



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

**“ESTUDIO Y CONFIGURACIÓN PARA LA  
INTEGRACIÓN DE ELEMENTOS DE SEGURIDAD  
BAJO LINUX, CONFIGURABLE MEDIANTE UNA  
INTERFAZ WEB QUE SOPORTE LOS  
PROTOCOLOS IPV4 E IPV6  
SIMULTÁNEAMENTE”**

**TESIS DE GRADO**

**PREVIA A LA OBTENCIÓN DEL TÍTULO DE**

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**PRESENTADO POR**

**GUALPA CAICEDO LEONARDO ENRIQUE  
MALÁN MULLO MARCO VINICIO**

**RIOBAMBA – 2008**

### **AGRADECIMIENTO MARCO**

*Quiero agradecer profundamente, a Dios, quien me ha bendecido con la actitud y el talento necesarios para alcanzar mis metas, con unos padres únicos, distinguidos, laboriosos y ejemplares, quienes me han dado su aliento constante, el soporte y consejos diario para poder encaminarme en la senda correcta, con una hermana cordial, atenta y presta a colaborar en mis horas de hastío y vacilación. Y a todas esas personas que con un poco o mucho, fueron en su momento el sustento anímico para caminar siempre hacia adelante, a todos uds. Gracias, que Dios los bendiga.*  
*Marco V. Malán M.*

### **DEDICATORIA MARCO**

*El presente trabajo, está dedicado a aquellos estudiantes que empiezan la vida estudiantil universitaria, a todas personas laboriosas de mi ciudad, provincia y país, que trabajan diariamente con el único afán de desarrollar y hacer grande el orgullo de ser ecuatoriano, a todas las personas que sin detenerse mucho tiempo en la circunstancia difícil, sacan fuerza y continúan haciendo patria, con esfuerzo y dedicación, a nuestros inmigrantes que con su esmero y talento dejan en alto el nombre nuestro amado país, y especialmente, a la gente de la tierra que me vio nacer, la comunidad el Troje, Cantón Colta, Provincia del Chimborazo, esto va para todos ustedes. Kaika kankunamanta kan.  
Marco V. Malán M.*

## **DEDICATORIA LEONARDO**

*La elaboración de esta tesis va dedicada, en primer lugar a Dios por permitirme consolidar una nueva meta en mi vida, a mis Padres por el constante sacrificio realizado, especialmente a mi Madre por sus enseñanzas, consejos y constante perseverancia para ser cada día mejor, como hijo y como persona. A mi hermano, con el que siempre puedo contar con su apoyo en todo momento. A mis amigos y compañeros de clases, que me supieron brindar su amistad sin esperar nada a cambio y con quienes compartí esta etapa universitaria.*

*Gracias a todos los que hicieron posible la realización de esta tesis.*

*Su amigo Leonardo E. Gualpa C.*

## Firmas de Responsabilidad

Autoridades	Firmas	Fecha
Dr. Romeo Rodríguez. <b>DECANO FACULTAD DE INFORMÁTICA Y ELECTRÓNICA.</b>	<hr/>	<hr/>
Ing. Iván Menes. <b>DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS.</b>	<hr/>	<hr/>
Ing. Patricio Moreno. <b>DIRECTOR DE TESIS.</b>	<hr/>	<hr/>
Ing. Jorge Menéndez. <b>MIEMBRO DEL TRIBUNAL.</b>	<hr/>	<hr/>
Tlgo. Carlos Rodríguez. <b>DIRECTOR CENTRO DE DOCUMENTACIÓN.</b>	<hr/>	<hr/>

“Nosotros, Marco V. Malán M. y Leonardo E. Gualpa C., somos los responsables de las ideas, doctrinas y resultados expuestos en esta Tesis, y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo”.

---

Leonardo E. Gualpa C.

---

Marco V. Malán M.

## ÍNDICE DE ABREVIATURAS

**ACL:** Listas de Control de Acceso  
**API:** Interface para Programas de Aplicación  
**ARP:** Protocolo de Resolución de Direcciones  
**BIA:** Bump en API  
**BIS:** Bump en Pila  
**CGI:** Interfaz Común de Salida  
**CentOS:** Acrónimo de Community ENTerprise Operating System  
**DHCP:** Procotocolo de Configuración de Host Dinámico  
**DMZ:** Zona Desmilitarizada  
**DNS:** Servidor de Nombres de Dominio  
**DNAT:** Traducción de Direcciones de Red de Destino  
**DSTM:** Mecanismo de Traducción Pila Dual  
**DSDM:** Metodología de Desarrollo de Sistemas Dinámico  
**EIS:** Escuela de Ingeniería en Sistemas  
**ESPOCH:** Escuela Superior Politécnica de Chimborazo  
**FTP:** Protocolo de Transferencia de Archivos  
**GNU:** Acrónimo de GNU is not Unix  
**HTTP:** Protocolo de Transmisión de Hipertexto  
**ICMP:** Protocolo de Control de Mensajes en Internet  
**IETF:** Fuerza Objetiva a la Ingeniería del Internet  
**IGMP:** Protocolo de Administración de Grupos en Internet  
**IP:** Protocolo de Internet  
**IPv4:** Protocolo de Internet versión 4  
**IPv6:** Protocolo de Internet versión 6  
**IPsec:** IP Security  
**IRC:** Charla Conferencia por Internet  
**ISATAP:** Protocolo de Direccionamiento para Túneles Automáticos  
**ISP:** Proveedor Servicios Internet  
**LAN:** Red de Área Local  
**MA:** Métodos Ágiles  
**MPLS:** Intercambiador de Etiquetas Multiprotocolo  
**NAT:** Traducción de Direcciones de Red  
**NAPT:** Traducción de Direcciones y Puertos de Red  
**PHP:** Acrónimo de PHP Hypertext Pre-processor  
**QoS:** Calidad de Servicio  
**RFC:** Documentos de borrador del IETF  
**SIT:** Traductor Simplificado para Internet  
**SIIT:** Stateless IP/ICMP Translation  
**SNAT:** Traducción de Direcciones de Red de Origen  
**TCP:** Protocolo de Control para Transmisión  
**UDP:** Protocolo de Datagrama de Usuario  
**XML:** Lenguaje de Marcado Extensible

## ÍNDICE GENERAL

AGRADECIMIENTO  
DEDICATORIA MARCO  
DEDICATORIA LEONARDO  
FIRMAS DE RESPONSABILIDAD  
DERECHOS DE AUTOR  
INDICE DE ABREVIATURAS  
INDICE GENERAL  
INDICE DE TABLAS  
INDICE DE FIGURAS

### INTRODUCCIÓN

### CAPÍTULO I

ESTRUCTURA DE LA INVESTIGACIÓN.....	17
1.1. Título de la Investigación.....	18
1.2. Justificación de la Investigación .....	18
1.3. Objetivos .....	19
1.3.1. Objetivo General .....	19
1.3.2. Objetivos Específicos .....	19
1.4. Hipótesis .....	19
1.5. Métodos y Técnicas .....	20
1.5.1. Métodos .....	20
1.5.2. Técnicas.....	20

### CAPÍTULO II

MARCO REFERENCIAL.....	21
2.1. Características Generales de IPv4 e IPv6 .....	22
2.1.1. IPv4.....	22
2.1.2. IPv6.....	23
2.1.2.1. Paquetes IPv6 .....	25
2.1.2.2. Fragmentación.....	31
2.1.2.3. Direccionamiento IPv6.....	31
2.1.2.4. Notación para las Direcciones .....	32
2.1.2.5. Tipos de Direcciones .....	33
2.1.2.6. Identificación de los Tipos de Direcciones .....	35
2.1.3. Diferencias entre IPv4 e IPv6 .....	36
2.1.4. Ventajas y Desventajas de IPv6 e IPv4 .....	38
2.1.4.1. IPv4 .....	38
2.1.4.2. IPv6 .....	39
2.1.5. Razones de Convivencia entre IPv4 e IPv6.....	40
2.1.6. Técnicas de Convivencia entre IPv4 e IPv6.....	40
2.1.6.1. Pila Dual.....	40
2.1.6.2. Túnel .....	42
2.1.6.3. Traducción de Direcciones .....	49
2.2. Linux CentOS 5.0.....	52
2.2.1. Características Generales .....	52
2.2.2. Soporte al Protocolo IPv6 .....	53



2.2.3.	Instalación mínima de Componentes.....	54
2.3.	Componentes de Seguridad.....	56
2.3.1.	Firewall.....	56
2.3.2.	Proxy.....	58
2.4.	Configuraciones.....	59
2.4.1.	Configuración Firewall.....	60
2.4.2.	Configuración Proxy.....	76
2.4.3.	Configuración Gestor Web.....	81
2.5.	Lenguaje PHP.....	88
2.5.1.	Características Generales.....	88
2.5.2.	Manejo de Archivos.....	90
2.5.3.	Manejo de Comandos Unix.....	91
2.6.	Metodología DSDM.....	96

### **CAPÍTULO III**

#### **ANÁLISIS DE LAS TÉCNICAS DE CONVIVENCIA EN LINUX CENTOS 5 Y**

POLÍTICAS DE RED DE LA E.I.S. ....	103
3.1. Situación Actual de la Intranet de la ESPOCH .....	104
3.1.1. Infraestructura IPv4.....	104
3.1.2. Infraestructura IPv6.....	105
3.2. Políticas de Red del Nodo de Red de la E.I.S.....	105
3.2.1. Infraestructura Tecnológica.....	106
3.2.2. Resultados Obtenidos.....	109
3.3. Aplicación de las Técnicas de convivencia y Selección de la más Idónea.....	110
3.3.1. Análisis de las Técnicas de Convivencia en CentOS 5.0.....	112
3.3.1.1. Pila Dual.....	112
3.3.1.2. Túnel.....	117
3.3.1.3. Traducción de Direcciones.....	125
3.3.2. Comparación de las Técnicas y Selección de la más Idónea.....	126
3.3.2.1. Determinación de los Parámetros a Evaluar.....	127
3.3.2.2. Análisis Comparativo entre las Técnicas de Convivencia.....	129
3.3.2.3. Resultados.....	133

### **CAPÍTULO IV**

DESARROLLO DE LA APLICACIÓN WEB “OPEN SOURCE SECURITY” .....	137
4.1. Acoplamiento de la Metodología DSDM para “O.S. Security” .....	138
4.1.1 DSDM – FASE DE VIABILIDAD.....	138
4.1.1.1. Determinación de Requerimientos Específicos.....	138
4.1.1.2. Determinación de Requerimientos de Interfaces.....	138
4.1.1.3. Determinación de Interfaces de Software.....	139
4.1.1.4. Análisis de Riesgos.....	140
4.1.1.5. Factibilidad Tecnológica.....	141
4.1.2 DSDM – FASE DE NEGOCIO.....	141
4.1.2.1. Definición de los Módulos.....	141
4.1.3 DSDM – INTERACCIÓN DEL MODELO FUNCIONAL.....	145
4.1.3.1 Esquema de interfaz web.....	145
4.1.4 DSDM – INTERACCIÓN DEL DISEÑO Y VERSIÓN.....	146
4.1.4.1 Desarrollo de la aplicación.....	146
4.1.4.2 Prototipo.....	148
4.1.5 DSDM - DESPLIEGUE.....	150
4.1.5.1 Implantación.....	150

4.1.5.2 Pruebas .....	150
4.1.5.3 Documentación.....	163

CONCLUSIONES  
RECOMENDACIONES  
RESUMEN  
SUMMARY  
GLOSARIO  
ANEXOS  
BIBLIOGRAFÍA

## ÍNDICE DE TABLAS

Tabla I.1. Problemas de IPv4 .....	24
Tabla I.2. Campos del encabezado de un paquete IPv6 .....	27
Tabla I.3. Dirección IPv6 válida .....	32
Tabla I.4. Compresión de un campo nulo .....	32
Tabla I.5. Equivalencias de direcciones IPv6.....	33
Tabla I.6. Dirección IPv6 inválida .....	33
Tabla I.7. Omisión de ceros iniciales .....	33
Tabla I.8. Dirección IPv4 camuflada en una dirección IPv6 .....	33
Tabla I.9. Diferencias entre IPv4 eIPv6 .....	36
Tabla I.10. Comandos para una regla IPTABLES.....	63
Tabla I.11. Parámetros para una regla de Iptables. ....	64
Tabla I.12. Opciones para paquetes ICMP en una regla de Iptables. ....	65
Tabla I.13. Opciones para paquetes TCP en una regla de Iptables.....	65
Tabla I.14. Opciones para paquetes UDP en una regla de Iptables .....	66
Tabla I.15. Opciones para objetivos en una regla de Iptables. ....	67
Tabla I.16. Comandos para reglas IP6TABLES.....	72
Tabla I.17. Parámetros para una regla Ip6tables.....	72
Tabla I.18. Opciones para paquetes ICMPv6 en una regla de Ip6tables. ....	73
Tabla I.19. Opciones para paquetes TCP en una regla de Ip6tables.....	73
Tabla I.20. Opciones para paquetes UDP en una regla de Ip6tables .....	74
Tabla I.21. Opciones para objetivos en una regla de Ip6tables. ....	75
Tabla I.22. Parámetros básicos de configuración de Squid. ....	78
Tabla I.23. Parámetros globales de configuración de APACHE.....	83
Tabla I.24. Directivas globales de configuración de APACHE. ....	84
Tabla I.25. Directivas principales de configuración de APACHE.....	85
Tabla I.26. Directivas de sección para la configuración de APACHE.....	87
Tabla I.27. Comandos para manejo de archivos en PHP. ....	90
Tabla I.28. Reglas metodología DSDM .....	98
Tabla II.29. Matriz de evaluación de criterios.....	129
Tabla II.30. Evaluación Configuración .....	130
Tabla II.31. Evaluación Compatibilidad Hardware .....	131
Tabla II.32. Evaluación Compatibilidad Software.....	131
Tabla II.33. Evaluación Integridad .....	132
Tabla II.34. Evaluación Interoperabilidad.....	132
Tabla II.35. Evaluación Rendimiento .....	133
Tabla II.36. Resultados Totales Comparación.....	134
Tabla III.37. Tabla de riesgos. ....	141

## INDICE DE FIGURAS

Fig. I.1. Datagramas IPv4 e IPv6.....	26
Fig. I.2. Cabecera IPv6.....	26
Fig. I.3. Jerarquía tipo de direcciones IPv6.....	34
Fig. I.4. Estructura Pila Dual.....	41
Fig. I.5. Estructura Túnel.....	42
Fig. I.6. Encapsulación Paquetes IPv6 en IPv4.....	43
Fig. I.7. Esquema de Túnel.....	45
Fig. I.8. Pantalla inicial instalación CentOS 5.....	54
Fig. I.9. Selección Componentes a instalar.....	55
Fig. I.10. Finalización instalación CentOS 5.....	55
Fig. I.11. Esquema Firewall.....	57
Fig. I.12. Componentes NetFilter.....	60
Fig. I.13. Tratamiento de paquetes.....	62
Fig. I.14. Proceso de desarrollo DSDM.....	102
Fig. II.15. Infraestructura IPv4 de la ESPOCH.....	104
Fig. II.16. Infraestructura tecnológica de la EIS.....	108
Fig. II.17. Infraestructura Laboratorios EIS.....	109
Fig. II.18. Ejecución ping6.....	111
Fig. II.19. Verificación interfaces red activas.....	111
Fig. II.20. Verificación interfaces red activas con Pila Dual.....	113
Fig. II.21. Asignación dirección IPv6 a una interfaz.....	114
Fig. II.22. Verificación IPv6 en navegador web.....	115
Fig. II.23. Captura de paquetes IPV6 durante pruebas con navegador web.....	115
Fig. II.24. Conexión mediante ping6 en Pila Dual.....	116
Fig. II.25. Captura de paquetes ICMPv6 usando Pila Dual.....	117
Fig. II.26. Escenario para túnel manual.....	119
Fig. II.27. Interconectividad entre redes IPv4 distintas.....	119
Fig. II.28. Interconectividad entre redes IPv4 distintas.....	120
Fig. II.29. Interfaces activas para configurar túneles.....	120
Fig. II.30. Pasos para configurar extremo 1 del túnel.....	121
Fig. II.31. Verificación túnel activo en extremo 1.....	121
Fig. II.32. Pasos para configurar extremo 2 del túnel.....	122
Fig. II.33. Verificación túnel activo en extremo 2.....	122
Fig. II.34. Interconectividad en los extremos del túnel.....	123
Fig. II.35. Acceso extremo túnel usando navegador.....	123
Fig. II.36. Captura de paquetes IPV6 durante acceso web con direcciones IPv6.....	124
Fig. II.37. Interconectividad en los extremos del túnel.....	124
Fig. II.38. Captura de paquetes durante interconectividad.....	125
Fig. II.39. Porcentajes entre Comparación de Mecanismos.....	134
Fig. III.40. Estructura web para O.S Security.....	145
Fig. III.41. Pantalla inicial "O.S Security".....	149
Fig. III.42. Módulo para configurar IPTABLES.....	149
Fig. III.43. Escenario Pruebas con IPTABLES.....	151
Fig. III.44. Módulo para configurar IPTABLES.....	151
Fig. III.45. Conectividad exitosa hacia enrutador.....	152
Fig. III.46. Conectividad fallida hacia enrutador.....	152
Fig. III.47. Reglas registradas en el firewall.....	153
Fig. III.48. Acceso hacia recurso web de equipo remoto desde equipo cliente.....	153
Fig. III.49. Restricción hacia intranet desde equipos remotos.....	154

Fig. III.50. Configuración para usar SSH .....	154
Fig. III.51. Denegación hacia SSH por regla firewall.....	154
Fig. III.52. Acceso a SSH por regla de firewall hacia una IP específica .....	155
Fig. III.53. Escenario Pruebas con IP6TABLES.....	156
Fig. III.54. Módulo para configurar IP6TABLES .....	156
Fig. III.55. Conexión exitosa hacia el enrutador.....	157
Fig. III.56. Reglas registradas en el firewall IPv6.....	157
Fig. III.57. Acceso equipo remoto desde equipo cliente.....	158
Fig. III.58. Escenario Pruebas para Proxy .....	159
Fig. III.59. Módulo para administrar SQUID.....	159
Fig. III.60. Redireccionamiento de puertos desde el firewall hacia el proxy .....	160
Fig. III.61. Redireccionamiento del navegador hacia el proxy.....	160
Fig. III.62. Acceso a equipo remoto desde equipo cliente usando proxy.....	161
Fig. III.63. Creación de nueva regla ACL.....	161
Fig. III.64. Listado de reglas ACL registradas.....	162
Fig. III.65. Ordenamiento de reglas ACL .....	162
Fig. III.66. Bloqueo a dirección remota por regla ACL registrada.....	163

# INTRODUCCIÓN

La evolución de Internet ha supuesto una revolución en el desarrollo de las comunicaciones y de la información, prueba inequívoca de ello es la inmensidad de información que existe en Internet. Cuando IPv4 fue estandarizado, nadie podía imaginar que se convertiría en lo que es hoy una arquitectura de amplitud mundial, con un número de usuarios superior al centenar de millones y que crece de forma exponencial. Aquella primera "Internet" fundada, sobre todo con fines experimentales, científico-técnicos y por supuesto con objetivos militares, no se parece en nada a la actual.

Debido a esa expansión vertiginosa de Internet ha provocado que el número de IPs públicas de 32 bits disminuya considerablemente. Fue entonces que la organización IETF; vio la necesidad de crear un nuevo protocolo, debido a la gran dimensión de las tablas de ruteo en el backbone de Internet, que lo hace ineficaz y perjudica los tiempos de respuesta. Debido a la multitud de nuevas aplicaciones en las que IPv4 es utilizado, ha sido necesario agregar nuevas funcionalidades al protocolo básico, aspectos que no fueron contemplados en el análisis inicial de IPv4, lo que genera complicaciones en su escalabilidad para nuevos requerimientos y en el uso simultáneo de dos o más de dichas funcionalidades.

Con las consideraciones mencionadas anteriormente, surgió el protocolo IPv6 cuya finalidad es cubrir el déficit de direcciones IPv4 mediante la inclusión de direcciones de 128 bits, además de incluir nuevas funcionalidades que hacen que IPv6 sea un protocolo robusto y seguro. Aún con todas las mejoras en IPv6 se tiene que seguir utilizando IPv4, porque IPv6 aún se encuentra en su fase de pruebas. Las implementaciones existentes de este protocolo aún presentan problemas de interoperabilidad con algunos otros protocolos de capas superiores, debido a que cada plataforma tiene su propia forma de implementación. Además se debe tomar en cuenta si se desea realizar una migración completa hacia IPv6, la infraestructura tendría un cambio vertiginoso; lo cual no es posible hacerlo de forma directa ya que la infraestructura con la que se cuenta actualmente se maneja bajo el protocolo IPv4, por ello dicha migración hacia IPv6 debe realizarse de una forma progresiva.

Por éstas razones surge la necesidad de crear mecanismos de coexistencia entre IPv4 e IPv6 que permitan la interoperabilidad entre los mismos, que permitan la comunicación entre infraestructuras que manejen el protocolo IPv4 y aquellas que manejen el protocolo IPv6. Entre los mecanismos de transición que han evolucionado paulatinamente, tenemos: Pila Dual, Túneles y Traducción de Direcciones; los mismos que analizaremos durante la investigación de esta tesis.

Por otro lado, con la aparición de estos nuevos paradigmas de comunicación se ha visto la necesidad de desarrollar herramientas que permitan la adecuada administración de los recursos de red, proveer mecanismos de seguridad que permitan la integridad de la información que se maneje. Por las razones citadas anteriormente, nuestra tesis se enfoca en analizar los mecanismos de transición desarrollados para los protocolos IPv4 e IPv6, además de diseñar una herramienta que incorpore mecanismos de administración de red, entre los que mencionamos el uso de Firewalls bajo el protocolo IPv4 e IPv6 que permitan la adecuada administración de los paquetes que se transportan en una red, conjuntamente con el uso de un administrador de contenidos Proxy bajo IPv4 el mismo que permite obtener un mejor rendimiento del ancho de banda usado.

La incorporación de éstos 2 mecanismos de administración de red: Firewall IPv4 e IPv6 y Proxy se lo realiza mediante una interfaz Web, la misma que permite la configuración de parámetros específicos que conlleven a la adecuada administración de los recursos de red. La selección de los parámetros de configuración, han sido tomados en referencia a las políticas de red existentes en el nodo de red de la Escuela de Ingeniería en Sistemas de la ESPOCH.

El desarrollo y ejecución de este proyecto de tesis, se lo efectúa en base a Open Source, es decir se lo realizará con herramientas Software Libre que para nuestro caso serán: Linux CentOS 5 como sistema operativo, Apache como Gestor Web y PHP como Lenguaje de Programación. La finalidad de usar Open Source es debido a que el uso de estas herramientas no involucra el pago de licencias para su utilización.

La estructura manejada en este proyecto investigativo es la siguiente:

En el Capítulo I se abordan temas relacionados con la estructuración de la investigación, en los cuales se describe: el título de la tesis, la justificación, los objetivos planteados, la hipótesis a demostrar y finalmente explicar los métodos y técnicas a utilizar durante el proceso de desarrollo de la tesis.

En el Capítulo II, se describe las características generales de los protocolos IPv4 e IPv6, además de las especificaciones manejadas por las Técnicas de Convivencia para los Protocolos IPv4 e IPv6, entre las técnicas de convivencia a analizar, se tiene: Pila Dual, Túneles y Traducción de Direcciones. Seguido de un estudio de las funcionalidades que incorporan los entornos Linux CentOS 5, Apache y PHP; para el desarrollo de la aplicación Web de seguridad denominada "Open Source Security – O. S Security". Y finalmente, se realiza un análisis de la Metodología DSDM (Dynamic System Development Method), que permite seguir un lineamiento adecuado durante el desarrollo de nuestra tesis.

En el Capítulo III, tomando como referencia las políticas de red existentes en el nodo de red de la Escuela de Ingeniería en Sistemas de la ESPOCH, se procede al análisis de la información obtenida, para en base a dicha información estructurar los parámetros de configuración que incorporará "O.S Security" en su entorno gráfico. Por otra parte, se analizará las configuraciones de las Técnicas de Convivencia descritas en el Capítulo II, bajo Linux CentOS 5, para de esta manera efectuar un Estudio Comparativo y determinar cual técnica se acopla de mejor manera a Linux CentOS 5 y a nuestra herramienta "O.S Security".

Finalmente en el Capítulo IV, una vez seleccionada la técnica de convivencia adecuada para CentOS 5 y delineados los parámetros de configuración para Firewalls en IPv4 e IPv6 y Proxy, se procede al acoplamiento de la Metodología DSDM en el desarrollo de la aplicación Web "O.S Security", proyecto de investigación y desarrollo de esta tesis.



---

# CAPÍTULO I

## ESTRUCTURA DE LA INVESTIGACIÓN

---

La elaboración de un trabajo de investigación se realiza con el soporte de una metodología que permita lograr los resultados esperados y no tener contratiempos posteriores, razón por lo que en este capítulo se da a conocer aspectos de la formulación de la tesis, para el estudio y configuración de elementos de seguridad bajo Linux, configurable mediante una interfaz Web que soporte los protocolos IPv4 e IPv6 simultáneamente. Además de dar una descripción de las técnicas y métodos utilizados para su desarrollo.

## 1.1. TÍTULO DE LA INVESTIGACIÓN

El título establecido para esta investigación es:

**“ESTUDIO Y CONFIGURACIÓN PARA LA INTEGRACIÓN DE ELEMENTOS DE SEGURIDAD BAJO LINUX, CONFIGURABLE MEDIANTE UNA INTERFAZ WEB QUE SOPORTE LOS PROTOCOLOS IPV4 E IPV6 SIMULTANEAMENTE”.**

Mediante la cual se define el alcance de éste proyecto investigativo, que es el estudio y configuración mediante un entorno Web de elementos de seguridad bajo Linux.

## 1.2. JUSTIFICACIÓN DE LA INVESTIGACIÓN

Para el desarrollo de soluciones informáticas, la tendencia de la comunidad informática es utilizar Open Source, ya que se ha comprobado el correcto desempeño de sus componentes, además que se disminuyen los costos de investigación, implementación y ejecución de las soluciones desarrolladas.

Debido a que hay implementaciones de seguridad utilizando instalaciones de Linux con varios servicios que ocupan los recursos de una PC, sería interesante el estudio de una instalación básica de Linux en su distribución CentOS 5.0 (ya que esta distribución es usada en la ESPOCH), que nos provea funciones específicas de seguridad y que además nos brinde una manera amigable de configurar IPTABLES en IPv4 e IPv6 y SQUID (motivo de nuestro estudio), tomando como referencia las políticas de red que se manejan en el nodo de red de la Escuela de Ingeniería en Sistemas de la ESPOCH.

En lo que a las mejoras de IPv6 se refiere, existen muchos estudios que mencionan las múltiples ventajas que tiene frente a su antecesor IPv4, nuestra investigación aspira lograr la convivencia de los protocolos IPv4 e IPv6 en la herramienta a desarrollar denominada “O.S. Security”, tras el estudio de las técnicas desarrolladas para esta

función, como son: Pila Dual, Túneles y Traducción de Direcciones; para la posterior selección de la más idónea, que permita acoplarse a los requerimientos de nuestra herramienta.

### 1.3. OBJETIVOS

#### 1.3.1. OBJETIVO GENERAL

Integrar la configuración de los elementos de seguridad Firewall y Proxy bajo Linux CentOS 5.0, en una sola herramienta que provea opciones de configuración personalizadas, a través de una interfaz Web y que soporte los protocolos IPv4 e IPv6 simultáneamente.

#### 1.3.2. OBJETIVOS ESPECÍFICOS

- Determinar los requerimientos básicos para instalar Linux CentOS 5.0, que soporte la adición posterior de los elementos de seguridad Firewall (Iptables) y Proxy (Squid) a usar.
- Analizar la configuración de los elementos de seguridad (firewall y proxy) manejados en Linux CentOS 5.0 en los protocolos de IPv4 e IPv6, tras examinar las políticas de seguridad manejadas en el nodo de red de la E.I.S.
- Caracterizar las técnicas de convivencia de los protocolos de IPv4 e IPv6, PILA DUAL, TÚNELES, TRADUCCIÓN DE DIRECCIONES y seleccionar la más apropiada para nuestra herramienta.
- Desarrollar una aplicación Web utilizando las herramientas APACHE como gestor Web, MYSQL como gestor de base de datos y PHP como lenguaje de desarrollo, que permita la configuración intuitiva de los parámetros necesarios de cada elemento de "O.S. Security".

### 1.4. HIPÓTESIS

Es posible crear una herramienta, integrando las funciones de seguridad como el Firewall y Proxy de Linux en su distribución CentOS 5.0, que soporte los protocolos

IPv4 e IPv6 y que permita una configuración personalizada de la misma, mediante una interfaz Web intuitiva.

## 1.5. MÉTODOS Y TÉCNICAS

### 1.5.1. MÉTODOS

El método a utilizar es el deductivo, debido a que existe información general y poco específica, dispersa por diferentes fuentes, por lo que se hace necesario analizarla y clasificarla para posteriormente ser utilizada durante las diferentes etapas de este proyecto investigativo.

El desarrollo este proyecto se basa en un proceso Sintético-Analítico, presentando definiciones, principios, leyes o normas generales de las cuales se extraerán conclusiones o consecuencias.

### 1.5.2. TÉCNICAS

**Libros y Revistas:** Esta técnica la usa para extraer información acerca de la estructura de los Protocolos IPv4 e IPv6, permitiendo determinar las funcionalidades que incluyen.

**Información en Internet:** Esta técnica se la usa para extraer información actualizada sobre estudios recientes referentes a las Técnicas de Convivencia entre los Protocolos IPv4 e IPv6, además de información relacionada a la configuración de Firewalls y Proxy bajo Linux.

**Estudios Similares:** Esta técnica nos permitirá tener una referencia acerca de estudios efectuados anteriormente al protocolo IPv6, para de esta manera lograr un lineamiento específico para nuestra investigación.

---

## CAPÍTULO II

### MARCO REFERENCIAL

---

En este capítulo se realiza un análisis del Protocolo IPv6, las funcionalidades que incluye frente a su antecesor el Protocolo IPv4. Además, se describe los tipos de direccionamiento que su estructura maneja. Por otra parte se analiza las razones de Convivencia entre los Protocolos IPv4 e IPv6, para su posterior estudio y análisis; identificando las técnicas más usadas y que puedan ser configuradas en Linux CentOS 5.

Se efectúa un análisis de los componentes de seguridad más usados para tener una administración eficiente de los recursos de red, entre los que podemos mencionar configuraciones para Firewalls basados en el Protocolo IPv4 e IPv6 y la configuración de un Servidor Proxy. Complementariamente se estudia la Metodología Dynamic Systems Development Method (DSDM) la misma que nos permitirá seguir un orden específico en el desarrollo de la tesis.

## 2.1. CARACTERÍSTICAS GENERALES DE IPV4 E IPV6

Internet se ha introducido en todos los ámbitos de nuestra vida. Precisamente, debido a su imparable crecimiento, la versión 4 del protocolo de Internet (IPv4) se está viendo limitada por el aumento progresivo de aplicaciones que necesitan direcciones IP públicas, globales, válidas para conexiones extremo a extremo, y por tanto, encaminables (enrutables). Por ello, el IETF (Internet Engineering Task Force), organización encargada de la evolución de la arquitectura en la Red; ha diseñado una nueva interpretación, denominada IPv6 (Internet Protocolo versión 6).

Este nuevo modelo tiene muchas soluciones y características que la hacen sucesor de la versión 4 puesto que resuelve sus deficiencias y aporta nuevas funciones acordes a la evolución actual de la red.

### 2.1.1. IPv4

Como se sabe, los nombres que se usa para conectarse a Internet (www.elmundo.es o www.google.com) se traducen en unos números (193.110.128.200 y 216.239.55.100, en nuestro ejemplo anterior) que son los que realmente usa la red. Es algo parecido a lo que pasa a los ciudadanos con la C.I., cada ciudadano tiene su nombre pero el identificativo único que se usa y/o se pide, en la vida diaria, es el número de la C.I.

Las direcciones en IPv4 --que son los que se conoce en la Internet actual-- tienen 32 bits agrupados en 4 grupos de 8 bits, por lo que el conjunto global va de 0.0.0.0 a 255.255.255.255. Por tanto, idealmente se podrían asignar 4.294.967.296 direcciones. Con esto en mente, quienes diseñaron la IPv4 pensaron que esto sería más que suficiente.

El problema está en que las direcciones se asignan en bloques o subredes; o sea, se agrupan, se asignan a alguien (empresa, Universidad, etc.) y todas ellas se consideran ya ocupados (se usen o no).

Las agrupaciones clásicas son:

- **Clase A:** donde se fija el primer octeto y se dejan los otros tres para que el usuario los maneje. Por ejemplo, se le asigna la subred "30.x.x.x". Las IPs asignadas al usuario son  $256*256*256=16.777.216$
- **Clase B:** se fijan los dos primeros octetos y los dos restantes quedan para el usuario. Por ejemplo, "156.23.x.x". Las IPs asignadas al usuario son  $256*256=65536$
- **Clase C:** se fijan los tres primeros octetos y el que resta queda para el usuario. Por ejemplo, "193.110.128.x". Las IPs asignadas al usuario son 256.

El problema, sobre todo en las primeras fases, fue que se asignaban con mucha facilidad y alegría Clases A y B, con lo que el espacio consumido y, sobre todo, el desperdiciado fue/es muy grande.

### 2.1.2. IPv6

Antes de definir con exactitud cuál es el nuevo protocolo de enrutamiento, se dieron propuestas de otros posibles sucesores. Los tres más importantes han sido **PIP** ('P' Internet Protocol), **TUBA** (TCP/UDP With Bigger Addresses) y **SIP/SIPP** (Simple Internet Protocol/Simple Internet Protocol Plus).

El motivo básico por el que surge IPv6 en el seno del IETF (Internet Engineering Task Force), es por la necesidad de crear un nuevo protocolo, debido a la gran dimensión de las tablas de ruteo en el backbone de Internet, que lo hace ineficaz y perjudica los tiempos de respuesta. Debido a la multitud de nuevas aplicaciones en las que IPv4 es utilizado, ha sido necesario agregar nuevas funcionalidades al protocolo básico, aspectos que no fueron contemplados en el análisis inicial de IPv4, lo que genera complicaciones en su escalabilidad para nuevos requerimientos y en el uso simultáneo de dos o más de dichas funcionalidades. Entre las más conocidas se pueden mencionar medidas para permitir la Calidad de Servicio (QoS), Seguridad (IPsec), movilidad y NAT.

Los cambios de escala y orientación del uso del protocolo suponen varios problemas para IPv4, ver **Tabla I.1**:

**Tabla I.1.** Problemas de IPv4

<b>Característica</b>	<b>Descripción</b>
<b>Escala</b>	Cada máquina presente en la red dispone de una dirección IP de 32 bits. Ello supone más de cuatro mil millones de máquinas diferentes. El número asignado a un ordenador no es arbitrario, sino que depende de una estructura más o menos jerárquica (en especial, si pertenece a una red), lo cual ocasiona que se desperdicie una enorme cantidad de direcciones.
<b>Enrutado</b>	Otro de los grandes problemas del crecimiento de Internet es la capacidad de almacenamiento necesaria en las pasarelas (routers) y el tráfico de gestión preciso para mantener sus tablas de encaminamiento. Existe un límite tecnológico al número de rutas que un nodo puede manejar, y como dado que Internet crece mucho más rápidamente que la tecnología que la mantiene, se vio que las pasarelas pronto alcanzarían su capacidad máxima y empezarían a desechar rutas, con lo que la red comenzaría a fragmentarse en subredes sin acceso entre sí.
<b>Multiprotocolo</b>	Cada vez resulta más necesaria la convivencia de diversas familias de protocolos: IP, OSI, IPX... Se necesitan mecanismos que permitan abstraer al usuario de la tecnología subyacente para permitir que concentre su atención en los aspectos realmente importantes de su trabajo. Se tiende, pues, hacia una red orientada a aplicaciones, que es con lo que el usuario interacciona, más que a una red orientada a protocolos (como hasta el momento).
<b>Seguridad</b>	Son necesarios esquemas de autenticación y privacidad, tanto para proteger a los usuarios en sí como la misma integridad de la red ante ataques malintencionados o errores.
<b>Tiempo Real</b>	IPv4 define una red pura orientada a datagramas y, como tal, no existe el concepto de reserva de recursos. Cada datagrama debe competir con los demás y el tiempo de tránsito en la red es muy variable y sujeto a congestión. A



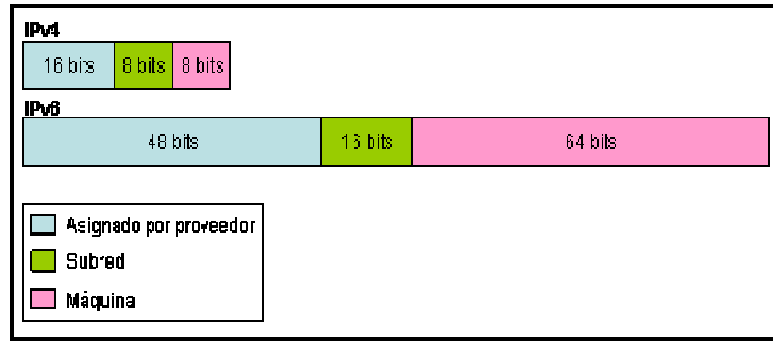
	pesar de que en la cabecera IP hay un campo destinado a fijar, entre otras cosas, la prioridad del datagrama, en la práctica ello no supone ninguna garantía. Se necesita una extensión que posibilite el envío de tráfico de tiempo real, y así poder hacer frente a las nuevas demandas en este campo.
<b>Comunicaciones Móviles</b>	Se necesita una nueva arquitectura con mayor flexibilidad topológica, capaz de afrontar el reto que supone la movilidad de sus usuarios. La seguridad de las comunicaciones, en este tipo de sistemas se ve además, especialmente comprometida.
<b>Facilidad de Gestión</b>	Con el volumen actual de usuarios y su crecimiento estimado, resulta más que obvio que la gestión de la red va a ser una tarea ardua. Es preciso que la nueva arquitectura facilite al máximo esta tarea. Un ejemplo de ello sería la autoconfiguración de los equipos al conectarlos a la red.
<b>Política de Enrutado</b>	Tradicionalmente los datagramas se han encaminado atendiendo a criterios técnicos tales como el minimizar el número de saltos a efectuar, el tiempo de permanencia en la red, etc. Cuando la red pertenece a una única organización eso es lo ideal, pero en el nuevo entorno económico en el que diferentes proveedores compiten por el mercado las cosas no son tan simples. Es imprescindible que la fuente pueda definir por qué redes desea que pasen sus datagramas, atendiendo a criterios de fiabilidad, coste, retardo, privacidad, etc.

La razón de un nuevo protocolo de direccionamiento entonces, se hace evidente debido a los problemas que se vienen suscitando tras las nuevas necesidades existentes y creadas.

#### 2.1.2.1. PAQUETES IPV6

El esquema usado de asignación es similar al anterior (IPv4), pero con los bloques y la capacidad de división mucho mayor. Pongamos el ejemplo de una empresa media que necesita crear muchas subredes para sus delegaciones. Con IPv4 a lo máximo que podría aspirar sería a una Clase B (recordemos, se

fijan los 16 primeros bits y los otros 16 quedarían para la empresa). En IPv6 lo común es que se asigne un /48, donde se fijan los primeros 48 bits, los 16 restantes para hacer subredes (por tanto, 65.535 posibles subredes) y los 64 restantes para la asignación de la máquina, **ver Fig. I.1.**



**Fig. I.1.** Datagramas IPv4 e IPv6

Vale recalcar, que la estructura del protocolo permite una escalabilidad, según las nuevas necesidades y/o aplicaciones y/o servicios lo vayan precisando.

**Estructura**

Un paquete en IPv6 está compuesto principalmente de dos partes:

- a) La cabecera y
- b) Los datos.

a) **La cabecera.**- Está en los primeros 40 bytes del paquete, **ver Fig. I.2:**



**Fig. I.2.** Cabecera IPv6

A continuación, describiremos cada uno de los campos que componen la cabecera de un paquete IPv6, **ver Tabla I.2:**

**Tabla I.2.** Campos del encabezado de un paquete IPv6

<b>Campo (Nº Bits)</b>	<b>Descripción</b>
<b>La versión de IP (4 bits)</b>	Todo el software IP debe verificar este campo antes de procesar el datagrama, para ver si el formato coincide con las especificaciones y la versión esperada. Este campo debería distinguir las versiones de IP, de forma que todas pudieran identificarse como un mismo protocolo a nivel de enlace con el mismo valor de Ethertype. Sin embargo, en la práctica muchas implementaciones de IPv4 no comprueban este campo sino que suponen que el paquete es IPv4 cuando el encabezado de nivel de enlace especifica protocolo IP. Por esto, a pesar de existir el campo versión es necesario asignar a IPv6 un valor propio en el nivel de enlace, como si se tratara de un protocolo diferente de IPv4. Durante el periodo de transición del IPv4 al IPv6, los enrutadores podrán examinar este campo para saber el tipo de paquete que tienen.
<b>La clase de tráfico (8 bits, Prioridad del Paquete)</b>	Utilizado para especificar parámetros de QoS de acuerdo a la especificación de la arquitectura Differentiated Services. Los valores del 0 al 7 indican poca sensibilidad al tiempo lo que permite encolar el tráfico. Los valores del 8 al 15 indican prioridad del tráfico fuera de flujo por lo que no se puede encolar este tipo de tráfico.
<b>Etiqueta de flujo (20 bits)</b>	Permite identificar los paquetes que pertenecen a una sesión concreta (conexión con prioridades y requisitos particulares) entre dos hosts (origen y destino), usado típicamente para solicitar una determinada QoS. Por ejemplo se quiere establecer una videoconferencia, por lo tanto se fija un flujo que garantice un retardo mínimo entre el audio y el video.

<b>Longitud de Carga Útil (16 bits)</b>	Indican el tamaño del paquete en bytes, sin considerar los 40 Bytes de encabezado. Como el valor máximo codificable es 65535, el paquete máximo será de 65575. Teniendo la posibilidad de transmitir paquetes tan grandes, en ciertas circunstancias puede significar un gran aumento en eficiencia. Cuando los paquetes son grandes, el número de paquetes necesarios para enviar cierta información es menor, y cuando hay menos paquetes para enrutar, entonces el enrutador tiene más tiempo para enrutar otros paquetes, o realizar otras tareas (manejo de cache, mantenimiento de tablas, etc). Los encabezados extendidos si se incluyen en la longitud de la carga útil.
<b>Cabecera siguiente (8 bits)</b>	Sirve para indicar si el encabezado está seguido por alguno de los encabezados opcionales (Cabeceras de extensión: Opciones salto por salto, enrutamiento, fragmentación, verificación de autenticidad, carga útil cifrada de seguridad, opciones de destino). Si no existen opciones, este campo indica el protocolo de nivel de transporte al que pertenece el paquete, utilizando los mismos códigos que en IPv4.
<b>Límite de saltos - Tiempo de Vida (8 bits)</b>	Equivalente al campo TTL de IPv4, especifica el número de saltos que un paquete, a nivel de la capa de red, puede tener, donde el máximo número de saltos especificables es 255. Al igual que en IPv4 este campo es de 8 bits y es inicializado en 255, decrementándose en 1 cuando pasa por un enrutador. Un límite de estos es de mucha importancia para que no se den ciclos infinitos cuando haya problemas de enrutamiento.
<b>La dirección origen (128 bits)</b>	Para especificar la IPv6 del nodo fuente.
<b>La dirección destino (128 bits)</b>	Para especificar la IPv6 del nodo destino. A diferencia de IPv4, esta no es necesariamente la

	dirección del destino, puede ser una dirección intermedia al destino, de acuerdo a los encabezados extendidos que se use (NEXT HEADER).
--	---

**a.1. Cabeceras de Extensión.-** El uso de un formato flexible de cabeceras de extensión opcionales es una idea innovadora que permite ir añadiendo funcionalidades de forma paulatina. Este diseño aporta gran eficacia y flexibilidad ya que se pueden definir en cualquier momento a medida que se vayan necesitando entre la cabecera fija y la carga útil.

Hasta el momento, existen 8 tipos de cabeceras de extensión, donde la cabecera fija y las de extensiones opcionales incluyen el campo de cabecera siguiente que identifica el tipo de cabeceras de extensión que viene a continuación o el identificador del protocolo de nivel superior. Luego las cabeceras de extensión se van encadenando utilizando el campo de cabecera siguiente que aparece tanto en la cabecera fija como en cada una de las citadas cabeceras de extensión. Como resultado de la secuencia anterior, dichas cabeceras de extensión se tienen que procesar en el mismo orden en el que aparecen en el datagrama. Todas o parte de estas cabeceras de extensión tienen que ubicarse en el datagrama en el orden especificado:

- i. **Cabecera principal:** Tiene el contrario que la cabecera de la versión IPv4 un tamaño fijo de 40 octetos.
- ii. **Cabecera de opciones de salto a salto (Hop-by-Hop):** Transporta información opcional, contiene los datos que deben ser examinado por cada nodo (cualquier sistema con IPv6) a través de la ruta de envío de un paquete. Su código es 0.

- iii. **Cabecera de encaminamiento (Routing):** Se utiliza para que un origen IPv6 indique uno o más nodos intermedios que han de visitar en el camino hacia el destino del paquete. El código que utiliza es 43.
- iv. **Encaminamiento desde la fuente.**
- v. **Cabecera de fragmentación (Fragment):** Hace posible que el origen envíe un paquete más grande de lo que cabría en la MTU (Maximum Transfer Unit - Unidad Máxima De Transferencia) de la ruta. Hay que tener en cuenta que al contrario que en IPv4, en IPv6 la fragmentación de un paquete solo se puede realizar en los nodos de origen. El código empleado en esta cabecera es 44.
- vi. **Cabecera de autenticación (Authentication Header):** Nos sirve para proveer servicios de integridad de datos, autenticación del origen de los datos, antireplay para IP. El código de esta cabecera es 51.
- vii. **Cabecera de encapsulado de seguridad de la carga útil (Encapsulating Security Payload):** Permiten proveer servicios de integridad de datos. El código al que hace referencia esta cabecera es el 50.
- viii. **Cabecera de opciones para el destino (Destination):** Se usa para llevar información opcional que necesita ser examinada solamente por los nodos destino del paquete. La última de las cabeceras utiliza el código 60.

**b) El campo de datos.-** Los datos que transporta el paquete, pueden llegar a 64k de tamaño en el modo normal, o más con la opción "jumbo payload".

### 2.1.2.2. FRAGMENTACIÓN

En IPv6 la fragmentación se realiza sólo en el nodo origen del paquete, al contrario que en IPv4 en donde los routers pueden fragmentar un paquete. En IPv6, las opciones también se salen de la cabecera estándar y son especificadas por el campo "Cabecera Siguiente" (Next Header), similar en funcionalidad en IPv4 al campo Protocolo. Un ejemplo: en IPv4 uno añadiría la opción "ruta fijada desde origen" (Strict Source and Record Routing) a la cabecera IPv4 si quiere forzar una cierta ruta para el paquete, pero en IPv6 uno modificaría el campo "Cabecera Siguiente" indicando que una cabecera de encaminamiento es la siguiente en venir. La cabecera de encaminamiento podrá entonces especificar la información adicional de encaminamiento para el paquete, e indicar que, por ejemplo, la cabecera TCP será la siguiente.

### 2.1.2.3. DIRECCIONAMIENTO IPV6

La característica distintiva más evidente de IPv6 es la utilización de direcciones de mucho mayor tamaño. El tamaño de una dirección en IPv6 es de 128 bits, que es cuatro veces mayor que el de una dirección de IPv4. Una dirección de 128 bits, permiten más de 340 sextillones de direcciones (340,282,366,920,938,463,463,374,607,431,768,211,456) únicas en Internet.

El tamaño relativamente grande de la dirección IPv6 está diseñado para subdividirse en dominios de enrutamiento jerárquicos que reflejen la topología de Internet en la actualidad. La utilización de 128 bits proporciona múltiples niveles de jerarquía y flexibilidad en el diseño del direccionamiento y enrutamiento jerárquicos, que son los elementos de los que carece actualmente la red Internet basada en IPv4.

#### 2.1.2.4. NOTACIÓN PARA LAS DIRECCIONES IPV6

Diferente a IPv4 que usa 4 octetos separados por puntos en notación decimal, IPv6 al tener que denotar una dirección de 128 bits usa 8 campos hexadecimales, de 16 bits cada uno. El uso de hexadecimales en IPv6 nos sirve para una notoria reducción en el tamaño de la dirección, ya que cada byte se puede denotar en 2 hexadecimales. Por ejemplo una dirección en IPv6 podría verse así:

**3FC2:43AB:3240:0000:85E2:0002:2900:00AC**, se usan dos puntos (:) para la delimitación de campos.

A veces las direcciones se pueden tornar un poco confusas por ser tan largas, pero se pueden utilizar convenciones adicionales para su reducción.

- Todos los ceros a la izquierda se pueden eliminar.
- Si uno de los campos tiene solo ceros se puede obviar el campo dejándolo vacío.
- Si hay varios campos vacíos, se eliminan los dos puntos de tal modo que solo queden dos consecutivos.

Por ejemplo:

**Tabla I.3.** Dirección IPv6 válida

<b>2001:0db8:85a3:08d3:1319:8a2e:0370:7334</b>	Es una dirección IPv6 válida.
--	-------------------------------

Si un grupo de cuatro dígitos es nulo (es decir, toma el valor "0000"), puede ser comprimido, ver **Tabla I.4**. Por ejemplo:

**Tabla I.4.** Compresión de un campo nulo

<b>2001:0db8:85a3:0000:1319:8a2e:0370:7344</b>	Es la misma dirección que: <b>2001:0db8:85a3::1319:8a2e:0370:7344</b>
--	--

Siguiendo esta regla, si más de dos grupos consecutivos son nulos, pueden comprimirse como "::". Si la dirección tiene más de una serie de grupos nulos consecutivos la compresión solo en uno de ellos. Ver **Tabla I.5**:



**Tabla I.5.** Equivalencias de direcciones IPv6

<b>2001:0DB8:0000:0000:0000:0000:1428:57ab</b>	Son todas válidas y significan lo mismo
<b>2001:0DB8:0000:0000:0000::1428:57ab</b>	
<b>2001:0DB8:0:0:0:0:1428:57ab</b>	
<b>2001:0DB8:0::0:1428:57ab</b>	
<b>2001:0DB8::1428:57ab</b>	

**Tabla I.6.** Dirección IPv6 inválida

<b>2001::25de::cade</b>	Es inválido porque no queda claro cuantos grupos nulos hay en cada lado.
-------------------------	--

**Tabla I.7.** Omisión de ceros iniciales

Los ceros iniciales en un grupo pueden ser omitidos	<b>2001:0DB8:02de::0e13</b>
	<b>2001:DB8:2de::e13</b>

**Tabla I.8.** Dirección IPv4 camuflada en una dirección IPv6

Si la dirección es una dirección IPv4 camuflada, los últimos 32 bits pueden escribirse en base decimal; así:	<b>::ffff:192.168.89.9</b>
es lo mismo que	<b>::ffff:c0a8:5909</b>
pero no lo mismo que	<b>::192.168.89.9</b> ó <b>::c0a8:5909</b>

El formato **::ffff:1.2.3.4** se denomina dirección IPv4 mapeada, y el formato **::1.2.3.4** dirección IPv4 compatible.

Las direcciones IPv4 pueden ser transformadas fácilmente al formato IPv6. Por ejemplo, si la dirección decimal IPv4 es **135.75.43.52** (en hexadecimal, **0x874B2B34**), puede ser convertida a **0000:0000:0000:0000:0000:0000:874B:2B34** o **::874B:2B34**.

#### 2.1.2.5. TIPOS DE DIRECCIONES

En IPv6 existen direcciones de enlace, privadas, globales; que son llamadas ámbitos (scope) de direcciones. Además lo conocido como unicast, multicast y broadcast, da lugar a unicast, anycast y multicast. El Broadcast ya no existe en IPv6, pues fue reemplazado por el Multicast.

Según el **ámbito**, las direcciones se pueden clasificar, **ver Fig. I.3:**

- **Link Local:** Se usan para direccionar un solo enlace, para utilizarlo en autoconfiguración de direcciones, descubrimiento de vecinos o ante la ausencia de un router.

Se lo representa **FE:80::<ID de interface>/10**

- **Site Local:** Direcciones de Sitio Local se usan en redes que no tienen una conexión a Internet, usadas para direccionar dentro de un sitio local u organización.

Se lo representa **FE:C0::<ID de subred>:<ID de interface>/10**

Donde: **ID de la interface** tiene un tamaño de 64 bits y se las puede configurar: con números aleatorios autogenerador, vía DHCP, manualmente configurada

- **Global:** Este tipo de direcciones tiene un prefijo identificador de grupo, o sea son direcciones validas que se usan para estar presente en Internet.



**Fig. I.3.** Jerarquía tipo de direcciones IPv6

Según el **tipo de direcciones** se pueden clasificar en:

- **UNICAST** Es el tipo general de direcciones. Sirve como identificador de una interfaz. Un paquete dirigido a una dirección unicast, será entregado a una única interfaz en la red.

- **MULTICAST** Identifican a un conjunto de interfaces de la red, de manera que cada paquete es enviado a todos y cada uno de ellos individualmente.
- **ANYCAST** Tipo de direcciones aplicables a un grupo. Identifican a un conjunto o grupo de interfaces de red. El paquete se enviara a cualquier interfaz que forme parte del conjunto. En realidad son direcciones unicast que se encuentran asignadas a varios interfaces. Un paquete IPv6 con una dirección destino anycast es encaminado a uno y sólo uno de los interfaces identificados por la dirección. El paquete será encaminado al interfaz más cercano, de acuerdo con las técnicas de medida de distancia de las estrategias de enrutamiento. Al contrario que con las direcciones Multicast, el paquete sólo se envía a un miembro del grupo que se determina mediante algún protocolo de routing.

#### 2.1.2.6. IDENTIFICACIÓN DE LOS TIPOS DE DIRECCIONES

Los tipos de direcciones IPv6 pueden identificarse tomando en cuenta los primeros bits de cada dirección.

- `::/128` – la dirección con todo ceros se utiliza para indicar la ausencia de dirección, y no se asigna ningún nodo.
- `::1/128` – la dirección de loopback es una dirección que puede usar un nodo para enviarse paquetes a sí mismo (corresponde con 127.0.0.1 de IPv4). No puede asignarse a ninguna interfaz física.
- `::/96` – La dirección IPv4 compatible se usa como un mecanismo de transición en las redes duales IPv4/IPv6.
- `::ffff:0:0/96` – La dirección IPv4 mapeada es usada como un mecanismo de transición en terminales duales.

- fe80::/10 – El prefijo de enlace local (link local) especifica que la dirección sólo es válida en el enlace físico local.
- ff00::/8 – El prefijo de multicast es usado para las direcciones multicast.
- Hay que resaltar que las direcciones de difusión (broadcast) no existen en IPv6, aunque la funcionalidad que prestan puede emularse utilizando la dirección multicast FF01::1.

### 2.1.3. DIFERENCIAS ENTRE IPV4 E IPV6

Dado que se han visto varias deficiencias en el protocolo anterior (IPv4), se han hecho algunos cambios que se citan como diferencias entre estos dos protocolos, algunas resultan sustanciales (seguridad), otros viene a mejorar el desempeño, haciéndolo más idóneo para las nuevas aplicaciones que se tienen hoy en día. A continuación, mencionaremos algunas de ellas, a fin de poder establecer los cambios específicos en el nuevo protocolo.

**Tabla I.9.** Diferencias entre IPv4 e IPv6

	<b>IPv4</b>	<b>IPv6</b>
<b>Direcciones</b>	Las direcciones de origen y destino tienen una longitud de 32 bits (4 bytes).	Las direcciones de origen y destino tienen una longitud de 128 bits (16 bytes).
<b>IPSec</b>	La compatibilidad es opcional.	La compatibilidad es obligatoria.
<b>Identificación del número de paquetes</b>	No existe ninguna identificación de flujo de paquetes para que los enrutadores controlen la QoS en el encabezado IPv4.	Se incluye la identificación del flujo de paquetes para que los enrutadores controlen la QoS en el encabezado IPv6, utilizando el campo Flow Label (etiqueta de flujo).
<b>Fragmentación</b>	La llevan a cabo los enrutadores y el host que realiza el envío.	No la llevan a cabo los enrutadores, sino únicamente el host que realiza el envío.
<b>Encabezado</b>	Incluye una suma de comprobación.	No incluye una suma de comprobación.

<b>Opciones</b>	El encabezado lo incluye.	Todos se trasladan a los encabezados de extensión IPv6.
<b>Marcos de solicitud ARP</b>	El Protocolo de resolución de direcciones (ARP) utiliza los marcos de solicitud ARP de difusión para resolver una dirección IPv4 como una dirección de capa de vínculo.	Los marcos de solicitud ARP se sustituyen por mensajes de solicitud de vecinos de multidifusión.
<b>Administrar la pertenencia a grupos locales de subred</b>	Se utiliza el Protocolo de administración de grupos de Internet (IGMP).	IGMP se sustituye con los mensajes de Descubrimiento de escucha de multidifusión (MLD).
<b>Determinar la dirección IPv4 de la mejor puerta de enlace predeterminada</b>	Se utiliza el Descubrimiento de enrutadores ICMP, y es opcional.	El Descubrimiento de enrutadores ICMP queda sustituido por la Solicitud de enrutadores ICMPv6 y los mensajes de anuncio de enrutador, y es obligatorio.
<b>Direcciones de multidifusión</b>	Se utilizan para enviar tráfico a todos los nodos de una subred.	No hay direcciones de multidifusión IPv6. De forma alternativa, se utiliza una dirección de multidifusión para todos los nodos de ámbito local del vínculo.
<b>Configuración manual</b>	Debe configurarse manualmente o a través de DHCP.	No requiere configuración manual o a través de DHCP.
<b>DNS</b>	Utiliza registros de recurso (A) de dirección de host en el Sistema de nombres de dominio (DNS) para correlacionar nombres de host con direcciones IPv4.	Utiliza registros de recurso (AAA) de dirección de host en el Sistema de nombres de dominio (DNS) para correlacionar nombres de host con direcciones IPv6.

Se ha establecido la diferencia de estos dos protocolos en varios detalles, que principalmente tienen que ver con el funcionamiento.

## 2.1.4. VENTAJAS Y DESVENTAJAS DE IPV6 E IPV4

Cada Protocolo tiene pros y contras, a continuación, haremos una mención a estos parámetros que no podemos pasar por alto.

### 2.1.4.1. IPv4

#### **VENTAJAS:**

- Redes actuales configuradas con este protocolo,
- Tratamiento decimal, fácilmente comprensible para las personas.

#### **DESVENTAJAS:**

- Fue creado hace 20 años, orientado a Datagramas, sin ser pensado para aplicaciones que existen en la actualidad (Servicios multimedia en tiempo real). Los datagramas deben competir con los demás, el tiempo de tránsito en la red es muy variable y sujeto a congestión.
- Provoca ruteo ineficiente, debido a que tiene demasiados campos de cabecera, haciendo variable el tamaño de la misma, lo que hace lento el procesamiento.
- La seguridad no es algo que esté bien cimentado, ya que permite la captura de paquetes, así también el posible camuflaje de hosts, además el sistema IPSec es tan solo una opción.
- Complejidad de coexistencia entre protocolos. Ya que las redes hoy en día trabajan orientadas a protocolos (IP, OSI, IPX, etc.), a veces resulta difícil, tedioso, permitir la comunicación entre redes de diferentes protocolos.
- Inexistencia de políticas de encaminamiento. Algunas empresas del mercado, desearían dirigir sus datagramas a determinadas redes.

## 2.1.4.2. IPV6

### **VENTAJAS:**

- Permite la convivencia con IPv4, lo que hace posible una fácil migración.
- Direcciones unicast, multicast y anycast.
- Formato de cabecera más flexible que en IPv4 para agilizar el encaminamiento.
- Nueva etiqueta de flujo para identificar paquetes de un mismo flujo.
- No se usa checksum.
- La fragmentación y reensamblado se realiza en los nodos finales.
- Nuevas características de seguridad. IPSEC formará parte del estándar.
- Nueva versión de ICMP, que incluye a MLD, el equivalente del IGMP de IPv4.
- Autoconfiguración de los nodos finales, que permite a un equipo aprender automáticamente una dirección IPv6 al conectarse a la red.
- Movilidad incluida en el estándar, que permitirá cambiar de red sin perder la conectividad.

### **DESVENTAJAS:**

- La necesidad de extender un soporte permanente para IPv6 a través de todo Internet y de los dispositivos conectados a ella, requerirá una gran inversión económica.
- Para estar enlazada al universo IPv4 durante la fase de transición, todavía se necesita una dirección IPv4 o algún tipo de NAT (compartición de direcciones IP) en los routers pasarela (IPv6<-->IPv4) que añaden complejidad y que significa que el gran espacio de direcciones prometido por la especificación no podrá ser inmediatamente usado.
- Problemas restantes de arquitectura, como la falta de acuerdo para un soporte adecuado de IPv6 multihoming (conexión de un host o sitio a más de un ISP a la vez).

### 2.1.5. RAZONES DE CONVIVENCIA ENTRE IPV4 E IPV6

Teniendo en cuenta que IPv4 tiene alrededor de 20 años de vigencia, se ha convertido en un aspecto que se lo maneja diariamente, como un aspecto de cultura informática.

Desde las empresas que cuentan con redes pequeñas para el intercambio de archivos, o la publicación de una página web de información, hasta las grandes empresas que cuentan con sistemas informáticos complejos con bases de datos y servicios on-line de atención al cliente, han hecho cierta inversión, mediana o grande, para la implementación de toda la infraestructura informática, y tales equipos manejan IPv4, y no se puede simplemente desechar toda esa implementación para cambiar a una nueva infraestructura para el funcionamiento de IPv6.

Existen sociedades que por falta de recursos, no podrían emigrar sus equipos al nuevo protocolo. Dadas las premisas que se han mencionado, se ha hecho necesaria la convivencia de estos protocolos, ya que la evolución, definitivamente deberá darse paulatinamente.

### 2.1.6. TÉCNICAS DE CONVIVENCIA ENTRE IPV4 E IPV6

Debido a las razones expuestas anteriormente, se promueve la convivencia de los protocolos IPv4 e IPv6, y para lograr este fin se han desarrollado técnicas que permitan lograr tal cometido, a manera general se tiene:

#### 2.1.6.1. PILA DUAL

Esta técnica implementa las pilas de ambos protocolos, IPv4 e IPv6 en cada nodo de la red. Cada nodo de pila dual en la red tendrá dos direcciones de red, una IPv4 y otra IPv6, está descrita en el RFC 2893.

