a las subredes que se tienen en este proyecto y denegando todo lo demás como se muestra en la figura 3.46.

```
#Recommended minimum configuration:
acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32
acl direccion src 192.168.1.0/24
acl salaprofesores src 192.168.2.0/24
acl aulacomputo src 192.168.3.0/24

#http_access allow localnet
http_access allow localhost
http_access allow direccion
http_access allow salaprofesores
http_access allow aulacomputo

# And finally deny all other access to this proxy
http_access deny all
```

Figura 3.46 Configuración de la acl

Otro punto que se tiene que configurar es la sección del puerto por el que escucha *Squid,* que por default es el puerto 3128, en esta línea se agrega la palabra *transparent,* esto se hace con la finalidad de evitar configurar cada uno de los dispositivos en su navegador, indicando que tiene que hacer uso de proxy. Por otra parte se redirecciona *SquidGuard* hacia el archivo de configuración *squidguard.conf*, para la consulta de los archivos en formato de base de datos que contienen los dominios y url´s de la páginas web que serán prohibidas mas adelante, como se muestra en la figura 3.47.

```
#



# Squid normally listens to port 3128
http_port 3128 transparent
redirect_program /usr/biin/squidGuard -c /etc/squid/squidGuard.conf
```

Figura 3.47 configuración de puerto de squid

Otra sección que se configura es la memoria cache la cual se dejará con 256 MB, como se muestra en la figura 3.48.

Figura 3.48 Configuración de la memoria cache

Por otra parte en la sección del disco cache se dejará con 1024 MB con 16 niveles y 256 subniveles cada uno, esto para tener un mayor almacenamiento de páginas web, como se muestra en la figura 3.49. Estas configuraciones se basan en las características del servidor para este proyecto.

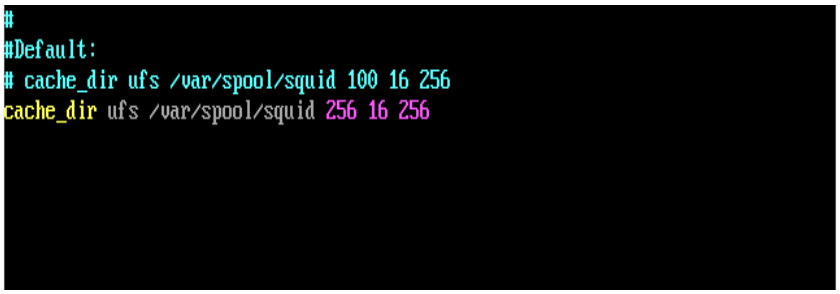

Figura 3.49 Configuración del disco cache

Al terminar de configurar el archivo guardar los cambios y salir. Para hacer funcionar al servidor proxy de Squid se hace dentro del directorio */etc/init.d* con el comando *service squid start*, antes de hacerlo funcionar se debe de detener el servicio con el siguiente comando *service squid stop* como se muestra en figura 3.50. En caso de que falle al iniciar el servicio de squid se tendrá que revisar las sintaxis que se realizó en el archivo de configuración squid.conf.

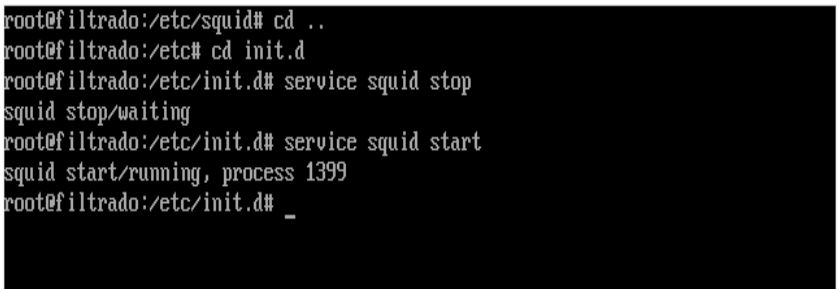

Figura 3.50 Funcionando Squid

3.5 Implementación de reglas de *firewall* y bit de reenvío

Para hacer que el servidor sea transparente no basta con la configuración de *squid*, se tendrá que hacer con la ayuda de las reglas de *iptables*, es decir, para todas las peticiones de las subredes que entren por la interfaz de red eth1 que se hagan hacia el puerto 80 (HTTP) se redireccionen al puerto 3128 (*squid*). Se tiene que hacer una regla para cada subred, como se muestra en la figura 3.51.

