



**UNIVERSIDAD DE GUAYAQUIL**  
**Facultad de Ciencias Matemáticas y Físicas**  
**Carrera de Ingeniería en Sistemas Computacionales**

**“IMPLEMENTACION DE UN SERVIDOR PROXY  
CON IPV6 EN UNA RED DE AREA LOCAL”**

**TOMO I**

**PROYECTO DE GRADO**

Previo la Obtención del Título de:

**INGENIERO EN SISTEMAS  
COMPUTACIONALES**

Autores:

**Andrés David Revelo Rodríguez**  
**José Orlando Fernandez Mora**  
**Javier Rolando Velasco Ruiz**

**GUAYAQUIL – ECUADOR**  
**Año: 2010**

## **DEDICATORIA**

Por sobre todas las cosas a Dios, por las bendiciones derramadas sobre mí. Mostrándome su grandeza, entendiendo que sin su ayuda nada de esto hubiese sido posible y sabiendo que solo soy un instrumento todo para la gloria de ti, padre Todopoderoso.

Andrés Revelo R.

A mi Madre amada, Elida Albertina Ruiz P. quien me forjó y enseñó lo necesario para ser un buen hombre y profesional en todos los aspectos, por su esfuerzo y apoyo incondicional en todo momento.

A mi Hermano Sammy Fernando, por alentarme desde arriba, porque sus sueños no se fueron, sino que me acompañaron para hacerlos realidad.

Javier R. Velasco Ruíz

Dedico principalmente a Dios, mi familia mi madre mi padre y mis hermanos a mi esposa y mi hijas y todos mis amigos y todas las personas que me apoyaron y me dieron fuerza moral.

A mi madre Nora Magdalena Mora que con su esfuerzo soy quien soy ahora, una persona que pudo alcanzar esta meta importante en mi vida.

A mis hijas Liah y Anahi quienes son la razón de mi vida, las que en los momentos más difíciles y agotadores me dieron ese impulso para seguir adelante.

José Fernández M.

## **AGRADECIMIENTO**

Después de Dios, a mis padres por su consejo y guía, desde pequeño supieron hacer de mí un hombre de bien. A mi madre Beatriz Rodríguez, por su respaldo, por ser mi amiga, mi todo, la vida entera no bastaría para darte gracias. A mi familia, mis hijas: Sarita, Daniela y Natalia, fuentes de mi inspiración. A mi esposa: Omayra O. por estar conmigo en las buenas y en las malas.

A mis familiares: tías Silvia, Amada y Elena, por su consejo y ayuda. Mis tíos y primos.

A mi grupo de tesis: José Fernández y Javier Velasco por su gran aporte también, lo logramos “pequeños saltamontes”.

A todos GRACIAS:

Andrés Revelo R.

A Dios y la Virgen María por mantenerme en pie.

A mi Familia, principalmente a mis Tíos y Abuelos: Alejandrina y Francisco, por los cuidados y apoyo recibidos desde mis primeros pasos hasta hoy.

A mis Amigos del Cole por siempre estar presentes en cualquier circunstancia.

A nuestro excelente equipo: Andrés y José, sin duda por todo el respeto y trabajo consciente efectuado para hacer realidad esta obra importante.

Y por todo el respaldo y confianza, al alentarme a continuar a paso firme, a mi Amor Diana Arana.

Javier R. Velasco Ruíz

Especialmente a Dios por que sin el nada sería posible. A mis maestros en la escuela, colegio y universidad por darme el conocimiento necesario para poder lograr esta meta cumplida y empezar una vida profesional, a mis padres por darme la formación como persona de bien.

A mis compañeros de grupo Javier Velasco, y Andrés Rebelo, a todos los compañeros de seminario y todos mis amigos entre ellos Jorge Luis Aray por su amistad.

*“Nunca debes renunciar a tus deseos, la suerte no existe, ni llega por si sola, lucha por tus ideales, y vence la adversidad”*

A todos GRACIAS:

José Fernández M.

## TRIBUNAL DE GRADUACIÓN

---

**PRESIDENTE DEL TRIBUNAL**

---

**PRIMER VOCAL**

---

**SEGUNDO VOCAL**

---

**SECRETARIO**

## DECLARACION EXPRESA

“La autoría de la tesis de grado corresponde exclusivamente a los suscritos, Perteneciendo a la Universidad de Guayaquil los derechos que generen la Aplicación de la misma”

Según el Reglamento de Graduación de la Carrera de Ingeniería en Sistemas Computacionales, Art. 26.

---

Andrés David Revelo Rodríguez

---

José Orlando Fernández Mora

---

Javier Rolando Velasco Ruiz



## **RESUMEN**

En cualquier medio en donde las Organizaciones intervengan, estarán ávidas de contar con herramientas que faciliten y mejoren procesos para conquistar sus metas. Siendo la tecnología y buenas prácticas de trabajo, factores inherentes al éxito, es necesario contar con una herramienta que combine dichos factores.

Servidor Proxy en IPv6 opera en una red local, bajo el protocolo de IP's versión 6, implementado sobre plataforma Linux, donde se incorporó los servicios y prestaciones de seguridad deseados para mantener a los activos de información. Este servidor cuenta con una aplicación desarrollada en Java y con BD MySQL, para la adecuada y correcta administración de accesos y restricciones que deben ser aplicados a todos los clientes internos de la organización.

**La aplicación Proxy Ipv6,** es una herramienta que cuenta con una interfaz fácil de usar, ejecuta políticas de accesos según los niveles jerárquicos de usuarios o grupos de navegación.

El Aplicativo brinda una solución optimizada para y de fácil entendimiento a la problemática planteada.

## **INDICE GENERAL**

<b><u>INDICE</u></b>	<b><u>PAGINAS</u></b>
DEDICATORIA.....	I-II
AGRADECIMIENTO.....	III-V
TRIBUNAL DE GRADUACIÒN.....	VI
DECLARACIÒN EXPRESA.....	VII
RESUMEN.....	VIII-IX

### **CAPITULO 1**

#### **1. INTRODUCCION**

1.1 Antecedentes.....	1
1.2 Problemática.....	4
1.3 Solución.....	5
1.4 Visión .....	6
1.5 Misión.....	6
1.6 Objetivos Generales.....	6
1.7 Objetivos Específico.....	7
1.8 Alcances.....	8
1.9 Ventajas.....	9
1.9.1 Acceso.....	9
1.9.2 Costo.....	9
1.10 Desventajas.....	10
1.11 Arquitectura.....	11
1.11.1 Arquitectura IPv6.....	11
a) Direcccionamiento.....	11

b) Paquetes.....	12
1.11.2 Arquitectura de tres capas.....	13
a) Capa Presentación.....	13
b) Capa de Negocios .....	14
c) Capa de Datos .....	16
1.12 Modelo de Desarrollo.....	16
1.13 Metodología.....	19
1.13.1 Encapsulación.....	19
1.13.2 Herencia.....	19
1.13.3 Polimorfismo.....	20
1.14 Recursos.....	21
1.14.1 Software.....	21
1.14.1.1 Plataforma.....	21
1.14.1.2 Servicios.....	21
1.14.1.3 Herramienta desarrollo.....	22
1.14.1.4 Base Dato.....	23
1.14.1.5 Herramienta diseño.....	23
1.14.2 Hardware.....	23
1.14.2.1 Computador Personal.....	23
1.14.3 Humano.....	23
1.15 Cronograma de actividades.....	24

## **CAPITULO 2**

### **2. ANÁLISIS**

2.1 Levantamiento de Información.....	25
2.1.1 ¿Por qué realizar el Proyecto?.....	25
2.1.2 ¿Quiénes son los Usuarios finales?.....	26
2.1.3 Formas de Levantar la Información utilizadas.....	26

2.1.3.1 investigación en Internet.....,,,	27
2.1 Caso de Uso.....	27
2.1.1 Diagrama de Caso de Uso.....	28
2.1.2 Detalle de los Caso de Uso.....	29
2.2 Diagrama entidad relación.....	32
2.3 Diagrama de especificación de Procesos.....	33

## **CAPITULO 3**

### **3 Diseño**

3.1 Definición y Objetivos.....	36
3.2 Diseño del Sistema.....	36
3.3 Estructura de la Base de Datos.....	38
3.3.1 Tabla usuarios de la aplicación.....	38
3.3.2 Tabla Permiso usuario de la aplicación.....	38
3.3.3 Tabla Permisos privilegios de usuario de la aplicación.....	39
3.3.4 Tabla creación y consulta de la aplicación.....	39
3.3.5 Tabla Reporte red interna.....	40
3.3.6 Tabla Reporte auditoria.....	40
3.3.7 Tabla creación y consulta usuario de navegación.....	41
3.3.8 Tabla parámetro de configuración.....	41
3.3.9 Tabla ayuda de la aplicación.....	42
3.4 Diseño a nivel de Componente.....	43
3.4.1 Proceso Autenticación.....	43
3.4.2 Proceso usuarios de la aplicación.....	44
3.4.3 Proceso levantamiento base de datos.....	45
3.4.4 Proceso realizar reportes.....	46
3.5 Diseño de Interfaz.....	47

3.5.1 Acceso.....	47
3.5.2 Menú principal.....	48
3.5.3. Pantalla creación de usuario de la aplicación.....	49
3.5.4 Pantalla consulta usuario de la aplicación.....	49
3.5.5 Pantalla privilegios de usuarios de la aplicación.....	50
3.5.6 Pantalla reporte consumo A/B.....	51
3.5.7 Pantalla reporte auditoria.....	52
3.5.8 Pantalla Red interna.....	52
3.5.9 Pantalla usuario de navegación.....	53
3.5.10 Pantalla Regla de navegación.....	54
3.5.11 Pantalla crear regla para el proxy- wizard.....	55
3.6 Flujo de ventanas.....	57

## **CAPITULO 4**

### **4 Desarrollo del Software**

4.1 Antecedentes.....	58
4.2 Procesos Principales.....	59
4.2.1 Clases.....	59
4.2.1.1 Conexión.....	59
4.2.1.2 configuración.....	60
4.2.1.2 Ejecutar procesos.....	60
4.2.2 Formas o Pantallas.....	60
4.2.2.1 Forma index.jsp.....	60
4.2.2.2 Forma principal.jsp.....	61
4.2.2.3 Forma mod_privilegios.jsp.....	61
4.2.2.4 Forma regla_proxy.jsp.....	61
4.2.2.5 Forma funciones_reportes.jsp.....	62
4.2.3 Procedimientos almacenados.....	62

4.2.3.1. Procedimiento mantenimiento_permisos.....	63
4.2.3.2. Procedimiento mantenimiento_usuarios.....	63
4.2.3.3. Procedimiento funciones.....	63
4.2.3.4. Procedimiento vistas.....	63

## **CAPITULO 5**

### **5 Conclusiones y Recomendaciones**

5.1 Conclusiones.....	64
5.2 Recomendaciones.....	65
5.3 Bibliografía.....	66
5.4 Anexo.....	67

## **TABLAS**

Tablas 2.3.1 especificación de procesos.....	33
Tablas 2.3.2 especificación de procesos creación de usuarios.....	33
Tablas 2.3.3 especificación de proceso de creación de privilegios.....	34
Tablas 2.3.4 especificación de proceso generación de reportes.....	34
Tablas 3.3.1 tabla usuario de la aplicación.....	38
Tablas 3.3.2 tabla permiso de la aplicación.....	38
Tablas 3.3.3 tabla permiso privilegios aplicación.....	39
Tablas 3.3.4 tabla usuario de la aplicación.....	39

Tablas 3.3.5 tabla reporte.....	40
Tablas 3.3.6 tabla auditoria.....	40
Tablas 3.3.7 tabla usuario de navegación.....	41
Tablas 3.3.8 tabla parámetros de configuración.....	41
Tablas 3.3.9 tabla ayuda de aplicación .....	42

## FIGURAS

Fig. 1.1 Pagina web squid confirmación de no transparencia.....	4
Fig. 1.2 Diagrama de red implementación Servidor Proxy.....	5
Fig. 1.3 Estructura de un paquete IPV6.....	12
Fig. 1.4 Confirmación de un paquete IPV6.....	13
Fig. 1.5 Diagrama arquitectura tres capas.....	16
Fig. 1.6 Diagrama Modelo espiral.....	18
Fig. 2.1 Diagrama de Caso de Uso Global.....	28
Fig. 2.2 Diagrama caso de uso Login.....	29
Fig. 2.3 Diagrama caso de uso creación de Usuario.....	30
Fig. 2.4 Diagrama caso de uso creación de privilegios.....	30
Fig. 2.5 Diagrama caso de uso Generación de reportes.....	31
Fig. 2.6 Diagrama caso de uso Administración.....	31
Fig. 2.7 Diagrama entidad relación.....	32
Fig. 3.3.1 Tabla Usuario de la aplicación.....	38
Fig. 3.3.2 Tabla Permiso usuario de la aplicación.....	38
Fig. 3.3.3 Tabla Permiso privilegios usuarios de la aplicación.....	39
Fig. 3.3.4 Tabla creación consulta de la aplicación.....	39
Fig. 3.3.5 Tabla reporte red interna.....	40
Fig. 3.3.6 Tabla reporte auditoria.....	40



Fig. 3.3.7 Tabla creación de consulta usuario de navegación.....	41
Fig. 3.3.8 Tabla parámetro de configuración.....	41
Fig. 3.3.9 Tabla ayuda de la aplicación.....	42
Fig. 3.10 Proceso de autenticación.....	43
Fig. 3.11 Proceso usuario de la aplicación.....	44
Fig. 3.12 Proceso levantamiento de base de datos.....	45
Fig. 3.13 Proceso realizar reporte.....	46
Fig. 3.14 Interfaz acceso.....	47
Fig. 3.15 Interfaz Pantalla principal.....	48
Fig. 3.16 Interfaz pantalla creación de usuario aplicación.....	49
Fig. 3.17 Interfaz pantalla consulta de usuario aplicación.....	50
Fig. 3.18 Interfaz pantalla privilegios de usuarios aplicación.....	50
Fig. 3.19 Interfaz pantalla consumo de ancho banda.....	51
Fig. 3.20 Interfaz pantalla reporte auditoria de usuario aplicación.....	52
Fig. 3.21 Interfaz pantalla reporte de red interna.....	53
Fig. 3.22 Interfaz pantalla usuario de navegación.....	54
Fig. 3.23 Interfaz pantalla regla de navegación.....	54
Fig. 3.24 Interfaz pantalla wizard creación de regla.....	56
Fig. 3.25 Diagrama Flujo de ventanas.....	57

# CAPITULO 1

# 1 INTRODUCCIÓN

## 1.1 Antecedentes

Desde los años 80's para los desarrolladores de plataformas de comunicación de datos, ha sido una preocupación constante la expansión exponencial de la World Wide Web (Internet), siendo así, y ya que esta realidad nos ha llevando hacia el agotamiento de las direcciones IP versión 4 (IPv4), es decir, a la progresiva disminución de la cantidad de direcciones IP disponibles.

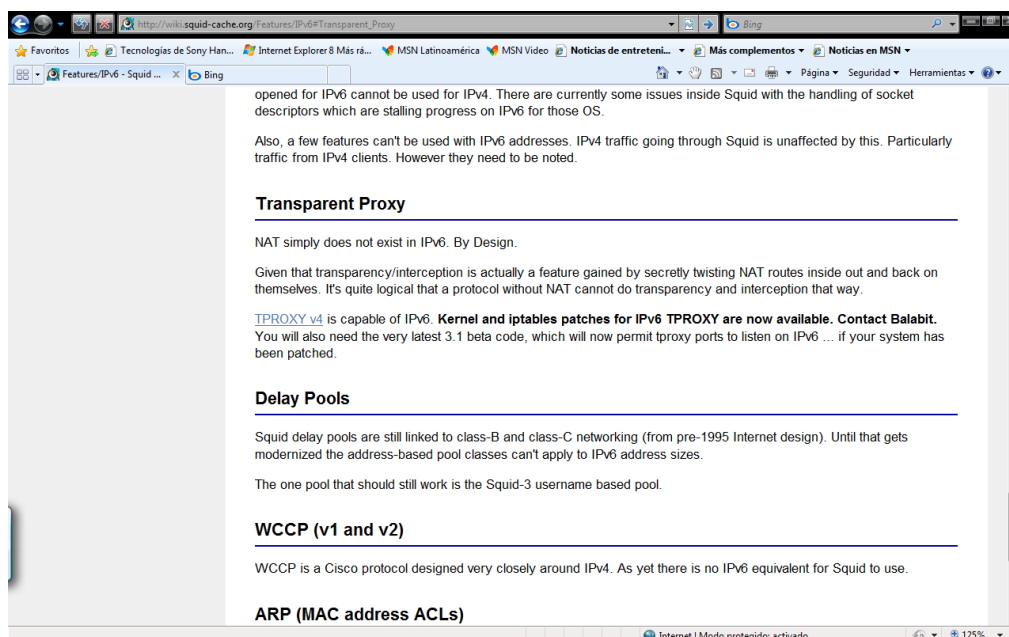
Las direcciones IP pueden ser de diferentes tipos o clases (A, B, C, D y E). Una red de clase A se reserva, en la actualidad para los gobiernos, y en el pasado también para grandes empresas como por ejemplo Hewlett Packard. Las direcciones de clase B se otorgan a medianas empresas. Los usuarios "normales" ó "públicos" usarán direcciones de clase C, se pierde así un rango de direcciones IP importante.

Es un hecho que actualmente no hay suficientes direcciones IPv4 para enrutar públicamente a los usuarios y proporcionar una dirección distinta para cada terminal con IPv4 (se incluyen computadores personales, teléfonos móviles, dispositivos integrados y hosts virtuales). Este problema se minimiza mediante la aplicación de NAT, Network Address Translation, por medio de la

cual diversos host de redes de área local (LAN) pueden compartir una única dirección IP pública de Internet.

Dada la imperiosa necesidad de implementar un nuevo protocolo de red, los desarrolladores de plataformas de comunicación se vieron en la obligación de crear un nuevo protocolo de red IP versión 6 (IPv6), y por consiguiente el desarrollo de servicios que realicen la transición entre estas dos versiones IPv4 e IPv6.

Hoy la gran mayoría de los ISP en el mercado Ecuatoriano trabajan bajo la versión IPv4, protocolo que fue creado en la década de los 70's, lo cual nos lleva a la implementación de un Proxy sobre un protocolo de red IPV6 que sea capaz de interactuar con el protocolo de red IPv4, para realizar dicha interacción entre dos tipo de versiones IPV4 del ISP y IPV6 de LAN interna se configura el proxy del servidor de forma adecuada por medio de una herramienta administrable y flexible para la persona que administra el sistema y a su vez darle al usuario el servicio de navegación, lo idóneo para el total desempeño de dicho servicio seria la transparencia en las conexiones del cliente a Internet, pero actualmente dicho servicio no está soportada para IPv6 hablando específicamente con respecto al Squid (Linux).



**Grafico 1.1 pagina web squid confirmación de no transparencia en IPv6**

## 1.2 Problemática

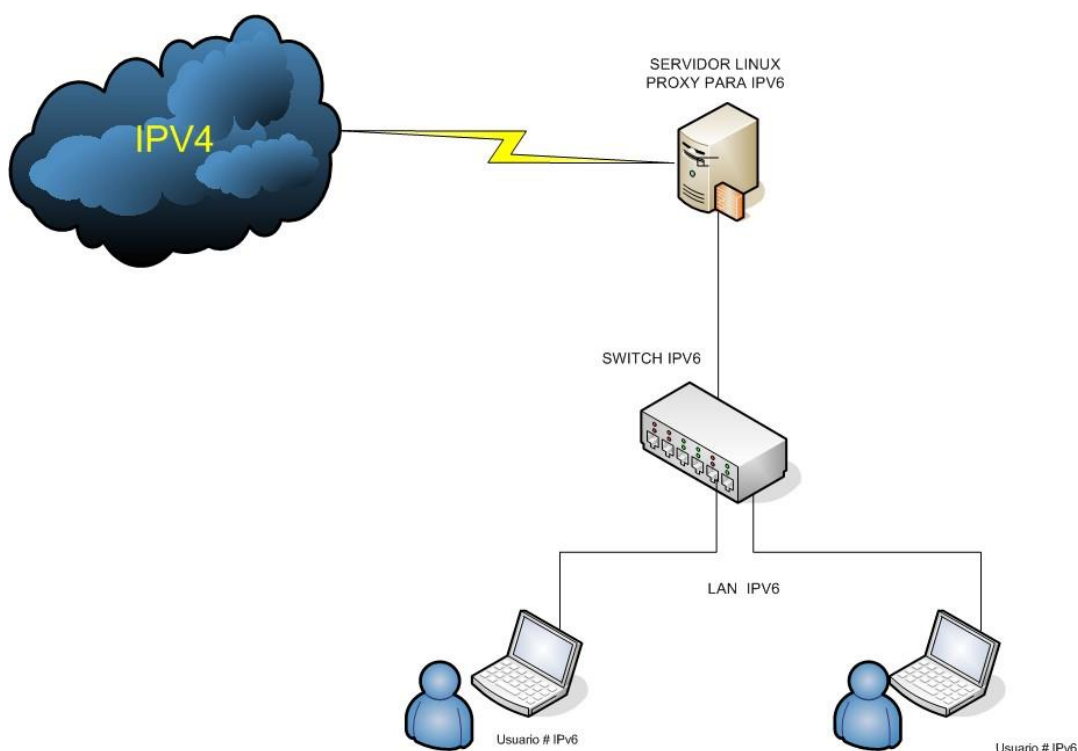
La Problemática fundamental del proyecto es la interacción entre los protocolos IPv4 e IPV6. De tal manera que cuando llegue el momento en que nuestros ISP (Internet Service Provider) migren al protocolo IPV6 no tengamos que efectuar mayores cambios a la configuración de nuestros servidores y Proxy's de acceso a la red.

En la actualidad en algunas partes del mundo, como: EUROPA, EEUU, MEXICO, CHINA e incluso VENEZUELA, han comenzado a ejecutar transición al protocolo IPV6, aunque hay tareas pendientes de implementación sobre los servicios disponibles de IPV6.

## 1.3 Solución

La solución planteada es la implementación de un servidor proxy que este en capacidad de comunicarse con los ISP con protocolos de red IPv4 que actualmente operan en nuestro medio, al mismo tiempo estará en capacidad de administrar los recursos de navegación, accesos a la LAN ( IPv6), tráfico de paquetes y control de usuarios.

Implementar una red IPV6 en nuestro medio no es tan promovido, pese a ello, aprovecharemos las ventajas que nos ofrece este renovado protocolo, tal como otorgarnos mayor disponibilidad de direcciones físicas



**Gráfico 1.2 - Diagrama de Red implementando Servidor Proxy**

## **1.4 Visión**

Convertirse en la principal herramienta de administración del Proxy para el Administrador de red local en IPv6.

## **1.5 Misión**

Cumplir con los requerimientos para el servicio de Internet para los usuarios internos en IPV6 mediante Proxy, en base a la aplicación de soluciones Open Source y herramientas disponibles de Hardware y Software.

## **1.6 Objetivo General**

Configuración de una red LAN IPV6 que permita la navegación en internet bajo la seguridad de un Proxy el mismo que a través de una cuenta de usuario administrador o consultor, gestionará diferentes procesos, como por ejemplo:

- Definición y restricción de permisos de navegación, a los usuarios de una red local (IPV6).
- Filtrado de contenido, por frases o tipos de archivos, para dar seguridad en caso de código malicioso en la Web que suele pasar inadvertido.

- Brindar seguridad a la empresa, mediante la restricción de servicios asociados al Hardware y al Software.
- Optimizar las prestaciones del servicio de Internet en la Red, para aprovechar de mejor manera el uso del ancho de banda.

## **1.7 Objetivo Especifico**

Implementación de un Sistema de Administración Proxy de fácil manejo sobre una red local IPV6, fabricado con herramientas de desarrollo Open Source y aplicaciones Orientadas a Objetos (Java), que trabajarán bajo la plataforma Linux, distribución Centos 5.3, la cual contribuirá a la configuración del mismo.

- Definición de Roles de Usuarios que interactúen con los servicios brindados por el Proxy.
- Restricción de páginas por usuarios o por IP, para establecer prioridades de aprovechamiento de los recursos de Internet.
- Restricción de necesidades de ancho de banda según convenga el administrador de la Red local.
- Denegar accesos a páginas o puertos que puedan ser focos de infección para nuestra red.



## 1.8 Alcance

1. Mantenimiento de Cuentas de Usuarios.
  - a. Creación de Roles.
  - b. Edición y eliminación de usuarios.
2. Restricciones:
  - a. Restricción por Frases.
  - b. Restricción por Clasificación.
  - c. Establecimiento de reglas por IP o por Grupos  
Restricciones de Ancho Banda por IP ó por Grupo
  - d. Puerto de navegación parametrizable
  - e. Restricciones por Tipos de Archivos
3. Tráfico de paquetes que envíe o reciba nuestra intranet
4. Generación de reportes.
  - a. Usuarios existentes.
  - b. Acceso de Usuarios a la aplicación.
  - c. Reglas establecidas por IP ó por Grupo.
  - d. Páginas de Navegación por IP ó Usuario.
5. Control de Acceso por IP y por usuario.
  - a. Creación de lista de permisos o negación de acciones.
6. Edición de parámetros de red.
  - a. Configuración de Puertos.

## **1.9 Ventajas**

### **1.9.1 Acceso:**

- a) Generación de negocio respecto a la implementación de la tecnología LAN IPV6 que está en expansión por la escasez de direcciones como es actualmente IPv4
- b) Acceso y configuración remota ya sea dentro o fuera de la red.
- c) Los servicios Squid serán de fácil manejo dentro de la aplicación que se administre en IPV6.
- d) Facilidad en la configuración del Proxy y filtro de contenido en el servidor principal.

### **1.9.2 Costo:**

- a) Ahorros por concepto de mantenimiento de servidor Proxy ya que al ser configurado en la implementación de forma óptima contribuirá a el buen funcionamiento de la red.
- b) Bajo consumo de recurso humano orientado al soporte Linux, ya que no necesita tener conocimientos avanzados para los servicios Squid.
- c) Optimizando los recursos de ancho de banda desde la aplicación no es imperioso invertir en su incremento.

## 1.10 Desventajas

- Debido a la problemática encontrada en la implementación de un servidor proxy transparente en ipv6 , se lo tiene que configurar en el explorador de forma manual, ya que la arquitectura IPv6 no soporta el NAT.
- Las direcciones IPv6 son más complicadas de recordar con respecto a las de IPv4, por la complejidad en la conformación de dichas direcciones.
- La escasez de equipos compatibles con la tecnología IPv6 que aun no son tan fáciles de conseguir en nuestro medio tales como Routers y Swichs y otros equipos necesarios para el buen funcionamiento.
- El soporte local para esta nueva versión de IPs es muy escaso ya que en la actualidad se encuentra poco difundida a pesar que en algunas partes de nuestra región ya han migrado a ella.

## 1.11 Arquitectura

### 1.11.1 Arquitectura IPv6:

#### a) Direccionamiento

El cambio más grande de IPv4 a IPv6 es la longitud de las direcciones de red. Las direcciones IPv6, definidas en el RFC 2373 y RFC 2374, son de 128 bits; esto corresponde a 32 dígitos hexadecimales, que se utilizan normalmente para escribir las direcciones IPv6, como se describe en la siguiente sección.

El número de direcciones IPv6 posibles es de  $2^{128} \approx 3.4 \times 10^{38}$ . Este número puede también representarse como  $16^{32}$ , con 32 dígitos hexadecimales, cada uno de los cuales puede tomar 16 valores (véase combinatoria).

En muchas ocasiones las direcciones IPv6 están compuestas por dos partes lógicas: un prefijo de 64 bits y otra parte de 64 bits que corresponde al identificador de interfaz, que casi siempre se genera automáticamente a partir de la dirección MAC de la interfaz a la que está asignada la dirección.

Notación para las direcciones IPv6 [editar] Las direcciones IPv6, de 128 bits de longitud, se escriben como ocho grupos de cuatro dígitos hexadecimales.

Por ejemplo

2001:0db8:85a3:08d3:1319:8a2e:0370:7334

## b) Paquetes

Un paquete en IPv6 está compuesto principalmente de dos partes: la cabecera y los datos.



Gráfico 1.3 – Estructura de un paquete de ipv6

Los primeros 40 bytes (320 bits) son la cabecera del paquete y contiene las direcciones de origen y destino (128 bits cada una), la versión del protocolo IP (4 bits), la clase de tráfico (8 bits, Prioridad del Paquete), la etiqueta de flujo (20 bits, manejo de la Calidad de Servicio), la longitud del campo de datos (16 bits), la cabecera siguiente (8 bits), y el límite de saltos (8 bits, Tiempo de Vida). Después viene el campo de datos, con los datos que

transporta el paquete, que puede llegar hasta 64k de tamaño en el modo normal, o más con la opción "jumbo payload".

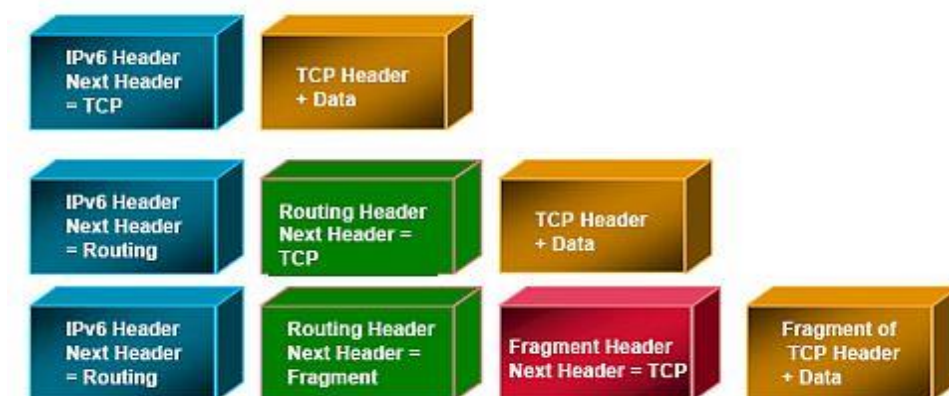


Gráfico 1.4 – Conformacion de un paquete ipv6

### 1.11.2 Arquitectura de tres capas:

Especialización de la Arquitectura Cliente Servidor, donde la carga se divide en tres partes con un reparto claro de funciones, Una capa para la Presentación, otra para el Negocio y otra para el Almacenamiento.

#### c) Capa de Presentación

Los servicios de presentación proporcionan la interfaz necesaria para presentar información y reunir datos. También aseguran los servicios de negocios necesarios para ofrecer las capacidades de transacciones requeridas e integrar al usuario con la aplicación para ejecutar un proceso de negocios.

Los servicios de presentación generalmente son identificados con la interfaz de usuario, y normalmente residen en un programa ejecutable localizado en

la estación de trabajo del usuario final. Aún así, existen oportunidades para identificar servicios que residen en componentes separados.

El cliente proporciona el contexto de presentación, generalmente un browser como Microsoft Internet Explorer o Netscape Navigator, que permite ver los datos remotos a través de una capa de presentación HTML, O también una aplicación WIN32 como ser los formularios de Visual Basic.

Mediante el uso de componentes, se separa la programación que da acceso a los datos en las bases de datos y aplicaciones desde el diseño y otros contenidos de la página Web. Esto ayuda a asegurar que los desarrolladores estén libres para enfocarse en escribir su lógica de negocios en componentes sin preocuparse acerca de cómo se muestra la salida. Recíprocamente, esto da libertad a los diseñadores de usar herramientas familiares para modificar la interfaz.

#### **d) Capa de Negocios**

Los servicios de negocios son el “puente” entre un usuario y los servicios de datos. Responden a peticiones del usuario (u otros servicios de negocios) para ejecutar una tarea de este tipo. Cumplen con esto aplicando procedimientos formales y reglas de negocio a los datos relevantes. Cuando los datos necesarios residen en un servidor de bases de datos, garantizan los servicios de datos indispensables para cumplir con la tarea de negocios o aplicar su regla. Esto aísla al usuario de la interacción directa con la base de datos.

Una tarea de negocios es una operación definida por los requerimientos de la aplicación, como introducir una orden de compra o imprimir una lista de clientes. Las reglas de negocio (business rules) son políticas que controlan el flujo de las tareas.

Como las reglas de negocio tienden a cambiar más frecuentemente que las tareas específicas de negocios a las que dan soporte, son candidatos ideales para encapsularlas en componentes que están lógicamente separados de la lógica de la aplicación en sí.

Para ayudar a los desarrolladores a construir la lógica de negocio basado en componentes Windows DNA incluye un conjunto muy poderoso de servicios que se encargan de la comunicación en una aplicación de tres capas. Estos servicios están altamente integrados unos con otros bajo un sistema operativo y expuestos de forma única a través de COM.

El nivel de servicios de negocios es responsable de:

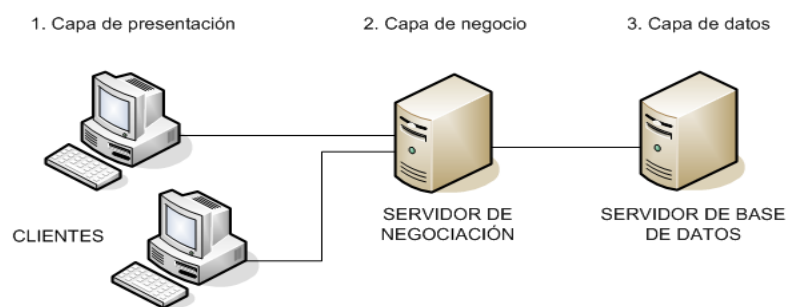
- Recibir la entrada del nivel de presentación.
- Interactuar con los servicios de datos para ejecutar las operaciones de negocios para los que la aplicación fue diseñada a automatizar (por ejemplo, la preparación de impuestos por ingresos, el procesamiento de ordenes y así sucesivamente).
- Enviar el resultado procesado al nivel de presentación.