Tiene un enfoque muy sencillo de implementar que requiere que los hosts y los routers soporten ambas versiones de IP y, por tanto, servicios y aplicaciones tanto IPv4 como IPv6. Estos nodos tienen la habilidad de enviar y recibir



paquetes IPv6 e IPv4, pudiendo así interoperar directamente con nodos IPv4 usando paquetes IPv4, y también operar con nodos IPv6 usando paquetes IPv6. Como algo negativo de la pila dual podríamos decir que la topología de red requiere dos tablas de encaminamiento y dos procesos de encaminamiento. Cada nodo en la red necesita tener actualizadas las dos pilas.

En estos momentos, este enfoque de doble pila es un mecanismo fundamental para introducir IPv6 en las arquitecturas IPv4 actuales y se prevé que siga siendo muy utilizado durante el próximo futuro. Su punto flaco es que obliga a que cada máquina retenga una dirección IPv4, cada vez más escasas. Así, a medida que se difunde IPv6, la técnica de doble pila tendrá que ser aplicada allí donde específicamente ayuda al proceso de transición.

Por ejemplo, un servidor de doble pila puede soportar clientes sólo IPv4 convencionales, nuevos clientes sólo IPv6, y por supuesto clientes de doble pila. Para aquellos casos en que haya insuficientes direcciones IPv4 se ha definido una combinación del modelo de conversión y de doble pila de protocolos, conocido como DSTM (Dual Stack Transition Mechanism), ver Fig.

I.4.

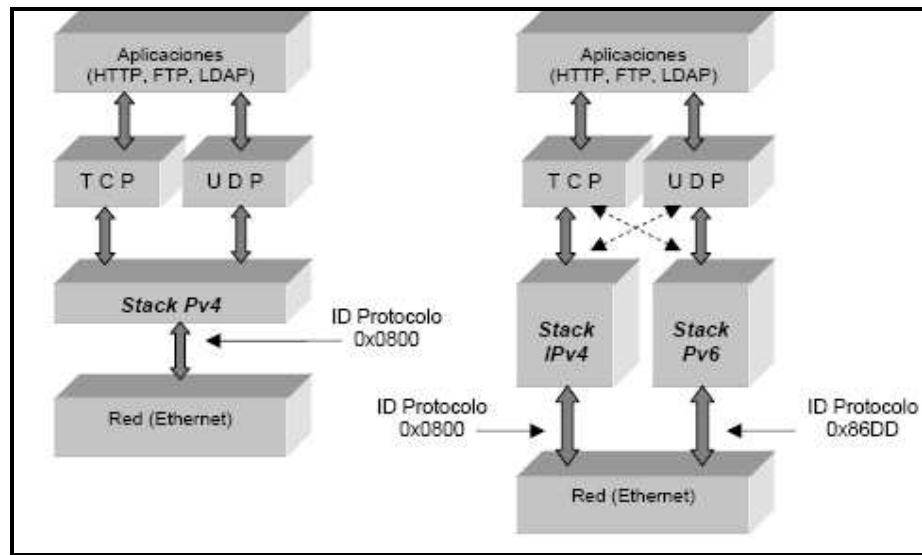


Fig. I.4. Estructura Pila Dual

### 2.1.6.2. TÚNEL

El túnel es un mecanismo en el que un paquete es encapsulado, dentro de otro tipo de paquete. Es decir podemos encapsular paquetes IPv6 dentro de paquetes IPv4. Este mecanismo es importante, ya que hay varios otros que se basan en este.

Los nodos o redes IPv6 que se encuentran separadas por infraestructuras IPv4 pueden construir un enlace virtual, configurando un túnel. Paquetes IPv6 que van hacia un dominio IPv6 serán encapsulados dentro de paquetes IPv4. Los extremos del túnel son dos direcciones IPv4 y dos IPv6. Se pueden utilizar dos tipos de túneles: configurados y automáticos. Los túneles configurados son creados mediante configuración manual. Los túneles automáticos no necesitan configuración manual. Los extremos se determinan automáticamente usando direcciones IPv6 IPv4-compatible, **ver Fig. I.5.**

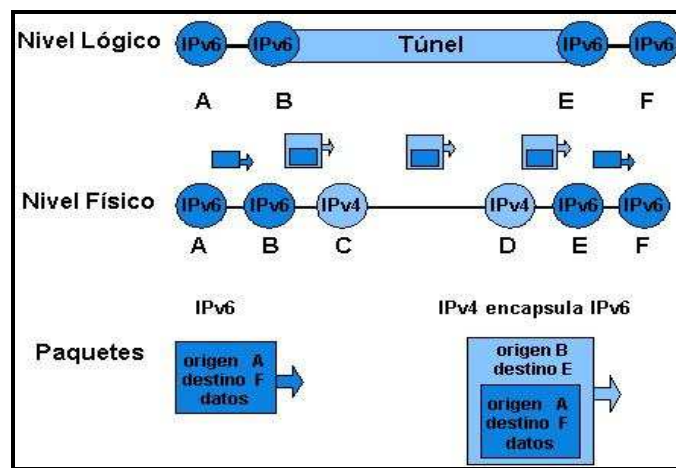


Fig. I.5. Estructura Túnel

Como hemos dicho los datagramas IPv6 se encapsulan sobre datagramas IPv4 para atravesar redes que aun no han sido migradas, de la siguiente manera, **ver Fig. I.6:**

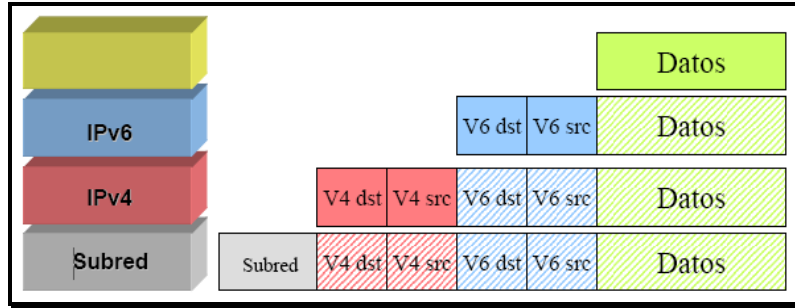


Fig. I.6. Encapsulación Paquetes IPv6 en IPv4

La principal ventaja de éste mecanismo de transición es que solo es necesario tener un Dual Stack en los nodos que servirán como extremos del túnel. Su principal desventaja es el retardo adicional ocasionado por el encapsulado y desencapsulado de paquetes IPv6 en datagramas IPv4, así como el tráfico de un mayor número de paquetes ocasionado por la reducción de espacio para datos en los datagramas IPv4 que contienen dentro paquetes IPv6.

Tunneling puede ser usado en una variedad de formas:

- **Router-to-Router:** Los routers IPv6/IPv4 interconectados con una infraestructura IPv4 pueden pasarse entre sí paquetes IPv6. En este caso el túnel abarca un segmento del trayecto que toma el paquete IPv6. Estos routers pueden ser utilizados para interconectar islas de *hosts* IPv6, por lo que cualquier *host* puede establecer sesiones IPv6 extremo a extremo con otro *host* de la otra isla IPv6.
- **Host-to-Router:** Los host IPv6/IPv4 pueden pasar paquetes IPv6 por un router IPv6/IPv4 intermediario que sea alcanzable por la infraestructura IPv4. Este tipo de túnel abarca el primer segmento del trayecto del paquete. El router puede tener conectividad IPv6 nativa sobre otra interfaz por lo que esta arquitectura permite el establecimiento de sesiones IPv6 extremo a extremo entre cualquier *host* de la isla IPv6 y el *host* aislado a través del enrutador.

- **Host-to-Host:** Los hosts IPv6/IPv4 interconectados con una infraestructura IPv4 pueden pasarse paquetes IPv6 entre sí. En este caso el túnel abarca el recorrido completo que toman los paquetes. Esta arquitectura requiere que ambos *hosts* tengan un *Dual Stack* configurado y solo permite el establecimiento de sesiones IPv6 extremo a extremo entre ellos.
- **Router-to-Host:** Los routers IPv6/IPv4 pueden pasar paquetes IPv6 hasta su host IPv6/IPv4 destinatario (final). Este túnel abarca el último segmento del recorrido.

Las técnicas de tunneling se clasifican según el mecanismo por el cual el nodo de encapsulamiento determina la dirección del nodo al final del túnel. En los primeros dos casos (Router-to-Router y Host-to-Router) el paquete IPv6 es pasado (tunneled) a un router. El endpoint de este tipo de túneles es un router intermediario el cual debe desencapsular el paquete IPv6 y reenviarlo a su destino final. Cuando se envían paquetes a un router, el endpoint del túnel es distinto del destino final del paquete que se está enviando. Así, la dirección del paquete IPv6 que se envía no provee la dirección IPv4 del endpoint del túnel.

Por esto, dicha dirección deberá obtenerse de la información de configuración en el nodo que ejecuta el tunneling. Por lo tanto se usa el termino tunneling configurado (configure tunneling) para describir el tipo de túneles donde el endpoint esta explícitamente configurado.

En los últimos dos casos (Host-to-Host y Router-to-Host) el endpoint del túnel es el nodo al cual el paquete IPv6 esta direccionado. Por lo tanto el endpoint puede ser determinado por la dirección IPv6 de destino del paquete. Si dicha dirección es una dirección IPv6 compatible con IPv4 entonces los últimos 32 bits especifican la dirección del nodo de destino y se puede usar como dirección del endpoint del túnel. De esta forma se evita configurar

explícitamente de la dirección del endpoint. Esta técnica es llamada tunneling automático.

Las dos técnicas de tunneling se diferencian principalmente en cómo se valen para determinar la dirección del endpoint del tunel, **ver Fig. I.7**. La mayor parte de estos mecanismos son lo mismo:

- El nodo de entrada del túnel (nodo de encapsulamiento) crea un paquete IPv4 en el que encapsula el paquete IPv6, y lo transmite encapsulado. El header IPv4 contiene las direcciones fuente y destino y el cuerpo del paquete contiene el header IPv6 seguido inmediatamente por los datos.
- El nodo de salida del túnel (nodo de desencapsulamiento) recibe el paquete encapsulado, elimina el header IPv4, actualiza el header IPv6 y procesa el paquete IPv6 recibido.

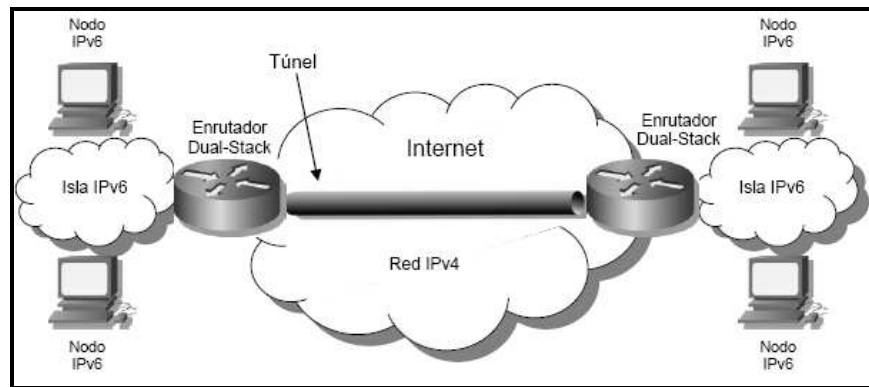


Fig. I.7. Esquema de Túnel.

Para poder configurar un túnel primero es necesario tomar en cuenta los siguientes aspectos:

- Habilitar el protocolo 41: Si se tiene configurado un cortafuegos sobre IPv4, es necesario establecer una regla que permita el acceso y salida al protocolo 41. Como está descrito en el RFC 2893 "*IPv6 Transition Mechanisms*" el número de protocolo asignado a la encapsulación de

paquetes IPv6 en IPv4 es el 41. Este valor es utilizado en el campo "Número de Protocolo" en el encabezado de IPv4 para especificar la encapsulación de un paquete IPv6 en un paquete IPv4.

- Manejo de mensajes de error (ICMPv4): Algunos viejos enrutadores en caso de error solo regresan ocho octetos de datos, sin embargo, los nodos emisores de los paquetes IPv6 necesitan conocer los campos de direcciones IPv6 en el error y cada uno de ellos ocupa 16 octetos.

Traducción de Direcciones de Red (NAT): No es posible establecer túneles IPv6 en IPv4 a través de NAT cuando éste está habilitado en modo traducción dinámica de puerto y redirección de puerto. Por otra parte, es posible establecer dichos túneles si NAT es configurado en modo estático como lo muestra el RFC 2766.

## **TIPOS DE TÚNELES**

### **a. TÚNELES MANUALES**

Es la forma más sencilla de configurar una conexión IPv6 a través de una red IPv4, aunque no es fácil de administrar. La mayoría de hosts doble pila y elementos de red soportan el estándar IPv6 en túneles IPv4, también conocidos como protocolo 41. El principio detrás de la "tunelación" es el encapsular paquetes IPv6 en paquetes IPv4, es decir, empaquetar paquetes dentro de otros paquetes, en realidad es una técnica muy poderosa.

Un túnel manual está compuesto por dos direcciones, un par de IPv4 y un par de IPv6. El par IPv4 es la pareja formada por la dirección de la máquina cliente o router y la dirección del servidor de túnel en el lado proveedor. Las direcciones IPv6 generalmente son proporcionados por las fuentes, ya sea como dos direcciones o como uno de prefijo /64.

Los Túneles Manuales son fáciles de configurar, ya que se encuentran ampliamente disponibles. Sin embargo, no ofrecen ningún tipo de autenticación y función de vigilancia. El mayor inconveniente de utilizar el túnel manual es la participación de las personas cada vez que un túnel se ha creado o cuando es modificado, pues debemos configurar manualmente algunas direcciones.

Este mecanismo es muy útil para conectar dos islas informáticas bien conocidas y muy poco probable que cambien, como pueden ser una sucursal y la oficina principal.

#### **b. TÚNELES AUTOMÁTICOS**

Los mecanismos más populares de túneles automáticos son: el túnel 6to4, Teredo e ISATAP, todos ellos se ejecutan en los sistemas operativos de MICROSOFT. La característica más importante de estos protocolos es que proporcionan una dirección IPv6 o prefijo basado en una dirección IPv4. Actualmente, la dirección IPv4 del nodo está incrustada en su dirección IPv6.

La ventaja de este enfoque es permitir direcciones IPv6 para ser configurada de inmediato y el intercambio de paquetes IPv6 directamente de un servidor a otro utilizando la red IPv4.

#### **TÉCNICAS PARA ESTABLECER TÚNELES**

El IETF definió protocolos y técnicas para establecer túneles entre nodos con dual-stack, entre estas técnicas se encuentran las siguientes:

##### **Túneles 6to4**

En esta técnica los extremos del túnel están determinados por las direcciones globales IPv4 embebidas dentro de direcciones IPv6 *6to4*. Las direcciones IPv6 6to4 están formadas por la combinación de un prefijo de enrutamiento global 2002::<16 y una dirección IPv4 globalmente única.

Los túneles 6to4 pueden ser configurados entre dos enrutadores en la orilla de sus respectivas redes, o entre un enrutador y un host. El único inconveniente de esta técnica para establecer túneles es que solo permiten enviar tráfico IPv6 entre hosts con prefijos de enrutamiento 2002. Para poder comunicarse con nodos con otros prefijos de enrutamiento tales como 2001::/16 y 3FFE::/16 es necesario utilizar un enrutador de reenvío

**Intransite Automatic Tunnel Addressing Protocol (ISATAP)**

Esta técnica permite crear túneles IPv6-in-IPv4 automáticamente dentro de un sitio IPv4. Cada *host* solicita a un enrutador dentro del sitio IPv4 una dirección IPv6 e información de enrutamiento, de esta manera, los paquetes enviados al Internet IPv6 son enrutados a través del enrutador ISATAP y los paquetes destinados hacia otros hosts dentro del mismo sitio son entregados directamente mediante túneles ISATAP. Las direcciones IPv6 se configuran automáticamente mediante el protocolo "descubrimiento de enrutador" ISATAP

**Teredo**

Es un mecanismo que "tunela" IPv6 a través de UDP en una manera que lo permite pasar a través de la mayoría de dispositivos que hacen NAT. Teredo se considera el último mecanismo a usar en el intento de permitir conectividad IPv6 desde una organización la cual los host finales no tengan otro método de comunicaciones capaz.

La operación de Teredo es algo similar a la de 6to4, ya que requiere cierta cantidad de infraestructura, como servidores y relays Teredo (los servidores operan en modo stateless y no es usual que redireccionen paquetes de data; su función principal es facilitar el direccionamiento entre clientes y relays Teredo, así que deben de estar en la red pública de internet IPv4. Los relays son puertas de enlace entre el internet IPv6 y los



clientes Teredo, redireccionan paquetes contactando a servidores Teredo si es necesario y por último deben de estar en el internet IPv4 e IPv6).

**Túnel Broker**

IETF definió este mecanismo para facilitar el desarrollo de túneles configurados sobre redes IPv4 ya que mediante esta técnica no se tiene que configurar manualmente cada extremo del túnel. Tal como está establecido en el RFC 3053 "IPv6 *Tunnel Broker*" el *tunnel broker* es un sistema externo que actúa como un servidor sobre la red IPv4 y recibe peticiones de nodos con dual stack para configurar túneles automáticamente (modelo cliente-servidor).

Estas peticiones son enviadas vía HTTP sobre IPv4 por el nodo que desea configurar dicho túnel. El *tunnel broker* entonces envía de vuelta al cliente información tal como la dirección IPv4 del servidor del túnel, la dirección IPv6 del servidor del túnel, la nueva dirección IPv6 que será asignada a este host con dual stack y las rutas IPv6 default para la configuración del túnel. Algunos *tunnel broker's* ya proporcionan *scripts* de configuración para los hosts clientes. Finalmente el *túnel broker* aplica comandos de manera remota sobre un enrutador con dual stack y que está conectado a un dominio IPv6 para habilitar el túnel configurado.

### 2.1.6.3. TRADUCCIÓN DE DIRECCIONES

Este mecanismo de transición permite la traducción de cabeceras la requieren los nodos sólo IPv6 para interoperar con nodos sólo IPv4. Es una parte opcional de SIT (*Simple Internet Transition*). La llevan a cabo los "routers" IPv6/IPv4 situados en las fronteras entre áreas IPv4-complete e IPv6-complete. El tráfico que cruza la frontera se clasifica de formas.

**Primero**, el tráfico es:

**IPv4:** Trafico de un área IPv4-complete que entra en un área IPv6-complete o

**IPv6:** Trafico de un área IPv6-complete que entra en un área IPv4-complete.

**Segundo**, cada uno de estos tipos se puede describir como:

**Terminal:** Dirigido a un nodo dentro del área o

**Tránsito:** Dirigido a un nodo fuera del área

Los "routers" traductores tienen que seleccionar la forma adecuada de direcciones IP, además de mapear correctamente las direcciones a traducir:

- Las direcciones IPv4 se obtienen tomando los 32 bits de orden inferior de la dirección IP. Si la fuente o el destino son sólo IPv6, la cabecera es intraducible.
- Las direcciones fuente IPv6 se crean añadiendo el prefijo de 96 bits 0:0:0:0:0 a la dirección IPv4.
- Las direcciones destino IPv6 se crean añadiendo el prefijo de 96 bits 0:0:0:0:0:ffff a la dirección IPv4 para generar una dirección IPv6 compatible con IPv4 para el tráfico terminal o el prefijo de 96 bits 0:0:0:0:0 para una dirección IPv6 mapeada a IPv4 para el tráfico de tránsito. En consecuencia, los traductores de cabeceras deben conocer los límites de su área.

Los mecanismos de traducción pueden ser divididos en dos grupos basándonos en si la información de estado está guardada:

**a. Con Estado**, se pueden implementar en los routers y hosts de extremo, el mecanismo más conocido :

**NAT-PT.-** Se coloca como una puerta de enlace entre dos redes, y este se encarga de traducir todas las direcciones de los paquetes que pasan a través de él, es decir entre hosts que son sólo IPv6 e IPv4 respectivamente. En

realidad es NAT (se usa el mecanismo NAT para la asignación de la dirección IPv4) más el Protocolo de Traducción (se usa el mecanismo SIIT). Utiliza el fondo de direcciones de IPv4, y mantiene la tabla de mapeado de la dirección IPv4/IPv6. Se realiza la traducción IPv4/IPv6 y se mantiene el estado mientras dura la sesión.

Este mecanismo tiene ciertas limitaciones:

- No se puede realizar la traducción de paquetes que contengan información sobre direcciones y/o puertos dentro del contenido del paquete. (ej. FTP y DNS).
- No se puede hacer mapeo dinámico de IPv4 a IPv6.
- El usuario o aplicación IPv4 debe ser consciente de si con quien se quiere comunicar es IPv6 o no.
- Pérdida de información intrínseca.

**b. Sin Estado**, para la conversión dentro de un host

**Bump-in-the-Stack.-** Es un caso particular del NAT-PT. A un nodo IPv4 se le agregan tres módulos a su pila para cuando necesite comunicarse con nodos IPv6, estos son unas extensiones al resolvidor de nombres, un mapeador de direcciones y un traductor. La idea es que cuando una máquina IPv4 necesite comunicarse con un nodo IPv6, a su dirección IPv6 se le asigna una dirección IPv4 de un rango de direcciones que tiene la máquina. La traducción completa del paquete se hace de acuerdo a SIIT.

BIS intercepta los paquetes IP al salir del nivel IP de la máquina, antes de enviarlos a la tarjeta de red, y realiza la traducción entre IPv4 e IPv6 según los paquetes sean entrantes o salientes.

**Bump-in-the-API.-** Sigue un funcionamiento análogo a BIS, salvo que la traducción de paquetes se realiza antes de construir el paquete, en la propia

interfaz de programación de aplicaciones. Por tanto, BIA es un mecanismo de transición más ligero ya que no necesita realizar una traducción del paquete, sino que construye el paquete IPv6 a partir de las funciones de la API IPv4,

BIA permite que las aplicaciones IPv4 sigan funcionando normalmente sobre una red IPv6, sin que haya que modificar el código fuente, ni recompilarlo. Dado que una aplicación IPv4 sólo es capaz de manejar direcciones IPv4, el mecanismo BIA proporcionará direcciones ficticias IPv4 a la aplicación para que sea transparente el hecho de que los nodos remotos son sólo direccionables mediante IPv6. BIA se encarga de mantener la correspondencia entre las direcciones IPv4 ficticias y las direcciones IPv6 reales.

## 2.2. LINUX CENTOS 5.0

Los sistemas GNU/Linux han llegado a un grado de madurez importante, que los hacen válidos para integrarlos en cualquier ambiente de trabajo, ya sea desde el escritorio del PC personal, hasta el servidor de una gran empresa. Los sistemas GNU/Linux ya no son una novedad, cuentan con una amplia variedad de usuarios y de ámbitos de trabajo donde son utilizados.

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

### 2.2.1. CARACTERÍSTICAS GENERALES

- CentOS (acrónimo de **C**ommunity **ENT**erprise **O**perating **S**ystem) es un clon a nivel binario de la distribución Red Hat Enterprise Linux, compilado por voluntarios a partir del código fuente liberado por Red Hat, empresa desarrolladora de RHEL.

CentOS es una distribución basada en Red Hat™ Enterprise Linux y está enfocada para el uso en servidores en producción.

- Kernel: 2.6.18
- Plataformas:
  - ✓ **i386:** Esta arquitectura soporta procesadores: AMD (K6, K7, Thunderbird, Athlon, Athlon XP, Sempron), Pentium (Classic, Pro, II, III, 4, Celeron, M, Xeon), VIA (C3, Eden, Luke, C7).
  - ✓ **x86\_64:** soporta procesadores: AMD 64 (Athlon 64, Opteron) and Intel Pentium (Xeon EM64T).
- Hardware recomendado:
  - ✓ Memoria RAM: 64 MB (mínimo).
  - ✓ Espacio en Disco Duro: 512 MB (mínimo) - 2 GB (recomendado).

CentOS 5.0 contiene muchos cambios respecto a versiones anteriores:

- CentOS 5.0 ha sido completamente construido usando un nuevo sistema de construcción de paquetes y chequeo de bibliotecas para confirmar la compatibilidad binaria al 100% con el proveedor.
- CentOS 5, ofrece nuevas actualizaciones que incluye:
  - ✓ La actualización de los siguientes paquetes software: Squid 2.6, Apache-2.2, php-5.1.6, kernel-2.6.18, Gnome-2.16, KDE-3.5, OpenOffice.org-2.0, Evolution-2.8, Firefox-1.5, Thunderbird-1.5, MySQL-5.0, PostgreSQL-8.1.
  - ✓ Un escritorio mejorado con soporte a compiz y AIGLX
- Virtualidad provista por Xen Hypervisor, con un administrador de máquinas virtuales y libvirt.
- Sabayon para simplificar la configuración de de perfiles de usuario.

### 2.2.2. SOPORTE AL PROTOCOLO IPV6

En Linux, IPv6 se implementa como un módulo del kernel. Así, se ha establecido el soporte a IPv6 desde las distribuciones con el kernel 2.2.x y 2.4.x y normalmente el

módulo IPv6 ya está instalado. Debido a este cambio en el módulo, algunas de las aplicaciones de este sistema operativo, también han hecho las adecuaciones necesarias para dar soporte a este nuevo protocolo IPv6.

Algunas de las aplicaciones que se han actualizado son: Apache, DNS, FTP, E-mail, Telnet, SSH, Firewall. Lo que ha hecho que Linux, en sus diferentes distribuciones, cuente desde hace algún tiempo ya, con el soporte necesario para poder manejar el tráfico por el nuevo protocolo IPv6.

### 2.2.3. INSTALACIÓN MÍNIMA DE COMPONENTES

Lo recomendado, sobre todo si se trata de un servidor, es realizar una instalación con el mínimo de paquetes, desactivando todas las casillas para todos los grupos de paquetes, en la **Fig. I.8** se aprecia la pantalla de instalación para CentOS 5.



**Fig. I.8.** Pantalla inicial instalación CentOS 5

El objeto de esto es solo instalar lo mínimo necesario para el funcionamiento del sistema operativo, y permitir instalar posteriormente solo aquello que realmente se requiera de acuerdo a la finalidad productiva que tendrá el sistema.

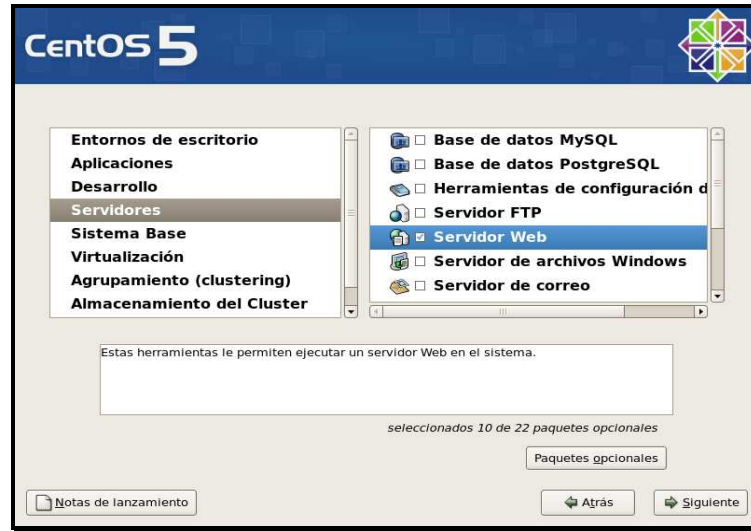


Fig. I.9. Selección Componentes a instalar

Como se desea tener un equipo dedicado para administrar el tráfico de red, entre los componentes requeridos para este fin, debemos instalar: kernel, Servidor Web para hacer uso de APACHE y Servidor de Red para utilizar IPTABLES, IP6TABLES y SQUID, ver Fig. I.9. Los cuales serán administrables remotamente mediante un entorno Web.

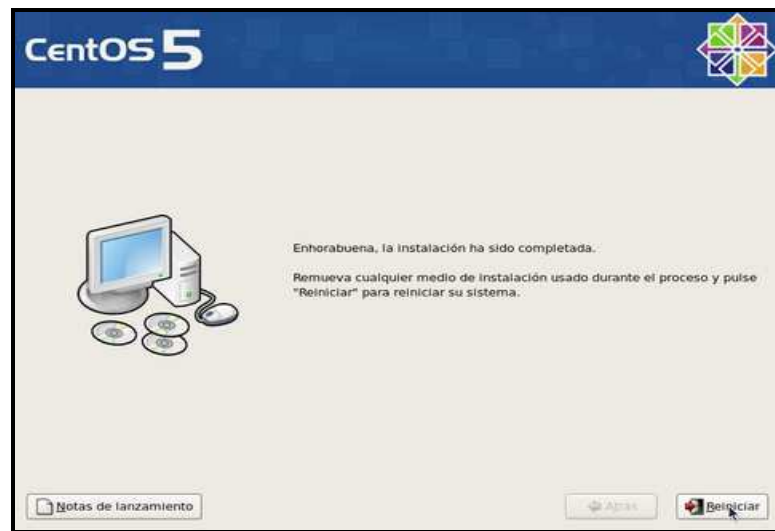


Fig. I.10. Finalización instalación CentOS 5

Una vez instalados los componentes necesarios, se obtiene un equipo dedicado exclusivamente a la administración del tráfico de red, como se ve en la Fig. I.10.

## 2.3. COMPONENTES DE SEGURIDAD

Como se ha mencionado, las aplicaciones con las que cuenta Linux distribución CentOS 5, manejan ya IPv6. De todas las aplicaciones que existen, el interés se centra en las aplicaciones de seguridad del sistema, como son: El Firewall (IPTABLES, IP6TABLES) y el Proxy (SQUID).

### 2.3.1. FIREWALL

Un firewall es la combinación de diferentes componentes: dispositivos físicos (hardware), programas (software) y actividades de administración, que, en conjunto, permitirán aplicar una política de seguridad de una red. El objetivo es protegerla de cualquier acción hostil proveniente de un host externo a la red, **ver Fig. I.11**.

Es apropiado referirse como “firewall” al conjunto de estrategias y políticas de seguridad y como “sistema firewall” a los elementos de hardware y software utilizados en la implementación de esas políticas, que es capaz de manejar las conexiones que entran y salen de una red.

***Función principal:*** Generalmente un firewall es utilizado para hacer de intermediario entre una red de una organización e Internet u otra red no confiable.

Estos mecanismos de control actúan sobre los medios de comunicación entre las dos redes, en particular, sobre la familia de protocolos utilizada para la comunicación de sistemas remotos. La más comúnmente usada es TCP/IP ya que dispone de amplios desarrollos de mecanismos estándares para su uso en varios aspectos, incluyendo en seguridad.

La protección que provee un firewall es de diferentes tipos:

- Bloquea tráfico no deseado.
- Redirecciona tráfico de entrada a sistemas internos de más confianza.
- Oculta sistemas vulnerables, que pueden ser fácilmente asegurados, de Internet.



- Puede registrar el tráfico desde y hacia la red privada.
- Puede ocultar información como ser nombres de sistemas, topología de la red, tipos de dispositivos de red, e identificadores de usuarios internos, de Internet.
- Puede proveer autenticación más robusta que las aplicaciones estándares.

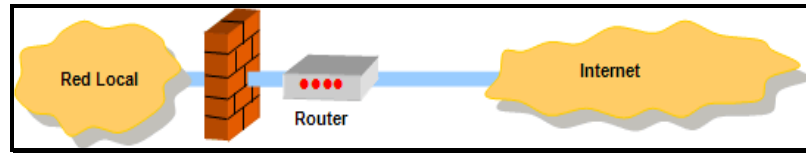


Fig. I.11. Esquema Firewall

## FUNDAMENTOS PARA IMPLEMENTAR UN FIREWALL

La seguridad de la información es pensada a menudo como un proceso y no como un producto. Sin embargo, las implementaciones de seguridad estándar usualmente emplean alguna forma de mecanismo dedicado para controlar los privilegios de acceso y restringir los recursos de la red a los usuarios autorizados, identificables y localizables. Junto a las soluciones que en el ámbito de la seguridad se han desarrollado, los cortafuegos o firewalls, que es uno de los componentes principales de la implementación de seguridad.

### Ventajas

- Personalizable a través de la inclusión o eliminación de reglas.
- No requiere ninguna personalización particular del lado del cliente, pues toda la actividad de la red es filtrada al nivel del servidor en vez de a nivel de la aplicación.
- Puesto que los paquetes no son transmitidos a través del proxy, el rendimiento de la red es más rápido debido a la conexión directa desde el cliente al host remoto.

### Desventajas

- No puede filtrar paquetes por contenido como los cortafuegos Proxy.

- Procesa los paquetes en la capa del protocolo pero no puede filtrar los paquetes en la capa de la aplicación.
- Las arquitecturas de red complejas pueden hacer el establecimiento de reglas de filtrado difíciles, especialmente si están usando enmascaramiento de IP o subredes locales y redes DMZ.

Teniendo en cuenta las ventajas y desventajas de un firewall con IPTABLES, ahora se puede proceder a habilitar este servicio y realizar las configuraciones pertinentes, y antes de ello se debe analizar estos tres puntos:

- Que para instalar un firewall, necesitamos un servidor con 2 tarjetas de red (Red Outside, Red Inside). O instalar un servidor con 3 tarjetas (Red DMZ)
- Determinar la Filosofía de la organización, se abre el tráfico que se necesita y se cierra todo los demás.
- Tráfico de entrada y salida mediante puertos TCP, UPD, ICMP.

Luego de haber analizado los tres puntos mencionados, se procede a realizar las configuraciones en los archivos correspondientes a los firewalls tanto para IPv4 como para IPv6.

### 2.3.2. PROXY

Es un programa o dispositivo que realiza una acción en representación de otro. La finalidad más habitual es la del servidor proxy, es un sistema de software que permite la conexión de una LAN entera al exterior con sólo una dirección IP de salida, es decir, si se monta en el servidor principal de la red un modem, tarjeta de Red, adaptador RDSI, etc, y se instala el Proxy (Configurando también las aplicaciones cliente en los terminales), se tendrá acceso al exterior de todos y cada uno de los terminales con una sola cuenta de acceso a internet.

Aunque existen también los proxy basados en hardware, en éstos lo único que se hace es montar muchos discos duros de gran capacidad (Esto sí es físico) y del orden de 128 Megas de Ram (Variable según necesidades, como todo claro). Pues

bien, éstas unidades solo tienen una función, que es la de almacenar datos en zonas intermedias o en sitios con gran flujo de peticiones para agilizar la transmisión de datos. Vale mencionar que el proxy actúa a nivel de la capa de aplicación para HTTP, HTTPS y FTP.

**Función Principal:** El Proxy es una aplicación que le permite a un cliente salir a internet utilizando un espacio en disco o caché que almacena las páginas visitadas por los usuarios de manera tal que si alguien quiere navegar a un sitio, no tiene que hacer la petición hasta el sitio sino que los ubica en el disco duro. Es decir que permite un acceso más rápido a los Web site con frecuencia usados. También puede ejecutar peticiones DNS bastante más rápido de lo que puede hacerlo la mayoría del software cliente y para controlar el acceso a sitios web (utilizando paquetes como squidGuard).

#### **FUNDAMENTOS PARA IMPLEMENTAR UN PROXY**

Si se hace notorio el acceso recurrente a la misma página desde una red, entonces sería ideal poder implementar un proxy que permita el acceso rápido a la página.

Para lograr implementar un proxy, se necesita un servidor DHCP o configurar las máquinas de los clientes con IPs privadas. Y en los navegadores de los clientes simplemente se especifica el nombre del proxy.

Vale acotar que Linux CentOS 5 incluye la versión 2.6 del archivo SQUID que cumplirá las funciones de PROXY. Esta versión sólo da soporte al protocolo IPv4, se están realizando investigaciones para la versión 3 de SQUID que incorporará soporte para el protocolo IPv6 a futuro.

## **2.4. CONFIGURACIONES**

Para que se pueda obtener un desempeño óptimo del sistema, se procede a realizar las configuraciones pertinentes, tanto de los componentes de seguridad (FIREWALL y

PROXY), como de los servicios que se necesitaran para que la aplicación opere correctamente.

## 2.4.1. CONFIGURACIÓN FIREWALL

Para nuestro estudio, Linux CentOS 5 provee soporte FIREWALL para los protocolos IPv4 e IPv6, por ello analizaremos los componentes IPTABLES e IP6TABLES, ya que cada uno va enfocado a su respectivo protocolo IPv4 e IPv6 respectivamente.

### IPTABLES

Proporciona una solución robusta y flexible para implementar firewalls. NetFilter es el nombre del proyecto e iptables el nombre del software de filtrado de paquetes en linux que viene integrado en el kernel 2.4.x y sus sucesores. Además implementa NAT (Network Address Traslation) y NAPT (Network Address and Port Traslation). Es el sucesor de ipchains e ipfwadmin.

El poder y flexibilidad de Netfilter es implementado a través de la interfaz de iptables. Esta herramienta de línea de comandos es similar en sintaxis a su predecesor, ipchains; sin embargo, iptables utiliza el subsistema Netfilter para mejorar la conexión de la red, inspección y procesamiento; mientras que ipchains usa conjuntos de reglas intrincados para filtrar rutas de fuentes y destino, así como también puertos de conexión o ambos. Iptables presenta funcionalidades como: registro avanzado, acciones previas y posteriores al enrutamiento, traducción de direcciones de red y reenvío de puertos, todo en una interfaz de línea de comandos, ver Fig. I.12.

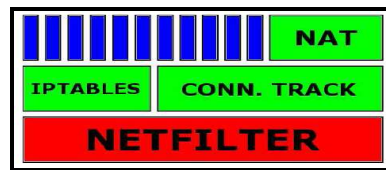


Fig. I.12. Componentes NetFilter

Con este software, los administradores de redes buscan lo siguiente:

- Control
- Seguridad
- Vigilancia

Iptables es el software ideal para cumplir los tres puntos anteriores permitiéndonos convertir computadoras obsoletas en potentes auxiliares de seguridad.

#### **Características Principales:**

- Filtrado de paquetes sin estado IPv4 e IPv6
- Filtrado de paquetes con estado (IPv4)
- Traducción de puertos y direcciones de todos tipos (NAT y NAPT)
- Infraestructura flexible y extensible
- Extensiones tripartitas de múltiple capa para distintos API's
- Gran número de plugins o módulos mantenidos en el repositorio "patch-o-matic"

#### **Filtro de paquetes**

El filtro de paquetes es un software que examina la cabecera de los mismos según van pasando y decide el destino por completo. Se puede elegir descartarlo o aceptarlo o bien cosas más complicadas.

En Linux dicho procedimiento está enganchado al núcleo a través de módulos o componentes estáticos, se podría realizar algunas cosas curiosas, pero el principio general es revisar las cabeceras para decidir la suerte que tendrá el paquete.

#### **Paso de los paquetes por los filtros**

El núcleo tiene contenido tres listas de reglas en la tabla de filtros llamadas cadenas corta fuegos, que son:

- **INPUT:** Si en la decisión, la máquina de destino es el firewall.
- **OUTPUT:** Si en la decisión, corresponde a una petición.
- **FORWARD:** si en la decisión, la máquina destino es otra.

### Proceso local

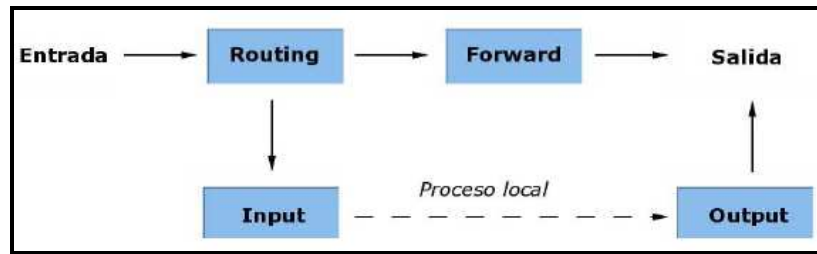


Fig. I.13. Tratamiento de paquetes

Cuando un paquete llega al firewall, primero pasa unos filtros, **ver Fig. I.13**. En primer lugar pasa al filtro de PREROUTING donde se puede manipular el paquete modificando sus datos de destino, por ejemplo redirigir a otra máquina o a otro puerto. Esto se conoce como DNAT (*destination network address translation*).

Una vez que el paquete de entrada está preparado se comprueba si va dirigido al propio ordenador en cuyo caso pasa al filtro de entrada (INPUT), o bien si va dirigido a otra máquina, en cuyo caso se dirige al filtro de reenvío (FORWARD).

Si el paquete iba destinado a la máquina local y ha pasado el filtro de entrada se le entrega al proceso local que lo solicitó. Si un proceso local genera un paquete que tiene que enviar a la red primero pasa por el filtro de salida (OUTPUT). El filtro de salida podría manipular el paquete modificando el destino.

Si el paquete ha pasado el filtro de reenvío (FORWARD) pasaría el filtro de POSTROUTING. Igualmente si el paquete local pasa el filtro de salida también pasa al filtro de POSTROUTING). En el filtro de POSTROUTING se pueden manipular los datos de origen de un paquete. En esta fase podemos realizar SNAT (*source network address translation*), es decir, manipular los datos de origen del paquete. Cada vez que se tenga que configurar un filtro se tiene que tener muy en cuenta el esquema descrito.

### Creación de las Reglas de Iptables

IPTABLES tiene una serie de comandos que permite manipular (insertar, eliminar) las reglas del conjunto predeterminado. En general, la estructura de un comando de iptables es el siguiente:

```
iptables [-t <table-name>] <command> <chain-name>  
<parameter-1> \<option-1> <parameter-n> <option-n>
```

Donde:

- *<table-name>*: permite al usuario seleccionar una tabla diferente de la tabla filter por defecto que se usa con el comando.
- *<command>*: es el centro del comando, dictando cuál es la acción específica a realizar, como pueda ser añadir o borrar una regla de una cadena particular, que es lo que se especifica en la opción *<chain-name>*
- *<parameter-n> <option-n>*: Los pares de parámetros y opciones que realmente definen la forma en la que la regla funcionará y qué pasará cuando un paquete cumpla una regla.

**Comandos.-** Para la creación de una regla podemos usar varios comandos, **ver**

**Tabla I.10:**

**Tabla I.10.** Comandos para una regla IPTABLES.

Comando	Descripción
-A	Añade la regla iptables al final de la cadena especificada. Este es el comando utilizado para simplemente añadir una regla cuando el orden de las reglas en la cadena no importa.
-D	Borra una regla de una cadena en particular. Puede también teclear la regla entera e iptables borrará la regla en la cadena que corresponda.
-F	Libera la cadena seleccionada, que borra cada regla de la cadena. Si no se especifica ninguna cadena, este comando libera cada regla de cada cadena.
-L	Lista todas las reglas de la cadena especificada tras el comando. Para ver una lista de todas las cadenas en la tabla filter por defecto.

<b>-N</b>	Crea una nueva cadena con un nombre especificado por el usuario.
<b>-P</b>	Configura la política por defecto para una cadena en particular de tal forma que cuando los paquetes atraviesen la cadena completa sin cumplir ninguna regla, serán enviados a un objetivo en particular, como puedan ser ACCEPT o DROP.
<b>-X</b>	Borra una cadena especificada por el usuario. No se permite borrar ninguna de las cadenas predefinidas para cualquier tabla.
<b>-Z</b>	Pone ceros en los contadores de byte y de paquete en todas las cadenas de una tabla en particular

**Parámetros.-** Para la creación de reglas se utiliza parámetros para una mejor estructura, algunos de ellos se los describe en la **Tabla I.11**:

**Tabla I.11.** Parámetros para una regla de iptables.

<b>Parámetro</b>	<b>Descripción</b>
<b>-d</b>	Configura el nombre de la máquina destino, dirección IP o red de un paquete que cumplirá la regla. Cuando se especifique una red, puede utilizar dos métodos diferentes para describir las máscaras de red, como 192.168.0.0/255.255.255.0 o 192.168.0.0/24.
<b>-i</b>	Configura las interfaces de entrada de red, como eth0 o ppp0, para ser usadas por una regla.
<b>-j</b>	Dice a iptables que salte a un objetivo en particular cuando un paquete cumple una regla en particular. Los objetivos válidos que se usarán tras la opción -j incluyen opciones estándar, ACCEPT, DROP, así como opciones extendidas que están disponibles a través de módulos que se cargan por defectos con el paquete RPM de IPTABLES de Red Hat Linux, como LOG y REJECT. En lugar de especificar la acción objetivo, puede también dirigir un paquete que cumpla la regla hacia una cadena definida por el usuario fuera de la cadena actual. Esto le permitirá aplicar otras reglas contra este paquete, y filtrarlo mejor con respecto a otros criterios.
<b>-o</b>	Configura la interfaz de red de salida para una regla en particular, y sólo puede ser usada con las cadenas OUTPUT y FORWARD en la tabla filter y la cadena POSTROUTING en las tablas nat y mangle. Estas opciones de los parámetros son los mismos que para los de la interfaz de red de entrada (opción -i).



<b>-p</b>	Configura el protocolo IP para la regla, que puede ser icmp, tcp, udp, o all (todos), para usar cualquier protocolo. Además, se pueden usar otros protocolos menos usados de los que aparecen en /etc/protocols. Si esta opción se omite al crear una regla, la opción all es la que se selecciona por defecto.
<b>-s</b>	Configura el origen de un paquete en particular usando la misma sintaxis que en el parámetro de destino (opción -d).

**Opciones de Paquetes icmp.-** Los paquetes que usan el protocolo de control de mensajes de internet (Internet Control Message Protocol, ICMP) pueden ser seleccionados usando la siguiente opción cuando se especifique -p icmp:

**Tabla I.12.** Opciones para paquetes ICMP en una regla de iptables.

Opción	Descripción
<b>--icmp-type</b>	Selecciona el nombre o el número del tipo ICMP que concuerde con la regla. Se puede obtener una lista de nombres válidos ICMP tecleando el comando iptables -p icmp -h.

**Opciones de Paquetes tcp.-** Estas opciones de identificación están disponibles en el protocolo TCP (opción -p tcp), **ver Tabla I.13:**

**Tabla I.13.** Opciones para paquetes TCP en una regla de iptables

Opción	Descripción
<b>--dport</b>	Configura el puerto de destino para el paquete. Puede utilizar o bien un nombre de servicio de red (como www o smtp), un número de puerto, o bien un rango de números de puertos para configurar esta opción. Para especificar un rango de números de puertos separe los dos números de puertos con dos puntos (:), como en -p tcp --dport 3000:3200. El rango válido más grande es 0:65535.
<b>--sport</b>	Configura el puerto de origen del paquete, usando las mismas opciones que --dport. También puede usar --source-port para especificar esta opción.
<b>--syn</b>	Provoca que todos los paquetes designados de TCP, comúnmente llamados paquetes SYN, cumplan esta regla. Cualquier paquete que esté llevando un payload de datos no será tocado. Si se sitúa un punto de exclamación (!) como flag tras la opción --syn se provoca que todos los paquetes no-SYN sean seleccionados.

<b>--tcp-flags</b>	Permite que los paquetes TCP con conjuntos de bits específicos, o flags, sean seleccionados para una regla. La opción de selección --tcp-flags acepta dos parámetros, que son los flags para los diferentes bits ordenados en una lista separada por comas. El primer parámetro es la máscara, que configura los flags que serán examinados en el paquete. El segundo parámetro se refiere a los flags que se deben configurar en el paquete para ser seleccionado. Los flags posibles son ACK, FIN, PSH, RST, SYN, y URG.
<b>--tcp-option</b>	Intenta seleccionar con opciones específicas de TCP que pueden estar activas en un paquete en particular.

**Opciones de Paquetes UDP.-** Estas opciones de selección están disponibles para el protocolo UDP (-p udp), ver **Tabla I.14**:

**Tabla I.14.** Opciones para paquetes UDP en una regla de Iptables

<b>Opción</b>	<b>Descripción</b>
<b>--dport</b>	Especifica el puerto destino del paquete UDP usando el nombre del servicio, el número del puerto, o un rango de puertos. La opción de selección de paquetes --destination-port se puede utilizar en lugar de --dport.
<b>--sport</b>	Especifica el puerto origen del paquete UDP usando el nombre del servicio número de puerto, o rango de puertos. La opción --source-port puede ser usada en lugar de --sport.

**Opciones del Objetivo.-** Una vez que un paquete cumple una regla en particular, la regla puede dirigir el paquete a un número de objetivos (destinos) diferentes que decidirán cuál será su destino y, posiblemente, las acciones adicionales que se tomarán, como el guardar un registro de lo que está ocurriendo.

Adicionalmente, cada cadena tiene un objetivo por defecto que será el que se utilice si ninguna de las reglas disponibles en dicha cadena se puede aplicar a dicho paquete, o si ninguna de las reglas que se aplican al mismo especifica un

objetivo concreto. Existen pocos objetivos estándar disponibles para decidir qué ocurrirá con el paquete, en la **Tabla I.15** se los describe:

**Tabla I.15.** Opciones para objetivos en una regla de iptables.

Opción	Descripción
<b>ACCEPT</b>	Permite que el paquete se mueva hacia su destino (o hacia otra cadena, si no ha sido configurado ningún destino ha sido configurado para seguir a esta cadena).
<b>DROP</b>	Deja caer el paquete al suelo. El sistema que envió el paquete no es informado del fallo. El paquete simplemente se borra de la regla que está verificando la cadena y se descarta.
<b>LOG</b>	<p>Guarda un registro de todos los paquetes que cumplen esta regla. Como estos paquetes son monitorizados por el kernel, el fichero <code>/etc/syslog.conf</code> determina dónde se escribirán esas entradas en el fichero de registro (log). Por defecto, se sitúan en el fichero <code>/var/log/messages</code>.</p> <p>Se pueden usar varias opciones tras el objetivo LOG para especificar la manera en la que tendrá lugar el registro:</p>
	<p><b>--log-level</b></p> <p>Configura un nivel de prioridad al evento de registro del sistema. Se puede encontrar una lista de los eventos del sistema en la página del manual de <code>syslog.conf</code>, y sus nombres se pueden usar como opciones tras la opción <code>--log-level</code>.</p>
	<p><b>--log-ip-options</b></p> <p>Cualquier opción en la cabecera de un paquete IP se guarda en el registro.</p> <p><b>--log-prefix</b></p> <p>Pone una cadena de texto antes de la línea de registro cuando ésta sea escrita. Acepta hasta 29 caracteres tras la opción <code>--log-prefix</code>. Esto puede ser útil para escribir filtros del registro del sistema para ser usados conjuntamente junto con el registro de paquetes.</p>
	<p><b>--log-tcp-options</b></p> <p>Cualquier opción en la cabecera de un paquete TCP se guarda en el registro.</p> <p><b>--log-tcp-sequence</b></p> <p>Escribe le número de secuencia TCP del paquete en el registro del sistema.</p>

<b>REJECT</b>	Envía un paquete de error de vuelta al sistema que envió el paquete, y lo deja caer (DROP). Este objetivo puede ser útil si queremos notificar al sistema que envió el paquete del problema. El objetivo REJECT acepta una opción --reject-with <type> para proporcionar más detalles para ser enviados junto con el paquete de error. El mensaje port-unreachable es el error <type> que se envía por defecto cuando no se utiliza junto con otra opción.
---------------	---

### Orden de las Reglas

El orden de las reglas de un firewall define su comportamiento. El filtro va comparando el paquete con cada una de las reglas hasta que se encuentra una que verifica y en ese caso se lleva a cabo lo que indique esa regla (aceptar o denegar); una vez realizada la acción no se comprueban más reglas. Si ponemos reglas muy permisivas entre las primeras del firewall, puede que las siguientes no se apliquen y no sirvan de nada.

### Política Predeterminada

También se puede dar el caso de que un paquete no haya verificado ninguno de los filtros una vez que se los ha comprobado todos. En este caso no se sabe si se tiene que aceptar o rechazar este paquete. Para resolver esta situación iptables dispone de una política predeterminada que permite indicar qué hacer con los paquetes que no hayan verificado ninguna regla. La política predeterminada será algunas de las acciones que hemos definido y se define con la opción "-P".

Por ejemplo:

**iptables -P INPUT -j ACCEPT**

Definiría como aceptar la política predeterminada del filtro INPUT.

La política predeterminada define un esquema distinto de cortafuegos. Si la política predeterminada es aceptar entonces se tendrá que denegar todo aquello que no interese. Si la política predeterminada es denegar entonces se tiene que permitir todo aquello que sea de interés.

Cada uno de los modelos de cortafuegos tiene sus ventajas e inconvenientes. En el primer caso, con la política predeterminada de aceptar es mucho más fácil la gestión del firewall. En este caso es útil cuando se sabe claramente qué puertos se quiere proteger y el resto no importa y se acepta. El problema que plantea es no poder controlar qué se tiene abierto o que en un momento dado se instale un software nuevo, un troyano por ejemplo, que abra un puerto determinado, o que no se sepa que determinados paquetes ICMP son peligrosos. Si la política predeterminada es ACEPTAR y no se protege explícitamente el sistema puede ser inseguro.

Cuando la política predeterminada es DENEGAR todo lo que no se acepte explícitamente será denegado por lo que el sistema puede fallar por despiste o desconocimiento en lo que se está permitiendo. Es más complicado establecer este tipo de cortafuegos, hay que tener muy claro cómo funciona el sistema y qué es lo que se tiene que aceptar sin caer en la tentación de introducir reglas muy permisivas. Esta configuración de cortafuegos es la recomendada, aunque no es aconsejable usarla si no se domina mínimamente el sistema.

### **IP6TABLES**

IP6TABLES se utiliza para configurar, mantener y examinar las tablas de las reglas de filtrado para paquetes IPv6 en el núcleo de Linux. Varias tablas pueden ser definidas. Cada tabla contiene un número de cadenas por defecto y puede también contener cadenas definidas por el usuario.

Cada cadena es una lista de reglas, que pueden ser emparejadas con un grupo de paquetes. Cada regla especifica lo que se debe hacer con los paquetes a los que se emparejó. Estos paquetes son llamados "objetivo", lo que puede ser también un salto a una cadena definida por el usuario, en la misma tabla.

Contrariamente al firewalling de Linux para IPv4, que contiene mecanismos de connection tracking (seguimiento de conexión), llegando incluso a perseguir las conexiones a nivel 5 (permitiendo el uso de DCC en IRC, o de FTP activo, o de Amanda, si se desea), IP6TABLES son un reglas sin estado básico (lo que se conoce como stateless packet filtering, filtrado de paquetes sin estado).

### **Desventaja**

Eso quiere decir que se puede aprovechar las reglas de `-m state --state ESTABLISHED` y `-m state --state RELATED` y se tiene que implementar un seguimiento de conexión muy básico con los flags de control de TCP. Además la tabla nat aún no es compatible. Esto significa que todavía no es posible realizar tareas de traducción de direcciones de red IPv6, tales como enmáscarado y reenvío de puertos.

### **Creando Reglas**

Si está instalado el paquete IPTABLES-IPV6, se puede filtrar la próxima generación del protocolo de Internet IPv6. El comando utilizado para manipular el filtrado de red de IPv6 es *IP6TABLES*. La mayoría de las directivas para este comando son idénticas a aquellas usadas por IPTABLES.

Las reglas guardadas para IP6TABLES son almacenadas en el archivo ***/etc/sysconfig/ip6tables***. Las reglas viejas guardadas por los scripts de inicio de ip6tables son guardadas con el comando ***/etc/sysconfig/ip6tables.save***.

El archivo de configuración para los scripts de inicio de ip6tables es ***/etc/sysconfig/ip6tables-config***, en el se guarda información usada por el kernel para configurar los servicios de filtrado de paquetes IPv6 en el momento de arranque o cuando se arranque el servicio ip6tables.

### Configuraciones Básicas

En primera instancia se debe activar el soporte para IPv6 en el kernel puesto que, por defecto esta desactivado. Para realizar esto, se debe ingresar al archivo network que se encuentra en el directorio **/etc/sysconfig/** y agregar la variable **networking\_ipv6=yes**. Entonces el archivo debe contener:

```
NETWORKING=yes  
HOSTNAME=nombre_estacion.dominio  
NETWORKING_IPV6=yes
```

De igual forma, se debe editar el archivo **ifcfg-eth0** que se encuentra en el directorio **/etc/sysconfig/network-scripts/**. El archivo se debe editar de la siguiente manera:

- La variable **IPV6INIT=yes** se encarga de inicializar el soporte de IPv6 en la interfase. Es necesaria para todos los casos de configuración.
- La variable **IPV6\_AUTOCONF=yes** habilita la auto configuración en la interfase.
- La variable **IPV6ADDR="dirección ipv6"** debe ser activada si se decide por la configuración estática. Se debe especificar la dirección IPv6 con la que se va a comunicar en la red seguida del prefijo. (Ej: **IPV6ADDR=2004:20:800::3/48**)

Por ultimo es conveniente reiniciar los servicios de red y realizar las pruebas convenientes.

Para reiniciar el servicio: **service network restart**

Ahora nos centraremos en conocer la estructura que IP6TABLES maneja al momento de crear reglas para filtrado de paquetes, cuya sintaxis es similar a la manejada por IPTABLES.

**Comandos.-** Para la creación de una regla podemos usar varios comandos, ver **Tabla I.16:**

**Tabla I.16.** Comandos para reglas IP6TABLES

Comando	Descripción
-A	Este es el comando utilizado para agregar una regla cuando el orden de las reglas en la cadena no importa.
-D	Borra una regla de una cadena en particular. Puede también teclear la regla entera e ip6tables borrará la regla en la cadena que corresponda.
-F	Libera la cadena seleccionada, que borra cada regla de la cadena. Si no se especifica ninguna cadena, este comando libera cada regla de cada cadena.
-L	Lista todas las reglas de la cadena especificada tras el comando ip6tables.
-N	Crea una nueva cadena con un nombre especificado por el usuario.
-P	Configura la política por defecto para una cadena en particular de tal forma que cuando los paquetes atraviesen la cadena completa sin cumplir ninguna regla, serán enviados a un objetivo en particular, como puedan ser ACCEPT o DROP.
-X	Borra una cadena especificada por el usuario. No se permite borrar ninguna de las cadenas predefinidas para cualquier tabla.
-Z	Pone ceros en los contadores de byte y de paquete en todas las cadenas de una tabla en particular

**Parámetros.-** Para la creación de reglas se utiliza parámetros para una mejor estructura, algunos de ellos se los describe en la **Tabla I.17**:

**Tabla I.17.** Parámetros para una regla Ip6tables.

Parámetro	Descripción
-d	Configura el nombre de la máquina destino, la dirección IPv6 como: <i>fff:fff:fff:fff:0000:0000:0000:0000</i> que un paquete cumplirá la regla.
-i	Configura las interfaces de entrada de red, como eth0 o ppp0, para ser usadas por una regla.
-j	Dice a ip6tables que salte a un objetivo en particular cuando un paquete cumple una regla en particular. Los objetivos válidos que se usarán tras la opción -j incluyen opciones estándar, ACCEPT, DROP. Esto le permitirá aplicar otras reglas contra este paquete, y filtrarlo mejor con respecto a otros criterios.



<b>-o</b>	Configura la interfaz de red de salida para una regla en particular, y sólo puede ser usada con las cadenas OUTPUT y FORWARD. Estas opciones de los parámetros son los mismos que para los de la interfaz de red de entrada (opción -i).
<b>-p</b>	Configura el protocolo IP para la regla, que puede ser icmpv6, tcp, udp o all, para usar cualquier protocolo. Además, se pueden usar otros protocolos menos usados de los que aparecen en /etc/protocols. Si esta opción se omite al crear una regla, la opción all es la que se selecciona por defecto.
<b>-s</b>	Configura el origen de un paquete en particular usando la misma sintaxis que en el parámetro de destino (opción -d).

**Opciones de Paquetes icmpv6.-** Los paquetes que usan el protocolo icmpv6, en la **Tabla I.18** se muestra la opción para icmpv6:

**Tabla I.18.** Opciones para paquetes ICMPv6 en una regla de Ip6tables.

Opción	Descripción
<b>--icmpv6-type</b>	Selecciona el nombre o el número del tipo ICMPv6 que concuerde con la regla. Se puede obtener una lista de nombres válidos ICMP tecleando el comando ip6tables -p ipv6-icmp -h.

**Opciones de Paquetes TCP.-** Estas opciones de identificación están disponibles en el protocolo TCP (opción -p tcp), **ver Tabla I.19:**

**Tabla I.19.** Opciones para paquetes TCP en una regla de Ip6tables

Opción	Descripción
<b>--dport</b>	Configura el puerto de destino para el paquete. Puede utilizar o bien un nombre de servicio de red (como www o smtp), un número de puerto, o bien un rango de números de puertos para configurar esta opción. Para especificar un rango de números de puertos separe los dos números de puertos con dos puntos (:), como en -p tcp --dport 3000:3200. El rango válido más grande es 0:65535.
<b>--sport</b>	Configura el puerto de origen del paquete, usando las mismas opciones que --dport. También puede usar --source-port para especificar esta opción.
<b>--syn</b>	Provoca que todos los paquetes designados de TCP, comúnmente llamados paquetes SYN, cumplan esta regla. Cualquier paquete que esté llevando un payload de datos no será tocado. Si se situa

	un punto de exclamación (!) como flag tras la opción --syn se provoca que todos los paquetes no-SYN sean seleccionados.
<b>--tcp-flags</b>	Permite que los paquetes TCP con conjuntos de bits específicos, o flags, sean seleccionados para una regla. La opción de selección --tcp-flags acepta dos parámetros, que son los flags para los diferentes bits ordenados en una lista separada por comas. El primer parámetro es la máscara, que configura los flags que serán examinados en el paquete. El segundo parámetro se refiere a los flags que se deben configurar en el paquete para ser seleccionado. Los flags posibles son ACK, FIN, PSH, RST, SYN, y URG.
<b>--tcp-option</b>	Intenta seleccionar con opciones específicas de TCP que pueden estar activas en un paquete en particular.

**Opciones de Paquetes UDP.-** Estas opciones de selección están disponibles para el protocolo UDP (-p udp), ver **Tabla I.20**:

**Tabla I.20.** Opciones para paquetes UDP en una regla de Ip6tables

<b>Opción</b>	<b>Descripción</b>
<b>--dport</b>	Especifica el puerto destino del paquete UDP usando el nombre del servicio, el número del puerto, o un rango de puertos. La opción de selección de paquetes --destination-port se puede utilizar en lugar de --dport.
<b>--sport</b>	Especifica el puerto origen del paquete UDP usando el nombre del servicio número de puerto, o rango de puertos. La opción --source-port puede ser usada en lugar de --sport.