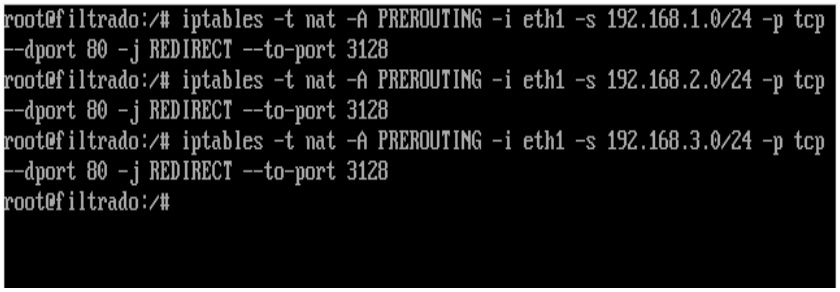

Figura 3.51Reglas para redireccionar al puerto 3128

El bit de reenvió se hace para que los paquetes sean enviados de una subred a otra, como se muestra en la figura 3.52.

Figura 3.52 Bit de reenvío.

3.6 Instalación de *SquidGuard*

Para instalar el programa *SquidGuard* situarse en el directorio raíz y escribir el siguiente comando *apt-get install squidguard,* como se muestra en la figura 3.53.

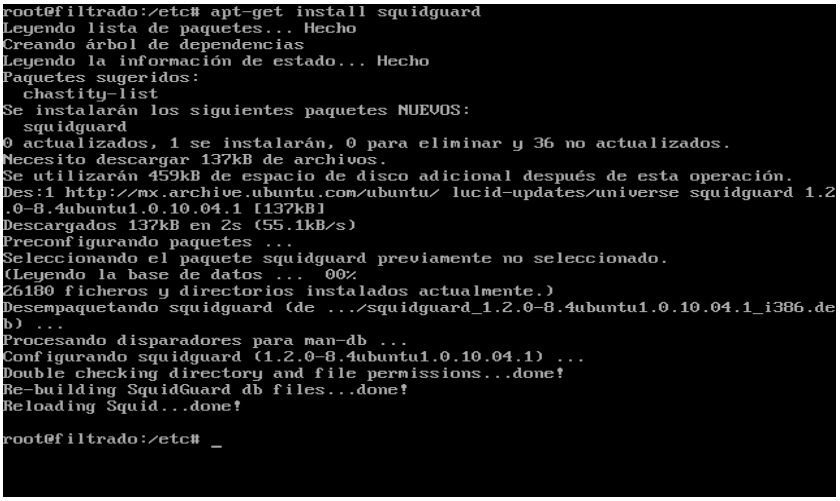

Figura 3.53 Instalación de squidguard

3.6.1 Descarga del archivo *blacklists*

Para descarga el archivo de las listas negras o *blacklists* que contiene millones de url y dominios que se encuentran divididos en diferentes categorías, se hace desde la página siguiente:

- http://cri.univ-tlse1.fr/blacklists/download/blacklists.tar.gz

Se debe de descargar en la ruta */var/lib/squidguard/db.* El comando para descargar el archivo es *wget [ruta completa del archivo a descargar]*, después de terminar la descarga del archivo blacklists se realiza la descompresión con el comando *tar xvzf blacklists.tar.gz* como se muestra en la figura 3.54.

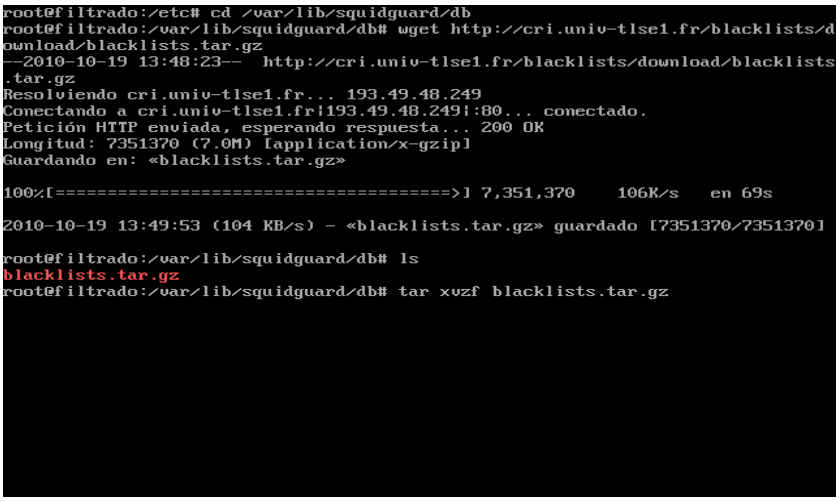

Figura 3.54 Descarga del archivo blacklists.tar.gz

En la figura 3.55 se muestran las categorías que están dentro del directorio blacklists. Solamente las categorías deben estar dentro del directorio db para copiar todos los archivos que contiene el directorio blacklists al directorio db se usa el siguiente comando:

cp –rv /var/lib/squidguard/db/blacklists/ /var/lib/squidguard/db.*

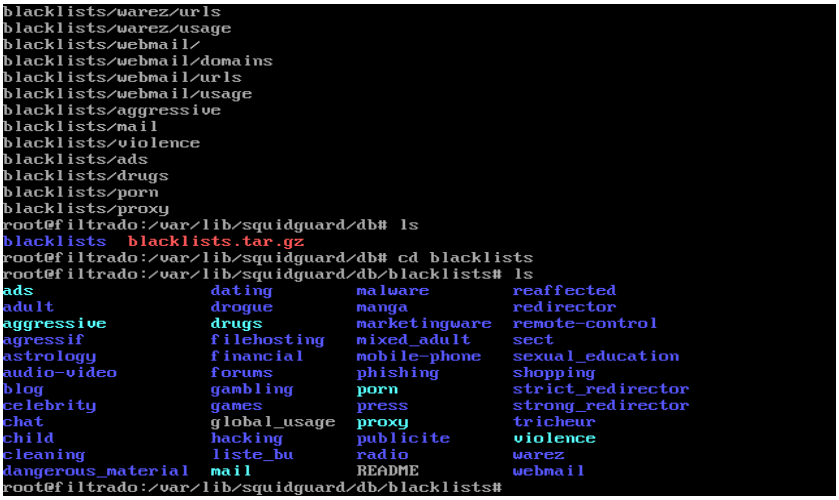

Figura 3.55 Descompresión del archivo *blacklists*.tar.gz

Al terminar de copiar todos los archivos se deberá de borrar el archivo blacklists.*tar*.gz y el directorio *blacklists* quedando de la siguiente manera, solamente las categorías dentro del directorio */db* como se muestra en la figura 3.56.

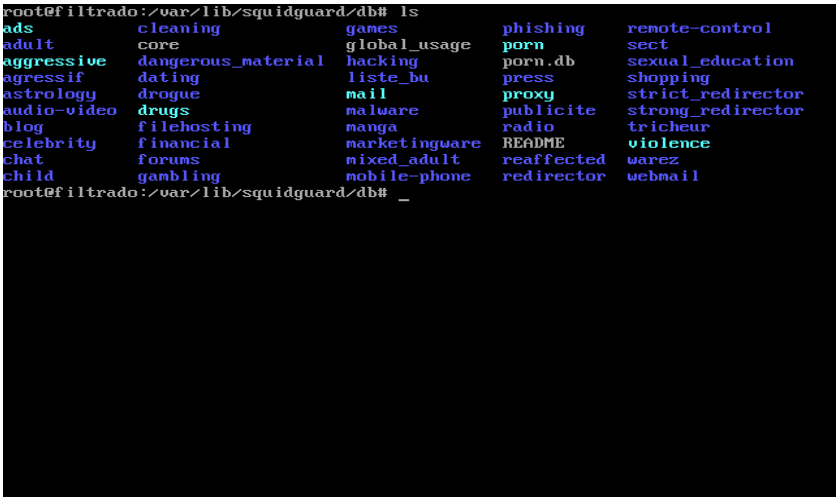

Figura 3.56 Conversión de categorías en formato de base de datos

3.6.2 Configuración del archivo *squidGuard*.conf

Para acceder al archivo de configuración squidGuard.conf situarse en la dirección */etc/squid.* Acceder al archivo por medio del editor de texto vi como se muestra en la figura 3.57.

Figura 3.57 Acceder al archivo de configuración *squidGuard*.conf

En el archivo de configuración *squidGuard.conf*, las primeras dos líneas no se modifican, ya que la primer línea muestran en donde se encuentra las categorías de la *blacklists*, y la segunda hace referencia de donde está el archivo log de *squidGuard*. Por otra parte se configura el horario de trabajo de *squidGuard* para este proyecto será de 7:30 de la mañana hasta las 8.30 de la noche como se muestra en la figura 3.58.