### e) Capa de Datos

El nivel de servicios de datos es responsable de:

- Almacenar los datos.
- Recuperar los datos.
- Mantener los datos.
- La integridad de los datos.



**Gráfico 1.5 - Diagrama de Arquitectura tres capas.**

## 1.12 Modelo de Desarrollo

Respecto al modelo de desarrollo la ingeniería de Software contiene varios modelos de los cuales usaremos la más apropiada para nuestro proyecto, sin descartar los otros modelos importantes como:

Modelo en Cascada, Modelo Prototipo, Modelo Espiral

Nuestro proyecto será basado en el Modelo de Espiral, por que se tiene que tener en cuenta los problemas y riesgos que se pueden presentar a lo largo del desarrollo del proyecto.

Si el administrador o cliente quiere aportar con mejoras en el software, se vuelve a evaluar las distintas nuevas alternativas y se realiza otra vuelta del espiral hasta que llegue un momento que el producto desarrollado sea aceptado y el mas idóneo con los mejores resultados .

En conclusión en modelo a seguir para nuestro caso es el Espiral es el más óptimo para el sistema, por su facilidad que da y permite ir comprobando fase a fase.

Para un mejor entendimiento a continuación esta la descripción de cada región del modelo antes mencionado:

Región 1 - Tareas requeridas para establecer la comunicación entre el cliente y el desarrollador.

Región 2 - Tareas inherentes a la definición de los recursos, tiempo y otra información relacionada con el proyecto.

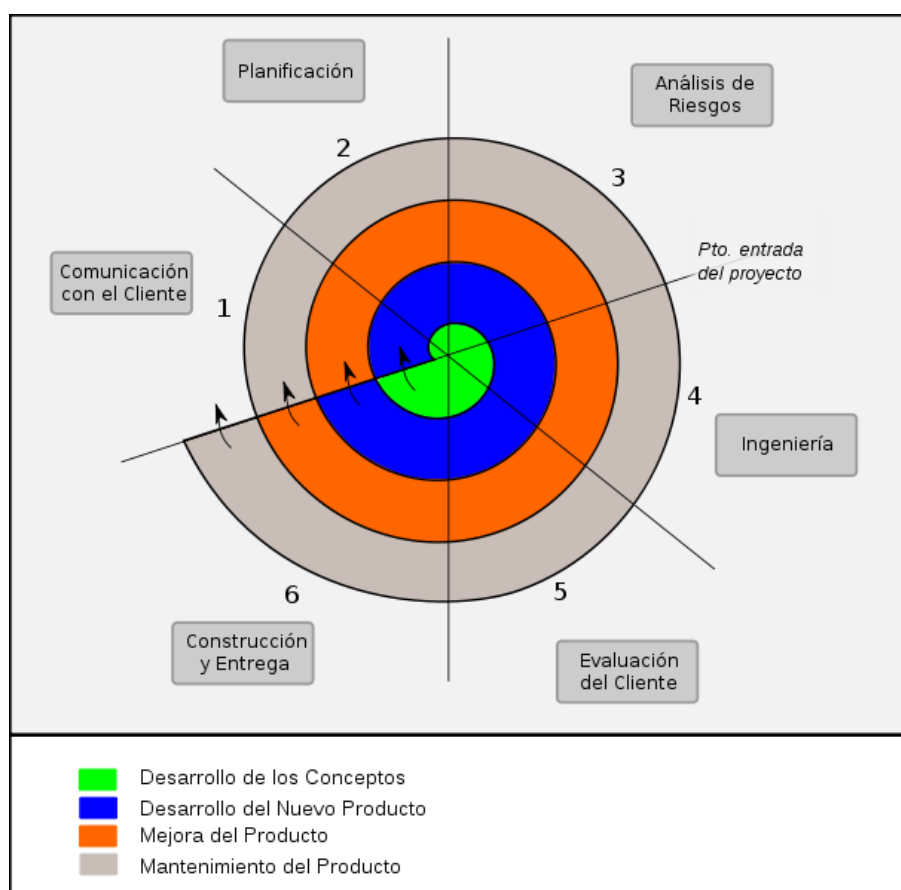
Región 3 - Tareas necesarias para evaluar los riesgos técnicos y de gestión del proyecto.

Región 4 - Tareas para construir una o más representaciones de la aplicación software.

Región 5 - Tareas para construir la aplicación, instalarla, probarla y proporcionar soporte al usuario o cliente (Ej. documentación y práctica).

Región 6 - Tareas para obtener la reacción del cliente, según la evaluación de lo creado e instalado en los ciclos anteriores.

Las regiones definidas en el modelo de la figura son:



**Gráfico 1.6 - Diagrama Modelo espiral**

En el caso de la realización de las Interfaces Gráficas de Usuarios se utiliza el Modelo de Prototipos.

## **1.13 Metodología**

La metodología usada para el desarrollo de este sistema, es la Orientada a Objeto.

En cuanto a la funcionalidad y la metodología interna, la programación orientada a objetos se fundamenta en la encapsulación, la herencia y el polimorfismo.

### **1) Encapsulación**

Una clase se declaran los tipos de datos y el medio de manipularlos (sus métodos).

### **2) Herencia**

La herencia es la propiedad que permite a los objetos construirse a partir de otros objetos.

Una clase se puede dividir en subclases. La clase original se denomina clase base; las clases que se definen a partir de la clase

base, compartiendo sus características y añadiendo otras nuevas, se denominan clases derivadas.

Las clases derivadas pueden heredar código y datos de su clase base añadiendo su propio código y datos a la misma.

La herencia impone una relación jerárquica entre clases en la cual una clase hija hereda de su clase padre. Si una clase sólo puede recibir características de otra clase base, la herencia se denomina herencia simple.

Si una clase recibe propiedades de más de una clase base, la herencia se denomina herencia múltiple.

### **3) Polimorfismo**

En un sentido literal, significa la cualidad de tener más de una forma.

En el contexto de POO, el polimorfismo se refiere al hecho de que una misma operación puede tener diferente comportamiento en diferentes objetos. Por ejemplo, consideremos la operación sumar. El operador + realiza la suma de dos números de diferente tipo. Además se puede definir la operación de sumar dos cadenas mediante el operador suma.

## 1.14 Recursos

### 1.14.1 Software

#### 1.14.1.1 Plataforma

1) **Linux CENTOS 5.3** – Distribución de Linux, basado en el Sistema Operativo Linux. Fue desarrollado por un gran numero de personas de la comunidad que se esfuerza en proveer y mantener lo mejor del software libre, el software de código abierto. Soporta arquitecturas de microprocesadores x86, x86-64 y Power PC. Tiene un administrador virtual de interfaz gráfica de usuario

#### 1.14.1.2 Servicios

1) **Squid 3.1.0.13** - software libre que implementa un servidor Proxy y un demonio para Web caché, publicado bajo licencia GPL Se considera muy completo y robusto, soporta muchos protocolos, aunque se usa principalmente para HTTP y FTP. Se añade soporte también a TLS, SSL, Internet Gopher y HTTPS.

### 1.14.1.3 Herramientas de Desarrollo

1) **Jdk 1.5.0.** - Incluye el API de Java, el JRE ( JVM ), compilador de Java y otras funcionalidades definidas por Sun.

2) **NetBeans IDE 6.5** - Plataforma de software de Código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Ofrece un ambiente gráfico en los que se tiene acceso a mayor número de herramientas no ofrecidas en los JDK's: Debuggers más elaborados, check-points dentro de la compilación, creación de WAR's (Web-Archives), "Wizards" para acelerar desarrollo, etc. Con él se puede desarrollar y desplegar bajo un mismo entorno aplicaciones empresariales compuestas por elementos de escritorio (aplicaciones J2SE que son las típicas ventanitas o aplicaciones de consola), y aplicaciones Web con sus respectivos jsp, ejb's, servlets, beans, xml, etc que son aplicaciones que ves en una ventana de explorador básicamente.

#### **1.14.1.4 Base de Datos**

- 1) MySQL 5.1.40.- MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) que funciona como un servidor que proporciona acceso multi-usuario a un número de bases de datos.

#### **1.14.1.5 Herramientas de Diseño**

- 1) **Adobe Dreamweaver CS3.** – Editor de páginas Web creado por Macromedia. Soporta estándares del World Wide Web Consortium.

### **1.14.2 Hardware**

#### **1.14.2.1 Computador Personal**

- 1) Procesador Pentium 4.
- 2) Disco duro 80 GB o superior.
- 3) Memoria Ram 1GB.
- 4) 2 Tarjetas de red.

### **1.14.3 Humano**

3 Analistas programadores responsables del proyecto de software.

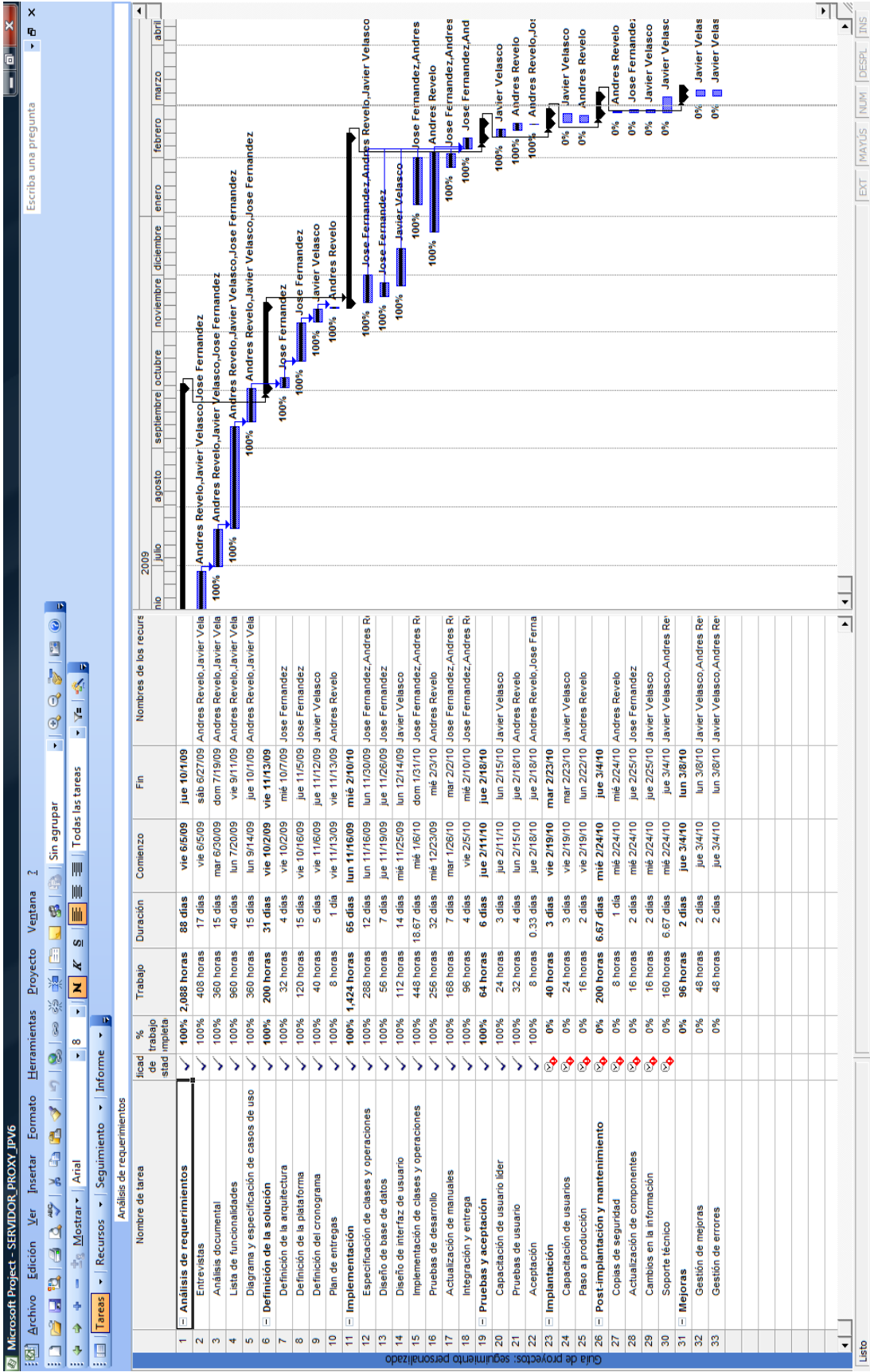
Jose Orlando Fernandez Mora.

Javier Rolando Velasco Ruiz.

Andrés David Revelo Rodríguez.



1.15 Cronograma de Actividades



# **CAPITULO 2**

## **ANÁLISIS**

### **2.1.- Levantamiento de la información**

#### **2.1.1.- Por que realizar este proyecto**

Este proyecto fue diseñado para que ante la migración hacia esta nueva tecnología IPV6 estemos preparados en cuanto a la compatibilidad entre la IPv4 y la IPv6 que esta siendo difundida cada vez alrededor del mundo.

He allí que la idea de la creación de esta herramienta es excelente y muy útil, para la administración de los recursos de internet en esta nueva versión del protocolo tcp/ip ya que nos estaríamos anticipando a la llegada de esta nueva tecnología la cual ofrece un sin número de prestaciones entre las más importantes podríamos citar la disponibilidad de direcciones ips disponibles para cada uno de los equipos de red . El uso de esta herramienta nos permitirá la toma de decisiones en cuanto al establecimiento de políticas de acceso a los recursos de conectividad de la empresa donde sea implementado.

### **2.1.2 Quienes son los usuarios finales.**

La herramienta va dirigida a cualquier tipo de auditor de sistemas y/o administradores de sistemas, los usuarios finales pueden tomar las mejores decisiones en base a estadísticas y reportes de los consumos realizados por los grupos de usuario, usuarios etc....

### **2.1.3 Formas de Levantar Información utilizadas**

Por ser una herramienta de servicio novedoso aquí en el Ecuador tenemos que buscar la manera de mejorarlo cada día para que pueda ser mas difundido en nuestro país.

### **2.1.3.1 Investigación en Internet**

El Internet es una de las herramientas que más hemos utilizado, a nivel de consulta e investigación en este proyecto; por medio de esta tecnología hemos podido concluir dudas con respecto a la transparencia del proxy en ipv6 ya que hasta el término de desarrollo de este proyecto no ha existido soporte de squid en cuanto a la transparencia. De ahí en más pudimos despejar esta interrogante y seguir con la investigación de las demás herramientas a utilizar.

## **2.2 Caso de Uso**

### **Caso de Uso**

Representa la forma en como un Usuario (Actor) opera con el sistema, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Es una operación o tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

## 2.1.1 Diagrama de Caso de Uso

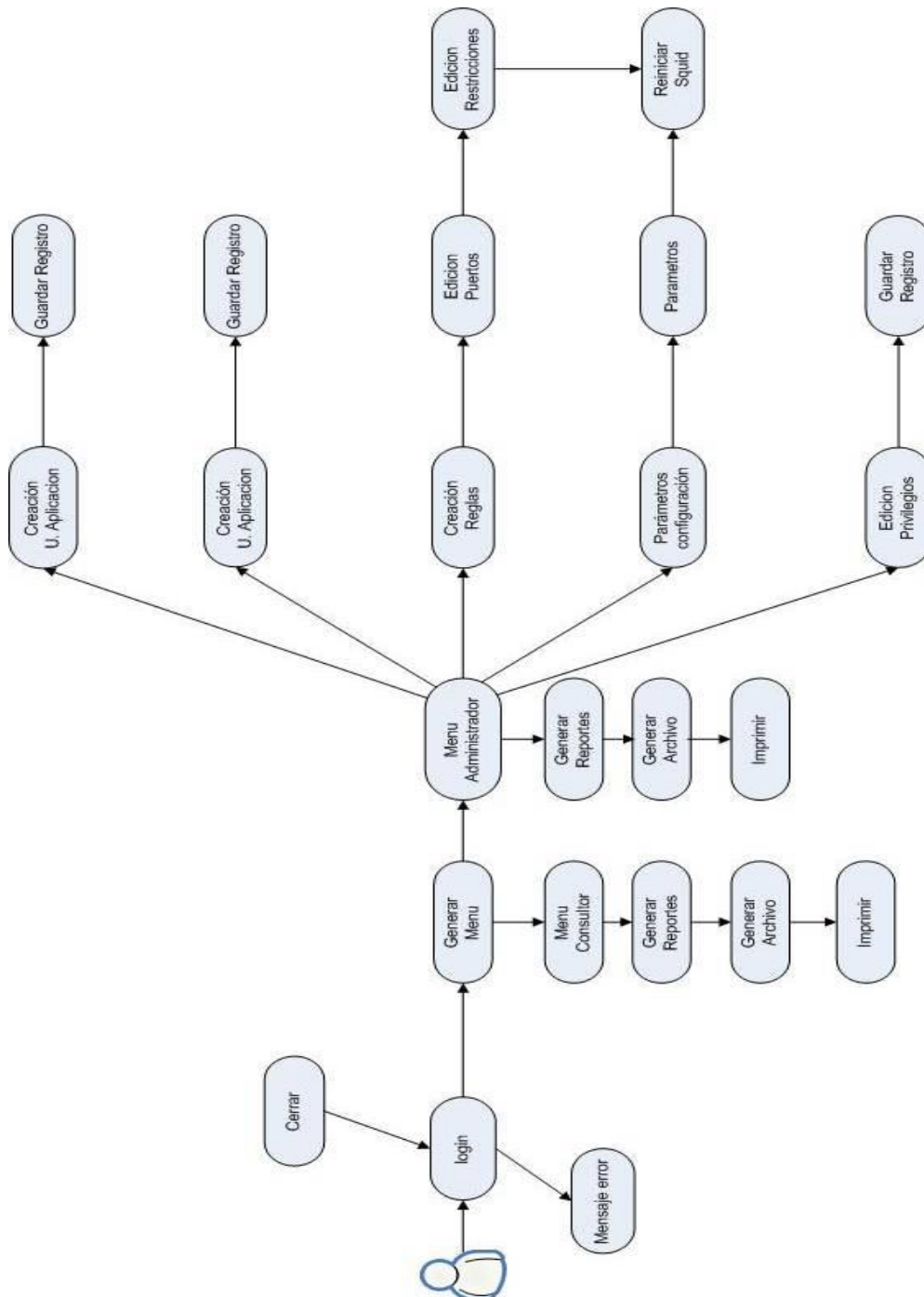
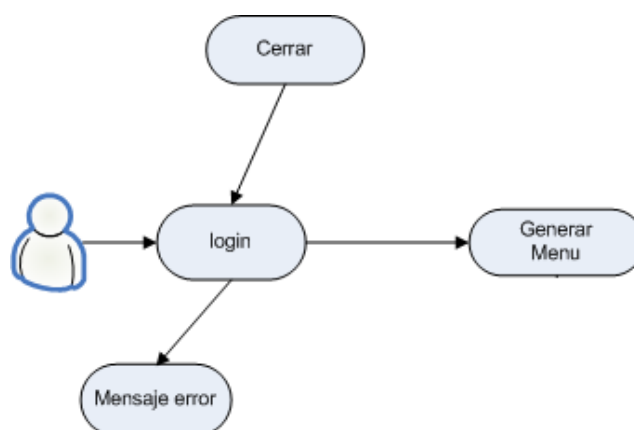


Gráfico 2.1 - Diagrama de Caso de Uso Global

## 2.1.2 Detalle de los Casos de Uso



**Gráfico 2.2 – Casos de Uso - Login**

DESCRIPCION DE CASOS DE USO	
Nombre:	Login
Actores:	Administrador de Sistemas, Administrador de Red, HelpDesk
Función:	Permitir el acceso al Sistema
Descripción:	El sistema debe validar que el usuario y contraseña sean los correctos, y de acuerdo a los permisos mostrar las diferentes opciones definidas en el Rol de Usuario.

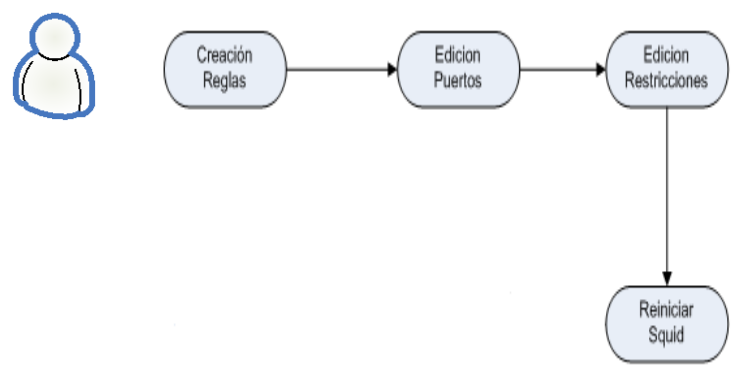


Gráfico 2.3 – Casos de Uso – Creación Reglas

DESCRIPCION DE CASOS DE USO	
Nombre:	Creación de Reglas
Actores:	Administrador
Función:	Creación de Reglas
Descripción:	Crea las reglas de navegación que gobiernan a los usuarios en base a las necesidades de los mismos y al análisis previo por parte del encargado del área de sistemas o el administrador del centro de computo sobre quien recaerá la responsabilidad de los accesos otorgados a dichos usuarios.

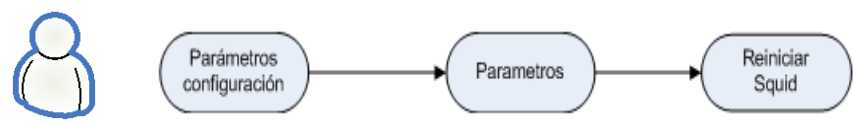
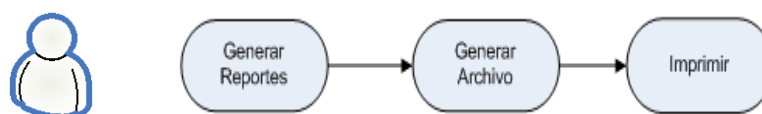


Gráfico 2.4 – Casos de Uso – Parámetros de Configuración

DESCRIPCION DE CASOS DE USO	
Nombre:	Parámetros de Configuración.
Actores:	Administrador.
Función:	Modificación de los parámetros de Configuración.
Descripción:	Asignación de los permisos de cada usuario creados previamente en el menú usuarios, quedara a criterio del encargado del departamento de sistemas las opciones de acceso de cada usuario de la aplicación.



**Gráfico 2.5 – Casos de Uso – Generación de Reportes**

DESCRIPCION DE CASOS DE USO	
Nombre:	Generación de Reportes.
Actores:	Administrador, Help Desk
Función:	Generación de Reportes.
Descripción:	Genera los reportes de acuerdo a los privilegios de usuario , indispensables para la toma de decisiones y verificación del correcto aprovechamiento de los recursos de internet.



## 2.2 Diagrama Entidad Relación

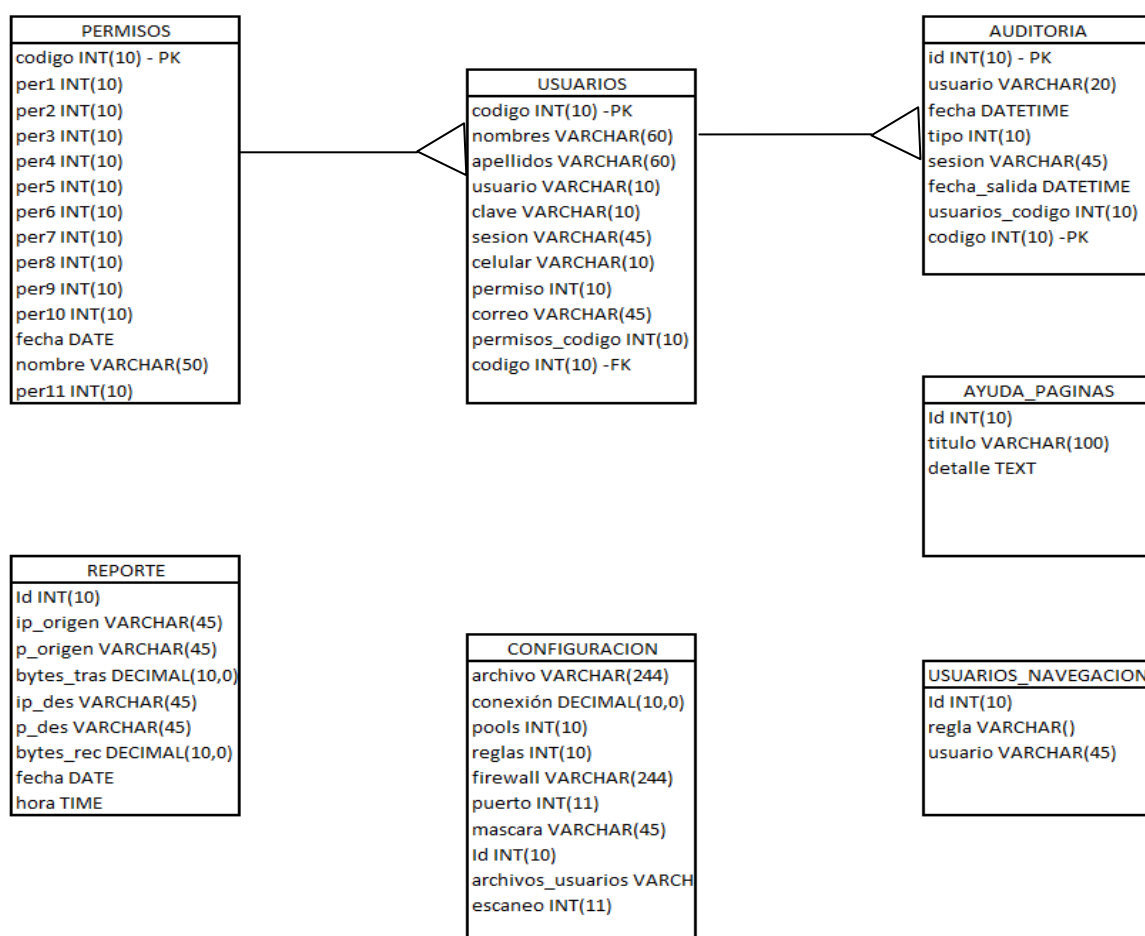


Gráfico 2.6 – Diagrama Entidad Relación

## 2.3 Diagrama de Especificación de Procesos

LOGIN	
Acciones de los Actores	Respuestas del Sistema
El Usuario ingresa el nombre de usuario y la contraseña.	El sistema confirma si el usuario y contraseña son correctos. Si es correcto, le muestra la siguiente Interfaz con un menú de opciones, de acuerdo al Rol.
El Usuario ingresa mal el nombre de usuario y contraseña.	El sistema le muestra un mensaje de error y le envía a la misma interfaz para pueda ingresarlos nuevamente.
El Usuario ingresa mal por tercera vez el nombre de usuario y la contraseña.	El sistema le muestra un mensaje de error y se cierra.

**Tabla 2.3.1 - Especificación de Proceso Login**

CREACIÓN DE USUARIOS	
Acciones de los Actores	Respuestas del Sistema
El Usuario Administrador ingresa nuevos valores de acuerdo a los parámetros establecidos.	El sistema le muestra un wizard amigable y sencillo de entender al usuario administrador para que proceda a la creación de las reglas según los criterios antes analizados.
El Usuario Administrador ingresa mal los parámetros de ingreso de las reglas.	El sistema le muestra un mensaje de error notificando cual fue el parámetro errado y le envía a la misma interfaz para pueda ingresarlos nuevamente.
El Usuario Administrador deja en blanco los datos de la regla.	El sistema le muestra un mensaje de error notificando que no ha ingresado los parámetros requeridos para el registro del usuario.

**Tabla 1.3.2 - Especificación de Proceso Creación de Reglas.**

<b>CREACIÓN DE PRIVILEGIOS</b>	
<b>Acciones de los Actores</b>	<b>Respuestas del Sistema</b>
El Usuario Administrador ingresa nuevos valores de puertos y censos de trafico.	El sistema almacena los valores ingresados de acuerdo al criterio del administrador ya que el cambio de puerto del squid es algo con lo que se tiene que tener cuidado. Lo mismo ocurre con el parámetro de censo de la herramienta jnettop.
El Usuario Administrador deja en blanco los parámetros requeridos por los servicios.	El sistema le muestra un mensaje de error notificando que no ha ingresado los parámetros adecuados para los servicios.

**Tabla 2.2.3 - Especificación del Proceso de Parámetros de Configuración.**

<b>GENERACION DE REPORTES</b>	
<b>Acciones de los Actores</b>	<b>Respuestas del Sistema</b>
El Usuario HelpDesk o Administrador selecciona el reporte del consumo de Ancho de Banda.	El sistema realiza las consultas en la base de datos necesarias para la visualización del reporte del historial del ancho de banda.
El Usuario HelpDesk o Administrador seleccionan la opción de Auditoria.	El sistema le muestra los usuarios que han sido creados para acceder a las opciones de la aplicación que administrara el servidor proxy.
El Usuario HelpDesk o Administrador solicita la opción de Red Interna.	El sistema verifica en sus archivos (los que existen en el directorio de reglas), el listado de las ip q salen por el proxy, genera un reporte con todas las paginas Web concurridas y el tiempo de navegación.

**Tabla 2.3.3 - Especificación del Proceso de Parámetros de Configuración.**

## **CAPITULO 3**

### **3 DISEÑO**

#### **3.1 Definición y objetivos**

El diseño de la aplicación es unas herramientas creada sobre la plataforma Centos y elaborada con una interfaz amigable dándole la mejor facilidad para la administración y manipulación de los diferentes controles y cambios que se le pueda dar a los usuarios ya sea tanto como los que lleguen a manipular el diseño como a los usuarios de la navegación.

#### **3.2 Diseño del sistema**

El diseño del sistema está compuesto por varias etapas desde la creación de usuarios hasta la parte administrativa del proyecto donde manipula directamente al proxy de la red interna en ipv6 agregando y modificando las reglas creadas por el administrador o consultor según sea el caso de quien ingrese a la aplicación.

La aplicación está compuesta por un menú y consta de los siguientes elementos

Página de inicio principal.- Aquí se ingresa usuario y contraseña

Menú principal.- aquí se visualiza el menú y las sub herramientas para el manejo de las aplicaciones.

Contiene de los sub elementos detallados a continuación:

Usuario del sistema	>creación
	>consultar
Privilegios	>Modificar
Reportes	>Consumo A/B
	>Auditoria
	>Red interna
Administración	>Usuarios de Navegación
	>Reglas de Navegación
	>Parámetros de configuración

### 3.3 Estructura Base de Datos

#### 3.3.1 Tabla Usuarios de la aplicación

select \* from usuarios x

Page Size: 20 Total Rows: 2 Page: 1 of 1 Matching Rows:

#	codigo	nombres	apellidos	usuario	clave	sesion	celular	permiso	correo
1	1	Administrador	Proxy	admin	admin	0.9054681942346525	1212	1	hgd@dsd.com
2	2	Andres	Revelo	andres	andres	<NULL>	0121...	2	andres@hotm...

Figura 3.3.1 Tabla Usuarios de la Aplicación

#### 3.3.2 Tabla Permisos de la aplicación

select \* from permisos x

</

Figura 3.3.2 Tabla Permisos Usuarios de la Aplicación

### 3.3.3 Tabla Permisos privilegio aplicación

select * from permisos x														
#	codigo	per1	per2	per3	per4	per5	per6	per7	per8	per9	per10	fecha	nombre	per11
1	1	1	1	1	1	1	1	1	1	1	1	0 2010-02-15	ADMINISTRADOR	1
2	2	0	1	0	1	1	0	0	1	1	1	0 2010-02-15	CONSULTOR	0

Figura 3.3.3 Tabla Permisos privilegios Usuarios de la Aplicación

### 3.3.4 Tabla Usuarios de la aplicación

select * from usuarios x									
#	codigo	nombres	apellidos	usuario	clave	sesion	celular	permiso	correo
1	1	Administrador	Proxy	admin	admin	0.9054681942346525	1212	1	hgd@dsd.com
2	2	Andres	Revelo	andres	andres	<NULL>	0121...	2	andres@hotmail...