**Opciones del Objetivo.-** Una vez que un paquete cumple una regla en particular, la regla puede dirigir el paquete a un número de objetivos (destinos) diferentes que decidirán cuál será su destino y, posiblemente, las acciones adicionales que se tomarán, como el guardar un registro de lo que está ocurriendo.

Adicionalmente, cada cadena tiene un objetivo por defecto que será el que se utilice si ninguna de las reglas disponibles en dicha cadena se puede aplicar a dicho paquete, o si ninguna de las reglas que se aplican al mismo especifica un

objetivo concreto. Existen pocos objetivos estándar disponibles para decidir qué ocurrirá con el paquete, en la **Tabla I.21** se los describe:

**Tabla I.21.** Opciones para objetivos en una regla de Ip6tables.

Opción	Descripción
<b>ACCEPT</b>	Permite que el paquete se mueva hacia su destino (o hacia otra cadena, si no ha sido configurado ningún destino ha sido configurado para seguir a esta cadena).
<b>DROP</b>	Deja caer el paquete al suelo. El sistema que envió el paquete no es informado del fallo. El paquete simplemente se borra de la regla que está verificando la cadena y se descarta.
<b>REJECT</b>	Envía un paquete de error de vuelta al sistema que envió el paquete, y lo deja caer (DROP). Este objetivo puede ser útil si queremos notificar al sistema que envió el paquete del problema. El objetivo REJECT acepta una opción --reject-with <type> para proporcionar más detalles para ser enviados junto con el paquete de error. El mensaje port-unreachable es el error <type> que se envía por defecto cuando no se utiliza junto con otra opción.

### Orden de las Reglas

El orden de las reglas de un firewall IPv6 define su comportamiento. El filtro va comparando el paquete con cada una de las reglas hasta que se encuentra una que verifica y en ese caso se lleva a cabo lo que indique esa regla (aceptar o denegar); una vez realizada la acción no se comprueban más reglas.

### Política Predeterminada

También se puede dar el caso de que un paquete no haya verificado ninguno de los filtros una vez que se los ha comprobado todos. En este caso no se sabe si se tiene que aceptar o rechazar este paquete. Para resolver esta situación IP6TABLES dispone de una política predeterminada que permite indicar qué hacer con los paquetes que no hayan verificado ninguna regla. La política predeterminada será algunas de las acciones que hemos definido y se define con la opción "-P".

Por ejemplo:

**Ip6tables -P INPUT -j ACCEPT**

Definiría como aceptar la política predeterminada del filtro INPUT.

La política predeterminada define un esquema distinto de cortafuegos. Si la política predeterminada es aceptar entonces se tendrá que denegar todo aquello que no interese. Si la política predeterminada es denegar entonces se tiene que permitir todo aquello que sea de interés.

Cada uno de los modelos de cortafuegos tiene sus ventajas e inconvenientes. En el primer caso, con la política predeterminada de aceptar es mucho más fácil la gestión del firewall IPv6. En este caso es útil cuando se sabe claramente qué puertos se quiere proteger y el resto no importa y se acepta. Si la política predeterminada es ACEPTAR y no se protege explícitamente el sistema puede ser inseguro.

Cuando la política predeterminada es DENEGAR todo lo que no se acepte explícitamente será denegado por lo que el sistema puede fallar por despiste o desconocimiento en lo que se está permitiendo. Es más complicado establecer este tipo de cortafuegos, hay que tener muy claro cómo funciona el sistema y qué es lo que se tiene que aceptar sin caer en la tentación de introducir reglas muy permisivas.

#### 2.4.2. CONFIGURACIÓN PROXY

El término proxy hace referencia a un programa o dispositivo que realiza una acción en representación de otro. La finalidad más habitual es la del servidor proxy, que sirve para permitir el acceso a Internet a todos los equipos de una organización cuando sólo se puede disponer de un único equipo conectado, esto es, una única dirección IP. Un proxy actúa como gateway (*puerta de acceso*) pero a un nivel más alto, en la llamada "capa de aplicación". Significa que entiende HTTP, FTP, u algún otro protocolo de alto nivel y que acepta por parte de un cliente (de la red interna, por ejemplo) solicitudes vinculadas a dicho protocolo.

El proxy realizará, a su vez, la solicitud al servidor de destino, tomará el resultado y lo devolverá. Al tener conocimiento del protocolo se pueden aplicar reglas mucho más interesantes, como restricciones basadas en contenido, partes del nombre de un sitio, usuario, grupo al que un usuario pertenece, IP de origen, etc.

### **Beneficios/Funciones**

En general (no sólo en informática), los proxies hacen posibles varias cosas nuevas:

- **Control.** Sólo el intermediario hace el trabajo real, por tanto se pueden limitar y restringir los derechos de los usuarios, y dar permisos sólo al proxy.
- **Ahorro.** Por tanto, sólo uno de los usuarios (el proxy) ha de estar equipado para hacer el trabajo real.
- **Velocidad.** Si varios clientes van a pedir el mismo recurso, el proxy puede hacer caché: guardar la respuesta de una petición para darla directamente cuando otro usuario la pida. Así no tiene que volver a contactar con el destino, y acaba más rápido.
- **Filtrado.** El proxy puede negarse a responder algunas peticiones si detecta que están prohibidas.
- **Modificación.** Como intermediario que es, un proxy puede falsificar información, o modificarla siguiendo un algoritmo.
- **Anonimato.** Si todos los usuarios se identifican como uno sólo, es difícil que el recurso accedido pueda diferenciarlos. Pero esto puede ser malo, por ejemplo cuando hay que hacer necesariamente la identificación.

### **Desventajas**

En general (no sólo en informática), el uso de un intermediario puede provocar:

- **Abuso.** Al estar dispuesto a recibir peticiones de muchos usuarios y responderlas, es posible que haga algún trabajo que no toque. Por tanto, ha de controlar quién tiene acceso y quién no a sus servicios, cosa que normalmente es muy difícil.
- **Carga.** Un proxy ha de hacer el trabajo de muchos usuarios.
- **Intromisión.** Es un paso más entre origen y destino, y algunos usuarios pueden no querer pasar por el proxy. Y menos si hace de caché y guarda copias de los datos.
- **Incoherencia.** Si hace de caché, es posible que se equivoque y dé una respuesta antigua cuando hay una más reciente en el recurso de destino.
- **Irregularidad.** El hecho de que el proxy represente a más de un usuario da problemas en muchos escenarios, en concreto los que presuponen una comunicación directa entre 1 emisor y 1 receptor (como TCP/IP).

#### Antes de Configurarlo

Para poder configurarlo, es necesario conocer los parámetros que el archivo de configuración `/etc/squid/squid.conf` contiene, pero solo son algunos los que deben ser modificados para poder obtener un funcionamiento adecuado, en la

**Tabla I.22** se detalla los mismos.

**Tabla I.22.** Parámetros básicos de configuración de Squid.

Parámetro	Descripción
<b>Puerto</b>	Se trata de la directiva <code>http_port</code> , que por defecto está en el puerto 3128. Aunque esto se puede cambiar al que más nos guste (en algunos casos se elige el 8080). Queda así: <b><i>http_port 8080</i></b>
<b>Tamaño del archivo cacheable</b>	Este es el tamaño máximo de un archivo de Squid va a cachear. Esta directiva ( <code>maximum_object_size</code> ) es por defecto de 4 Mb, lo que resulta totalmente chico para una red corporativa. Lo vamos a reemplazar por 150 Mb (153600 bytes) ya que es un valor más apropiado. Los archivos de más de ese tamaño no serán cacheados. <b><i>Maximum_object_size 153600 KB</i></b>
<b>Uso de memoria</b>	El primer parámetro a tocar es <code>cache_mem</code> . Especifica la cantidad de memoria RAM, que Squid utilizará. Según recomendación de los

	autores, si tenemos "X" RAM que querramos dedicar al Squid, pongamos aqui un TERCIO de dicho valor (X/3). <p style="text-align: center;"><b>cache_mem 6 MB</b></p>	
<b>Tamaño del directorio de cache</b>	Luego podemos seguir por cache_dir, donde especificaremos el donde y el cuanto de la Cache. Por ejemplo: <p style="text-align: center;">cache_dir aufs /var/spool/squid 3072 16 256</p>	
	aufs	Es una directiva que aumenta el rendimiento de squid. Ya que lo vuelve multithread. La opción por defecto es ufs.
	/var/spool/squid  El valor 3072	Es el directorio raiz donde estarán los archivos cacheados Indica la cantidad de Mb que dispondremos de disco para caché, como máximo. Al 16 y al 256 es probable que no necesitemos nunca cambiarlos, e indican al Squid como utilizar los 3072 Mb

**Patrones de refresco:** Los patrones de refresco es la parte fundamental para hacer eficiente a Squid. Esto sucede porque su función es determinar que archivos se consideran frescos (y no deben volver a solicitarse a la url), separándolos de los que no cumplen dicha condición (y sí deben volver a solicitarse).

El mecanismo es complejo, porque la función de esta etiqueta es compleja. Por eso es que se incluye una buena cantidad de sugerencias que podrán servir de ejemplo.

La sinopsis de la etiqueta es la siguiente:

***refresh\_pattern [-i] regex min porcentaje max [opciones]***

- Como vemos -i es opcional (porque está entre corchetes). Lo que hace es activar la opción "case-insensitive", o sea, que no distinga mayúsculas de minúsculas. Si no ponemos esta opción, por defecto, es case-sensitive, como todo en Linux.

- regex se refiere a una expresión regular que define a qué tipo de objeto apuntamos con este patrón.
- min es el tiempo (en minutos) que un objeto, sin tiempo explícito de expiración, debe considerarse fresco.
- porcentaje es (valga la redundancia) el porcentaje de vida promedio de un objeto, en el que éste debe considerarse fresco.
- max es el tiempo máximo (en minutos) que un objeto, sin tiempo explícito de expiración, debe considerarse fresco.
- Las opciones permiten cambiar el comportamiento natural de estos tres parámetros de maneras específicas. Son muy poco utilizadas.

### **LISTAS DE CONTROL DE ACCESO (ACL)**

Tiene que ver con la posibilidad de fijar reglas en forma de listas (acl) que mostrarán los criterios a aplicarse en la red corporativa (como complemento del firewall). Por defecto se quiere permitir la utilización del Proxy a una red interna solamente. El siguiente bloque es un ejemplo para una red x.x.x.0 con máscara 255.255.255.0. Se permitirá acceso toda esta red y al localhost (127.0.0.1), esto se logra definiendo 4 acl's:

La del administrador (de uso interno del Squid), la de localhost, una global que hable de TODA dirección IP posible y la de permitidos (nuestra red privada).

En este ejemplo, se asume una red privada clase C x.x.x.0 a la cual se denomina "permitidos". La regla acl "todos" generalmente se denomina "all" en Squid, y viene definida por defecto.

```
acl localhost src 127.0.0.1/255.255.255.255  
acl todos src 0.0.0.0/0.0.0.0  
acl permitidos src x.x.x.0/255.255.255.0  
http_access allow permitidos localhost  
http_access deny todos  
icp_access allow permitidos localhost  
icp_access deny todos
```



Los indicadores **'deny'** y **'allow'** significan "denegar" y "permitir", respectivamente, a las solicitudes que concuerden con las ACL definidas más arriba en cada protocolo.

El orden es de 'allow, deny'. Primero se indica a quienes se permite, luego se deniega a todos los demás. Quizá se quiera hacer esto por cada {PROTOCOLO}\_access que se considere pertinente. Quizá se quiera modificar el 'cache\_mgr', para indicar la dirección de e-mail de una organización. Así, si algún usuario tiene problemas, sabe a quién contactar.

'visible\_hostname' indica el nombre del Host que se publicara en páginas de error y otras generadas por Squid.

Luego, se debe indicar con que usuario y grupo debe funcionar el Squid, luego de haber sido iniciado con root desde los scripts de inicio (o a mano por el root mismo). Esto se lo hace con 'cache\_effective\_user' y 'cache\_effective\_group'.

### 2.4.3. CONFIGURACIÓN GESTOR WEB

El servidor HTTP Apache es un software (libre), servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo.

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

Apache tiene amplia aceptación en la red, porque está diseñado para ser un servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios web elegir que características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor.

### **Ventajas**

- Modular
- Módulos de autenticación: mod\_access, mod\_auth y mod\_digest.
- Open source
- Multi-plataforma. Puede soportar de una forma más fácil y eficiente una amplia variedad de sistemas operativos.
- Extensible
- Popular (fácil conseguir ayuda/soporte)
- Gratuito
- El servidor puede personalizarse mejor para las necesidades de cada sitio web.
- Soporte para los lenguajes perl, python, tcl y PHP.
- Soporte para SSL y TLS.
- Permite la configuración de mensajes de errores personalizados y negociación de contenido.
- Provee soporte para el protocolo IPv6

### **Archivo de Configuración**

Cuando se instala Apache, se copia el archivo "**httpd.conf**", que está compuesto por tres bloques fundamentales, que bloques son:

- ❖ Parámetros globales

- ❖ Directivas de funcionamiento
- ❖ Hosts virtuales

❖ **Parámetros globales**

Algunos parámetros son propósito general, y otros son configurables de forma independiente para cada conjunto de directorios o de ficheros o incluso para un servidor virtual específico. En tales casos, estos parámetros se encuentran dentro de secciones en las que se indica el contexto de aplicación de dicho parámetro. Las secciones fundamentales son, **ver Tabla I.23:**

**Tabla I.23.** Parámetros globales de configuración de APACHE.

Parámetro	Descripción
<Directory>	Los parámetros que se encuentran dentro de la sección Directory sólo se aplican al directorio indicado y sus subdirectorios.
<DirectoryMatch>	Igual que Directory, aunque acepta expresiones regulares en el nombre del directorio.
<Files>	Los parámetros de configuración facilitan control de acceso a los ficheros mediante su nombre.
<FilesMatch>	Igual que Files, pero acepta en el nombre del fichero expresiones regulares.
<VirtualHost>	Los parámetros sólo se aplican a aquellas peticiones dirigidas a este host (nombre de servidor, dirección IP o puerto TCP).
<Proxy>	Sólo se aplican estos parámetros a aquellas peticiones de proxy (requiere que esté instalado "mod proxy") coincidentes con la especificación de URL.
<ProxyMatch>	Igual que proxy, pero acepta en la URL indicada el uso de expresiones regulares.

❖ **Directivas globales de configuración**

Algunas directivas de configuración nunca se aplican a las secciones antes mencionadas (directorios, etc.), sino que afectan al conjunto del servidor web. Las más destacables son, **ver Tabla I.24:**

**Tabla I.24.** Directivas globales de configuración de APACHE.

<b>Directiva</b>	<b>Descripción</b>
<b>ServerRoot</b>	Especifica la localización del directorio raíz en el que se encuentra instalado el servidor web. Partiendo de este directorio, se encuentran los ficheros de configuración, etc. Si la instalación del servidor es correcta, no debería modificarse nunca.
<b>Timeout</b>	Número de segundos tras los cuales el servidor cierra la conexión.
<b>KeepAlive</b>	Especifica si se deben utilizar conexiones persistentes para atender las peticiones de un mismo usuario mediante la misma conexión TCP.
<b>MaxKeepAliveRequest</b>	El número máximo de peticiones permitidas durante una conexión persistente.
<b>MinSpareServers</b>	Define el número de procesos servidores hijo desocupados que no atienden una petición.
<b>MaxSpareServers</b>	Define el nº máximo de procesos servidor hijo desocupados que no manejan una petición. Si existieran más de lo que define la directiva, el proceso padre mataría los procesos que exceden.
<b>StartServers</b>	Número de procesos servidor hijo que serán creados cuando arranca Apache por primera vez.
<b>Maxclients</b>	Define el número de peticiones simultáneas que apache puede soportar. Como máximo se crean este nº de procesos servidores hijo.
<b>MaxRequestPerChild</b>	Define el nº de peticiones que cada proceso hijo tiene permitido procesar antes de morir.
<b>Listen</b>	Especifica el puerto en que se atenderán las peticiones. Por defecto el servidor "escucha" en el puerto 80 de TCP. Permite especificar las direcciones IP que se utilizarán (en caso de que el servidor tuviese más de una). Por defecto se utilizarán todas las disponibles.

❖ **Directivas principales**

Hay algunas directivas que, generalmente, no suelen aparecer en las secciones anteriormente mencionadas (algunas de ellas no deben estar en

ninguna sección, y es obligatorio que aparezcan en la sección principal), sino que se encuentran en la sección principal, **ver Tabla I.25.**

**Tabla I.25.** Directivas principales de configuración de APACHE.

<b>Directiva</b>	<b>Descripción</b>
<b>Port</b>	Define el puerto en el cual escucha el servidor (0 - 65535). Hay que tener en cuenta la relación que tiene esta directiva con el fichero /etc/services y que algunos puertos, especialmente los situados por debajo del 1024, están reservados para protocolos específicos. El puerto estándar para el protocolo HTTP es el 80.
<b>User/Group</b>	Definen el usuario y el grupo con el que el servidor contestará las peticiones. Para poder utilizar esta directiva, el servidor standalone debe ejecutarse inicialmente como usuario root. El usuario no debería poseer privilegios que otorguen acceso a ficheros que no deseemos.
<b>ServerAdmin</b>	Especifica la dirección de correo electrónico del administrador. Esta dirección puede mostrarse en los mensajes de error a modo de dirección de contacto para que los usuarios notifiquen el error al administrador. No debe estar dentro de ninguna sección.
<b>ServerName</b>	Sirve para especificar el nombre y el puerto TCP que el Apache utiliza para identificarse. Se puede determinar de forma automática, pero se recomienda especificarlo. Si el servidor no tuviera un nombre DNS, es recomendable incluir su dirección IP. No debe incluirse dentro de ninguna sección. Su sintaxis es:  <b>ServerName nombre direccion: puerto</b> como en:  <b>ServerName www.algo.com:80</b> <b>ServerName 192.168.1.1:80</b>
<b>DocumentRoot</b>	Directorio raíz desde el cual se servirán los documentos. Por defecto es "htdocs", dentro de la carpeta de instalación de Apache. No debe aparecer dentro de ninguna sección, a excepción de la sección VirtualHost. Le corresponde una sección <Directory> en la cual se marcan los parámetros de configuración de este directorio.

<b>Directory</b>	<Directory></Directory> se utilizan para encerrar un grupo de directivas que se aplicarán al directorio en cuestión y sus sub-directorios. El parámetro directorio, puede ser una trayectoria completa o un metacaracter.
<b>AllowOverride override</b>	Cuando el servidor encuentra un fichero .htaccess (definido por la directiva AccessFileName) necesita conocer que directivas declaradas en este fichero pueden sobrescribir información de acceso.  El parámetro override puede ser definido a None y en tal caso el servidor no leerá el fichero o puede ser definido a All, de forma que permitirá todas las directivas.  El parámetro override también puede ser definido a: AuthConfig, FileInfo, Indexes, Limit, Options.
<b>UserDir</b>	Permite indicar a Apache que un subdirectorio dentro del directorio de trabajo de los diferentes usuarios del sistema sirva para que estos almacenen su página personal.  Por ejemplo:  <b>UserDir public</b>  hará que la página almacenada en el directorio del usuario "test", dentro del subdirectorio "público", sea accesible como:  <a href="http://www.algo.com/~test/indice.html">http://www.algo.com/~test/indice.html</a>
<b>DirectoryIndex</b>	Especifica el fichero que Apache servirá por defecto para cada directorio en caso de que no se especifique ningún fichero concreto en la URL de la petición. Por defecto es "index.html". Es decir, si se solicita en la barra de direcciones del navegador: <i>www.algo.com</i> el servidor enviará por defecto <i>www.algo.com/index.html</i> . Es posible especificar más de un fichero y el orden con que se especifican los ficheros determinará la prioridad para determinar cuál se debe servir. Es posible encontrar la directiva fuera de cualquier sección o dentro de alguna de ellas.
<b>Alias</b>	Las directivas Alias y AliasMatch permiten la definición de accesos a directorios que están fuera del DocumentRoot. Su sintaxis es: Alias url directorio. Por ejemplo:  <b>Alias /docs /home/documentos</b>  Hará que una petición a <a href="http://www.algo.com/docs/manual">http://www.algo.com/docs/manual</a> se sirva desde /home/documentos/manual.

<b>Location url</b>	La directiva proporciona control de acceso por URL. Es similar a la directiva Directory.
---------------------	--

❖ **Directivas de sección**

Casi todas las secciones de localización (Directory, Location, etc.) incluyen una serie de directivas en su configuración que permiten controlar el acceso al contenido, **ver Tabla I.26.**

**Tabla I.26.** Directivas de sección para la configuración de APACHE.

<b>Directiva</b>	<b>Descripción</b>
<b>Allow</b>	Permite especificar quién tiene autorización para acceder a un recurso. Se pueden especificar direcciones IP, nombres de máquina, fragmentos del nombre o de la dirección o variables de la petición. Existe la palabra clave "all" que indica "todos los clientes".
<b>Deny</b>	Permite especificar a quién no permitimos el acceso a un recurso. Cuenta con las mismas opciones que Allow.
<b>Order</b>	Permite afinar el funcionamiento de las anteriores directivas: Allow y Deny. Existen 2 opciones: <b>Allow, Deny.</b> Por defecto se deniega el acceso y sólo podrán acceder aquellos clientes que cumplan las especificaciones de Allow y en cambio no cumplan las especificaciones de Deny. <b>Deny, Allow.</b> Por defecto se permite el acceso y sólo podrán entrar los clientes que no cumplan las especificaciones de Deny o sí cumplan las especificaciones de Allow.

**Servidores Virtuales**

Apache permite servir varios sitios web con un único servidor. Para ello permite la creación de dominios virtuales en función de diferentes direcciones IP o diferentes nombres por IP. Apache fue de los primeros servidores que soportó servidores virtuales sin necesidad de distinguir por IP, sino en función de nombre. Esta capacidad simplifica enormemente la administración de los

servidores, y supone un ahorro de direcciones IP, que normalmente son escasas. Los servidores virtuales que distinguen en función del nombre son perfectamente transparentes para el cliente, con la posible excepción de aquellos navegadores muy antiguos que no envíen la cabecera "Host:" con cada petición.

## 2.5. LENGUAJE PHP

**PHP** es un acrónimo recursivo que significa "**PHP** Hypertext **P**re-processor" (inicialmente PHP Tools, o, *Personal Home Page Tools*), es un lenguaje de programación usado normalmente para la creación de contenido para sitios web con los cuales se puede programar las páginas html y los códigos de fuente. Se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando las librerías Qt o GTK+.

El PHP inicio como una modificación a Perl escrita por Rasmus Lerdorf a finales de 1994. Su primer uso fue el de mantener un control sobre quien visitaba su curriculum en su web. En los siguientes tres años, se fue convirtiendo en lo que se conoce como PHP/FI 2.0. Esta forma de programar llego a muchos usuarios, pero el lenguaje no tomo el peso actual hasta que Zeev Surasky y Andi Gutmans le incluyeron nuevas características en 1997, que dio por resultado el PHP 3.0. En Linux CentOS 5 la versión de PHP utilizada es la 5.1.6.

### 2.5.1. CARACTERÍSTICAS GENERALES

PHP es un preprocesador de hipertextos, la misma que tiene las siguientes características:

- Soporte sólido y REAL para Programación Orientada a Objetos (OOP) con PHP Data Objects. En PHP 5 hay un nuevo modelo de Objetos. El manejo de PHP



de objetos ha sido reescrito por completo, permitiendo un mejor desempeño y más características.

- Mejoras de rendimiento.
- Mejor soporte para MySQL con extensión completamente reescrita.
- Mejor soporte a XML.
- Soporte nativo para SQLite.
- Soporte integrado para SOAP.
- Iteradores de datos.
- Excepciones de errores.
- Más potencia que los lenguajes convencionales.
- Facilidad de aprendizaje.
- Escasez de consumo de recursos.

En el nivel más básico PHP es equiparable a un CGI cualquiera. Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, manipulación de cookies y páginas dinámicas. Un sitio con paginas dinámicas es el que permite interactuar con el visitante, de modo que cada usuario que visita la pagina vea la información modificada para requisitos articulares.

Las aplicaciones dinámicas para el Web son frecuentes en los sitios comerciales (e-commerce), donde el contenido visualizado se genera de la información alcanzada en una base de datos u otra fuente externa. La mayor fuerza de PHP es que está preparado para soportar accesos a muchos tipos de bases de datos como:

<b>1. Adabas D</b>	<b>6. InterBase</b>	<b>11. mSQL</b>
<b>2. dBase</b>	7. Solid	<b>12. MySQL</b>
<b>3. Empress</b>	8. Sybase	<b>13. Oracle</b>
<b>4. FiclePro</b>	9. Velocis	<b>14. PosgreSQL</b>
<b>5. informix</b>	<b>10. Unix dbm</b>	

El código PHP puede incluirse dentro del código html de la página. Para delimitar la sección de código PHP podemos hacerlo de varias formas:

- Usando las etiquetas `<?php` y `?>`

- Usando las etiquetas <? y ?>
- Mediante <script lenguaje="php"> </script>

## 2.5.2. MANEJO DE ARCHIVOS

Se nos vuelve necesaria la manipulación de archivos, para poder editarlos y configurar así CentOS, a fin de poder establecer la funcionalidad que deseamos del sistema.

PHP nos ofrece una serie de funciones que facilitan esta tarea, para poder acceder a archivos, leer los mismos e incluso editarlos, varias de estas funciones son las siguientes, ver **Tabla I.27**:

**Tabla I.27.** Comandos para manejo de archivos en PHP.

Comando	Descripción														
<b>copy</b>	Copia un archivo, su sintaxis es: <b>copy(\$origen,\$destino)</b>														
<b>rename</b>	Cambia el nombre de un archivo de \$nombre1 a \$nombre2, la sintaxis es: <b>rename(\$antes,\$despues)</b>														
<b>unlink</b>	Borra un archivo, y la sintaxis de esta función es: unlink(\$archivo)														
<b>fopen</b>	Abre un archivo y le asigna un identificador id. La sintaxis es la siguiente: \$id = fopen(\$archivo, \$modo)														
	Donde el modo puede ser uno de los siguientes:														
	<table border="1"> <thead> <tr> <th>Comando</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>'r'</td> <td>Sólo lectura</td> </tr> <tr> <td>'r+'</td> <td>Lectura y escritura</td> </tr> <tr> <td>'w'</td> <td>Sólo escritura</td> </tr> <tr> <td>'w+'</td> <td>Lectura y escritura. Suprime el contenido anterior si se escribe. El archivo es creado si no existe.</td> </tr> <tr> <td>'a'</td> <td>Sólo escritura. El archivo es creado si no existe y el puntero se coloca al final.</td> </tr> <tr> <td>'a+'</td> <td>Lectura y escritura. El archivo es creado si no existe y el puntero se coloca al final.</td> </tr> </tbody> </table>	Comando	Descripción	'r'	Sólo lectura	'r+'	Lectura y escritura	'w'	Sólo escritura	'w+'	Lectura y escritura. Suprime el contenido anterior si se escribe. El archivo es creado si no existe.	'a'	Sólo escritura. El archivo es creado si no existe y el puntero se coloca al final.	'a+'	Lectura y escritura. El archivo es creado si no existe y el puntero se coloca al final.
	Comando	Descripción													
	'r'	Sólo lectura													
	'r+'	Lectura y escritura													
	'w'	Sólo escritura													
'w+'	Lectura y escritura. Suprime el contenido anterior si se escribe. El archivo es creado si no existe.														
'a'	Sólo escritura. El archivo es creado si no existe y el puntero se coloca al final.														
'a+'	Lectura y escritura. El archivo es creado si no existe y el puntero se coloca al final.														
<b>fgets</b>	Lee una línea de un archivo hasta un número máximo de caracteres. La sintaxis es la siguiente: fgets(\$id,\$max)														
<b>fwrite</b>	Escribe una cadena dentro del archivo. La sintaxis es la siguiente: fwrite(\$id, \$cadena)														
<b>fseek</b>	Avanza o retrocede el puntero del archivo un cierto número de posiciones. La sintaxis es la siguiente: fseek(\$id,\$posiciones)														

<b>feof</b>	Comprueba si el puntero que lee el archivo ha llegado al final. La sintaxis es la siguiente: feof(\$id)
<b>fpass thru</b>	Lee completamente el archivo y lo muestra. La sintaxis es la siguiente: fpass thru(\$id)
<b>fclose</b>	Cierra el archivo abierto previamente. La sintaxis es la siguiente: fclose(\$id)

### 2.5.3. MANEJO DE COMANDOS UNIX

PHP, tiene varias funciones que nos permiten ejecutar comandos del shell de Linux. Pero antes de seguir con la caracterización de las funciones que PHP posee, definiremos rápidamente lo que es un Shell.

**Shell.**- Es un intérprete de comandos, es un programa especializado para aceptar comandos ingresados por el usuario, traduciendo aquellos en programas para ejecutarse, ejecutando esos programas, y mostrándolos (o haciendo algo) con los resultados.

Diferentes shells existen, que ofrecen diferentes características. Dos familias de shell existen:

- Shell Bourn y sus variantes (sh, bash, ksh)
- shell C y sus variantes (csh, tcsh).

Aunque muchas shells tienen características comunes a otras, la manera en que las usan es única, de manera que (por ejemplo) las convenciones de la shell Bourne usualmente no se aplican a las shells C.

**Bash.**- (la shell bourne-again shell) es la shell predeterminada para los usuarios de Linux. Es compatible con la tradicional shell Bourne (sh), los scripts de hechos en sh funcionarán en bash, aunque hay algunas características específicas a bash que no funcionarán en shells Bourne más viejas.

Otro de los aspectos importantes que se deben analizar, es que para la ejecución de comandos a través del browser, es que esta tarea es efectuada por el usuario apache.

Este usuario no tiene todos los privilegios como si lo tiene "root", por lo que es requerido hablar de "sudo".

**Sudo.**- Es una herramienta de sistema que permite a los usuarios realizar la ejecución de mandatos como superusuario u otro usuario de acuerdo a como se especifique en el fichero /etc/sudoers, donde se determina quien está autorizado. Los números de identidad de usuario y de grupo (UID: *identificador de usuario* y GID: *identificador de grupos*) reales y efectivas se establecen para igualar a aquellas del usuario objetivo como esté especificado en el fichero /etc/passwd.

De modo predeterminado sudo requiere que los usuarios se autenticen así mismos con su propia clave de acceso (nunca la clave de acceso de root). Una vez que el usuario se ha autenticado, el usuario podrá utilizar nuevamente sudo sin necesidad de volver a autenticarse durante 5 minutos, salvo que se especifique lo contrario en el fichero /etc/sudoers. Si el usuario ejecuta el mandato sudo -v podrá refrescar éste periodo de tiempo sin necesidad de tener que ejecutar un mandato, en cuyo caso contrario expirará esta autenticación y será necesario volver a realizarla.

Si un usuario no listado en el fichero /etc/sudoers trata de ejecutar un mandato a través de sudo, se registra la actividad en la bitácora de sistema (a través de syslogd) y se envía un mensaje de correo electrónico al administrador del sistema (root).

El fichero /etc/sudoers se edita con el mandato **visudo**, herramienta que a través de vi permite realizar cambios y verificar sintaxis y errores. Si se trata de modificar directamente /etc/sudoers, éste tiene permisos de solo lectura.

La sintaxis básica de una regla es:

***[usuario, %grupo, NOMBRELISTA] [anfitrión] = (id de usuario a usar) mandatos***

En nuestro caso, queremos darle a apache varios permisos, por lo que tendríamos entonces que realizar el siguiente procedimiento:

- Bajo la sintaxis "*Defaults requiretty*", añadimos la línea:

**Default logfile = /var/log/sudo.log**

- Bajo la línea “*root ALL = (ALL) ALL*”, aumentamos los permisos que le daremos al usuario apache, un caso de ejemplo:

***apache ALL = (root) NOPASSWD: /bin/ls, /sbin/service***

Luego de estas modificaciones, el usuario apache, está en capacidad de ejecutar por ejemplo, ls y puede iniciar o detener servicios.

Con la noción clara de lo que es un Shell y como darle permisos al usuario apache para que pueda ejecutar comando, veamos cuales son las funciones de PHP que nos permiten interactuar con el bash de Linux.

## **FUNCIONES PARA EJECUCIÓN DE COMANDOS**

**exec.-** Ejecutar un programa externo. La sintaxis es la siguiente:

```
string exec ( string comando [, array &salida [, int &var_retorno]] )
```

Donde los parámetros son:

- **Comando.-** El comando que será ejecutado.
- **Salida.-** Si el argumento salida está presente, entonces la matriz especificada será llenada con cada línea de la salida del comando. El espacio en blanco extra, como \n, no es incluido en esta matriz. Note que si la matriz ya contiene algunos elementos, exec() anexará sus resultados al final de la matriz. Si no desea que la función anexe los elementos, use unset() sobre la matriz antes de pasarla a exec().
- **var\_retorno.-** Si el argumento var\_retorno está presente junto con el argumento salida, entonces el status de retorno del comando ejecutado será escrito en esta variable. El valor retornado por la función es la última línea de los resultados del comando.

**shell\_exec.-** Ejecutar un comando mediante el intérprete de comandos y devolver la salida completa como una cadena. La sintaxis es la siguiente:

string **shell\_exec** ( string cmd )

Donde el parámetro es:

- **cmd**.- El comando que será ejecutado.

El valor retornado por la función, será la salida misma del comando. Con las funciones anteriores, podemos ejecutar comandos generales, pero debido a la necesidad de editar algunos scripts, necesitamos también de funciones que nos permitan manipular archivos. Para este cometido, veremos cuáles son las funciones de PHP que nos permitirán lograr este objetivo:

## **FUNCIONES PARA MANEJO DE ARCHIVOS**

**chmod ()**: Trata de cambiar los permisos del archivo especificado por nombre\_archivo a los permisos dados por modo. La sintaxis es la siguiente:

int **chmod** ( string nombre\_archivo, int modo )

Donde los parámetros son los siguientes:

- **nombre\_archivo**: Con él se especifica el nombre del archivo que se trata de cambiar los permisos.
- **modo**: Consiste de tres componentes de valor octal que especifican las restricciones de acceso para el propietario, el grupo de usuarios al que pertenece el propietario del archivo, y todo el mundo, en ese orden. Uno de los componentes puede ser calculado al agregarle los permisos necesarios para ese usuario en específico, El número 1 significa que tiene permisos de ejecución, el número 2 significa que puede modificar el contenido del archivo, el número 4 significa que puede leer el contenido del archivo. Para asegurar que se hace la operación esperada se necesita anteponer un cero (0) como prefijo del parámetro modo.

**fopen ()**.- Asocia un recurso con nombre, especificado por nombre\_archivo, a una secuencia. Si nombre\_archivo es de la forma "esquema://...", se asume que es una URL y PHP buscará por un gestor de protocolo (también conocido como envoltura)

para tal esquema. Si no hay envolturas registradas para ese protocolo, PHP emitirá una noticia para ayudarle a rastrear problemas potenciales en su script, y luego continúa como si nombre\_archivo indicara un archivo corriente.

La sintaxis de es la siguiente:

```
resource fopen ( string nombre_archivo, string modo [, bool usar_ruta_inclusion  
[, resource contexto_z]] )
```

Donde los parámetros son:

- **nombre\_archivo**: Es el nombre del archivo que se desea abrir, como se ha explicado en la definición de la función, puede ser la ruta de un archivo, o una URL.
- **modo**: Es el tipo de acceso que se tendrá al archivo especificado, puede tomar los siguientes valores:

'r': Apertura para sólo lectura; ubica el apuntador de archivo al comienzo del mismo.

'r+': Apertura para lectura y escritura; ubica el apuntador de archivo al comienzo del mismo.

'w': Apertura para sólo escritura; ubica el apuntador de archivo al comienzo de éste y lo trunca a una longitud de cero. Si el archivo no existe, intenta crearlo.

'w+': Apertura para lectura y escritura; ubica el apuntador de archivo al comienzo de éste y lo trunca a una longitud cero. Si el archivo no existe, intenta crearlo.

'a': Apertura para sólo escritura; ubica el apuntador de archivo al final del mismo. Si el archivo no existe, intenta crearlo.

'a+': Apertura para lectura y escritura; ubica el apuntador de archivo al final del mismo. Si el archivo no existe, intenta crearlo.

'x': Creación y apertura para sólo escritura; ubica el apuntador de archivo al comienzo de éste. Si el archivo ya existe, la llamada a fopen() fallará devolviendo FALSE y generando un error de nivel E\_WARNING.

'x+': Creación y apertura para lectura y escritura; ubica el apuntador de archivo al comienzo de éste. Si el archivo ya existe, la llamada a fopen() fallará devolviendo FALSE y generando un error de nivel E\_WARNING.

- **usar\_ruta\_inclusion**.- puede definirse como '1' o **TRUE** si desea buscar por el archivo en include\_path, también.

**fwrite ()**.-Escritura sobre archivos, segura con material binario. La sintaxis es la siguiente:

```
int fwrite ( resource gestor, string cadena [, int longitud] )
```

Donde los parámetros son:

- **gestor**: Es la fuente que contiene el archivo en el que se escribirá, es un almacén de datos.
- **cadena**: Es el contenido que se va a añadir en el archivo señalado por gestor.
- **Longitud**: Cuando el argumento longitud es entregado, la escritura se detendrá después de que longitud bytes hayan sido escritos, o al alcanzar el final de cadena, aquello que ocurra primero.

## 2.6. METODOLOGÍA DSDM

La metodología Dynamic Systems Development Method (DSDM) puede complementar metodologías de XP, RUP o Microsoft Solutions Framework, o combinaciones de todas ellas. DSDM es relativamente antiguo en el campo de los MAs (Métodos Ágiles) y constituye una metodología madura, que ya va por su cuarta versión. Se dice que ahora las iniciales DSDM significan Dynamic Solutions Delivery Method. Ya no se habla de sistemas sino de soluciones, y en lugar de priorizar el desarrollo se prefiere enfatizar la entrega.



### **Ventajas**

- Rápido desarrollo de aplicaciones.
- Eliminación de código repetitivo.
- Elaboración de aplicaciones más estables.
- El desarrollador cuenta con una metodología para el desarrollo de este tipo de aplicaciones.
- Interactividad entre el equipo de desarrollo, el usuario final y el gerente.
- Las pruebas se realizan a lo largo del ciclo de vida del proyecto. De esta manera garantiza cubrir los requerimientos del usuario.
- La probabilidad de que los usuarios acepten el uso del sistema en el computador.

### **Potenciales riesgos utilizando DSDM**

- Falta de involucramiento del usuario
- El tiempo excesivo consumido para tomar decisiones
- Incrementos irreversibles son desarrollados
- Las pruebas no se integran a lo largo del ciclo de vida
- Usuarios asignados al proyecto son "no queridos" por la organización
- Los usuarios consiguen involucrarse en el proyecto.
- Las estructuras de los datos son monolíticas e inflexibles debido al rápido prototipado de datos

## **PRINCIPIOS**

En DSDM las prácticas se llaman Principios, y son nueve:

- a. Es imperativo el compromiso activo del usuario.
- b. Los equipos de DSDM deben tener el poder de tomar decisiones.
- c. El enfoque radica en la frecuente entrega de productos.
- d. El criterio esencial para la aceptación de los entregables es la adecuación a los propósitos de negocios.
- e. Se requiere desarrollo iterativo e incremental.
- f. Todos los cambios durante el desarrollo son reversibles.

- g. La línea de base de los requerimientos es de alto nivel. Esto permite que los requerimientos de detalle se cambien según se necesite y que los esenciales se capten tempranamente.
- h. La prueba está integrada a través de todo el ciclo de vida. La prueba también es incremental. Se recomienda particularmente la prueba de regresión, de acuerdo con el estilo evolutivo de desarrollo.
- i. Es esencial una estrategia colaborativa y cooperativa entre todos los participantes. Las responsabilidades son compartidas y la colaboración entre usuario y desarrolladores no debe tener fisuras.

## REGLAS DE DSDM

La idea dominante detrás de DSDM es explícitamente inversa a la que se encuentra en otras partes, y al principio resulta contraria a la intuición; en lugar de ajustar tiempo y recursos para lograr cada funcionalidad, en esta metodología tiempo y recursos se mantienen como constantes y se ajusta la funcionalidad de acuerdo con ello. Esto se expresa a través de reglas que se conocen como “reglas MoSCoW” por las iniciales de su estipulación en inglés. Las reglas se refieren a rasgos del requerimiento:

**Tabla I.28.** Reglas metodología DSDM

<b>Must have</b>	Debe tener. Son los requerimientos fundamentales del sistema. De éstos, el subconjunto mínimo ha de ser satisfecho por completo
<b>Should have</b>	Debería tener. Son requerimientos importantes para los que habrá una resolución en el corto plazo.
<b>Could have</b>	Podría tener. Podrían quedar fuera del sistema si no hay más remedio.
<b>Want to have but won't have this time around</b>	Se desea que tenga, pero no lo tendrá esta vuelta. Son requerimientos valorados, pero pueden esperar.

## ROLES

DSDM define quince roles, algo más que el promedio de los Mas (métodos ágiles). Los más importantes son:

Programadores y Programadores Senior. Son los únicos roles de desarrollo. El título de Senior indica también nivel de liderazgo dentro del equipo. Ambos títulos cubren todos los roles de desarrollo, incluyendo analistas, diseñadores, programadores y verificadores.

**a. Coordinador técnico.**

Define la arquitectura del sistema y es responsable por la calidad técnica del proyecto, el control técnico y la configuración del sistema.

**b. Usuario embajador.**

Proporciona al proyecto conocimiento de la comunidad de usuarios y disemina información sobre el progreso del sistema hacia otros usuarios. Se define adicionalmente un rol de Usuario Asesor (Advisor) que representa otros puntos de vista importantes; puede ser alguien del personal de IT o un auditor funcional.

**c. Visionario.**

Es un usuario participante que tiene la percepción más exacta de los objetivos del sistema y el proyecto. Asegura que los requerimientos esenciales se cumplan y que el proyecto vaya en la dirección adecuada desde el punto de vista de aquéllos.

**d. Patrocinador Ejecutivo.**

Es la persona de la organización que detenta autoridad y responsabilidad financiera, y es quien tiene la última palabra en las decisiones importantes.

**e. Facilitador.**

Es responsable de administrar el progreso del taller y el motor de la preparación y la comunicación.

**f. Escriba.**

Registra los requerimientos, acuerdos y decisiones alcanzadas en las reuniones, talleres y sesiones de prototipado.

## FASES

DSDM consiste en cinco fases:

**1. Estudio de factibilidad.**

- 2. Estudio del negocio.**
- 3. Iteración del modelo funcional.**
- 4. Iteración de diseño y versión.**
- 5. Implementación.**

Las últimas tres fases son iterativas e incrementales. De acuerdo con la iniciativa de mantener el tiempo constante, las iteraciones de DSDM son cajas de tiempo. La iteración acaba cuando el tiempo se consume. Se supone que al cabo de la iteración los resultados están garantizados. Una caja de tiempo puede durar de unos pocos días a unas pocas semanas.

DSDM ha desarrollado sistemáticamente el problema de su propia implantación en una empresa. El proceso de Examen de Salud (Health Check) de DSDM se divide en dos partes que se interrogan, sucesivamente, sobre la capacidad de una organización para adoptar el método y sobre la forma en que éste responde a las necesidades una vez que el proyecto está encaminado. Un Examen de Salud puede insumir entre tres días y un mes de trabajo de consultoría.

- a. Estudio de factibilidad.** Se evalúa el uso de DSDM o de otra metodología conforme al tipo de proyecto, variables organizacionales y de personal. Si se opta por DSDM, se analizan las posibilidades técnicas y los riesgos. Se preparan como productos un reporte de viabilidad y un plan sumario para el desarrollo. Si la tecnología no se conoce bien, se hace un pequeño prototipo para ver qué pasa. No se espera que el estudio completo insuma más de unas pocas semanas. Es mucho para un método ágil, pero menos de lo que demandan algunos métodos clásicos.
- b. Estudio del negocio.** Se analizan las características del negocio y la tecnología. La estrategia recomendada consiste en el desarrollo de talleres, donde se espera que los expertos del cliente consideren las facetas del sistema y acuerden sus prioridades de desarrollo. Se describen los procesos de negocio y las clases de usuario en una Definición del Área de Negocios. Se espera así reconocer e involucrar a gente clave de la organización en una etapa temprana. La Definición utiliza descripciones de alto nivel, como diagramas de entidad-relación o modelos de objetos de negocios.

- c. Iteración del modelo funcional.** En cada iteración se planea el contenido y la estrategia, se realiza la iteración y se analizan los resultados pensando en las siguientes. Se lleva a cabo tanto el análisis como el código; se construyen los prototipos y en base a la experiencia se mejoran los modelos de análisis. Los prototipos no han de ser descartados por completo, sino gradualmente mejorados hacia la calidad que debe tener el producto final. Se produce como resultado un Modelo Funcional, conteniendo el código del prototipo y los modelos de análisis. También se realizan pruebas constantemente.
- d. Iteración de diseño y construcción.** Aquí es donde se construye la mayor parte del sistema. El producto es un Sistema Probado que cumplimenta por lo menos el conjunto mínimo de requerimientos acordados conforme a las reglas MoSCoW. El diseño y la construcción son iterativos y el diseño y los prototipos funcionales son revisados por usuarios. El desarrollo ulterior se atiene a sus comentarios.
- e. Despliegue.** El sistema se transfiere del ambiente de desarrollo al de producción. Se entrena a los usuarios, que ponen las manos en el sistema. Eventualmente la fase puede llegar a iterarse. Otros productos son el Manual de Usuario y el Reporte de Revisión del Sistema. A partir de aquí hay cuatro cursos de acción posibles: (1) Si el sistema satisface todos los requerimientos, el desarrollo ha terminado. (2) Si quedan muchos requerimientos sin resolver, se puede correr el proceso nuevamente desde el comienzo. (3) Si se ha dejado de lado alguna prestación no crítica, el proceso se puede correr desde la iteración funcional del modelo en adelante. (4) si algunas cuestiones técnicas no pudieron resolverse por falta de tiempo se puede iterar desde la fase de diseño y construcción.

## CICLO DE VIDA

Como ya hemos visto el ciclo de vida contempla cinco fases:

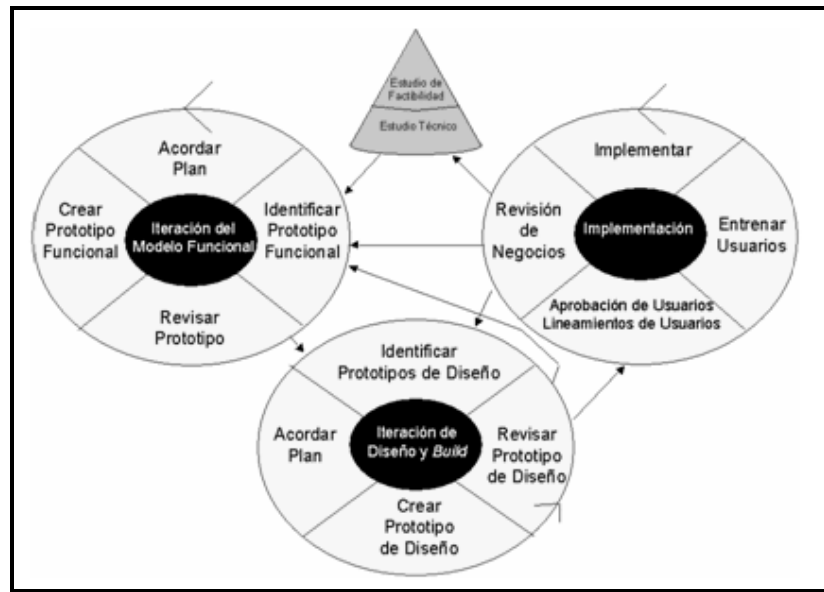
- Estudio viabilidad,
- Estudio del negocio,
- Modelado funcional,

- Diseño y
- Construcción e implementación.

El método empieza con un estudio de viabilidad y negocio. El estudio de viabilidad considera si DSDM es apropiado para el proyecto. El estudio de negocio es una serie corta de talleres para entender el área de negocio dónde tiene lugar el desarrollo. También propone arquitecturas de esbozos del sistema y un plan del proyecto. Las tres últimas son iterativas, además de existir retro alimentación a todas las fases.

Se ha notado que DSDM tiene principios subyacentes que incluyen una interacción activa del usuario, entregas frecuentes, equipos autorizados, pruebas a lo largo del ciclo. Igual que otros métodos ágiles, usan ciclos de plazos cortos de entre dos y seis semanas. Hay un énfasis en la alta calidad y adaptabilidad hacia requisitos cambiantes.

La configuración del ciclo de vida de DSDM se representa con un diagrama característico que vale la pena reproducir, **ver Fig. I.14:**



**Fig. I.14.** Proceso de desarrollo DSDM

---

## CAPÍTULO III

# ANÁLISIS DE LAS TÉCNICAS DE CONVIVENCIA EN LINUX CENTOS 5 Y POLÍTICAS DE RED DE LA E.I.S.

---

La temática tratada en este capítulo es determinar los parámetros necesarios para configurar Firewalls IPv4 e IPv6 y Proxy, tomando como referencia las políticas de red existentes en el nodo de red de la EIS, y en base a la información obtenida estructurar los módulos de configuración necesarios para ser administrados mediante un entorno Web.

Por otra parte, se analiza las configuraciones y se realiza pruebas con las Técnicas de Convivencia entre IPv4 e IPv6 bajo Linux CentOS 5, para de ésta manera realizar un Estudio Comparativo y determinar cual técnica de convivencia se acopla de mejor manera a nuestros requerimientos.

### 3.1. SITUACIÓN ACTUAL DE LA INTRANET DE LA ESPOCH

La Escuela Superior Politécnica de Chimborazo, ha ido creciendo en los últimos años, buscando mejorar toda su infraestructura, tanto física como tecnológica, eso ha permitido mantener latente la posibilidad de estar siempre a la par a los diferentes avances que se dan en el mundo informático.

#### 3.1.1. INFRAESTRUCTURA IPv4

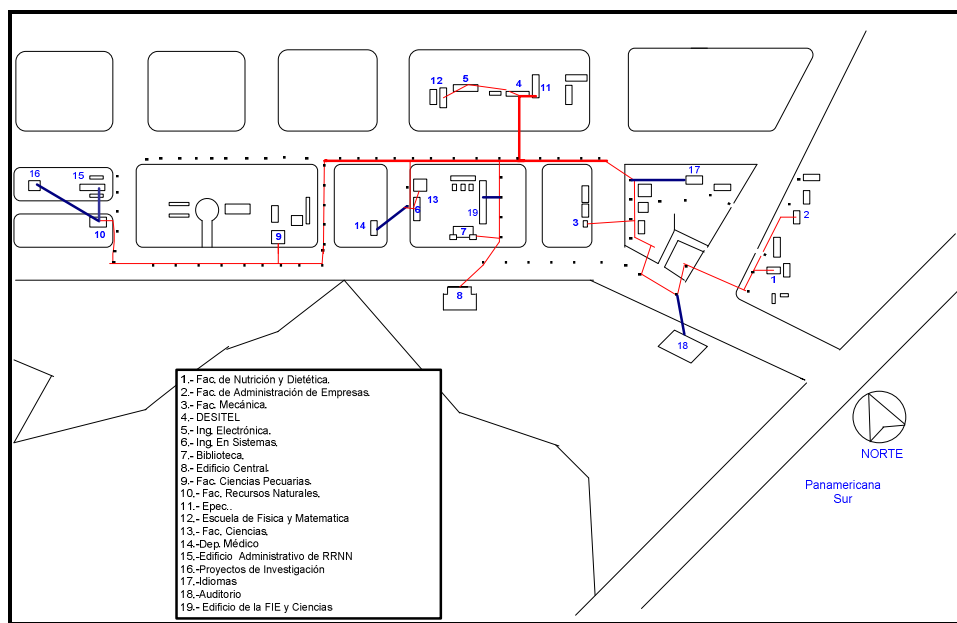


Fig. II.15. Infraestructura IPv4 de la ESPOCH.

El campus de la ESPOCH cuenta con un tendido de fibra óptica en gran parte de su instalación, permitiendo la comunicación e interconexión de todas las subredes existentes en cada facultad. Este detalle es muy importante, ya que es la columna vertebral de todas las comunicaciones dentro de la institución, como se puede observar en la Fig. II.15.



Permitiendo de esta manera hacer uso de los diferentes servicios de red que la ESPOCH brinda a cada una de sus dependencias, como son: Telefonía IP, Internet, Internet Inalámbrico, videoconferencias, etc.

### **3.1.2. INFRAESTRUCTURA IPV6**

Debido a que la ESPOCH está en un crecimiento paulatino, la infraestructura IPv6 está en los planes futuros de la institución, para poder estar acorde a la situación tecnológica del mundo.

Cabe mencionar que la migración de la infraestructura IPv6 se la realiza de forma paulatina, es decir en determinados puntos de la Intranet se está realizando esta migración, por ello se ha realizado la adquisición de varios equipos tecnológicos que cuentan con el soporte necesario para IPv6 entre ellos Switch Cisco que soportan el nuevo protocolo, además de la adecuación de determinados laboratorios cuya infraestructura soporte IPv6. Un caso específico de estas adecuaciones lo tenemos en el Laboratorio de Investigación de la Escuela de Ingeniería en Sistemas, cuyo tendido de red está formado en cableado categoría 6, además los equipos existentes en dicho laboratorio cuentan con tarjetas de 1Gbps permitiendo un mejor desempeño de la red.

De igual forma, se cuenta con software necesario que proporcione soporte al Protocolo IPv6, entre los que podemos mencionar el uso de las últimas versiones de Linux, entre otros componentes complementarios.

## **3.2. POLÍTICAS DE RED DEL NODO DE RED DE LA E.I.S**

Nuestra investigación se enfoca en examinar y determinar que políticas de administración de red son usadas en un nodo de red específico. Por ello se ha tomado como referencia el nodo de red de la Escuela de Ingeniería en Sistemas, debido a que la infraestructura que

posee se acopla a los requerimientos de nuestra investigación, que es el soporte a los Protocolos IPv4 e IPv6.

El análisis a efectuarse en el nodo de red de la Escuela de Ingeniería en Sistemas, permite conocer que recursos y/o servicios, herramientas, etc. son utilizadas para la adecuada administración del tráfico de red.

Para la recopilación de dicha información, se efectuó una entrevista a uno de los administradores de red de la Escuela de Ingeniería en Sistemas, el cual nos dio a conocer los recursos existentes en dicho nodo de red y que explicaremos a continuación.

### 3.2.1. INFRAESTRUCTURA TECNOLÓGICA

Para tener un mejor conocimiento de la infraestructura vigente en el nodo de la Escuela de Ingeniería en Sistemas, analizaremos los componentes que la conforman.

#### **SERVIDORES**

**Servidor Proxy**, este equipo es el encargado de proveer el servicio de Internet a la EIS, además de denegar el acceso a ciertos recursos, como:

- Descarga de archivos multimedia (mp3, wmv, avi, mp3, iso, etc.).
- Acceso a páginas de contenido no apropiado.
- Uso de programas P2P, como: Messenger, ares, entre otros.

Este equipo trae incluido el Sistema Operativo CentOS 4.6, que incorpora el kernel 2.6.9 y la versión del archivo SQUID es la 2.5, dicho fichero tiene configurados parámetros específicos que cumplen la finalidad de realizar las tareas enunciadas anteriormente.

Para la realización de nuestro proyecto se tomará como referencia las configuraciones realizadas en el fichero SQUID, para de esta manera crear un entorno gráfico que permita la configuración de dichos parámetros. Entre los que están:

- Configuración de Puertos
- Tamaño de Memoria Caché
- Directorio Caché
- Usuario FTP
- Servidor DNS
- Listas de Control de Acceso (ACL)

**Servidor Web**, este equipo es usado con fines didácticos por el Ing. Danilo Pástor en la asignatura Aplicaciones Web.

**Servidor Descargas**, este equipo es usado como soporte del servidor Proxy para descargar ficheros.

**Servidor Software**, este equipo provee un repositorio de elementos software, necesarios en la Escuela de Ingeniería en Sistemas, entre el software que provee este servidor están: Office, Dreamweaver, Flash, Firefox, entre otros.

## **ENTORNO DE RED**

Actualmente los equipos de red que existen en la EIS están orientados hacia los protocolos IPv4 e IPv6, por ejemplo el switch principal es un Cisco Catalyst 3750G dicho modelo incluye soporte para el protocolo IPv6 y es el encargado de administrar las peticiones de los usuarios de la EIS hacia el servidor principal ubicado en el Departamento de Sistemas y Telemática (DESITEL) mediante una conexión directa de fibra óptica.



**Fig. II.16. Infraestructura tecnológica de la EIS.**

Además se utiliza varios switch adicionales de respaldo aparte del switch Cisco principal, dichos switch adicionales reciben las peticiones de los diferentes laboratorios existentes en la EIS, **ver Fig. II.16.**

Vale mencionar que la asignación de direcciones IP las efectúa el DESITEL, mediante el uso de VLAN's ya que ellos son los encargados de distribuir los recursos hacia cada dependencia existente en la ESPOCH.

### **ADMINISTRACIÓN DE RED**

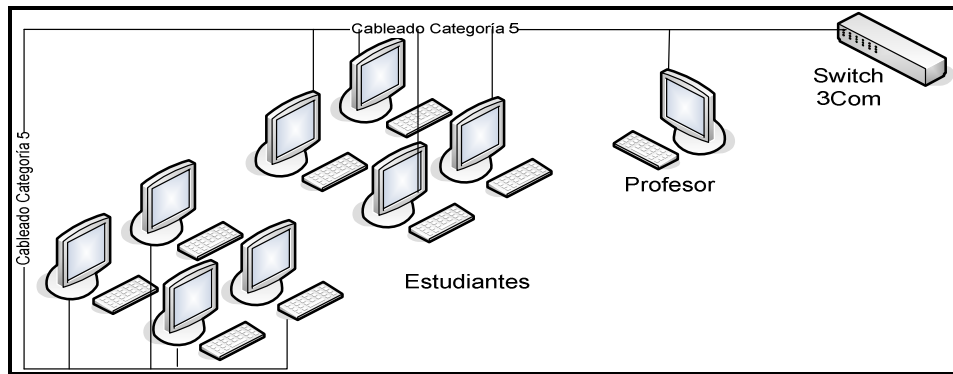
Los contenidos del tráfico de red de todos los laboratorios son filtrados mediante la utilización del fichero SQUID que está incluido en el servidor Proxy de la EIS, es decir cada petición que un usuario efectúe hacia Internet es filtrado por el SQUID.

Es importante mencionar que no está implantado un filtrador de paquetes que permita administrar de una manera adecuada el tráfico generado en la red, todo lo referente a políticas para filtrado de paquetes lo realiza directamente el firewall de la ESPOCH ubicado en el Departamento de Sistemas y Telemática (DESITEL).

## INFRAESTRUCTURA FISICA

Cada laboratorio de la EIS cuenta con una infraestructura basada en el protocolo IPv4, es decir cableado estructurado categoría 5 a excepción del Laboratorio de Investigación y Desarrollo que cuenta con infraestructura orientada al Protocolo IPv6, es decir cableado estructurado categoría 6. La infraestructura de red completa de la Escuela de Ingeniería en Sistemas se encuentra en el **Anexo A**.

Además cada laboratorio posee un switch local que se conecta con el switch principal ubicado en el nodo de la EIS para de ésta manera proveer los servicios a las peticiones efectuadas por los usuarios de la red, **ver Fig. II.17**.



**Fig. II.17.** Infraestructura Laboratorios EIS.

### 3.2.2. RESULTADOS OBTENIDOS

Una vez recopilada la información necesaria tras el análisis de las Políticas de Red de la Escuela de Ingeniería en Sistemas, se obtuvo los siguientes resultados:

- En lo referente a mecanismos que permitan el filtrado de paquetes de red, no existe una herramienta que cumpla con esta finalidad; es decir, un Firewall. Por lo que sería conveniente la utilización de un Firewall que cumpla con este propósito.

- Para el filtrado de contenidos de información para acceso a Internet, se utiliza un Servidor Proxy, cuyo servicio de administración es SQUID, el cual tiene configurado los siguientes parámetros:
  - a. Configuración de Puertos
  - b. Tamaño de Memoria Caché
  - c. Directorio Caché
  - d. Usuario FTP
  - e. Servidor DNS
  - f. Listas de Control de Acceso (ACL)
  
- Con la información recopilada tras examinar las Políticas de Red de Nodo de la Escuela de Ingeniería en Sistemas, podemos delinear los módulos y los parámetros de configuración para administrar el tráfico de red usando Firewalls (IPv4 e IPv6) y Proxy, administrables mediante un entorno Web.

### 3.3. APLICACIÓN DE LAS TÉCNICAS DE CONVIVENCIA Y SELECCIÓN DE LA MÁS IDÓNEA

.Las distribuciones actuales de Linux incluyen el soporte necesario para IPv6 en sus respectivos Kernels, es decir, versiones del kernel a partir de la 2.6.x incluyen soporte para el protocolo IPv6.

La investigación utiliza Linux CentOS 5 y para verificar el soporte IPv6 en su kernel, se comprueba de la siguiente manera: *Nos logueamos como root y abrimos una terminal y tipeamos la siguiente línea:*

```
[root@ejm ~]# test -f /proc/net/if_inet6 && echo "existe soporte IPv6 en el kernel"
existe soporte IPv6 en el kernel
```

Si existiera algún error, cargamos el módulo que nos brinda el soporte necesario, de la siguiente manera:

```
[root@ejm ~]# modprobe ipv6
```

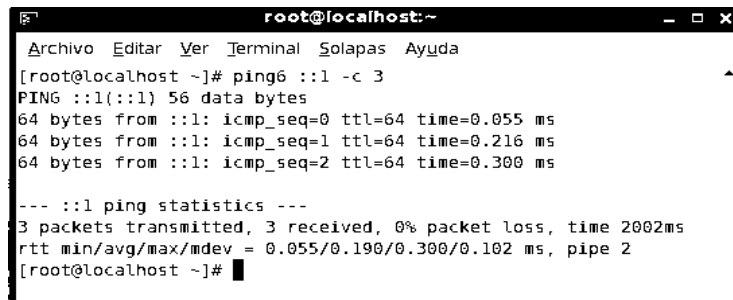
Una vez hecha la carga del módulo, la comprobamos de la siguiente manera:

```
[root@ejm ~]# lsmod | grep -w 'ipv6' && echo "modulo IPv6 cargado exitosamente"
ipv6 251136 19 ip6t_REJECT
modulo IPv6 cargado exitosamente
```

Con los pasos efectuados anteriormente se ha comprobado el soporte IPv6 en el Kernel

2.6.18 que incorpora Linux CentOS 5.