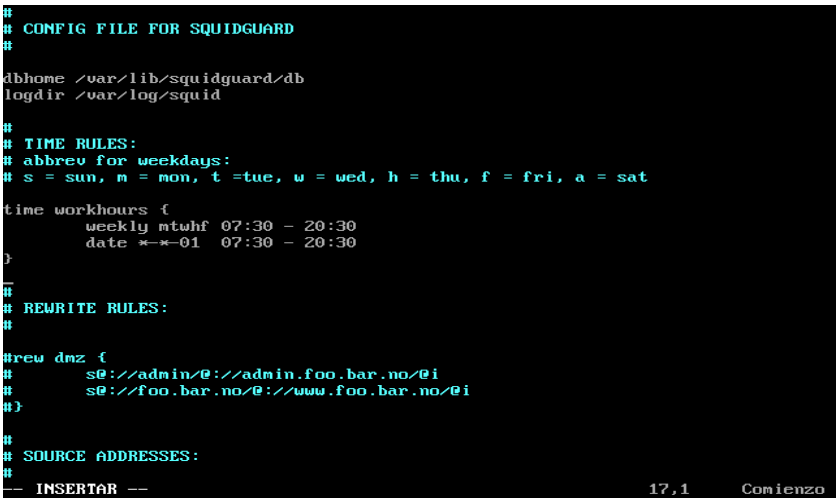

Figura 3.58 Configuración del horario squidGuard.conf

Se agregan los nombres de cada una de las subredes con sus respectivos rangos de direcciones IP a las cuales se restringirá el acceso a páginas web, como se muestra en la figura 3.59.

```
src direccion {
        ip      192.168.1.30-192.168.1.40
                }
src salaprofesores {
        ip      192.168.2.30-192.168.2.40
        }
src aulacomputo {
        ip      192.168.3.30-192.168.3.70
        }
```

Figura 3.59 Rango de direcciones a restringir el acceso a páginas web

Se agregan las categorías por bloques a las que se negará el acceso. como son adult, astrology, chat, games, porn, drugs, violence, aggressive, manga, y shopping como se muestra en la figura 3.60

```
dest adult {
        domainlist      adult/domains
        urllist         adult/urls
        }
dest porn {
        domainlist      porn/domains
        urllist         porn/urls
        }
dest games {
        domainlist      games/domains
        urllist         games/urls
}
dest astrology {
        domainlist      astrology/domains
        urllist         astrology/urls
        }
dest violence {
        domainlist      violence/domains
        urllist         violence/urls
        }
dest chat {
        domainlist      chat/domains
        urllist         chat/urls
        }
dest drugs {
        domainlist      drugs/domains
        urllist         drugs/urls
        }
dest aggressive {
-- INSERTAR --                                  98,9          58%
```

Figura 3.60 Categorías por dominio y url

A continuación se prohíben las categorías ya definidas para cada una de las subredes como se muestra en la figura 3.61

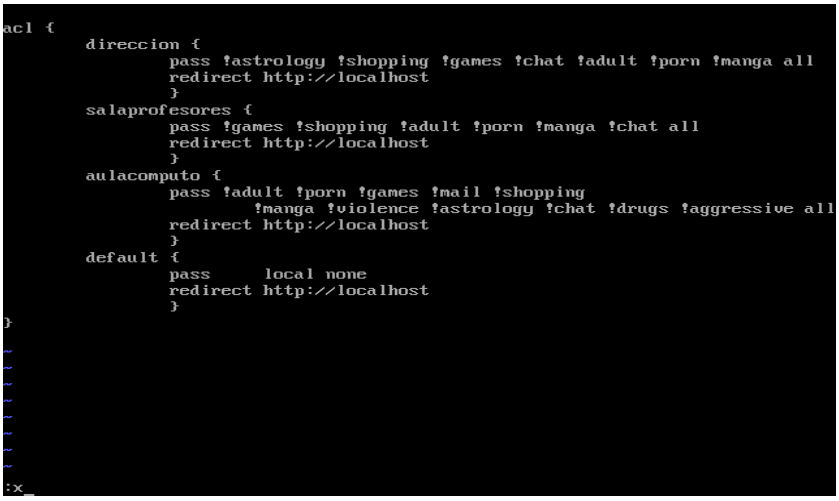

Figura 3.61 Denegando accesos por subred

Al terminar de configurar el archivo guardar los cambios y salir. Por último para confirmar que la configuración del archivo squidGuard.conf funciona correctamente, se convierten las categorías en formato de base de datos para que la consulta sea rápida y efectiva, con el comando *squidGuard –C all*, como se muestra en la figura 3.62.