Figura 3.3.4 Tabla Creación y consulta de la Aplicación



### 3.3.5 Tabla Reporte








select * from reporte x								
       Page Size: 20    Total Rows: 6723    Page: 1 of 337    Matching Rows:								
#	id	ip_origen	p_origen	bytes_tras	ip_des	p_des	bytes_rec	fecha
1	1365	fe80::216:76ff:fe42:48a3	3128	3232	fe80::f90d:2f8:7134:6e1a	49816	6250	2010-02-15
2	1366	fe80::216:76ff:fe42:48a3	3128	1434	fe80::f90d:2f8:7134:6e1a	49882	2335	2010-02-15
3	1367	fe80::216:76ff:fe42:48a3	3128	74	fe80::f90d:2f8:7134:6e1a	49881	699	2010-02-15
4	1368	fe80::216:76ff:fe42:48a3	3128	17412	fe80::f90d:2f8:7134:6e1a	49883	1842	2010-02-15
5	1369	fe80::216:76ff:fe42:48a3	3128	268258	fe80::f90d:2f8:7134:6e1a	49883	30387	2010-02-15
6	1370	fe80::216:76ff:fe42:48a3	3128	128081	fe80::f90d:2f8:7134:6e1a	49881	19634	2010-02-15
7	1371	fe80::216:76ff:fe42:48a3	3128	49411	fe80::f90d:2f8:7134:6e1a	49881	7986	2010-02-15
8	1372	fe80::216:76ff:fe42:48a3	3128	34824	fe80::f90d:2f8:7134:6e1a	49884	3684	2010-02-15
9	1373	fe80::216:76ff:fe42:48a3	3128	26204	fe80::f90d:2f8:7134:6e1a	49890	2923	2010-02-15
10	1374	fe80::216:76ff:fe42:48a3	135	0	fe80::f90d:2f8:7134:6e1a	135	86	2010-02-15
11	1375	fe80::216:76ff:fe42:48a3	3128	16989	fe80::f90d:2f8:7134:6e1a	49888	3740	2010-02-15
12	1376	fe80::216:76ff:fe42:48a3	3128	5252	fe80::f90d:2f8:7134:6e1a	50034	1006	2010-02-15
13	1377	fe80::216:76ff:fe42:48a3	3128	687	fe80::f90d:2f8:7134:6e1a	50044	906	2010-02-15
14	1378	fe80::216:76ff:fe42:48a3	3128	152970	fe80::f90d:2f8:7134:6e1a	50050	7870	2010-02-15
15	1379	fe80::216:76ff:fe42:48a3	3128	4644	fe80::f90d:2f8:7134:6e1a	50047	148	2010-02-15
16	1380	fe80::216:76ff:fe42:48a3	135	0	fe80::f90d:2f8:7134:6e1a	135	86	2010-02-15

Figura 3.3.5 Tabla Reporte de Red Interna

### 3.3.6 Tabla Auditoria








select * from auditoria x						
       Page Size: 20    Total Rows: 115    Page: 1 of 6    Matching Rows:						
#	id	usuario	fecha	tipo	sesion	fecha_salida
1	13	admin	2010-01-26 15:02:36.0	0	F3D5F1888008A08C838D528C858F8906	<NULL>
2	14	admin	2010-01-26 15:08:10.0	0	8A883F25825C1F92122703178C1F33EE	<NULL>
3	15	admin	2010-01-26 15:09:07.0	0	94F5D481E9C5315CD695335E47ECD1...	<NULL>
4	16	admin	2010-01-26 15:14:10.0	0	419E744271EED513F2F283533170579E	<NULL>
5	17	admin	2010-01-26 15:20:53.0	0	49A798FCDEFADE928ADA3DE176984...	2010-01-26 15:21:56.0
6	18	admin	2010-01-26 15:22:12.0	0	DC89D2A158270F4EC7624FA12D8597...	2010-01-26 15:22:34.0
7	19	admin	2010-01-26 15:22:46.0	0	938A9E658C4D7F05C312DA41D2103E...	<NULL>
8	20	admin	2010-01-26 15:40:45.0	0	87FCFA81E822C0C43D5EDC8994E775E0	<NULL>
9	21	admin	2010-01-26 15:46:34.0	0	16857653A448A6B1D4F5C860F89205...	<NULL>
10	22	admin	2010-01-26 16:11:25.0	0	01AD225828E08C19FFA90467C78022...	<NULL>
11	23	admin	2010-01-26 16:14:04.0	0	2031044F8876D24ED2EA478A9F83560D	<NULL>

Figura 3.3.6 Tabla Reporte Auditoria

### 3.3.7 Tabla Usuarios navegación

#	id	regla	usuario
1	7	rrhh	fox
2	11	full	fox
3	12	all	fox
4	13	prueba	fox

Figura 3.3.7 Tabla Creación y consulta usuarios de Navegación

### 3.3.8 Tabla Parámetros de Configuración

select \* from configuraci... x

Figura 3.3.8 Tabla Parámetros de configuración



### 3.4 Diseño a nivel de Componentes

#### 3.4.1 Proceso Autentificación

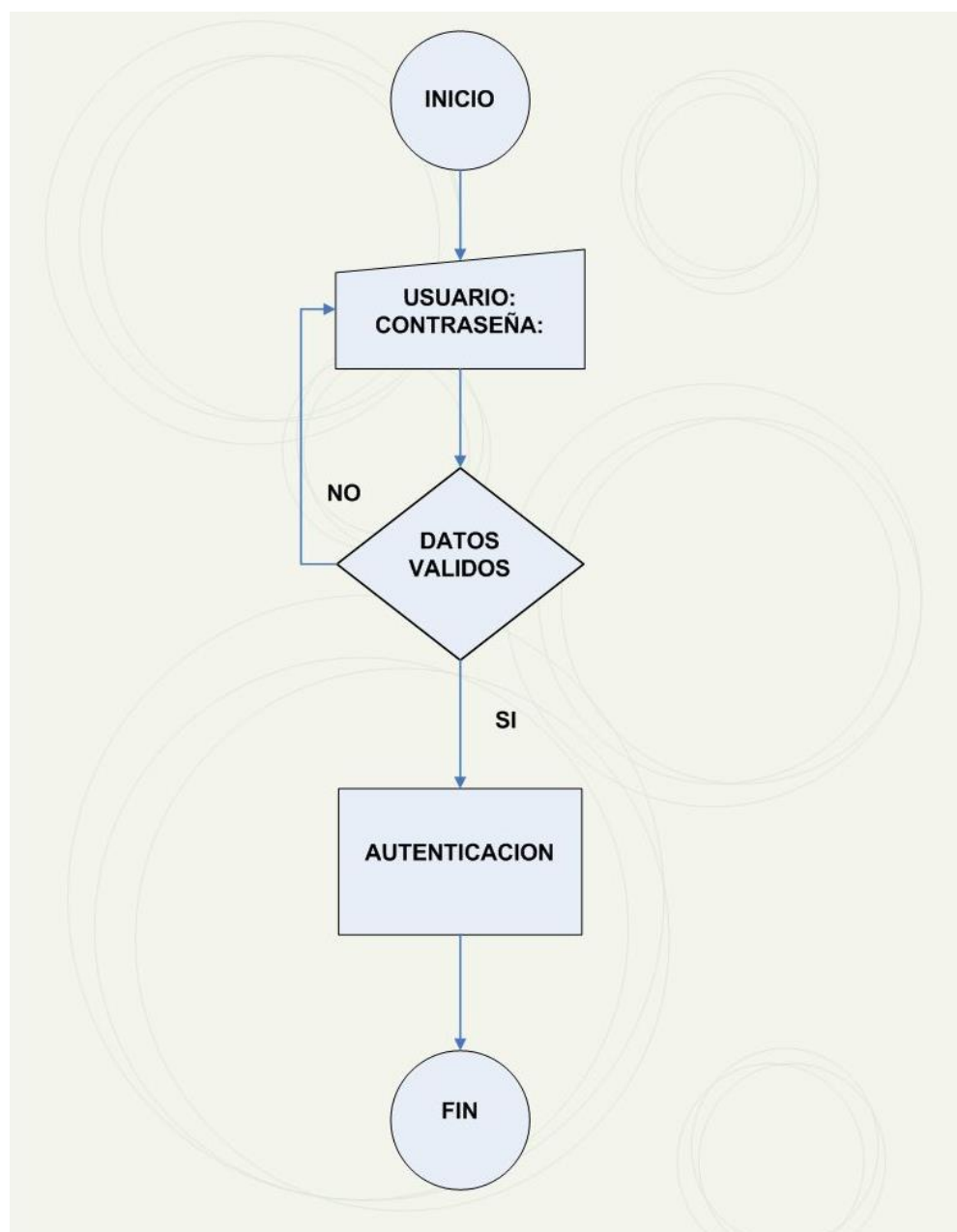


Figura 3.10 Proceso Autentificación

### 3.4.2 Proceso Usuario de la aplicación

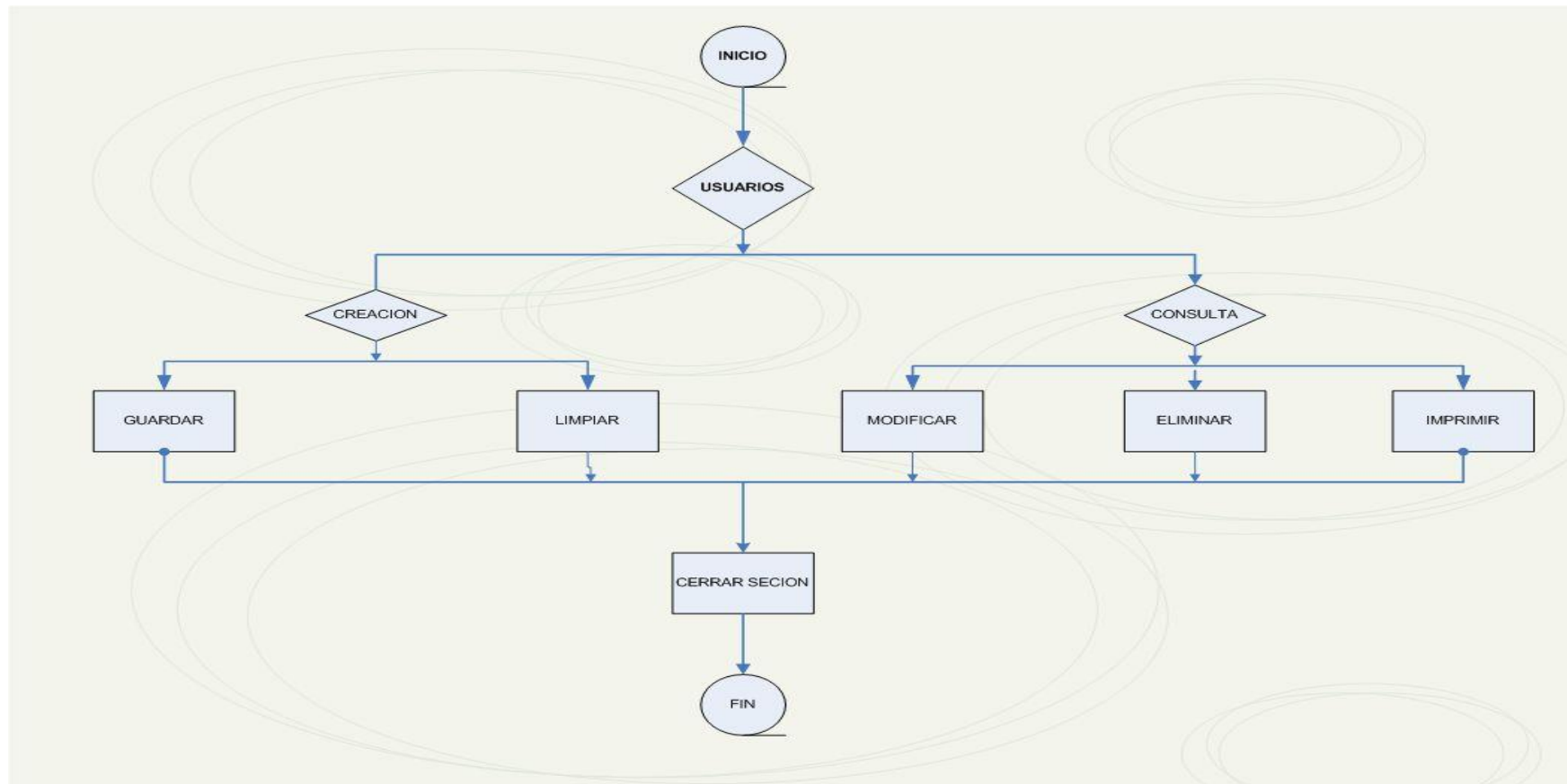


Figura 3.11 Proceso Usuario de la Aplicación



### 3.4.3 Proceso Levantamiento de a Base de Datos

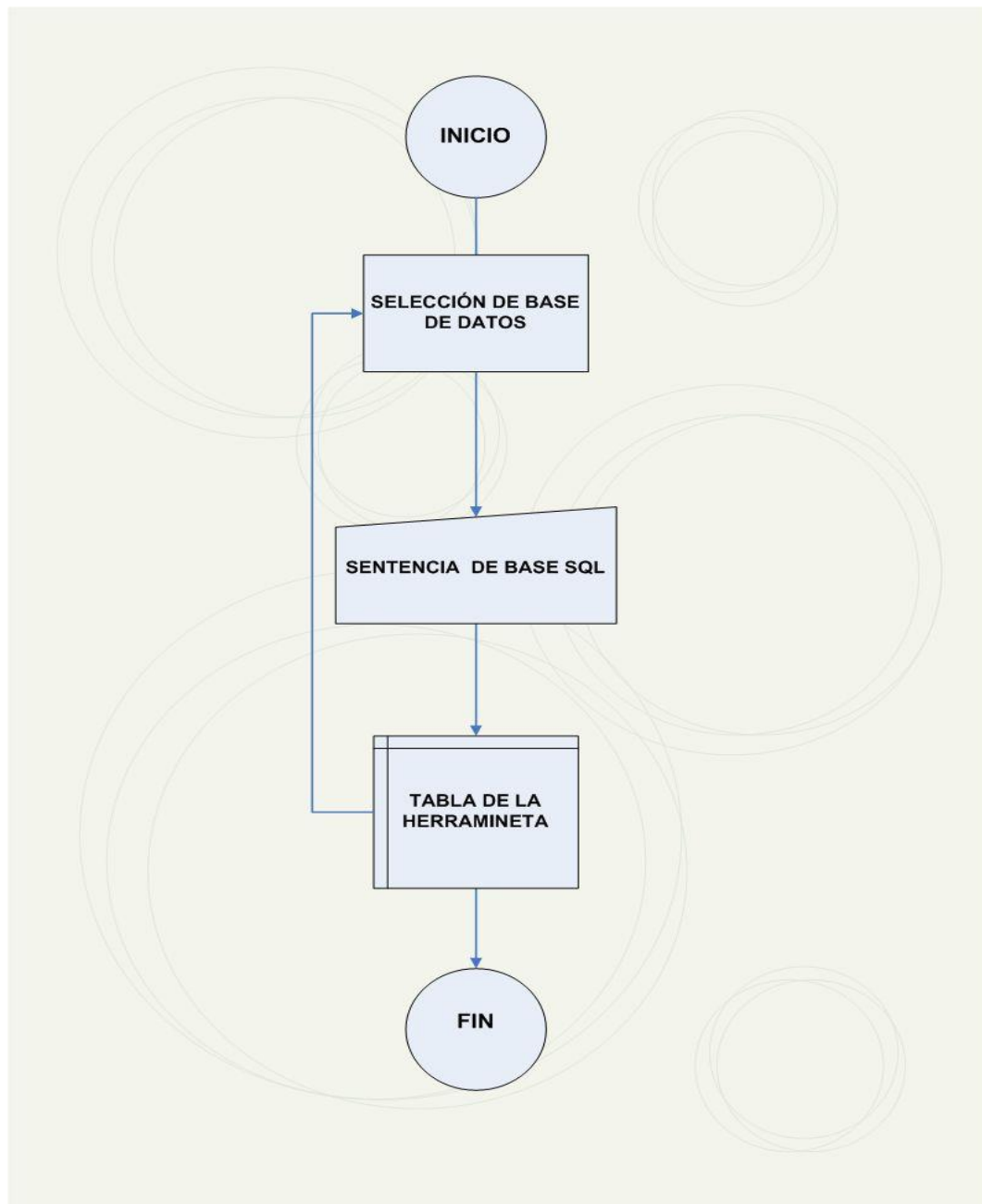


Figura 3.12 Proceso levantamiento de base de dato

### 3.4.4 Proceso Realizar Reporte

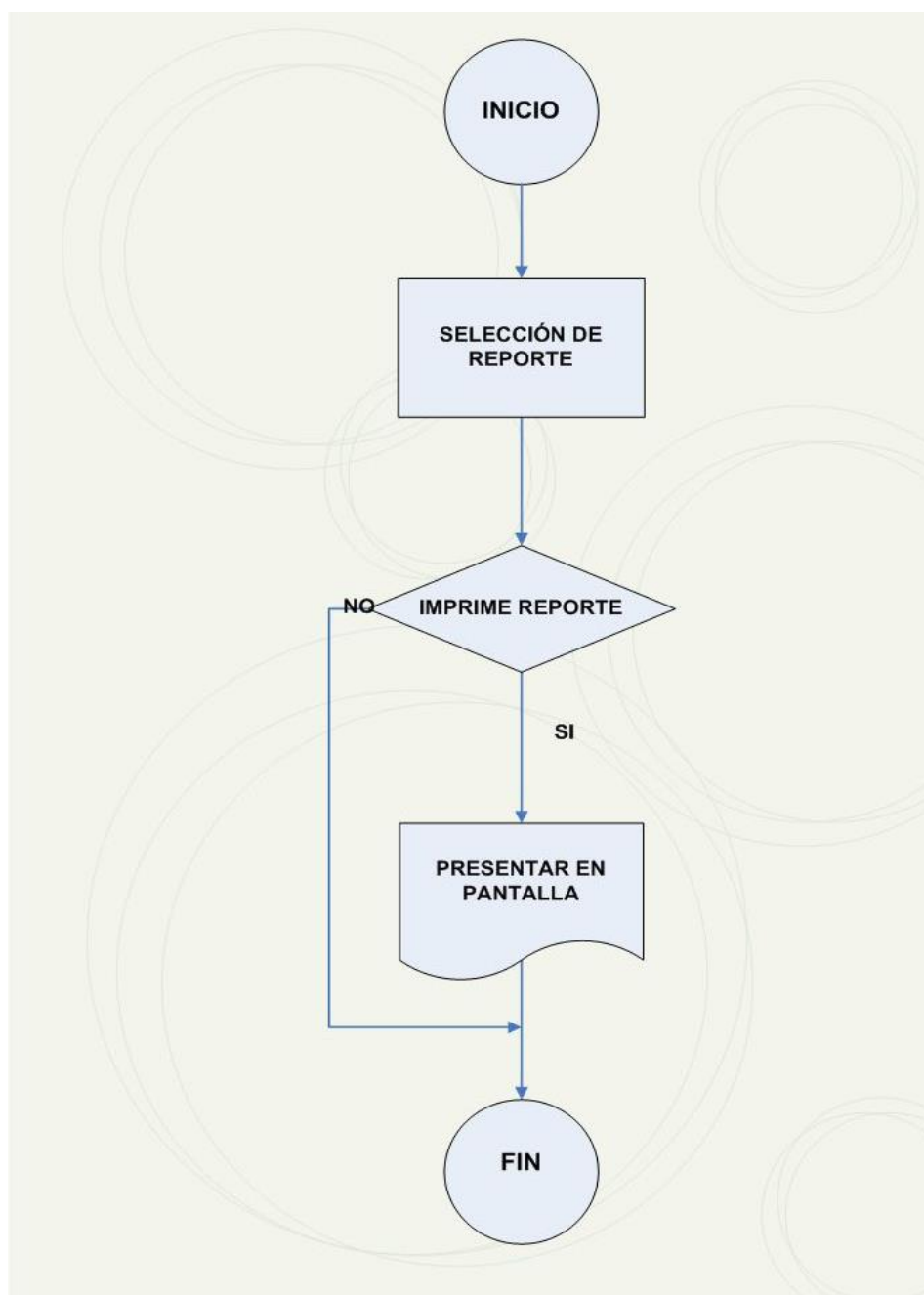


Figura 3.13 Proceso Realizar reporte



### 3.5 Diseño de Interfaz

A continuación veremos las pantallas que contiene la herramienta  
De la administración y manipulación de Proxy en IPv6

#### 3.5.1 Acceso

La aplicación que maneja el proxy en IPV6 es una herramienta que va a permite administrar los usuarios de navegación, en la pantalla principal nos muestra el acceso del usuario ya sea administrador o bien el consultor según sea el caso del tipo de rol con que ingrese el usuario de la aplicación.

El usuario tiene que entrar a la identificación ingresando el nombre y clave para poder ingresar a la pantalla principal



Identificación de Usuario

Usuario :

Clave :

 Ingresar

Done

Figura 3.14 interfaz Acceso

### 3.5.2 Menú Principal

Después que el usuario accedió a la aplicación se le presentara la pantalla del menú principal está compuesta por todas las opciones que va a manipular el administrador o consultor, se va distinguir por los colores:

Color amarillo: Corresponde a los roles, usuarios y privilegios

Color Negro: Corresponde a los reportes

Color Verde: Administración del Proxy, usuarios, reglas, parámetros



Figura 3.15 Interfaz Pantalla principal

### 3.5.3 Pantalla creación Usuario de la aplicación

En esta pantalla se crea los usuarios de la aplicación ingresando todos los datos que se muestra en la pantalla y se le designa que tipo de rol se le dará para su próximo acceso .

The screenshot shows a web application interface with a navigation bar at the top containing links: Principal, Usuarios, Privilegios, Reportes, and Administracion. Below the navigation bar, a welcome message 'Bienvenido : Administrador Proxy' is displayed. The main content area is titled 'CREACION DE USUARIOS DEL SISTEMA' and contains the following form fields:

- Nombres :
- Apellidos :
- Celular :
- Correo :
- Usuario :
- Clave :
- Confirmar Clave :
- Privilegio :

At the bottom of the form are two buttons: 'Guardar' and 'Limpiar'. A 'Cerrar sesión' button is also visible in the top right corner of the application area. A 'Done' status bar is at the very bottom of the browser window.

Figura 3.16 Interfaz Pantalla creación de usuario de la aplicación

### 3.5.4 Pantalla consulta Usuario de la aplicación

En esta pantalla se muestra todos los usuarios del aplicativo, estos usuarios se muestran de forma ordenada según el código de ingreso que el sistema le otorga en el momento de la creación del usuario, en el campo Rol se muestra el nombre del mismo, y se puede tener una idea del tipo de acceso que tendría dicho usuario, se puede modificar, eliminar y también imprimir.



Figura 3.17 Interfaz Pantalla consulta de usuario de la aplicación

### 3.5.5 Pantalla Privilegios de Usuarios de la aplicación

Esta pantalla muestra las diferentes opciones de acceso al menú.

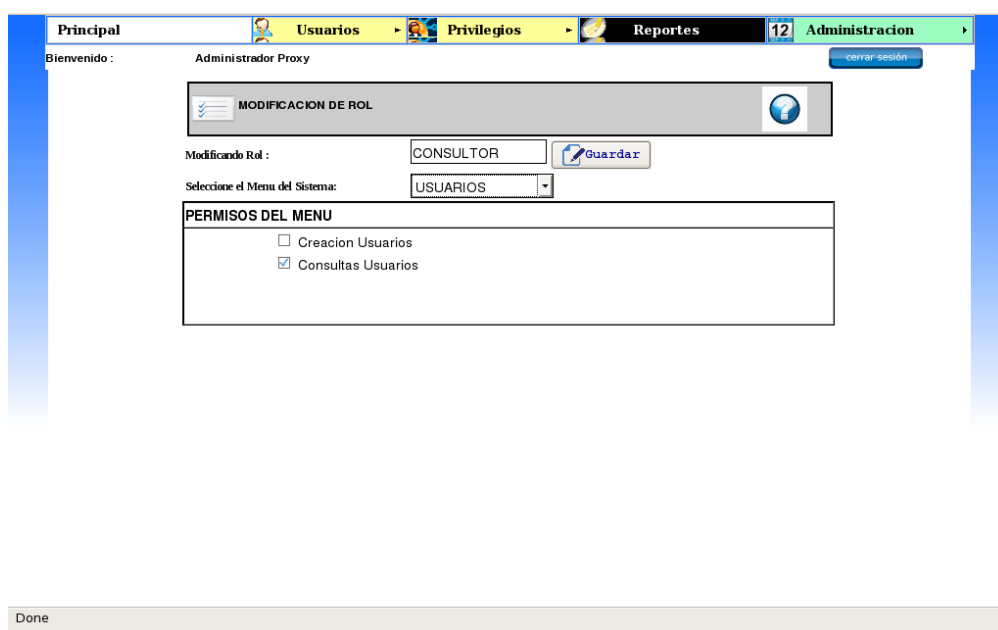


Figura 3.18 Interfaz Pantalla privilegios de usuario de la aplicación

### 3.5.6 Pantalla Reporte consumo A/B

En esta pantalla se muestra de forma grafica el consumo de los usuarios ya sea por clientes o por consumo total.

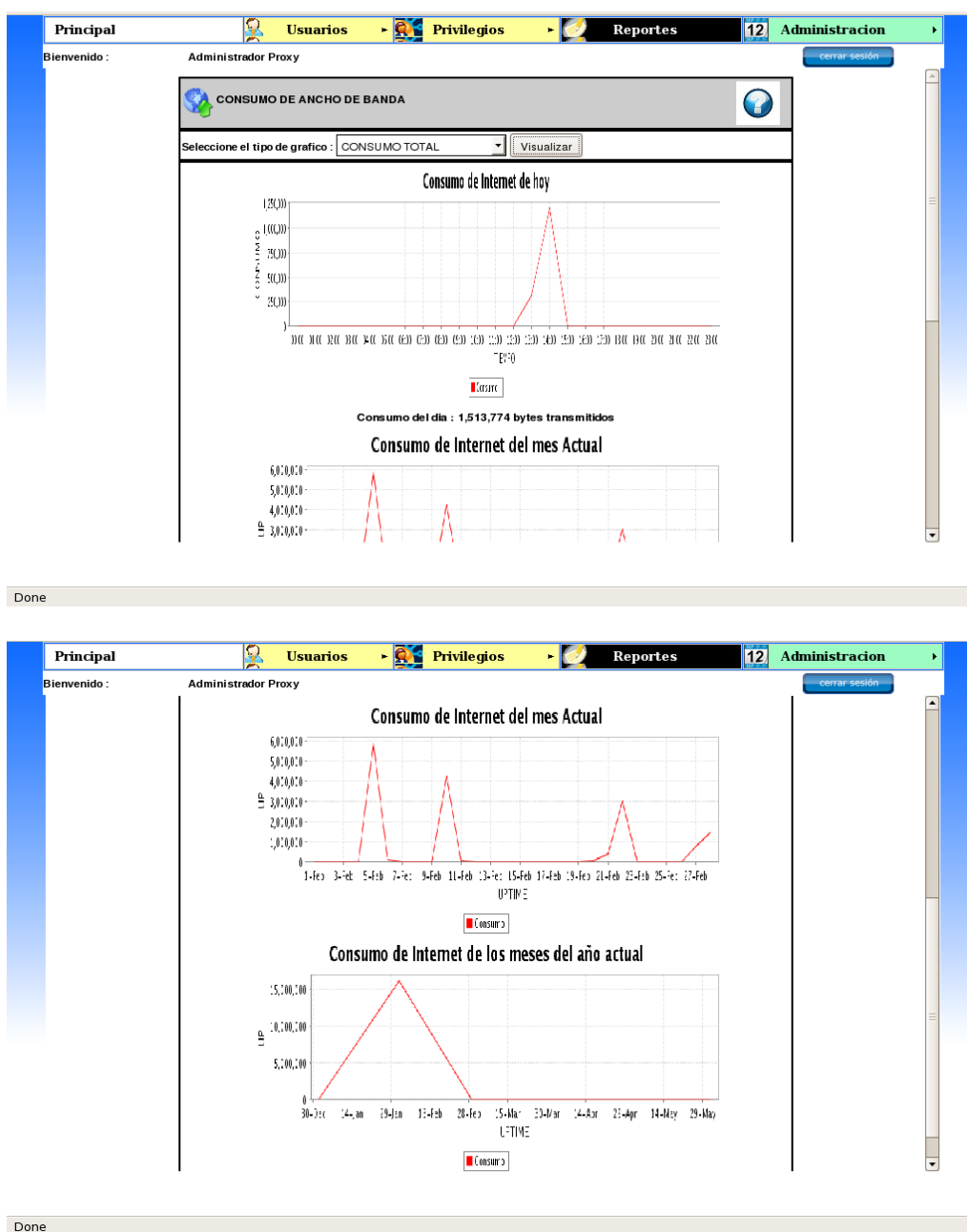
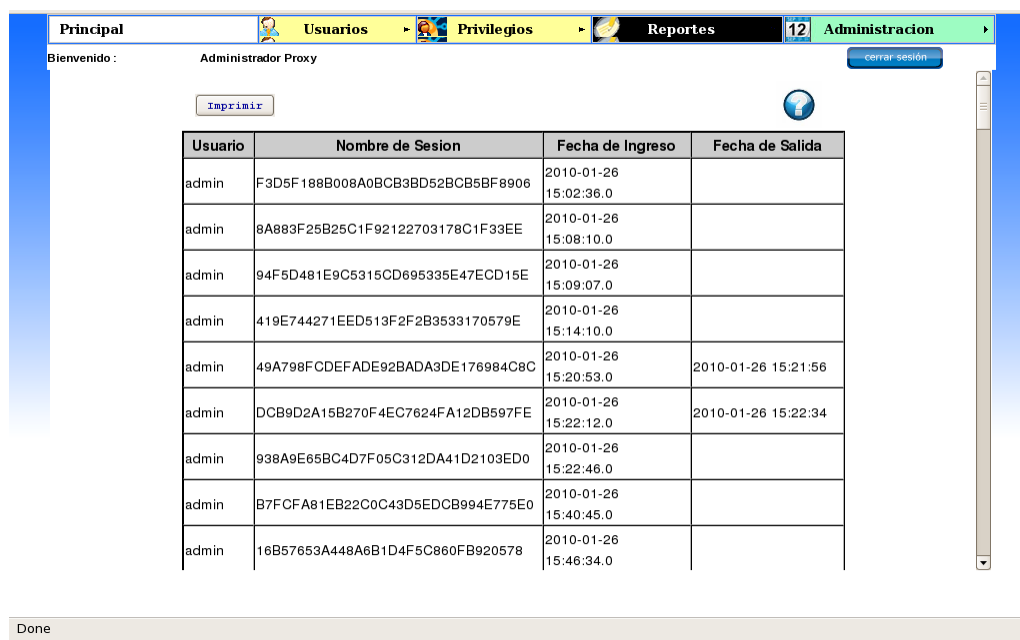


Figura 3.19 Interfaz Pantalla consumo de ancho de banda

### 3.5.7 Pantalla Reporte Auditoria

En esta pantalla nos presenta el listado de los usuarios que ingresan al sistema con la fecha de ingreso y la fecha de salida.



Usuario	Nombre de Sesion	Fecha de Ingreso	Fecha de Salida
admin	F3D5F188B008A0BCB3BD52BCB5BF8906	2010-01-26 15:02:36.0	
admin	8A883F25B25C1F92122703178C1F33EE	2010-01-26 15:08:10.0	
admin	94F5D481E9C5315CD695335E47ECD15E	2010-01-26 15:09:07.0	
admin	419E744271EED513F2F2B3533170579E	2010-01-26 15:14:10.0	
admin	49A798FCDEFAD92BADA3DE176984C8C	2010-01-26 15:20:53.0	2010-01-26 15:21:56
admin	DCB9D2A15B270F4EC7624FA12DB597FE	2010-01-26 15:22:12.0	2010-01-26 15:22:34
admin	938A9E65BC4D7F05C312DA41D2103ED0	2010-01-26 15:22:46.0	
admin	B7FCFA81EB22C0C43D5EDCB994E775E0	2010-01-26 15:40:45.0	
admin	16B57653A448A6B1D4F5C860FB920578	2010-01-26 15:46:34.0	

Figura 3.20 Interfaz Pantalla reporte auditoria usuarios aplicaci3n

### 3.5.8 Pantalla Red interna

En esta pantalla se presentara la informaci3n de la red interna, se ver3 las IP's de origen y destino y los puertos, bytes correspondiente, tambi3n se ver3 a que usuario o grupo corresponde las IPs mostradas se podr3 tener la opci3n de imprimir dicho reporte.

Principal

Usuarios

Privilegios


Reportes


12


Administración

Bienvenido : Administrador Proxy

Cerrar sesión

 TRAFICO RED INTERNA

 Imprimir



IP ORIGEN	PUERTO PROXY	BYTES ENV.	IP DESTINO	PUERTO DESTINO	BYTES RECIB.	USUARIO/GRUPO
fe80::216:76ff:fe42:48a3	3128	357218	fe80::f90d:2f8:7134:6e1a	50157	16280	fox
fe80::216:76ff:fe42:48a3	3128	21792	fe80::f90d:2f8:7134:6e1a	50171	666	fox
fe80::216:76ff:fe42:48a3	3128	3688	fe80::f90d:2f8:7134:6e1a	50169	3283	fox
fe80::216:76ff:fe42:48a3	3128	1997	fe80::f90d:2f8:7134:6e1a	50153	3417	fox
fe80::216:76ff:fe42:48a3	3128	1397	fe80::f90d:2f8:7134:6e1a	50168	3052	fox
fe80::216:76ff:fe42:48a3	3128	1042	fe80::f90d:2f8:7134:6e1a	50174	2468	fox
fe80::216:76ff:fe42:48a3	3128	1324	fe80::f90d:2f8:7134:6e1a	50170	2333	fox
fe80::216:76ff:fe42:48a3	3128	1322	fe80::f90d:2f8:7134:6e1a	50162	2274	fox
fe80::216:76ff:fe42:48a3	3128	1043	fe80::f90d:2f8:7134:6e1a	50173	2473	fox
fe80::216:76ff:fe42:48a3	3128	1207	fe80::f90d:2f8:7134:6e1a	50172	810	fox

Done

Figura 3.21 Interfaz Pantalla reporte red interna

### 3.5.9 Pantalla Usuario de navegación

Esta pantalla estará en la opción de administración se podrá crear y eliminar y con la opción de imprimir si es necesario, los campos a mostrar son usuario y la clave se presentaran en asteriscos para interpretar el cifrado.