**Prueba:** Verificamos el soporte IPv6 en CentOS 5, realizando un ping6 a la dirección loopback de IPv6 que es la "::1", como se aprecia en la **Fig. II.18:**

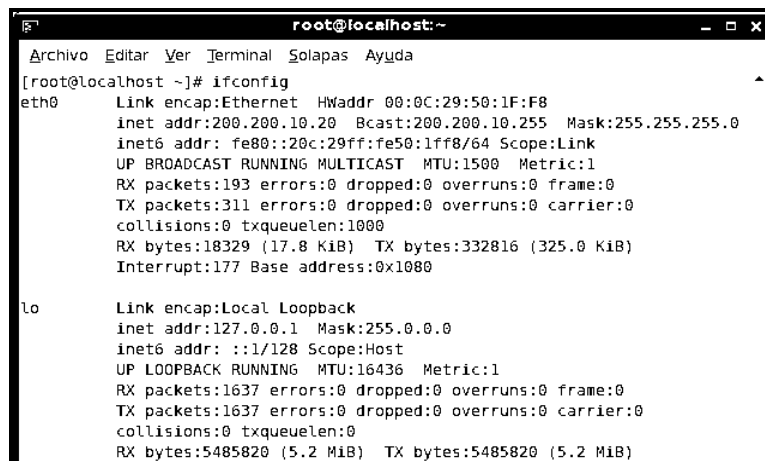


```
root@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost ~]# ping6 ::1 -c 3
PING ::1(::1) 56 data bytes
64 bytes from ::1: icmp_seq=0 ttl=64 time=0.055 ms
64 bytes from ::1: icmp_seq=1 ttl=64 time=0.216 ms
64 bytes from ::1: icmp_seq=2 ttl=64 time=0.300 ms

--- ::1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.055/0.190/0.300/0.102 ms, pipe 2
[root@localhost ~]#
```

Fig. II.18. Ejecución ping6

Adicionalmente, verificamos el direccionamiento IPv6 en las interfaces de red del equipo:



```
root@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:50:1F:F8
          inet addr:200.200.10.20  Bcast:200.200.10.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe50:1ff8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:193 errors:0 dropped:0 overruns:0 frame:0
          TX packets:311 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:18329 (17.8 KiB)  TX bytes:332816 (325.0 KiB)
          Interrupt:177 Base address:0x1080

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1637 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1637 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5485820 (5.2 MiB)  TX bytes:5485820 (5.2 MiB)
```

Fig. II.19. Verificación interfaces red activas

En la **Fig. II.19**, se aprecia que en cada interfaz de red; existen 2 tipos de direcciones, una corresponde a la dirección IPv4 y la otra corresponde a una dirección IPv6 cuyo prefijo es **fe80** que significa que es una dirección IPv6 local.

### 3.3.1. ANÁLISIS DE LAS TÉCNICAS DE CONVIVENCIA EN CENTOS 5.0

Una vez verificado el soporte IPv6 en Linux CentOS 5, se procede a caracterizar las Técnicas de Convivencia entre los protocolos IPv4 e IPv6 bajo CentOS 5, elegidas para nuestra investigación.

#### 3.3.1.1. PILA DUAL

Tiene un enfoque muy sencillo de implementar que requiere que los hosts y los routers soporten ambas versiones de IP y por tanto, servicios y aplicaciones tanto IPv4 como IPv6. Estos nodos tienen la habilidad de enviar y recibir paquetes IPv6 e IPv4, pudiendo así interoperar directamente con nodos IPv4 usando paquetes IPv4, y también operar con nodos IPv6 usando paquetes IPv6.

#### CONFIGURACIÓN DE PILA DUAL EN CentOS 5

El soporte IPv6 en Linux CentOS 5 viene incorporado en su kernel, simplemente nos resta determinar las direcciones para cada protocolo.

El proceso es sencillo, si queremos determinar direcciones temporales, que se eliminarán al reiniciar el servicio de red, simplemente nos logeamos como **root** al arrancar el sistema y digitamos lo siguiente en una terminal.

**Para IPv4:** [root@ejm ~]# ip addr add xxx.xxx.xxx.xxx/xx dev ethx

**Para IPv6:** [root@ejm ~]# ip addr add xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xx dev ethx

Donde ethx, hace referencia a la tarjeta de red que se desea configurar con los dos protocolos.

**Por ejemplo:** [root@ejm ~]# ip addr add 3ffe:1234:5:5::1/64 dev eth0

Verificamos si la dirección IPv6 ha sido ingresada, realizando un **ifconfig** como se puede apreciar en la **Fig. II.20**.



```
root@localhost:~  
Archivo Editar Ver Terminal Solapas Ayuda  
[root@localhost ~]# ifconfig  
eth0      Link encap:Ethernet  HWaddr 00:0C:29:50:1F:F8  
          inet addr:200.200.10.20  Bcast:200.200.10.255  Mask:255.255.255.0  
          inet6 addr: 3ffe:1234:5:5::1/64 Scope:Global  
          inet6 addr: fe80::20c:29ff:fe50:1ff8/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:331 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:518 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:33036 (32.2 KiB)  TX bytes:558778 (545.6 KiB)  
          Interrupt:177 Base address:0x1080  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:16436  Metric:1  
          RX packets:1649 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:1649 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:5487068 (5.2 MiB)  TX bytes:5487068 (5.2 MiB)  
[root@localhost ~]#
```

Fig. II.20. Verificación interfaces red activas con Pila Dual

Si se desea que estas direcciones IPv6 queden establecidas permanentemente, es necesaria la edición de un archivo del sistema, ubicado en:

`/etc/sysconfig/network-script/ifcg-ethx,`

Donde x, puede ser 0, 1, 2, etc. Es decir la tarjeta de red que se desee establecer la dirección. Solo añadimos lo siguiente:

`IPV6ADDR= xxxx:xxxx:xxxx:xxxx:xxxx:xxxx` , para fijar la dirección IPv6, y `IPV6INIT=yes,` para que arranque el módulo al iniciar el sistema.

Finalmente digitamos:

`Service network restart,` para reiniciar el servicio de red.

Con esto, ya tenemos la pila dual activada en una determinada interfaz de red, bajo Linux CentOS 5.

## PILA DUAL – PRUEBAS REALIZADAS

Para asegurarnos de la correcta configuración de Pila Dual, se realizo pruebas simples para verificar el desempeño de este mecanismo de convivencia:

- Uso del navegador para acceder a direcciones IPv6
- Envío de paquetes ICMPv6

#### a. Pruebas con el navegador

Para realizar este tipo de pruebas fue necesario configurar al equipo con una dirección IPv6 global, como se puede apreciar en la **Fig. II.21**. Por defecto la versión de Apache que incluye CentOS 5 viene con soporte para IPv6, por ello no fue necesario realizar ningún tipo de configuración adicional al servidor web Apache.

```
[root@localhost ~]# ip addr add 3ffe:1234:5:5::3 dev eth0
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:16:76:C8:6C:FF
          inet addr:192.168.10.5  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: 3ffe:1234:5:5::3/128 Scope:Global
          inet6 addr: fe80::216:76ff:fec8:6cff/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:116 errors:0 dropped:0 overruns:0 frame:0
          TX packets:124 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:28605 (27.9 KiB)  TX bytes:24845 (24.2 KiB)
          Base address:0x20c0  Memory:50300000-50320000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2733 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2733 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4102652 (3.9 MiB)  TX bytes:4102652 (3.9 MiB)

[root@localhost ~]# █
```

**Fig. II.21.** Asignación dirección IPv6 a una interfaz

Para verificar el soporte de Apache hacia IPv6 abrimos un navegador y colocamos la dirección IPv6 entre corchetes asignada al equipo, cuyos pasos se encuentran descritos en la **Fig. II.21**.

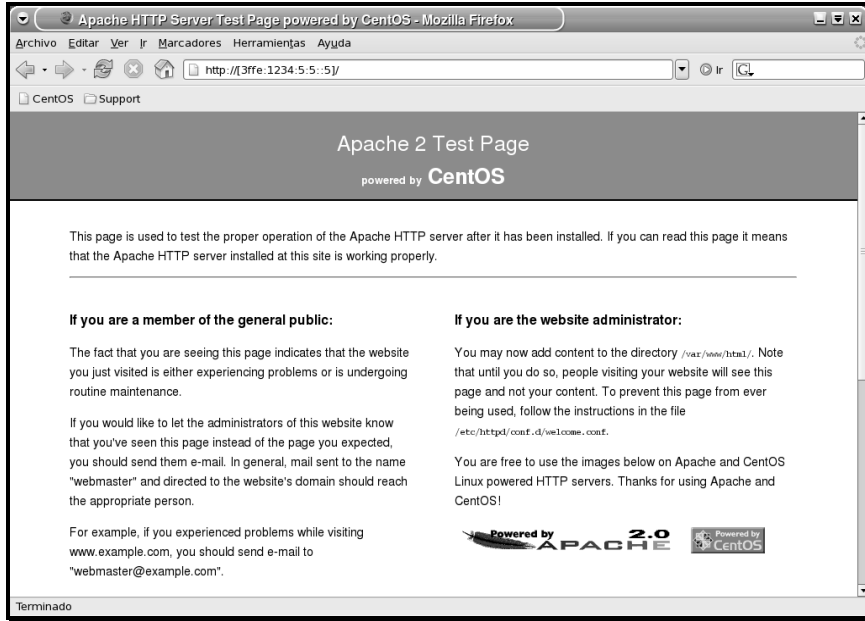


Fig. II.22. Verificación IPv6 en navegador web.

Usando la herramienta Ethereal se procedió a la captura de paquetes IPv6 al momento que se efectuaban las pruebas con Pila Dual, ver Fig. II.23.

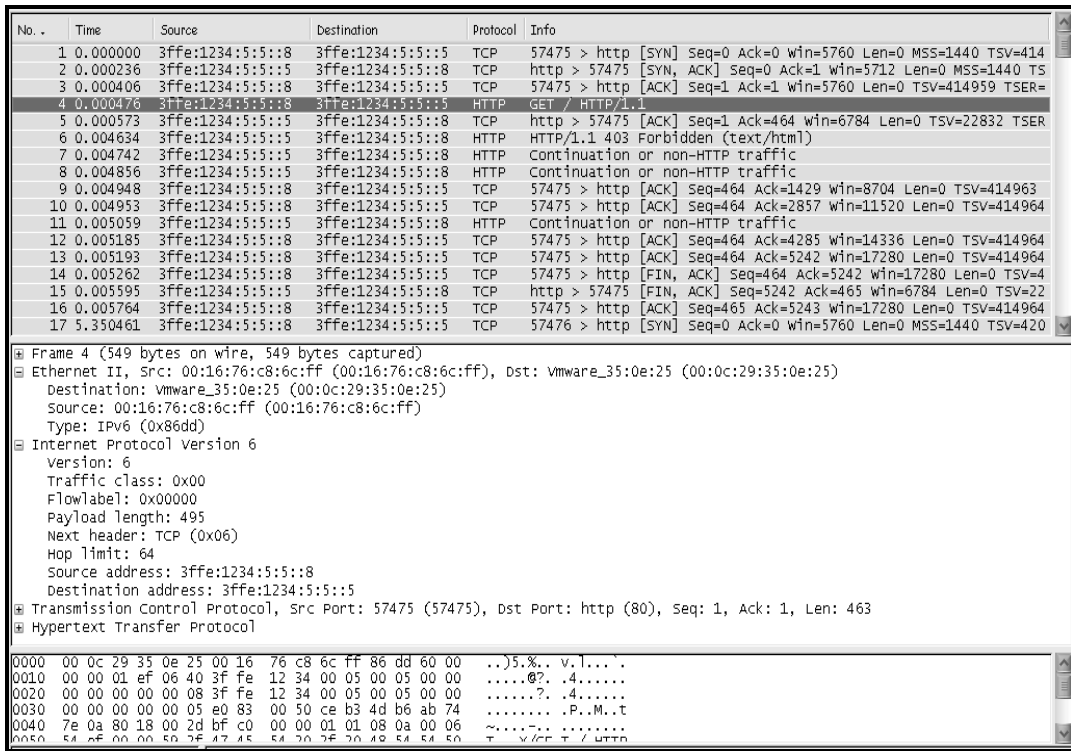


Fig. II.23. Captura de paquetes IPV6 durante pruebas con navegador web

Para la realización de esta prueba se debió tener en cuenta las siguientes consideraciones: la asignación de una dirección IPv6 válida, además de tener el respectivo soporte IPv6 en el navegador.

### b. Envío Paquetes ICMPv6

La siguiente prueba consiste en enviar paquetes ICMPv6 entre 2 hosts que tengan configurado Pila Dual, de la siguiente manera:

**PC 1: Dir. IPv6 3ffe:1234:5:5::5**

**PC 2: Dir. IPv6 3ffe:1234:5:5::8**

Luego se procede a realizar un Ping6 para verificar la conectividad en ambos hosts. El nodo emisor envía mensajes del tipo “*Echo Request*” al destino especificado y si este se encuentra disponible envía de vuelta mensajes del tipo “*Echo Reply*”. Cuyos resultados se aprecian en la **Fig. II.24**

```
[root@localhost ~]# ping6 -I eth0 3ffe:1234:5:5::5
PING 3ffe:1234:5:5::5(3ffe:1234:5:5::5) from 3ffe:1234:5:5::8 eth0: 56 data bytes
64 bytes from 3ffe:1234:5:5::5: icmp_seq=0 ttl=64 time=0.339 ms
64 bytes from 3ffe:1234:5:5::5: icmp_seq=1 ttl=64 time=0.291 ms
64 bytes from 3ffe:1234:5:5::5: icmp_seq=2 ttl=64 time=0.343 ms
64 bytes from 3ffe:1234:5:5::5: icmp_seq=3 ttl=64 time=0.386 ms
64 bytes from 3ffe:1234:5:5::5: icmp_seq=4 ttl=64 time=0.386 ms
64 bytes from 3ffe:1234:5:5::5: icmp_seq=5 ttl=64 time=0.360 ms
64 bytes from 3ffe:1234:5:5::5: icmp_seq=6 ttl=64 time=0.341 ms
64 bytes from 3ffe:1234:5:5::5: icmp_seq=7 ttl=64 time=0.503 ms
64 bytes from 3ffe:1234:5:5::5: icmp_seq=8 ttl=64 time=0.344 ms
64 bytes from 3ffe:1234:5:5::5: icmp_seq=9 ttl=64 time=0.353 ms
64 bytes from 3ffe:1234:5:5::5: icmp_seq=10 ttl=64 time=0.395 ms
64 bytes from 3ffe:1234:5:5::5: icmp_seq=11 ttl=64 time=0.357 ms
64 bytes from 3ffe:1234:5:5::5: icmp_seq=12 ttl=64 time=0.350 ms
64 bytes from 3ffe:1234:5:5::5: icmp_seq=13 ttl=64 time=0.395 ms
--- 3ffe:1234:5:5::5 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 12997ms
rtt min/avg/max/mdev = 0.291/0.367/0.503/0.048 ms, pipe 2
[root@localhost ~]#
```

**Fig. II.24.** Conexión mediante ping6 en Pila Dual

Usando la herramienta ethereal se procedió a la captura de paquetes ICMPv6, para verificar la conectividad del protocolo IPv6, dicha captura se la muestra en la **Fig. II.25**.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.10.10	192.168.10.255	BROWSE	Domain/workgroup Announcement LAN, NT workstation, Domain Enum
2	18.832857	3ffe:1234:5:5::5	3ffe:1234:5:5::8	ICMPv6	Echo request
3	18.834410	3ffe:1234:5:5::8	ff02::1:ff00:5	ICMPv6	Neighbor solicitation
4	18.835401	3ffe:1234:5:5::5	3ffe:1234:5:5::8	ICMPv6	Neighbor advertisement
5	18.837327	3ffe:1234:5:5::8	3ffe:1234:5:5::5	ICMPv6	Echo reply
6	20.395084	3ffe:1234:5:5::5	3ffe:1234:5:5::8	ICMPv6	Echo request
7	20.395267	3ffe:1234:5:5::8	3ffe:1234:5:5::5	ICMPv6	Echo reply
8	22.059898	3ffe:1234:5:5::5	3ffe:1234:5:5::8	ICMPv6	Echo request
9	22.060085	3ffe:1234:5:5::8	3ffe:1234:5:5::5	ICMPv6	Echo reply
10	22.706384	3ffe:1234:5:5::8	3ffe:1234:5:5::5	ICMPv6	Echo request
11	22.706536	3ffe:1234:5:5::5	3ffe:1234:5:5::8	ICMPv6	Echo reply
12	23.705334	3ffe:1234:5:5::5	3ffe:1234:5:5::8	ICMPv6	Echo request
13	23.705520	3ffe:1234:5:5::8	3ffe:1234:5:5::5	ICMPv6	Echo reply
14	23.706474	3ffe:1234:5:5::8	3ffe:1234:5:5::5	ICMPv6	Echo request
15	23.706502	3ffe:1234:5:5::5	3ffe:1234:5:5::8	ICMPv6	Echo reply

# Frame 6 (118 bytes on wire (118 bytes captured))  
# Ethernet II, Src: Vmware\_35:0e:25 (00:0c:29:35:0e:25), Dst: 00:16:76:c8:6c:ff (00:16:76:c8:6c:ff)  
# Internet Protocol Version 6  
  Version: 6  
  Traffic class: 0x00  
  Flow label: 0x00000  
  Payload length: 64  
  Next header: ICMPv6 (0x3a)  
  Hop limit: 64  
  Source address: 3ffe:1234:5:5::5  
  Destination address: 3ffe:1234:5:5::8  
# Internet Control Message Protocol v6  
  Type: 128 (Echo request)  
  Code: 0  
  Checksum: 0xc819 [correct]  
  ID: 0x7312  
  Sequence: 0x0001  
  Data (56 bytes)

```
0000 00 16 76 c8 6c ff 00 0c 29 35 0e 25 86 dd 60 00  ..v.1... }5%..
0010 00 00 00 40 3a 40 3f fe 12 34 00 05 00 05 00 00  ..@:0?..4.....
0020 00 00 00 00 00 05 3f fe 12 34 00 05 00 05 00 00  .....?..4.....
0030 00 00 00 00 00 08 80 00 c8 19 73 12 00 01 9a a4  .....5.....
0040 d5 47 42 e3 02 00 08 09 0a 0b 0c 0d 0e 0f 10 11  .GB.....
0050 13 12 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21  |
```

Fig. II.25. Captura de paquetes ICMPv6 usando Pila Dual

### 3.3.1.2. TÚNEL

El túnel es un mecanismo en el que un paquete es encapsulado, dentro de otro tipo de paquete. Es decir podemos encapsular paquetes IPv6 dentro de paquetes IPv4. Es la forma más sencilla de configurar una conexión IPv6 a través de una red IPv4, aunque no es fácil de administrar. La mayoría de hosts doble pila y elementos de red soportan el estándar IPv6 en túneles IPv4

#### CONFIGURACIÓN DE TÚNEL MANUAL EN LINUX CentOS 5

Para la configuración de este tipo de túneles es necesario tener los extremos configurados el mecanismo de Pila Dual que permita la conectividad entre IPv4 e IPv6, de una manera simultánea.

Los pasos para la creación de un túnel manual son los siguientes:

- Crear un nuevo dispositivo de tunel (especificando un TTL porque el valor por defecto es 0):

**Sintaxis:**

```
# /sbin/ip tunnel add <nombreTunel> mode sit ttl <ttdefault> remote  
<dirIPv4Remota> local <dirIPv4Local>
```

**Ejemplo:**

```
# /sbin/ip tunnel add tunelPrueba mode sit ttl 80 remote 200.200.10.1 local  
192.1.2.3
```

- Levantamos la interfaz.

**Sintaxis:**

```
# ifconfig <nombreTunel> up
```

**Ejemplo:**

```
# ifconfig tunelPrueba up
```

- Agregar una dirección IPv6 al túnel creado

**Sintaxis:**

```
# /sbin/ip -6 addr add <dirIPv6>/prefijo dev <nombreTunel>
```

**Ejemplo:**

```
# /sbin/ip -6 addr add 3ffe:1234:5:6::1/64 dev tunelPrueba
```

## ESCENARIO DE PRUEBAS

Para la configuración de un túnel manual se ha efectuado el siguiente escenario:

*Dos host que pertenecen a 2 redes IPv4 distintas, el primer host pertenece a la red 192.168.10.0/24 y el segundo host pertenece a la red 200.192.10.0/24. Ambos host tienen habilitado el mecanismo de pila dual como mecanismo de convivencia entre IPv4 e IPv6, el esquema gráfico se lo especifica en la **Fig. II.26.***

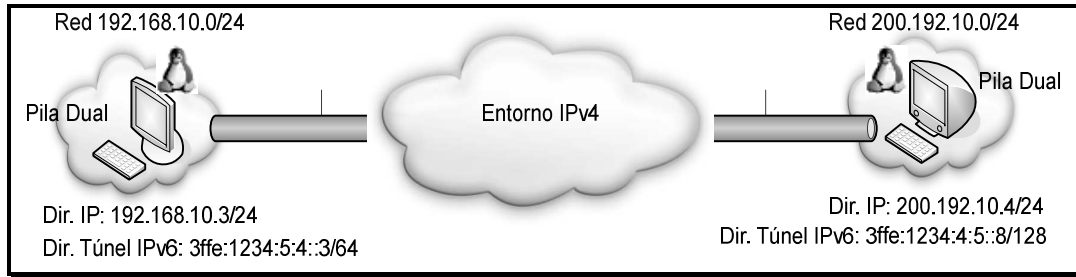


Fig. II.26 Escenario para túnel manual

Como se trata de 2 redes distintas se procede a realizar un enrutamiento estático para que exista conectividad en ambas redes, esto se lo efectúa mediante el siguiente comando:

**Sintaxis:**

```
# route add -net <red_a_conectar> netmask <dirMáscara> gw <dirIPv4Gateway>
```

**Ejemplo:** Desde la red 200.192.10.0/24 se desea tener conectividad con la red 200.200.10.0/24

```
# route add -net 192.168.10.0 netmask 255.255.255.0 gw 200.200.10.4
```

Configurado el enrutamiento estático entre las 2 redes, verificamos si existe o no conectividad entre ellas.

Desde la red 200.192.10.0 hacia la red 192.168.10.0, ver Fig. II.27

```
lroot@localhost ~]# ping 192.168.10.3
PING 192.168.10.3 (192.168.10.3) 56(84) bytes of data.
64 bytes from 192.168.10.3: icmp_seq=1 ttl=64 time=1.39 ms
64 bytes from 192.168.10.3: icmp_seq=2 ttl=64 time=0.345 ms
64 bytes from 192.168.10.3: icmp_seq=3 ttl=64 time=0.348 ms
64 bytes from 192.168.10.3: icmp_seq=4 ttl=64 time=0.344 ms
```

Fig. II.27 Interconectividad entre redes IPv4 distintas

Desde la red 192.168.10.0 hacia la red 200.192.10.0, ver Fig. II.28.

```
[root@localhost ~]# route add -net 200.192.10.0 netmask 255.255.255.0 gw 192.168.10.3
[root@localhost ~]# ping 200.192.10.4
PING 200.192.10.4 (200.192.10.4) 56(84) bytes of data:
64 bytes from 200.192.10.4: icmp_seq=1 ttl=64 time=1.01 ms
64 bytes from 200.192.10.4: icmp_seq=2 ttl=64 time=0.749 ms
64 bytes from 200.192.10.4: icmp_seq=3 ttl=64 time=0.689 ms
64 bytes from 200.192.10.4: icmp_seq=4 ttl=64 time=1.00 ms
64 bytes from 200.192.10.4: icmp_seq=5 ttl=64 time=0.956 ms
64 bytes from 200.192.10.4: icmp_seq=6 ttl=64 time=0.991 ms

--- 200.192.10.4 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5005ms
rtt min/avg/max/mdev = 0.689/0.901/1.018/0.137 ms
[root@localhost ~]#
```

Fig. II.28 Interconectividad entre redes IPv4 distintas

Una vez verificada la conectividad en ambas redes IPv4 procedemos a la creación del túnel IPv6 sobre IPv4, no sin antes verificar si existe algún túnel creado anteriormente. Mediante el uso del comando *ifconfig*, ver Fig. II.29.

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:16:76:C8:6C:FF
          inet addr:200.192.10.4  Bcast:200.192.10.255  Mask:255.255.255.0
          inet6 addr: fe80::216:76ff:fec8:6cff/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:760 errors:0 dropped:0 overruns:0 frame:0
          TX packets:781 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:319998 (312.4 KiB)  TX bytes:132875 (129.7 KiB)
          Base address:0x20c0 Memory:50300000-50320000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1556 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1556 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5333073 (5.0 MiB)  TX bytes:5333073 (5.0 MiB)

[root@localhost ~]#
```

Fig. II.29. Interfaces activas para configurar túneles

Una vez verificado que no existe túneles creados procedemos a la creación de un nuevo túnel.

**Extremo 1 del Túnel, host perteneciente a la red 200.192.10.0/24**

- Creación de nuevo túnel

```
# /sbin/ip tunnel add mytunnel mode sit ttl 12 remote 192.168.10.3 local 200.192.10.4
```

- Levantamos la interfaz del túnel creado



```
# ifconfig mytunnel up
```

Asignamos una dirección IPv6 válida al túnel creado

```
# ip -6 addr add 3ffe:1234:4:5::8 dev mytunnel
```

Creamos rutas estáticas entre las redes IPv6

```
# ip -6 route add ::/0 dev mytunnel metric 1
```

En la **Fig. II.30** se resumen los pasos a seguir para configurar un túnel

```
[root@localhost ~]# ip tunnel add mytunnel mode sit ttl 12 remote 192.168.10.3 local 200.192.10.4
[root@localhost ~]# ifconfig mytunnel up
[root@localhost ~]# ip -6 route add ::/0 dev mytunnel
[root@localhost ~]# ip -6 addr add 3ffe:1234:4:5::8 dev mytunnel
[root@localhost ~]#
```

**Fig. II.30.** Pasos para configurar extremo 1 del túnel

En la **Fig. II.31** se verifica la creación del nuevo túnel denominado **mytunnel**

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:16:76:C8:6C:FF
          inet addr:200.192.10.4  Bcast:200.192.10.255  Mask:255.255.255.0
          inet6 addr: fe80::216:76ff:fec8:6cff/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1069 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1102 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:518304 (506.1 KiB)  TX bytes:181347 (177.0 KiB)
          Base address:0x20c0  Memory:50300000-50320000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1556 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1556 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5333073 (5.0 MiB)  TX bytes:5333073 (5.0 MiB)

mytunnel  Link encap:IPv6-in-IPv4
          inet6 addr: fe80::c8c0:a04/128 Scope:Link
          inet6 addr: 3ffe:1234:4:5::8/128 Scope:Global
          UP POINTOPOINT RUNNING NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

[root@localhost ~]#
```

**Fig. II.31.** Verificación túnel activo en extremo 1

### **Extremo Túnel, host perteneciente a la red 192.168.10.0/24**

Creación de nuevo túnel

```
# /sbin/ip tunnel add mitun mode sit ttl 12 remote 200.192.10.4 local 192.168.10.3
```

Levantamos la interfaz del túnel creado

```
# ifconfig mitun up
```

Asignamos una dirección IPv6 válida al túnel creado

```
# ip -6 addr add 3ffe:1234:5:4::3/64 dev mitun
```

Creamos rutas estáticas entre las redes IPv6

```
# ip -6 route add ::/0 dev mytunnel metric 1
```

En la **Fig. II.32** se resumen los pasos a seguir para configurar un túnel

```
[root@localhost ~]# ip tunnel add mitun mode sit ttl 12 remote 200.192.10.4 local 192.168.10.3
[root@localhost ~]# ifconfig mitun up
[root@localhost ~]# ip -6 route add ::/0 dev mitun
[root@localhost ~]# ip -6 addr add 3ffe:1234:5:4::3/64 dev mitun
[root@localhost ~]#
```

**Fig. II.32.** Pasos para configurar extremo 2 del túnel

En la **Fig. II.33** se verifica la creación del nuevo túnel denominado **mitun**

```
eth1      Link encap:Ethernet  HWaddr 00:0C:29:35:0E:25
          inet addr:192.168.10.3  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe35:e25/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:920 errors:0 dropped:0 overruns:0 frame:0
          TX packets:989 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:147706 (144.2 KiB)  TX bytes:489483 (478.0 KiB)
          Interrupt:169 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1898 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1898 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3327056 (3.1 MiB)  TX bytes:3327056 (3.1 MiB)

mitun     Link encap:IPv6-in-IPv4
          inet6 addr: 3ffe:1234:5:4::3/64 Scope:Global
          inet6 addr: fe80::c0a8:a03/128 Scope:Link
          UP POINTOPOINT RUNNING NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

[root@localhost ~]#
```

**Fig. II.33.** Verificación túnel activo en extremo 2

Una vez configurados los extremos del túnel procedemos a verificar si existe conectividad entre los 2 extremos del túnel, cuyos resultados se muestran en la

**Fig. II.34:**

```
[root@localhost ~]# ping6 -I mytunnel 3ffe:1234:5:4::3 -c 4
PING 3ffe:1234:5:4::3(3ffe:1234:5:4::3) from 3ffe:1234:4:5::8 mytunnel: 56 data bytes
64 bytes from 3ffe:1234:5:4::3: icmp_seq=0 ttl=64 time=0.520 ms
64 bytes from 3ffe:1234:5:4::3: icmp_seq=1 ttl=64 time=0.412 ms
64 bytes from 3ffe:1234:5:4::3: icmp_seq=2 ttl=64 time=0.371 ms
64 bytes from 3ffe:1234:5:4::3: icmp_seq=3 ttl=64 time=0.397 ms

--- 3ffe:1234:5:4::3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.371/0.425/0.520/0.056 ms, pipe 2
[root@localhost ~]#
```

Fig. II.34. Interconectividad en los extremos del túnel

## PRUEBAS REALIZADAS

Para asegurarnos del correcto funcionamiento del túnel, se realizó pruebas simples, para verificar el desempeño de este mecanismo de convivencia:

- Uso del navegador para acceder a direcciones IPv6
- Envío de paquetes ICMPv6

### a. Uso del navegador para acceder al extremo del túnel

Para verificar la conectividad entre los extremos del túnel, procedemos a colocar la dirección IPv6 válida entre corchetes en el navegador Web de cualquier extremo del túnel configurado, ver Fig. II.35.

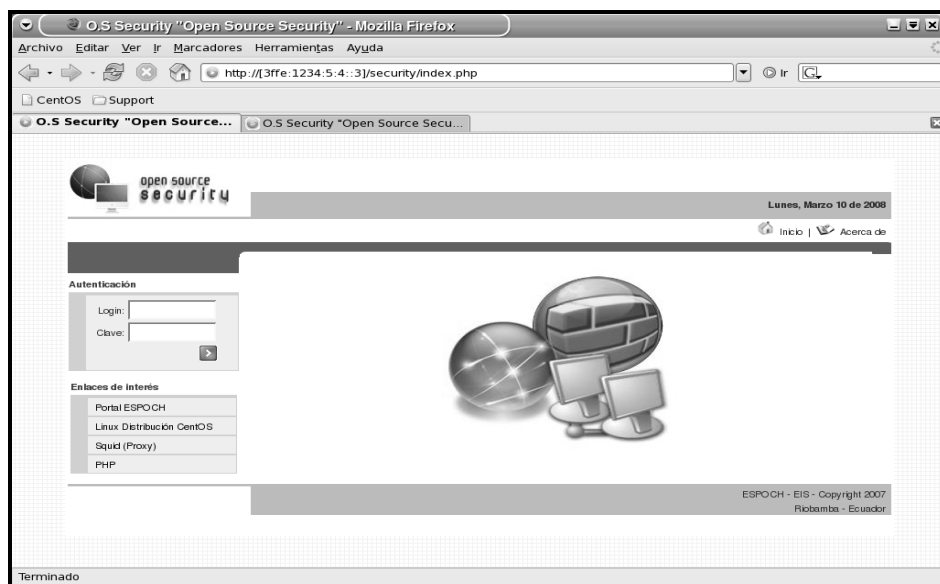
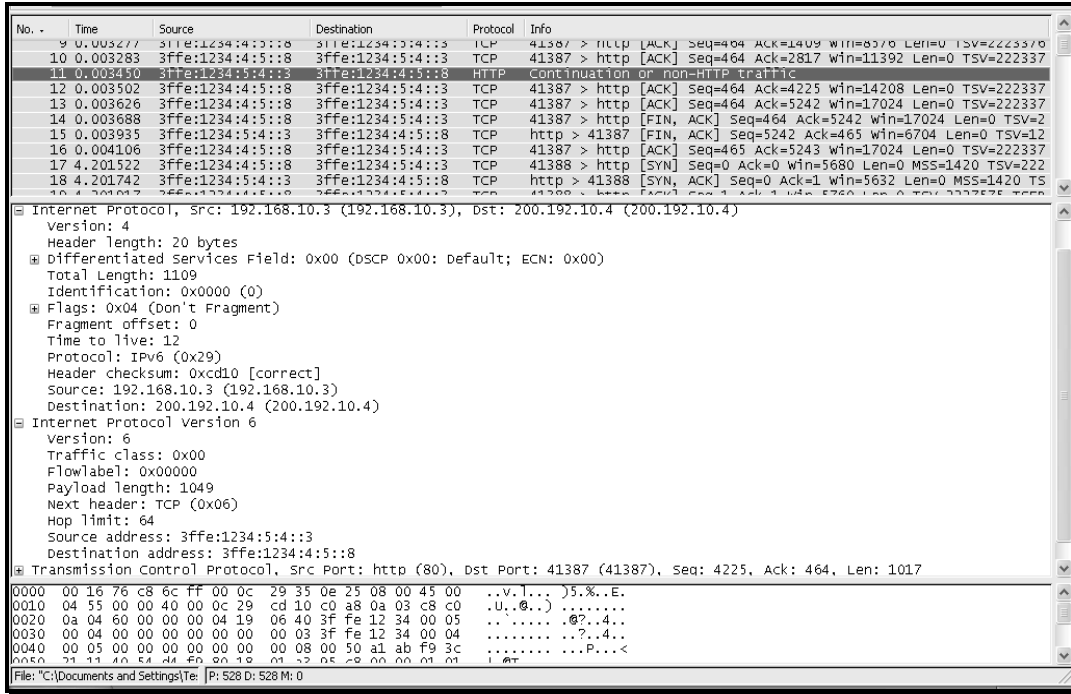


Fig. II.35. Acceso extremo túnel usando navegador

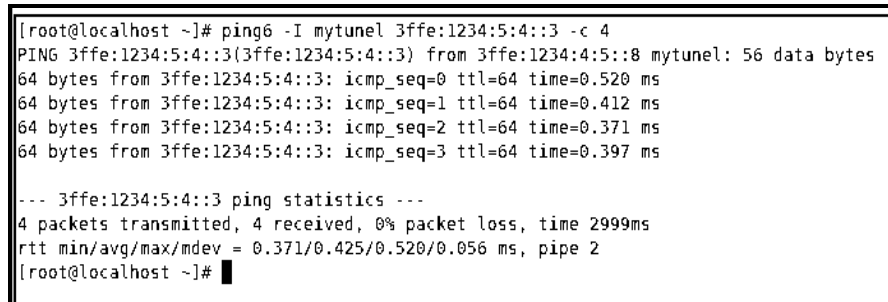
Complementariamente, durante la realización de las pruebas se procedió a la captura de paquetes IPv6, verificando de esta manera la conectividad existente en los extremos de túnel creado, dicha captura se encuentra en la **Fig. II.36**.



**Fig. II.36.** Captura de paquetes IPv6 durante acceso web con direcciones IPv6

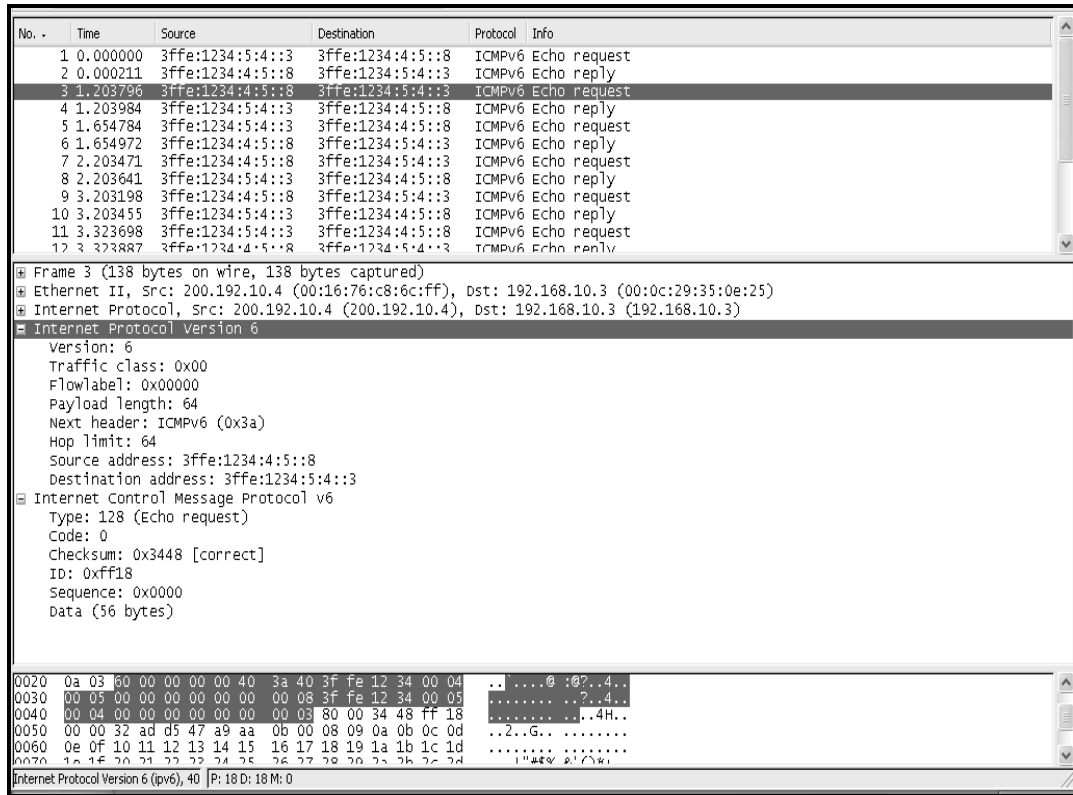
### b. Envío de paquetes ICMPv6

Esta prueba permite verificar la conectividad existente en los extremos del túnel creado, mediante la ejecución del comando ping6, ver **Fig. II.37**.



**Fig. II.37.** Interconectividad en los extremos del túnel

Adicionalmente, durante la ejecución de esta prueba se procedió a la captura de paquetes ICMPv6, para analizar la estructura de dichos paquetes y comprobar la conectividad existente, dicha captura se muestra en la **Fig. II.38**



**Fig. II.38.** Captura de paquetes durante interconectividad.

### 3.3.1.3. TRADUCCIÓN DE DIRECCIONES

Este mecanismo de transición permite a un nodo que solo cuenta con el stack IPv6 habilitado dentro de una red IPv6 comunicarse con otro nodo que solo tiene el stack IPv4 habilitado dentro de una red IPv4. La cabecera IP ha de ser convertida y puede ser requerido un pool de direcciones IPv4 para proporcionar un alias al host IPv6 durante la comunicación.

Sin embargo, ésta técnica requiere tener también habilitados mecanismos de traducción entre IPv4 e IPv6 en las orillas de ambas redes (enrutadores). La

conversión será más compleja si la aplicación procesa las direcciones IP. La principal desventaja es que todo el peso de este mecanismo de transición recaerá en los dispositivos.

Para hacer uso de este mecanismo de transición es necesario instalar un software específico que permita realizar las configuraciones respectivas. No obstante, el software requerido para realizar las pruebas en Linux CentOS 5 usando Traducción de Direcciones, no está disponible para esta distribución debido a que es una versión nueva de Linux. El software que permite realizar Traducción de Direcciones tiene sus versiones hasta Fedora Core 4. Por tal motivo, no es posible realizar las respectivas configuraciones y pruebas usando Traducción de Direcciones.

Por tal motivo, para el posterior Análisis y Selección de la Técnica de Convivencia más adecuada no será tomada en cuenta el mecanismo Traducción de Direcciones, por lo tanto el Análisis Comparativo será realizado entre los mecanismos: Pila Dual y Túneles.

### **3.3.2. COMPARACIÓN DE LAS TÉCNICAS Y SELECCIÓN DE LA MÁS IDÓNEA**

La transición no siempre es la solución. Es importante tener presente que la transición no es la solución a todos los problemas. De hecho, algunas aplicaciones innovadoras necesitan IPv6 para su despliegue masivo.

Desplegar mecanismos de transición a gran escala puede además implicar problemas de escalabilidad que podrían limitar enormemente el rendimiento de IPv6 en comparación una solución nativa.

### **Coexistencia de IPv4 e IPv6.**

Es esencial mantener bajo control la transición para evitar el despliegue de dos infraestructuras Internet paralelas. Las aplicaciones IPv6 se beneficiarán de las enormes inversiones realizadas en el despliegue de redes IPv4.

### **Continuidad del servicio.**

La transición de IPv4 a IPv6 no es sólo una cuestión de direcciones o de routing. Cualquiera que sea la infraestructura habrá que proporcionar progresivamente los nuevos servicios IPv4, como IP QoS, seguridad, etc.

## **3.3.2.1. DETERMINACIÓN DE LOS PARÁMETROS A EVALUAR**

### **CONFIGURACIÓN**

Cada técnica de transición tiene su propia manera de estructurar sus procedimientos de comunicación con los dispositivos que soporten la utilización de un determinado mecanismo, que varía según el Sistema Operativo. Esta información ha sido recopilada en los respectivos RFC's que describen los pasos a seguir para lograr la integración de los protocolos IPv4 e IPv6.

### **COMPATIBILIDAD (HARDWARE Y SOFTWARE)**

#### **Hardware**

Debido al avance tecnológico, los equipos de última generación incorporan funcionalidades que facilitan la configuración de las técnicas de convivencia, es decir, soportan la utilización del Protocolo IPv4 e IPv6 simultáneamente.

#### **Software**

Los Sistemas Operativos actuales incorporan el soporte necesario en sus núcleos, para facilitar la configuración de las técnicas de convivencia más idónea.

## **INTEGRIDAD**

La integridad del flujo de información garantiza que la misma no ha sido modificada durante la transmisión de los paquetes, por medio de los mecanismos de transición.

## **INTEROPERABILIDAD**

Es la capacidad de comunicarse, ejecutar programas o transferir datos entre distintas unidades funcionales de forma que se requiera el mínimo o nulo conocimiento del usuario sobre las características particulares de dichas unidades.

Este concepto ha adquirido gran trascendencia porque la penetración de Internet a nivel universal ha hecho que se convierta en una importante necesidad la interacción entre todos los sitios conectados a la red de redes, en la actualidad se esta dando una progresiva migración del protocolo IPv4 hacia IPv6 y es necesario encontrar el mecanismo de transición que cumpla esta tarea de una manera efectiva.

## **DESEMPEÑO**

Hace referencia al comportamiento que tiene un mecanismo de convivencia específico, una vez que cumple con todos los argumentos y/o especificaciones establecidas para su utilización. Permitiendo de esta manera, comprobar su funcionalidad en entornos reales.



### 3.3.2.2. ANÁLISIS COMPARATIVO ENTRE LAS TÉCNICAS DE CONVIVENCIA

A continuación se presenta la matriz de valores, ver **Tabla II.29**; bajo la cual se procede a evaluar los mecanismos de convivencia IPv4/IPv6. Que permita seleccionar el mecanismo más adecuado:

**Tabla II.29.** Matriz de evaluación de criterios.

<b>Factor</b>	<b>Escala</b>	<b>Interpretación</b>
Óptimo	5	Los argumentos establecidos se cumplen en su totalidad.
Satisfactorio	4	La mayoría de los argumentos establecidos se cumplen.
Aceptable	3	La mitad de los argumentos establecidos se cumplen.
Regular	2	Ciertos argumentos son cumplidos parcialmente.
Inaplicable	1	No se cumple con ningún argumento establecido.

#### **EVALUACIÓN – REQUERIMIENTOS**

Para el análisis a efectuarse, se debe tomar en cuenta varios factores:

- La información documentada sobre cada mecanismo, que permita una configuración adecuada del mismo.
- Los inconvenientes surgidos durante las pruebas de configuración y uso de cada mecanismo de convivencia, en los escenarios de prueba.
- Infraestructura física que permitió la realización de las respectivas pruebas de verificación.

Para la selección de la técnica de convivencia más adecuada se analiza los mecanismos Pila Dual y Túneles, debido que ambos mecanismos permitieron efectuarse sus respectivas configuraciones y pruebas en Linux CentOS 5, no así el mecanismo Traducción de Direcciones, debido a que este mecanismo requiere la instalación de software especializado que permita su configuración y pruebas respectivas, dicho software no está disponible para Linux CentOS 5.

### EVALUACIÓN – PARAMETRIZACIÓN

Ahora se procede a la evaluación de cada parámetro con los mecanismos de convivencia seleccionados (Pila Dual y Túnel), con la finalidad de efectuar un análisis específico que permita una evaluación objetiva, tomando como referencia la **Matriz de Evaluación de Criterios (Tabla II.29)** y los escenarios de pruebas realizados para verificar su funcionamiento.

#### PARÁMETRO 1: CONFIGURACIÓN

Este parámetro evalúa, si los procedimientos empleados por cada mecanismo de convivencia son los adecuados para su funcionamiento, ver **Tabla II.30**.

#### Evaluación Parcial

**Tabla II.30.** Evaluación Configuración

Técnica	Factor	Escala	Justificación
Pila Dual	Óptimo	5	No presenta complicaciones de configuración, puesto que su funcionamiento consiste en tener soporte IPv6 en el kernel para su utilización.
Túnel	Satisfactorio	4	Este mecanismo esta relacionado con Pila Dual puesto que su configuración necesita que los equipos que actúen como extremos del túnel tengan soporte IPv4 e IPv6 simultáneamente. Caso contrario, no es posible su configuración.

## PARÁMETRO 2: COMPATIBILIDAD HARDWARE

Este parámetro permite evaluar si existe algún tipo de complicación en el hardware, al usar un mecanismo de convivencia IPv4 e IPv6, ver **Tabla II.31**.

### Evaluación Parcial

**Tabla II.31.** Evaluación Compatibilidad Hardware

Técnica	Factor	Escala	Justificación
Pila Dual	Óptimo	5	No hay complicaciones en este método, ya que los dispositivos de última generación cuentan con el soporte necesario para los protocolos IPv4 e IPv6.
Túnel	Óptimo	5	No hay complicaciones en este método, ya que los dispositivos de última generación cuentan con el soporte necesario para los protocolos IPv4 e IPv6.

## PARÁMETRO 3: COMPATIBILIDAD SOFTWARE

Este parámetro permite evaluar si existe algún tipo de complicación respecto al software, cuando se usa un mecanismo de convivencia IPv4 e IPv6, ver **Tabla II.32**.

### Evaluación Parcial

**Tabla II.32.** Evaluación Compatibilidad Software

Técnica	Factor	Escala	Justificación
Pila Dual	Óptimo	5	Tomando como referencia el uso de versiones actualizadas de sistemas operativos que brinden soporte hacia IPv6, el uso de este mecanismo de convivencia es factible.
Túnel	Óptimo	5	Tomando como referencia el uso de versiones actualizadas de sistemas operativos que brinden soporte hacia IPv6, el uso de este mecanismo de convivencia es factible.

#### PARÁMETRO 4: INTEGRIDAD

Se encarga de evaluar que durante el uso de un mecanismo de convivencia, la información que se transmite por dicho mecanismo no sufre ningún tipo de modificación, ver **Tabla II.33**.

##### Evaluación Parcial

**Tabla II.33.** Evaluación Integridad

Técnica	Factor	Escala	Justificación
Pila Dual	Óptimo	5	Los equipos que tengan habilitado Pila Dual, podrán transmitir información entre sí, sin ningún tipo de problema. Asegurando la integridad en los datos que se envíen.
Túnel	Satisfactorio	4	Debido a que este mecanismo hace uso de períodos de tiempo para mantener su conectividad activa. Puede darse el caso, en que durante el envío de paquetes, el tiempo de actividad para el túnel se termine, perdiéndose la información que se estaba transmitiendo.

#### PARÁMETRO 5: INTEROPERABILIDAD

Permite evaluar si al hacer uso de un mecanismo de transición determinado, permite la comunicación con dispositivos de red distintos que tengan el soporte necesario a los protocolos IPv4 e IPv6, ver **Tabla II.34**.

##### Evaluación Parcial

**Tabla II.34.** Evaluación Interoperabilidad

Técnica	Factor	Escala	Justificación
Pila Dual	Óptimo	5	Cualquier dispositivo de red administrable que tenga el soporte debido a los protocolos IPv4 e IPv6 puede hacer uso de este mecanismo.
Túnel	Óptimo	5	Cualquier dispositivo de red administrable que tenga el soporte debido a los protocolos IPv4 e IPv6 puede hacer uso de este mecanismo.

### PARÁMETRO 6: DESEMPEÑO

Se encarga de evaluar el comportamiento que tiene un mecanismo de transición, durante su utilización, ver **Tabla II.35**.

#### Evaluación Parcial

**Tabla II.35.** Evaluación Rendimiento

Técnica	Factor	Escala	Justificación
Pila Dual	Satisfactorio	4	Dado que IPv4 e IPv6 son protocolos distintos, la información que se transmite no afecta su desempeño, sin embargo se debe tener en cuenta que este mecanismo debe verificar el tipo de dirección que se usa; es decir, si se trata de una dirección IPv4 o IPv6.
Túnel	Satisfactorio	4	Dado que este mecanismo se encarga de encapsular y desencapsular paquetes IPv6 en paquetes IPv4 y viceversa. El tiempo usado en realizar esta tarea, influye en cierta manera en su desempeño.

### 3.3.2.3. RESULTADOS

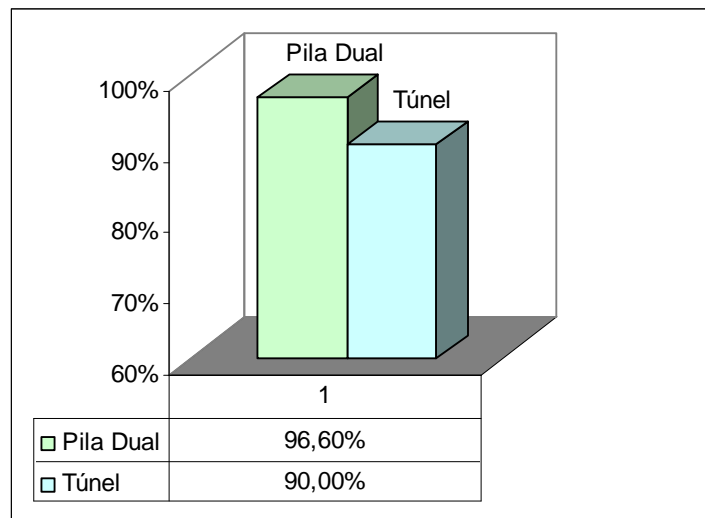
A partir del análisis efectuado a los Parámetros de Evaluación (Configuración, Compatibilidad Hardware y Software, Integridad, Interoperabilidad y Desempeño) con los mecanismos de convivencia seleccionados (Pila Dual y Túnel), se efectuaron evaluaciones parciales por cada uno (**ver Tabla II.30, Tabla II.31, Tabla II.32, Tabla II.33, Tabla II.34, Tabla II.35**), permitiendo un análisis más detallado de dichos parámetros con los mecanismos Pila Dual y Túnel.

Los resultados parciales se los agrupa en la **Tabla II.36**; con la finalidad de obtener un promedio total de los parámetros evaluados para cada mecanismo

de convivencia (Pila Dual y Túnel) analizado, permitiendo de esta manera obtener sus porcentajes totales de operabilidad respectivos, ver **Fig. II.39**.

**Tabla II.36.** Resultados Totales Comparación

		<b>Técnicas de Convivencia Evaluadas</b>	
		Pila Dual	Túnel
<b>Parámetros de Comparación</b>	Configuración	5	4
	Compatibilidad Hardware	5	5
	Compatibilidad Software	5	5
	Integridad	5	4
	Interoperabilidad	5	5
	Desempeño	4	4
	<b>Promedio General:</b>	<b>4.83</b>	<b>4.5</b>
	<b>Porcentaje Total:</b>	<b>96.6%</b>	<b>90%</b>



**Fig. II.39.** Porcentajes entre Comparación de Mecanismos

Los resultados finales obtenidos de la evaluación, son los siguientes:

**Pila Dual** presenta un porcentaje de operabilidad del **96.6%** mientras que el mecanismo de **Túnel** presenta un porcentaje del **90%**. Dichos porcentajes se

los obtuvo del promedio general obtenido por cada mecanismo, realizando una aproximación con el valor de escala más alto.

Con los resultados obtenidos se deduce lo siguiente:

- Las evaluaciones parciales efectuadas a cada parámetro, permitieron establecer las condiciones de operación bajo las cuales cada mecanismo se desenvuelve.
- Los argumentos requeridos para la utilización de ambos mecanismos de convivencia, cumplen con los requerimientos establecidos. Sin embargo, la diferencia existente en los porcentajes totales de ambos mecanismos, se debe a las evaluaciones parciales realizadas, en las cuales se presentaron ciertos argumentos que diferencian un mecanismo del otro. Por ejemplo, en la integridad de los datos el mecanismo Túnel marca una diferencia respecto a Pila Dual, debido a que su utilización se basa en períodos de tiempo y si dicho período de tiempo expira, los datos que se transmiten en ese momento tienen algún grado de alteración. Con respecto a la Configuración de cada mecanismo; Pila Dual no presenta inconvenientes, debido que para su utilización basta con tener el soporte IPv6 en el kernel, mientras que el mecanismo Túnel debe incorporar equipos con Pila Dual en sus extremos, para efectuar posteriormente la configuración que el mecanismo túneles usa. Por estas razones, ambos mecanismos se distinguen uno del otro, de ahí la diferencia existente en los porcentajes finales entre ambos mecanismos de convivencia evaluados.
- La utilización del mecanismo de convivencia más adecuado, va enfocado al propósito de incorporar dicho mecanismo en una herramienta que provea el soporte necesario para los protocolos IPv4 e IPv6 simultáneamente. Por esta razón y basándonos en los resultados finales de la comparación

efectuado entre Pila Dual y Túneles, podemos establecer que el mecanismo más adecuado para nuestros requerimientos es **Pila Dual**, debido a que en el análisis efectuado en las evaluaciones parciales, se demuestra un mejor desempeño de Pila Dual frente al mecanismo de Túneles.

- Como una acotación final, vale mencionar que la utilización de Pila Dual es óptimo para entornos donde se conoce explícitamente la cantidad de equipos que se usa. Mientras que el uso de Túneles, permite la conectividad entre entornos que se encuentran separados y que se desea la comunicación entre ellos. Como nuestro estudio se enfoca en administrar entornos pequeños como una LAN, Pila Dual cumple con esta conectividad satisfactoriamente, proporcionando el soporte necesario para los protocolos IPv4 e IPv6 al mismo tiempo.



---

## CAPÍTULO IV

# DESARROLLO DE LA APLICACIÓN WEB “OPEN SOURCE SECURITY”

---

En este capítulo se detalla los aspectos generales para el Análisis, Diseño y Desarrollo efectuadas a “O.S Security”, mediante el acoplamiento de la metodología DSDM (Dynamic Systems Development Method), que es una metodología ágil que permite mantener un desarrollo evolutivo, apegado a los requerimientos del proyecto.

Para la documentación, especificación de requerimientos y modelado del sistema se usará el Lenguaje de modelado unificado (UML), además se especifica la funcionalidad de la aplicación en su primera versión; con la finalidad de proyectar los beneficios del uso de la herramienta desarrollada.

## 4.1. ACOPLAMIENTO DE LA METODOLOGÍA DSDM PARA “O.S. SECURITY”

Para lograr enfoque claro y conciso, se ha elegido la Metodología DSMD; la cual permite una descripción apropiada de los diferentes módulos que “O.S. Security” está constituida.

### 4.1.1 DSDM – FASE DE VIABILIDAD

#### 4.1.1.1. DETERMINACIÓN DE REQUERIMIENTOS ESPECÍFICOS

**REQ (01)** La aplicación deberá permitir un ingreso seguro, mediante la validación de los datos registrados en el sistema operativo, con los que ingrese el usuario. Y se mantendrá la seguridad, haciendo uso de sesiones.

**REQ (02)** La aplicación web deberá acceder y manipular de una manera idónea los archivos de configuración (iptables, ip6tables, squid), de tal forma que permita administrarlos correctamente. Además que la aplicación tiene que ser capaz de conservar los permisos que posea el archivo.

**REQ (03)** La aplicación deberá proveer la posibilidad de iniciar y/o detener servicios, dependiendo del tipo de configuración que el root efectúe sobre el sistema operativo.

#### 4.1.1.2. DETERMINACIÓN DE REQUERIMIENTOS DE INTERFACES

Las interfaces del Administrador están constituidas esencialmente por las ventanas Web, cuadros de diálogo, gráficos, etc., el propósito es crear un entorno gráfico con características para configuraciones intuitivas, dando lugar a los siguientes requerimientos que debe cumplir:

- Menús Interactivos (Cajas de diálogo y respuestas del sistema)
- Gráficos

## **CARACTERÍSTICAS DE LOS USUARIOS**

“O.S Security” está diseñado en base a crear un entorno que facilite la configuración vía Web de los elementos de seguridad: iptables (IPv4, IPv6), squid; mediante una interfaz intuitiva. Dado a que la mayoría de configuraciones en entornos Linux se lo efectúa vía consola, ésta aplicación pretende de cierta manera cambiarlo hacia un entorno gráfico.

“O.S Security” puede ser usado por usuarios que tengan conocimientos básicos y/o avanzados sobre Linux, especialmente sobre el manejo y configuración de los ficheros iptables y squid.

### **4.1.1.3. DETERMINACIÓN DE INTERFACES DE SOFTWARE**

Dando una visión general citamos a continuación los requerimientos planteados:

#### **RECURSO SOFTWARE: PHP**

**OBJETIVO:** Desarrollar un entorno Web que permita la configuración de elementos de seguridad.

**OBSERVACIONES:** Los módulos que conformarán la aplicación Web deberán ser implementados siguiendo el paradigma de la Programación Orientada a Objetos (POO).

#### **RECURSO SOFTWARE: APACHE**

**OBJETIVO:** Gestionar los recursos necesarios para el correcto desempeño de la aplicación Web.

**OBSERVACIONES:** Para el correcto desempeño de la aplicación Web, es necesario realizar una configuración previa en el Sistema Operativo Linux,

asignándole los permisos necesarios para la manipulación de ficheros y la activación de servicios vía Web.

**RECURSO SOFTWARE:** LINUX CentOS 5

**OBJETIVO:** Interactuar con la aplicación Web de tal forma que permita la configuración y administración de los ficheros del sistema.

**OBSERVACIONES:** Es el encargado de integrar los módulos necesarios para el normal desempeño de la aplicación.

#### 4.1.1.4. ANÁLISIS DE RIESGOS

Aquí se trata de localizar los posibles inconvenientes que de cierta manera no permitan la normal ejecución de la aplicación Web en el desarrollo de sus módulos. Por ello, se ha optado a una clasificación en la cual se analizara los riesgos más comunes en los cuales nuestra aplicación Web pueda incurrir para de esta manera optar por el plan de contingencia que supere los mismos.

##### **NIVELES DE RIESGO**

**Físicos**, causados por el acceso no autorizado de personas a un centro de cómputo, en este caso el servidor Proxy de la E.I.S.

**Software**, causados por errores de mala configuración del sistema.

**Aspectos de compatibilidad**, causados por la mala planeación de la integración de la aplicación Web con el sistema operativo.

**Falta de políticas de seguridad**, es necesario considerar posibles ataques y las acciones a realizar frente a estos eventos. Con lo descrito anteriormente, se ha creado un listado de los posibles riesgos y la manera como afrontar los mismos de una manera adecuada, ver **Tabla III.37**.

**Tabla III.37.** Tabla de riesgos.

<b>Riesgo</b>	<b>Solución</b>
No contar con el hardware necesario	Una solución a este riesgo, es enviar una solicitud previa de requerimiento de equipos para el correcto desempeño de la aplicación.
Readecuación de la planificación.	La solución más oportuna para este riesgo es realizar un cronograma de trabajo bastante real
Fijar una solución no conveniente	Si realizamos un análisis muy bien detallado de la situación y examinamos correctamente los requerimientos este riesgo puede ser controlado en su totalidad.
Inconvenientes con la interfaz del portal por parte del usuario final	Esto se puede evitar si se presenta continuamente avances del sistema en donde el usuario pueda ir viendo como va quedando y donde los desarrolladores puedan observar si el usuario esta satisfecho con la interfaz y manejo de las ventanas Web presentadas.

#### 4.1.1.5. FACTIBILIDAD TECNOLÓGICA

En este ítem se hace referencia a la infraestructura física (hardware) en la cual la aplicación “O.S Security” se desenvolverá. Vale mencionar que la infraestructura con la que cuenta la E.I.S cumple con las exigencias para el correcto desempeño de la aplicación Web.

#### 4.1.2 DSDM – FASE DE NEGOCIO

##### 4.1.2.1. DEFINICIÓN DE LOS MÓDULOS

En esta etapa se describen los módulos que conforman la aplicación Web “O.S. Security”.

**MÓDULO 1:** “Autenticación directa con el Sistema Operativo”

**Objetivo General**

- Validar la información ingresada por el usuario con la información registrada en el sistema operativo.

**Objetivo Específico**

- Fijar los parámetros necesarios que permitan la interacción de la aplicación Web con el sistema.

**MÓDULO 2: “Configuración del Firewall IPv4”**

**Objetivo General**

- Administrar mediante un entorno Web los parámetros necesarios para el correcto funcionamiento de un firewall en el protocolo IPv4.

**Objetivos Específicos**

- Establecer los parámetros a configurar en el firewall IPv4.
- Detallar las opciones de configuración que permitan el correcto desempeño del firewall IPv4.

**MÓDULO 3: “Configuración del Firewall IPv6”**

**Objetivo General**

- Administrar mediante un entorno Web los parámetros necesarios para el correcto funcionamiento de un firewall en el protocolo IPv6.

**Objetivos Específicos**

- Establecer los parámetros a configurar en el firewall IPv6.
- Detallar las opciones de configuración que permitan el correcto desempeño del firewall IPv6.
- Diferenciar la estructura del fichero firewall IPv6 respecto a IPv4

**MÓDULO 4: “Administración y Configuración del Proxy”**

**Objetivo General**

- Especificar y definir los parámetros que permitan la adecuada administración del proxy

### **Objetivos Específicos**

- Describir los parámetros más utilizados dentro del fichero del Proxy.
- Analizar la manera de configurar estos parámetros en un entorno Web.
- Precisar los resultados que se espera obtener mediante la configuración de estos parámetros.