Figura 3.62 Creación de archivos en formato de base de datos

Este proceso tarda algunos minutos dependiendo de la cantidad de categorías que se están convirtiendo Para observar este proceso se abre otra consola con la combinación de teclas *alt* + *flecha derecha* se ingresa usuario y contraseña de root y con el comando *tail –f /var/log/squid/squidGuard.log* se observará la creación de las categorías en archivos de base de datos como se muestran en las figuras 3.63.

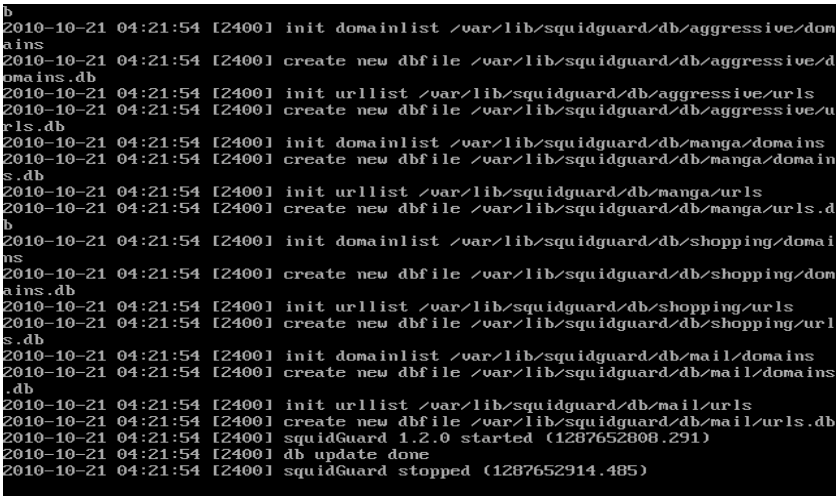

Figura 3.63 Comprobación de los archivos de base de datos

En caso de que se muestre algún error en la línea de comandos, se deberá de revisar la sintaxis en el archivo de configuración de *squidGuard.*

3.7 Instalación de sarg

Para instalar el programa *Sarg* situarse en el directorio raíz y escribir el siguiente comando *apt-get install sarg* como se muestra en la figura 3.64

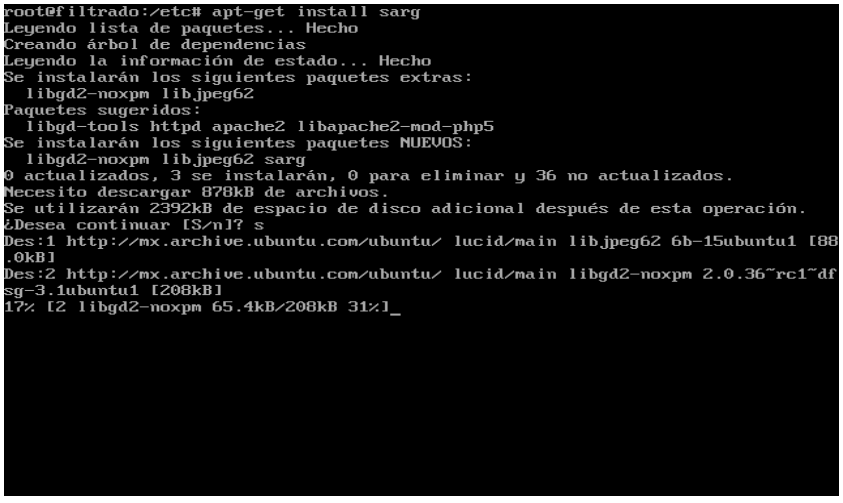

Figura 3.64 Instalación de sarg

3.7.1 Configuración del archivo sarg.conf

Para acceder al archivo de configuración sarg.conf situarse en la dirección */etc/sarg.* Acceder al archivo por medio del editor de texto vi como se muestra en la figura 3.65.

```
root@filtrado:/etc/apache2/sites-available# cd /etc/sarg
root@filtrado:/etc/sarg# ls
css.tpl        exclude_hosts  sarg.conf         user_limit_block
exclude_codes  exclude_users  sarg-reports.conf  usertab
root@filtrado:/etc/sarg# vi sarg.conf
```