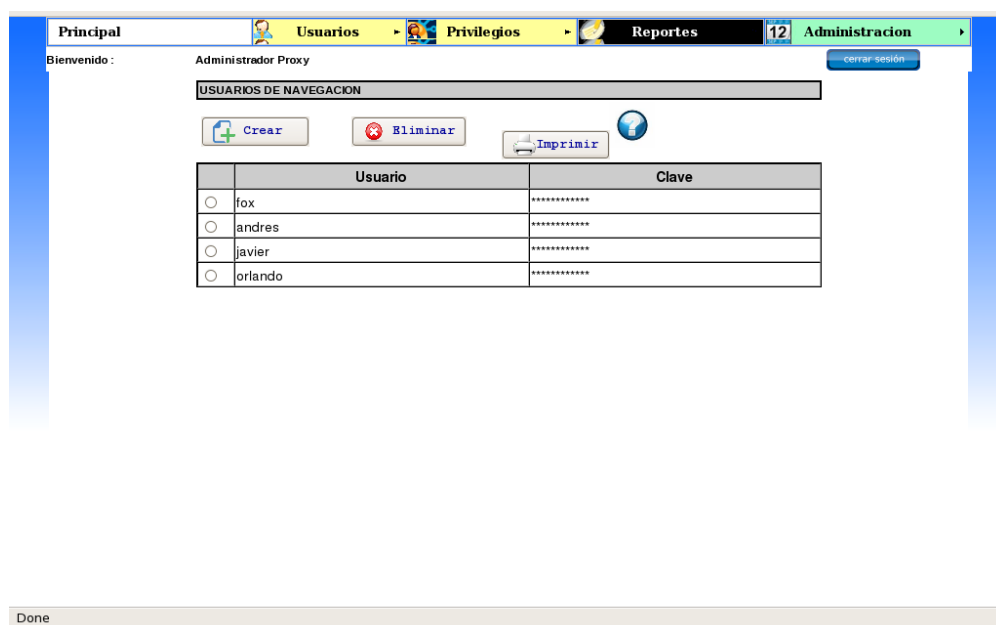


Figura 3.22 Interfaz Pantalla usuarios de navegación

### 3.5.10 Pantalla Regla de Navegación

En esta pantalla se mostrara la consulta de reglas del proxy se podrá crear, modificar y eliminar las reglas.

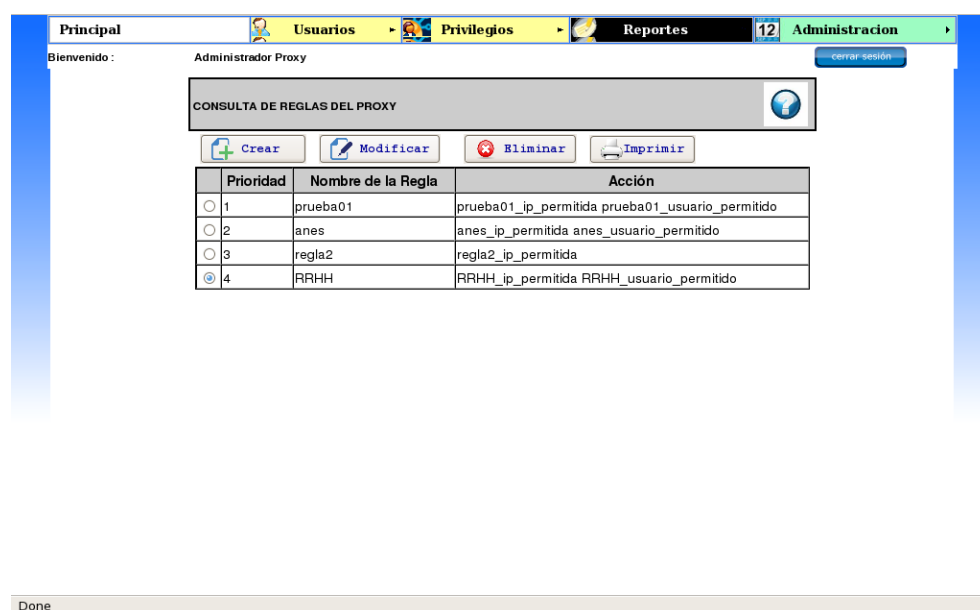


Figura 3.23 Interfaz Pantalla Regla de navegación



### 3.5.11 Pantalla Crear Regla para el proxy-Wizard

En estas pantallas se ingresara el nombre de la regla, las diferentes reglas que se crearan se presentaran como especie de un Wizard paso a paso se podar añadir o quitar permisos o restricciones que se aplicaran a la regla, esta misma terminara en la regla del ancho de banda.

Principal Usuarios Privilegios Reportes 12 Administracion

Bienvenido : Administrador Proxy

CREAR REGLA PARA EL PROXY

Ingrese el nombre de la nueva regla : RRHH

Por favor ingrese el nombre de la regla sin espacio

Guardar y Seguir Consultar Reglas

Principal Usuarios Privilegios Reportes 12 Administracion

Bienvenido : Administrador Proxy

Ingrese direccion IP: [d472:2af9:7fe5:1bab]

fe80::f90d:2f8:7134:6e1a  
fe80::5c9b:4b6e:acd1:1346

Listado de direcciones permitidas para la navegación

☒ Direcciones IP denegadas

Ingrese direccion IP:

fe80::d472:2af9:7fe5:1bab

Listado de direcciones denegadas para la navegación

Guardar y Seguir Guardar y Salir Siguiente Restriccion

Principal Usuarios Privilegios Reportes 12 Administración

Bienvenido : Administrador Proxy Cerrar sesión

Parametros de Configuración

Puerto actual del Proxy 3128 Ingrese el nuevo puerto del Proxy 4444

Tiempo actual de escaneo del jnettop 5 Ingrese el nuevo tiempo de escaneo del jnettop 10

Modificar Datos

Done

Figura 3.24 Interfaz Pantalla Wizard creación de reglas

### 3.6 Flujo de ventanas

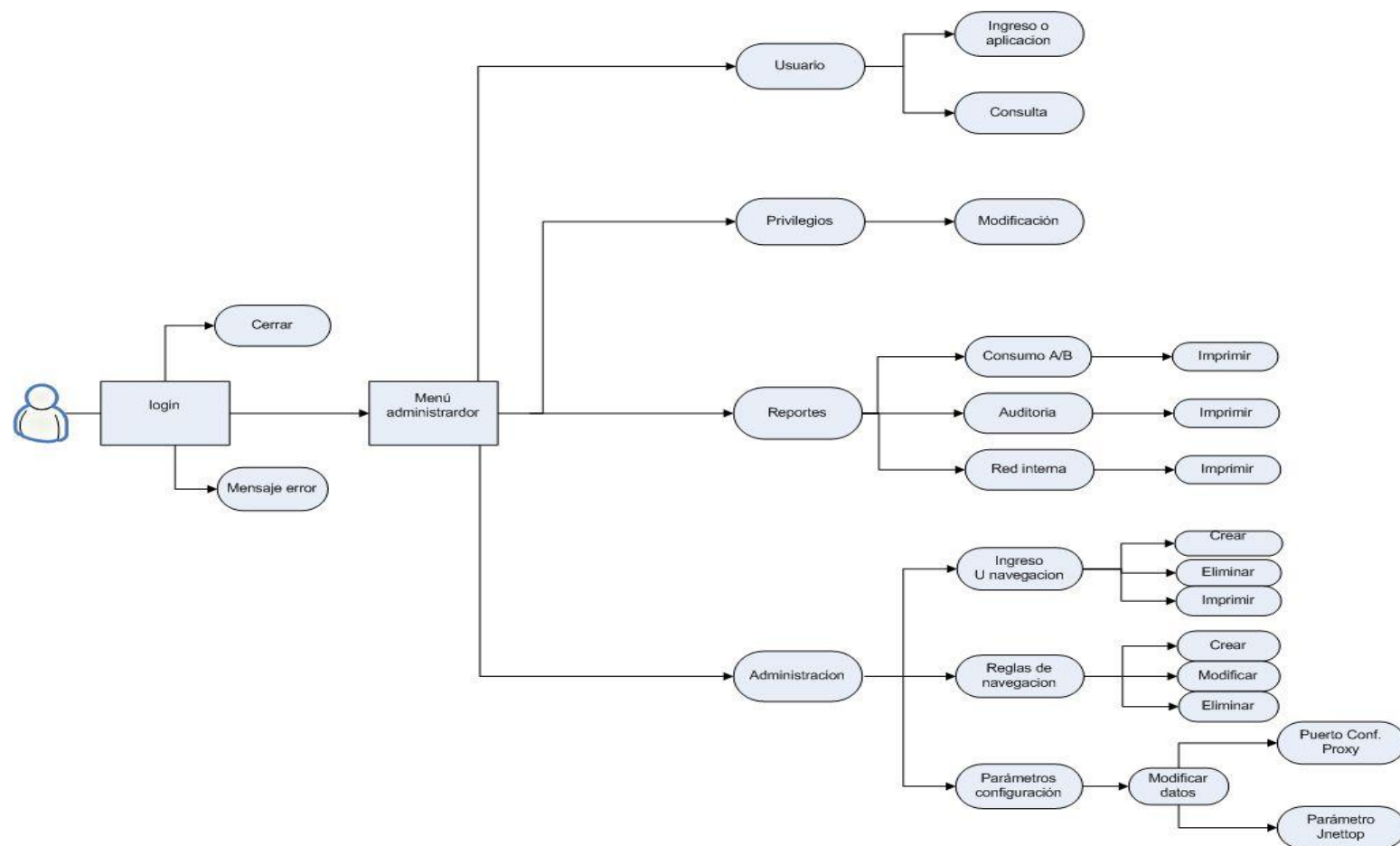


Figura 3.25 diagrama flujo de ventanas



# **CAPITULO 4**

## **DESARROLLO DEL SOFTWARE**

### **4.1. Antecedente**

Para el desarrollo de esta herramienta utilizamos el método de cascada del desarrollo, es decir, dividimos la herramienta por micro módulos dependiendo de la importancia de los mismo y realizando las pruebas pertinentes.

## **4.2 Procesos Principales**

La herramienta que desarrollamos tiene como proceso esencial la administración visual del servidor proxy, creación de usuarios de navegación delimitaciones de ancho de banda, creación de reglas de navegación aplicando restricciones que dependiendo de la necesidades de los usuarios serán establecidas para el correcto aprovechamiento de los recursos de conectividad hacia el internet.

Dicha herramienta tiene como eje principal la manipulación de los archivos de configuración del servidor implementado en la distribución de centos 5.2. el mismo que complementado con la base de datos de My SQL Server 5 brindan una herramienta poderosa. En cuanto al diseño nos hemos ayudado de la herramienta dreamweaver para desarrollar la parte del front- end de la aplicación.

### **4.2.1 Clases**

#### **4.2.1.1. Clase “Conexion”**

Esta clase tiene la función específica de realizar la conexión de la herramienta con la base de datos levantada en la herramienta My SQL SERVER. No solo esta clase brinda la comunicación con la base sino que en ella tambien se gestionan las verificaciones de las peticiones de las consultas, mantenimiento.

#### **4.2.1.2 Clase “Configuracion”**

En esta clase contiene los procesos de modificacion de puertos parametrizables los mismos que eson ingresados por el usuario los mismos que afectaran a los archivos principales de configuracion del servidor.

.

#### **4.2.1.3 Clase “Ejecuta Procesos”**

En esta clase se van a llevar a cabo los procesos de reseteo de los servicios mas importantes de nuestro servidor proxy al momento de realizar cualquier cambio, como el proceso de reseteo del firewall y reseteo del squid además de eso el proceso de ejecución del servicio jnettop el cual nos ayuda a capturar los requerimientos de navegación en datos mas específicos como puerto origen, ip origen , bytes trasmitidos etc.....

### **4.2.2. Formas o Pantallas**

#### **4.2.2.1 “Index.jsp”**

La primera pantalla de la herramienta, esta es la que permite dar o no el acceso al usuario. Esta pantalla utiliza la clase Conexión para realizar el acceso del usuario según la autenticación del mismo, además que carga datos a las diferentes variables globales que utiliza la herramienta y que serán utilizadas para los diferentes procesos.

#### **4.2.2.2 “Principal.jsp”**

Esta pantalla es la que va a contener el menu que se generara luego de validar al usuario y su contrasena dependiendo de las opciones que tenga el usuario de desplegara el menu correspondiente a las acciones habilitadas.

#### **4.2.2.3 “mod\_privilegios.jsp”**

En esta pantalla se ponen a consideracion las opciones de asiganacion de cada unos de los privilegios o permisos que tendran los usuarios de la aplicación , previo a esto logicamente el usuario debio haberse creado y al mismo tiempo haberse definido a que rol pertenecera dicho usuario. Para de esta manera proceder a la selección de las opciones presentadas en esta pantalla

#### **4.2.2.4 “reglas\_proxy.jsp”**

Esta pantalla es la raiz de nuestra aplicación ya que en ella se gestionan todas las posibilidades de creacion de reglas que estaran sujetas a crietrio del usuario q tenga estas opciones disponibles en su cuenta en las mismas podra definir restricciones de paginas, archivos de descarga , frases, puertos, ancho de banda



#### 4.2.2.5 “funciones\_reportes.jsp”

Esta pantalla nos muestra las opciones de reportes tales como:

- Red Interna: en esta opción podemos visualizar el tráfico recurrente por la red mediante datos específicos tales como la ip de origen, puerto de origen, puerto proxy, bytes enviados, ip destino , puerto destino, bytes recibidos, usuario o grupos.
- Auditoria: este reporte nos permite controlar que usuarios ingresaron a la aplicación, hora y fecha.
- Consumo A/B: este reporte nos indica la cantidad de consumo registrado por los usuarios de navegación previamente registrados en el menú administración en la opción que lleva el mismo nombre .  
Además se podrán visualizar estadísticas de navegación por día , mes y año.

#### 4.2.3. Procedimientos Almacenados (SQL SERVER)

Los procedimientos son los encargados de realizar el mantenimiento dentro de la base de datos con los parámetros enviados desde las clases

Los procedimientos almacenados llevan el mismo nombre de las tablas a las que afectan.

#### **4.2.4.1 Procedimiento Almacenado “mantenimiento\_permisos”**

Este procedimiento se encarga de dar mantenimiento a la tabla permisos utilizado con el fin de realizar las operaciones básicas de consulta, ingreso, modificación y eliminación.

#### **4.2.4.2 Procedimiento Almacenado “mantenimiento\_usuarios”**

Este procedimiento se encarga de dar mantenimiento a la tabla usuarios utilizado con el fin de realizar las operaciones básicas de consulta, ingreso, modificación y eliminación.

#### **4.2.4.3 Procedimiento Almacenado “funciones”**

Este procedimiento trae los datos de los usuarios de navegacion los cuales estan asociados a las reglas de navegacion .

#### **4.2.4.4 Procedimiento Almacenado “vistas”**

Este procedimiento almacenado se encarga de traer de la base las sesiones cargadas por los usuarios que usan la aplicación , desde la tabla auditoria.

# **CAPITULO 5**

## **CONCLUSIONES Y RECOMENDACIONES**

### **5.1 Conclusiones**

La Herramienta tiene el fin de realizar la administración de un proxy en ipv6 estableciendo así la comunicación de nuestra red ipv6 con los ISP V4 que actualmente se mantienen en nuestro medio ya que la migración hacia esta

Esta herramienta nos permitirá el correcto aprovechamiento de los recursos de internet estableciendo políticas de navegación para los usuarios que tengan la opción de poder realizarlo , esto nos permitirá poder evaluar y tomar decisiones en cuanto al consumo de recursos por aéreas, departamentos o usuarios de la red, es decir poder medir la eficiencia de la personas .

Con dichas políticas de navegación también se contribuye a la seguridad de nuestra red por que nos permitirá restringir la descarga de probables archivos maliciosos, posibles páginas con contenido sospechoso, etc.... Todo esto apoyado en un servidor estable correctamente configurado para soportar esta nueva generación de ipv6 .

De ésta manera se logra tener una confiabilidad de que cuando llegue la migración de los ISP que están en nuestro entorno ya estar preparados y sin mayores cambios a efectuar en nuestra red .

## **5.2 Recomendaciones**

Esta herramienta esta diseñada para funcionar en cualquier distribución de centos pero siempre y cuando utilice la versión de squid 3.1.0.13 o superior ya que esta es la versión en la que esta soportada la ipv6.

Adicional a esto las maquinas clientes tienen que soportar también ipv6 la cual la podemos encontrar en sistemas operativos Windows XP SP3, Windows Vista , Windows 7.

### **5.3 Bibliografía**

Para la elaboración de nuestra herramienta nos basamos en información recogida de diferentes websites de interés además de entrevistas con persona conocedoras del entorno de redes y administración de centros de computo así como también de conocedores de plataformas LINUX.

#### **WEBSITES:**

[http://www.6sos.org/que\\_es\\_ipv6.php](http://www.6sos.org/que_es_ipv6.php)

<http://www.centos.org/>

<http://dev.mysql.com/tech-resources/articles/dispelling-the-myths.html>

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

<http://es.wikipedia.org/wiki/IPV6>

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

<http://www.ipv6.org/>

<http://java.sun.com>

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

<http://www.netbenas.org>

<http://www.netfilter.org/>

[http://www.nettix.com.pe/instalacion\\_de\\_servidor\\_centos\\_5.2.html](http://www.nettix.com.pe/instalacion_de_servidor_centos_5.2.html)

<http://www.linuxparatodos.net/portal/staticpages/index.php?page=19-0-como-squid-general>

<http://www.squid-cache.org/>

<http://www.pello.info/filez/firewall/iptables.html>

## 5.4 Anexos

### ENTREVISTAS:

*Ing. Jorge Aray de la Cruz, Administración de Redes ESPOLTEL.*

**Tema:** Instalación de Servidores LINUX en IPV6.

**Contenido:** Recomendaciones de Instalación de la distribución Centos 5.2.

*Ing. Carlos Montes, Consultor de Sistemas.*

**Tema:** Transparencia en IPV6, Soporte de Squid, Iptables.

**Contenido:** En esta entrevista fueron expuestas recomendaciones importantes para la consecución del proyecto tales como las validaciones por usuario, el servicio de control de ancho de banda, manipulación de datos del servicio jnettop para la generación de reportes, configuración de firewall.

*Ing. Carlos Alvarado, Administración de Redes.*

**Tema:** Interfaz Gráfica de la Aplicación y recomendaciones.

**Contenido:** La ayuda brindada en esta entrevista sirvió para darnos una idea final de la presentación de nuestro proyecto en la misma también fue definida la forma de crear las reglas de navegación mediante un wizard.

## **INDICE GENERAL**

<b><u>INDICE</u></b>	<b><u>PAGINAS</u></b>
<b>DEDICATORIA.....</b>	<b>I-II</b>
<b>AGRADECIMIENTO.....</b>	<b>III-V</b>
<b>TRIBUNAL DE GRADUACIÒN.....</b>	<b>VI</b>
<b>DECLARACIÒN EXPRESA.....</b>	<b>VII</b>
 <b>CAPITULO 1</b>	
PROXY EN IPV6.....	2
1 Interfaz de la herramienta .....	4
2 Pantalla de inicio.....	5
3 Menú usuarios .....	7
3.1 Creación de usuarios del sistema.....	7
3.2 Consulta de usuarios al sistema.....	9
4 Menús privilegios.....	12
4.1 Modificar rol consultor.....	12
5 Reportes.....	14
5.1 Reporte consumo ancho de banda.....	14
5.1.1 Reporte consumo por cliente.....	15
5.1.2 Reporte consumo total.....	16
5.2 Auditoria de usuarios de la aplicación.....	18
5.3 Reporte red interna de navegación.....	20
6 Administración.....	22
6.1 Usuario de Navegación.....	22
6.1.1 Creación de usuario de navegación.....	24
6.2 Reglas de navegación.....	25
6.2.1 Creando nueva regla.....	29
6.3 Parámetros de configuración.....	35



## CAPITULO 2

Introducción.....	37
Rol Administrador.....	38
Rol Consultor.....	39
Procesos Críticos.....	39
Clases.....	40
Clases “conexión”.....	40
Clases “Auditoria”.....	43
Clases “Configuracion”.....	45
Clases “Ejecutar_Procesos”.....	47
Clases “Funciones”.....	51
Clases “Objetos”.....	53
Clases “Permisos”.....	81
Clases “Proceso_Capturador”.....	84
Clases “Usuarios”.....	86

## FIGURAS

Fig. 1.1 Acceso al sistema.....	5
Fig. 1.2 Pantalla menú principal.....	5
Fig. 1.3 Pantalla menú Usuarios.....	6
Fig. 1.4 Pantalla creación de usuarios del sistema.....	7
Fig. 1.5 Pantalla consulta de usuarios del sistema.....	9
Fig. 1.6 Pantalla menú Privilegios.....	11
Fig. 1.7 Pantalla modificar privilegios del consultor.....	12
Fig. 1.8 Pantalla menú reporte.....	13
Fig. 1.9 Pantalla de selección de reporte de A/B .....	14
Fig. 1.10 Pantalla reporte consumo por cliente.....	15
Fig. 1.11 Pantalla reporte consumo total (1).....	17
Fig. 1.12 Pantalla reporte consumo total (2).....	17
Fig. 1.13 Pantalla reporte auditoria.....	19
Fig. 1.14 Pantalla red interna.....	21
Fig. 1.15 Pantalla menú administración.....	22
Fig. 1.16 Pantalla administración.....	23
Fig. 1.17 Pantalla creación de usuarios de navegación.....	25
Fig. 1.18 Pantalla menú reglas de navegación.....	27
Fig. 1.19 Pantalla consulta de reglas del Proxy.....	27
Fig. 1.20 Pantalla creación de la regla.....	29
Fig. 1.21 Pantalla creación de la regla Proxy – IP permitidas.....	30
Fig. 1.22 Pantalla crear regla Proxy –IP restringidas.....	30
Fig. 1.23 Pantalla crear regla Proxy – puertos.....	31
Fig. 1.24 Pantalla crear regla Proxy – usuarios.....	31
Fig. 1.25 Pantalla crear regla Proxy - dominios.....	32

Fig. 1.27 Pantalla crear regla Proxy – tipos de archivos.....	33
Fig. 1.28 Pantalla crear regla Proxy – ancho de banda.....	33
Fig. 1.29 Pantalla crear regla Proxy – Reiniciar servicios .....	34
Fig. 1.30 Pantalla crear regla Proxy – Final.....	35
Fig. 1.31 Pantalla parámetros de configuración.....	36

## **ANEXOS**

### **ANEXO 1**

INSTALACION CENTOS 5.2 Y CONFIGURACION.....	91
---	----

### **ANEXO 2**

PASOS PARA CONFIGURAR AUTOMATICAMENTE PROXY EN EL EXPLORADOR.....	118
--	-----

# **CAPITULO 1**

## **MANUAL DE USUARIO**

### **PROXY EN IPV6**

El proyecto está dedicado al desarrollo de un servidor Proxy en IPv6, en el cual se puede permitir o restringir la navegación en internet en función de varios tipos de acceso, como restricciones de:

- Palabras
- IP's
- Frases
- Puertos
- Dominios
- Tipos de archivos.

Este proyecto fue creado usando la herramienta de desarrollo Netbeans 6.5 y utilizando como base de datos MySQL versión 5, además está enfocado a operar en una red interna en IPV6, todos los usuarios que necesiten navegar deben estar registrados con el administrador de seguridades, quien será la persona que otorgará y negará accesos al usuario de navegación, esto se pondrá en práctica en cualquier organización que promulgue el buen uso y manejo del servicio web.

En la aplicación existe un usuario por defecto, el “administrador”, el cual es denominado “Administrador del Proxy” quien efectuará las configuraciones iniciales de parámetros y reglas.

Cuando el usuario ingresa por primera vez a la herramienta, este tendrá la opción de crear dos tipos de roles existentes:

- Rol Administrador
- Rol Consultor

Cada rol tiene funciones específicas y sus características son las siguientes:

- Rol Administrador.- Al ingresar con este tipo de rol tendrá todos los permisos de la aplicación, es decir, podrá tener acceso a todas las opciones del menú, así como los privilegios para crear otros usuarios y sus restricciones de acceso en la aplicación, ejecutar diferentes reportes como los de acceso de usuarios, reporte de gráficas de consumo de ancho de banda y el reporte de red interna de navegación con los nombres de los grupos y usuarios.
- El Consultor.- Al ingresar con este tipo de rol estará en dependencia de la configuración que haya establecido el

administrador, solo accederá a la ampliación de forma restringida según sea el caso de los privilegios otorgados para el rol consultor, en algunos casos si el administrador lo permitiese le podría otorgar el manejo de toda la aplicación, esto puede suceder el caso de ausentismo de un tiempo prolongado del administrador y concede permisos al rol de confianza.

## **1. Interfaz de la herramienta**

La interfaz define el contenido de las diferentes pantallas de la aplicación, las cuales se detallarán seguidamente, así también todos los pasos y procedimientos a seguir para el correcto manejo de la herramienta.

### **El administrador de la herramienta Proxy**

- El tendrá la responsabilidad de crear los usuarios de la aplicación y de otorgar los permisos necesarios al rol consultor para el uso de determinadas funciones.
- Sera responsable de la emisión de reportes solicitados por la gerencia de la organización respecto de los ingresos y salidas de la aplicación de usuarios administradores y consultores,

emitir reportes de consumo de recursos de internet en bytes, y navegación de los usuarios web.

- Tendrá la responsabilidad de realizar el mantenimiento de usuarios de navegación, tanto para nuevos usuarios como también para modificar los existentes y de la eliminación según sea necesario.
- Creará y manipulará el proxy añadiendo reglas con sus respectivas restricciones de acceso para cada usuario, podrá formar reglas para grupos o restringir usuarios individualmente.

## **2. Pantalla de inicio**

En esta pantalla se digita el nombre de usuario que va a entrar a la aplicación y su respectiva contraseña de acceso (figura 1.1), para poder tener acceso a las funciones de la herramienta administradora del proxy, los datos del usuario y contraseña digitados deberán ser validados y estar registrados en la base de datos MySQL, los datos de usuario y contraseña digitados anteriormente serán comparados y validados con los datos almacenados en la base, usando el algoritmo de encriptación MD5, de no ser compatibles con los datos almacenados, el programa presentará un mensaje de alerta que indicará que el usuario no es el correcto y tendrá que volver a ingresar los datos correctos para poder acceder de forma correcta a la aplicación.





**Fig. 1.1 Acceso al Sistema**

Después de validados y aceptados los datos de usuario y contraseña, se mostrará la pantalla principal con todos los menús y submenús correspondientes a cada opción (figura 1.2).



**Fig. 1.2 Pantalla Menú principal**

### 3. Menú Usuarios

Al pasar el puntero del mouse sobre el menú **USUARIOS** se desplazará un submenú que contiene las opciones de **Creación** y **Consultar** (figura 1.3).



**Fig. 1.3 Pantalla Menú Usuarios**

A continuación se procederá a describir el submenú.

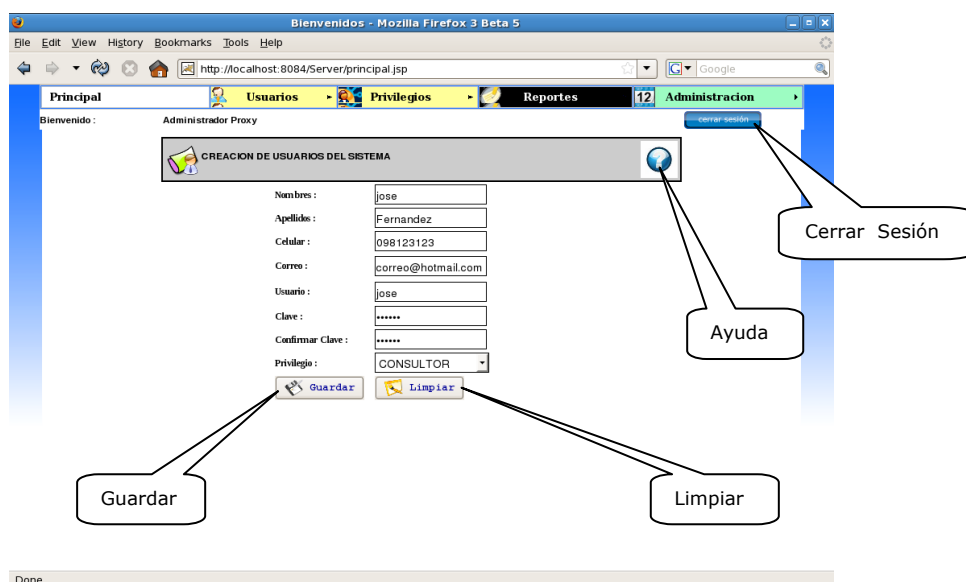
#### 3.1 Creación de Usuarios del Sistema

En esta pantalla (figura 1.4) se presenta como encabezado el nombre de la función que se está ejecutando, en la cual debemos ingresar por teclado todos los datos solicitados que son obligatorios:

- Nombres

- Apellidos
- Cédula
- Correo
- Usuario
- Clave
- Confirmar clave
- Privilegio (Consultor / Administrador)

**Nota:** Es de suma importancia que al llenar todos los datos de usuario se preste atención a la opción **Privilegio**, ya que este campo desplazará una lista con las dos únicas opciones de rol por asignar al usuario en proceso de creación: **ADMINISTRADOR** y **CONSULTOR**.



**Fig. 1.4** Pantalla Creación de Usuarios del sistema

Función de botones:



: Guardará los datos en la base del sistema.



: Limpiará los datos escritos previamente.



: Cierra la sesión del usuario actual.

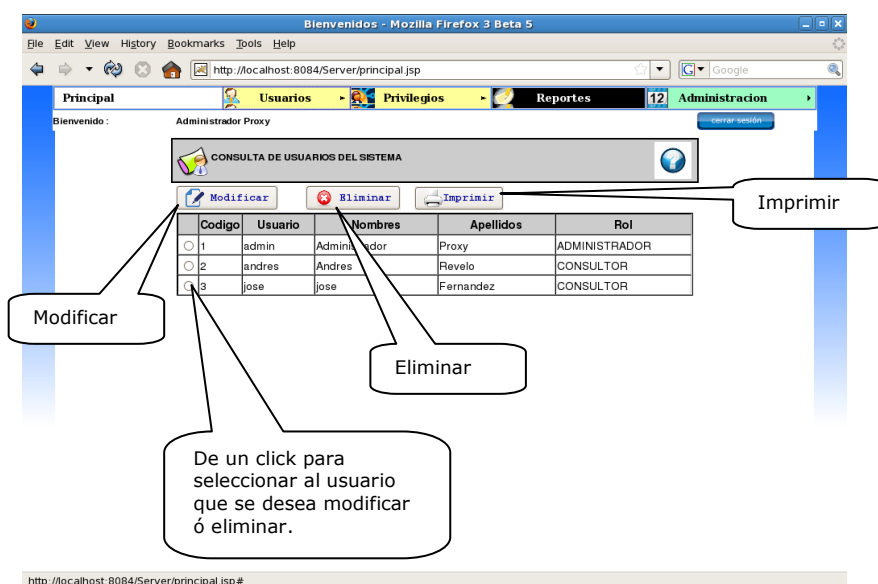


: Desplazará un cuadro de texto con ayuda sobre la pantalla actual.

Después de guardar los datos correctamente en la base de datos, el sistema lo enviará a la pantalla **Consulta de usuarios del sistema** la misma que se detalla a continuación.

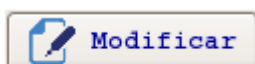
### 3.2 Consulta de Usuarios del sistema

En esta pantalla (figura 1.5) se muestran todos los usuarios de la aplicación que previamente hayan sido creados, los cuales se presentarán de forma ordenada según su asignación secuencial en la base de datos, es decir, numerados, de presentarse la necesidad de eliminar a algún usuario del sistema desde esta pantalla, se debe seleccionar el código secuencial asignado, y hacer un click en el botón **Eliminar**, el cual ejecutará el proceso de eliminación, el usuario será borrado y su código no será vuelto a presentar, ya que el sistema otorga un código único para cada usuario.

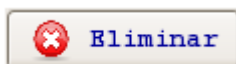


**Fig. 1.5 Pantalla Consulta de Usuarios del sistema**

Función de botones:

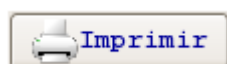


: Seleccione haciendo un click en el botón de selección correspondiente al usuario que desea modificar y luego presione este botón, lo llevará a la pantalla modificar y le presentará los datos actuales del usuario seleccionado, se reemplazarán dichos datos con los cambios que se consideren necesarios, se procederá a guardar los nuevos datos en la base de datos haciendo click en el botón Guardar.



: Seleccione haciendo un click en el botón de selección correspondiente al usuario que desea eliminar y luego presione este botón, aparecerá un mensaje para confirmar la acción,

si acepta el mensaje ejecutará el proceso de eliminación, el usuario será borrado y su código no será vuelto a presentar, ya que el sistema otorga un código único para cada usuario.



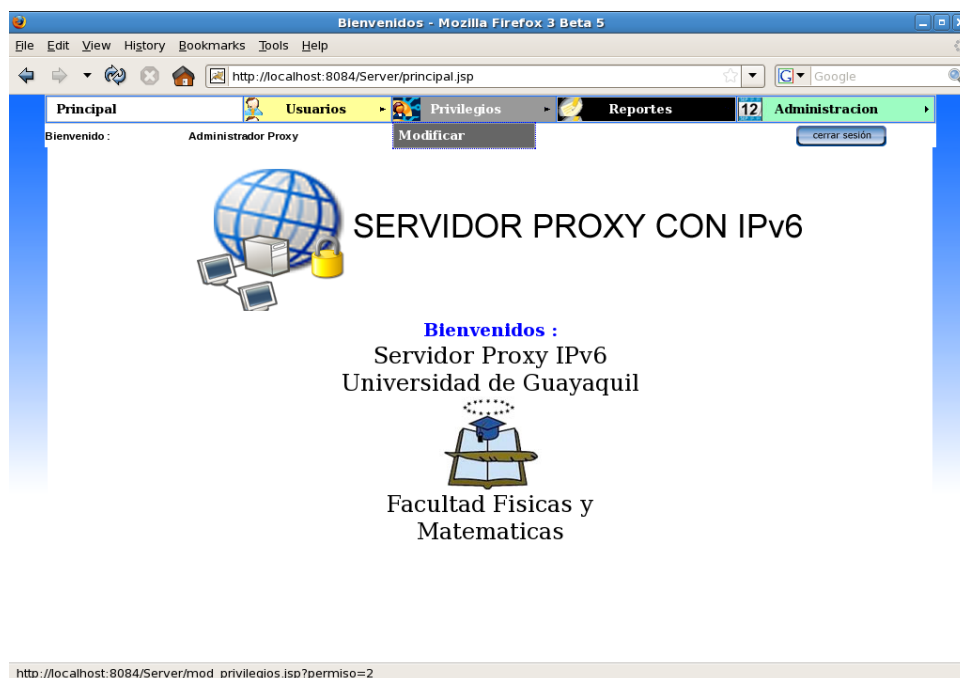
: Presione este botón en caso de requerir imprimir la información en pantalla, se desplegará el asistente de impresión y usted debe seleccionar la impresora a la cual desea enviar el documento, de lo contrario se direccionará a la impresora predeterminada del sistema operativo.