### **MÓDULO 5: “Información General del Equipo”**

#### **Objetivo General**

- Obtener la información específica referente al equipo donde se ejecute “O.S. Security”.

#### **Objetivo Específico**

- Detallar específicamente las características del entorno en donde se desenvuelve la aplicación Web.

### **MÓDULO 6: Escaneo de puertos**

#### **Objetivo General**

- Analizar que servicios del sistema operativo permiten la utilización de determinados puertos para sus operaciones.

#### **Objetivos Específicos**

- Observar el comportamiento del sistema operativo cuando un determinado servicio hace uso de un puerto determinado.
- Elaborar políticas de seguridad que permitan reforzar la integridad del Firewall y Proxy de la E.I.S.
- Identificar los posibles puntos de riesgo al tener un puerto abierto, sin control.

### **MÓDULO 7: Conectividad entre Host**

#### **Objetivo General**

- Interactuar mediante un entorno Web si un equipo esta o no conectado a la red.

**Objetivo Específico**

- Analizar los resultados obtenidos con los equipos existentes en la intranet.

**MÓDULO 8: Administración Tareas Programadas (CRON)**

**Objetivo General**

- Proveer un entorno gráfico en el cual permita configurar tareas para que el sistema operativo las realice automáticamente.

**Objetivos Específicos**

- Fijar los parámetros necesarios que permitan la correcta configuración de una determinada tarea
- Comprobar las acciones realizadas por las tareas que han sido configuradas anteriormente.

**MÓDULO 9: Administración Direcciones IP Prohibidas (BLACKLIST)**

**Objetivo General**

- Administrar mediante un entorno gráfico las direcciones IPv4 que contengan un contenido no apropiado.

**Objetivo Específico**

- Identificar las redes y/o direcciones IP de contenido no apropiado.
- Registrar estas direcciones prohibidas para posteriormente ser acopladas al firewall que contiene "O.S Security"



## 4.1.3 DSDM – INTERACCIÓN DEL MODELO FUNCIONAL

### 4.1.3.1 ESQUEMA DE INTERFAZ WEB

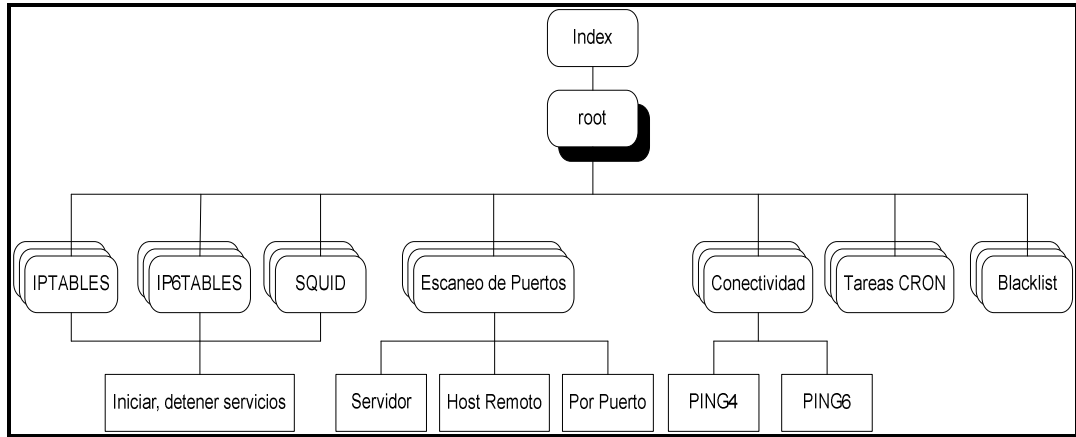


Fig. III.40. Estructura web para O.S Security

La estructura de la aplicación Web (ver Fig. III.40) será diseñada de tal manera que para hacer uso de la misma el usuario tenga que ingresar sus datos para que el sistema los valide. La página de intruso se presentará en caso que los datos ingresados sean incorrectos; caso contrario se le presentará la página principal de administración, la misma que está conformada por las siguientes opciones:

- **Administrar IPTABLES**

Presenta una página para la configuración de Firewall en IPv4 los cambios se lo realiza sobre un script bash. En la misma, se presenta una opción Iniciar Servicio, seleccionada la opción se le presenta una página informando los detalles del proceso realizado.

- **Administrar IP6TABLES**

Presenta una página para la configuración de cambios sobre un script bash que contiene la estructura del Firewall para IPv6. En la misma, se presenta una opción Iniciar Servicio, seleccionada la opción se le presenta una página informando los detalles del proceso realizado.

- **Administrar SQUID**

Se presentará una página que incluye las opciones respectivas de configuración de cambios sobre el archivo SQUID. En la misma, se presenta una opción Iniciar Servicio, seleccionada la opción se le presenta una página informando los detalles del proceso realizado.

- **Escaneo de Puertos**

Ésta opción permite verificar los puertos utilizados los distintos servicios de Linux en el equipo donde se encuentra la aplicación Web.

- **Conectividad**

A través de un entorno Web esta opción permite verificar la conectividad entre los host.

- **Administrar Tareas Cron**

Ésta opción permite configurar tareas para que el sistema operativo las realice automáticamente.

- **Administrar Blacklist**

Permite mediante un entorno gráfico las direcciones IPv4 que contengan un contenido no apropiado, las mismas que servirán para incorporarlas al firewall IPv4.

#### 4.1.4 DSDM – INTERACCIÓN DEL DISEÑO Y VERSIÓN

##### 4.1.4.1 DESARROLLO DE LA APLICACIÓN

Este ítem se lo divide en 3 fases, que se los detallará a continuación:

**Fase 1:** Establecimiento de las Clases y Diseño de Interfaces

En esta fase se estructuran las clases que serán utilizadas para satisfacer los requerimientos establecidos. Además, se establecen también los parámetros que corresponden al diseño de la interfaz (colores, opciones, gráficos, secciones, etc.).

Los responsables de efectuar las actividades de ésta fase, son:

- **Marco Malán:**

Se encargará de desarrollar los módulos: IPTABLES, IP6TABLES y SQUID

- **Leonardo Gualpa:**

Se encargará de desarrollar los módulos: ESCANEO DE PUERTOS, CONECTIVIDAD, ADMINISTRACIÓN DE TAREAS y ADMINISTRACIÓN DE BLACKLIST

**Nota:** El desarrollo de la interfaz es un trabajo conjunto de ambas personas.

**Fase 2:** Desarrollo de los Módulos establecidos

- **Marco Malán:**

Desarrolla las siguientes clases:

class\_acceso\_ficheros.php  
class\_iptables.php  
class\_ip6tables.php  
class\_squid.php  
firewall.php  
iptables.php  
ipv4.php  
firewall6.php  
ip6tables.php  
ipv6.php  
proxy.php  
mngproxy.php  
manual\_proxy.php

Todas estas clases están encaminadas a cumplir los requerimientos de los módulos asignados.

- **Leonardo Gualpa:**

Desarrolla las siguientes clases:

class\_controls.php  
class\_cron.php  
class\_blacklist.php  
class\_ports.php  
class\_portsService.php  
administracion.php

Todas estas clases están encaminadas a cumplir los requerimientos de los módulos asignados.

### **Fase 3: Integración**

Tras el desarrollo de los diferentes módulos, éstos se encuentran funcionales individualmente. En ésta etapa se dará paso a la integración que conlleve a la obtención de una sola aplicación que cumpla cabalmente con los requerimientos establecidos y que además contiene funciones adicionales logrando más funcionalidades de las que se esperaban.

Ésta fase requiere del trabajo conjunto de ambas personas, distribuidas en varias sesiones de trabajo.

#### **4.1.4.2 PROTOTIPO**

Esta fase presenta el entorno gráfico que “O.S Security” tiene:

- La primera parte es el encabezado ( header ), en la que se puede distinguir el nombre del sistema y la fecha.
- Se tiene dispuesto al lado izquierdo de la pantalla el menú principal ( main menú ), para acceder a las diferentes opciones que brinda el sistema.
- En la parte central, se muestran los diferentes contenidos que se requieran, dada la selección de una opción del menú principal.
- Finalmente se tiene el pie de página, en el que se destacan, el nombre de la institución y la ubicación de la misma.

A continuación se pueden ver dos ejemplos gráficos de OS Security, en las **Figuras III.41 y III.42**

Encabezado

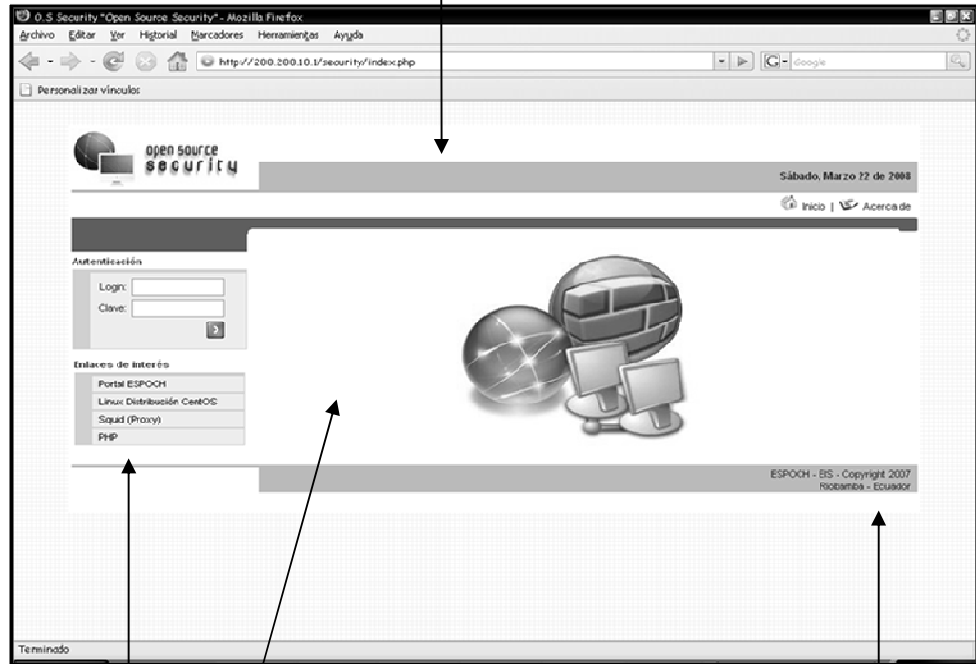


Fig. III.41. Pantalla inicial "O.S Security"

Menú

Contenidos

Píe de página

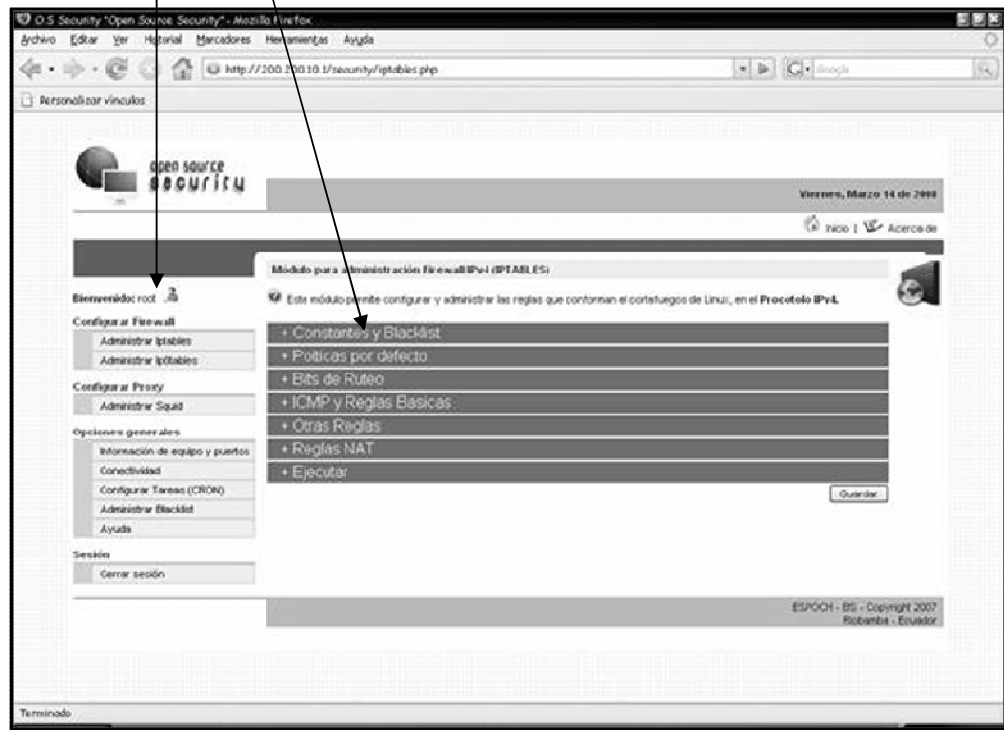


Fig. III.42. Módulo para configurar IPTABLES

## 4.1.5 DSDM - DESPLIEGUE

### 4.1.5.1 IMPLANTACIÓN

La aplicación Web “O.S Security” fue colocada en un servidor cuya plataforma es Linux CentOS 5, la misma que contiene los servicios básicos de funcionamiento como son: Servidor Web Apache y PHP; el mismo que es un requerimiento de la investigación.

Vale mencionar que todas las pruebas de la aplicación Web se las hicieron en el entorno descrito en ésta investigación; hecho por el que no se puede asegurar su eficiencia en ambientes Linux distintos.

### 4.1.5.2 PRUEBAS

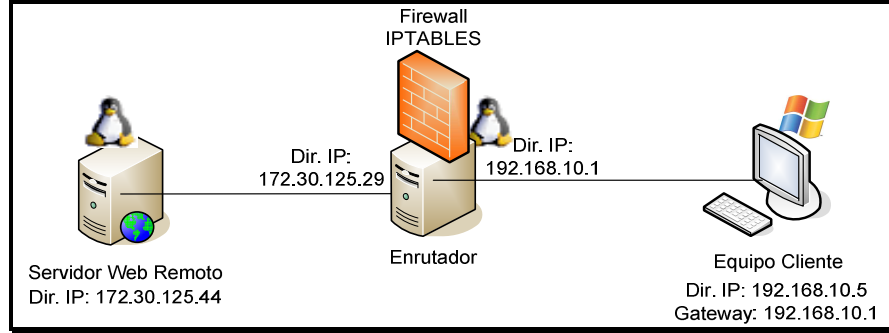
#### **PRUEBA 1: Uso del Firewall IPv4 (IPTABLES)**

Se efectuaron pruebas básicas de configuración para el módulo Firewall, que “O.S Security” incorpora con la finalidad de demostrar la administración de determinados parámetros mediante la utilización de un entorno Web.

El escenario planteado para esta prueba es el siguiente: 3 equipos, un equipo cumple con la función de un Equipo Remoto (Internet).

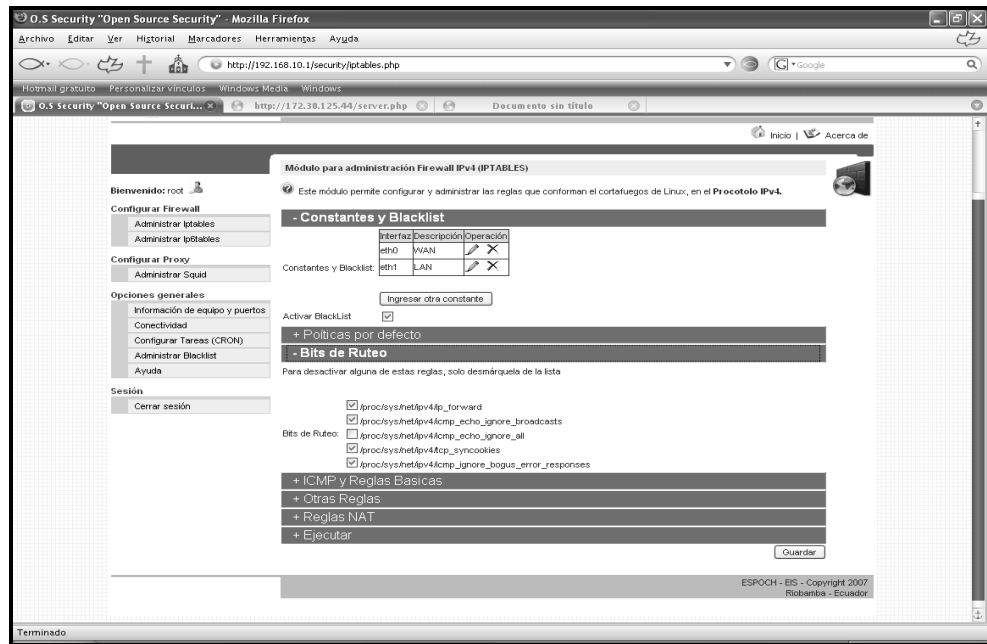
El segundo equipo cumple con la función de enrutador, es decir, permite la comunicación entre un equipo local y un equipo remoto, este equipo tiene incorporado 2 tarjetas de red, la primera tarjeta permite la conexión hacia la intranet y la segunda tarjeta permite la conexión hacia nuestro equipo remoto (Internet). Además este equipo tiene incorporado el Firewall IPTABLES para realizar el filtrado de paquetes.

Finalmente el tercer equipo es el equipo cliente perteneciente a la Intranet y cuyo Gateway es el equipo que cumple con las funciones de enrutador. El esquema gráfico se lo puede apreciar en la **Fig. III.43**



**Fig. III.43.** Escenario Pruebas con IPTABLES

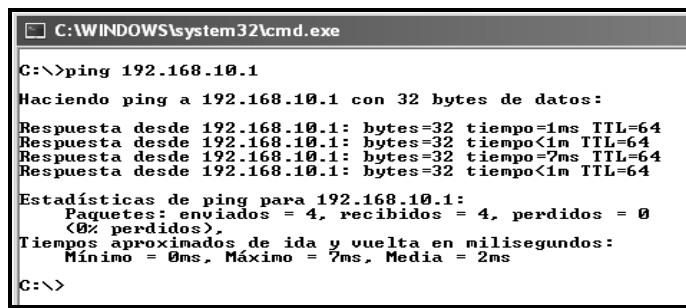
Una vez especificado el escenario, procedemos a las pruebas respectivas. Desde el equipo cliente accedemos al enrutador donde se encuentra la aplicación "O.S Security" y configuramos el módulo Administrar Iptables, que corresponde al Firewall para el Protocolo IPv4 (ver **Fig. III.44**)



**Fig. III.44.** Módulo para configurar IPTABLES

### PRUEBAS CONECTIVIDAD

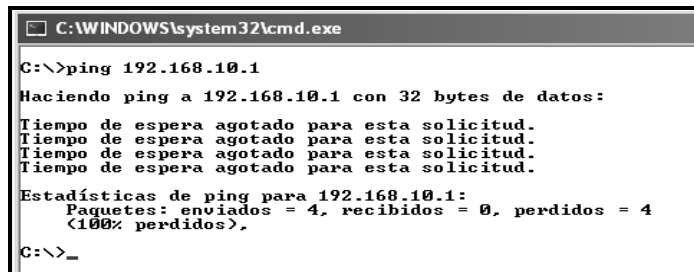
En la **Fig. III.44.** se detalla la configuración correspondiente a las interfaces activas en el enrutador, así como la sección para configurar Bits de Ruteo. En donde se puede apreciar que el tercer casillero se encuentra deshabilitado, es decir, cuando esta opción se encuentra desactivada los equipos clientes y remotos pueden realizar ping hacia el enrutador, los resultados se los aprecia en la **Fig.III.45.**



```
C:\WINDOWS\system32\cmd.exe
C:\>ping 192.168.10.1
Haciendo ping a 192.168.10.1 con 32 bytes de datos:
Respuesta desde 192.168.10.1: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.10.1: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.10.1: bytes=32 tiempo=7ms TTL=64
Respuesta desde 192.168.10.1: bytes=32 tiempo<1m TTL=64
Estadísticas de ping para 192.168.10.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 7ms, Media = 2ms
C:\>
```

**Fig. III.45.** Conectividad exitosa hacia enrutador

Al momento que se active este bit (**ver Fig. III.44**), ningún equipo ya sea cliente o remoto puede hacer ping al enrutador, esto se lo demuestra en la **Fig.III.46**



```
C:\WINDOWS\system32\cmd.exe
C:\>ping 192.168.10.1
Haciendo ping a 192.168.10.1 con 32 bytes de datos:
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Estadísticas de ping para 192.168.10.1:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
    (100% perdidos).
C:\>_
```

**Fig. III.46.** Conectividad fallida hacia enrutador

Ahora se procederá al filtrado de paquetes de entrada y salida, para lograr este objetivo se han agregado ciertas reglas que permiten el acceso del equipo cliente hacia los recursos que posee un equipo remoto (**ver Fig.III.47**).



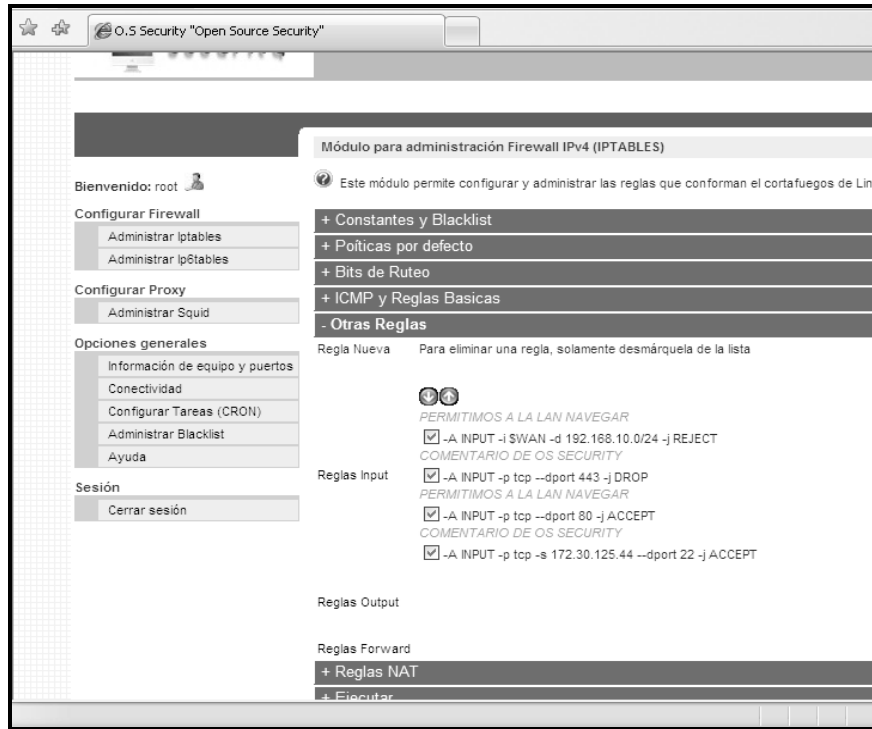


Fig. III.47. Reglas registradas en el firewall

### PRUEBAS DE ACCESO WEB

Complementariamente a las pruebas de conectividad se efectuaron pruebas mediante el uso de navegadores Web, una de ellas fue el acceso de un equipo de la intranet hacia un equipo remoto vía Web (ver Fig. III.48)



Fig. III.48. Acceso hacia recurso web de equipo remoto desde equipo cliente

Además se crearon reglas que eviten el acceso de redes externas hacia la Intranet, es decir, que un usuario externo intente acceder a un equipo perteneciente a la intranet será bloqueado, este resultado se lo puede verificar en la Fig. III.49

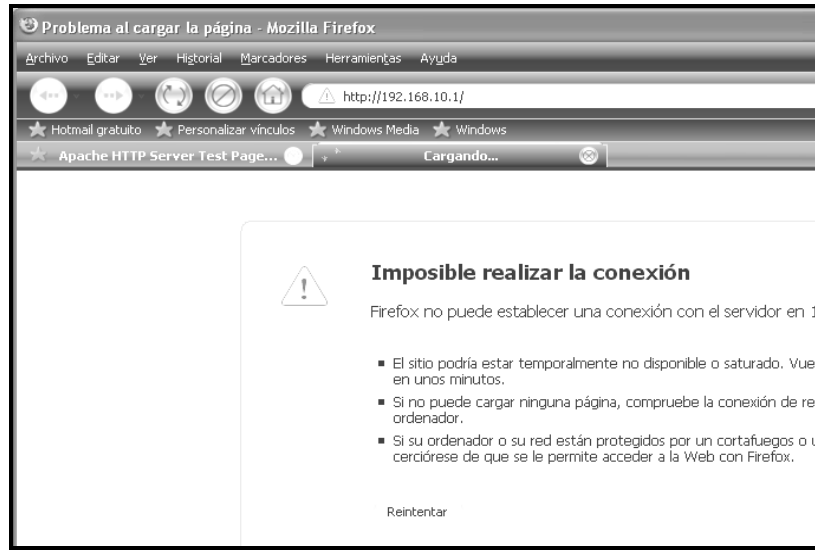


Fig. III.49. Restricción hacia intranet desde equipos remotos

### PRUEBAS USANDO SSH

En la Fig.III.47 se especificó una regla mediante la cual una dirección IP específica puede acceder remotamente al servidor usando SSH. Caso contrario no se le permite el acceso hacia este servicio.

En la Fig. III.50 se procede a la conexión remota de un equipo vía SSH

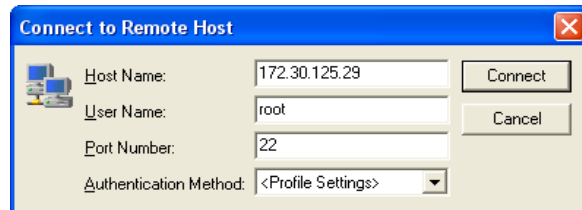


Fig. III.50. Configuración para usar SSH

Si la dirección IP del equipo cliente que intentano concuerda con la IP registrada en la regla IPTABLES de la Fig. III.47, se le prohíbe el acceso como se muestra en la Fig. III.51

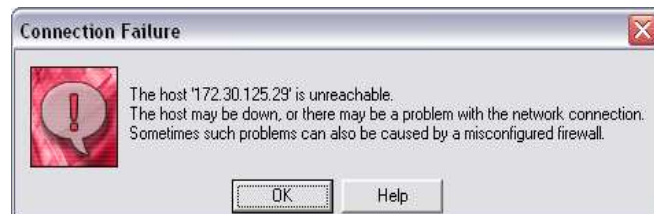
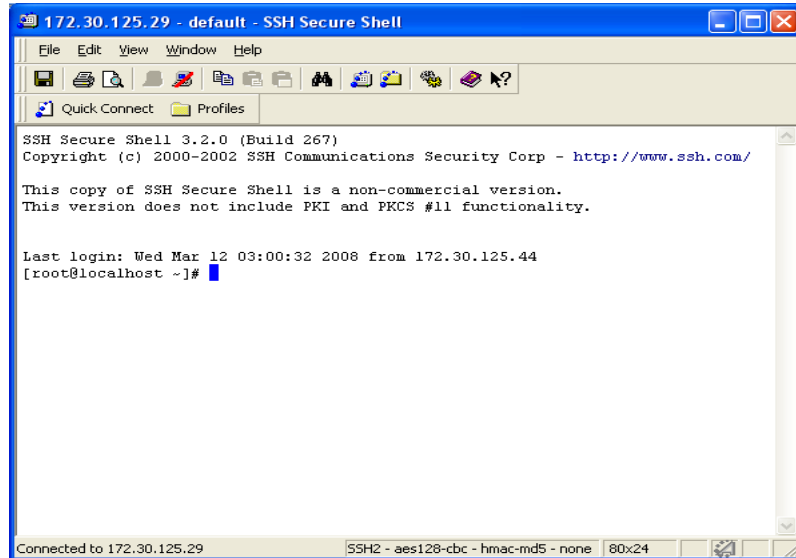


Fig. III.51. Denegación hacia SSH por regla firewall

Por otra parte si la dirección IP del equipo cliente concuerda con la IP registrada en la regla IPTABLES de la **Fig. III.47** se le autoriza el acceso vía SSH, esto se lo puede verificar en la **Fig. III.52**



**Fig. III.52.** Acceso a SSH por regla de firewall hacia una IP específica

## **PRUEBA 2: Uso del Firewall IPv6 (IP6TABLES)**

Se efectuaron pruebas básicas de configuración para el módulo Firewall para IPv6, que "O.S Security" incorpora con la finalidad de demostrar la administración de determinados parámetros mediante la utilización de un entorno Web.

El escenario planteado para esta prueba es el siguiente: 3 equipos, un equipo cumple con la función de un Equipo Remoto (Internet) con su respectiva dirección IPv6, tal como se detalla en la **Figura III.53**.

El segundo equipo cumple con la función de enrutador; es decir, permite la comunicación entre un equipo local y un equipo remoto, este equipo tiene incorporado 2 tarjetas de red, la primera tarjeta permite la conexión hacia la intranet y la segunda tarjeta permite la conexión hacia nuestro equipo remoto (Internet). El proceso de ruteo se basa en el uso del servicio radvd, el cual

permite la comunicación entre prefijos distintos IPv6. Además este equipo tiene incorporado el Firewall IP6TABLES para realizar el filtrado de paquetes.

Finalmente el tercer equipo es el equipo cliente perteneciente a la Intranet y cuyo Gateway es el equipo que cumple con las funciones de enrutador, el cual se encarga de asignar automáticamente direcciones IPv6 hacia los clientes. A continuación se presenta el esquema gráfico de este escenario:

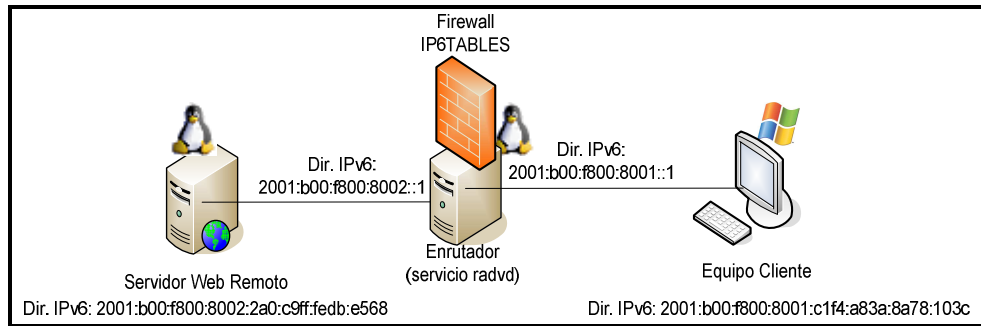


Fig. III.53. Escenario Pruebas con IP6TABLES

Una vez especificado el escenario, procedemos a las pruebas respectivas. Desde el equipo cliente accedemos al enrutador donde se encuentra la aplicación “O.S Security” y configuramos el módulo Administrar Ip6tables, que corresponde al Firewall IPv6, el mismo que puede verse en la **Figura III.54**

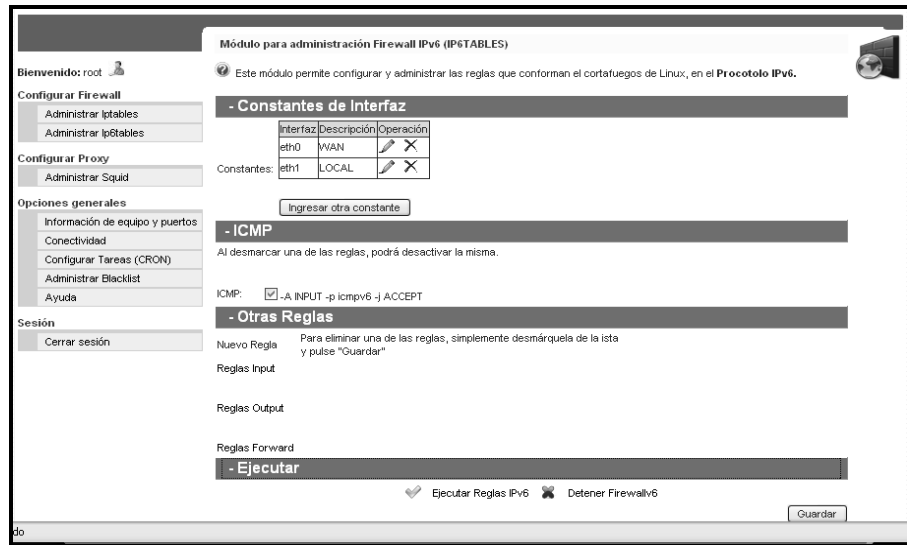
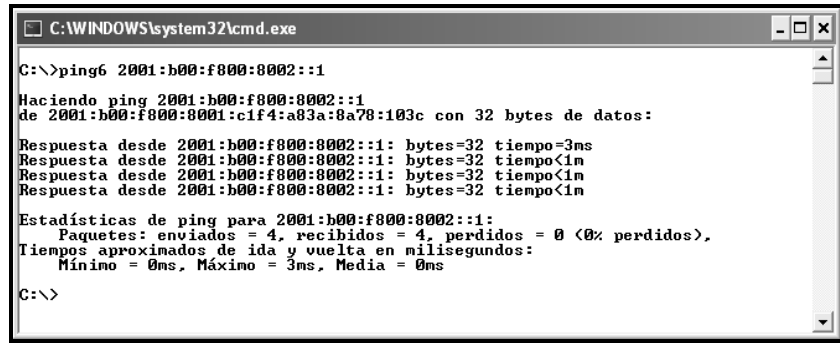


Fig. III.54. Módulo para configurar IP6TABLES

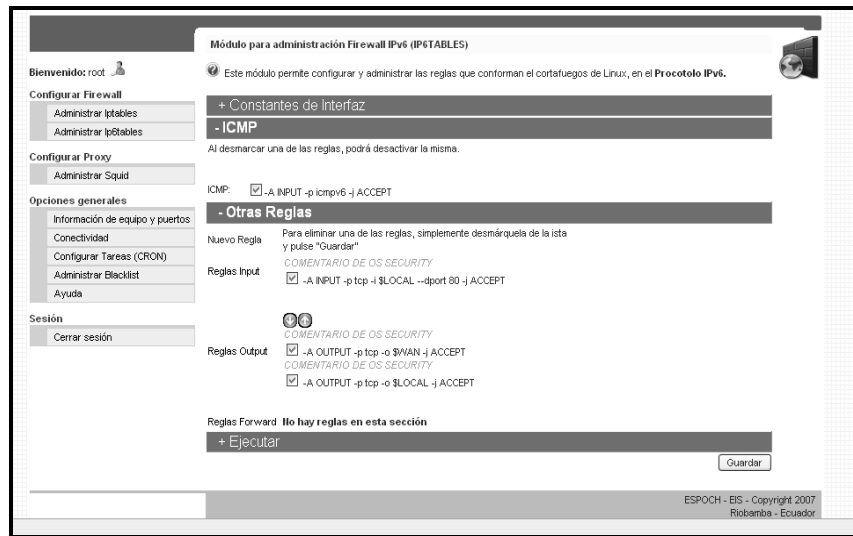
### PRUEBAS CONECTIVIDAD

En la **Fig. III.54.** se detalla la configuración correspondiente a las interfaces activas en el enrutador, así como la sección para configurar Paquetes ICMPv6. Cuando la opción `icmpv6` se encuentra desactivada los equipos clientes y remotos pueden realizar ping hacia el enrutador, tal como vemos en la **Figura III.55.**



**Fig. III.55.** Conexión exitosa hacia el enrutador

Ahora se procederá al filtrado de paquetes de entrada y salida, para lograr este objetivo se han agregado ciertas reglas (ver **Figura III.56**) que permiten el acceso del equipo cliente hacia los recursos que posee un equipo remoto.



**Fig. III.56.** Reglas registradas en el firewall IPv6

### PRUEBAS WEB

Ahora verificamos el acceso web ipv6, y vemos que existe conectividad (ver **Figura III.57**), la página que solicitamos es una con los detalles de la versión instalada de PHP.

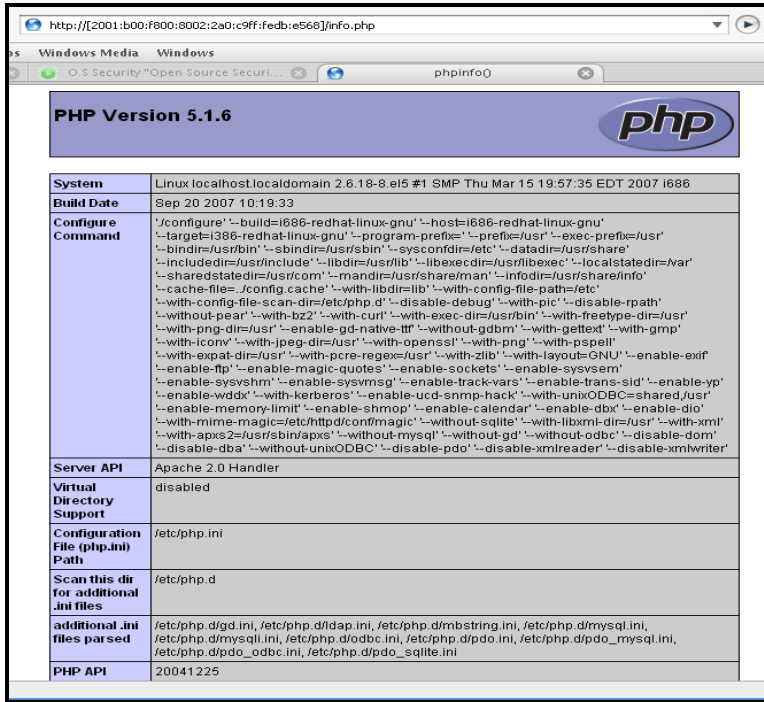


Fig. III.57. Acceso equipo remoto desde equipo cliente

### PRUEBA 3: Proxy IPv4

Se efectuaron pruebas básicas de configuración para el módulo Proxy, que "O.S Security" incorpora con la finalidad de demostrar la administración de determinados parámetros mediante la utilización de un entorno Web.

El escenario planteado para esta prueba es igual al usado para Firewall IPv4 con la diferencia que ahora se agrega el servicio squid para administrar el acceso hacia recursos Web.

A continuación, en la **Figura III.58**, se presenta el esquema gráfico de este escenario:

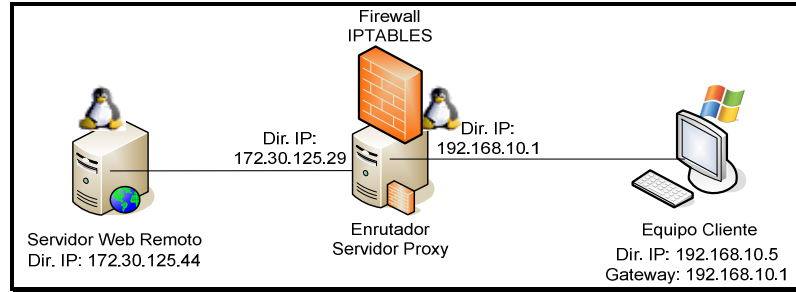


Fig. III.58. Escenario Pruebas para Proxy

Una vez especificado el escenario, procedemos a las pruebas respectivas. Desde el equipo cliente accedemos al enrutador donde se encuentra la aplicación “O.S Security” y configuramos el módulo Administrar Squid, el cual permite administrar el contenido hacia recursos Web. Para este fin, podemos ver en la **Figura III.59**, varias de las opciones que se pueden manipular con el sistema.

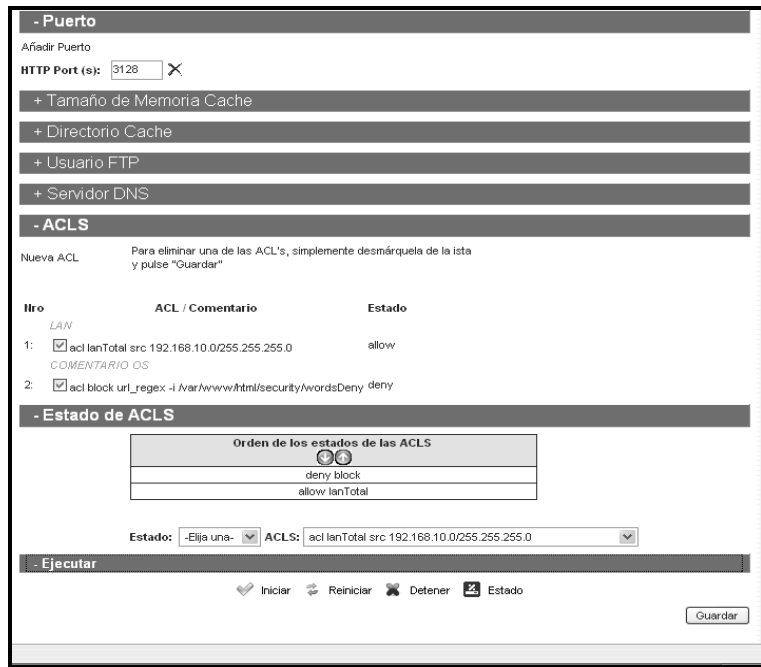


Fig. III.59. Módulo para administrar SQUID

Para tener un mejor soporte respecto al tráfico de paquetes, al módulo para configurar Firewall IPv4 se especifica ciertas reglas que permitan redireccionar todo el tráfico HTTP (ver **Figura III.60**) proveniente de la intranet apunte hacia

el servidor Proxy, el cual será el encargado de administrar el contenido existente en dicho tráfico.

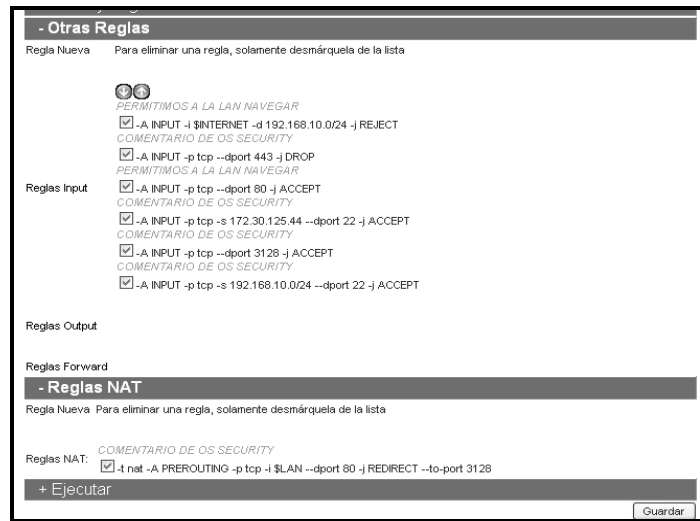


Fig. III.60. Redireccionamiento de puertos desde el firewall hacia el proxy

Para que la configuración del Squid funcione en los equipos clientes, debemos configurar en el navegador web, la dirección IP y el puerto de nuestro servidor Proxy, tal como vemos en la **Figura III.61**

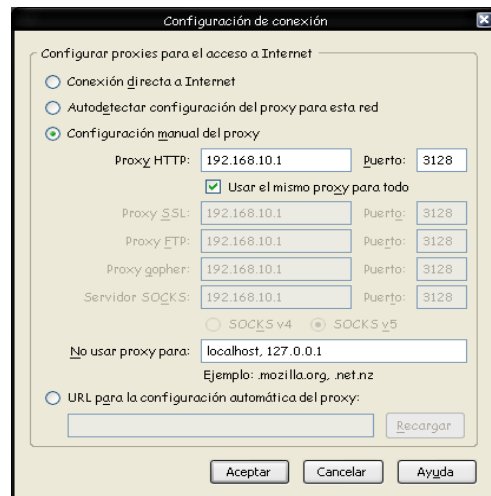


Fig. III.61. Redireccionamiento del navegador hacia el proxy

Por el momento se tiene creado una regla, la cual permite el acceso por parte de los clientes hacia sitios web remotos. Esto se verifica accediendo desde el

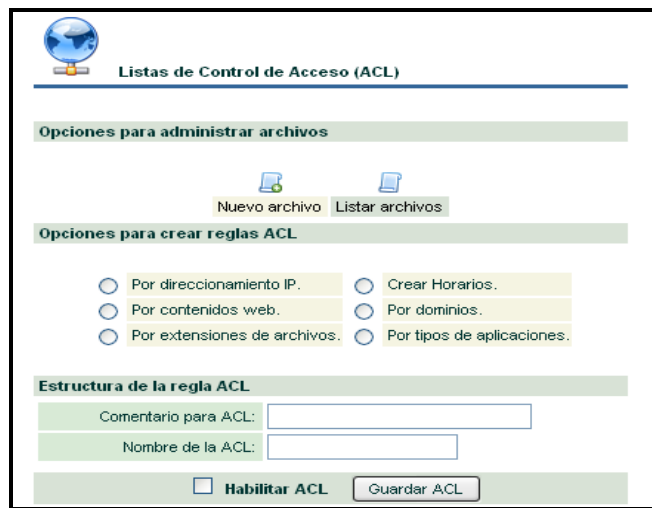


cliente en el que se configuró el proxy, (en nuestro ejemplo) solicitando la página de inicio del servidor web, como se ve en la **Figura III.62**



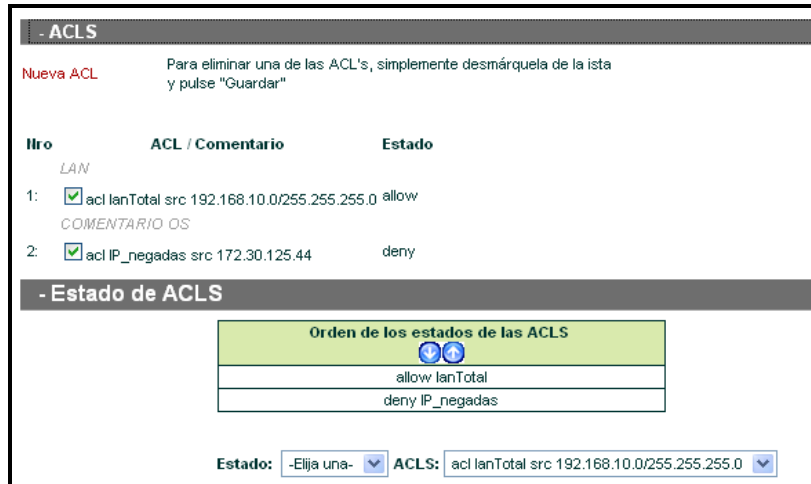
**Fig. III.62.** Acceso a equipo remoto desde equipo cliente usando proxy

Si deseamos restringir el acceso hacia una determinada dirección IP por parte de nuestros clientes, configuramos una regla que cumpla este propósito, mediante nuestra aplicación usando el formulario de creación de ACL's mostrada en la **Figura III.63**. Creamos una acl cuyo prefijo será IP\_negada en la cual especificaremos que dirección IP deseamos bloquear el acceso, para nuestro caso la dirección IP será la 172.30.125.44.




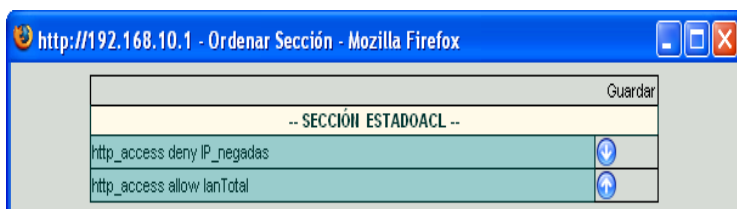
**Fig. III.63.** Creación de nueva regla ACL

Verificamos la creación de nuestra acl **IP\_negadas**, accediendo a la sección de las ACL's, mostrada en la **Figura III.64**



**Fig. III.64.** Listado de reglas ACL registradas

Debemos tomar en consideración que las reglas acl se ejecutan desde la más específica hacia la más general, por ello es importante ordenarlas, esto se consigue al dar click sobre el ícono "Ordenar" , lo que nos permitirá ver una ventana con la lista de las ACL's existentes, con los botones correspondientes para variar su ubicación, tal como vemos en la **Figura III.65**



**Fig. III.65.** Ordenamiento de reglas ACL

Guardamos los cambios y verificamos los resultados.

Con ello se podrá lograr nuestro objetivo, bloquear la navegación a un cliente determinado. Para verificar esta premisa, desde el cliente intentamos acceder al servidor web (en nuestro ejemplo) y podemos ver (**Figura III.66**) que el acceso ya ha sido restringido.

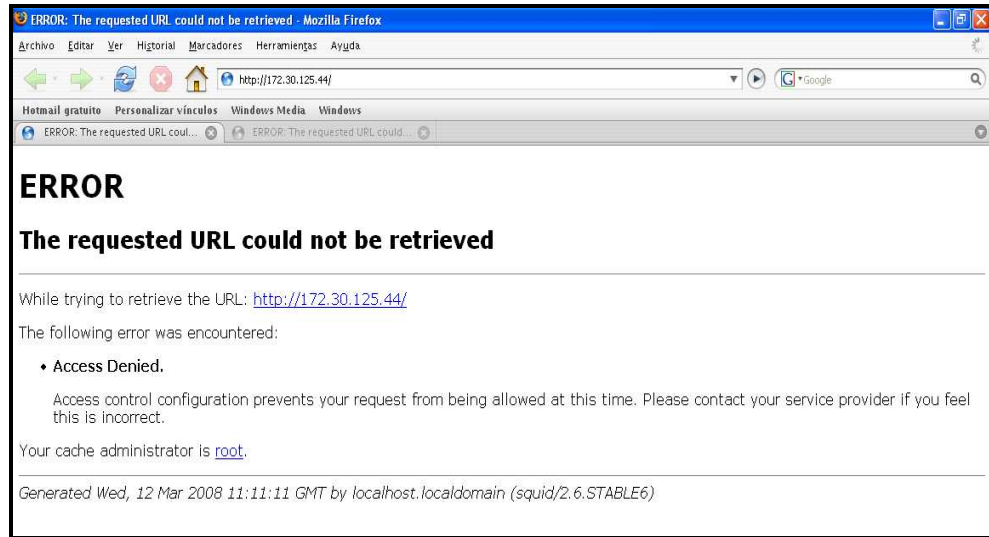


Fig. III.66. Bloqueo a dirección remota por regla ACL registrada

#### 4.1.5.3 DOCUMENTACIÓN

La información recopilada en el desarrollo de ésta investigación, se encuentra descrita en los Manuales Técnico y del Usuario, los mismos que están incluidos en los Anexos de este documento.

# CONCLUSIONES

La realización de este trabajo de tesis permite mencionar las siguientes conclusiones:

- a. Para la realización de este proyecto investigativo, se aplica los conocimientos adquiridos a lo largo de nuestra carrera; adquiriendo la experiencia necesaria en el ámbito del software libre; es decir, conocer más sobre las ventajas, funcionalidades, rendimiento, etc., que brinda el uso de las herramientas Open Source para la administración de recursos de red.
  
- b. Se verifica que la convivencia entre los Protocolos IPv4 e IPv6 es posible, mediante la utilización de los diferentes mecanismos desarrollados, como son: Pila dual, Túneles, Traducción de Direcciones; aplicando las configuraciones respectivas para cada técnica, sobre la plataforma Linux CentOS 5. Esta verificación permite establecer, en primera instancia, que las Técnicas Pila Dual y Túneles están relacionadas, es decir que la una requiere de las características de la otra para lograr un funcionamiento óptimo. Concretamente, la técnica de convivencia Túnel, requiere que el sistema operativo a utilizar mantenga un funcionamiento dual de los protocolos, IPv4 e IPv6. Con lo que respecta a la técnica traducción de direcciones, se pudo establecer que para lograr su correcto funcionamiento se tiene que instalar un software especializado que le permita realizar la laborar de traducción, dicho software ya tiene implementaciones para varias distribuciones, pero no se pudo hallar una específica para nuestra distribución Linux CentOS 5, debido a que es una distribución nueva.
  
- c. Tras efectuar un análisis de las Políticas de Red manejadas en el nodo de red de la Escuela de Ingeniería en Sistemas, con el objetivo de conocer que tipo de herramientas y configuraciones son usadas para la administración del tráfico de red. Se pudo constatar, que el servicio utilizado para administrar el acceso a sitios web, es mediante el uso de un Servidor Proxy; por otra parte, no se realiza un control para filtrado de paquetes en la red.

Con la información recopilada, se pudo estructurar los módulos necesarios que permitan la creación y desarrollo de una herramienta que incorpore elementos de seguridad como son: Filtrador de paquetes – Firewalls (IPv4 e IPv6) y Proxy, los cuales pueden ser configurados y administrados mediante una interfaz Web, brindando soporte a los Protocolos IPv4 e IPv6.

- d. Se obtuvo un equipo dedicado al filtrado de paquetes (firewall) para los protocolos IPv4 e IPv6, incluyendo un gestor de contenidos (Proxy) para acceso a sitios web; efectuándose una instalación mínima de componentes en Linux CentOS 5, tales como: Kernel, Servidor Web para hacer uso de APACHE y Servidor de Red que permite hacer uso de IPTABLES, IP6TABLES y SQUID. Permitiendo su configuración, administración y acoplamiento en un entorno Web, mediante el desarrollo de la herramienta “O.S Security”, objetivo de este proyecto de tesis.
  
- e. La utilidad de tener una herramienta para administración de tráfico de red remota, es con la finalidad de permitir al administrador de red efectuar las modificaciones necesarias, cuando se note algún comportamiento extraño en el rendimiento de la red, sin la necesidad de estar presente físicamente frente al servidor, permitiendo una configuración y/o administración remota.

## RECOMENDACIONES

- Efectuado el análisis de las Políticas de red en la Escuela de Ingeniería en Sistemas, se constató que no se cuenta con un plan de administración para tráfico de red adecuado. Es conveniente que se tomen las consideraciones necesarias, que permitan obtener el máximo rendimiento a la infraestructura existente en dicho punto de red.
- Es conveniente la actualización de los sistemas encargados de administrar el tráfico de red en la Escuela de Ingeniería en Sistemas, ya que la administración del tráfico de red no solo involucra el tener uno o varios servicios activos, sino realizar un constante seguimiento (monitoreo) de dichos servicios que permitan determinar si el rendimiento de la red es el óptimo para las necesidades de los usuarios que hacen uso de ella.

## TRABAJO A FUTURO

- El proyecto desarrollado se enfocó en administrar y configurar Firewalls (IPv4 e IPv6) complementariamente un Servidor Proxy en un entorno Web. Este proyecto puede ser tomado como referencia para la realización de nuevos proyectos investigativos, que incorporen nuevas funcionalidades a la herramienta desarrollada “O.S Security”, por ejemplo el uso de Sistemas para Detección de Intrusos (IDS), Analizador de tráfico de paquetes en tiempo real, Antivirus, Antispam, etc. Obteniendo de esta manera una herramienta más robusta y segura, que permita administrar los recursos de red eficientemente mediante la utilización de componentes Open Source – GNU/Linux.

## RESUMEN

El objetivo es el estudio, diseño y configuración de una aplicación Web de seguridad que integre los elementos: barreras de seguridad (Firewalls) y servidor de contenidos (Proxy) bajo Linux, proporcionando soporte a los protocolos IPv4 e IPv6 simultáneamente. Esta aplicación Web tiene la finalidad de permitir una administración gráfica para Firewalls y Proxy, basados en las políticas de red existentes en la Escuela de Ingeniería en Sistemas de la ESPOCH, proporcionando al gestor de red una herramienta Web administrable para seguridades.

Para la coexistencia de los protocolos IPv4 e IPv6 se efectuó un estudio comparativo entre las técnicas de convivencia: Pila Dual y Túneles, efectuándose escenarios de pruebas evaluándose los parámetros: Configuración, Compatibilidad, Escalabilidad, Seguridad e Integridad, Interoperabilidad. Usando el método analítico y matrices de valores, Pila Dual obtuvo un porcentaje del 96.6% mientras que la otra técnica obtuvo el 90%, demostrando que Pila Dual cumple nuestros requerimientos.

Open Source Security se denomina a la aplicación Web, cuya interfaz permite administrar y configurar Firewalls y Proxy gráficamente, se la realizó usando: Linux CentOS 5 como Sistema Operativo, Apache como Gestor Web y PHP como Lenguaje de Programación. Obteniéndose resultados favorables al configurar Firewalls y Proxy en un entorno gráfico frente a la configuración tradicional bajo consola.

Concluyendo el estudio de Firewalls y Proxy integrando mecanismos de convivencia entre IPv4 e IPv6, permitieron desarrollar una herramienta Web para administración de seguridades. Recomendamos a los administradores de red que usen "Open Source Security", monitoreen regularmente el tráfico de red para verificar el desempeño.

## SUMMARY

The aim is to study, design and configuration of a web application security that integrates elements: security barriers (Firewalls), and content server (Proxy) under Linux, providing support for both IPv4 and IPv6 protocols simultaneously. This web application is designed to allow a graphical management for firewalls and proxy, policy-based network in the School of Engineering Systems at the ESPOCH, providing the network manager a tool for Web manageable assurances.

For the coexistence of both IPv4 and IPv6 protocols will make a comparative study between the techniques of coexistence: Dual Stack and Tunnels, conducted test scenarios evaluated parameters: Configuration, Compatibility, Scalability, Security and Integrity, Interoperability. Using the analytical method and matrix of values, Dual Stack earned a percentage of 96.6%, whereas the other technique gained 90%, showing that Stack Dual perform our requirements.

Open Source Security referred to the Web application, which interface to manage and configure Firewalls and Proxy graphically, are performed using the Linux CentOS 5 as operating system CentOS 5, Apache as Web Server and PHP as Programming Language. Get favorable results when setting up Firewalls and Proxy in a graphical environment compared to the traditional configuration under console.

Concluding the study of firewalls and proxy integrating mechanisms of coexistence between IPv4 and IPv6, allowed to develop a tool for managing Web assurances. We encourage network administrators to use "Open Source Security", regularly monitor network traffic to check performance.



## **GLOSARIO**

### **A**

#### **Ancho de Banda**

Este término define la cantidad de datos que puede ser enviada en un periodo de tiempo determinado a través de un circuito de comunicación dado.

#### **Autenticación**

Proceso en el que se da fe de la veracidad y autenticidad de un producto, de unos datos o de un servicio.

### **C**

**CGI:** (Common Gateway Interface) es una importante tecnología de la World Wide Web que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa.

#### **Confidencialidad**

Calidad de secreto, que no puede ser revelado a terceros o personas no autorizadas.

**Contenido:** Información disponible en una página

### **D**

**DSDM:** (Dynamic Systems Development Method) Es una metodología para el desarrollo de software, apoyado por su continua implicación del usuario en un desarrollo iterativo y creciente. Como extensión del Desarrollo rápido de aplicaciones (RAD), DSDM se centra en los proyectos de sistemas de información que son caracterizados por presupuestos y agendas apretadas.

### **E**

#### **Estándar**

Norma que se utiliza como punto de partida para el desarrollo de servicios, aplicaciones, protocolos, etc.

## **F**

**Filtrado:** Técnica informática por medio de la cual se puede regular el acceso a los contenidos en Internet, bloqueando aquellos que se consideren indebidos o indeseables. Puede realizarse tanto a nivel de servidor.

**Firewall:** Es la combinación de diferentes componentes: dispositivos físicos (hardware), programas (software) y actividades de administración, que, en conjunto, permitirán aplicar una política de seguridad de una red. El objetivo es protegerla de cualquier acción hostil proveniente de un host externo a la red.

## **H**

**Host:** Es un nombre único que se le da a un dispositivo conectado a una red informática. Puede ser un ordenador, un servidor de archivos, un dispositivo de almacenamiento por red, una máquina de fax, impresora, etc. Este nombre ayuda al administrador de la red a identificar las máquinas sin tener que memorizar una dirección IP para cada una de ellas.

## **I**

**Iptables:** es el nombre de la herramienta de espacio de usuario mediante la cual el administrador puede definir políticas de filtrado del tráfico que circula por la red.

## **L**

**Linux:** es la denominación de un sistema operativo tipo Unix (también conocido como GNU/Linux) y el nombre de un núcleo. Es uno de los ejemplos más prominentes del software libre y del desarrollo del código abierto, cuyo código fuente está disponible públicamente, para que cualquier persona pueda libremente usarlo, estudiarlo, redistribuirlo, comercializarlo y, con los conocimientos informáticos adecuados, modificarlo.

## **M**

**Multicast:** Identifican un conjunto de interfaces de red

## U

**Unicast:** Sirve como identificador de una interfaz

## P

**Pila Dual:** Esta técnica implementa las pilas de ambos protocolos, IPv4 e IPv6 en cada nodo de la red. Cada nodo de pila dual en la red tendrá dos direcciones de red, una IPv4 y otra IPv6.

**Proxy:** Es un programa o dispositivo que realiza una acción en representación de otro. La finalidad más habitual es la del servidor proxy, es un sistema de software que permite la conexión de una LAN entera al exterior con sólo una dirección IP de salida

## S

### Sniffers

Programa y/o dispositivo que monitoriza la circulación de datos a través de una red. Los sniffers pueden emplearse tanto con funciones legítimas de gestión de red como para el robo de información. Los sniffers no autorizados pueden ser extremadamente peligrosos para la seguridad de una red ya que virtualmente es casi imposible detectarlos y pueden ser emplazados en cualquier lugar, convirtiéndolos en un arma indispensable de muchos piratas informáticos. Algunas herramientas sniffers conocidas son: WepCrack, Airtort o NetStumbler, entre otras.

## T

**Túnel:** Es un mecanismo en el que un paquete es encapsulado, dentro de otro tipo de paquete. Es decir podemos encapsular paquetes IPv6 dentro de paquetes IPv4.