Figura 3.65 Acceso al archivo sarg.conf

En este archivo se configura la presentación del formato que tendrá el reporte de la actividad de la red, como por ejemplo el nombre del reporte, el tipo de letra, el tamaño de la letra, el fondo, etc. como se muestra en la figura 3.66

```
# TAG:  access_log file
#       Where is the access.log file
#       sarg -l file
#
access_log /var/log/squid/access.log

# TAG: graphs yes¦no
#       Use graphics where is possible.
#           graph_days_bytes_bar_color blue¦green¦yellow¦orange¦brown¦red
#
#graphs yes
#graph_days_bytes_bar_color orange

# TAG:  title
#       Especify the title for html page.
#
title "Uso de la red de la escuela"

# TAG:  font_face
#       Especify the font for html page.
#
font_face Tahoma,Verdana,Arial

# TAG:  header_color
#       Especify the header color
#
header_color darkblue

# TAG:  header_bgcolor
-- INSERTAR --                                    48,35         5%
```

Figura 3.66 Archivo de configuración sarg.conf

Los reportes se pueden generar por hora, diario, semanal y mensual. Para fines de este proyecto el reporte de cada hora se genera cada 17 minutos,

para el reporte diario se genera a las 6:25 hrs de cada día, para el reporte semanal se genera el séptimo día de cada semana a las 6:47 hrs y por último para el reporte mensual se genera el primer día de cada mes a las 6:52 hrs. Todas estas configuraciones se realizan en el archivo *crontab* que se encuentra dentro del directorio */etc*, como se muestra en la figura 3.67.

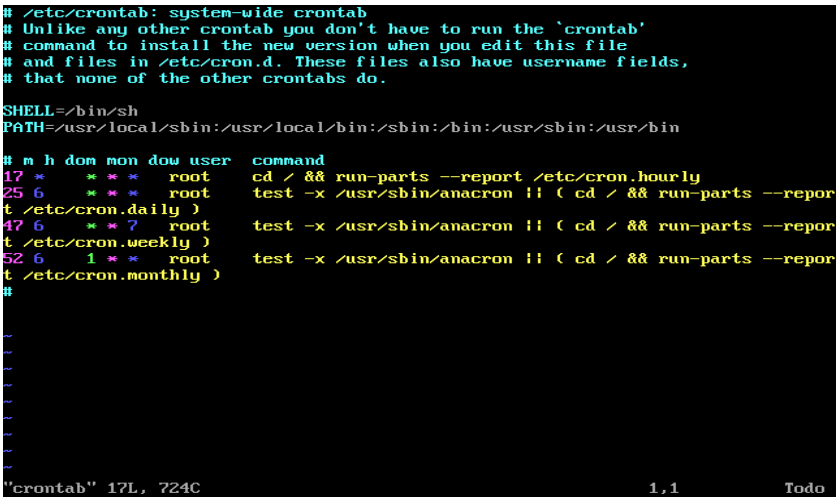

Figura 3.67 Archivo crontab.

3.8 Instalación del servidor apache2

Para instalar el servidor apache2 situarse en el directorio raíz y escribir el siguiente comando *apt-get install apache2* como se muestra en la figura 3.68

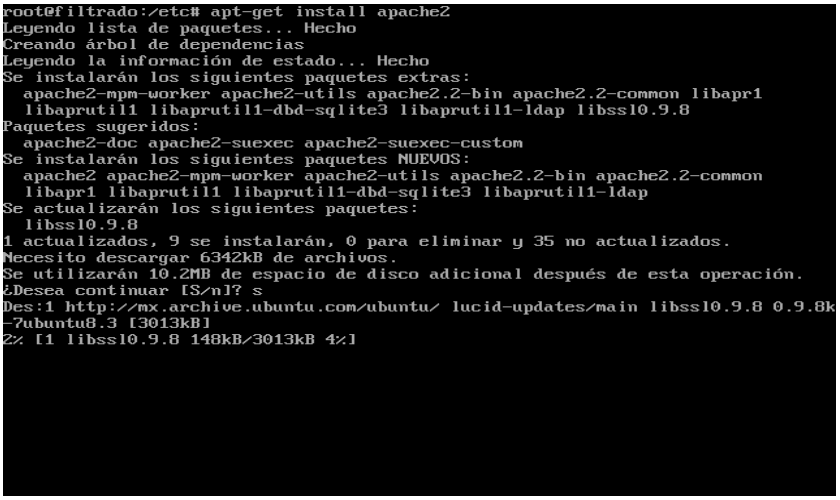

Figura 3.68 Instalación del servidor apache2

3.8.1 Configuración del archivo *default*

Para acceder al archivo de configuración *default* del servidor Apache situarse en la dirección /etc/apache2/sites-available. Acceder al archivo por medio del editor de texto vi. Prácticamente en el archivo default de *Apache* no se tiene que configurar nada, como se muestra en la figura 3.69. Sólo si se requiere cambiar el mail del administrador o en su defecto la ruta del archivo en formato *HTML* que mostrara al usuario que la pagina a la que quiere acceder está bloqueada por defecto se encuentra el archivo *index.html* en la ruta */var/www*.