: Desplazará un cuadro de texto con ayuda sobre la pantalla actual.

## 4. Menú Privilegios

Para activar el menú privilegios se procede a pasar el puntero del mouse sobre él, lo cual desplazará el submenú **modificar** (figura 1.6).

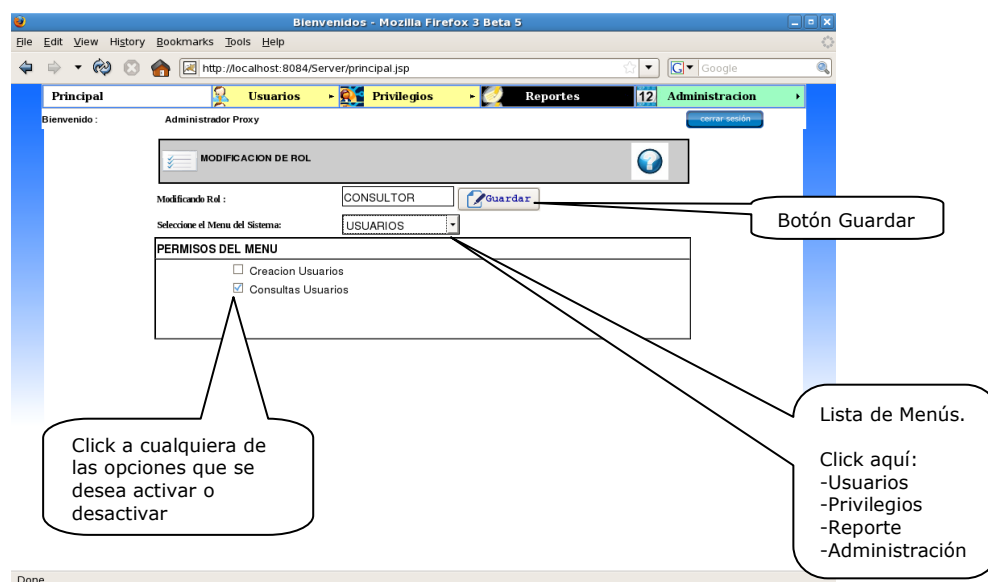


**Fig. 1.6 Pantalla Menú Privilegios**

### 4.1 Modificar Rol Consultor

Hacemos un click para que se muestre la pantalla de modificación de rol (figura 1.7), es importante tomar en cuenta que el único rol al cual se podrán realizar modificaciones es al **CONSULTOR**, por lo tanto, en la etiqueta que contiene **Modificando Rol** podremos observar tan solo al rol consultor, y en la etiqueta **Seleccione el menú del sistema** se desplazará una lista que contiene los nombres de los menús de la aplicación.

Para modificar los privilegios del rol seleccionamos haciendo un click en cada uno de los menús de la lista. Y en la sección de **PERMISOS DEL MENU** encontraremos el contenido de cada menú principal, es decir, sub menú, representados en casilleros de opción con lo cual al hacer click en cada uno de ellos, según las restricciones que el administrador imponga se otorgarán o restringirán los permisos que el rol requiera, los cambios surtirán efecto solo cuando hagamos un click en el botón GUARDAR.



**Fig. 1.7 Pantalla modificar privilegios del consultor**

Función de botones:



: Cuando se procede a modificar ya sea añadiendo más permisos ó quitando alguno de ellos, los cambios realizados solo surtirán efecto después de hacer click en este botón y con la posterior confirmación de la acción respectiva.



## 5. Reportes

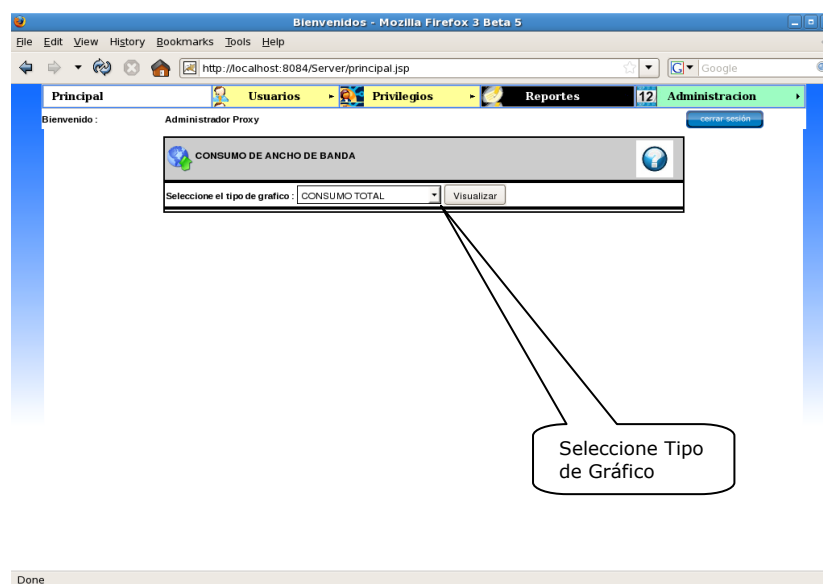
El menú de reportes contiene tres submenús, cada uno de ellos y sus respectivas funciones se describirán paso a paso para aprovechar sus bondades posteriormente (figura 1.8).



**Fig. 1.8 Pantalla Menú reporte**

### 5.1 Reporte Consumo de Ancho de Banda (A/B)

En el primer submenú de Reportes, tenemos los reportes de consumo de ancho de banda, en la etiqueta **Seleccione el tipo de de grafico** nos presentará mediante una lista dos opciones de reportes, que son: **CONSUMO TOTAL y CONSUMO POR CLIENTE** (figura 1.9), al seleccionar cualquiera de ellos debemos presionar el botón **“Visualizar”** haciendo un click en este para que genere el gráfico requerido para análisis.



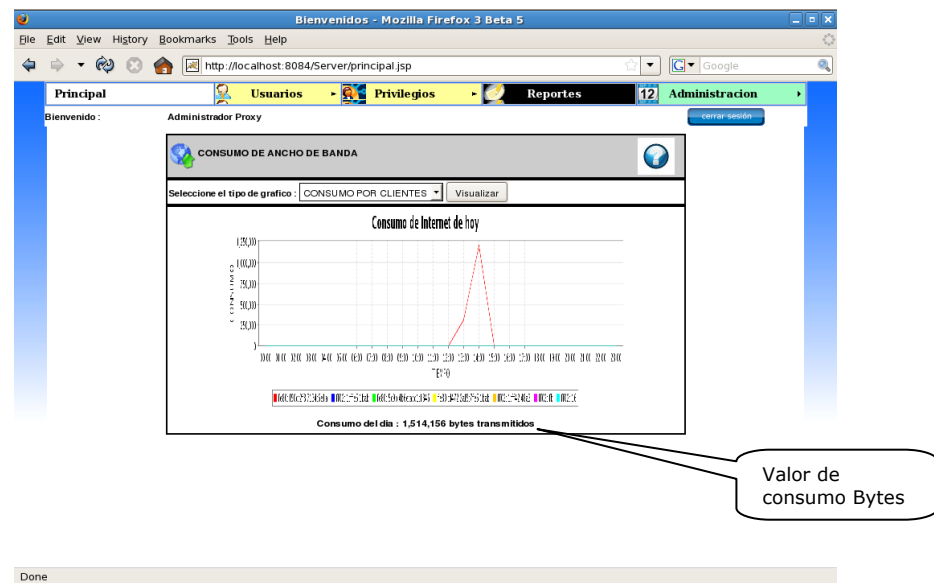
**Fig. 1.9 Pantalla Selección de Reportes de A/B**

### 5.1.1 Reporte Consumo por Cliente

El gráfico de **CONSUMO POR CLIENTES** (fig. 1.10) muestra los bytes consumidos en el día por cada una de las IP's registradas en la aplicación y en que rango de horas sucedió este consumo, datos que sirven de referencia directa para el administrador de la red.

Detalles del gráfico:

El eje horizontal es representado por los valores de tiempo en horas y el eje vertical es representado por los valores de consumo en bytes, en la sección inferior se visualiza la leyenda en colores que representan a cada IP, y a continuación de forma numérica se muestra el valor total de los bytes consumidos por todos los usuarios de la red interna.



**Fig. 1.10 Pantalla reporte Consumo por cliente**

### 5.1.2 Reporte Consumo Total

Al seleccionar el tipo de gráfico CONSUMO TOTAL desde la pantalla de selección de reportes de A/B (figura 1.9) y al hacer un click en el botón **“Visualizar”** se generan tres gráficos:

- Consumo de bytes totales del día, tiempo establecido en horas.  
(figura 1.11).
- Consumo de bytes totales en el mes corriente, tiempo en semanas y
- Consumo de Bytes totales en el año en curso, tiempo en quincenas  
(figura 1.12).

Bienvenidos - Mozilla Firefox 3 Beta 5

File Edit View History Bookmarks Tools Help

http://localhost:8084/Server/principal.jsp

Principal Usuarios Privilegios Reportes 12 Administracion

Bienvenido : Administrador Proxy

Imprimir

Usuario	Nombre de Sesión	Fecha de Ingreso	Fecha de Salida
admin	F3D5F188B008A0BCB9BD52BCB5BF8906	2010-01-26 15:02:36.0	
admin	8A883F25B25C1F92122703178C1F33EE	2010-01-26 15:08:10.0	
admin	94F5D481E9C5315CD695335E47ECD15E	2010-01-26 15:09:07.0	
admin	419E744271EED513F2F2B3533170579E	2010-01-26 15:14:10.0	
admin	49A798FCDEFAD92BADA3DE176984C8C	2010-01-26 15:20:53.0	2010-01-26 15:21:56
admin	DCB9D2A15B270F4EC7624FA12DB597FE	2010-01-26 15:22:12.0	2010-01-26 15:22:34
admin	938A9E65BC4D7F05C312DA41D2103ED0	2010-01-26 15:22:46.0	
admin	B7FCFA81EB22C0C43D5EDCB994E775E0	2010-01-26 15:40:45.0	
admin	16B57653A448A6B1D4F5C860FB920578	2010-01-26 15:46:34.0	

Done

Fig. 1.11 Pantalla reporte Consumo Total (1)

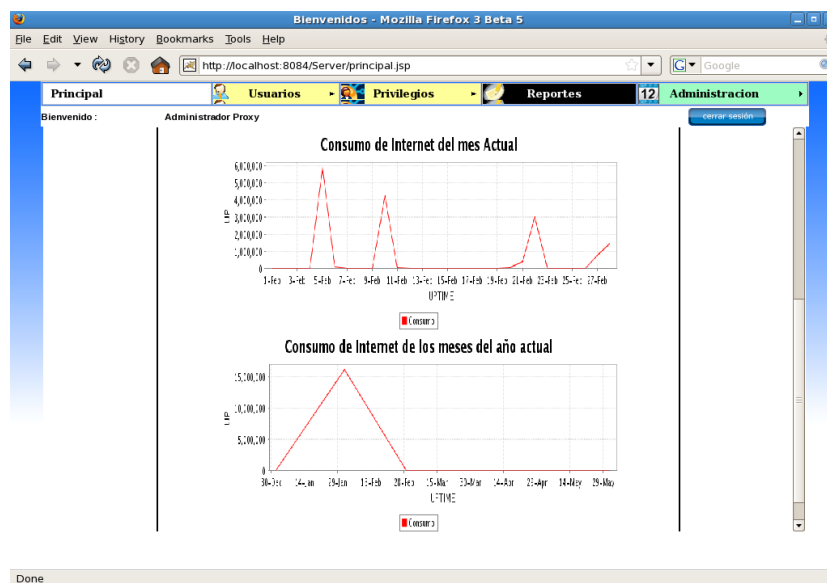
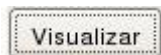


Fig. 1.12 Pantalla reporte Consumo Total (2)

Función de botones:



: Al hacer click aquí se presentará el tipo de gráfico seleccionado desde la lista, ya sea el reporte **CONSUMO POR CLIENTES** ó **CONSUMO TOTAL** expresados en bytes.

## 5.2 Auditoría de usuarios de la aplicación.

En este reporte se genera el registro de acceso de usuarios a la aplicación (figura 1.13), compuesto por cuatro campos de datos:

- Usuario
- Nombre de Sesión
- Fecha de Ingreso
- Fecha de Salida

**Usuario:** Muestra el nombre del usuario que accedió al sistema

**Nombre de sesión:** Se muestra el código de acceso único que le otorga el sistema al usuario que ingresó

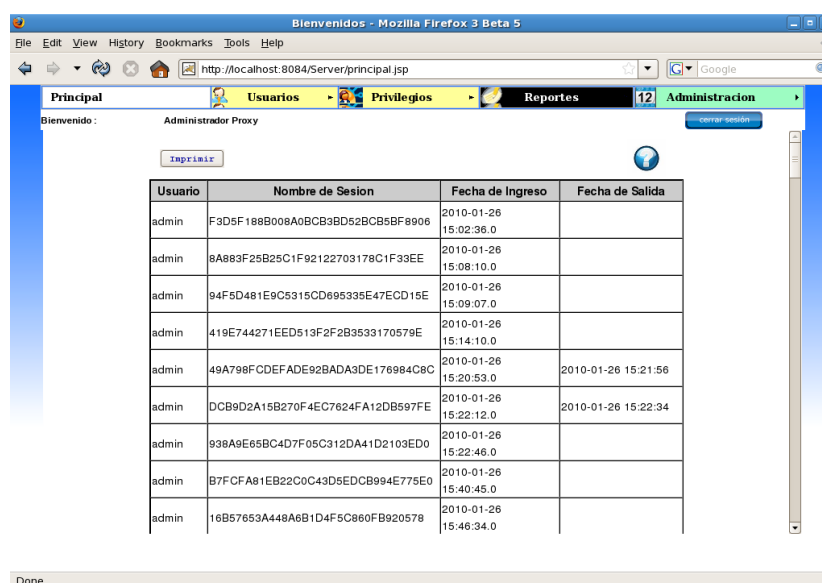
**Fecha de ingreso:** Muestra la fecha y la hora de que el usuario accedió

**Fecha de salida:** Muestra la fecha y la hora de que el usuario cerró la sesión.

**Nota:** Es obligatorio que los usuarios que dejen de usar la aplicación en un momento determinado, luego de realizadas las gestiones dentro de la misma,

salgan de la aplicación haciendo un click en el botón cerrar sesión para que se realice el registro de salida del usuario.

Si el usuario cierra la aplicación de forma incorrecta no se registra el dato de salida de la aplicación.



Usuario	Nombre de Sesión	Fecha de Ingreso	Fecha de Salida
admin	F3D5F188B008A0BCB3BD52BCB5BF8906	2010-01-26 15:02:36.0	
admin	8A883F25B25C1F92122703178C1F33EE	2010-01-26 15:08:10.0	
admin	94F5D481E9C5315CD695335E47ECD15E	2010-01-26 15:09:07.0	
admin	419E744271EED513F2F2B3533170579E	2010-01-26 15:14:10.0	
admin	49A798FCDEFAD92BADA3DE176984C8C	2010-01-26 15:20:53.0	2010-01-26 15:21:56
admin	DCB9D2A15B270F4EC7624FA12DB597FE	2010-01-26 15:22:12.0	2010-01-26 15:22:34
admin	938A9E65BC4D7F05C312DA41D2103ED0	2010-01-26 15:22:46.0	
admin	B7FCFA81EB22C0C43D5EDCB994E775E0	2010-01-26 15:40:45.0	
admin	16B57653A448A6B1D4F5C860FB920578	2010-01-26 15:46:34.0	

**Fig.1.13 Pantalla reporte Auditoría**

Función de botones:



: Al hacer click a este botón se imprimirá el reporte presentado en pantalla en su totalidad con la impresora predeterminada del sistema operativo ó seleccionando la impresora destino para el reporte.

### 5.3 Reporte Red interna de navegación.

En esta pantalla de reporte se presentará el tráfico de paquetes en la red interna (figura 1.14), compuesto por los siguientes 7 campos:

- IP ORIGEN
- PUERTO PROXY
- BYTES ENVIADOS
- IP DESTINO
- PUERTO DESTINO
- BYTES RECIBIDOS
- USUARIO / GRUPO

**IP origen:** Es la IP del servidor en versión IPV6 de la red interna.

**Puerto Proxy:** Este es el puerto del servidor específicamente del proxy toda petición de cualquier usuario será escuchado por este puerto.

**Bytes enviados:** Es la cantidad de bytes enviados del servidor hacia el usuario de navegación.

**IP destino:** Esta son las IP's de los usuarios de navegación de igual manera todas estas IP's son en versión IPV6 y pertenece a la red interna

**Puerto destino:** Este es el puerto del usuario que hizo la petición al proxy esto puede variar el numero de puerto según la petición.

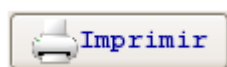
**Bytes recibidos:** Son los bytes recibidos del usuario de navegación

**Usuario /Grupo:** Aquí se coloca el nombre del usuario o los usuarios que están navegando o en el caso de nombre de grupo que es un conjunto de usuarios.

IP ORIGEN	PUERTO PROXY	BYTES ENV.	IP DESTINO	PUERTO DESTINO	BYTES RECIB.	USUARIO/GRUPO
fe80::216:76ff:fe42:48a3	3128	357218	fe80::f90d:2f8:7134:6e1a	50157	16280	fox
fe80::216:76ff:fe42:48a3	3128	21792	fe80::f90d:2f8:7134:6e1a	50171	666	fox
fe80::216:76ff:fe42:48a3	3128	3688	fe80::f90d:2f8:7134:6e1a	50169	3283	fox
fe80::216:76ff:fe42:48a3	3128	1997	fe80::f90d:2f8:7134:6e1a	50153	3417	fox
fe80::216:76ff:fe42:48a3	3128	1397	fe80::f90d:2f8:7134:6e1a	50168	3052	fox
fe80::216:76ff:fe42:48a3	3128	1042	fe80::f90d:2f8:7134:6e1a	50174	2468	fox
fe80::216:76ff:fe42:48a3	3128	1324	fe80::f90d:2f8:7134:6e1a	50170	2333	fox
fe80::216:76ff:fe42:48a3	3128	1322	fe80::f90d:2f8:7134:6e1a	50162	2274	fox
fe80::216:76ff:fe42:48a3	3128	1043	fe80::f90d:2f8:7134:6e1a	50173	2473	fox
fe80::216:76ff:fe42:48a3	3128	1207	fe80::f90d:2f8:7134:6e1a	50172	810	fox

**Fig. 1.14 Pantalla red interna**

Función de botones:



: Al hacer click a este botón se imprimirá el reporte presentado en pantalla en su totalidad con la impresora predeterminada del sistema operativo ó seleccionando la impresora destino para el reporte.



## 6. Administración

En la pantalla del menú Administración se presentan los siguientes tres submenús (figura 1.15):

- Usuarios de navegación
- Reglas de navegación y
- Parámetros de configuración.

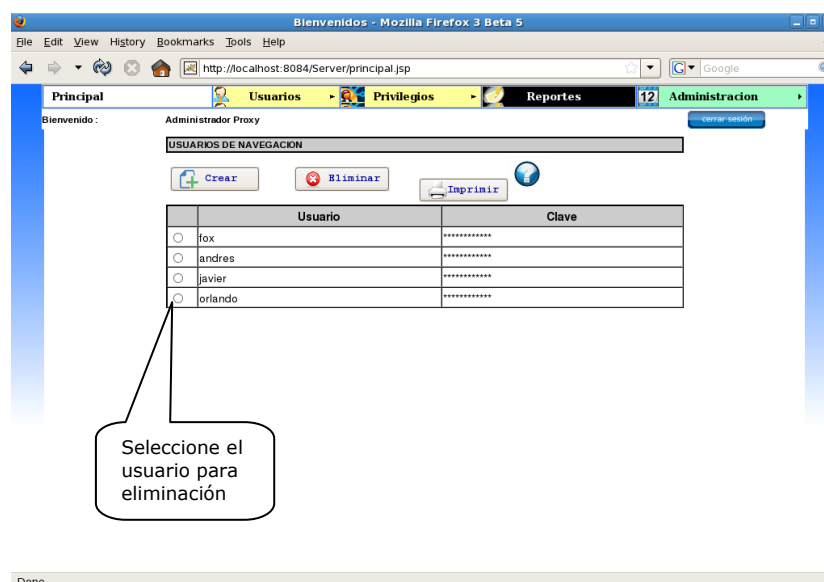


**Fig. 1.15 Pantalla Menú Administración**

### 6.1 Usuarios de navegación

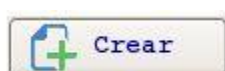
Al ejecutar el sub menú Usuarios de navegación (figura 1.16), este le permitirá al administrador de la red dar acceso de navegación a usuarios de la red interna, asignando a cada uno, ó según sea el caso, un usuario y contraseña para que les sea posible navegar en internet.

Las funciones incluidas en esta opción permitirán al Administrador de la aplicación crear y eliminar usuarios, e incluso enviarlos como reporte a la impresora predeterminada del sistema, mediante el botón **“Imprimir”**.

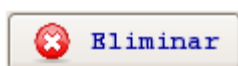


**Fig. 1.16 Pantalla Administración**

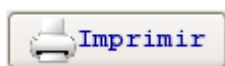
Función de botones:



: Con este botón se podrá crear un usuario nuevo de navegación.



: Previamente seleccionado un usuario de navegación, haciendo un click en este botón se iniciará el proceso de eliminación, el cual después de confirmados los anuncios de advertencia definitivamente borrará el registro del usuario seleccionado anteriormente.

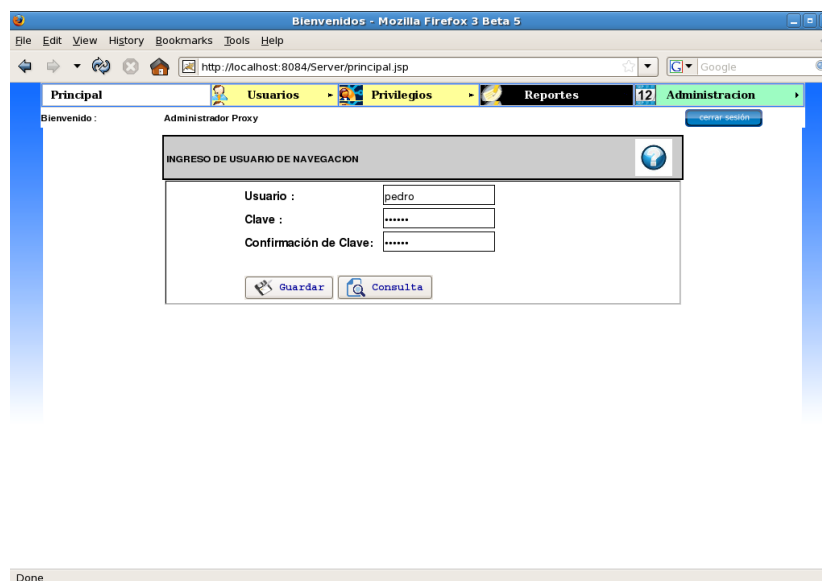


: Al hacer click a este botón se imprimirá el reporte presentado en pantalla en su totalidad con la impresora predeterminada del sistema operativo ó seleccionando la impresora destino para el reporte.

### 6.1.1 Creación de Usuario de Navegación

En esta pantalla (figura 1.17) se podrá crear usuarios de navegación, solo se llenarán los campos visibles que son el nombre del usuario a crear, la *clave* única del para el usuario y la *confirmación* de la misma, estos usuarios son los que se tomarán en cuenta para la creación de reglas de navegación, las cuales son manipuladas por el administrador ó consultor según sea el caso, dichos usuarios podrán formar parte de un grupo.

Usuarios que serán visibles en los reportes anteriormente descritos.



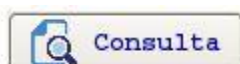
**Fig. 1.17 Pantalla creación de usuario de navegación**

Función de botones:



**Guardar**

: Se guardarán los datos para la creación del nuevo usuario de navegación .



**Consulta**

: Dar un click a este botón para regresar a la pantalla de consulta de usuarios de navegación.

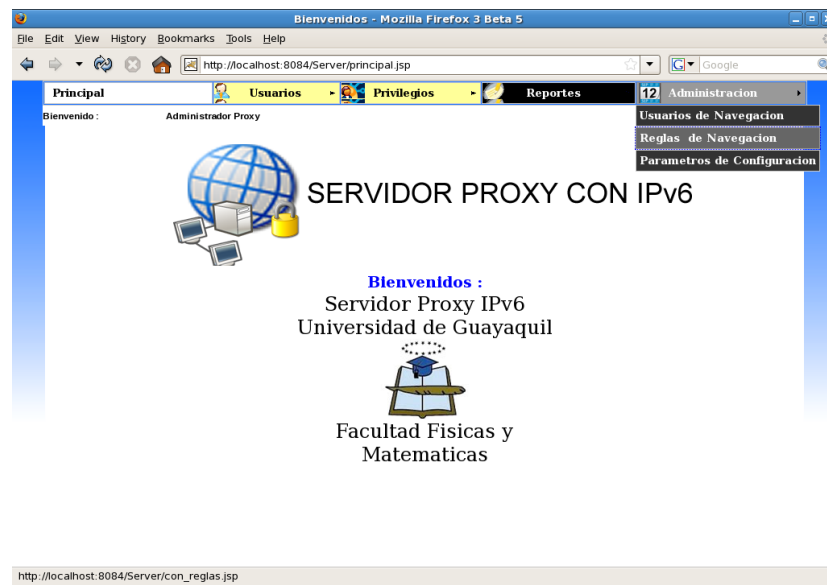
## 6.2 Reglas de navegación.

Esta es la herramienta más importante de la aplicación (figura 1.18) y será la más utilizada por el administrador ó consultor, según sea el caso, aquí se crearán, eliminarán y modificarán las reglas del proxy (figura 1.19), así como

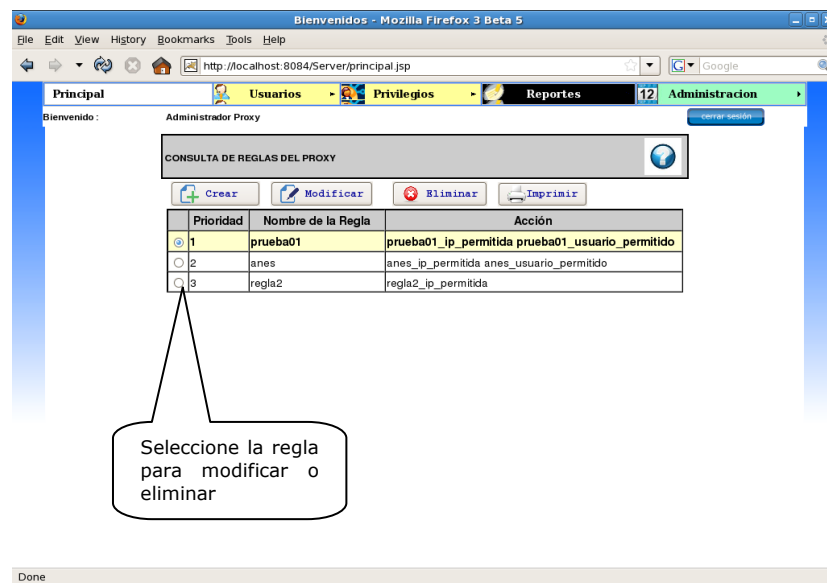
dar permisos y restricciones a los usuarios según las políticas establecidas en la organización para el cumplimiento del buen manejo de los recursos de Internet para la red interna en IPV6. Cabe recalcar que un requisito indispensable e irremplazable para la creación de las reglas de navegación en el proxy es contar con al menos una dirección IP, además esta mencionar que esta debe estar expresada en versión 6 para poder continuar con los siguientes pasos en la creación de las reglas.

La herramienta de creación de reglas de navegación está diseñada en un formato estilo **Asistente** ó **Wizard**, de modo que el usuario Administrador pueda prestar la debida atención a cada paso que debe seguir para concluir con éxito la creación de una regla.

Entre los pasos que contiene el asistente de creación de reglas podemos mencionar añadir *usuarios*, permitir/restringir *puertos*, permitir/restringir *dominios*, permitir/restringir *frases*, permitir/restringir *tipos de archivos*, y así mismo segmentar el ancho de banda de la red para dicha regla tomando en consideración el ancho de banda general de la red, en caso de no manipular la segmentación de ancho de banda, el sistema asignará de forma dinámica para todas las reglas creadas el ancho de banda que sea necesario.



**Fig. 1.18 Pantalla Menú Reglas de Navegación**

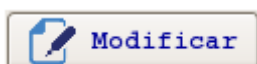


**Fig. 1.19 Pantalla Consulta de Reglas del Proxy**

### Función de botones:



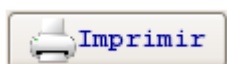
: Al darle click a este botón se inicia un servicio de la aplicación que contempla un asistente de creación de nuevas reglas (figura 1.20).



: Antes de hacer click en este botón previamente se debe de seleccionar la regla para la modificación, se presentará la pantalla de la regla ya creada con los datos antes ingresados para su respectiva modificación.



: De igual manera antes de hacer click en este botón previamente se debe seleccionar la regla que se ha decidido borrar, se presentarán los mensajes de confirmación para la completa eliminación de la regla.

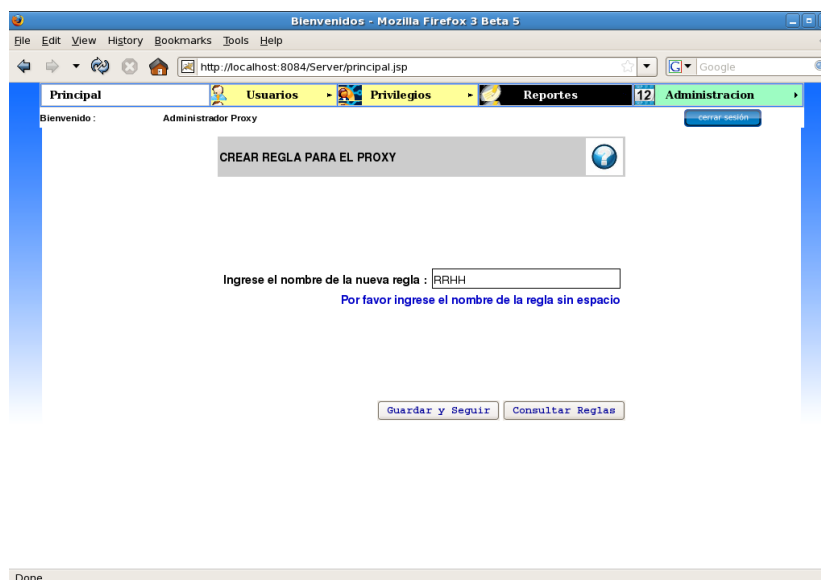


: Al hacer click a este botón se imprimirá el reporte presentado en pantalla en su totalidad con la impresora predeterminada del sistema operativo ó seleccionando la impresora destino para el reporte.

### 6.2.1 Creando nueva Regla.

Mostraremos de forma gráfica paso a paso como crear una regla desde el inicio hasta el final de la misma.

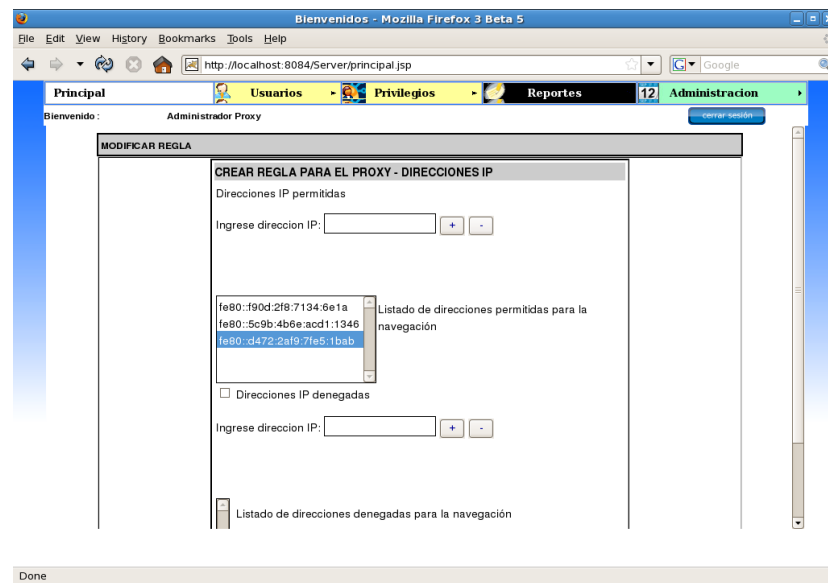
Se ingresará por teclado un nombre a la regla (figura 1.20) según el objetivo de la misma y se procesará con el ingreso de la dirección IPv6 para la cual será asignada la misma regla, para continuar hacemos click en el botón **Guardar y seguir**, de esta manera ya daremos el primer paso para la creación de la nueva regla de navegación.