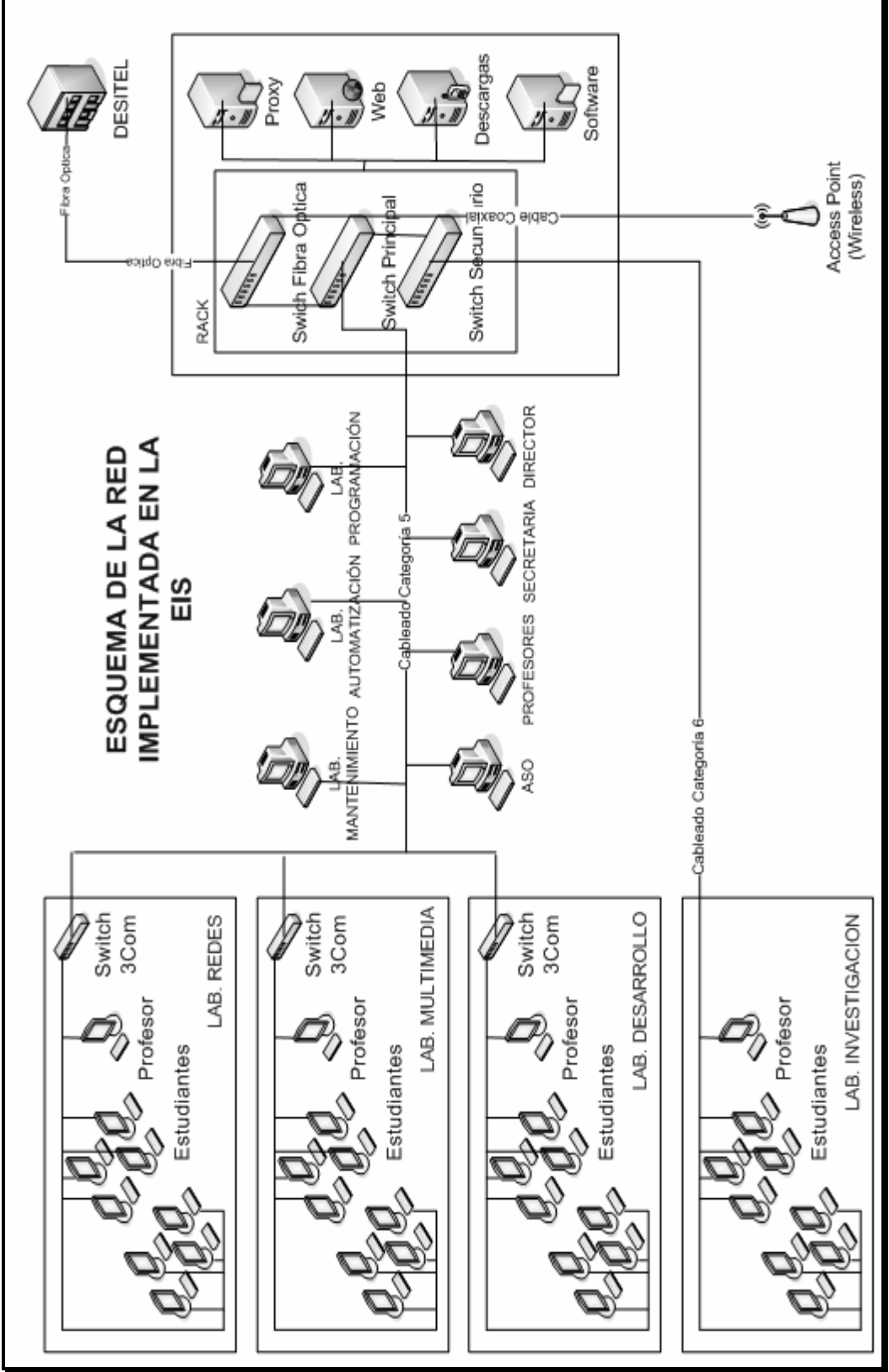
**Traducción de Direcciones:** Este mecanismo de transición permite a un nodo que solo cuenta con el stack IPv6 habilitado dentro de una red IPv6 comunicarse con otro nodo que solo tiene el stack IPv4 habilitado dentro de una red IPv4.

# **ANEXOS**

## **ANEXO A**

### **Infraestructura de Red de la Escuela Ingeniería en Sistemas**

# ESQUEMA DE LA RED IMPLEMENTADA EN LA EIS



## **ANEXO B**

### **“O.S Security” Manual Técnico**

# **Manual Técnico “Open Source Security”**

## **COPYRIGHT**

Bajo las leyes de derecho de autor, la Documentación, el Software, junto con sus elementos no pueden ser copiados, fotocopiados, reproducidos, trasladados o reducidos a cualquier medio electrónico, en forma parcial o total, sin la previa autorización escrita de Leonardo Gualpa, Marco Malán y a la ESPOCH, desarrolladores del producto.

© 2007 - 2008 “O.S. Security”

**Todos los derechos reservados.**



## 1. INTRODUCCIÓN

“O.S. Security”, se ha desarrollado con el fin de contribuir a la comunidad open source, haciendo una nueva herramienta que ayude a los usuarios del sistema operativo Linux CentOS 5, a configurar los archivos IPTABLES, IP6TABLES y SQUID.CONF, necesarios para los servicios FIREWALL, FIREWALL6 y SQUID respectivamente, a través de una interfaz web completamente amigable e intuitiva.

### 1.1. Alcance

El desarrollo de este manual está destinado para quienes deseen consultar aspectos inherentes al desarrollo de esta aplicación web.

### 1.2. Objetivos

- ↳ Describir un conjunto de actividades racionales y sistemáticas en cuanto a los procesos de desarrollo.
- ↳ Hacer uso de estándares de documentación que permitan describir cada una de las fases utilizadas para el desarrollo de un sistema.
- ↳ Considerar conceptos y términos concretos que facilitarán un mayor grado de captación y entendimiento por parte de quienes hagan uso del manual.

### 1.3. Definiciones y Acrónimos

Los términos utilizados en este manual se describen a continuación:

HW	Hardware
SW	Software
O.S. Security	Open Source Security
SO	Sistema Operativo
ESPOCH	Escuela Superior Politécnica de Chimborazo

### 1.4. Referencias

La información tomada como referencia se basa en lo siguiente:

- ↳ Estándar de Documentación: ANSI/IEEE 830-1984
- ↳ Documentación general de la Ingeniería de Software.
- ↳ Apuntes de las materias relacionadas con el desarrollo de software.

## 2. ESPECIFICACIÓN DE REQUISITOS SW

En esta primera fase, se especifican los requerimientos que debe cumplir la aplicación, para que esta pueda ser una ayuda cuando de manipular los servicios Firewall y Proxy se trata.

### 2.1. INTRODUCCIÓN

Lo que se desarrolla en esta fase, es la descripción de las diferentes funciones que tendrá la aplicación web “O.S. Security”. Para esto se ha tomado como referencia el punto de red de la Escuela de Ingeniería en Sistemas Informáticos de la ESPOCH.

#### 2.1.1. Propósito

La parte crucial de esta fase, es determinar las funcionalidades que la aplicación deberá tener, así también como de las posibles restricciones que la misma tendrá.

### 2.1.2. **Ámbito del Sistema**

“O.S. Security”, ha sido concebido como el resultado de la investigación de tesis de los señores Leonardo Gualpa y Marco Malán, con la finalidad de ser una ayuda para la configuración vía web de el firewall y Proxy de un sistema Linux CentOS 5.

## 2.2. DESCRIPCIÓN GENERAL

### 2.2.1. **Perspectiva del Software**

Esta aplicación debe permitir acceder a un equipo con el sistema operativo Linux CentOS 5, y facilitar así, la configuración rápida de los archivos Firewallv4, Firewallv6 y squid.conf.

Así también, sería útil que mediante la aplicación misma, se puedan dar ciertos pasos, comúnmente usados para determinar la correcta configuración de un servidor, como por ejemplo: el hacer ping, iniciar servicios de firewall firewall6 y Proxy, escaneo de puertos, bloqueo de listas negras (blacklist), ejecución de tareas.

### 2.2.2. **Funciones del Software**

**Acceso a los Ficheros de Configuración.**- Se debe acceder a los archivos de configuración de los servicios que se desean administrar.

**Inicio de Servicios.**- Tras realizadas las configuraciones, se debe permitir el iniciar los servicios configurados.

**Pruebas de Conectividad.**- Ya que usualmente se utiliza el comando ping para la verificación de conectividad, la aplicación web, también podrá efectuar la ejecución de tal comando.

**Escaneo de Puertos.**- Otra de las operaciones que a menudo se utiliza para la verificación de la efectividad de un firewall, “O.S. Security” deberá proveer esta opción.

**Bloqueo de Blacklist.**- En la actualidad se conocen ya algunas direcciones no apropiadas para el acceso a una red, por lo que la aplicación también podrá leer un lista (blacklist) que contenga éstas direcciones nocivas para una red.

**Ejecución de Tareas.**- Adicionalmente, la aplicación estará en la capacidad de ir almacenando tareas en el sistema, para poder ejecutarlas en un momento dado.

### 2.2.3. **Limitaciones Generales**

**Configuración de Usuario Apache.**- Debido a que las operaciones se llevarán a cabo a través de una aplicación web, el proceso en el sistema mismo se lo hará como usuario apache, este es un usuario con privilegios limitados en un sistema Linux, por lo que habrá que hallar la manera de poder ofrecerle a este usuario las capacidades necesarias para que la aplicación pueda cumplir su objetivo.

**Autenticación de Usuarios.**- Por cuestiones de seguridad, la aplicación no podrá ser acezada por otros usuarios que no sean el administrador del

sistema, por lo que el único usuario capacitado para acceder a la aplicación, será el usuario root.

**Tiempo de Sesión.**- Por ser una aplicación dirigida a manipular archivos de configuración del sistema Linux CentOS 5, esta aplicación no podría quedar abierta por mucho tiempo en un browser, ya que podría ser empleada para actos de vandalismo informático. Por lo que tras 5 minutos de inactividad, la sesión se caducará.

**Movilidad.**- En vista de cada distribución de Linux tiene, generalmente sus propias características, la investigaciones realizadas para el desarrollo de esta aplicación son comprobadas únicamente para Linux CentOS 5, no podemos asegurar que funcione apropiadamente en otra distribución.

## **2.3. REQUISITOS ESPECÍFICOS**

En este apartado, se detallaran los requisitos que deberán ser satisfechos por nuestra aplicación, que solo se regirán al correcto desempeño de la misma.

### **2.3.1. Requisitos Funcionales**

#### **REQ (01) Validación del usuario root y sesiones**

La aplicación deberá permitir un ingreso seguro, mediante la validación de los datos registrados en el sistema operativo, con los que ingrese el usuario. Y se mantendrá la seguridad, haciendo uso de sesiones.

#### **REQ (02) Acceso a los ficheros de configuración de los servicios establecidos**

La aplicación web deberá acceder y manipular de una manera idónea los archivos de configuración (iptables, ip6tables, squid), de tal forma que permita administrarlos correctamente. Además que la aplicación tiene que ser capaz de conservar los permisos que posea el archivo.

#### **REQ (03) Inicio de servicios**

La aplicación deberá proveer la posibilidad de iniciar y/o detener servicios, dependiendo del tipo de configuración que el root efectúe sobre el sistema operativo.

#### **REQ (04) scaneo de Puertos**

Otra de las operaciones que a menudo se utiliza para la verificación de la efectividad de un firewall, "O.S. Security" deberá proveer esta opción.

#### **REQ(05) Bloqueo de Blacklist**

En la actualidad se conocen ya algunas direcciones no apropiadas para el acceso a una red, por lo que la aplicación también podrá leer una lista (blacklist) que contenga éstas direcciones nocivas para una red.

#### **REQ (06) Ejecución de Tareas**

Adicionalmente, la aplicación estará en la capacidad de ir almacenando tareas en el sistema, para poder ejecutarlas en un momento dado.

### 2.3.2. Requisitos de Interfaz Externa

#### REQ (07) Interfaz de usuario

Debido a que la aplicación debe ser manejable fácilmente, esta deberá tener una interfaz que sea completamente amigable, se manejarán ventanas, los datos podrán se ingresados con el teclado, elegir opciones con el Mouse, tal que el usuario tenga una visión comprensible.

### 2.3.3. Requisitos de Rendimiento

#### REQ (08) Tiempo de respuesta

Ya que la aplicación debe ser una ayuda de administración, las respuestas que la aplicación devuelva, deben ser en el menor tiempo posible.

## 3. ANÁLISIS

En esta fase se desea representar de mejor manera la interacción del usuario con la aplicación, para conseguir esto, se hacen uso de casos de uso y diagramas UML, que son completamente comprensibles.

### 3.1. Casos de Uso

#### Caso de Uso 1: Validación de datos con el sistema

<b>Caso de Uso:</b>	Validación de datos con el sistema
<b>Actores:</b>	Administrador, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	"O.S Security" permite un ingreso seguro, mediante la validación de los datos registrados en el sistema operativo, con los que ingrese el usuario.
<b>Referencias:</b>	REQ (01)

#### Curso típico de eventos

Actor	Sistema
1. El administrador ingresa sus datos (login, clave)	2. Procede a verificar los datos ingresados. 2.1. Si los datos son correctos, se le redirecciona a la página principal de administración para el root 2.2. Si los datos son incorrectos, se le redirecciona a una página de usuario no autorizado
	3. Despliega el menú de opciones disponibles para el administrador.

**Nota:** Como "O.S Security" es una aplicación que se ejecutará en entornos Linux, el usuario que podrá hacer uso de ella será el root, debido a que este usuario es el que tiene todos los privilegios para la manipulación de ficheros, inicio/detención de servicios, etc.

**Caso de Uso 2: Acceso y manipulación de archivos**

<b>Caso de Uso:</b>	Acceso y manipulación de archivos
<b>Actores:</b>	Administrador, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	Acceder y manipular de una manera idónea los archivos de configuración del sistema (iptables, ip6tables, squid), de tal forma que permita a la aplicación "O.S Security" configurarlos y administrarlos correcta, mediante un entorno Web.
<b>Referencias:</b>	REQ (02)

**Curso típico de eventos**

<b>Actor</b>	<b>Sistema</b>
1. El administrador selecciona la opción a configurar del menú desplegado por el sistema	2. Dependiendo de la opción seleccionada por el administrador, el sistema procede a interactuar con los ficheros del sistema operativo, con el objetivo de realizar su respectiva configuración.
	3. Dicha configuración se la efectúa a través de una interfaz Web, la cual contendrá los parámetros de configuración.
	4. Los parámetros de configuración variarán dependiendo de la opción que el administrador desee configurar, además debe verificar los permisos que el fichero posea.
	5. Interacción constante con el administrador y muestra al administrador los resultados obtenidos.

**Caso de Uso 3: Manipulación de Iptables**

<b>Caso de Uso:</b>	Manipulación de Iptables
<b>Actores:</b>	Administrador, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	"O.S Security" le permite al administrador del sistema, poder acceder a las reglas de iptables y le ofrece opciones para la administración del mismo.
<b>Referencias:</b>	REQ (02)

### Curso típico de eventos

Actor	Sistema
1. El administrador ingresa sus datos (login, clave)	2. Procede a verificar los datos ingresados. 2.1. Si los datos son correctos, se le redirecciona a la página principal de administración para el root 2.2. Si los datos son incorrectos, se le redirecciona a una página de usuario no autorizado
	3. Despliega el menú de opciones disponibles para el administrador.
4. Elige administrar iptables	5. Le ofrece las opciones automática y manual.
6. Elige la opción automática	7. Se muestran las diferentes secciones del archivo iptables.
8. Realiza los cambios deseados y presiona Guardar	9. Guarda los cambios efectuados en el archivo 9.1. Muestra las secciones del archivo.
10. Presiona iniciar servicio.	11. Ejecuta el servicio iptables. 11.1. Muestra un mensaje de confirmación.
12. Presiona cerrar sesión.	13. Muestra una pantalla de confirmación.
14. Elige la opción aceptar y se va.	

### Caso de Uso 4: Manipulación de Ip6tables

<b>Caso de Uso:</b>	Manipulación de Ip6tables
<b>Actores:</b>	Administrador, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	"O.S Security" le permite al administrador del sistema, poder acceder a las reglas de Ip6tables y le ofrece opciones para la administración del mismo.
<b>Referencias:</b>	REQ (02)

### Curso típico de eventos

Actor	Sistema
1. El administrador ingresa sus datos (login, clave)	2. Procede a verificar los datos ingresados. 2.1. Si los datos son correctos, se le redirecciona a la página principal de

	administración para el root 2.2. Si los datos son incorrectos, se le redirecciona a una página de usuario no autorizado
	3. Despliega el menú de opciones disponibles para el administrador.
4. Elige administrar ip6tables	5. Le ofrece las opciones automática y manual.
6. Elige la opción automática	7. Se muestran las diferentes secciones del archivo ip6tables.
8. Realiza los cambios deseados y presiona Guardar	9. Guarda los cambios efectuados en el archivo 9.1. Muestra las secciones del archivo.
10. Presiona iniciar servicio.	11. Ejecuta el servicio ip6tables. 11.1. Muestra un mensaje de confirmación.
12. Presiona cerrar sesión.	13. Muestra una pantalla de confirmación.
14. Elige la opción aceptar y se va.	

#### Caso de Uso 5: Manipulación de Squid

<b>Caso de Uso:</b>	Manipulación de Squid
<b>Actores:</b>	Administrador, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	"O.S Security" le permite al administrador del sistema, poder acceder a las reglas de Squid y le ofrece opciones para la administración del mismo.
<b>Referencias:</b>	REQ (02)

#### Curso típico de eventos

Actor	Sistema
1. El administrador ingresa sus datos (login, clave)	2. Procede a verificar los datos ingresados. 2.1. Si los datos son correctos, se le redirecciona a la página principal de administración para el root 2.2. Si los datos son incorrectos, se le redirecciona a una página de usuario no autorizado
	3. Despliega el menú de opciones disponibles para el administrador.

4. Elige administrar squid	5. Le ofrece las opciones automática y manual.
6. Elige la opción automática	7. Se muestran las diferentes secciones del archivo squid.
8. Realiza los cambios deseados y presiona Guardar	9. Guarda los cambios efectuados en el archivo 9.1. Muestra las secciones del archivo.
10. Presiona iniciar servicio.	11. Ejecuta el servicio squid. 11.1. Muestra un mensaje de confirmación.
12. Presiona cerrar sesión.	13. Muestra una pantalla de confirmación.
14. Elige la opción aceptar y se va.	

#### Caso de Uso 6: Inicio y detención de servicios

<b>Caso de Uso:</b>	Inicio y detención de servicios
<b>Actores:</b>	Root, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	“O.S Security” provee la posibilidad de iniciar y/o detener servicios, dependiendo del tipo de configuración que el root efectúe sobre el sistema.
<b>Referencias:</b>	REQ (03)

#### Curso típico de eventos

Actor	Sistema
1. Si el root una vez que ha configurado un archivo del sistema, puede establecer la manera en que el servicio opere.	2. Si la configuración de un fichero depende o no de la inicialización de servicios, el sistema será capaz de realizar estas tareas de forma automática.
	3. Visualizará los resultados obtenidos al root para que compruebe su correcto desempeño.

#### Caso de Uso 7: Seguridad e Integridad

<b>Caso de Uso:</b>	Seguridad e Integridad
<b>Actores:</b>	Root, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	“O.S Security” como mecanismo de protección, emplea el uso de sesiones. Además para el manejo de archivos, guardará los cambios efectuados en un determinado fichero conservando los permisos correspondientes.



<b>Referencias:</b>	REQ (01)
---------------------	----------

#### Curso típico de eventos

Actor	Sistema
1. Cada vez que el root haga uso de "O.S Security", por ejemplo al ingresar su login y clave	2. Validará los datos ingresados y generará una sesión exclusiva para él.
	3. En caso de entrar en un estado de inactividad, la aplicación procederá a eliminar esa sesión transcurridos 5 minutos de inactividad.
	4. Provee integridad de los datos, al momento de ser manipulados vía Web, solicitando al root la acción a realizar.
5. El root configurará los parámetros necesarios de un determinado fichero	6. Procesará la información ingresada por el root y realizará las modificaciones correspondientes al fichero.

#### Caso de uso 8: Datos del Equipo

<b>Caso de Uso:</b>	Datos del Equipo
<b>Actores:</b>	Root, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	Obtener la información específica referente al equipo donde se ejecute "O.S. Security".
<b>Referencias:</b>	Ninguna

#### Curso típico de eventos

Actor	Sistema
1. El root selecciona la opción <b>Información de equipo y puertos</b> del menú principal	2. Despliega una pantalla con las opciones: <b>Datos del Equipo</b> <b>Escanear Puertos</b>
	3. Solicita al root elegir la opción a efectuar
4. El root elige la opción: <b>Datos del Equipo.</b>	5. Despliega un formulario que contiene información específica del equipo en donde se está ejecutando "O.S Security"

#### Caso de uso 9: Escaneo de Puertos

<b>Caso de Uso:</b>	Escaneo y/o Sondeo de Puertos
<b>Actores:</b>	Root, Sistema

<b>Tipo:</b>	Primario
<b>Propósito:</b>	Analizar que servicios del sistema operativo permiten la utilización de determinados puertos para sus operaciones.
<b>Referencias:</b>	Ninguna

#### Curso típico de eventos

Actor	Sistema
1. El root selecciona la opción <b>Información de equipo y puertos</b> del menú principal	2. Despliega una pantalla con las opciones: <b>Datos del Equipo</b> <b>Escanear Puertos</b>
	3. Solicita al root elegir la opción a efectuar
4. El root elige la opción: <b>Escanear Puertos.</b>	5. Muestra una pantalla con los tipos de escaneo que se puede realizar. <b>Servidor</b> <b>Host Remoto</b> <b>Por Puerto</b>
	5.1. La opción <b>Servidor</b> realiza un escaneo de los puertos habilitados en los protocolos TCP y UDP
	5.2. La opción <b>Host Remoto</b> permite realizar el escaneo especificando la dirección IP del host remoto a ser escaneado además del rango de puertos a escanear. En el protocolo TCP.
	5.3. La opción <b>Por Puerto</b> es similar a la opción Host Remoto excepto que para realizar un escaneo no se especifica un rango de puertos. En el protocolo TCP.
	6. Solicita al root elegir el tipo de escaneo a realizar.
7. El root elige la opción que el necesite	8. Muestra los resultados obtenidos del escaneo de puertos.

#### Caso de uso 10: Conectividad entre Host

<b>Caso de Uso:</b>	Conectividad entre Host
<b>Actores:</b>	Root, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	Interactuar mediante un entorno Web si un equipo esta o no conectado a la red.
<b>Referencias:</b>	Ninguna

### Curso típico de eventos

Actor	Sistema
1. El root selecciona la opción <b>Conectividad</b> del menú principal	2. Despliega una pantalla con las opciones: <b>Exec Ping</b> <b>Exec Ping6</b>
	2.1. La opción <b>Exec Ping</b> permite realizar ping en el protocolo IPv4
	2.2. La opción <b>Exec Ping6</b> permite realizar ping6 en el protocolo IPv6
	3. Solicita al root elegir la opción a efectuar
4. El root elige la opción que el crea conveniente	5. Carga un formulario en el cual solicita el ingreso de una dirección, ya sea IPv4 o IPv6
6. Ingresa la dirección IP a la cual desee verificar si existe conectividad	7. Procesa la dirección IP ingresada y muestra los resultados obtenidos.

### Caso de uso 11: Agregar Tarea

<b>Caso de Uso:</b>	Agregar Tarea
<b>Actores:</b>	Root, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	Fijar los parámetros necesarios que permitan la correcta configuración de una determinada tarea.
<b>Referencias:</b>	Ninguna

### Curso típico de eventos

Actor	Sistema
1. El root selecciona la opción <b>Configurar Tareas (CRON)</b> del menú principal	2. Despliega una pantalla con las opciones: <b>Agregar, Listar y Ejecutar Tareas</b>
	3. Solicita al root la acción a realizar.
4. Elige la opción <b>Agregar Tarea</b>	5. Carga un formulario con los parámetros de configuración para agregar una nueva tarea al sistema.
	5.1. Los parámetros que carga son: minutos, horas, Día del mes, mes, día de la semana y la tarea a realizar.
	6. Solicita al root ingresar los parámetros.
7. Ingresa los datos solicitados	8. Procesa los datos ingresados y los registra

	en el archivo /etc/crontab para que el sistema los ejecute automáticamente.
--	---

#### Caso de uso 12: Listar Tareas

<b>Caso de Uso:</b>	Listar Tareas.
<b>Actores:</b>	Root, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	Listar el conjunto de tareas programadas para que el sistema las realice automáticamente.
<b>Referencias:</b>	Ninguna

#### Curso típico de eventos

Actor	Sistema
1. El root selecciona la opción <b>Configurar Tareas (CRON)</b> del menú principal	2. Despliega una pantalla con las opciones: <b>Agregar, Listar y Ejecutar Tareas</b>
	3. Solicita al root la acción a realizar.
4. Elige la opción <b>Listar Tareas</b>	5. Carga un formulario con las tareas registradas en el archivo /etc/crontab.
	5.1. Permite la posibilidad de eliminar una tarea específica.

#### Caso de uso 13: Ejecutar Tareas

<b>Caso de Uso:</b>	Ejecutar Tareas
<b>Actores:</b>	Root, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	Permite ejecutar el demonio encargado de la ejecución de tareas en el sistema.
<b>Referencias:</b>	Ninguna

#### Curso típico de eventos

Actor	Sistema
1. El root selecciona la opción <b>Configurar Tareas (CRON)</b> del menú principal	2. Despliega una pantalla con las opciones: <b>Agregar, Listar y Ejecutar Tareas</b>
	3. Solicita al root la acción a realizar.
4. Elige la opción <b>Ejecutar Tareas</b>	5. Se encarga de reiniciar el demonio crond para que actualice el registro de tareas.
	6. Muestra un mensaje de la acción realizada por el sistema.

**Caso de uso 14: Agregar Blacklist**

<b>Caso de Uso:</b>	Agregar Blacklist
<b>Actores:</b>	Root, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	Registrar direcciones prohibidas para posteriormente ser acopladas al firewall que contiene "O.S Security"
<b>Referencias:</b>	Ninguna

**Curso típico de eventos**

Actor	Sistema
1. El root selecciona la opción <b>Administrar Blacklist</b> del menú principal	2. Despliega una pantalla con las opciones: <b>Agregar y Listar Blacklist</b>
	3. Solicita al root la acción a realizar.
4. Elige la opción <b>Agregar Blacklist</b>	5. Carga un formulario en el cual se ingresará la dirección IPv4 considerada por el root blacklist
	6. Solicita al root ingresar la dirección IP
7. El root ingresa la dirección IP requerida.	8. Procede a validar la dirección IP y verifica si no está registrada anteriormente.
	8.1. Una vez procesada la dirección IP se la registra en un archivo que será usado por el firewall que incorpora "O.S Security"

**Caso de uso 15: Listar Blacklist**

<b>Caso de Uso:</b>	Listar Blacklist.
<b>Actores:</b>	Root, Sistema
<b>Tipo:</b>	Primario
<b>Propósito:</b>	Listar el conjunto de direcciones IPs no adecuadas (Blacklist) para posteriormente ser incorporadas al firewall de "O.S Security"
<b>Referencias:</b>	Ninguna

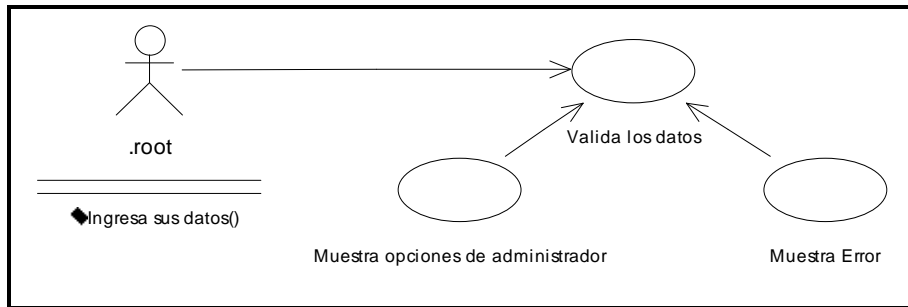
**Curso típico de eventos**

Actor	Sistema
1. El root selecciona la opción <b>Administrar Blacklist</b> del menú principal	2. Despliega una pantalla con las opciones: <b>Agregar y Listar Blacklist</b>
	3. Solicita al root la acción a realizar.
4. Elige la opción <b>Listar</b>	5. Carga un formulario

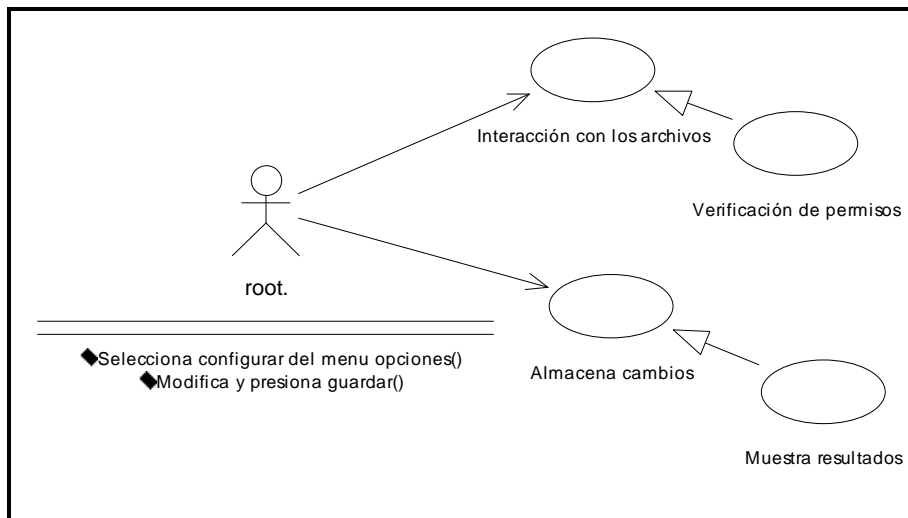
<b>Blacklist</b>	con las direcciones IPs registradas, cuyo contenido no es el apropiado.
	5.1. Permite la posibilidad de eliminar una dirección IP específica.

### 3.2. Diagramas de Casos de Uso

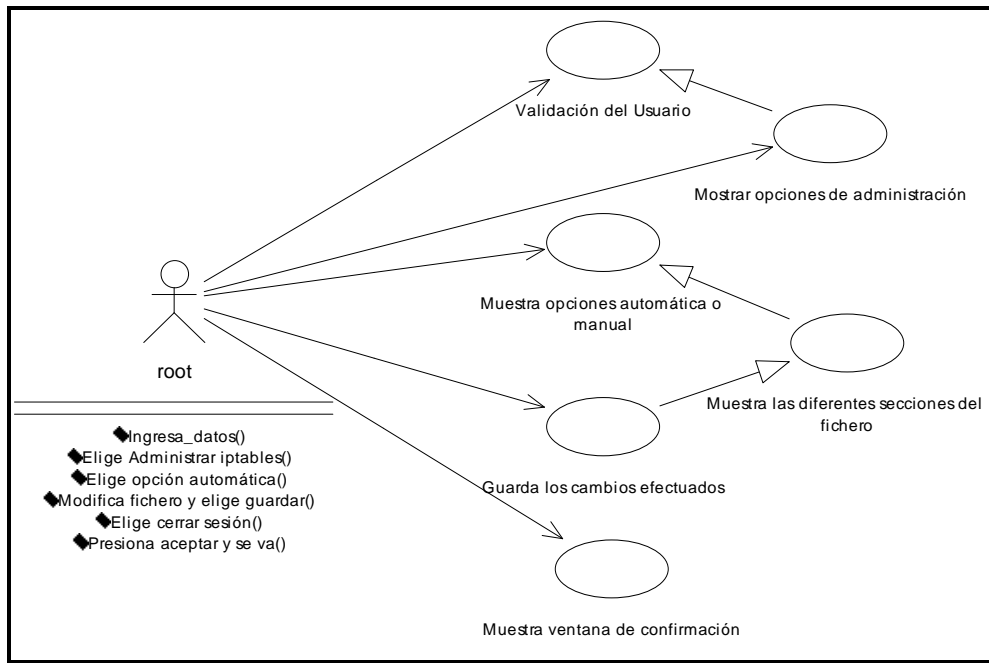
#### Caso de Uso 1: Validación de datos con el sistema



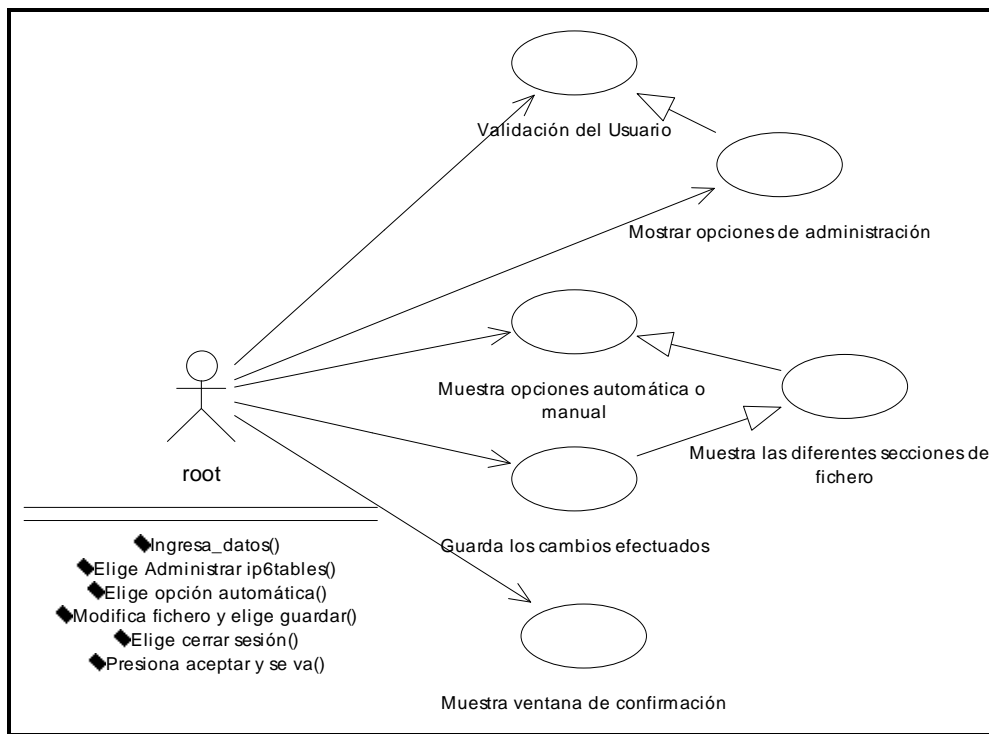
#### Caso de Uso 2: Acceso y manipulación de archivos



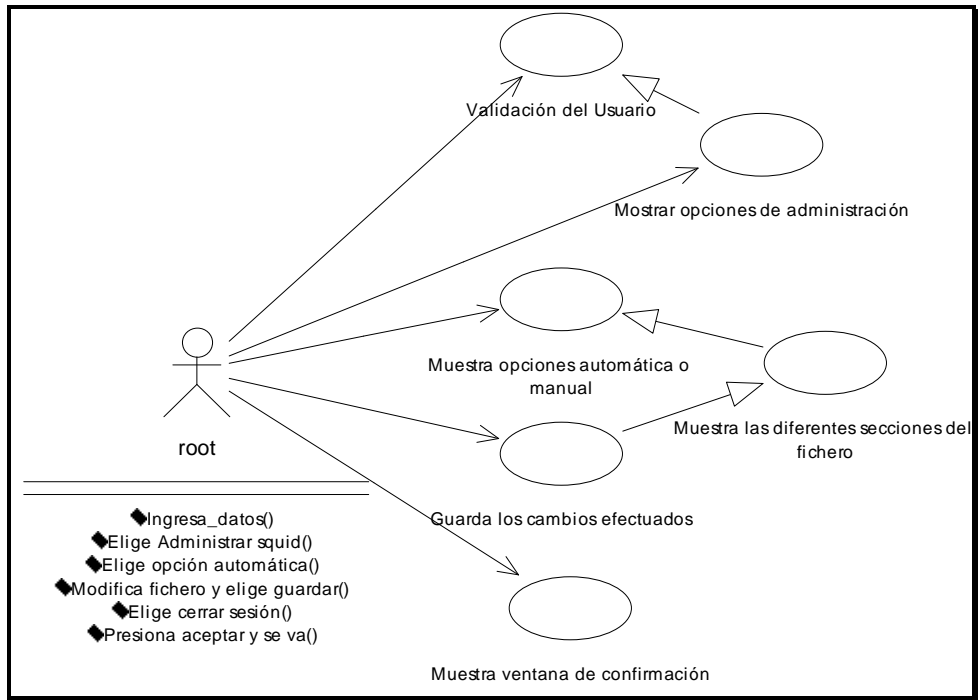
### Caso de Uso 3: Manipulación de Iptables



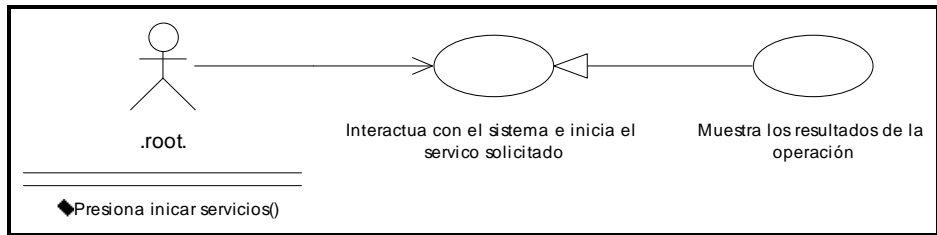
### Caso de Uso 4: Manipulación de Ip6tables



### Caso de Uso 5: Manipulación de Squid

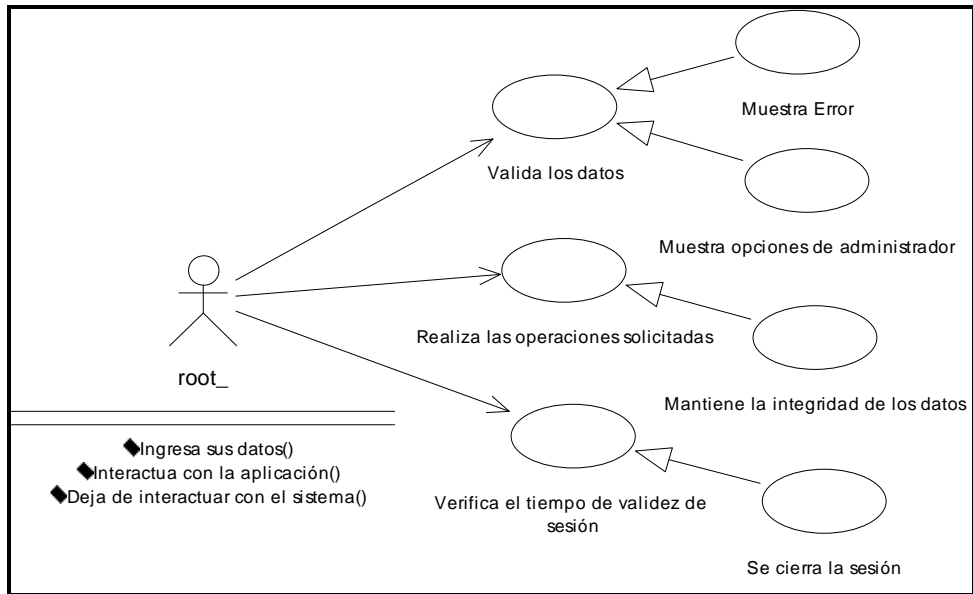


### Caso de Uso 6: Inicio y detención de servicios

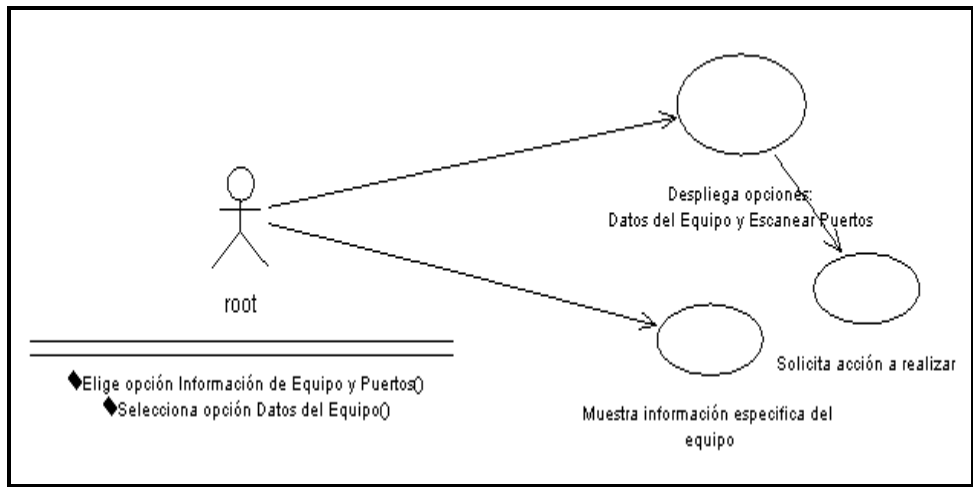




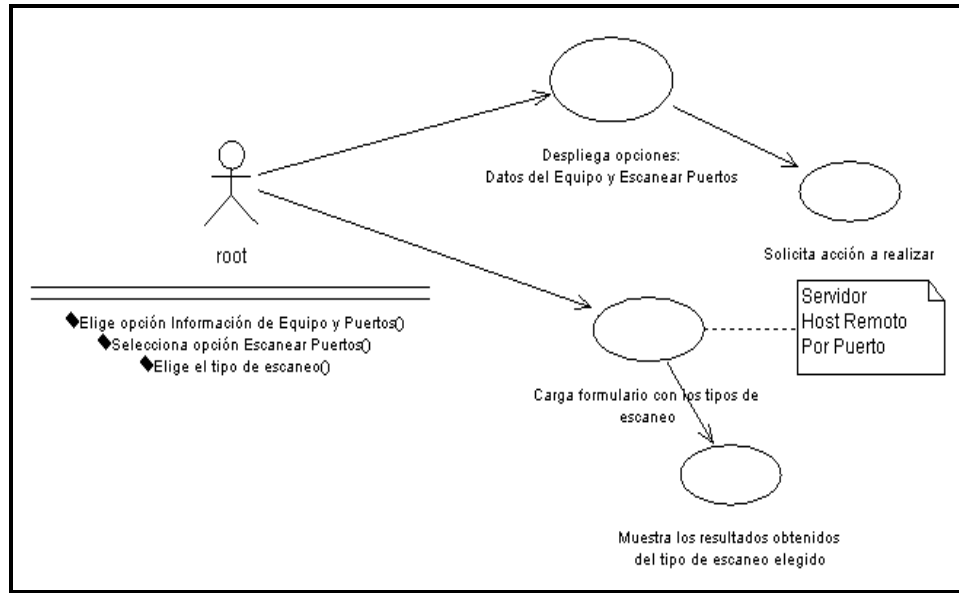
### Caso de Uso 7: Seguridad e Integridad



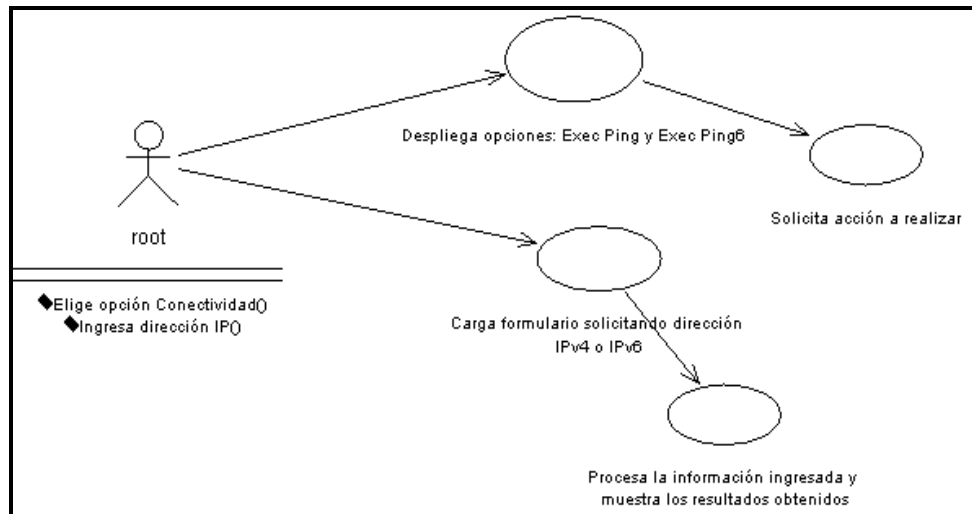
### Caso de uso 8: Datos del Equipo



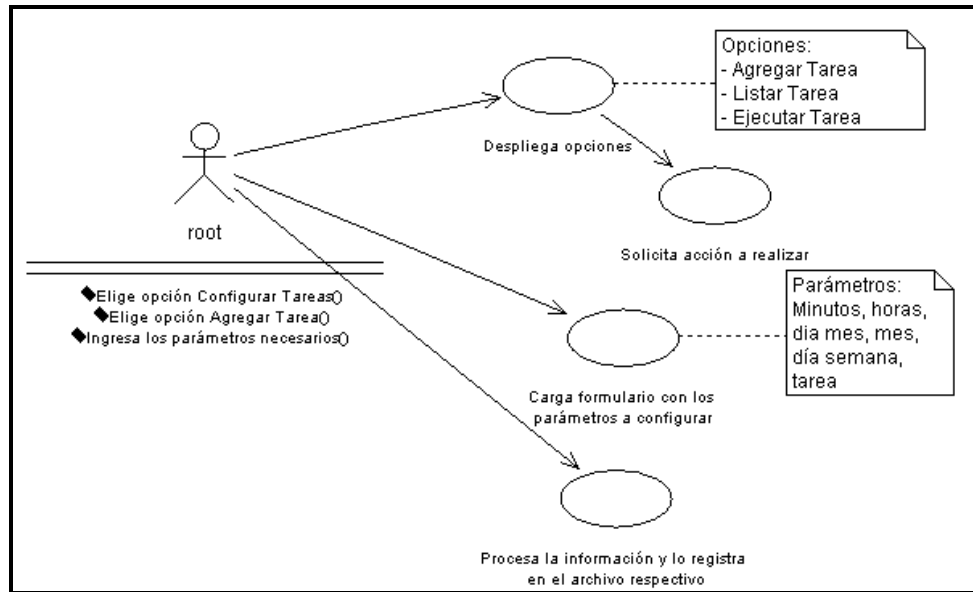
### Caso de uso 9: Escaneo de Puertos



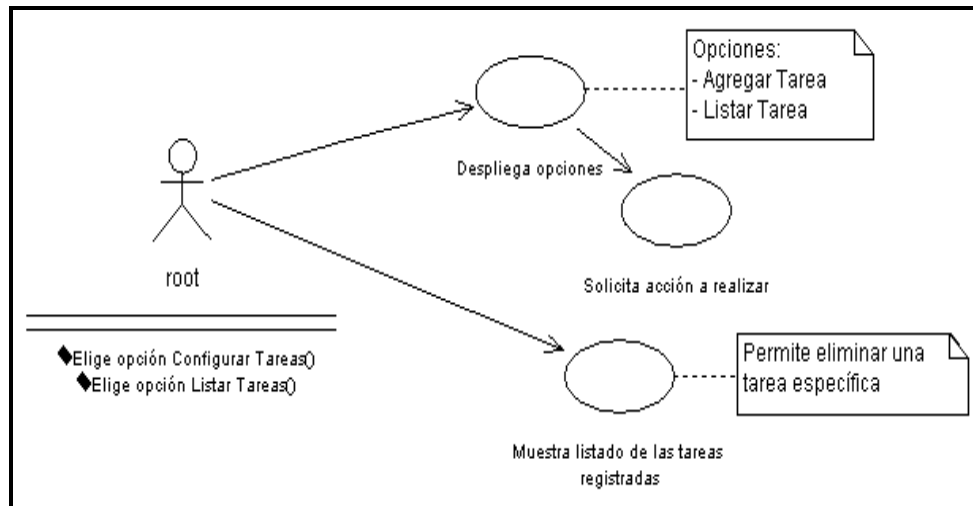
### Caso de uso 10: Conectividad entre Host



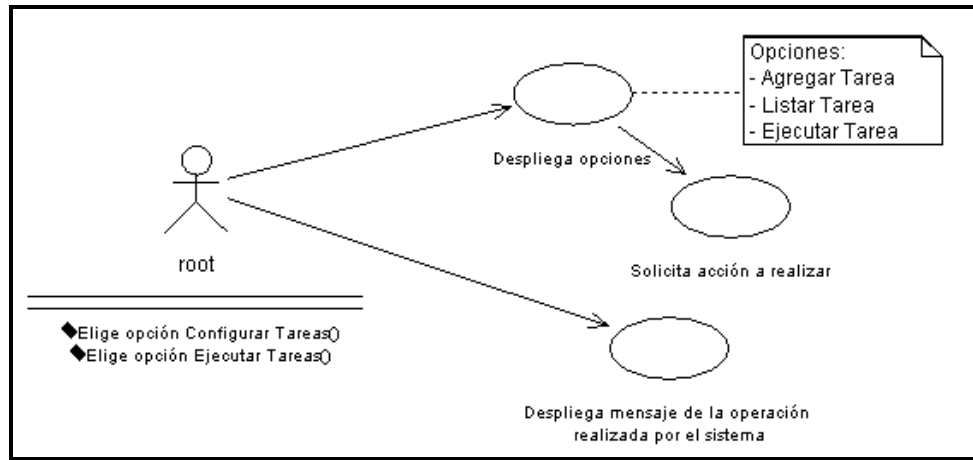
### Caso de uso 11: Agregar Tarea



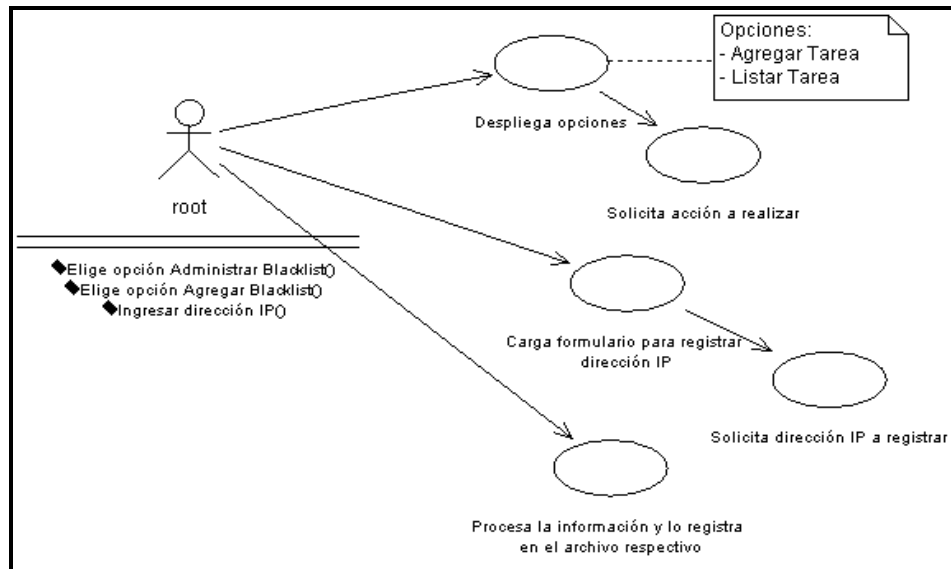
### Caso de uso 12: Listar Tareas



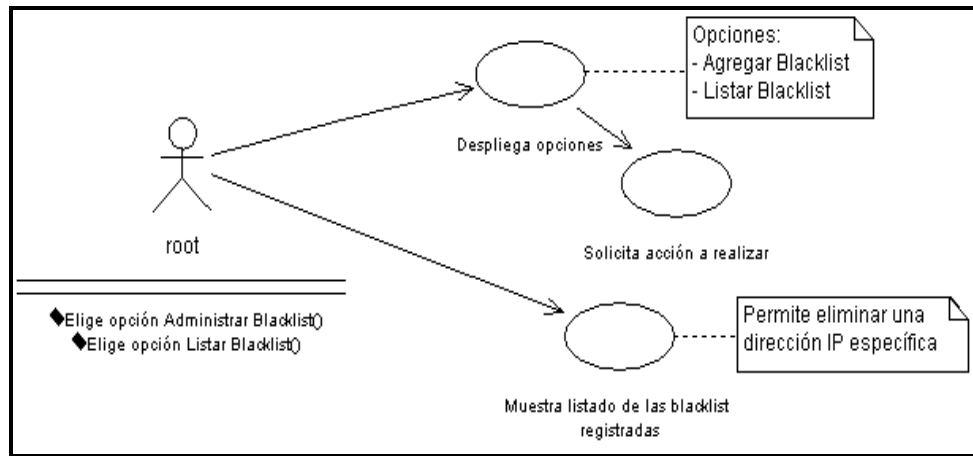
### Caso de uso 13: Ejecutar Tareas



### Caso de uso 14: Agregar Blacklist



### Caso de uso 15: Listar Blacklist



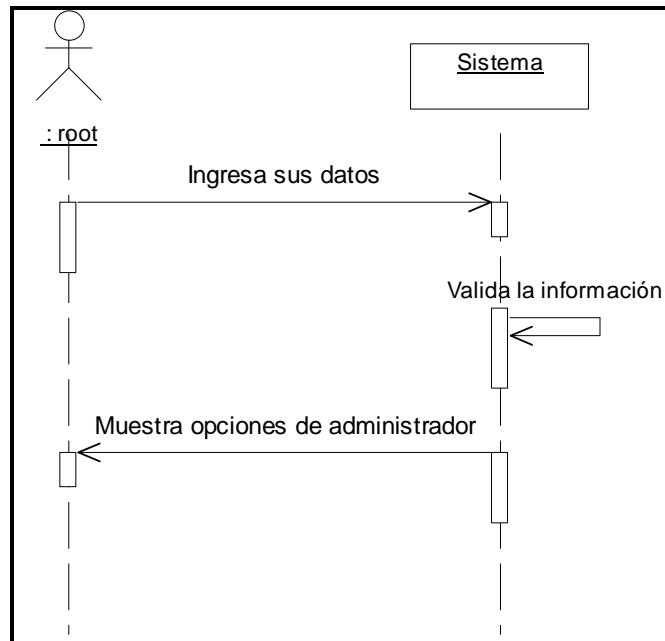
### 3.3. Diagramas de Interacción

#### 3.3.1. Diagrama de Secuencia

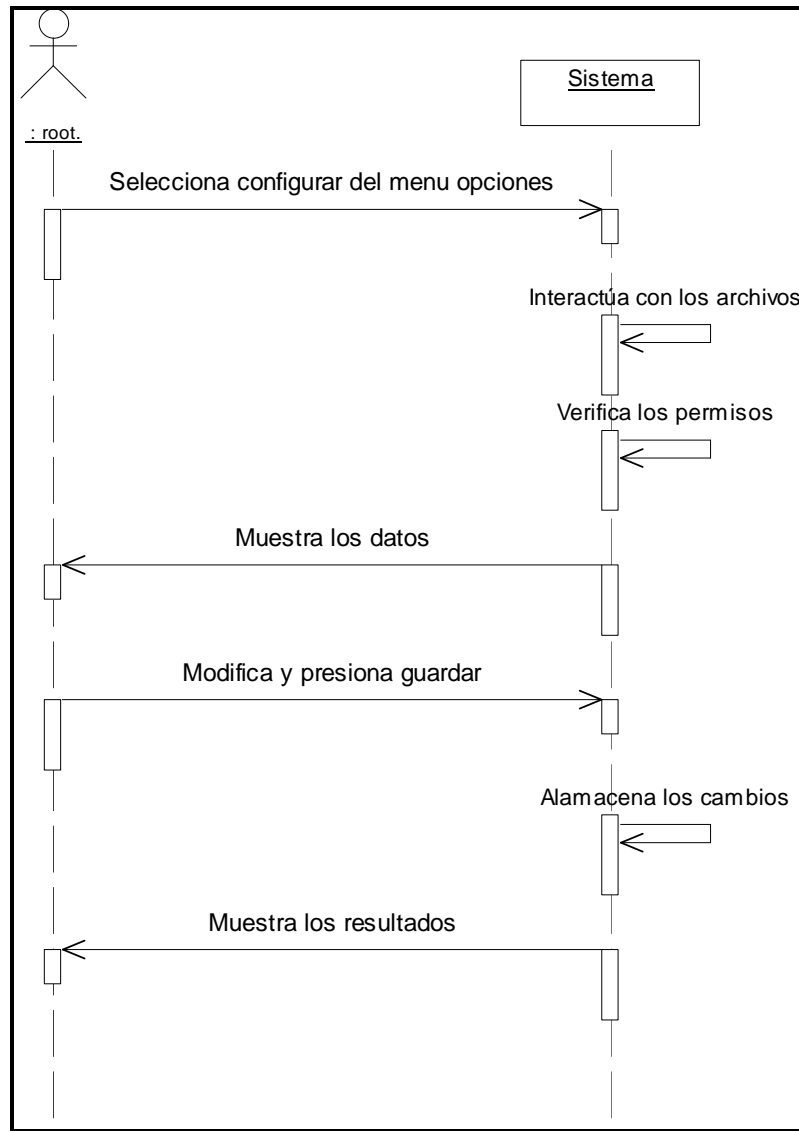
Muestran una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según una secuencia en el tiempo.

Los diagramas de secuencia desarrollados para "O.S. Security" son presentados en los siguientes gráficos.