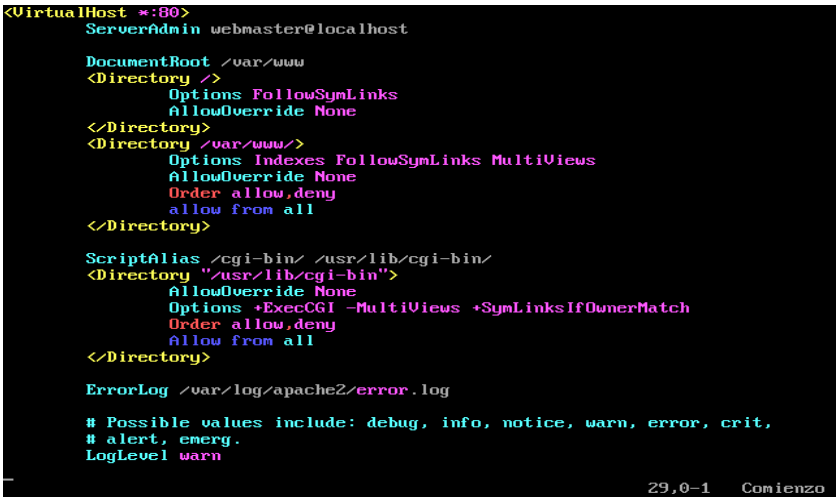

Figura 3.69 Configuración de archivo default de apache2

Para fines de este proyecto editar el archivo index.html que se encuentra en la ruta */var/www,* el código fuente se muestra en la figura 3.70, para que al momento que un usuario quiera acceder a una página que esta etiquetada como prohibida se despliegue el mensaje de error como se muestra en la figura 3.71.

```
<html>
        <head>
        <title> ¡ ACCESO DENEGADO ! </title>
        </head>
        <style type = "text/CCS">
                body {
                        background:black
                    }
                p {
                        font-weight:bold;
                        font-family:Elephant;
                        font-size:60;
                        color:red;
                  }
                h1 {
                        font-weight:bold;
                        font:size:40;
                        font-family:arial;
                        color:white;
                   }
        </style>
        <marquee width = 830 behavor = slide loop = 10 bgcolor = yeloow>
        <p aling = center> * ACCESO DENEGADO * </p>
        <br>
        <br>
        <hr>
        <br>
        <br>
        <br>
        <h1 aling = center> ERROR: El contenido de la página a la cual
                            se quiere acceder está bloqueada, en estos
                            momentos se está enviando un reporte.
        </h1>
</html>
```

Figura 3.70 Archivo index.html

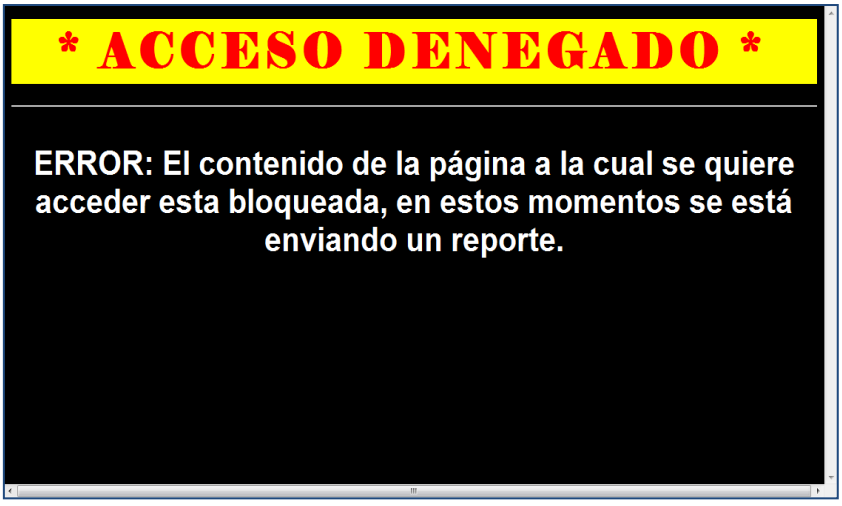

Figura 3.71 Página de error.