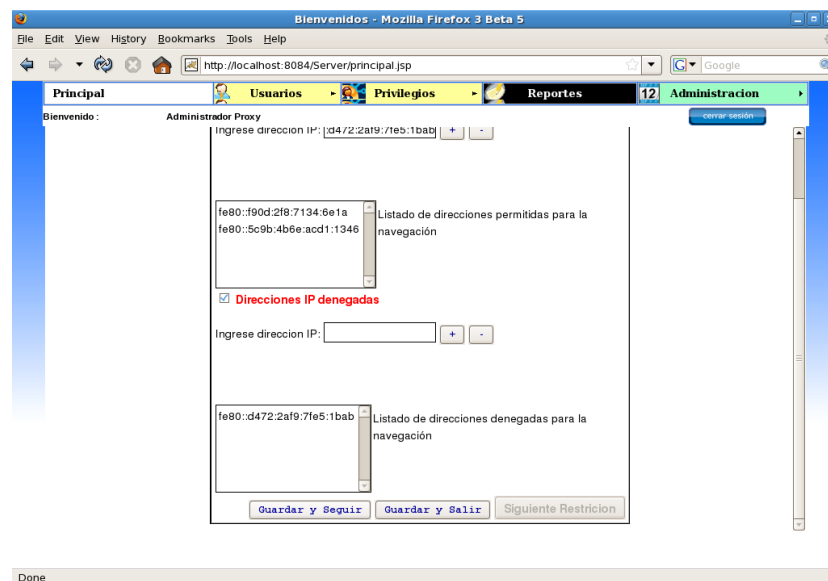
**Fig. 1.20 Pantalla creación de Regla**

A continuación en la siguiente pantalla (figura 1.21) incluiremos las direcciones IP que serán permitidas en la regla, así mismo podremos agregar las direcciones IP que serán restringidas en la misma regla (figura 1.22).



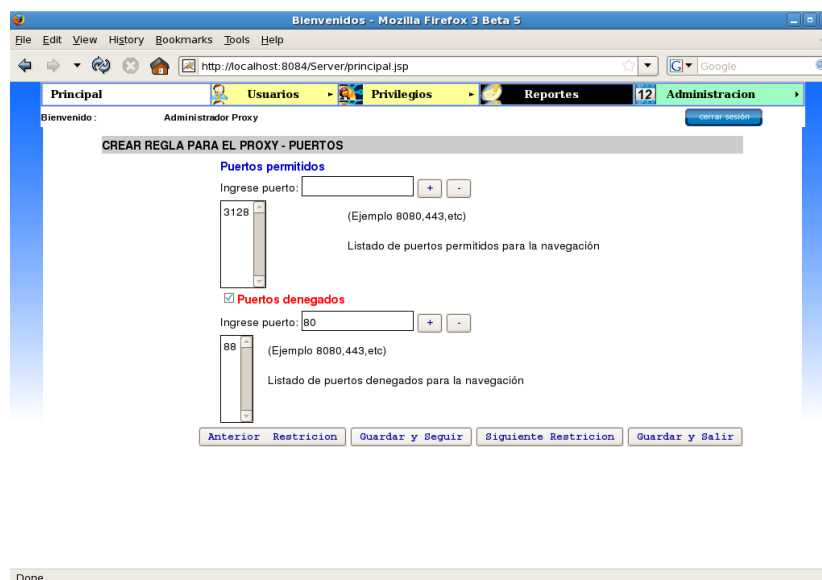


**Fig. 1.21 Pantalla Crear Regla Proxy - IP permitidas**



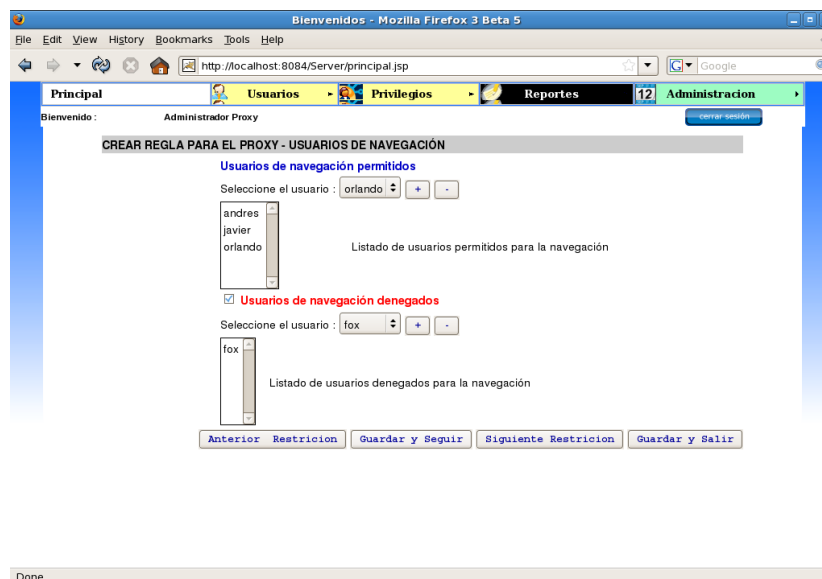
**Fig. 1.22 Pantalla Crear Regla Proxy - IP restringidas**

Una vez agregadas las direcciones IP permitidas y restringidas, hacemos un click en el botón **“Guardar y Seguir”**. El siguiente paso será especificar si la regla incluirá restricciones en algún puerto (figura 1.23).



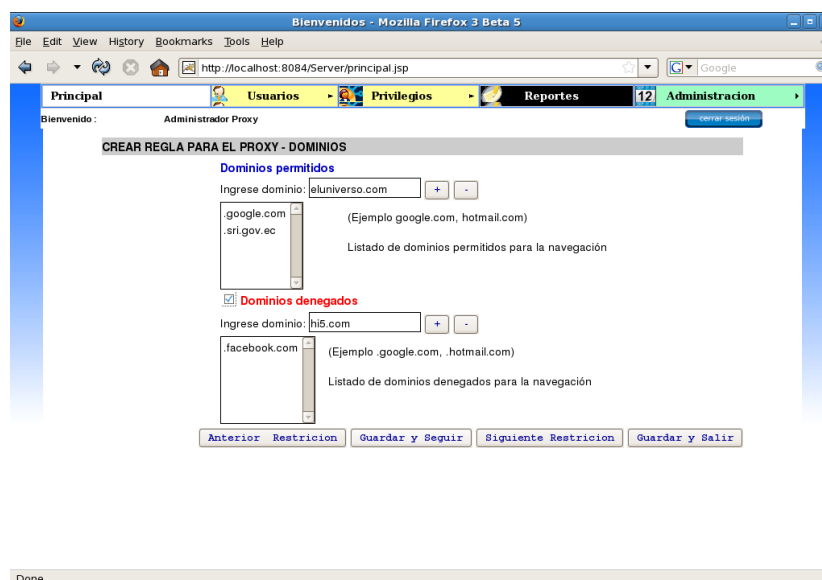
**Fig. 1.23 Pantalla Crear Regla Proxy - Puertos**

Paso siguiente es incluir o restringir usuarios de navegación (figura 1.24).



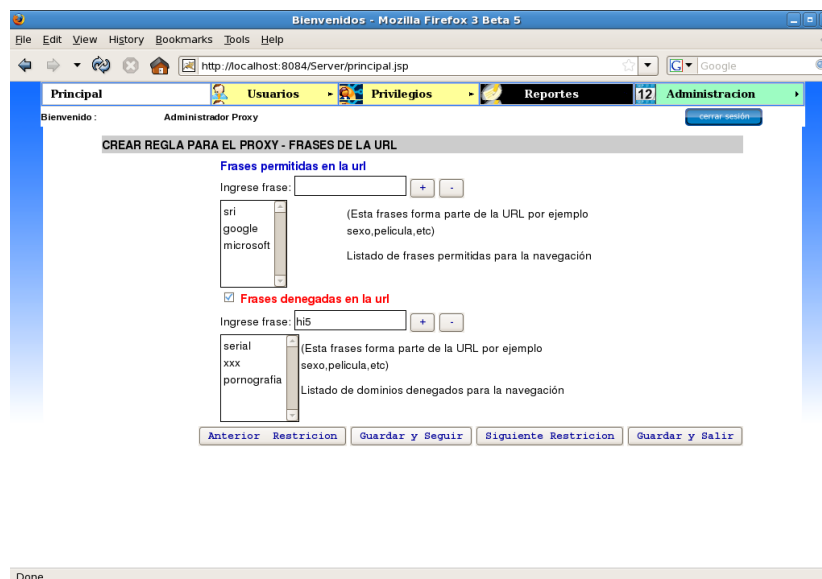
**Fig. 1.24 Pantalla Crear Regla Proxy - Usuarios**

A continuación, se detallan los dominios que se desean permitir o denegar en la regla. Se reflejará en capacidad de acceder a estos dominios. (figura 1.25)



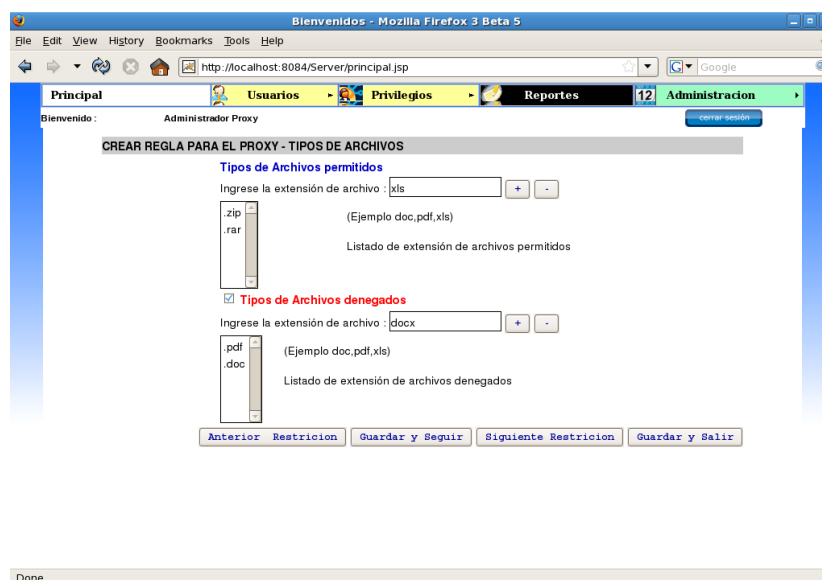
**Fig. 1.25 Pantalla Crear Regla Proxy – Dominios**

Seguidamente se podrá especificar las restricciones en las URL de palabras que deben excluirse de la navegación (figura 1.26) . Ej: SEXO.



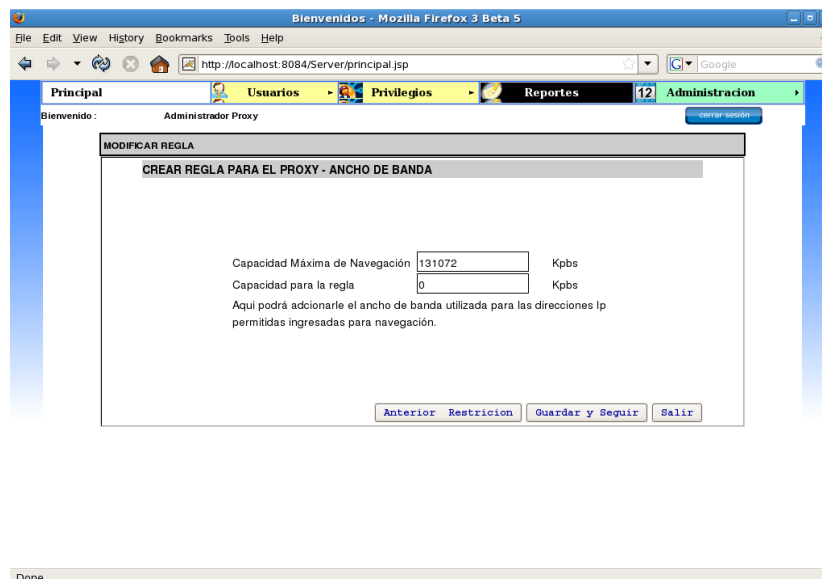
**Fig. 1.26 Pantalla Crear Regla Proxy – Frases URL**

La siguiente restricción a establecer, es la inclusión o restricción de los tipos de archivos que pueden acceder los usuarios de navegación (figura 1.27).



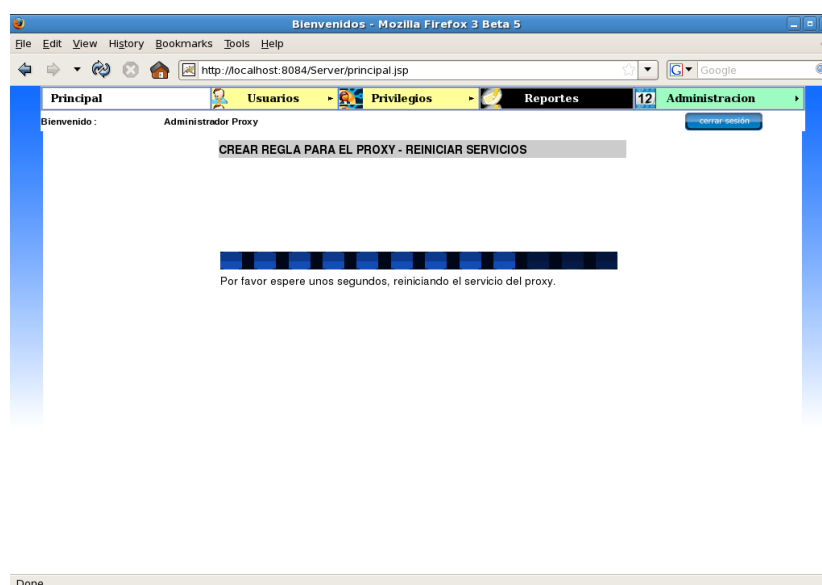
**Fig. 1.27 Pantalla Crear Regla Proxy – Tipos de Archivos.**

Seguidamente el asistente presenta la opción de especificar el ancho de banda en bytes asignados a la regla en proceso de creación (figura 1.28).



**Fig. 1.28 Pantalla Crear Regla Proxy – Ancho de Banda.**

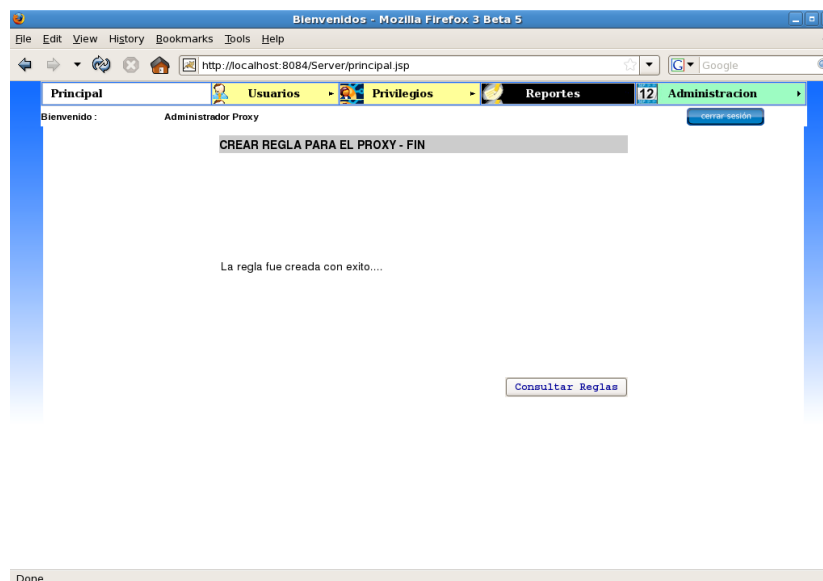
Al finalizar el anterior proceso, procedemos a la conclusión de la creación de la regla haciendo click en el botón **“Guardar y Seguir”** (figura 1.28), lo cual reiniciará los servicios del proxy en el servidor, mostrándonos la siguiente pantalla de espera (figura 1.29).



**Fig. 1.29 Pantalla Crear Regla Proxy – Reiniciar Servicios.**

Luego la pantalla de confirmación de la acción se presenta (figura 1.30), culminando así el Asistente de creación de reglas del Proxy.

Una vez creada la regla, automáticamente las restricciones en ella incluidas se ven reflejadas en la navegación de los usuarios de la red interna en IPv6.



**Fig. 1.30 Pantalla Crear Regla Proxy – Final**

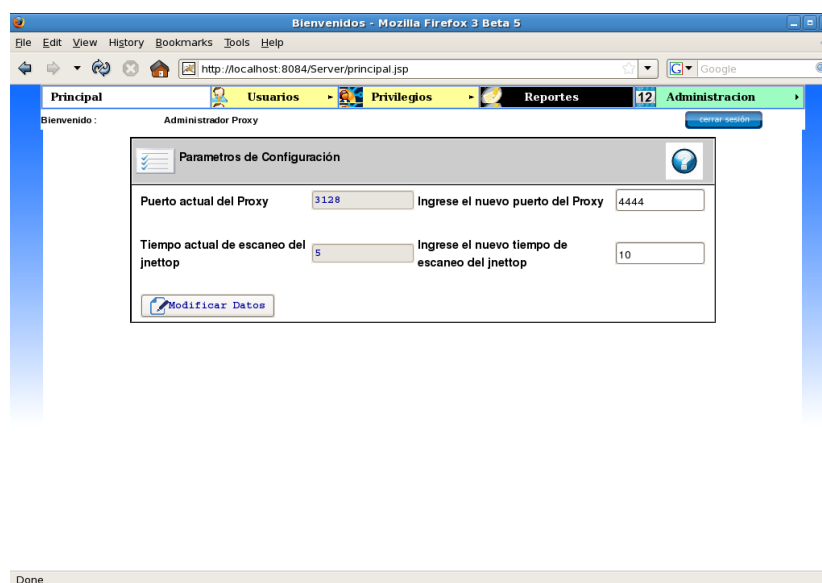
### 6.3 Parámetros de configuración.

En esta pantalla podemos realizar la configuración de dos de los parámetros por defecto que usará el Servidor Proxy (figura 1.31):

- PUERTO DEL PROXY
- TIEMPO DE ESCANEO DE TRAFICO EN LA RED.

Con más detalle podemos mencionar que en el campo de texto con la etiqueta **“Ingrese el nuevo puerto del Proxy”** se puede cambiar dinámicamente el puerto del Servidor Proxy (Squid.conf), es decir, cambiar a voluntad el puerto por donde el servidor recibirá los requerimientos, así como también podremos interactuar con el tiempo de escaneo del trafico de la red

interna IPv6 en la etiqueta **“Ingrese el nuevo tiempo de escaneo del Jnettop”**, una vez ingresados los datos hacemos un click al botón **“Modificar Datos”** para que los cambios se hagan efectivos, previamente observaremos los mensajes de confirmación de la acción de modificación.



**Fig. 1.31 Pantalla Parámetros de configuración**

## **CAPITULO # 2**

### **MANUAL TÉCNICO**

El propósito de este manual es proporcionar una guía específica de cómo está constituida la herramienta de administración del Proxy en IPV6 para el usuario final, en este caso el administrador del centro de computo.



La aplicación que administra el servidor proxy cuenta con dos roles específicos de usuarios:

- Rol Administrador.
- Rol Consultor.

Cada uno de estos roles tienen sus cuentas de usuario, password y sus respectivos privilegios (atributos).

### **Rol Administrador**

Este rol contará con todas las opciones dentro de la aplicación, para la navegación dentro de la aplicación, modificación en las reglas del proxy, etc. El cual estará asignado a la persona encargada de la administración de seguridades en la red ó Responsable del centro de cómputo.

Los siguientes son privilegios del rol administrador:

- Creación y consulta de Usuarios
- Modificar Rol de Consultor.
- Visualizar reportes de consumo de recursos.
- Mantenimiento de Usuarios de Navegación
- Reglas del Proxy
- Parámetros de Configuración

El rol Administrador es quien posee acceso a todas las opciones contenidas en esta aplicación por lo cual debe asignarse al perfil del administrador del sistema.

### **Rol Consultor**

Este rol presenta restricciones dentro de la aplicación, pues su función en el sistema será la de monitorear el tráfico de la red hacia la internet y en construir reglas que optimicen el uso adecuado de los recursos de la empresa.

Los siguientes son privilegios del rol Consultor:

- Consultas de Usuarios.
- Visualizar reportes de consumo de recursos.

### **Procesos críticos**

Las funciones serán reutilizadas en cada módulo y la aplicación del código será adecuada a las necesidades del mismo, modificando sus características, complementando las rutinas con funciones adicionales necesarias para el correcto funcionamiento, excluyendo aquellas funciones innecesarias y actualizando las sentencias SQL a las necesidades concretas de cada caso.

## CLASES

Una clase es una plantilla para un objeto. Por lo tanto define la estructura de un objeto y su interfaz funcional, en forma de métodos. Cuando se ejecuta un programa en Java, el sistema utiliza definiciones de clase para crear instancias de las clases, que son los objetos reales.

### CLASE “Conexion”

Esta clase se utiliza para realizar la conexión y desconexión de la base de datos durante todas las operaciones de la aplicación, así como la ejecución de las respectivas consultas SQL y comandos de actualización, que se realizarán en los procesos de reportes y de información almacenada.

```
import java.sql.*;

public class Conexion {
    String error;
    Connection con;
    public Conexion() {}

    //Este procedimiento se usa para creación de una conexión a la DB

    public void conectar() throws
    ClassNotFoundException, SQLException, Exception{
        try{

            //Clase usada para llamar al driver de conexión
            Class.forName("org.gjt.mm.mysql.Driver").newInstance();
            //Permite realizar la conexión a la BD
            con=DriverManager.getConnection("jdbc:mysql://localhost/proxy_server?us
            er=root&password=");
        }catch(ClassNotFoundException uno ){
            error="No puede localizar la base de datos";
            throw new ClassNotFoundException(error);
        }catch(SQLException dos ){
```

```

        error="No puede realizar la conexion de la base de datos";
        throw new SQLException(error+dos.toString());
    }catch(Exception tres ){
        error="Error desconocido";
        throw new Exception(error+tres.toString());
    }
}

//Esta función devuelve un objeto de tipo conexión usado para reportes
public Connection getConexion(){
    return con;
}

//Procedimiento para desconectar o cerrar la conexión a la BD
public void desconectar () throws SQLException{
    try{
        //Esta función nos permite saber si existe un objeto conectado a la BD
        if(con!=null){
            //Este proceso permite cerrar la conexión
            if(!con.isClosed()){
                con.close();
            }
        }
    }catch(SQLException dos ){
        error="Imposible desconectar la base de datos";
        throw new SQLException(error);
    }
}

//Esta función tiene como objetivo retornar objeto resulset para
//cualquier tipo de consulta a la BD
public ResultSet Consultas(String sql) throws
SQLException,Exception{
    ResultSet rs=null;
    try{
        //Ejecuta la consulta en la BD
        Statement stmt=con.createStatement();
        rs=stmt.executeQuery(sql);
    }catch(SQLException dos ){
        error="No puede ejecutar la consulta";
        throw new SQLException(error);
    }catch(Exception tres ){
        error="Generación de excepción en la consulta";
        throw new Exception(error);
    }
    return rs;
}

```

```
//Esta función tiene como objetivo retornar una cadena de caracteres
//por medio de una consulta a la BD, puede ser general, pero solo
//obtendremos un campo
```

```
    public String Consultas_objeto(String sql) throws
SQLException,Exception{
    ResultSet rs=null;
    String retorna="";
    try{
    Statement stmt=con.createStatement();
    rs=stmt.executeQuery(sql);
    while(rs.next()){
        retorna=rs.getString("Objeto");
    }
    }catch(SQLException dos){
        error="No puede ejecutar la consulta";
        throw new SQLException(error);
    }catch(Exception tres ){
        error="Generacion de execepción en la consulta";
        throw new Exception(error);
    }
    return retorna;
}
```

```
//Esta función tiene como objetivo consultar a la BD si dicha
//transacción tiene registros, si tiene registros retorna un valor
//booleano verdadero, caso contrario falso
```

```
    public Boolean verificaconsultas(String sql) throws
SQLException,Exception{
    boolean ban=false;
    ResultSet rs=null;
    try{
    Statement stmt=con.createStatement();
    rs=stmt.executeQuery(sql);
    if(rs.next()){
        ban=true;
    }
    }catch(SQLException dos ){
        error="No puede ejecutar la consulta";
        throw new SQLException(error);
    }catch(Exception tres ){
        error="Generacion de execepcion en la consulat";
        throw new Exception(error);
    }
    return (new Boolean(ban));
}
```

```
//Tiene como objeto ejecutar todas las instrucciones de actualización
//inserción y eliminación de datos por medio de comandos SQL

    public void mantenimiento(String sql) throws
SQLException,Exception{

        if(con!=null){
            try{
                PreparedStatement sentencia;
                sentencia=con.prepareStatement(sql);
                sentencia.execute();

            }catch(SQLException dos ){
                error="No puede ejecutar la mantenimiento";
                throw new SQLException(error);
            }catch(Exception tres ){
                error="Generacion de execepcion en la mantenimiento";
                throw new Exception(error);
            }
        }
    }

    public static void main(String[] args)throws SQLException,Exception
    {
        Conexion p= new Conexion();
        p.conectar();
    }
}
```

### CLASE “Auditoría”

Esta clase se utiliza para el registro de los accesos a la aplicación así como de los cierres de sesión de cada usuario registrado en la base, para de esta manera realizar procesos de trazabilidad en el sistema.

```
import java.sql.*;
import java.util.Vector;

public class Auditoria extends Conexion {
    CallableStatement proceso = null;
```

```
//Tiene como objeto quitar el character ":" a una determinada IP

public String formatea(String direccion){
    String dir="";
    String h="";
    for(int i=0;i<direccion.length();i++){
        if((i%2)==0 && i>0){
            dir=dir.concat(":");
            dir=dir.concat(String.valueOf(direccion.charAt(i)));
        }else{
            dir=dir.concat(String.valueOf(direccion.charAt(i)));
        }
    }
    return dir;
}

//Esta función retorna un objeto vector en el cual realiza una consulta
//a la BD por medio de un procedimiento almacenado

public Vector Consulta_acceso(){
    Vector arreglo =new Vector();
    try{
        conectar();
        proceso=con.prepareCall("{ Call VISTAS(?,?) }");
        proceso.setInt(1,5);
        proceso.setString(2,"");
        proceso.execute();
        while(proceso.getResultSet().next()){
            Vector objeto=new Vector();

            objeto.addElement(proceso.getResultSet().getString("usuario"));

            objeto.addElement(proceso.getResultSet().getString("sesion"));

            objeto.addElement(proceso.getResultSet().getString("fecha"));

            objeto.addElement(proceso.getResultSet().getString("fecha_salida"));

            arreglo.addElement(objeto);
        }
        proceso.close();
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage());
    }
    return arreglo;
}
```

```
    }
}
```

### CLASE “Configuración”

La clase Configuración es dedicada a realizar la interacción con los servicios del servidor de acceso a internet, es decir, ejecutar los cambios de configuración que se necesiten de acuerdo a las necesidades de restricción o de medición de consumo en la red.

Esta clase esta asociada a la funcionalidad de Parámetros de Configuración dentro de la aplicación, lo cual nos deja ver las validaciones que se realizan previo a los cambios que solicite el administrador.

```
import java.sql.*;
import java.util.*;
import java.net.*;
import java.io.IOException;

public class Configuracion extends Conexion{
    CallableStatement proceso = null;

    //Esta función permite ver si un puerto esta asociado con un servicio
    //de nuestro servidor, Ej: 21 puerto FTP

    public Boolean preguntar_puerto_abierto(String puerto) {
        boolean ban=true;
        try {
            //Clase que permite obtener un objeto de direccion en este caso el
            //objeto LocalHost
            InetAddress ia = InetAddress.getByName("localhost");
            //Esta clase permite realizar la conexion de un Puerto especifico de
            //nuestro servidor
            Socket s = new Socket(ia, (new Integer(puerto)).intValue());
            //Este ciomnado permite medir el tiempo de espera de la conexión
            s.setSoTimeout(2000);
            s.close();
        } catch (Exception e) {
            return false;
        }
        return true;
    }
}
```



```

    }catch (UnknownHostException ex1) {
        ban=false;
    }
    catch (IOException ex) {
        ban=false;
    }
    return (new Boolean(ban));
}
//Función que nos permite obtener un objeto vector de una consulta de
//la tabla Configuración, los datos que obtendremos: Tiempo de escaneo,
//el puerto que usa el Firewall y el Puerto que usa el Proxy.

public Vector datos_configuracion(){
    Vector datos=new Vector();
    try{
        conectar();
        proceso=con.prepareStatement("select * from configuracion");
        proceso.execute();
        while(proceso.getResultSet().next()){

datos.addElement(proceso.getResultSet().getString("escaneo"));

datos.addElement(proceso.getResultSet().getString("firewall"));

datos.addElement(proceso.getResultSet().getString("puerto"));

        }
        desconectar();
    }catch (java.sql.SQLException dos){
        System.out.println(dos.getMessage());
    }catch (Exception tres ){
        System.out.println(tres.getMessage());
    }
    return datos;
}
//Este procedimiento permite modificar los archivos de configuración
//del Proxy y Firewall, permite cambiar el puerto de uso de Proxy.

public void modifica_puerto(String t3,String t4){
    try{
        Objetos archivo=new Objetos();
        archivo.ingresa_delay("http_port "+t3,"http_port "+t4,"0");
        archivo.modifica_firewall("$IPT6 -A INPUT -i $PUBIF -p tcp --
destination-port "+t4+" -j ACCEPT","$IPT6 -A INPUT -i $PUBIF -p tcp --
destination-port "+t3+" -j ACCEPT");
        conectar();
        proceso=con.prepareStatement("update configuracion set
puerto='"+t3+"' ");
    }
}

```

```

        proceso.execute();
        desconectar();
    }catch(Exception tres ){
        System.out.println(tres.getMessage());
    }
}
//Permite modificar el tiempo de escaneo de Jnettop, el mismo que
//realiza una actualización a la tabla configuración en el campo
//escaneo.

public void modifica_jnettop(String t6,String t5){
    try{
        conectar();
        proceso=con.prepareStatement("update  configuracion set
escaneo='"+t6+"' ");
        proceso.execute();
        desconectar();
    }catch(Exception tres ){
        System.out.println(tres.getMessage());
    }
}
}

```

### CLASE “Ejecutar Procesos”

La clase `Ejecutar_Procesos` es la llamada a relizar la interacción con los servicios: Squid que son los encargados de manipular las reglas de acceso, Firewall IPV6, Registro de usuarios de navegación de red, Captura de datos del servicio Jnettop para reportes de consumo de internet.

```

import java.io.*;
import java.util.StringTokenizer;
import java.util.Vector;
import java.sql.*;

public class Ejecutar_Procesos {
    //Este procedimiento tiene como finalidad cerrar el proceso de Squid
    //utilizando el runtime de java y también reiniciar el servicio.
    public void restart_squid(){
        try {
            Process proc =
Runtime.getRuntime().exec("/opt/squid/sbin/squid -k kill");
            proc.waitFor();

```

```

//El bucle es ejecutado para dar tiempo al proceso de refrescamiento
//del servicio SQUID, en este caso para bajar el servicio.
        for(int i=0;i<20000;i++){
            for(int j=0;j<500;j++){
                }
            }
        proc =
Runtime.getRuntime().exec("/opt/squid/sbin/squid");
        proc.waitFor();
//El bucle es ejecutado para dar tiempo al proceso de refrescamiento
//del servicio SQUID, en este caso para levantar el servicio.

        for(int i=0;i<20000;i++){
            for(int j=0;j<500;j++){
                }
            }
        } catch (InterruptedException ex) {
            ex.printStackTrace();
        } catch (IOException eproc){
            System.out.println(eproc);
        }
    }
//Permite controlar el firewall, levantando y bajando el servicio.

    public void restart_firewall(){
        try {
//Proceso usado para detener el firewall en caso de actualización
            Process proc = Runtime.getRuntime().exec("service
iptables stop");
            proc.waitFor();
//Proceso usado para levantar el firewall en caso de actualización
            proc =
Runtime.getRuntime().exec("/etc/rc.d/firewallip6");
            proc.waitFor();
        } catch (InterruptedException ex) {
            ex.printStackTrace();
        } catch (IOException eproc){
            System.out.println(eproc);
        }
    }
//Permite crear los usuarios de navegación que son usados en el archivo
//de configuración del proxy, usando el runtime de java para ejecutar
//commandos htpasswd.

    public void crear_usuario(String usuario,String clave){
        try {
            System.out.println("bash crea_usuarios "+usuario+"
"+clave+" ");

```

```

//Proceso que hace posible la introducción de usuario y contraseña
//para permitir la navegación desde el browser
        Process proc = Runtime.getRuntime().exec("htpasswd -b
/opt/squid/clave_squid "+usuario+" "+clave);
        proc.waitFor();
        String linea="";
        BufferedReader buffer=new BufferedReader(new
InputStreamReader(proc.getInputStream()));
        while((linea = buffer.readLine())!=null)
        {
            System.out.println(linea);
        }

        } catch (InterruptedException ex) {
            ex.printStackTrace();
        } catch (IOException eproc) {
            System.out.println(eproc);
        }
    }

//Tiene como fin ejecutar y registrar las actividades de la red interna
//por medio del runtime de java
public Vector ejecuta_jnettop(){
    String respuesta = null;
    String linea;
    respuesta="";
    Vector datos=new Vector();
    try{
//Proceso que captura datos registrados de navegación: IP, bytes
//transmitidos, y los ubica ordenados en un vector para presentación

        Clases.Conexion base =new Clases.Conexion();
        base.conectar();
        String nu=base.Consultas_objeto("select escaneo as objeto
from configuracion");
        base.desconectar();

        Process proc = Runtime.getRuntime().exec("jnettop -i eth1 --
display text -t "+nu+" --format '$src$-$srcport$-$srcbytes$-$dst$-$
dstport$-$dstbytes$'");
        proc.waitFor();
        BufferedReader buffer=new BufferedReader(new
InputStreamReader(proc.getInputStream()));
        while((linea = buffer.readLine())!=null)
        {
            StringTokenizer tokens = new StringTokenizer(linea,"-");
            while(tokens.hasMoreTokens()){
                String ip_or=tokens.nextToken();