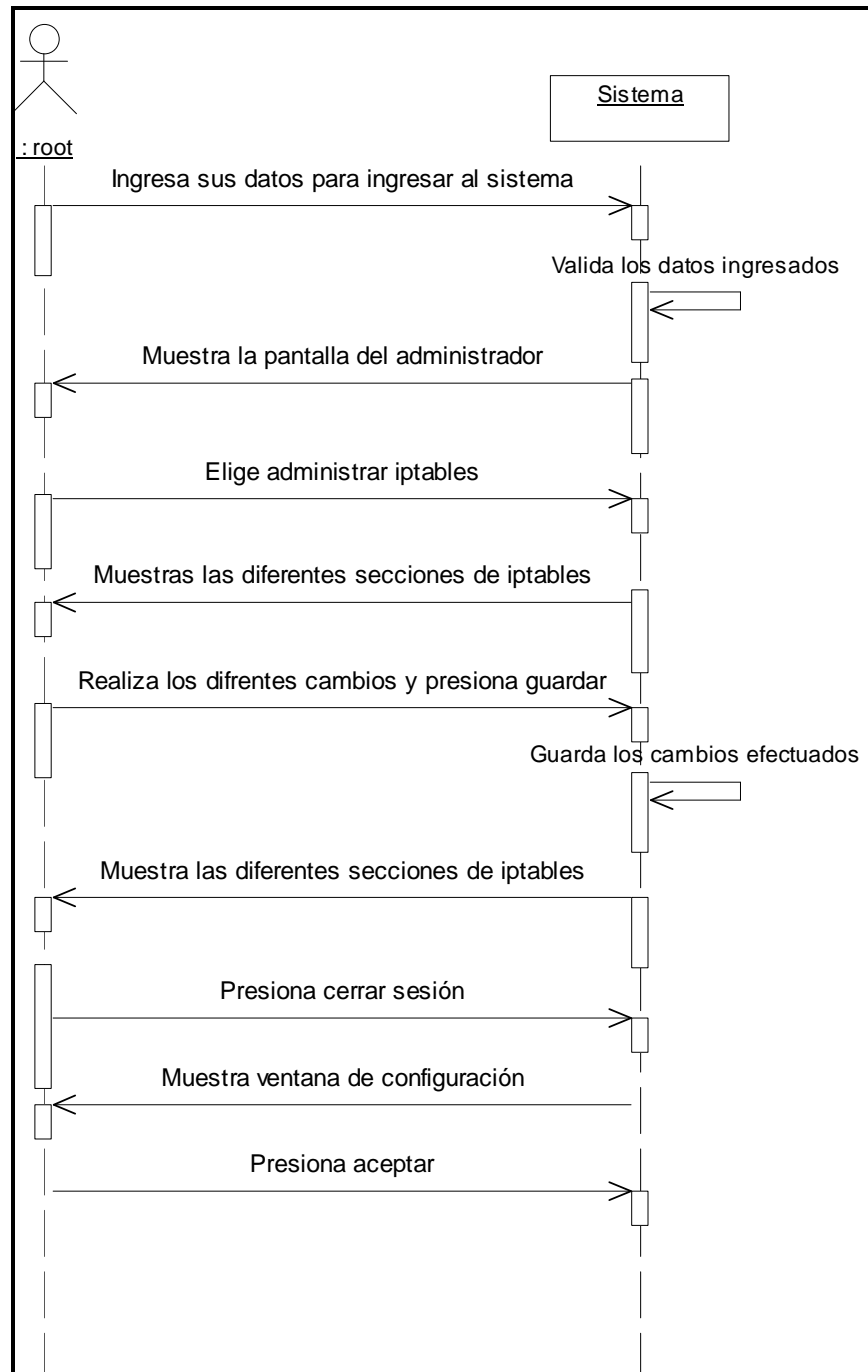
#### Caso de Uso 1: Validación de datos con el sistema



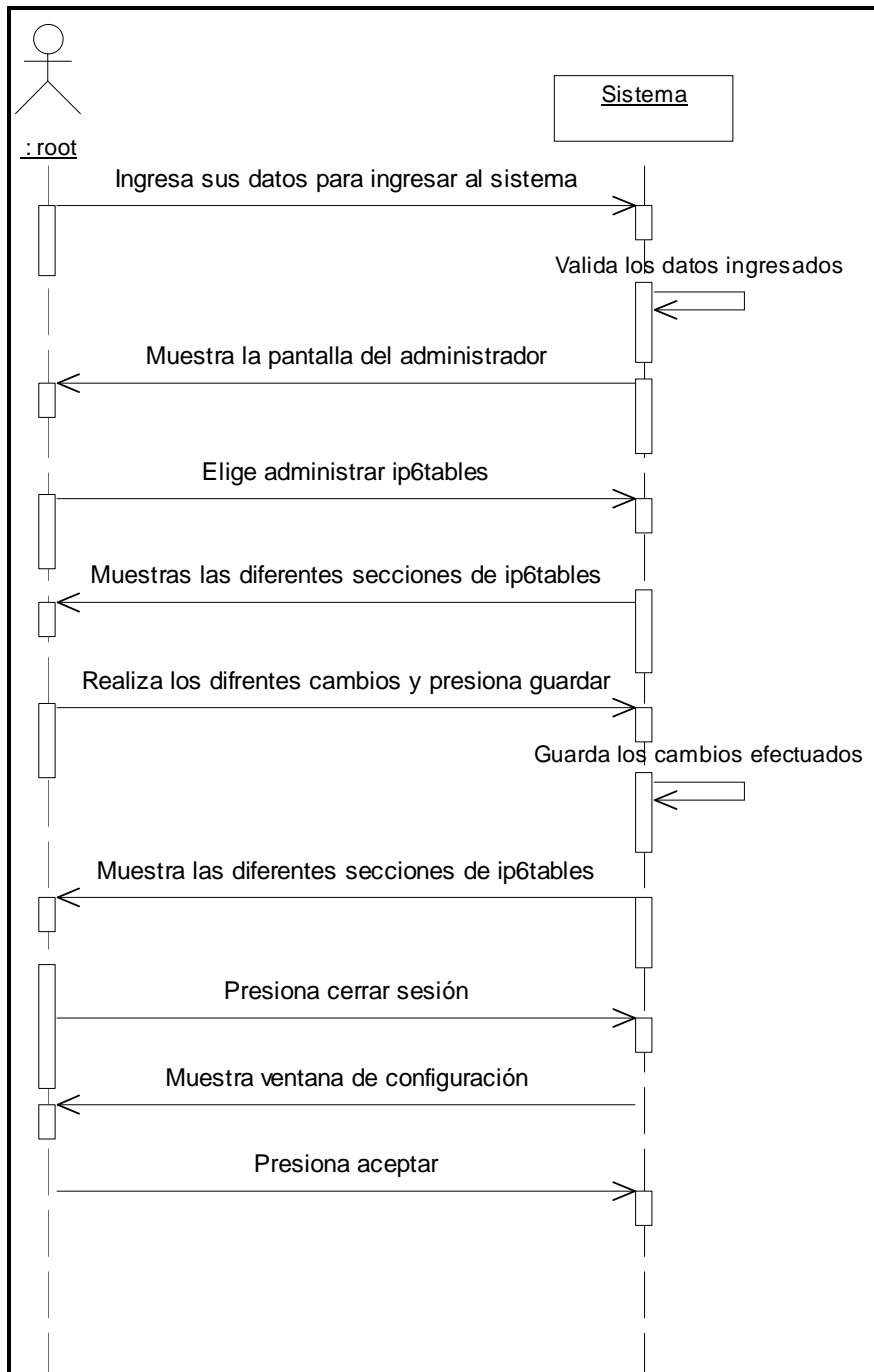
**Caso de Uso 2: Acceso y manipulación de archivos**



### Caso de Uso 3: Manipulación de Iptables

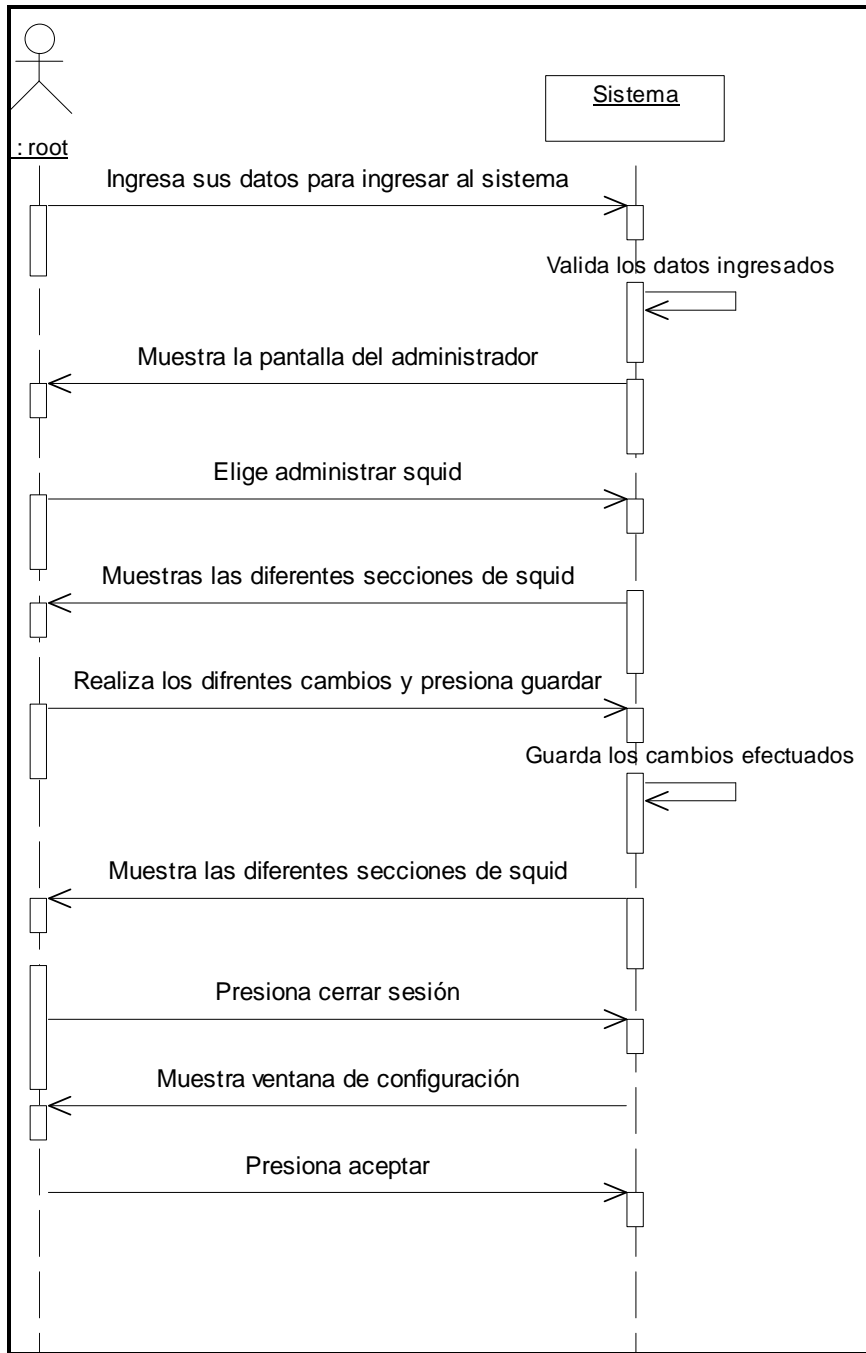


#### Caso de Uso 4: Manipulación de Ip6tables

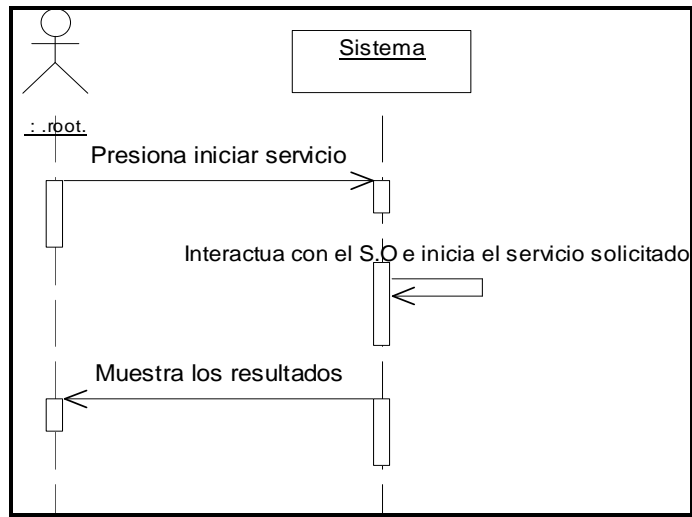




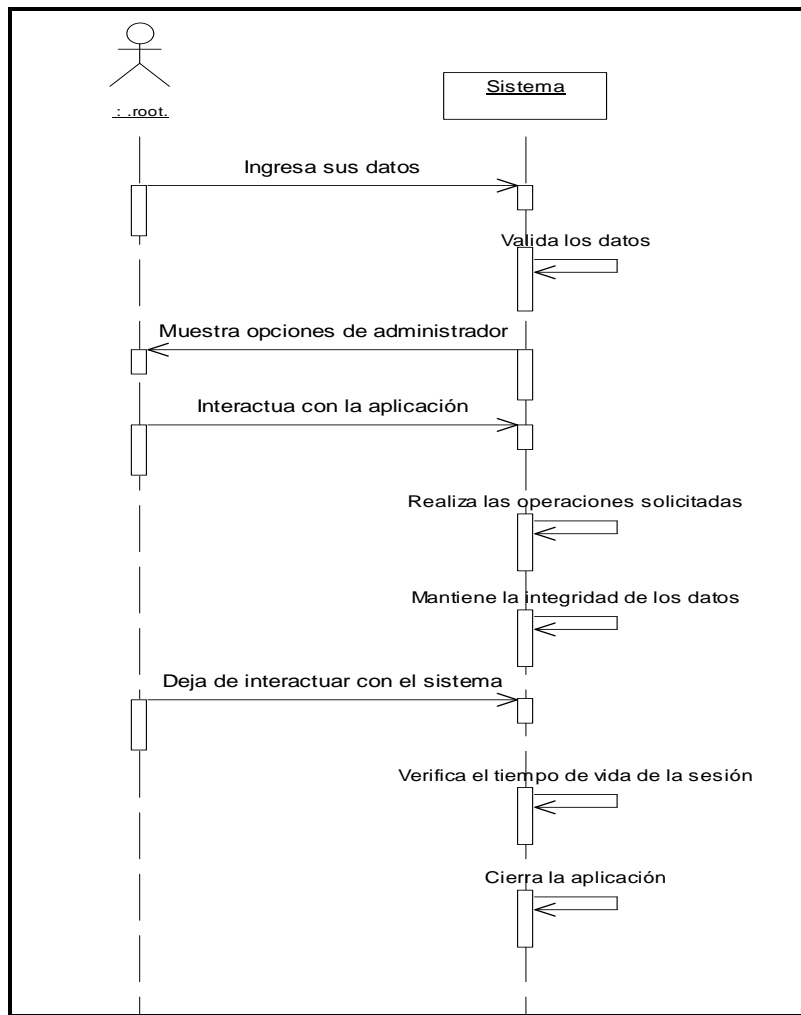
### Caso de Uso 5: Manipulación de Squid



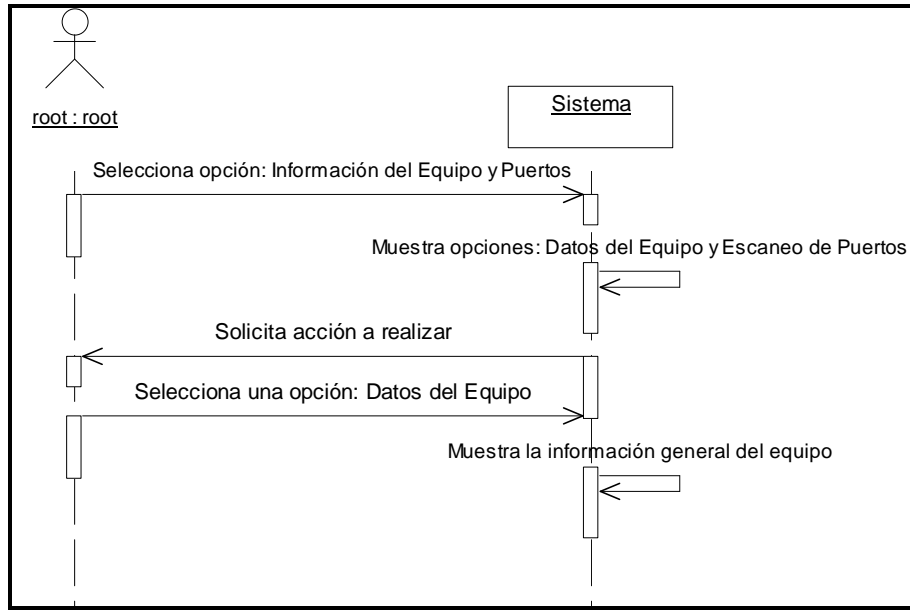
### Caso de Uso 6: Inicio y detención de servicios



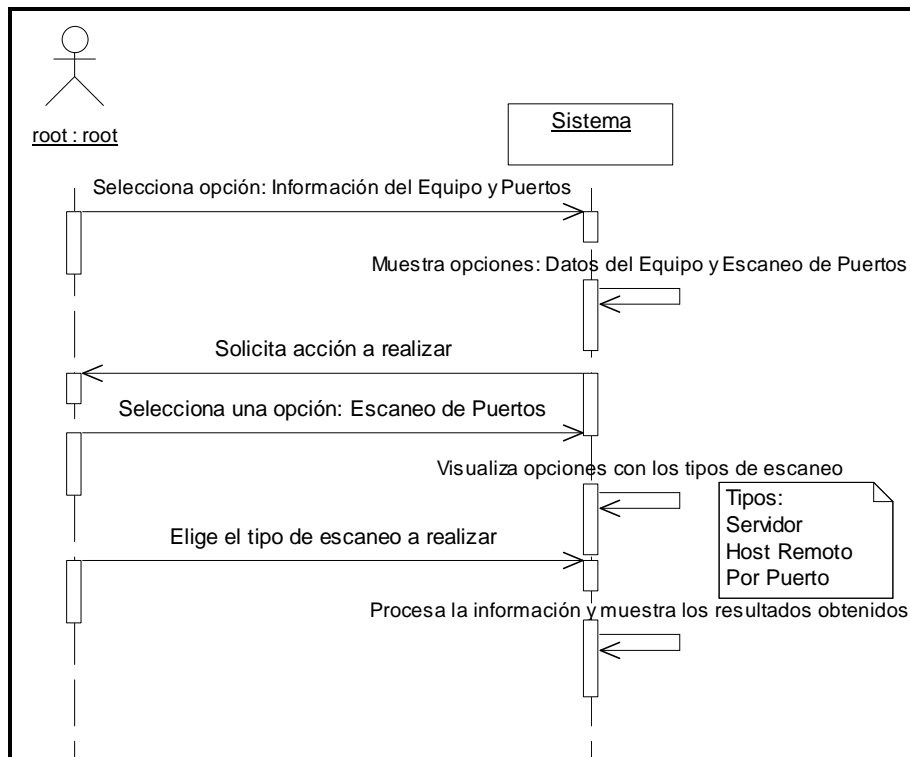
### Caso de Uso 7: Seguridad e Integridad



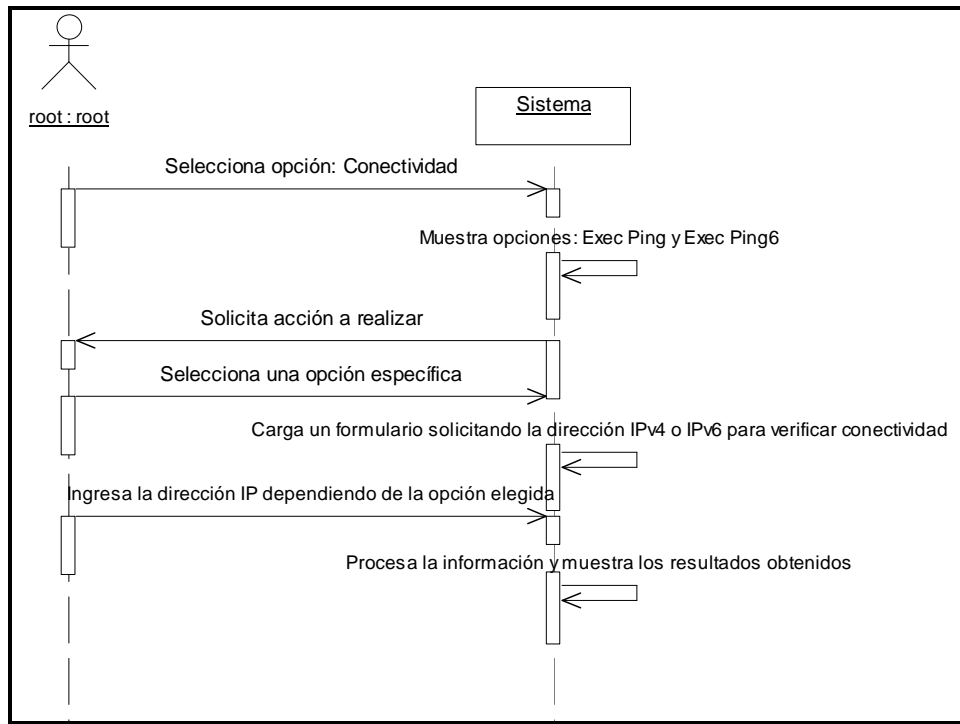
### Caso de uso 8: Datos del Equipo



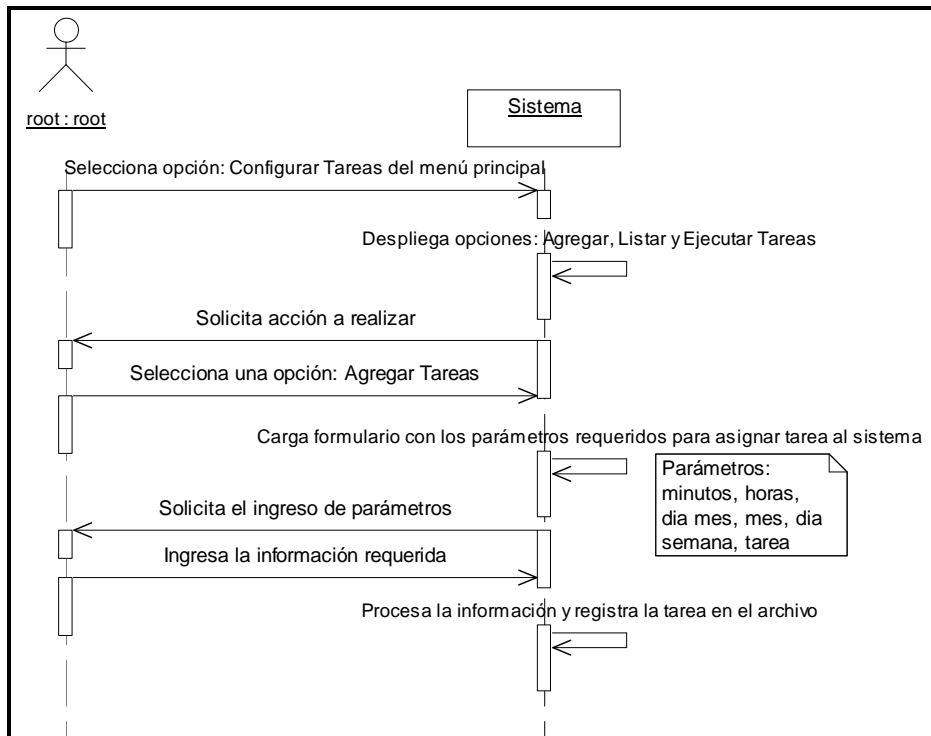
### Caso de uso 9: Escaneo de Puertos



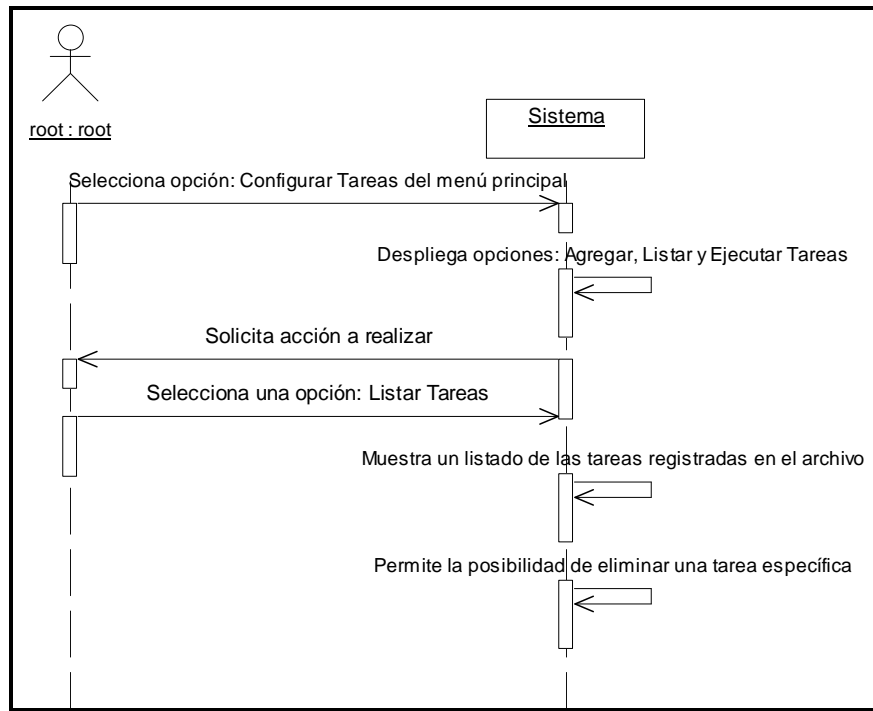
### Caso de uso 10: Conectividad entre Host



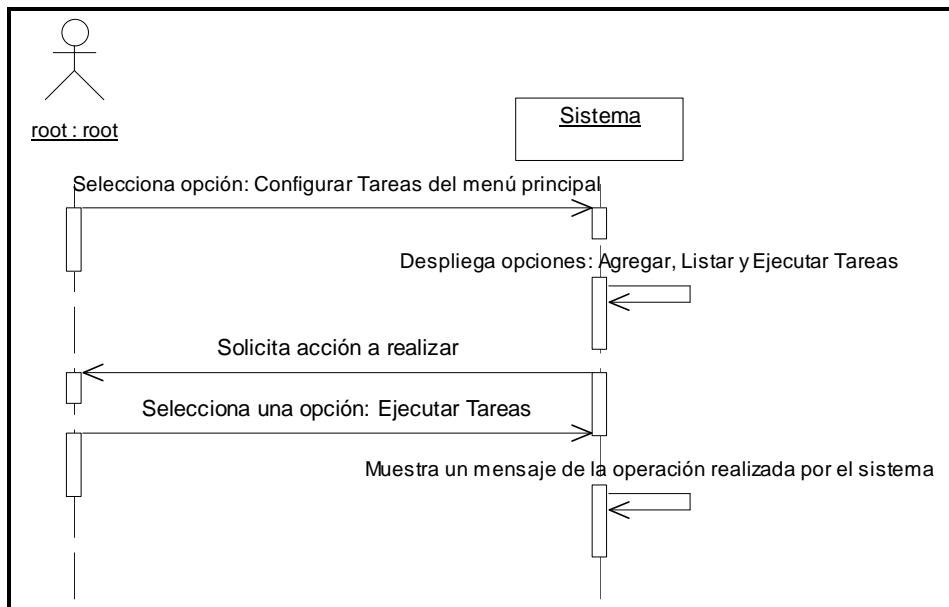
### Caso de uso 11: Agregar Tarea



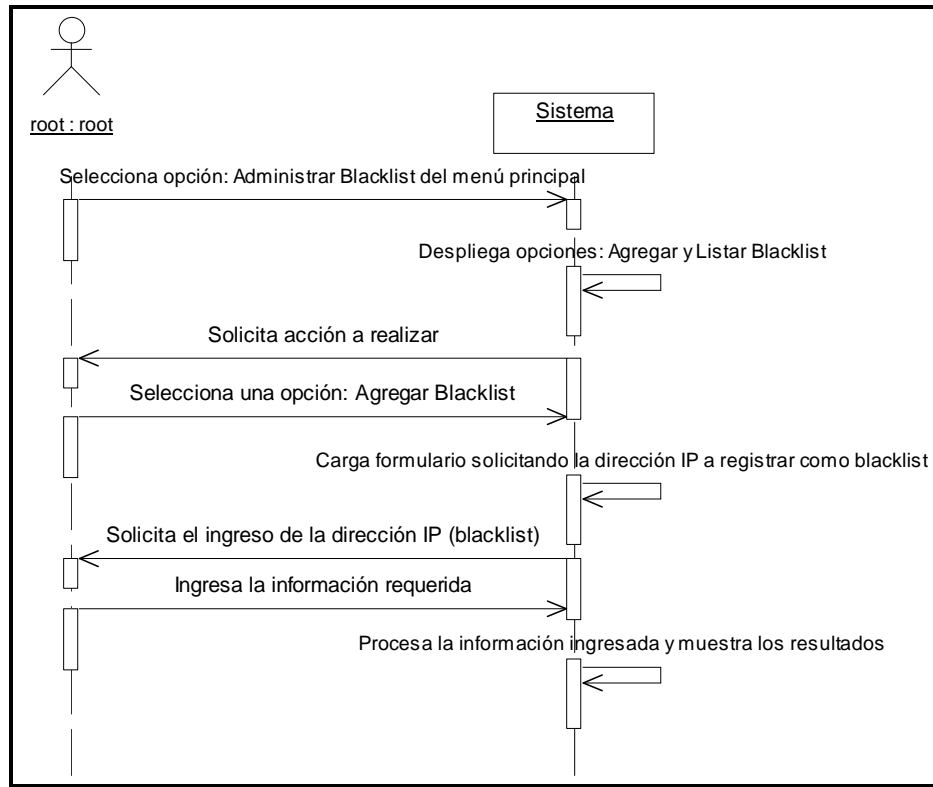
### Caso de uso 12: Listar Tareas



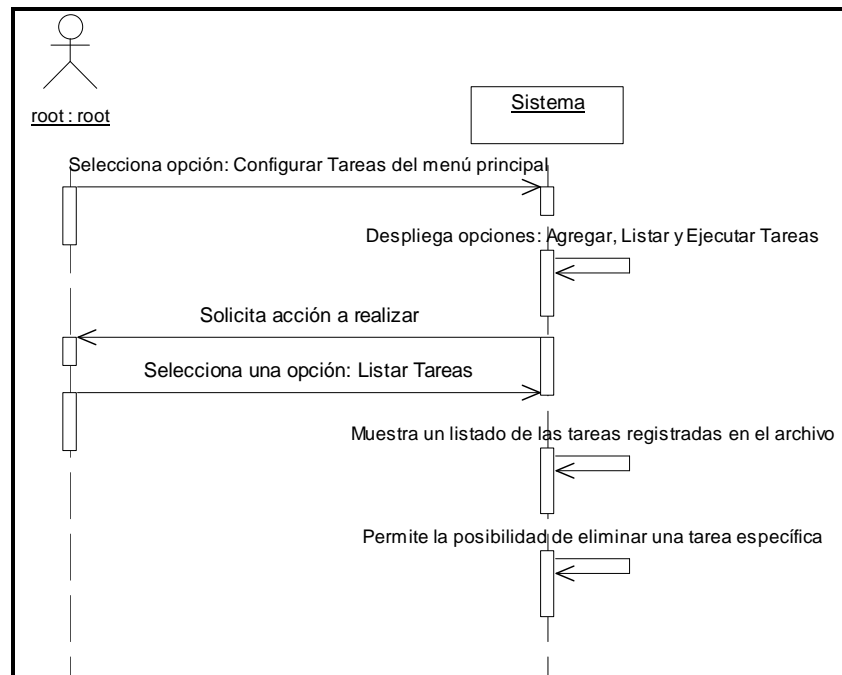
### Caso de uso 13: Ejecutar Tareas



### Caso de uso 14: Agregar Blacklist



### Caso de uso 15: Listar Blacklist



### 3.4. Diccionario de Clases y objetos

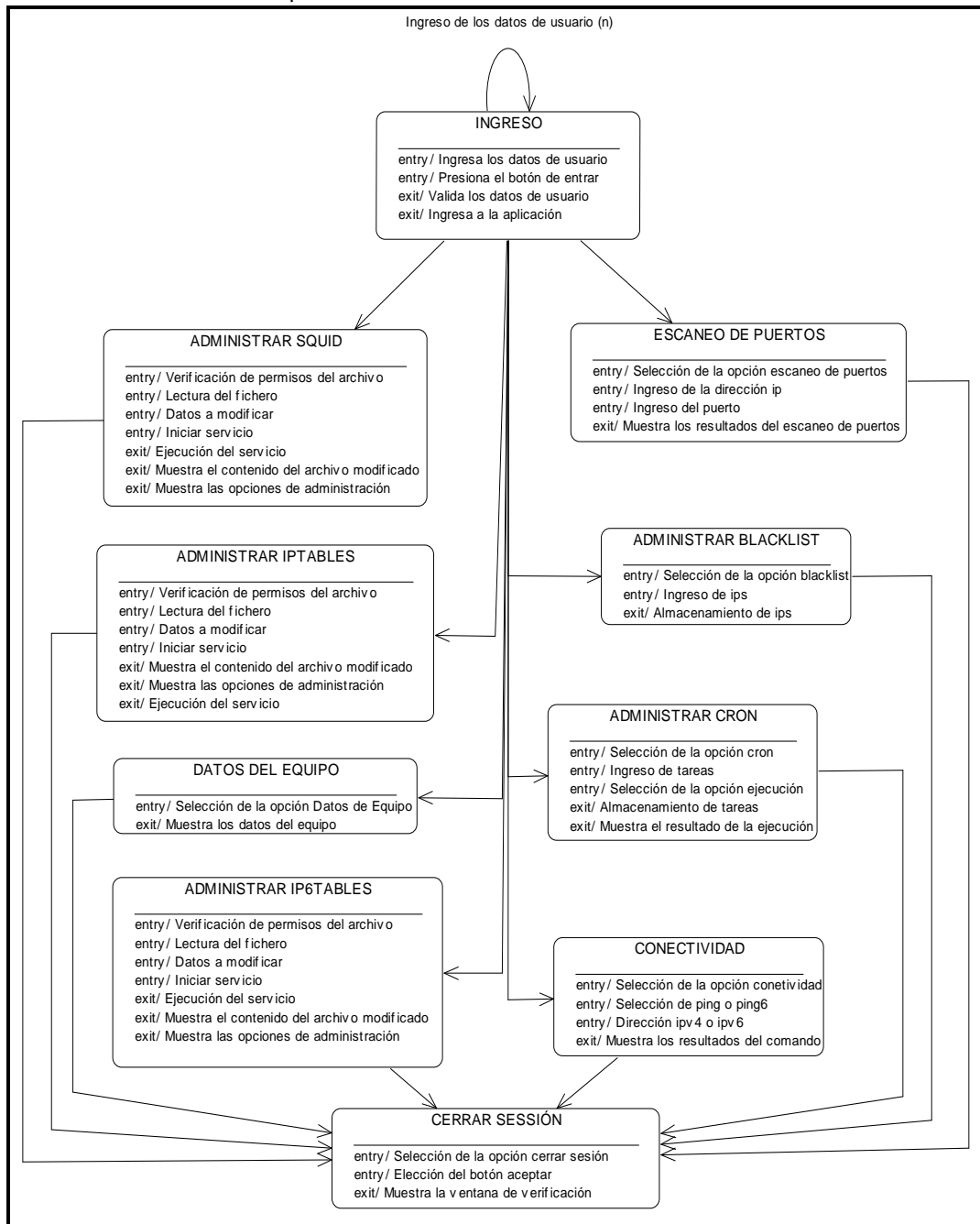
Clase	Característica	Comportamiento
Acceso_ficheros	Nombre: string[10] Ruta: string[100] Permisos_antiguos: string [4]	_constructor(nombre_archivo) Get_permisos() Set_permisos(permisos) Leer_archivo() Actualizar() Insertar_linea() Ordenar_linea() Eliminar_linea() Get_seccion(nombre_secc) Set_seccion (cambios)
lptables		Get_tabs() Get_form_ipv4() Inicio_fichero() Agregar_linea(fichero) Ordenar_fichero(fichero) Result_ejecucion(mensaje) Fin_fichero() Operac_mos(mensaje,operac, linea) Operación(mensaje,operac, linea) Star_service()
lp6tables		Get_tabs() Get_form_ipv6() Inicio_fichero() Agregar_linea(fichero) Ordenar_fichero(fichero) Result_ejecucion(mensaje) Fin_fichero() Operac_mos(mensaje,operac, linea) Operación(mensaje,operac, linea) Star_service()
Squid		Get_tabs() Get_form_squid() Inicio_fichero() Agregar_linea(fichero) Result_ejecucion(mensaje) Fin_fichero() Operac_mos(mensaje,operac, linea) Operación(mensaje,operac, linea) Star_service()
Controls		exec_script_fir(script,tipo) exec_squid(action) check_paquete(paquete) check_service(service) get_form_infoPC() get_form_Conex() duracion_sesion() local_scan_ports() Info_ping() Info_ping6()
Cron		diasMes() meses() diaSemana() CheckDias_Mes(month, year) ExisteFichero(path) NuevaTarea(path,cadena)

		updateFichero(linea,path) procesoAddTask(min,hordMes,mes,dSemana,comnd) ListadoTareas(ruta,param) CronOptions() execTaskCron(path)
Root		check_info_user(log,pass) get_user();
Blacklist		Busqueda(dato,vec) ListadoBlacklist(ruta,param) RegistrarIP(path,cadena)
page		draw_page() page_headers() get_head() contenido_left() get_foot() page_content()
Ports	Ports: string [20] Addr: string [36] Sockets: string[40] Services: strign[60] Ban: string[50]	sockets_enabled() set_ports(ports) set_addr(addr) tcp_scan_port(addr,port) udp_scan_port(addr,port) run(type) ScanPuertoEspecifico(sp,dir) RemoteScanPorts(ip,pi,pf,t)



### 3.5. Diagramas de Estado

Este diagrama nos permite ver, cuales son los estados por los que pasa el usuario al interactuar con la aplicación web.

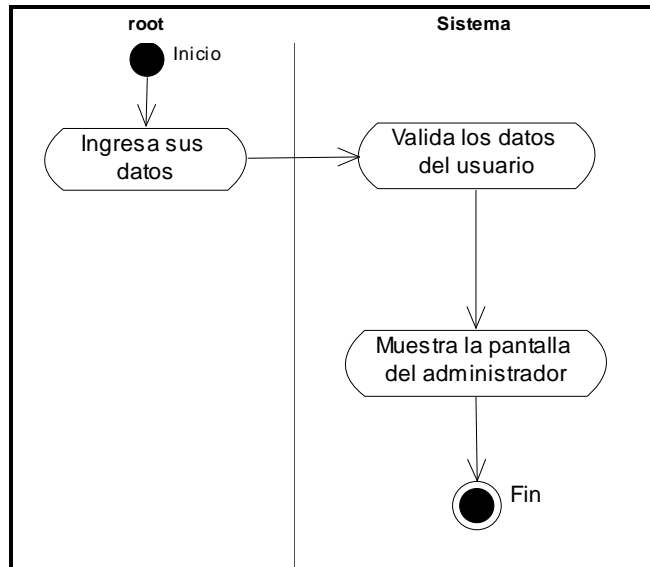


### 3.6. Diagrama de Calles

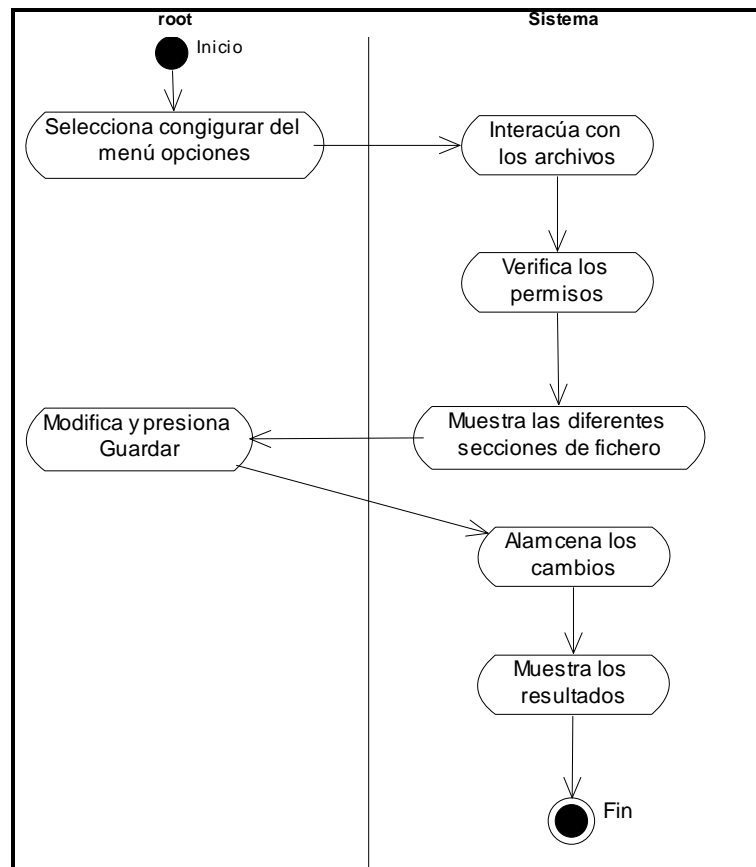
Cuando se modelan flujos de trabajo de organizaciones, es especialmente útil dividir los estados de actividades en grupos, cada grupo tiene un nombre completo y se denominan calles. Cada calle representa a la parte de la Organización responsable de

las actividades que aparecen en esa calle, gráficamente quedaría representado como se muestra en el siguiente gráfico.

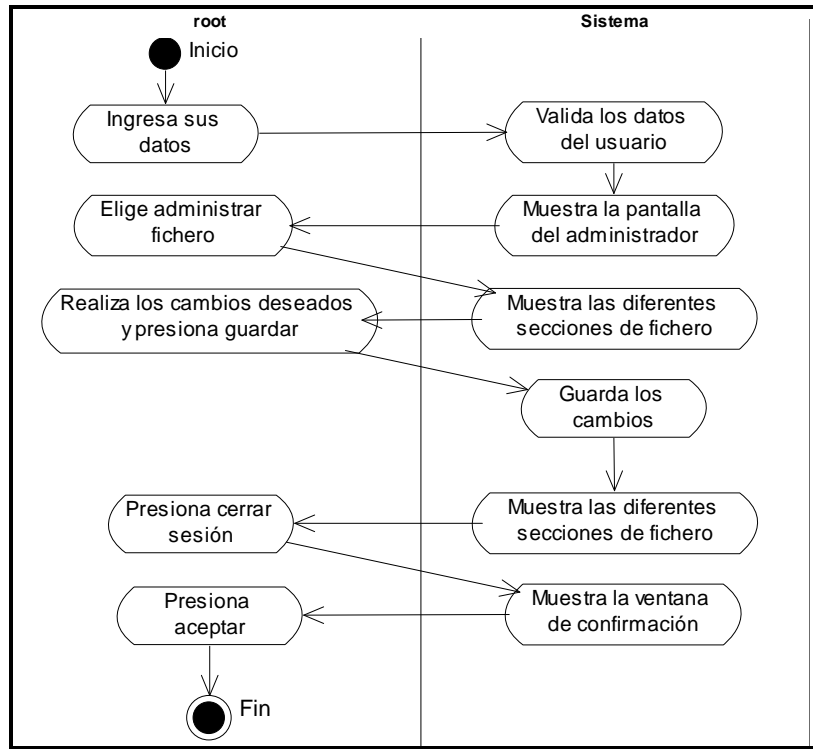
**Caso de Uso 1:** Validación de datos con el sistema



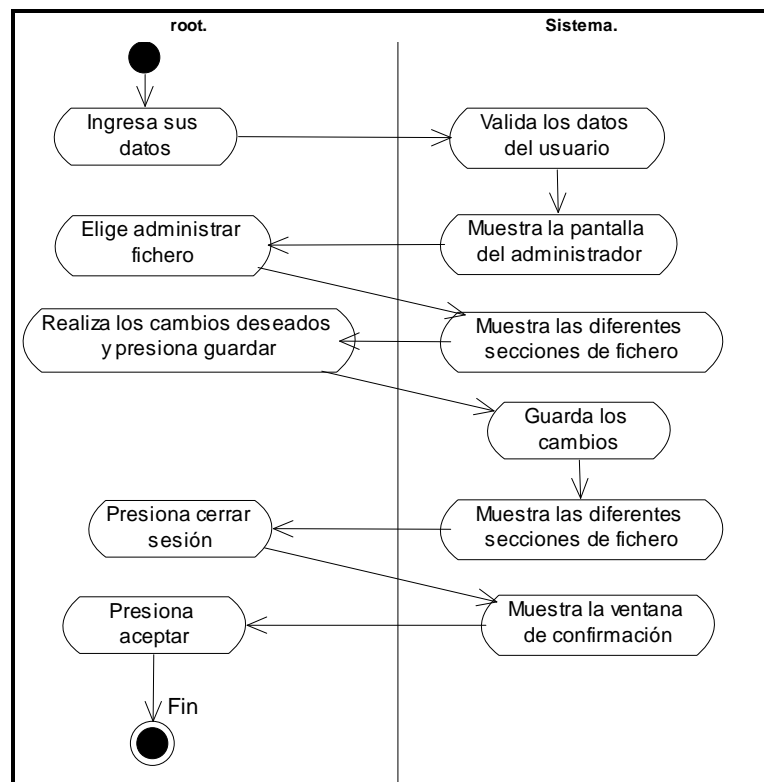
**Caso de Uso 2:** Acceso y manipulación de archivos



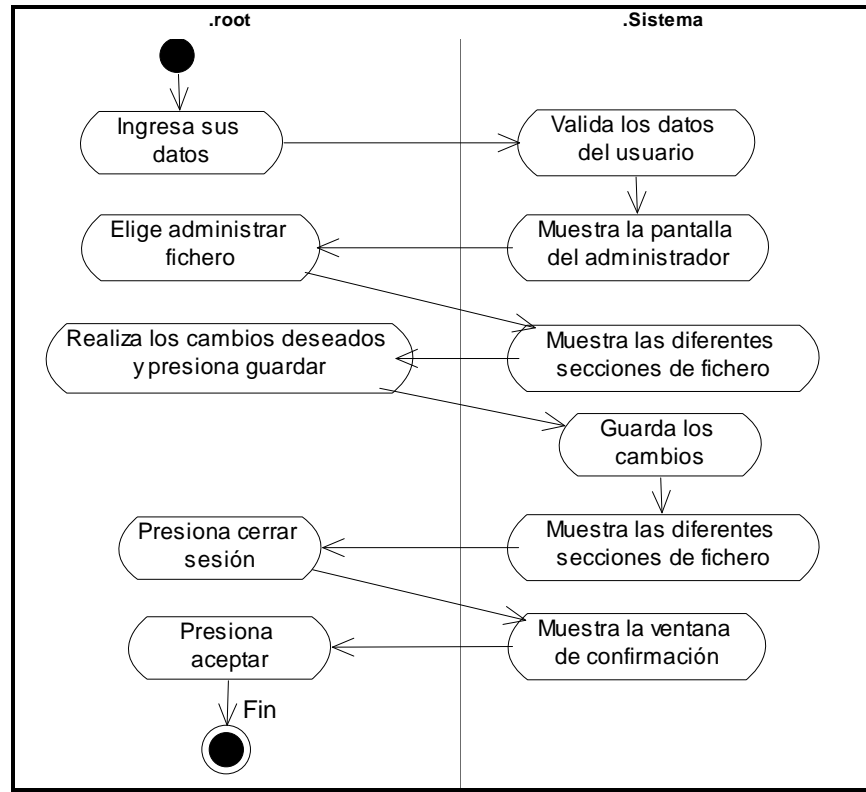
### Caso de Uso 3: Manipulación de Iptables



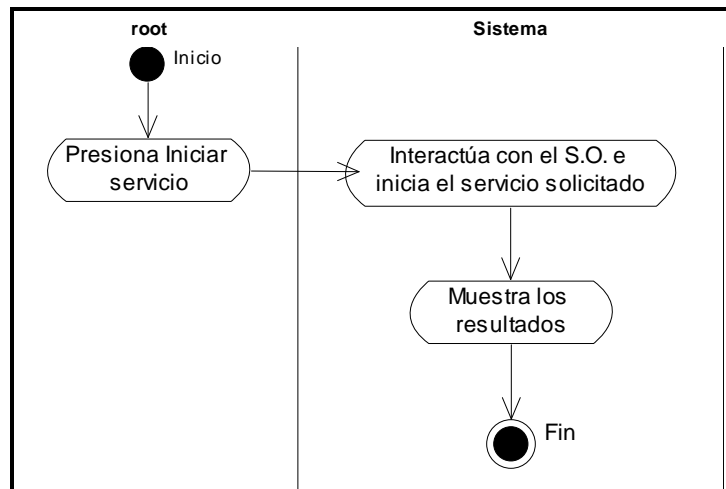
### Caso de Uso 4: Manipulación de Ip6tables



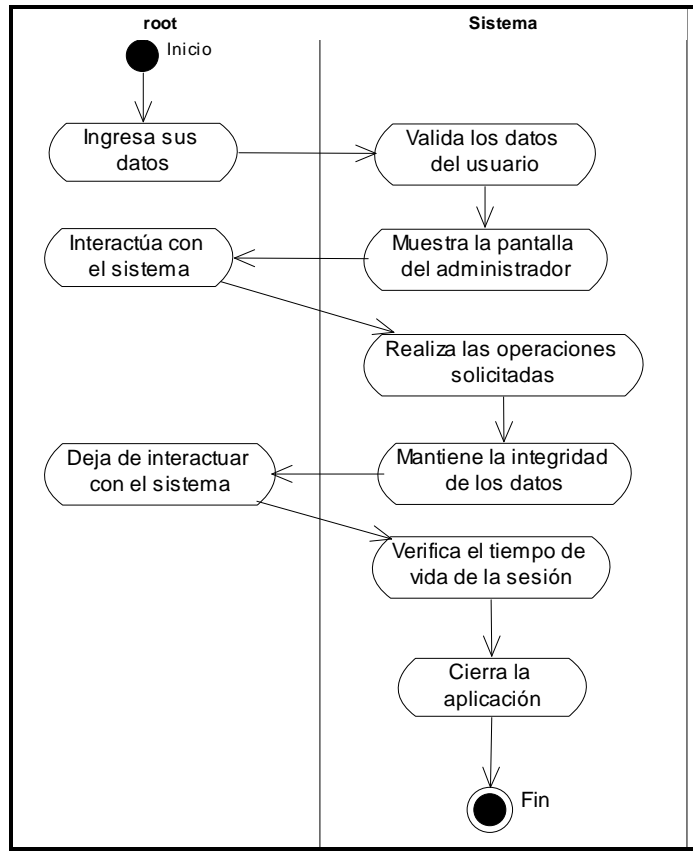
### Caso de Uso 5: Manipulación de Squid



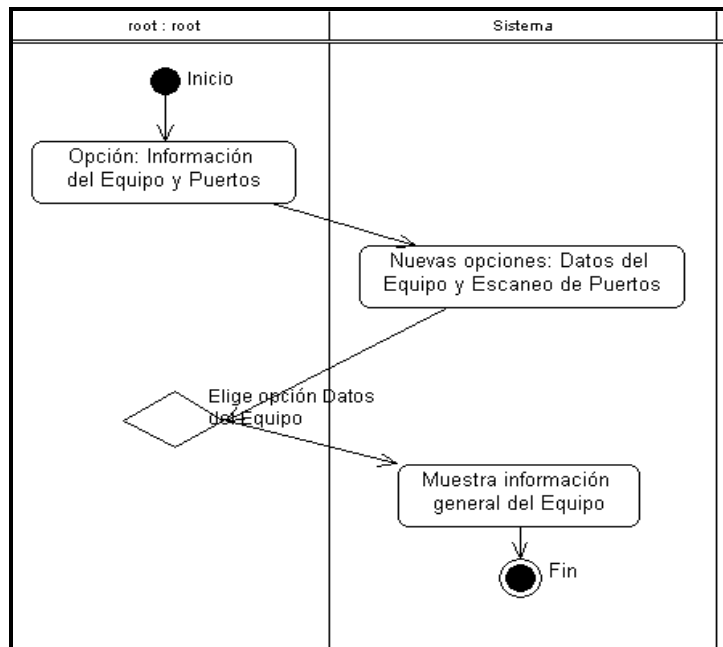
### Caso de Uso 6: Inicio y detención de servicios



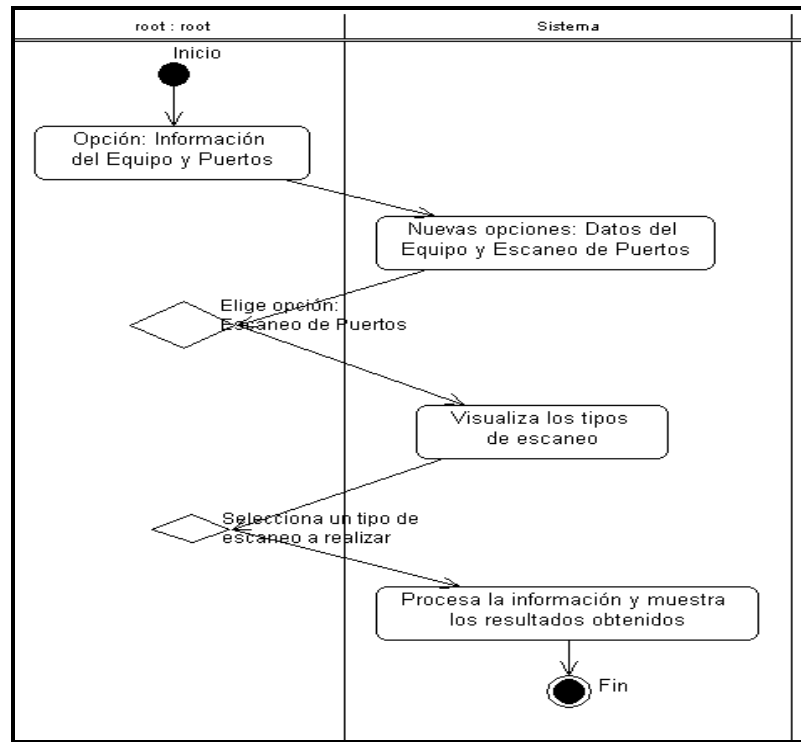
### Caso de Uso 7: Seguridad e Integridad



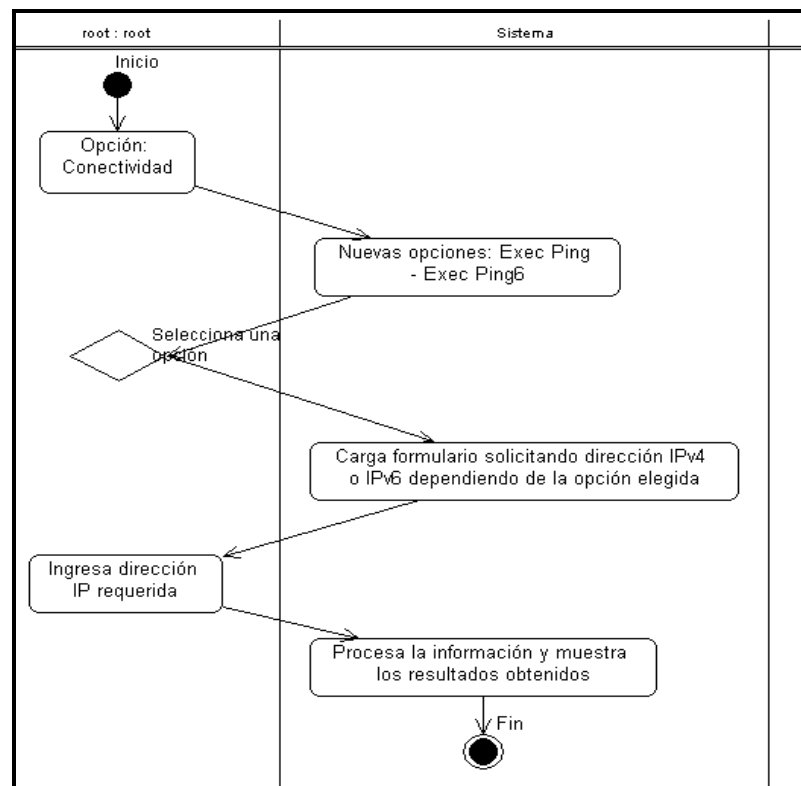
### Caso de uso 8: Datos del Equipo



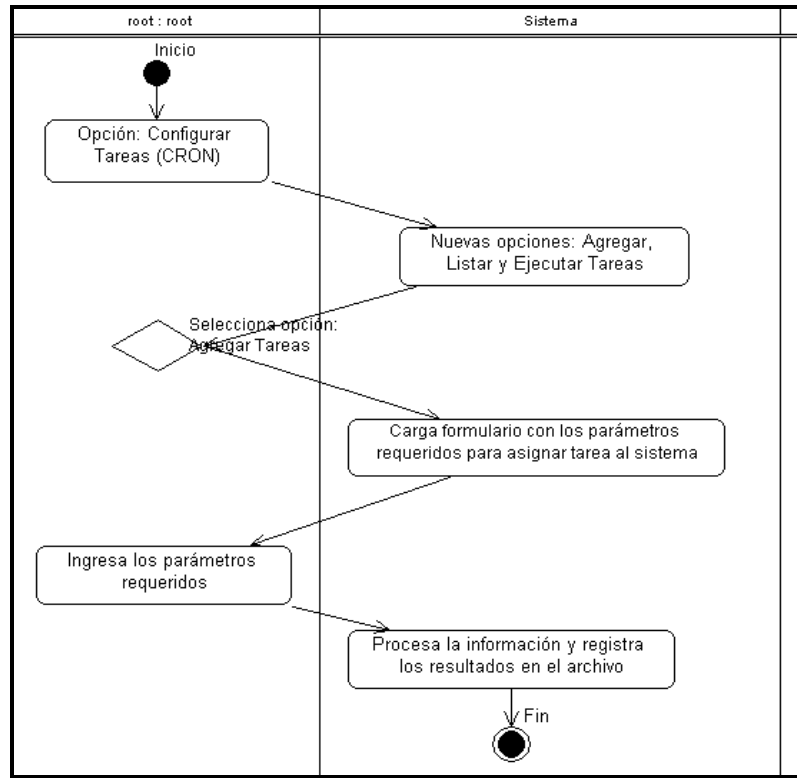
### Caso de uso 9: Escaneo de Puertos



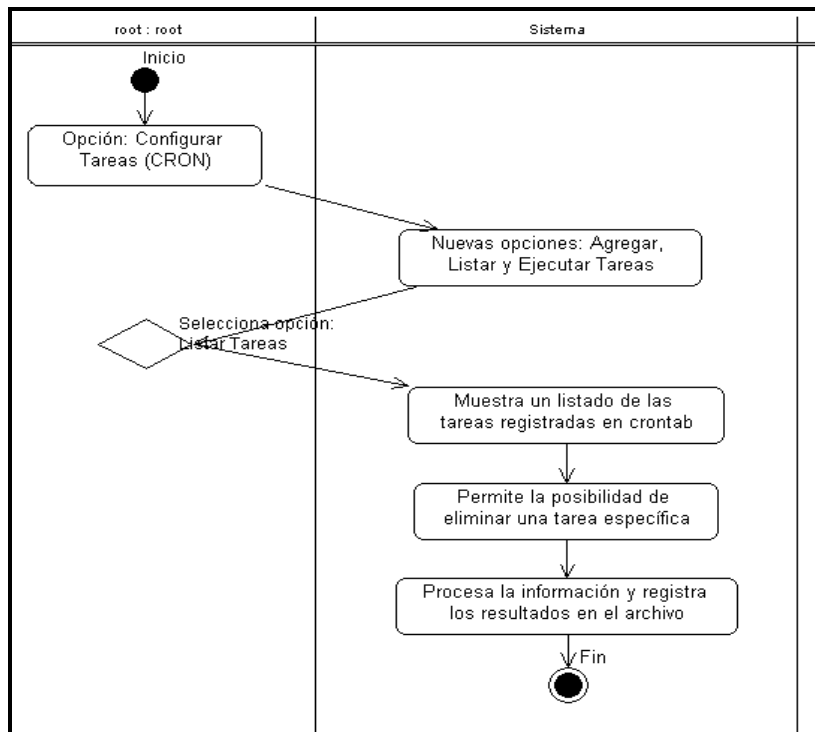
### Caso de uso 10: Conectividad entre Host



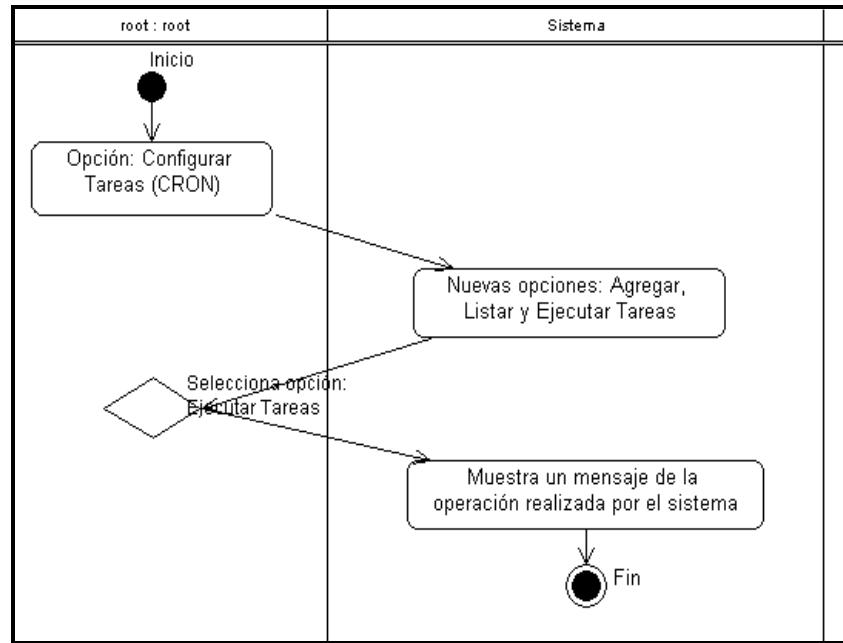
### Caso de uso 11: Agregar Tarea



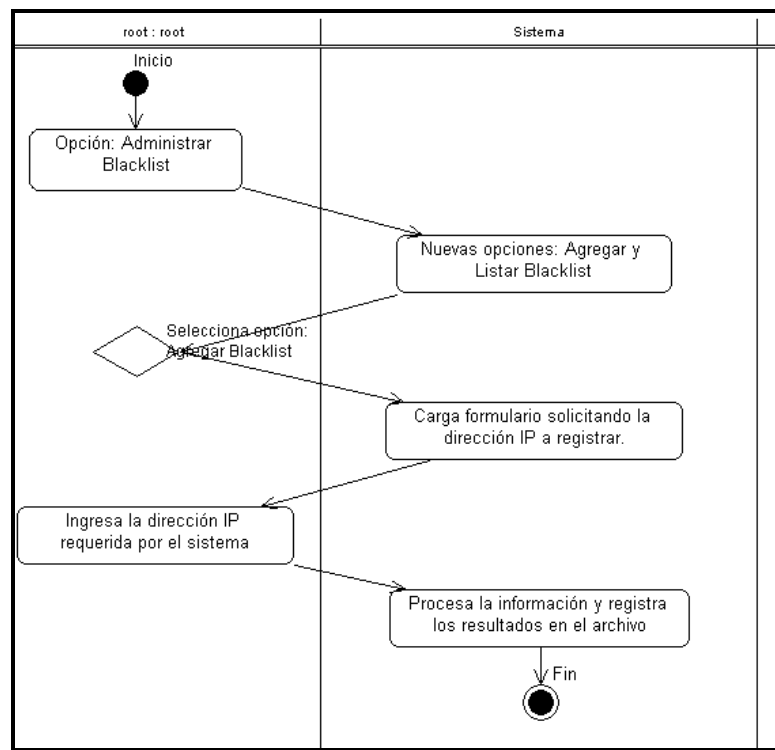
### Caso de uso 12: Listar Tareas



### Caso de uso 13: Ejecutar Tareas

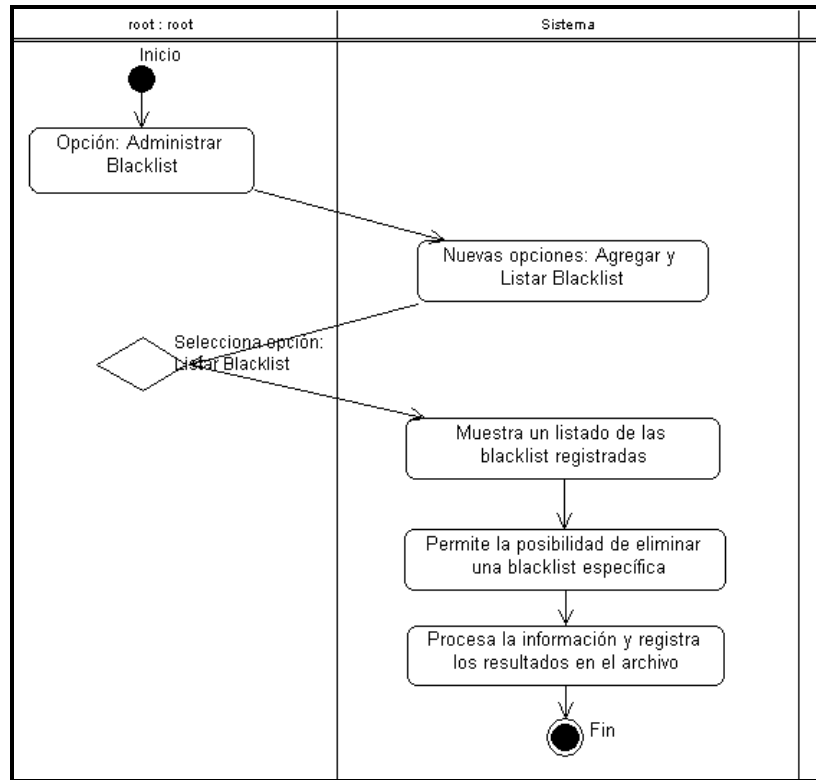


### Caso de uso 14: Agregar Blacklist





### Caso de uso 15: Listar Blacklist

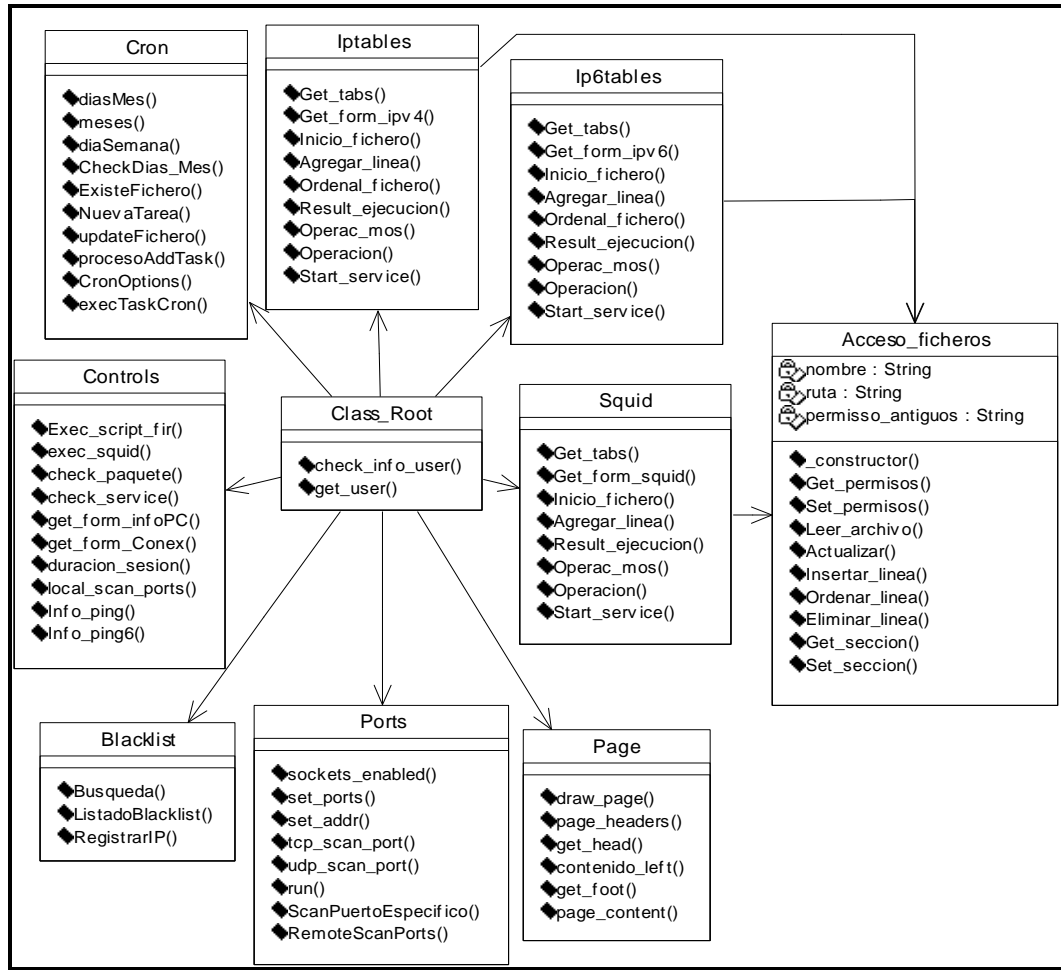


## 4. DISEÑO

En esta fase se pretende crear una solución a nivel lógico para poder lograr satisfacer todos los requisitos, basándose en el conocimiento reunido en la fase de análisis. Esta fase se constituirá en la más importante pues la especificación de cada una de sus actividades se las hace con más detalle de tal forma que se pueda ver la utilización física del sistema.