CAPÍTULO IV

RESULTADOS

4.1 Pruebas

Las pruebas que se efectuaron en este proyecto se dividieron en dos etapas:

Para primera etapa se requirieron de tres personas para cada una de las subredes: dirección, sala de profesores y aula de cómputo explicando previamente el funcionamiento del filtrado de contenido web, teniendo en cuenta la explicación comenzaron a navegar a través de la internet. En la navegación se les comento de las categorías que se prohibieron anteriormente para cada una de las subredes.

Como habría de esperarse el resultado fue el óptimo ya que al acceder a algunas páginas se desplegaba el mensaje de error en cada una de las subredes.

Por ejemplo en la subred de la dirección se intento acceder a las siguientes páginas:

- http://www.animextremist.com/mangas.htm
- http://www.babosas.com
- http://www.juego.com

En la subred de la sala de profesores se intentó acceder a las siguientes páginas:

- http://www.quehoroscopo.com/
- http://www.juegos7.com/
- http://www.suburbia.com.mx/index.html

Para la subred de aula de cómputo si intentó acceder a las siguientes páginas:

- http://www.orgasmatrix.com
- http://www.maxi-juegos.com/
- http://www.soyunazorra.com/

Cabe mencionar que este último usuario accedió a las propiedades del navegador intentado deshabilitar la opción de usar un servidor proxy, esto no se llevo a cabo ya que esta opción esta deshabilitada por default porque el servidor proxy de *Squid* es transparente.

En la segunda etapa de las pruebas se requirieron de otras tres personas para cada una de las subredes, sin explicar que navegarían a través de un filtrado de contenido web.

Cada uno de los usuarios pretendía acceder a las siguientes páginas web:

- http://www.youtube.com/
- http://prodigy.msn.com/
- Messenger

Como era de esperarse los usuarios que se encontraban en las subredes de la dirección y la sala de profesores no tuvieron ningún problema con las dos primeras páginas ya que éstas no se encuentran en la categoría de acceso denegado, con respecto al Messenger no se conectaron puesto que este se encuentra dentro de las categorías de acceso denegado

En particular el usuario que se encontraba en la subred de la aula de cómputo no pudo acceder a ninguna de las páginas web, ni conectarse al Messenger, dando como resultado la pagina de error, al darse cuenta que los demás usuarios podían acceder las páginas web antes mencionadas decidió cambiarse de computadora, pero esta también pertenecía a la subred de la aula de computo.

Finalmente se les explico el motivo por cual se desplegaba la pagina de error y no podían acceder a las páginas web que consultaban.

CONCLUSIONES

Como se puede observar se cubrieron en su totalidad tanto el objetivo general y los objetivos particulares, con base a la información y configuraciones que se realizaron en cada uno de los programas que integran el filtrado de contenido web.

Las pruebas que se realizaron anteriormente arrojan resultados favorables, puesto que la implementación del servidor de filtrado de contenido web funciona correctamente para cada una de las subredes, teniendo así el control de las páginas web con contenido inapropiado para los niños, a demás de que otro aspecto favorable para esta implementación, es que reduce el riesgo de infección viral causada por páginas con código malicioso.

Un punto a resaltar en este trabajo es que sirve como base para implementar un filtrado de contenido web, no nada más en una escuela pequeña, sino también en cualquier escenario como por ejemplo una empresa pequeña y mediana, en una universidad, en una biblioteca, hasta en lugares pequeños como un café internet, etc.

BIBLIOGRAFÍA

Behrouz A. ,F. (2002). *Transmisión de datos y redes de comunicaciones*. Madrid, España: Mc Graw Hill.

Honeycutt,J, (1998). *La biblia de internet.* Madrid, España: Anaya Multimedia.

Kurose, J.F. y Ross, K.W.(2004). *Red de computadores: Un enfoque descendente basado en Internet* .México:Addison Wesley.

Pérez, C.M. y Pérez I.C. (1998).*Linux: Guía práctica para usuarios.* Madrid, España: Anaya multimedia.

Peterson, R. (2001). *Linux: Fundamentos de programación.* Bogotá, Colombia: Mc Graw Hill.

Sánchez, S. (1999). *UNIX y Linux: Guía práctica.* México: Alfaomega.

Shah, S. y Soyinka W. (2007). *Manual de administración de Linux* México: Mc Graw Hill.

La guía definitiva para proteger de hackers a sus servidores *Linux*.(2000) *Linux máxima seguridad*. Madrid, España:Prentice Hall.