```

```

        ip_or=ip_or.replace("\\", "");

        String port_or=tokens.nextToken();
        String bytes_or=tokens.nextToken();
        String ip_ds=tokens.nextToken();

        String port_ds=tokens.nextToken();
        String bytes_ds=tokens.nextToken();
        bytes_ds=bytes_ds.replace("\\", "");
        String usuario="";
        if(ip_ds.indexOf("::")>0){
            Vector dato=new Vector();
            dato.addElement(ip_or);
            dato.addElement(port_or);
            dato.addElement(bytes_or);
            dato.addElement(ip_ds);
            dato.addElement(port_ds);
            dato.addElement(bytes_ds);

            usuario=(new
Objetos()).busca_nombre_objeto(ip_ds);
            if(usuario.length()==0){
                dato.addElement(" ");
            }else{
                usuario=usuario.substring(0,
usuario.indexOf("_"));
                base.conectar();
                usuario=base.Consultas_objeto("select
funciones(9, '"+usuario+"', '') as objeto");
                if(usuario.length()==0){
                    dato.addElement(" ");
                }else{
                    dato.addElement(usuario);
                }
                base.desconectar();
            }
            datos.addElement(dato);
        }
    }
} catch (SQLException ex) {

} catch (IOException eproc){
    System.out.println(eproc);
} catch (InterruptedException ex) {
    ex.printStackTrace();
} catch (ClassNotFoundException ex) {

} catch (Exception ex) {

```

```

        }
        return datos;
    }

    public static void main(String[] args){
        Ejecutar_Procesos ob=new Ejecutar_Procesos();
        ob.crear_usuario("jose", "222");
    }
}

```

### CLASE “Funciones”

La clase Funciones realiza las verificaciones de seguridad y registros de auditoria en los accesos para los usuarios administrador o consultor de la aplicación.

```

import java.security.*;

public class Funciones {

    //Función usada para la identificación de usuarios de la página web

    public Boolean verifica_login(String usuario, String clave,String
aletorio){
        boolean bandera=true;
        String clave_usuario="";
        try{
            Conexion conectar =new Conexion();
            conectar.conectar();

            //Proceso que pregunta a la base si el usuario esta registrado
            if(true==conectar.verificaconsultas("select * from usuarios
where usuario='"+usuario+"' ").booleanValue()){
                clave_usuario=conectar.Consultas_objecto("select clave as
objecto from usuarios where usuario='"+usuario+"'");
                if(clave.toUpperCase().compareTo(convertir_clave(clave_usuario,aletorio
))!=0){
                    bandera=false;
                }else{

```

```
//Proceso que suma un código aleatorio en texto al dato de usuario

        conectar.mantenimiento("update usuarios set
sesion='"+aleatorio+"' where usuario='"+usuario+"' ");
    }
    }else{
        bandera=false;
    }
    conectar.desconectar();
}catch(java.sql.SQLException dos){
}catch(ClassNotFoundException uno ){
}catch(Exception tres ){
}

    return (new Boolean(bandera));
}

//Función usada para registrar en la tabla auditoría accesos de
//usuarios a la página con sus respectivas capturas de fecha y hora.

public String graba_acceso(String usuario,String tipo){
    String sesion="";
    try{
        Conexion conectar =new Conexion();
        Fechas fecha =new Fechas();
        conectar.conectar();
        if(tipo.length()==0){
            sesion=convertir_clave(String.valueOf(fecha.se),"");
        }

//Proceso que registra en la tabla auditoría los accesos de usuarios

        conectar.mantenimiento("insert into auditoria
(usuario,fecha,tipo,sesion)value ('"+usuario+"',now(),0,'"+sesion+"')");
    }else{

//Proceso que registra en la tabla auditoría las salidas de usuarios

        conectar.mantenimiento("update  auditoria set
fecha_salida=now() where sesion='"+tipo+"'");
    }
    conectar.desconectar();

    }catch(java.sql.SQLException dos){
    }catch(ClassNotFoundException uno ){
    }catch(Exception tres ){
    }
    }
    return sesion;
}
}
```

```

//Esta función permite cambiar una cadena de caracteres a un conjunto
//de caracteres especiales usando algoritmos de encriptación MD5, en el
//cual se usan 4 octetos para transformarlos en bytes hasta
//convertirlos en una clave casi indecifrable.
private String convertir_clave(String clave, String numero){
    byte[] hash = null;
    byte[] uniqueKey =null;
        clave=clave.concat(numero);
    uniqueKey = clave.getBytes();
    try {
        hash = MessageDigest.getInstance("MD5").digest(uniqueKey);
    }catch (NoSuchAlgorithmException e) {

        throw new Error("no MD5 support in this VM");
    }
    StringBuffer hashString = new StringBuffer();
    for ( int i = 0; i < hash.length; ++i ) {
        String hex = Integer.toHexString(hash[i]);
        if ( hex.length() == 1 ) {

            hashString.append('0');
            hashString.append(hex.charAt(hex.length()-1));
        } else {
            hashString.append(hex.substring(hex.length()-2));
        }
    }

    return hashString.toString().toUpperCase();
}
}

```

## **CLASE “Objetos”**

En la clase Objetos encontramos las funciones que realizan el barrido y manipulación del archivo de configuración del Squid que contiene las reglas de restricción de acceso a la navegación en la internet.

Es la clase más importante pues dentro de ella se encuentran los procesos que hacen posible la ejecución de la administración de Squid y por consiguiente del servidor proxy.



```

import java.io.*;
import java.sql.SQLException;
import java.util.StringTokenizer;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Objetos extends Conexion {

    //Permite buscar un SRC en el archivo de configuración del proxy,
    //mediante un barrido de todo el texto

    public String busca_nombre_objeto(String nombre){
        String cont="";
        try {
            this.conectar();
            String archivo = Consultas_objeto("select archivo as objeto from
configuracion");
            this.desconectar();
            File f = new File(archivo);
            BufferedReader entrada;
            String linea="";
            entrada = new BufferedReader( new FileReader( f )
);
            while(entrada.ready()){
                linea = entrada.readLine();
                if(linea.indexOf(nombre)>=0 &&
linea.indexOf("src")>=0){
                    int j=linea.indexOf("src");
                    cont=linea.substring(3, j).trim();
                    break;
                }
            }
            entrada.close();
        } catch (Exception ex) {
            Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
        }
        return (cont);
    }

    //Permite ver usuarios que estan ingresados en el archivo de
    //navegación para utilización de proxy y son puestos en formato HTML.

    public String lista_usuarios_combo(){
        String items="";
        try {
            this.conectar();

```

```

        String archivo = Consultas_objecto("select archivo_usuarios as
objecto from configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        entrada = new BufferedReader( new FileReader( f )
);
        while(entrada.ready()){
            linea = entrada.readLine();
            String usuario=linea.substring(0,
linea.indexOf(":"));
            items=items.concat("<option
value='"+usuario+"'>"+usuario+"</option>");
        }
        entrada.close();
    } catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return items;
}
//Función que tiene como finalidad buscar solo un Usuario en todo el
//archivo de configuración del Proxy.
public Boolean busca_usuarios(String nombre){
    boolean ban=false;
    try {
        this.conectar();
        String archivo = Consultas_objecto("select archivo_usuarios as
objecto from configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        entrada = new BufferedReader( new FileReader( f )
);
        while(entrada.ready()){
            linea = entrada.readLine();
            if(linea.indexOf(nombre)>=0 ){
                ban=true;
                break;
            }
        }
        entrada.close();
    } catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE, null, ex);
    }
    return (new Boolean(ban));
}

```

```

//Tiene como fin buscar un objeto en el archivo de configuración del
//proxy
    public Boolean busca_objeto(String nombre){
        boolean ban=false;
        try {
            this.conectar();
            String archivo = Consultas_objeto("select archivo as objeto from
configuracion");
            this.desconectar();
            File f = new File(archivo);
            BufferedReader entrada;
            String linea="";
            entrada = new BufferedReader( new FileReader( f )
);
            while(entrada.ready()){
                linea = entrada.readLine();
                if(linea.indexOf(nombre)>=0 ){
                    ban=true;
                    break;
                }
            }
            entrada.close();
        } catch (Exception ex) {
            Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
        }
        return (new Boolean(ban));
    }

//Permite buscar los delay registrados en el archivo de configuración
//mediante un barrido del texto

    public Boolean busca_delay_access(String nombre,Vector num){
        boolean ban=false;
        try {
            this.conectar();
            String archivo = Consultas_objeto("select archivo as objeto from
configuracion");
            this.desconectar();
            File f = new File(archivo);
            BufferedReader entrada;
            String linea="";
            entrada = new BufferedReader( new FileReader( f )
);
            while(entrada.ready()){
                linea = entrada.readLine();
                if(linea.indexOf(nombre)>=0 &&
linea.indexOf("delay_access")>=0){

```

```

num.addElement(linea.substring(12,linea.indexOf("allow")-1));

                ban=true;
                break;
            }
        }
        entrada.close();
    } catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return (new Boolean(ban));
}
//Busca un delay, pero obtiene el valor del delay
public String busca_parametro_access2(String num){
    String valor="";
    try {
        this.conectar();
        String archivo = Consultas_objecto("select archivo as objeto from
configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        entrada = new BufferedReader( new FileReader( f )
);
        while(entrada.ready()){
            linea = entrada.readLine();
            if(linea.indexOf("delay_parameters "+num+"
")>=0){
                valor=linea.substring(linea.lastIndexOf("
"),linea.lastIndexOf("/"));
                break;
            }
        }
        entrada.close();
    } catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return (valor);
}
//Tiene como fin buscar el número de puerto que esta registrado en una
//ACL
public Boolean busca_objeto_tipo_puerto(String nombre){
    boolean ban=false;
    try {
        this.conectar();

```

```

        String archivo = Consultas_objecto("select archivo as objeto from
configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        entrada = new BufferedReader( new FileReader( f )
);
        while(entrada.ready()){
            linea = entrada.readLine();
            if(linea.indexOf(nombre)>=0 &&
linea.indexOf("acl")>=0 && linea.indexOf("port")>=0){
                ban=true;
                break;
            }
        }
        entrada.close();
    } catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return (new Boolean(ban));
}

//Tiene como fin buscar una palabra especifica en el archivo de
//configuración

public Boolean busca_palabra(String nombre){
    boolean ban=false;
    try {
        this.conectar();
        String archivo = Consultas_objecto("select archivo as objeto from
configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        entrada = new BufferedReader( new FileReader( f )
);
        while(entrada.ready()){
            linea = entrada.readLine();
            if(linea.indexOf(nombre)>=0 ){
                ban=true;
                break;
            }
        }
        entrada.close();
    } catch (Exception ex) {

```

```

        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return (new Boolean(ban));
}

//Tiene como finalidad buscar una frase entera en el archivo de
//configuración del proxy

public String retorna_frases(String nombre){
String contenido="";
try {
    this.conectar();
    String archivo = Consultas_objecto("select
archivo as objeto from configuracion");
    this.desconectar();
    File f = new File(archivo);
    BufferedReader entrada;
    String linea="";
    entrada = new BufferedReader( new
FileReader( f ) );

    while(entrada.ready()){
        linea = entrada.readLine();
        if(linea.indexOf(nombre)>=0 ){
            contenido=linea;
            break;
        }
    }
    entrada.close();

} catch (Exception ex) {
    Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
}
return contenido;
}

//Verifica el ancho de banda ingresado en el archivo de configuración
//del proxy mediante los delay, y luego los retorna en formaro HTML
public String consulta_administrador_ancho_banda(){
String contenido="";

try {
    this.conectar();
    String archivo = Consultas_objecto("select
archivo as objeto from configuracion");
    String pools = Consultas_objecto("select pools
as objeto from configuracion");
    this.desconectar();
    int j=Integer.parseInt(pools);

```

```

        for(int i=1;i<=j;i++){
            File f = new File(archivo);
            BufferedReader entrada;
            String linea="";
            entrada = new BufferedReader(
new FileReader( f ) );
            while(entrada.ready()){
                linea =
entrada.readLine();

            if(linea.indexOf("delay_parameters "+String.valueOf(i))>=0 ){
                StringTokenizer tokens
= new StringTokenizer(linea," ");
                tokens.nextToken();
                tokens.nextToken();
                String
valor=tokens.nextToken();

                valor=valor.substring(0, valor.indexOf("/"));

                contenido=contenido.concat(" <tr><td width='31' ><div
align='center'>" +String.valueOf(i)+"</div></td><td width='170' ><div
align='center'>" +valor+"</td>");
            }

            if(linea.indexOf("delay_access "+String.valueOf(i))>=0 ){
                StringTokenizer tokens
= new StringTokenizer(linea," ");
                tokens.nextToken();
                tokens.nextToken();
                tokens.nextToken();
                String
valor=tokens.nextToken();

                contenido=contenido.concat("<td width='418' '><div
align='center'>" +valor+"</div></td></tr>");
            }
        }
        entrada.close();
    }

    }catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return contenido;
}

```

```

//Permite modificar una ACL

public void señala_objeto(String nombre){
    boolean ban=false;
    String contenido="";
    try {
        this.conectar();
        String archivo = Consultas_objeto("select archivo as
objecto from configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        entrada = new BufferedReader( new FileReader( f )
    );
        while(entrada.ready()){
            linea = entrada.readLine();
            if(linea.indexOf(nombre)>=0 &&
linea.indexOf("acl")>=0){
                contenido=contenido.concat("*modifica*"+linea+"\n");
            }else{
                contenido=contenido.concat(linea+"\n");
            }
        }
        entrada.close();
        crea_archivos(archivo,contenido);
    } catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

//Permite modificar un objeto del archivo de configuración del proxy

public void eli_mod_objeto(String nombre){
    boolean ban=false;
    String contenido="";
    try {
        this.conectar();
        String archivo = Consultas_objeto("select archivo as
objecto from configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        entrada = new BufferedReader( new FileReader( f )
    );
        while(entrada.ready()){

```



```

        linea = entrada.readLine();
        if(linea.indexOf(nombre)>=0 &&
linea.indexOf("*modifica*")>=0) {
            }else{
                contenido=contenido.concat(linea+"\n");
            }
        }
        entrada.close();
        crea_archivos(archivo,contenido);
    } catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
//Permite borrar los valores del ancho de banda del archivo de
//configuración del proxy

public void elimina_banda(String pools){
    ingresa_delay("", "delay_access "+pools, "3");
    ingresa_delay("", "delay_parameters "+pools, "3");
    ingresa_delay("", "delay_class "+pools, "3");
    ingresa_delay("delay_pools
"+String.valueOf(Integer.parseInt(pools)-1), "delay_pools "+pools, "0");
    if(Integer.parseInt(pools)-1==0) {
        ingresa_delay("", "delay_pools ", "3");
    }
    try {
        this.conectar();
        this.mantenimiento("update configuracion set pools=pools-
1");
        this.desconectar();
    } catch (SQLException ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

//Permite eliminar los usuarios de navegación del proxy

public void elimina_usuario(String usuario,String clave ){
boolean ban=false;
String contenido="";
try {
    this.conectar();
    String archivo = Consultas_objeto("select archivo_usuarios as
objecto from configuracion");

```

```

        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        entrada = new BufferedReader( new FileReader( f )
);
        while(entrada.ready()){
            linea = entrada.readLine();
            if(!(linea.indexOf(usuario)>=0 &&
linea.indexOf(clave)>=0 )){
                contenido=contenido.concat(linea+"\n");
            }
        }
        entrada.close();
        crea_archivos(archivo,contenido);
    } catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }

}

//Elimina una ACL con sus reglas

public void elimina_objetos(String nombre ){
    boolean ban=false;
    String contenido="";
    try {
        this.conectar();
        String archivo = Consultas_objeto("select archivo as objeto from
configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        entrada = new BufferedReader( new FileReader( f )
);
        while(entrada.ready()){
            linea = entrada.readLine();
            if(!(linea.indexOf(nombre)>=0 &&
linea.indexOf("acl")>=0 )){
                contenido=contenido.concat(linea+"\n");
            }
        }
        entrada.close();
        crea_archivos(archivo,contenido);
    } catch (Exception ex) {

```

```

        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }

}

//Elimina los http Access del archivo de configuración del proxy
public void elimina_objetos_reglas(String nombre){
boolean ban=false;
String contenido="";
try {
    this.conectar();
    String archivo = Consultas_objeto("select archivo as objeto from
configuracion");
    this.desconectar();
    File f = new File(archivo);
    BufferedReader entrada;
    String linea="";
    entrada = new BufferedReader( new FileReader( f )
);
    while(entrada.ready()){
        linea = entrada.readLine();
        if(linea.indexOf(nombre)>=0 &&
linea.indexOf("http_access")>=0 ){
            linea=linea.replaceAll(nombre,"");
            if(linea.compareTo("http_access allow
")!=0){
                contenido=contenido.concat(linea+"\n");
            }
        }else{
            contenido=contenido.concat(linea+"\n");
        }
    }
    entrada.close();
    crea_archivos(archivo,contenido);
} catch (Exception ex) {
    Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
}
}

//Permite verificar si la regla existe en el archivo de configuración

public Boolean busca_objeto_reglas(String nombre){
boolean ban=false;
try {
    this.conectar();
    String archivo = Consultas_objeto("select archivo as objeto from
configuracion");

```

```

        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        entrada = new BufferedReader( new FileReader( f )
);
        while(entrada.ready()){
            linea = entrada.readLine();
            if(linea.indexOf(nombre)>=0 &&
linea.indexOf("http_access")>=0){
                ban=true;
                break;
            }
        }
        entrada.close();
    } catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return (new Boolean(ban));
}

//Permite presentar todas las reglas que fueron creadas

public String consulta_objeto(){
    String contenido="";
    try {
        this.conectar();
        String archivo = Consultas_objecto("select archivo_usuarios
as objeto from configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        int e=0;
        entrada = new BufferedReader( new FileReader( f )
);
        while(entrada.ready()){
            linea = entrada.readLine();
            StringTokenizer tokens = new
StringTokenizer(linea,":");
            String usuario=tokens.nextToken();
            String clave=tokens.nextToken();
            contenido=contenido.concat("<tr
id='fila'+String.valueOf(e)+' '
onMouseOver='enciende('"+String.valueOf(e)+"',1) '
onMouseOut='apaga('"+String.valueOf(e)+"',1) '>");
            contenido=contenido.concat("<td><input
type='radio' name='codigo' value='1' id='RadioGroup1_0'

```

```

onClick='elementos (" "+usuario+"\\"," "+clave+"\\") '
/></td><td>" +usuario+"</td><td>*****</td></tr>\n");
        e++;

    }
    entrada.close();
} catch (Exception ex) {
    Logger.getLogger (Objetos.class.getName()).log (Level.SEVERE,
null, ex);
}
return contenido;
}

//Presenta una regla individual dependiendo de la ACL.

public String consulta_objeto_individual(String objeto){
    String contenido="";
    try {
        this.conectar();
        String archivo = Consultas_objeto("select archivo as
objecto from configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        int e=0;
        entrada = new BufferedReader( new FileReader( f )
);
        while(entrada.ready()){
            linea = entrada.readLine();
            if(linea.indexOf("acl")==0 &&
linea.indexOf(objeto)>0 ){
                StringTokenizer tokens = new
StringTokenizer(linea, " ");
                tokens.nextToken();
                String nombre=tokens.nextToken();
                String tipo=tokens.nextToken();
                String valor=tokens.nextToken();
                if(tipo.compareTo("url_regex")==0 ){
                    valor=tokens.nextToken();
                }
                if(tipo.compareTo("urlpath_regex")==0){
                    valor=valor.replace("$", "");
                    valor=valor.replace("\\", "");
                }
            }
        }
    }
}

```

```

                                contenido=contenido.concat(" <option
value='"+valor+"'>" + valor + "</option>");
                                e++;
                                }else{
                                    if(linea.indexOf("#fin_reglas")>=0) {
                                        break;
                                    }
                                }
                            }
                        }
                        entrada.close();
                    } catch (Exception ex) {
                        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
                    }
                    return contenido;
                }

//Presenta la información como un vector de todas las reglas del
//archivo de configuración

public Vector consulta_regla_individual(String numero){
    Vector datos =new Vector();
    try {
        this.conectar();
        String archivo = Consultas_objeto("select archivo as
objecto from configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        int num=Integer.parseInt(numero);
        int e=0;
        entrada = new BufferedReader( new FileReader( f )
);

        while(entrada.ready()){
            linea = entrada.readLine();
            if(linea.indexOf("http_access")==0){
                if(e==num){
                    StringTokenizer tokens = new
StringTokenizer(linea," ");
                    tokens.nextToken();
                    String nombre=tokens.nextToken();
                    datos.addElement(nombre);
                    Vector objetos=new Vector();
                    String objeto=tokens.nextToken();
                    objetos.addElement(objeto);
                    while(tokens.hasMoreElements()){
                        String ve=tokens.nextToken();
                        objetos.addElement(ve);

```

```

        }
        datos.addElement(objetos);
        break;

    }
    e++;
} else {
    if (linea.indexOf("#fin_reglas") >= 0) {
        break;
    }
}
}
    entrada.close();
} catch (Exception ex) {
    Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
}
    return datos;
}

//Esta función lista todas las ACL en el formato de un Combo para
//presentarlas en la página Web

public String consulta_objeto_src() {
    String contenido = "";
    try {
        this.conectar();
        String archivo = Consultas_objeto("select archivo as
objeto from configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea = "";
        int e = 0;
        entrada = new BufferedReader( new FileReader( f )
);
        while (entrada.ready()) {
            linea = entrada.readLine();
            if (linea.indexOf("acl") == 0 &&
linea.indexOf("src") > 0 ) {
                StringTokenizer tokens = new
StringTokenizer(linea, " ");
                tokens.nextToken();
                String nombre = tokens.nextToken();
                String tipo = tokens.nextToken();
                String valor = tokens.nextToken();
                if (contenido.indexOf(nombre) < 0) {

```

```

                                contenido=contenido.concat(" <option
value='"+nombre+"'>" + nombre + "</option>");
                                }
                                e++;
                                }else{
                                if(linea.indexOf("#fin_reglas")>=0) {
                                break;
                                }
                                }
                                }
                                entrada.close();
                                } catch (Exception ex) {
                                Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
                                null, ex);
                                }
                                return contenido;
                                }

//Permite presentar los ACL en el Wizard

    public String consulta_objeto_global(){
        String contenido="";
        try {
            this.conectar();
            String archivo = Consultas_objeto("select archivo as
objeto from configuracion");
            this.desconectar();
            File f = new File(archivo);
            BufferedReader entrada;
            String linea="";
            int e=0;
            entrada = new BufferedReader( new FileReader( f )
);
            while(entrada.ready()){
                linea = entrada.readLine();
                if(linea.indexOf("acl")==0 ) {
                    StringTokenizer tokens = new
StringTokenizer(linea," ");
                    tokens.nextToken();
                    String nombre=tokens.nextToken();
                    String tipo=tokens.nextToken();
                    String valor=tokens.nextToken();
                    if(contenido.indexOf(nombre)<0) {
                        contenido=contenido.concat(" <option
value='"+nombre+"'>" + nombre + "</option>");
                    }
                    e++;
                }else{
                    if(linea.indexOf("#fin_reglas")>=0) {

```



```

                                break;
                            }
                        }
                    }
                }
                entrada.close();
            } catch (Exception ex) {
                Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
            }
            return contenido;
        }

//Retorna el valor del ancho de banda que fue configurado anteriormente
//en el archivo de configuración del proxy

    public String valor_banda() {
        String contenido="";
        try {
            this.conectar();
            contenido = Consultas_objecto("select conexion as objeto
from configuracion");
            this.desconectar();
        } catch (SQLException ex) {
            Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (Exception ex) {
            Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
        }
        return contenido;
    }

//Retorna el número de pools que estan ingresados en el archivo de
//configuración
    public String valor_pool() {
        String contenido="";
        try {
            this.conectar();
            contenido = Consultas_objecto("select pools as objeto from
configuracion");
            this.desconectar();
        } catch (SQLException ex) {
            Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (Exception ex) {
            Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
        }
        return contenido;
    }

```

```

//Permite consultar y eliminar objetos del archivo de configuración
public void consulta_objetos(String nombre) {
    if(busca_objeto(nombre).booleanValue()){
        elimina_objetos(nombre);
    }
}

//Permite registrar los usuarios de navegación en la BD
public void ingresa_usuario_base(String regla,String usuario){
    try {
        this.conectar();
        mantenimiento("insert into usuarios_navegacion
(regla,usuario)values ('"+regla+"','"+usuario+"')");
        this.desconectar();
    } catch (SQLException ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

//Permite eliminar los usuarios de navegación en la BD
public void borrar_usuario_base(String regla){
    try {
        this.conectar();
        mantenimiento("delete from usuarios_navegacion where
regla='"+regla+"'");
        this.desconectar();
    } catch (SQLException ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

//Permite crear los usuarios de navegación
public void ingresa_usuarios_navegacion(String nombre,String
tipo){
    Clases.Ejecutas_Procesos objeto = new
Clases.Ejecutas_Procesos();
    objeto.crear_usuario(nombre, tipo);
}

```

```

//Permite crear los nombres de las ACL
    public void  ingresa_objetos(String nombre,String tipo,String
valor){
        try {
            this.conectar();
            String contenido="";
            String archivo = Consultas_objeto("select archivo as
objecto from configuracion");
            this.desconectar();
            File f = new File(archivo);
            BufferedReader entrada;
            String linea="";
            entrada = new BufferedReader( new FileReader( f )
);
            while(entrada.ready()){
                linea = entrada.readLine();
                int num=linea.indexOf("#politicas_objetos");
                if(num>=0){
                    if(tipo.compareTo("urlpath_regex")==0){
                        valor="\\ "+valor+"$";
                    }
                    contenido=contenido.concat("acl "+nombre+"
"+tipo+" "+valor+" \n");
                    contenido=contenido.concat(linea+"\n");
                }else{
                    contenido=contenido.concat(linea+"\n");
                    if(linea.indexOf("#fin_reglas")>=0){
                        // break;
                    }
                }
            }
            entrada.close();
            crea_archivos(archivo,contenido);
        } catch (SQLException ex) {
            Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (Exception ex) {
            Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }

//Este procedimiento permite modificar los valores de las ACL
    public void  modifica_objetos(String nombre,String tipo,String
valor){
        try {
            this.conectar();
            boolean ban=true;
            String contenido="";

```

```

        String archivo = Consultas_objecto("select archivo as
objecto from configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        entrada = new BufferedReader( new FileReader( f )
);

        while(entrada.ready()){
            linea = entrada.readLine();
            int num=linea.indexOf("*modifica*");
            if(num>=0 && ban==true){
                if(tipo.compareTo("url_regex")==0){
                    tipo=tipo.concat(" -i");
                }
                if(tipo.compareTo("urlpath_regex")==0){
                    valor="\\ "+valor+"$";
                }
                contenido=contenido.concat("acl "+nombre+"
"+tipo+" "+valor+" \n");

                contenido=contenido.concat(linea+"\n");
                ban=false;
            }else{
                contenido=contenido.concat(linea+"\n");
                if(linea.indexOf("#fin_reglas")>=0){
                }
            }
        }
        entrada.close();
        crea_archivos(archivo,contenido);
    } catch (SQLException ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

//Permite crear archivos de configuración
public void crea_archivos(String archivo,String contenido){
    try {
        FileWriter fw= new FileWriter(archivo);
        for(int i=0;i<contenido.length();i++){
            fw.write(contenido.charAt(i));
        }
        fw.close();
        (new Ejecutas_Procesos()).restart_squid();
    }
}

```

```

    }catch (IOException e) {
        e.printStackTrace();
    }
}

//Este procedimiento tiene multiples tareas a realizar, como: insertar
//antes de una frase, despues de una frase, en la misma frase, borrar
//la frase y concatenar frase

    public void  ingresa_delay(String palabra_inserta,String
antes_palabra,String antes ){
    try {
        this.conectar();
        String contenido="";
        String archivo = Consultas_objeto("select archivo as
objecto from configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        entrada = new BufferedReader( new FileReader( f )
);

        while(entrada.ready()){
            linea = entrada.readLine();
            int num=linea.indexOf(antes_palabra);
            if(num>=0) {
                //en la misma posicion
                if(antes.compareTo("0")==0) {

contenido=contenido.concat(palabra_inserta+"\n");
                }
                //antes de lo encontrado
                if(antes.compareTo("1")==0) {

contenido=contenido.concat(linea+"\n");

contenido=contenido.concat(palabra_inserta+"\n");
                }
                //despues de lo encontrado
                if(antes.compareTo("2")==0) {

contenido=contenido.concat(palabra_inserta+"\n");

contenido=contenido.concat(linea+"\n");
                }
                //eliminado
                if(antes.compareTo("3")==0) {

```

```

//adicionar
if(antes.compareTo("4")==0) {

contenido=contenido.concat(linea+" "+palabra_inserta+"\n");
}
}else{
    contenido=contenido.concat(linea+"\n");
}
}
entrada.close();
crea_archivos(archivo,contenido);
} catch (SQLException ex1) {
    Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex1);
} catch (Exception ex) {
    Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
}
}

//Permite crear regla en el archivo de configuración

public void agrega_reglas_ancho_banda(String pool,String
numero,String objeto){
    Vector dato=new Vector();
    if(busca_palabra("delay_pools").booleanValue()){
        if(busca_delay_access(objeto,dato).booleanValue()==false) {
            ingresa_delay("delay_class
"+String.valueOf(Integer.parseInt(pool)+1)+" 1","delay_class
"+String.valueOf(Integer.parseInt(pool)+1)+" 1","1");
            ingresa_delay("delay_pools
"+String.valueOf(Integer.parseInt(pool)+1),"delay_pools
"+String.valueOf(Integer.parseInt(pool)),"0");
            ingresa_delay("delay_parameters
"+String.valueOf(Integer.parseInt(pool)+1)+" "+numero+"/8192","#-
fin_delay_parameters","2");
            ingresa_delay("delay_access
"+String.valueOf(Integer.parseInt(pool)+1)+" allow
"+objeto,"#*fin_delay_access","2");
        }else{
            ingresa_delay("delay_parameters
"+String.valueOf(Integer.parseInt(pool))+"
"+numero+"/8192","delay_parameters
"+String.valueOf(Integer.parseInt(pool)),"0");
        }

    }else{
        ingresa_delay("delay_pools 1","#+fin_delay","2");
        ingresa_delay("delay_class 1 1","#+fin_delay","2");
    }
}

```

```

        ingresa_delay("delay_parameters 1 "+numero+"/8192","#-
fin_delay_parameters", "2");
        ingresa_delay("delay_access 1 allow
"+objeto, "#*fin_delay_access", "2");
    }
    try {
        this.conectar();
        this.mantenimiento("update configuracion set
pools=pools+1");
        this.desconectar();
    } catch (SQLException ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
//REGLAS DEL PROXY
//Permite ingresar las reglas en el archivo de configuración

    public void ingresa_reglas(String accion, String objetos,String
nombre_regla){
        String numero ="";
        try {
            this.conectar();
            numero = Consultas_objeto("select reglas as objeto from
configuracion");
            this.mantenimiento("update configuracion set
reglas=reglas+1");
            this.desconectar();
        } catch (SQLException ex) {
            Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (Exception ex) {
            Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }

    if(busca_objeto_reglas(nombre_regla).booleanValue()==false){
        ingresa_delay("http_access allow
"+objetos, "#fin_reglas", "2");
    }else{
        if(busca_objeto_reglas(objetos).booleanValue()==false){
            ingresa_delay(objetos, "http_access allow
"+nombre_regla, "4");
        }
    }
}

```

```

        ingresa_delay(objetos,"http_access allow
!" + nombre_regla,"4");
    }
}

}

//Permite consultar las reglas en el archivo de configuración

public String consulta_reglas(){
    String contenido=" ";
    int i=1;
    try {
        this.conectar();
        String archivo = Consultas_objeto("select archivo as
objecto from configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        int e=0;
        entrada = new BufferedReader( new FileReader( f )
);
        while(entrada.ready()){
            linea = entrada.readLine();

            if(linea.indexOf("http_access")==0 ){
                StringTokenizer tokens = new
StringTokenizer(linea," ");

                String nombre=tokens.nextToken();
                String tipo=tokens.nextToken();
                String objeto=tokens.nextToken();
                if(tokens.hasMoreElements()){
                    objeto=objeto.concat("
" + tokens.nextToken());
                }

                String nom=objeto.substring(0,
objeto.indexOf("_"));

                if(nom.indexOf("!")>=0){
                    nom=objeto.substring(1, nom.length());
                }
                contenido=contenido.concat("<tr
id='fila'+String.valueOf(e)+' '
onMouseOver='enciende('"+String.valueOf(e)+"',1) '
onMouseOut='apaga('"+String.valueOf(e)+"',1)'>");
                contenido=contenido.concat("<td><input
type='radio' name='codigo' value='1' id='RadioGroup1_0'

```



```

onClick='elementos_reglas("'" + nom + "'" ) '
/></td><td>" + String.valueOf(i) + "</td><td>" + nom + "</td><td>" + objeto + "</td>
</tr>\n");

        e++;
        i++;
    }else{
        if(linea.indexOf("#fin_reglas")>=0) {
        }
    }
}
    entrada.close();
} catch (Exception ex) {
    Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
}
    return contenido;
}
//Realiza la llamada a función elimina_regla

    public void elimina_reglas(String numero){
        elimina_regla(numero);
    }

    public void modifica_reglas(String accion, String objetos,String
numero)
    {
        int num1=Integer.parseInt(numero);
        try {
            this.conectar();
            String contenido="";
            String archivo = Consultas_objeto("select archivo as
objecto from configuracion");
            this.desconectar();
            File f = new File(archivo);
            BufferedReader entrada;
            String linea="";
            int i=0;
            entrada = new BufferedReader( new FileReader( f )
);
            while(entrada.ready()){
                linea = entrada.readLine();
                int num= linea.indexOf("http_access");
                if(num>=0) {
                    if(i==num1) {

contenido=contenido.concat("http_access "+accion+" "+objetos+"\n");
                    }else{

contenido=contenido.concat(linea+"\n");