### 4.1. Diagrama de Clases

En la especificación del Diagrama de Clases de Diseño se mostrará la especificación para las clases de Software del Sistema



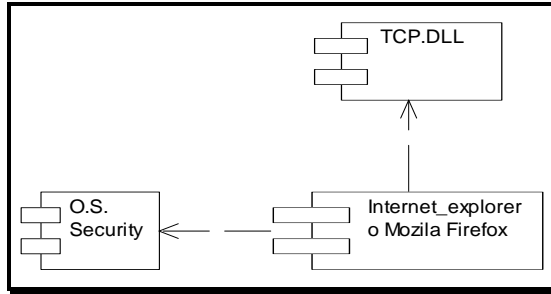
## 4.2. Modelo Físico

En esta actividad presentaremos los 2 tipos de Diagramas para modelar los aspectos físicos de un sistema orientado a objetos:

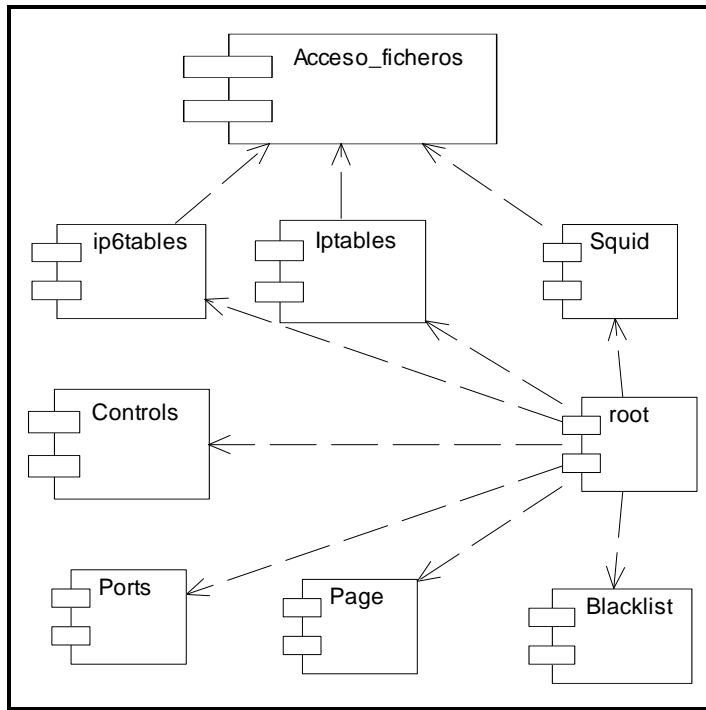
1. Diagrama de Componentes
2. Diagrama de Despliegue.

### 4.2.1. Diagrama de Componentes

En primer lugar daremos el diagrama correspondiente de los componentes requeridos para poder acceder a la aplicación.

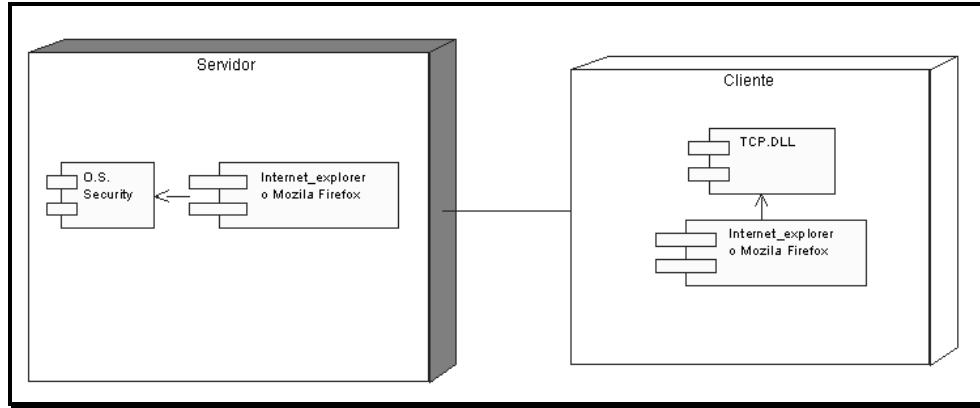


Ahora veremos la constitución del componente O.S. Security:



#### 4.2.2. Diagrama de Despliegue

En estos diagramas se representan dos tipos de elementos, *nodos* y *conexiones*, así como la distribución de componentes del sistema de información con respecto a la partición física del sistema.



## 5. IMPLEMENTACIÓN

Tras haber definido como desarrollar la aplicación web "O.S. Security", procedemos a implementar la misma.

## 5.1. Codificación

La codificación del Sistema se realizó en Macromedia Dreamweaver 8, utilizando el lenguaje PHP, parte del código se puede ver en las imágenes mostradas a continuación:

### Código PHP

Página índice:

Se puede observar que se hace una clase entrada la cual corresponde a la Página de Entrada al Sistema, esta es una página conocida como manejadora, la cual hace uso de clases para poder realizar las operaciones debidas.

```

<?php
require_once ( $_SERVER['DOCUMENT_ROOT'] . "/security/config.php" );
require_once ( CLASS_PATH . "class.page.php");
require_once ( CLASS_PATH . "class.contenido.php");
require_once ( CLASS_PATH . "class.date.php" );
include("javascript.php");
//include_once("mensajes.php");

//ojo si uso estas 2 lineas, es como colocar register_globals=on, no es correcto
//extract($_GET);
//extract($_POST);

$cont=new class_content_page();
$page=new class_page();

$op=$_REQUEST['action'];

if(!isset($op)) $op='inicio';

switch($op)
{
    case 'inicio': $page->content=$cont->get_form(); break;
    case 'acerca': $page->content=$cont->acerca(); break;
    default: echo "Error";
}
$page->draw_page();
?>
```

La página contiene referencias de las clases que esta manejadora hace uso, para poder mostrarnos la página index, que es la página en la que un usuario ingresa sus datos para ingresar a la aplicación.

Ahora veremos varios ejemplos de programación en PHP.

### Procedimiento Leer Archivo

```
//
//leemos el archivo
public function leer_archivo()
{
    $retval = "";
    $aux = "";

    $gestor = fopen($this->ruta,'rb');
    while (!feof($gestor))
    {
        $aux .= fread($gestor, filesize($this->ruta));
    }
    fclose($gestor);
    $retval = split("\n",$aux);
    return $retval;
}
```

### Procedimiento Actualizar Archivo

```
//  
//para actualizar el archivo con cualquier cambio  
public function actualizar($actual)  
{  
    $retval = "";  
    $aux = join("\n", $actual);  
    $gestor = fopen($this->ruta, 'w+');  
    if (fwrite($gestor, $aux) == FALSE)  
        $retval = 'No se pudo escribir en el archivo';  
    else  
        $retval = 'Hecho';  
    fclose($gestor);  
    return $retval;  
}
```

### Procedimiento Get Sección

```
//  
//  
//para obtener una sección determinada del archivo  
public function get_seccion( $seccion )  
{  
    //$archivo = $this->leer_archivo();  
    $archivo = file($this->ruta);  
    $begin_title = "# BEGIN " . $seccion;  
    $end_title = "# END " . $seccion;  
    $band = 0;  
  
    foreach( $archivo as $indi => $value )  
    {  
        if ( trim($value) == $end_title )  
            break;  
        if ( $band )  
            $retval[] = $value;  
        if ( trim($value) == $begin_title )  
            $band = 1;  
    }  
    return $retval;  
}
```

### Uno de los procedimientos utilizados para mostrar la interfaz

```
// En esta funcion presentaremos todo lo referente a la cabecera de un documento  
HTML  
// referencias a hojas de estilo titulo de la pagina  
public function page_headers()  
{  
    $retval = "<html>\n";  
    $retval .= "    <head>\n";  
    $retval .= "        <link rel=\"shortcut icon\" href=\"favicon.ico\">";  
    $retval .= "        <title> O.S Security \"Open Source Security\" </title>\n";  
    $retval .= "        <link href=\"style.css\" type=\"text/css\"  
rel=\"stylesheet\"></link>\n";  
    //  
    //linea añadida 08/01/12
```

```

        $retval .= "        <link rel=\"stylesheet\" type=\"text/css\" href=\"http://\" .
$_SERVER['HTTP_HOST'] . "/security/scripts/tab-view.css\" />";
        $retval .= "        <link rel=\"stylesheet\" type=\"text/css\" href=\"http://\" .
$_SERVER['HTTP_HOST'] . "/security/scripts/menu.css\" />";
        $retval .= "        <script type=\"text/javascript\" src=\"http://\" .
$_SERVER['HTTP_HOST'] . "/security/scripts/tab-view.js\"></script>";

        $retval .= '
<script lenguaje="javascript">
function OpenClose(btnId, divId)
{
if(document.getElementById(btnId).style.fontWeight == \'bold\')
{
document.getElementById(btnId).value =
document.getElementById(btnId).value.replace("-", "+");
document.getElementById(btnId).style.fontWeight= \'normal\';
document.getElementById(divId).style.display= \'none\';
}
else
{
document.getElementById(btnId).value =
document.getElementById(btnId).value.replace("+", "-");
document.getElementById(btnId).style.fontWeight= \'bold\';
document.getElementById(divId).style.display= \'block\';
}
}
}
</script>
';

//
$retval .= "    </head>\n";
return $retval;
}

```

## 5.2. Ejecución de los Programas

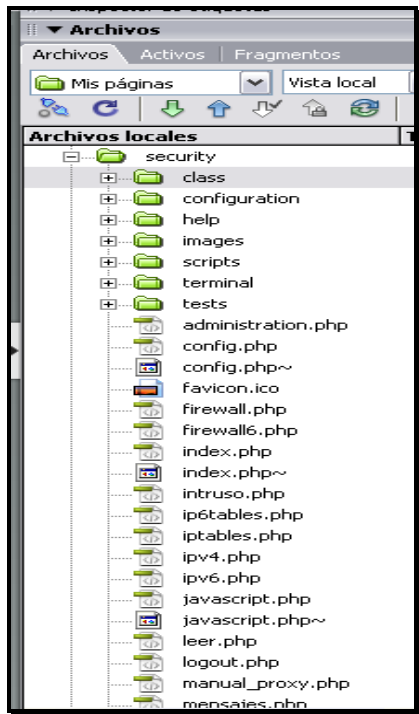
Para ejecutar la aplicación se lo puede hacer mediante el uso del teclado (F12), teniendo configurado el servidor de pruebas, que está en el submenú “Administrar Sitios” del menú “Sitios” ó presionando el ícono con forma de globo terráqueo ubicado en la parte superior de la pantalla como se muestra a continuación:



## 5.3. Organización de Datos

Dreamweaver 8 nos permite organizar los archivos en forma de árbol, y para poder tenerlos siempre al alcance, se dispone de un árbol que inicia en la carpeta configurada como raíz al momento de administrar los sitios. Podemos ver al lado derecho de la ventana de Dreamweaver 8 como están ubicados nuestras carpetas y archivos.

En nuestro caso tenemos varias carpetas que contienen los archivos que tienen el mismo propósito, podemos ver la organización de nuestros archivos en la gráfica siguiente:





## **ANEXO C**

### **“O.S Security” Manual de Usuario**

# **Manual de Usuario “Open Source Security”**

## **COPYRIGHT**

Bajo las leyes de derecho de autor, la Documentación, el Software, junto con sus elementos no pueden ser copiados, fotocopiados, reproducidos, trasladados o reducidos a cualquier medio electrónico, en forma parcial o total, sin la previa autorización escrita de Leonardo Gualpa C. y Marco Malán M, desarrolladores del producto.

© 2007 – 2008 “Open Source Security”

**Todos los derechos reservados**

## 1. Introducción

El presente Manual expone el ambiente de trabajo de la aplicación Web “**Open Source Security**”, realizando una descripción de sus interfaces así como los principales objetos que forman parte del producto; de tal manera que el administrador (root) no tenga inconvenientes al usar el mismo.

## 2. Ejecución

“O.S Security” es una aplicación Web diseñada para ejecutarse en entornos Linux, por ello los requerimientos necesarios para su correcto funcionamiento son los siguientes:

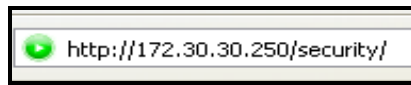
- **Sistema Operativo:** Linux CentOS 5
- **Servidor Web:** Apache 2.0
- **Entorno de Desarrollo:** PHP 5.0
- **Navegador Web:** Mozilla Firefox

Para usar “O.S Security” siga los siguientes pasos:

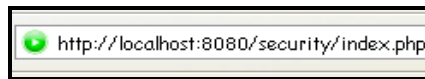
- Típear en el browser la ruta donde se encuentra “O.S Security” para de esta manera cargar el contenido de la aplicación.

Por ejemplo:

**Acceso remoto:** <http://172.30.30.250/security/index.php>



**Acceso local:** <http://localhost:8080/security/index.php>



- Una vez realizado el paso anterior se cargará la pantalla inicial de O.S Security

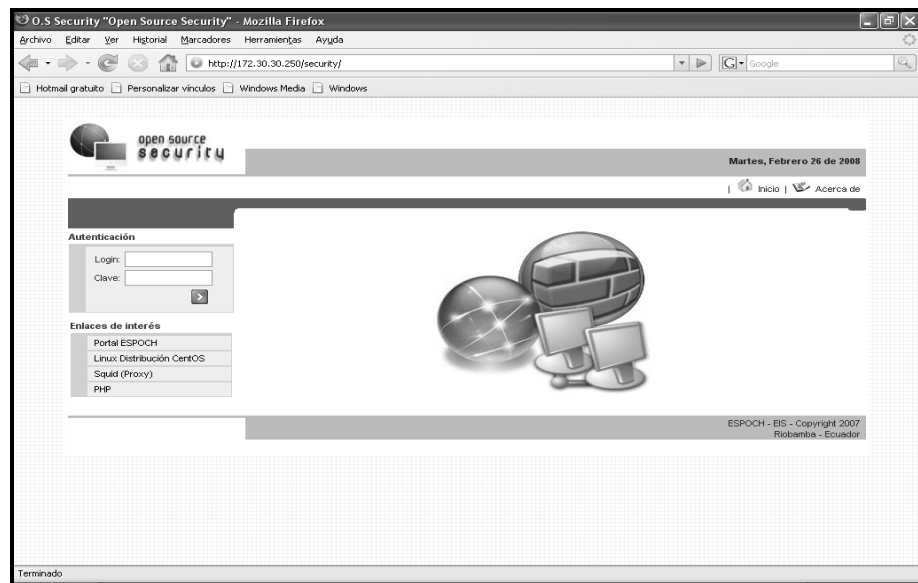


Fig. 1 – Pantalla Inicial

### 3. Modo de Uso

En esta sección usted se familiarizará con la interfaz gráfica, así como también aprenderá a manejar los diferentes módulos que conforman “O.S Security”, los mismos que mencionaremos a continuación:

- Administrar IPTABLES
- Administrar IP6TABLES
- Administrar SQUID
- Información de Equipo y Puertos
- Conectividad
- Configurar Tareas (CRON)
- Administrar Blacklist

Los módulos descritos anteriormente se visualizarán una vez que el administrador valide sus datos en el sistema.

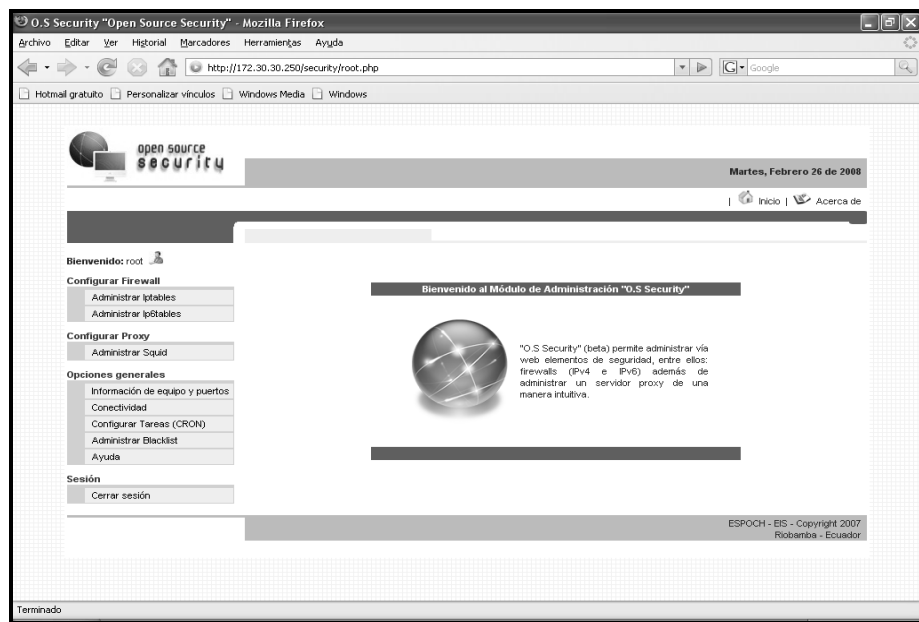


Fig. 2 – Menú Principal

A continuación procederemos a describir el esquema de cada uno de los módulos de “O.S Security”:

- **Módulo 1: Administrar IPTABLES**

La finalidad de este módulo es la de administrar mediante una interfaz gráfica el firewall (cortafuegos) que incluye “O.S Security” para el protocolo IPv4, para de esta manera lograr un filtrado de paquetes que permitan un mejor desempeño de la red que “O.S Security” administre.

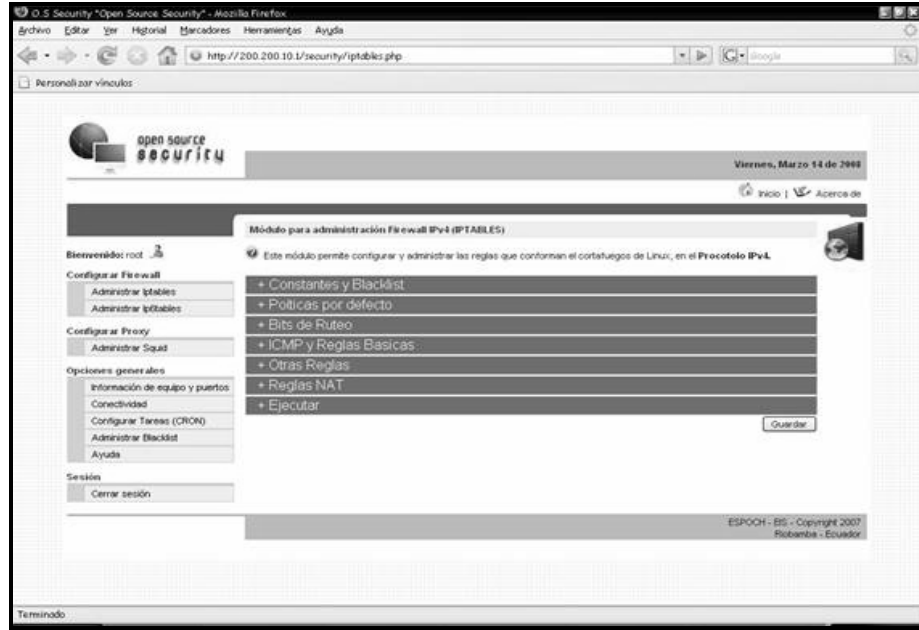


Fig. 3 – Opciones Configuración Firewall IPv4

Las opciones que incluye este módulo son: Constantes y Blacklist, Políticas por Defecto, Bits de ruteo, ICMP y Reglas básicas, Otras reglas, Reglas NAT y Ejecutar. A continuación explicaremos cada opción:

**Constantes y Blacklist**

Muestra las interfaces de red activas en el servidor, permitiendo la posibilidad de dar un nombre descriptivo (Constante) a cada interfaz. Además permite activar el módulo de direcciones prohibidas por el administrador (blacklist).

Blacklist son un listado de direcciones IP consideradas como no permitidas debido a la cantidad de spam que envían por Internet congestionando la red. Para evitar acceder hacia dichas IPs este módulo permite bloquearlas.

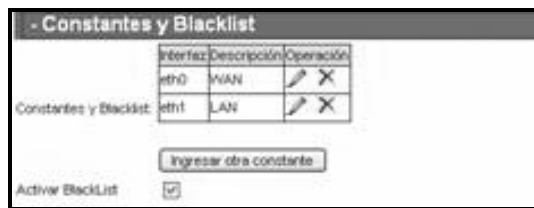


Fig. 4 – Opciones para constantes

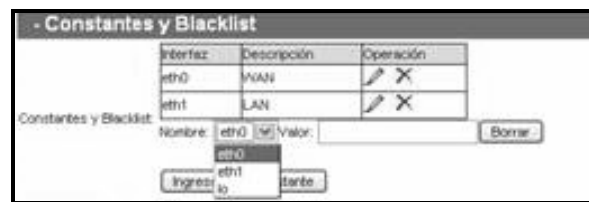


Fig. 5 – Ingreso de nueva constante



Fig. 6 – Edición de una constante específica

**Políticas por Defecto**

Permite establecer los parámetros por defecto que la estructura del Firewall tendrá para realizar el filtrado de paquetes.

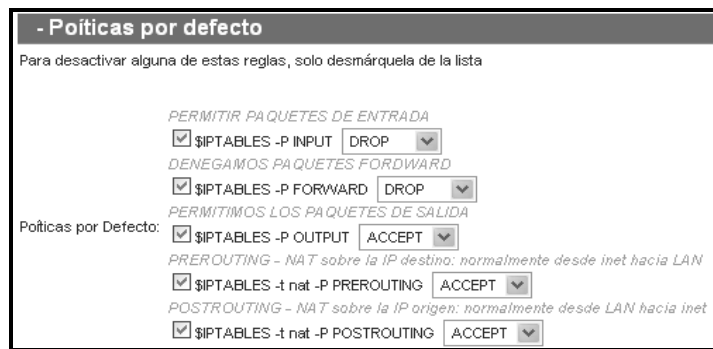


Fig. 7 – Políticas por defecto

**Bits de ruteo**

Habilita determinados bits que permiten reforzar la seguridad en el firewall, por ejemplo: habilitar forward, evitar que un equipo cliente haga ping al servidor, habilitar el uso de cookies al protocolo tcp, entre otras

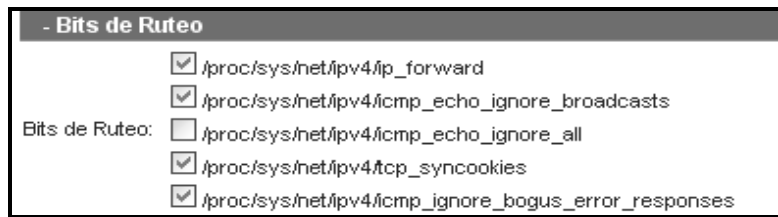


Fig. 8 – Parámetros Bits de Ruteo

**Reglas ICMP y Reglas Básicas**

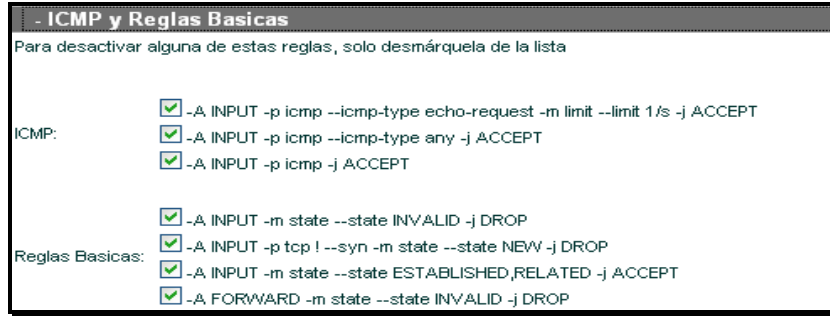
Esta sección tiene 2 subcategorías

**Reglas ICMP**

Permite o no la utilización del protocolo ICMP al momento de realizar pings

**Reglas Básicas**

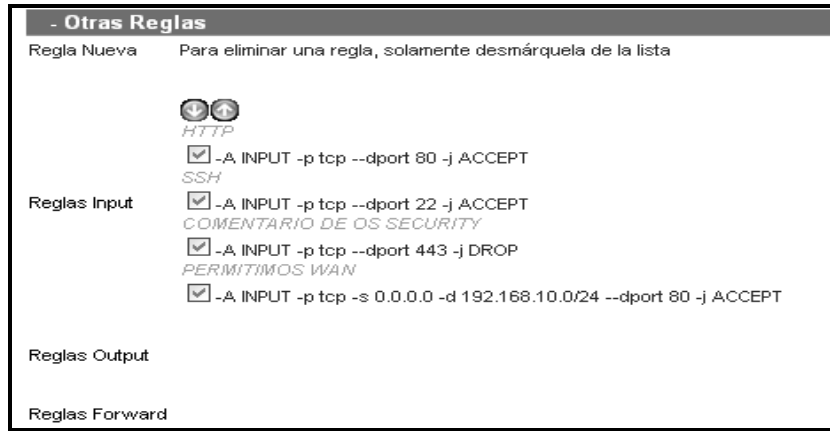
Establece los permisos respectivos al momento de la creación y adición de una regla respectiva al firewall.



**Fig. 9 – Parámetros ICMP y Reglas Básicas**

**Otras Reglas**

Contiene el listado de las reglas registradas en el firewall IPTABLES, como son: INPUT, OUTPUT, FORWARD. Además permite la posibilidad de agregar una nueva regla al hacer clic en el enlace **Nueva Regla**, desplegando un formulario detallado con los parámetros necesarios para crear una regla adecuada para filtrado de paquetes.



**Fig. 10 – Listado de reglas**

DATOS PARA LA CREACIÓN DE UNA REGLA NUEVA

Interfaz	Descripción	Comentario de la Regla:	
eth0	WAN	<input type="text"/>	
eth1	LAN	<input type="text"/>	

Input/Output	Protocolo	Opciones de Regla		Accept/Drop	Log
INPUT	tcp	Elija el origen	Elija el destino	ACCEPT	<input checked="" type="checkbox"/>
		<input type="radio"/> Interfaces: <input type="radio"/> IP:	<input type="radio"/> Interfaces: <input type="radio"/> IP:		
		<input type="radio"/> Colocar banderas <input checked="" type="radio"/> No colocar banderas	<input type="radio"/> Permitir <input type="radio"/> Denegar ()		
		<input type="checkbox"/> Puerto:	<input checked="" type="checkbox"/> Puerto: <input type="text"/> <input type="radio"/> Permitir <input type="radio"/> Denegar ()		

Ingrese los siguientes datos:  
Prefijo:   
(29 caracteres)

**Fig. 11 – Formulario para crear nueva regla**

☑ **Reglas NAT**

Esta sección muestra un listado de las reglas NAT registradas, este tipo de reglas permite realizar redirecciones de tráfico dependiendo del tipo de paquete que se filtre. Para agregar una regla NAT, se debe hacer clic en el enlace **Regla Nueva** para que cargue un formulario detallado con los parámetros necesarios para crear una regla NAT.



Fig. 12 – Listado Reglas NAT

Interfaz	Descripción
eth0	WAN
eth1	LAN

Comentario de la Regla:

Pre/Post Routing	Protocolo	Opciones de Regla		Redirect/DIAT	Log
PREROUTING	tcp	Elija el origen	Elija el destino	REDIRECT	<input checked="" type="checkbox"/>
		<input type="radio"/> Interfaces:	<input type="radio"/> Interfaces:		
		<input type="radio"/> IP:	<input type="radio"/> IP:		
		<input checked="" type="radio"/> Puerto:	<input type="text"/>	Ingrese el puerto: (numero de puerto)	
		<input type="radio"/> Denegar (!)	<input type="text"/>	Ingrese los siguientes datos:	
		<input type="checkbox"/> Puerto:	<input type="text"/>	Nivel de Prioridad: <input type="text" value="crit"/>	
			<input type="text"/>	Prefijo: <input type="text"/>	(29 caracteres)

Fig. 13 – Formulario para crear reglas NAT

☑ **Ejecutar**

Guarda los cambios realizados y pone en ejecución el firewall en el equipo donde se encuentra "O.S Security"



Fig. 14 – Ejecutar Firewall

▪ **Módulo 2: Administrar IP6TABLES**

La finalidad de este módulo es la de administrar mediante una interfaz gráfica el firewall IP6TABLES que incluye "O.S Security" para el protocolo IPv6, para de esta manera lograr un filtrado de paquetes que permitan un mejor desempeño de la red.



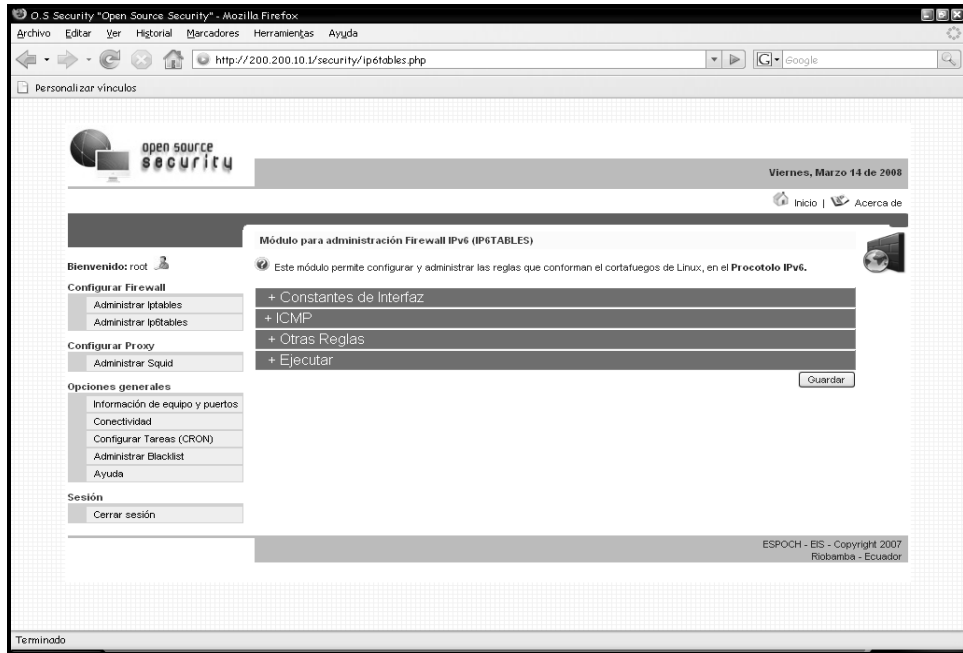


Fig. 15 – Opciones configuración Firewall IPv6

Las opciones que incluye este módulo son: Constantes de interfaz, ICMP, Otras reglas y Ejecutar. A continuación explicaremos cada opción:

**Constantes de Interfaz**

Muestra las interfaces de red activas en el servidor, permitiendo la posibilidad de dar un nombre descriptivo (Constante) a cada interfaz.



Fig.16 – Parámetros de interfaz

**ICMP**

Habilita la conectividad del Protocolo ICMPv6, al momento de realizar ping hacia direcciones IPv6.



Fig. 17 – Parámetro activación ICMPv6

**Otras reglas**

Contiene el listado de las reglas registradas en el firewall IP6TABLES, como son: INPUT, OUTPUT, FORWARD. Además permite la posibilidad de agregar una nueva regla al hacer clic en el enlace **Nueva Regla**, desplegando un formulario

detallado con los parámetros necesarios para crear una regla adecuada para filtrado de paquetes.

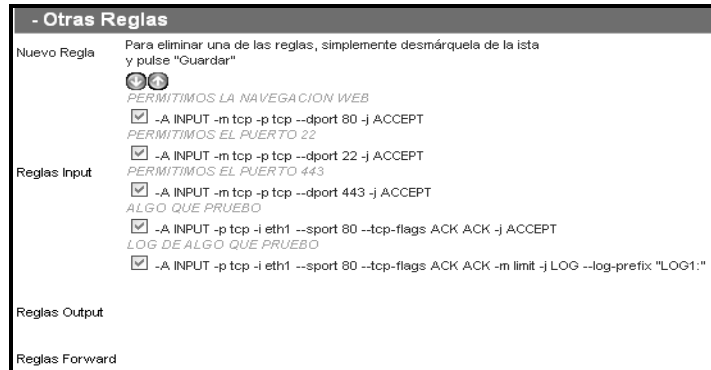


Fig. 18 – Listado de reglas registradas en el firewall IP6TABLES

DATOS PARA LA CREACIÓN DE UNA REGLA NUEVA										
<table border="1"> <thead> <tr> <th>Interfaz</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>eth0</td> <td>WAN</td> </tr> <tr> <td>eth1</td> <td>LAN</td> </tr> </tbody> </table>		Interfaz	Descripción	eth0	WAN	eth1	LAN	Comentario de la Regla:		
Interfaz	Descripción									
eth0	WAN									
eth1	LAN									
Input/Output	Protocolo	Opciones de Regla		Accept/Drop						
INPUT	tcp	Elija el origen	Elija el destino	ACCEPT						
<input type="radio"/> Colocar banderas <input checked="" type="radio"/> No colocar banderas		<b>Interfaz/Constante Origen:</b> WAN <input checked="" type="radio"/> Permitir <input type="radio"/> Denegar (!)	<b>Interfaz/Constante Destino:</b> WAN <input checked="" type="radio"/> Permitir <input type="radio"/> Denegar (!)							
		<input type="radio"/> IP: <input type="text"/>	<input type="radio"/> IP: <input type="text"/>							
		<input type="checkbox"/> Puerto: <input type="text"/>	<input checked="" type="checkbox"/> Puerto: <input type="text"/> <b>Puerto Destino:</b> <input type="text"/> <input checked="" type="radio"/> Permitir <input type="radio"/> Denegar (!)							
<input type="button" value="Guardar"/>										

Fig. 19 – Formulario para crear reglas IP6TABLES

**Ejecutar**

Guarda los cambios realizados en las reglas y pone en ejecución las reglas del firewall para IP6TABLES.

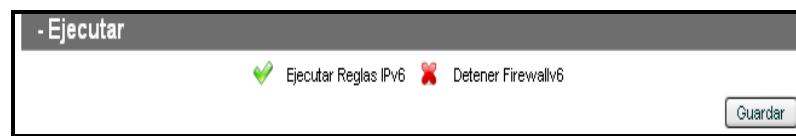
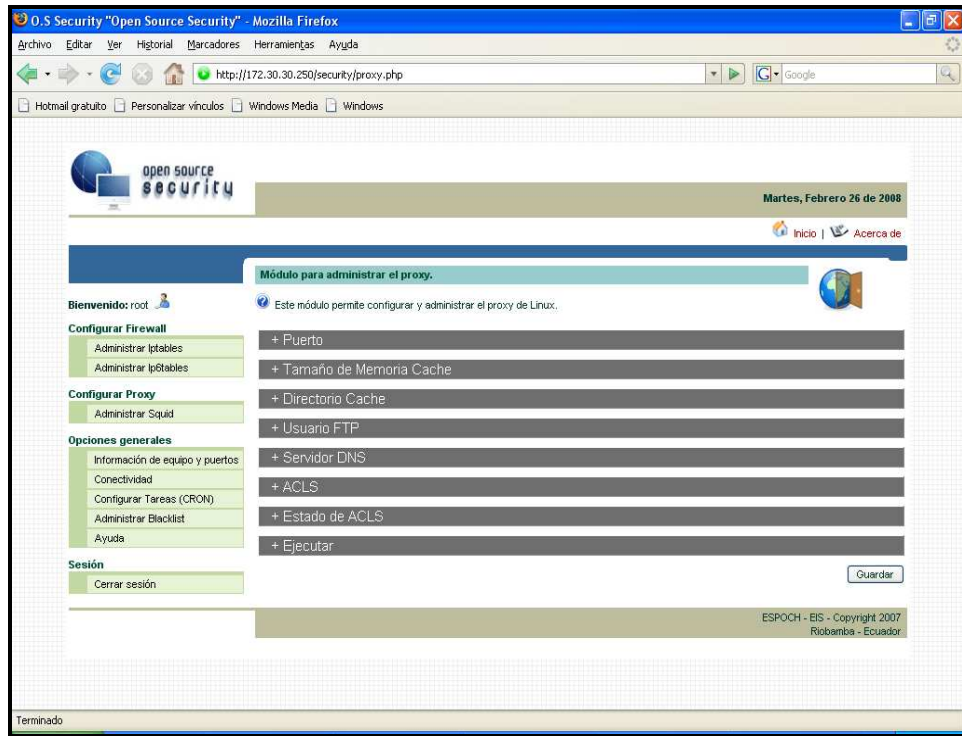


Fig. 20 – Ejecución de reglas del firewall IP6TABLES

- **Módulo 3: Administrar SQUID**

La finalidad de este módulo es la de administrar mediante una interfaz gráfica el servidor proxy, para de esta permitir el acceso de los equipos clientes de una intranet hacia Internet.



**Fig. 21 – Opciones Configuración Proxy**

Los parámetros de configuración que incluye este módulo son: Puerto, Tamaño de memoria Caché, Directorio Caché, Usuario FTP, Servidor DNS, ACL's, Estado de ACL's y Ejecución.

A continuación daremos una descripción de las opciones que incluye este módulo:

- **Puerto**

Esta sección permite registrar los puertos que el squid escucha al momento que el firewall IPTABLES redirecciona los paquetes al squid, cuando los clientes acceden a Internet.



**Fig. 22 – Listado de Puertos registrados**

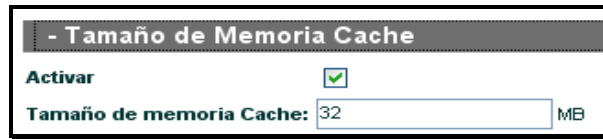
Para agregar un nuevo puerto hay que hacer un clic en el enlace **Añadir Puerto** para que se visualice un formulario solicitando el puerto a registrar.



**Fig. 23 – Formulario para agregar un nuevo puerto**

**Tamaño de memoria caché**

Permite separar un espacio de memoria específico para almacenar las páginas Web.



- Tamaño de Memoria Cache

Activar

Tamaño de memoria Cache: 32 MB

Fig. 24 – Formulario para configurar el tamaño de memoria caché

**Directorio caché**

Permite separar un espacio en disco del servidor para almacenar las páginas Web que los clientes acceden desde Internet.



- Directorio Cache

El actual es:

Tipo	Directorio	Tamaño Memoria	N° Objetos	N° Clientes
ufs	/var/spool/squid	1000	16	256

Activar

Puede modificar la configuración:

Tipo	Directorio	Tamaño Memoria	N° Objetos	N° Clientes
ufs	/var/spool/squid	1000	16	256

Fig. 25 – Formulario para configurar el Directorio Caché

**Usuario FTP**

Permite especificar el e-mail del usuario encargado de la administración del servidor Proxy, cuando ocurra un inconveniente en el mismo el sistema enviará un mensaje de alerta a éste e-mail.



- Usuario FTP

Nombre de usuario: mmalan@epoch.edu.ec

Fig. 26 – Registro de usuario FTP

**Servidor DNS**

Hace referencia al o los servidores DNS que el Proxy necesita para salir a Internet.



- Servidor DNS

Para poder ingresar un nuevo servidor DNS, haga click [aquí](#)

Servidores DNS: 172.30.60.5  
192.6.1.1

Fig. 27 – Registro de Servidor DNS

**ACL**

Permite registrar las reglas encargadas de administrar correctamente el contenido que los clientes de la intranet obtienen desde Internet. Permitiendo de esta manera un mejor desempeño.

- ACLS		
Nuevo		
Nro.	ACL / Comentario	Estado
<i>REGLA DE MARCO</i>		
1:	<input checked="" type="checkbox"/> acl lan src 172.30.40.0/255.255.255.224	deny
<i>CON PROHIBICIONES DE EXTENSIONES</i>		
2:	<input checked="" type="checkbox"/> acl extensiones url_regex -i \.ppppppppp\$	deny
<i>USAMOS UNA FUENTE</i>		
3:	<input checked="" type="checkbox"/> acl fuente src "/var/www/html"	allow

Fig. 28 – Reglas ACL registradas

Para registrar una nueva ACL basta con hacer clic en el enlace **Nuevo**

Fig. 29 – Formulario para registro de nueva ACL

**Estado de ACLs**

Permite asignar los respectivos permisos de acceso y/o deniego a las reglas ACL registradas en la opción anterior.

Fig. 30 – Asignación de permisos de acceso

**Ejecutar**

Una vez realizado los cambios respectivos en el fichero, se le permite poner en ejecución al servicio squid.

Fig. 31 – Opciones servicio Squid

- **Módulo 4:** Información del Equipo y Puertos

La finalidad de este módulo es la de obtener información específica del equipo mediante una interfaz gráfica, para de esta manera determinar los recursos que el equipo utiliza.

Este módulo está conformado por 2 opciones: **Datos del Equipo y Escaneo de Puertos**, cada opción presentará ambientes distintos de configuración para el administrador.

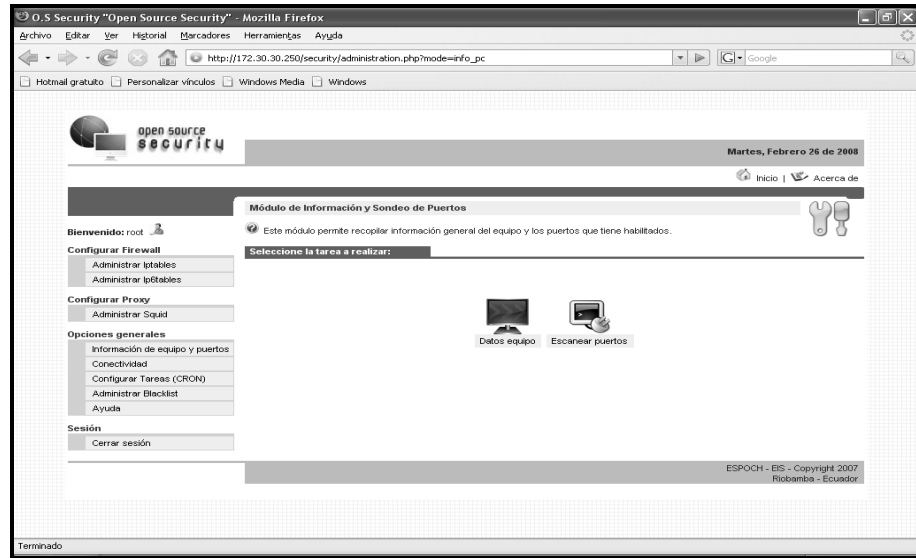


Fig. 32 – Opciones de Información del Equipo y Puertos

A continuación daremos una descripción de las opciones que contiene éste módulo:

### DATOS DEL EQUIPO

Recopila información general del entorno del servidor donde se encuentra instalado “O.S Security”.

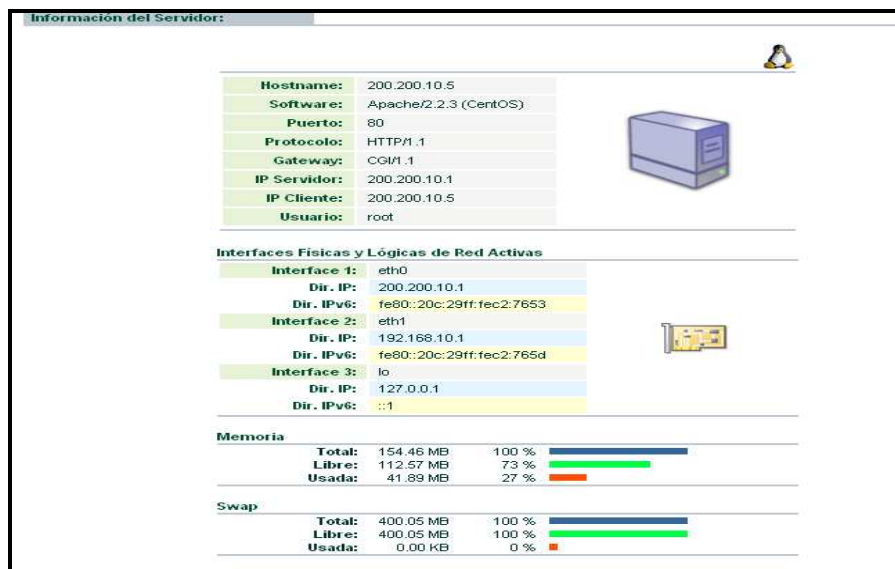


Fig. 33 – Información del Servidor

## ESCANEO DE PUERTOS

Permite determinar los puertos que los servicios de Linux utilizan durante su ejecución, hay 3 tipos de escaneo:



Fig. 34 – Opciones para Escaneo de puertos

### Servidor

Muestra los puertos utilizados por los distintos servicios de Linux, que se encuentran en el equipo



Fig. 35 – Resultados obtenidos del escaneo de puertos del servidor

### Host Remoto

Permite realizar un escaneo de un rango de puertos de un equipo específico

Una captura de pantalla de una herramienta de escaneo de puertos que muestra un listado de puertos escaneados en un host remoto. El título es "Resultados obtenidos:". El listado muestra los puertos del 21 al 40, todos deshabilitados, excepto el puerto 22, que está habilitado.

Puerto	Estado	IP
21	deshabilitado	172.30.30.250
22	habilitado	172.30.30.250
23	deshabilitado	172.30.30.250
24	deshabilitado	172.30.30.250
25	deshabilitado	172.30.30.250
26	deshabilitado	172.30.30.250
27	deshabilitado	172.30.30.250
28	deshabilitado	172.30.30.250
29	deshabilitado	172.30.30.250
30	deshabilitado	172.30.30.250
31	deshabilitado	172.30.30.250
32	deshabilitado	172.30.30.250
33	deshabilitado	172.30.30.250
34	deshabilitado	172.30.30.250
35	deshabilitado	172.30.30.250
36	deshabilitado	172.30.30.250
37	deshabilitado	172.30.30.250
38	deshabilitado	172.30.30.250
39	deshabilitado	172.30.30.250
40	deshabilitado	172.30.30.250

Fig. 36 – Listado de rango de puertos escaneados

**Por Puerto**

A diferencia de la opción anterior solo ejecuta el escaneo de un puerto específico.



Fig. 37 – Resultado obtenido a un puerto específico [1]

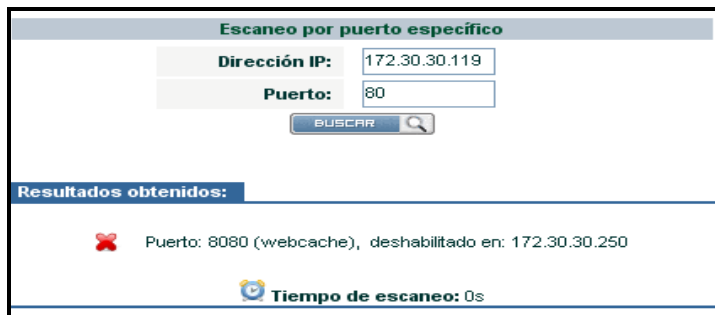


Fig. 38 – Resultado obtenido a un puerto específico [2]

▪ **Módulo 5: Conectividad**

La finalidad de este módulo es la de obtener información referente a los equipos que se encuentran conectados en la intranet.

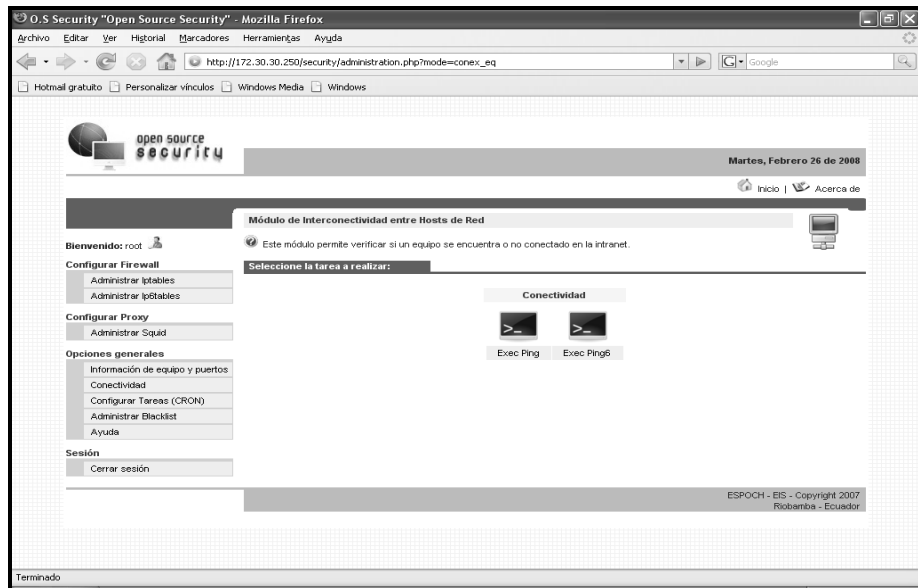


Fig. 39 – Opciones Conectividad



Este módulo está conformado por 2 opciones:

**Exec Ping**

Permite verificar la existencia de conectividad a un host específico en el protocolo IPv4, para ello se debe ingresar la dirección IPv4 a consultar.

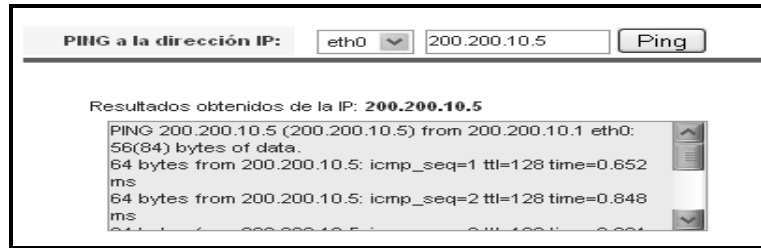


Fig. 40 – Formulario conectividad IPv4

**Exec Ping6**

Permite verificar la existencia de conectividad a un host específico en el protocolo IPv6, para ello se debe ingresar la dirección IPv6 a consultar.



Fig. 41 – Formulario conectividad IPv6

**Nota:** Verificar que el host de destino tenga habilitado el protocolo IPv6, además del tipo de dirección IPv6 a consultar por ejemplo si se trata de una dirección local o global, ya que ambas no son iguales.

▪ **Módulo 6:** Configurar Tareas (CRON)

Éste módulo tiene la finalidad de registrar y administrar tareas específicas para que el sistema operativo las ejecute automáticamente sin necesidad de la intervención del administrador.

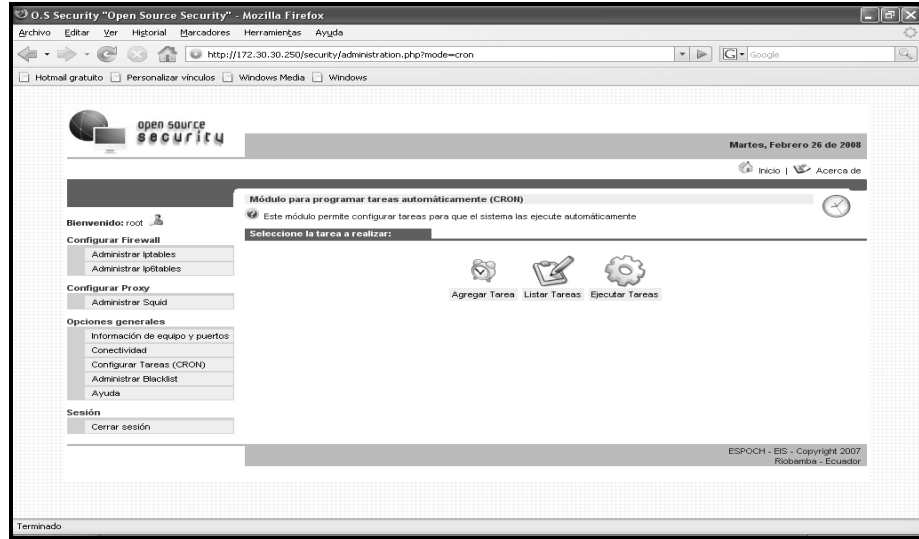


Fig. 42 – Opciones Configurar Tareas (CRON)

Está conformado por 3 opciones de administración:

**Agregar Tarea**

Incorpora 2 funcionalidades:

➤ **Tareas preprogramadas para Blacklist**

Permiten configurar la ejecución automática de scripts bash relacionados a la descarga de blacklist de un repositorio de Internet, además de actualizar la información de dicha descarga con la bitácora de blacklist que incorpora el Firewall IPv4 de “O.S Security”.

Mediante la configuración de determinados parámetros que logren cumplir con ésta finalidad.

Fig. 43 – Formulario para registrar Tarea Preprogramada

➤ **Configurar Tarea Especifica**

Permite agregar una tarea específica al sistema mediante la configuración de determinados parámetros que logren cumplir con ésta finalidad. Por

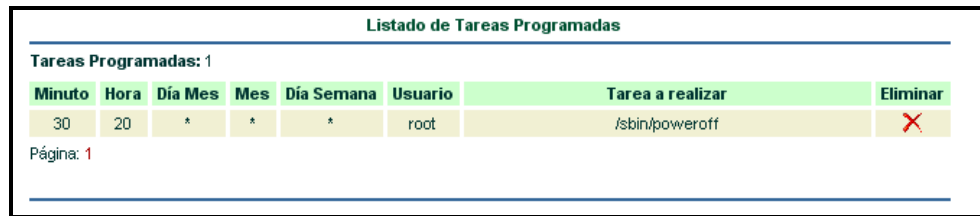
ejemplo: Si deseamos que el computador se apague automáticamente debemos especificar la hora y el minuto de ejecución



Fig. 44 – Formulario para registro de nueva tarea al sistema

**Listar Tareas**

Muestra un listado de las tareas registradas en el archivo /etc/crontab, además se le permite la posibilidad de eliminar una tarea específica que el administrador crea conveniente no sea ejecutada automáticamente por el sistema operativo.



Listado de Tareas Programadas							
Tareas Programadas: 1							
Minuto	Hora	Día Mes	Mes	Día Semana	Usuario	Tarea a realizar	Eliminar
30	20	*	*	*	root	/sbin/poweroff	X

Página: 1

Fig. 45 – Listado de tareas registradas

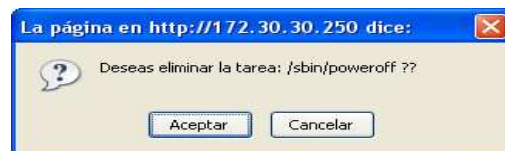


Fig. 46 – Mensaje de confirmación para eliminar una tarea específica

**Ejecutar Tareas**

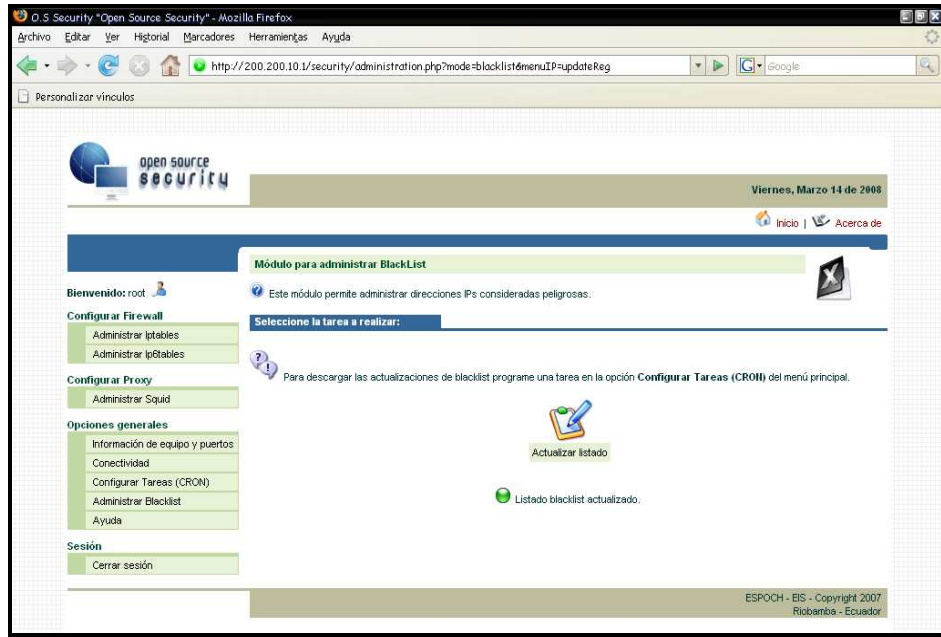
Permite efectuar los cambios realizados en el fichero /etc/crontab. Para que el servicio crond lo ejecute automáticamente.



Fig. 47 – Ejecución de tareas

- **Módulo 7: Administrar Blacklist**

Éste módulo incorpora la funcionalidad descargar actualizaciones de blacklist de un repositorio de Internet con la finalidad de actualizar dicho contenido con la bitácora de blacklist que “O.S Security” incorpora para el Firewall IPTABLES para filtrar paquetes que provengan de direcciones inapropiadas, debido a su contenido o a funcionalidades que bajen el rendimiento de la red.



**Fig. 63 – Opciones Blacklist**

#### 4. Nuevas Versiones

Debido a las expectativas generadas con “O.S Security” y el surgimiento de nuevos requerimientos se ve la posibilidad de anexar nuevos módulos que permitan un mejor desempeño del mismo.

## BIBLIOGRAFÍA

- KENDALL, R. Análisis y Diseño de Sistemas. México – México: Prentice Hall, 1995. pp. 910.
- MORENO Patricio. Redes de Computadores II. Riobamba – Ecuador, 2004. pp.105.
- MORENO Patricio. Integración de Sistemas. Riobamba – Ecuador, 2005. pp.114.
- PRESSMAN, R. Ingeniería de Software: un enfoque práctico. Madrid – España: Mc Graw Hill, 2002. pp. 620.

### **Bibliografía de Internet**

#### **Blacklist**

1. <http://www.checkipblacklists.com/Blacklist-Directory.php>  
(2008-01-08)
2. <http://www.xaraya.com/documentation/phpxref/nav.html?modules/mailbag/xaradminapi/mailbag.php.source.html>  
(2008-01-10)
3. <http://www.frozenminds.com/spamblacklist.html>  
(2008-01-23)

#### **Iptables**

4. <http://www.linuxguruz.com/iptables/>  
(2007-10-16)
5. <http://www.no-net.org/ip6wall/download/>  
(2007-10-16)

6. <http://www.netfilter.org/documentation/index.html>  
(2007-10-04)
7. <http://www.zdnet.co.uk/tsearch/Firewall+iptables.htm>  
(2007-10-11)
8. <http://www.securitydocs.com/Firewall/2>  
(2007-10-15)
9. <http://securitydocs.bitpipe.com/rlist/term/Firewall-Appliances.html>  
(2007-10-15)
10. [http://www.sans.org/reading\\_room/whitepapers/firewalls/807.php](http://www.sans.org/reading_room/whitepapers/firewalls/807.php)  
(2007-10-17)

### **Metodología DSDM**

11. [http://www.ifi.uzh.ch/rrerg/fileadmin/downloads/teaching/seminars/seminar\\_ws0304/14\\_Voigt\\_DSMD\\_Ausarbeitung.pdf](http://www.ifi.uzh.ch/rrerg/fileadmin/downloads/teaching/seminars/seminar_ws0304/14_Voigt_DSMD_Ausarbeitung.pdf)  
(2007-08-22)
12. [http://en.wikipedia.org/wiki/Dynamic\\_Systems\\_Development\\_Method](http://en.wikipedia.org/wiki/Dynamic_Systems_Development_Method)  
(2007-09-12)

### **Pila Dual**

13. [http://www.viagenie.qc.ca/en/ipv6/presentations/IPv6-transition-mechanisms\\_v1.pdf](http://www.viagenie.qc.ca/en/ipv6/presentations/IPv6-transition-mechanisms_v1.pdf)  
(2007-08-15)
14. [http://www.it.uc3m.es/~fvalera/int\\_red/trabajos/IRSBv1.pdf](http://www.it.uc3m.es/~fvalera/int_red/trabajos/IRSBv1.pdf)  
(2007-08-15)
15. [http://www.cu.ipv6f.org/pdf/carlos\\_ralli\\_transitiontutorial.pdf](http://www.cu.ipv6f.org/pdf/carlos_ralli_transitiontutorial.pdf)  
(2007-05-16)

### **Proxy (Squid)**

16. <http://www.squid-cache.org>  
(2007-10-15)
17. [http://www.deckle.co.za/squid-users-guide/Browser\\_Configuration](http://www.deckle.co.za/squid-users-guide/Browser_Configuration)  
(2007-10-17)
18. <http://www.linuxparatodos.org>

(2007-09-19)

19. <http://www.ecualug.org>

(2007-09-26)

### **Técnicas de Convivencia**

20. <http://www.uv.es/femenia/TransicionIPv6enUV.ppt>

(2007-10-03)

21. <http://www.ipv6.uv.es/TransicionIPv6enUV.ppt>

(2007-08-15)

### **Túneles**

22. <http://ipv6.fedora-es.com/?q=node/5>

(2007-08-14)

23. <http://web.frm.utn.edu.ar/codarec/Cursos/IPv6%20Fundamento%20e%20Implementacion.pdf>

(2007-08-20)

24. <http://www.inet6.dk/thesis.pdf>

(2007-08-21)

25. <http://www.linux.com/base/ldp/howto/Linux+IPv6-HOWTO/chapter-configuring-ipv6-in-ipv4-tunnels.html>

(2007-09-04)

26. [http://www.nic.ve/view/docs/CNTI\\_IPv6.pdf](http://www.nic.ve/view/docs/CNTI_IPv6.pdf)

(2007-09-10)

### **Traducción de Direcciones**

27. <http://oasis.dit.upm.es/~omar/tesis/tesis1/taro/26-Migracion-IPv6-Omar-Wallid.pdf>

(2007-09-10)

28. <http://www.rediris.es/red/ponenciasValencia/PortabilidadApli.pdf>

(2007-09-12)

29. <http://internetng.dit.upm.es/ponencias-jing/2002/fernandez/Evolucion-IPv4-IPv6-David-Fernandez.PDF>

(2007-09-12)