```

```

        }
        i++;
    }else{
        contenido=contenido.concat(linea+"\n");
    }
    }
    entrada.close();
    crea_archivos(archivo,contenido);
} catch (SQLException ex1) {
    Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex1);
} catch (Exception ex) {
    Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
}
}

//Permite eliminar las reglas en el archivo de configuración

public void elimina_regla(String numero){
    try {
        this.conectar();
        String contenido="";
        String archivo = Consultas_objeto("select archivo as
objecto from configuracion");
        this.desconectar();
        File f = new File(archivo);
        BufferedReader entrada;
        String linea="";
        int i=0;
        entrada = new BufferedReader( new FileReader( f )
);

        while(entrada.ready()){
            linea = entrada.readLine();
            int num= linea.indexOf(numero);
            if(num<0){
                contenido=contenido.concat(linea+"\n");
            }
        }
        entrada.close();
        crea_archivos(archivo,contenido);
    } catch (SQLException ex1) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex1);
    } catch (Exception ex) {
        Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
}

```

```
//Permite modificar el firewall

    public void modifica_firewall(String antiguo_puerto,String
nuevo_puerto){
        try {
            this.conectar();
            String contenido="";
            String archivo = Consultas_objeto("select firewall as
objecto from configuracion");
            this.desconectar();
            File f = new File(archivo);
            BufferedReader entrada;
            String linea="";
            entrada = new BufferedReader( new FileReader( f )
);
            while(entrada.ready()){
                linea = entrada.readLine();
                int num=linea.indexOf(antiguo_puerto);
                if(num>=0){

contenido=contenido.concat(nuevo_puerto+"\n");
                }else{
                    contenido=contenido.concat(linea+"\n");
                }
            }
            entrada.close();
            crea_archivos(archivo,contenido);
            ( new Ejecutas_Procesos()).restart_firewall();
        } catch (SQLException ex1) {
            Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex1);
        } catch (Exception ex) {
            Logger.getLogger(Objetos.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
}
```

## CLASE “Permisos”

Esta clase proporciona los accesos necesarios para los usuarios que operan la aplicación del proxy.

```
import java.sql.*;

//Verifica los permisos que tienen los usuarios

public class Permisos extends Conexion {
    CallableStatement proceso = null;

    public Boolean verifica_permisos(String nombre){
        Boolean objeto=null;
        try{
            conectar();
            objeto= verificaconsultas("select * from permisos where
nombre='"+nombre+"'");
            desconectar();
        }catch (java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch (Exception tres ){
            System.out.println(tres.getMessage());
        }
        return objeto;
    }

    //Verifica si el usuario tiene permisos

    public Boolean permisos_usuarios(String codigo){
        Boolean objeto=null;
        try{
            conectar();
            objeto= verificaconsultas("select * from usuarios where
permiso="+codigo);
            desconectar();
        }catch (java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch (Exception tres ){
            System.out.println(tres.getMessage());
        }
        return objeto;
    }
}
```

```

    }
    public Boolean verificar_permisos(String nombre,String codigo){
        Boolean objeto=null;
        try{
            conectar();
            objeto= verificaconsultas("select * from permisos where
nombre='"+nombre+"' and not(codigo='"+codigo+"' )");
            desconectar();
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            System.out.println(tres.getMessage());
        }
        return objeto;
    }

//Ingresa permisos a los usuarios

    public void ingresar_permisos(String nombre,String op1,String
op2,String op3,String op4,String op5,String op6,String op7,String
op8,String op9,String op10,String op11){
        try{
            conectar();
            proceso=con.prepareCall("{ Call
MANTENIMIENTO_PERMISOS(?,?,?,?,?,?,?,?,?,?,?,?,?)}");
            proceso.setInt(1,(new Integer(op1)).intValue());
            proceso.setInt(2,(new Integer(op2)).intValue());
            proceso.setInt(3,(new Integer(op3)).intValue());
            proceso.setInt(4,(new Integer(op4)).intValue());
            proceso.setInt(5,(new Integer(op5)).intValue());
            proceso.setInt(6,(new Integer(op6)).intValue());
            proceso.setInt(7,(new Integer(op7)).intValue());
            proceso.setInt(8,(new Integer(op8)).intValue());
            proceso.setInt(9,(new Integer(op9)).intValue());
            proceso.setInt(10,(new Integer(op10)).intValue());
            proceso.setInt(11,(new Integer(op11)).intValue());
            proceso.setString(12,nombre);
            proceso.setInt(13,0);
            proceso.setInt(14,1);
            proceso.execute();
            proceso.close();
        }catch(java.sql.SQLException dos){
            dos.printStackTrace();
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            tres.printStackTrace();
            System.out.println(tres.getMessage());
        }
    }
}

```

```
//Elimina permisos a los usuarios
```

```
public void eliminar_permisos(String codigo){
    try{
        conectar();
        proceso=con.prepareStatement("{ Call
MANTENIMIENTO_PERMISOS(?,?,?,?,?,?,?,?,?,?,?,?,?) }");
        proceso.setInt(1,0);
        proceso.setInt(2,0);
        proceso.setInt(4,0);
        proceso.setInt(3,0);
        proceso.setInt(5,0);
        proceso.setInt(6,0);
        proceso.setInt(7,0);
        proceso.setInt(8,0);
        proceso.setInt(9,0);
        proceso.setInt(10,0);
        proceso.setInt(11,0);
        proceso.setString(12, "");
        proceso.setInt(13,Integer.parseInt(codigo));
        proceso.setInt(14,3);

        proceso.execute();
        proceso.close();
    }catch(java.sql.SQLException dos){
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        System.out.println(tres.getMessage());
    }
}
```

```
//Modifica permisos a los usuarios
```

```
public void modificar_permisos(String nombre,String op1,String
op2,String op3,String op4,String op5,String op6,String op7,String
op8,String op9,String op10,String op11,String codigo){
    try{
        conectar();
        proceso=con.prepareStatement("{ Call
MANTENIMIENTO_PERMISOS(?,?,?,?,?,?,?,?,?,?,?,?,?) }");
        proceso.setInt(1,(new Integer(op1)).intValue());
        proceso.setInt(2,(new Integer(op2)).intValue());
        proceso.setInt(3,(new Integer(op3)).intValue());
        proceso.setInt(4,(new Integer(op4)).intValue());
        proceso.setInt(5,(new Integer(op5)).intValue());
        proceso.setInt(6, (new Integer(op6)).intValue());
        proceso.setInt(7, (new Integer(op7)).intValue());
        proceso.setInt(8, (new Integer(op8)).intValue());
```

```

        proceso.setInt(9, (new Integer(op9)).intValue());
        proceso.setInt(10, (new Integer(op10)).intValue());
        proceso.setInt(11, (new Integer(op11)).intValue());
        proceso.setString(12, nombre);
        proceso.setInt(13, (new Integer(codigo).intValue() ));
        proceso.setInt(14,2);
        proceso.execute();
        proceso.close();
    }catch(java.sql.SQLException dos){
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        System.out.println(tres.getMessage());
    }
}
}

```

### CLASE “Proceso Capturador”

Clase que captura los datos desde Jnettop para proporcionar un reporte de consulmo de red, en base a requerimientos de los clientes y respuestas del servidor.

```

import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

//Registra la actividad de la red interna

public class Proceso_Capturador extends Thread {
    Clases.Ejecutas_Procesos consola=new Clases.Ejecutar_Procesos();
    Clases.Conexion conexion= new Clases.Conexion();

    public void run () {
        try {
            conexion.conectar();
            do{
                System.out.println("entro");
                java.util.Vector consultas=consola.ejecuta_jnettop();
                if(consultas.size()>0){
                    for(int i=0;i<consultas.size();i++){
                        java.util.Vector
d=(java.util.Vector)consultas.get(i);

```

```

        if(d.get(3).toString().indexOf("::")>0){
            conexion.mantenimiento("insert into
reporte
(ip_origen,p_origen,bytes_tras,ip_des,p_des,bytes_rec,fecha,hora)values
('"+d.get(0).toString()+"','"+d.get(1).toString()+"','"+d.get(2).toStrin
g()+"','"+d.get(3).toString()+"','"+d.get(4).toString()+"','"+d.get(5).to
String()+"',curdate(),curtime())");
        }
    }
    }
    this.sleep(5000);
}while(true);
} catch (ClassNotFoundException ex) {

Logger.getLogger(Proceso_Capturador.class.getName()).log(Level.SEVERE,
null, ex);

    } catch (SQLException ex) {

Logger.getLogger(Proceso_Capturador.class.getName()).log(Level.SEVERE,
null, ex);

    } catch (Exception ex) {

Logger.getLogger(Proceso_Capturador.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

```



## CLASE “Usuarios”

La clase Usuarios manipula los objetos referentes a los usuarios del sistema, para así generar los procesos de mantenimiento de usuarios de la aplicación.

```
package Clases;
import java.sql.*;
import java.util.Vector;

public class Usuarios extends Conexion {
    CallableStatement proceso = null;

    public Boolean verifica_usuarios(String usuario){
        Boolean objeto=null;
        try{
            conectar();
            objeto= verificaconsultas("select * from usuarios where
usuario='"+usuario+"'");
            desconectar();
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            System.out.println(tres.getMessage());
        }
        return objeto;
    }
    public Boolean verificar_usuarios(String usuario,String codigo){
        Boolean objeto=null;
        try{
            conectar();
            objeto= verificaconsultas("select * from usuarios where
usuario='"+usuario+"' and not(codigo='"+codigo+"' )");
            desconectar();
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            System.out.println(tres.getMessage());
        }
        return objeto;
    }
    public void ingresar_usuarios(String nombre,String apellido,String
celular,String usuario, String clave,String permiso,String correo){
```

```

        try{
            conectar();
            proceso=con.prepareStatement("{ Call
MANTENIMIENTO_USUARIOS(?,?,?,?,?,?,?,?,?) }");
            proceso.setString(1,nombre);
            proceso.setString(2,apellido);
            proceso.setString(4,celular);
            proceso.setString(3,usuario);
            proceso.setString(5,clave);
            proceso.setInt(6, (new Integer(permiso).intValue()));
            proceso.setInt(7, 0);
            proceso.setInt(8, 1);
            proceso.setString(9,correo);
            proceso.execute();
            proceso.close();
            desconectar();
        }catch(java.sql.SQLException dos){
            dos.printStackTrace();
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            tres.printStackTrace();
            System.out.println(tres.getMessage());
        }
    }
    public void eliminar_usuarios(String codigo){

        try{
            conectar();
            proceso=con.prepareStatement("{ Call
MANTENIMIENTO_USUARIOS(?,?,?,?,?,?,?,?,?) }");
            proceso.setString(1,"");
            proceso.setString(2,"");
            proceso.setString(3,"");
            proceso.setString(4,"");
            proceso.setString(5,"");
            proceso.setInt(6, 0);
            proceso.setInt(7, Integer.parseInt(codigo));
            proceso.setInt(8, 3);
            proceso.setString(9,"");
            proceso.execute();
            proceso.close();
            desconectar();
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            System.out.println(tres.getMessage());
        }
    }
}

```

```

    public void modificar_usuarios(String nombre,String apellido,String
celular,String usuario, String clave,String codigo,String
permiso,String correo){
        try{
            conectar();
            proceso=con.prepareStatement("{ Call
MANTENIMIENTO_USUARIOS(?,?,?,?,?,?,?,?,?) }");
            proceso.setString(1,nombre);
            proceso.setString(2,apellido);
            proceso.setString(3,usuario);
            proceso.setString(5,clave);
            proceso.setString(4,celular);
            proceso.setInt(6, (new Integer(permiso).intValue()));
            proceso.setInt(7, Integer.parseInt(codigo));
            proceso.setInt(8, 2);
            proceso.setString(9,correo);
            proceso.execute();
            proceso.close();
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            System.out.println(tres.getMessage());
        }
    }
    public String datos_personas(String codigo){
        String nombres="";
        try{
            conectar();
            proceso=con.prepareStatement("{ Call VISTAS(?,?) }");
            proceso.setInt(1,1);
            proceso.setString(2,codigo);
            proceso.execute();
            while(proceso.getResultSet().next()){
                nombres=proceso.getResultSet().getString("nombres")+
"+proceso.getResultSet().getString("apellidos");
            }
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            System.out.println(tres.getMessage());
        }
        return nombres;
    }
    public Vector permisos_usuario(String usuario){
        Vector permiso=new Vector(10);
        try{
            conectar();
            proceso=con.prepareStatement("{ Call VISTAS(?,?) }");
            proceso.setInt(1,2);

```

```

        proceso.setString(2, usuario);
        proceso.execute();
        while (proceso.getResultSet().next()) {

permiso.addElement(proceso.getResultSet().getString("per1"));
permiso.addElement(proceso.getResultSet().getString("per2"));
permiso.addElement(proceso.getResultSet().getString("per3"));
permiso.addElement(proceso.getResultSet().getString("per4"));
permiso.addElement(proceso.getResultSet().getString("per5"));
permiso.addElement(proceso.getResultSet().getString("per6"));
permiso.addElement(proceso.getResultSet().getString("per7"));
permiso.addElement(proceso.getResultSet().getString("per8"));
permiso.addElement(proceso.getResultSet().getString("per9"));
permiso.addElement(proceso.getResultSet().getString("per10"));
permiso.addElement(proceso.getResultSet().getString("per11"));

        }
        desconectar();
    } catch (java.sql.SQLException dos) {
        System.out.println(dos.getMessage());
    } catch (Exception tres) {
        System.out.println(tres.getMessage());
    }
    return permiso;
}
}

```

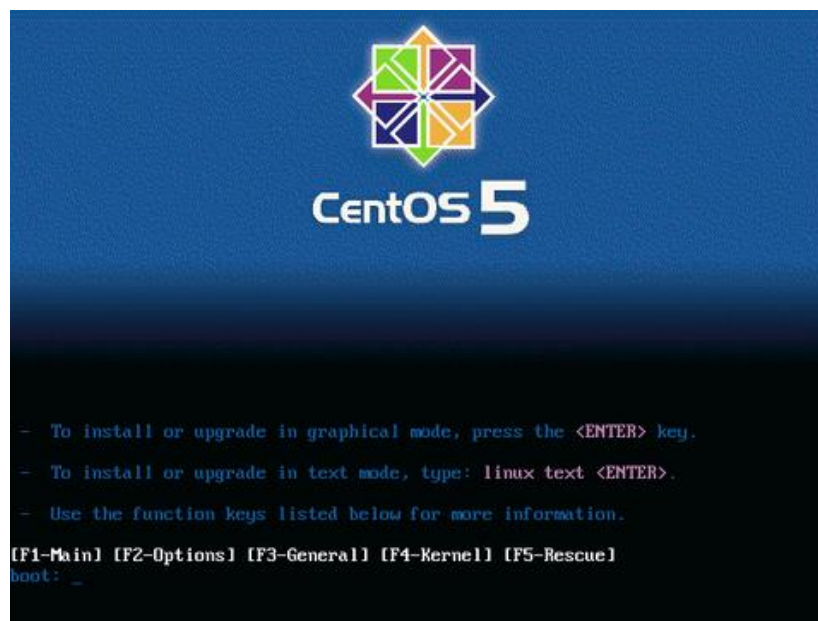


## ANEXO 1

### INSTALACION LINUX CENTOS 5.2

#### SISTEMA OPERATIVO

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



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

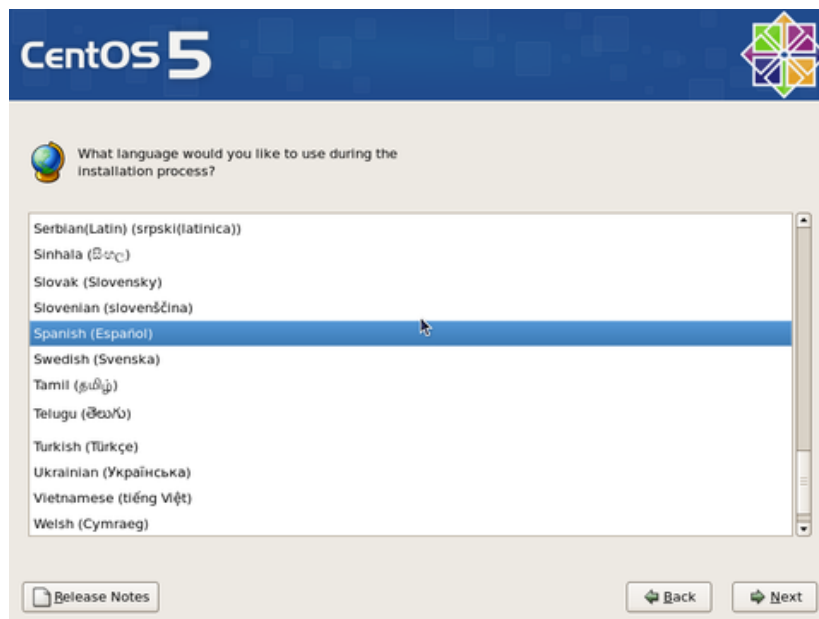
cuales se realizará la instalación están en buen estado, seleccione «**Skip**» y pulse la tecla **ENTER**.



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

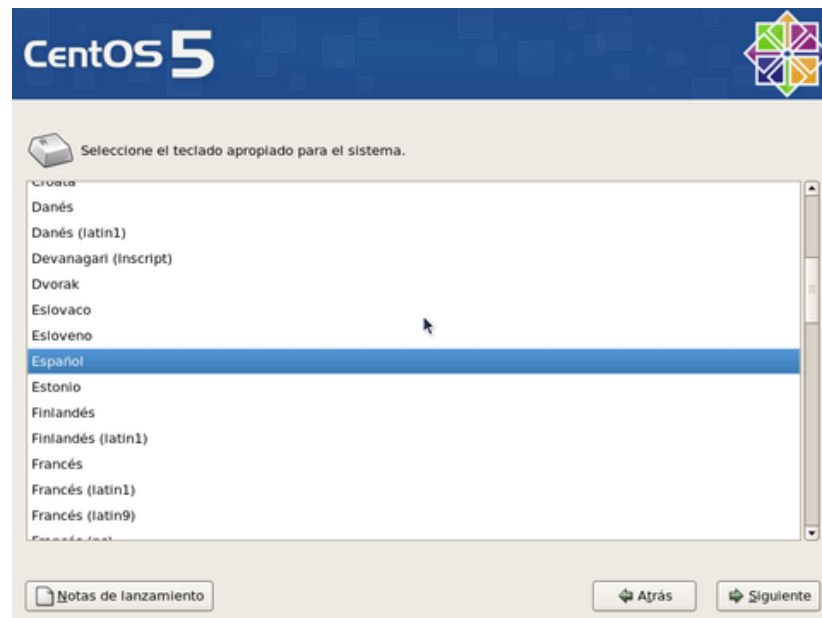


Seleccione «**Spanish**» como idioma para ser utilizado durante la instalación.



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



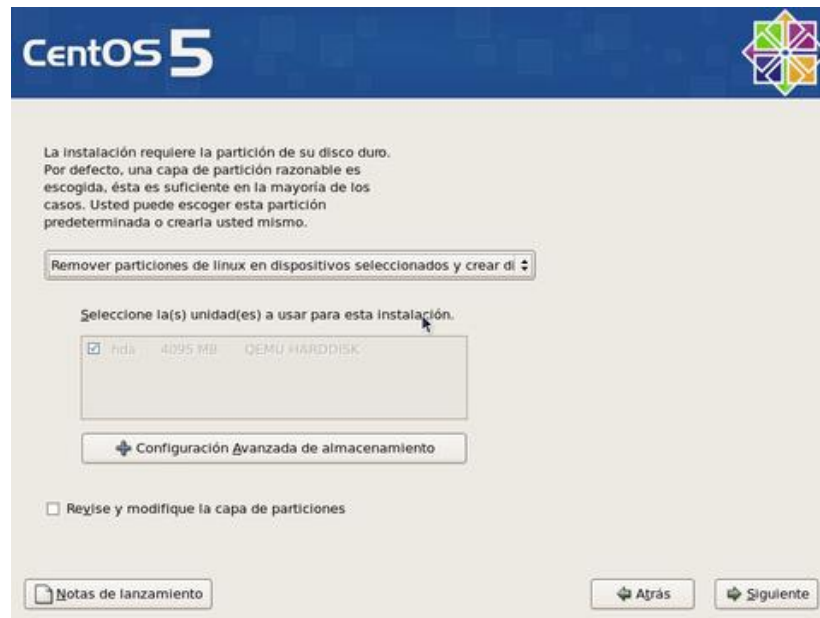


Salvo que exista una instalación previa que se desee actualizar (no recomendado), deje seleccionado «**Instalar CentOS**» y haga clic en el botón «**Siguiente**» a fin de realizar una instalación nueva.

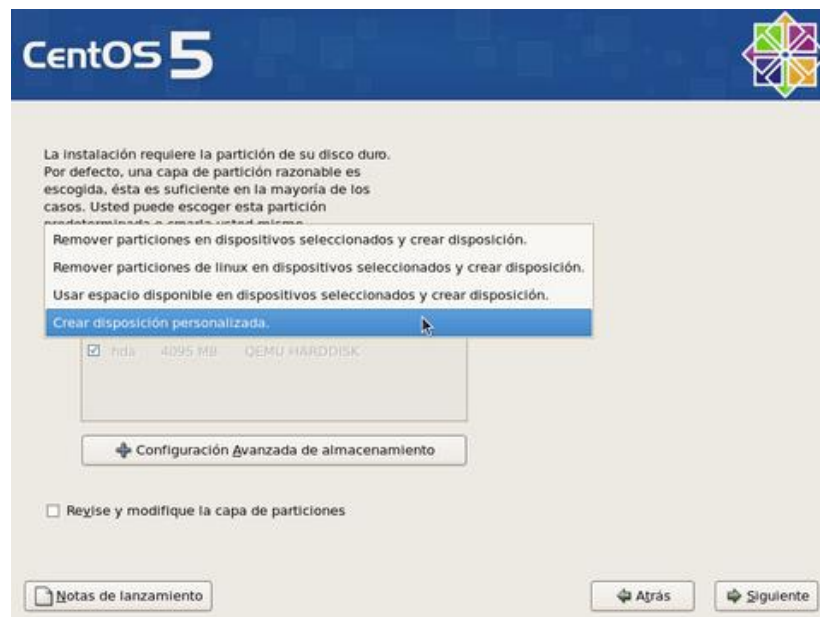


Para crear las particiones de forma automática, lo cual puede funcionar para la mayoría de los usuarios, puede seleccionar:

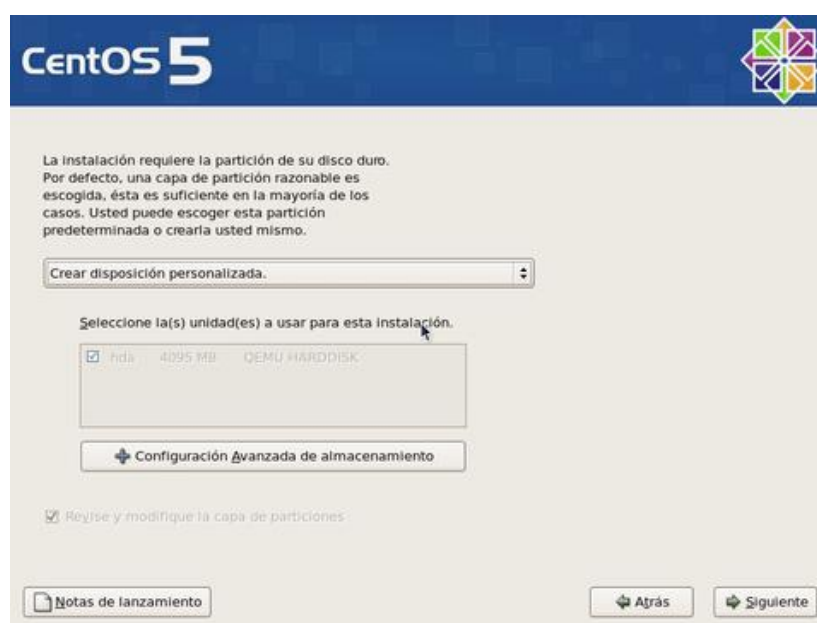
- **«Remove particiones en dispositivos seleccionados y crear disposición»**, lo cual eliminaría cualquier partición de cualquier otro sistema operativo presente, y creará de forma automática las particiones necesarias.
- **«Remove particiones de linux en dispositivos seleccionados y crear disposición»**,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



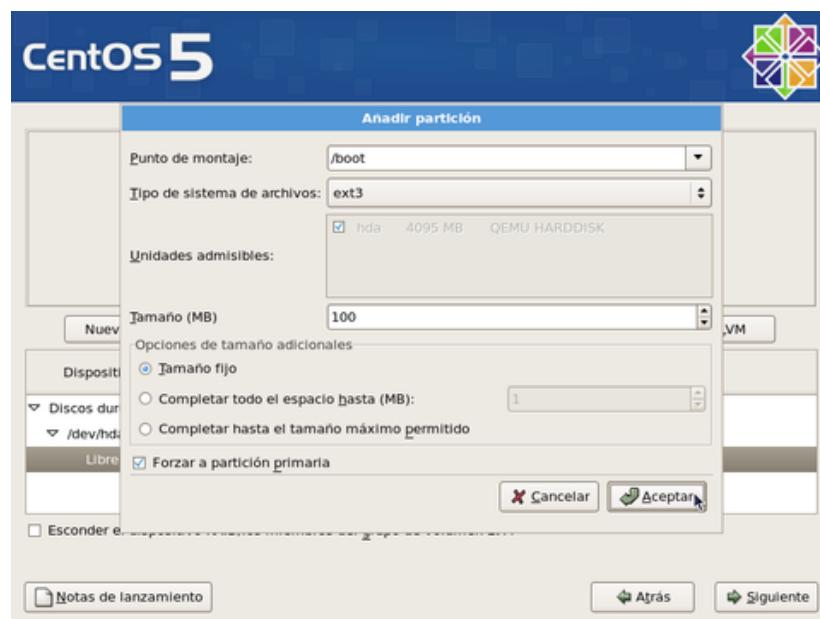
Conviene crear una disposición que permita un mayor control. Seleccione **«Crear disposición personalizada»**.



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



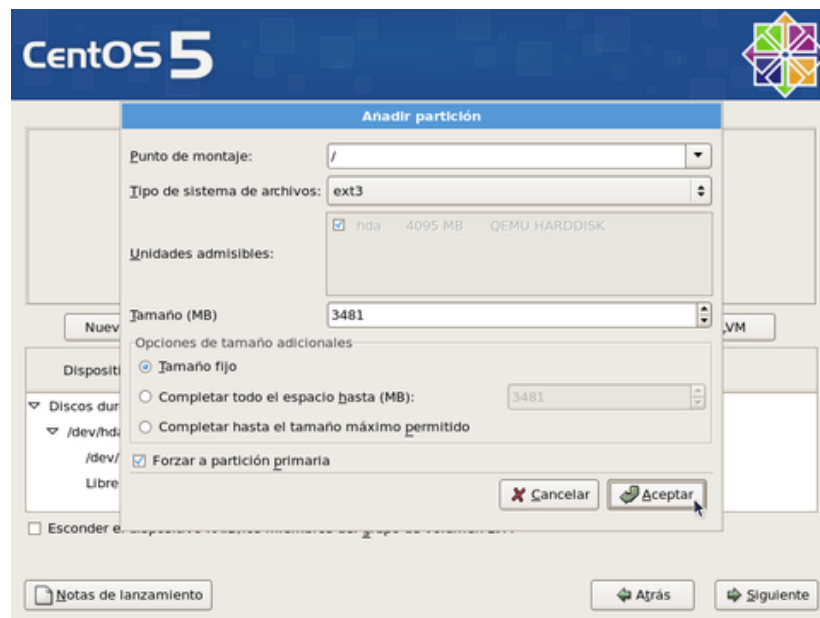
La herramienta de particiones mostrará el espacio disponible. Haga clic en el botón «**Nuevo**».



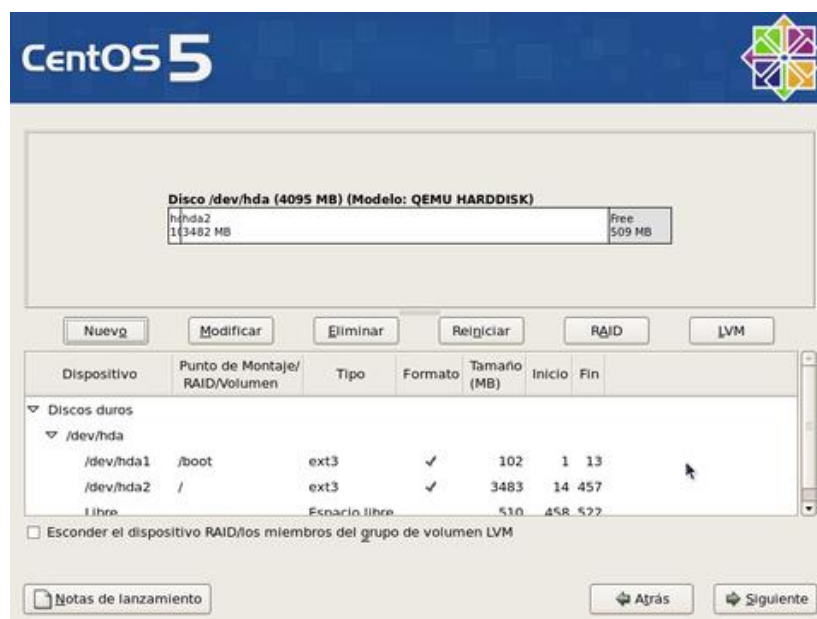
Si está conforme, haga clic otra vez en el botón «**Nuevo**» y proceda a crear la siguiente partición.



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

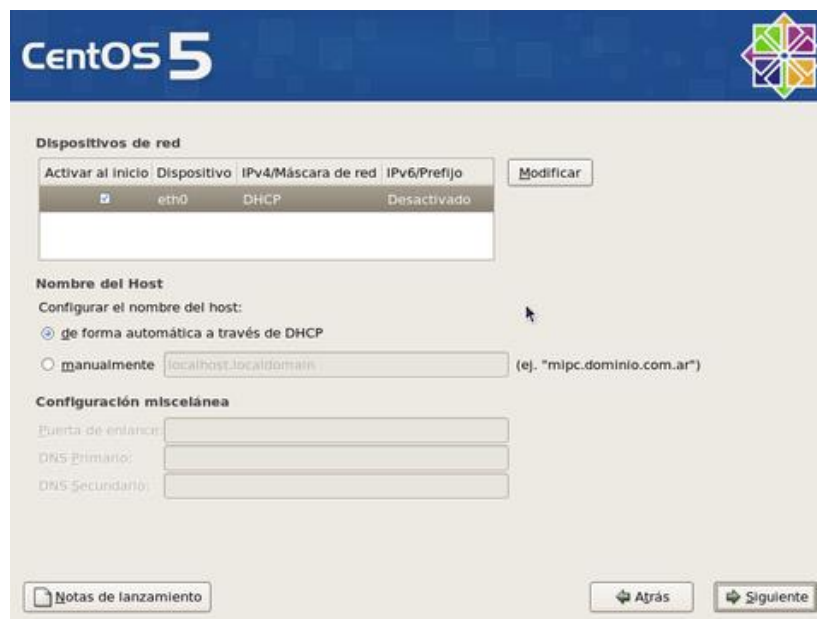


Si está conforme, haga clic otra vez en el botón «**Nuevo**» y proceda a crear la siguiente partición.



Crear 4 particiones de la siguiente manera:

/boot	Ext3	100MB
/cache	Ext3	16000MB (El tamaño del disco duro multiplicado por 200).
/swap	swap	1024 (El doble de la RAM).
/	Ext3	El resto de espacio que sobra.



The image shows the CentOS 5 Network Configuration window. At the top, there's a blue header with the 'CentOS 5' logo and a colorful geometric icon. Below the header, the window is titled 'Dispositivos de red'. It contains a table with columns: 'Activar al inicio', 'Dispositivo', 'IPv4/Máscara de red', 'IPv6/Prefijo', and a 'Modificar' button. The table has one row for 'eth0' with 'DHCP' and 'Desactivado' settings. Below the table, there's a section 'Nombre del Host' with a label 'Configurar el nombre del host:' and two radio buttons: 'de forma automática a través de DHCP' (selected) and 'manualmente'. The 'manualmente' option has a text input field containing 'localhost.localdomain' and a hint '(ej. "mipc.dominio.com.ar")'. Below this is a section 'Configuración miscelánea' with three text input fields: 'Puerta de enlace', 'DNS Primario', and 'DNS Secundario'. At the bottom left is a button 'Notas de lanzamiento' and at the bottom right are 'Atrás' and 'Siguiente' buttons.

Debe existir dos interfaces de red dentro del servidor la eth0 para la salida hacia internet y la eth1 para la comunicación con la red LAN.

**ETH0:** Utilizar la configuración de IP Dinamica (DHCP).

Activar Soporte para IPV4.

Activar Soporte para IPV6.

Activar al Inicio.

**ETH1:** No Utilizar la configuración de IP Dinamica (DHCP).

Configuración manual IPV4: 192.168.1.1 / 255.255.255.0

Activar Soporte para IPV4.

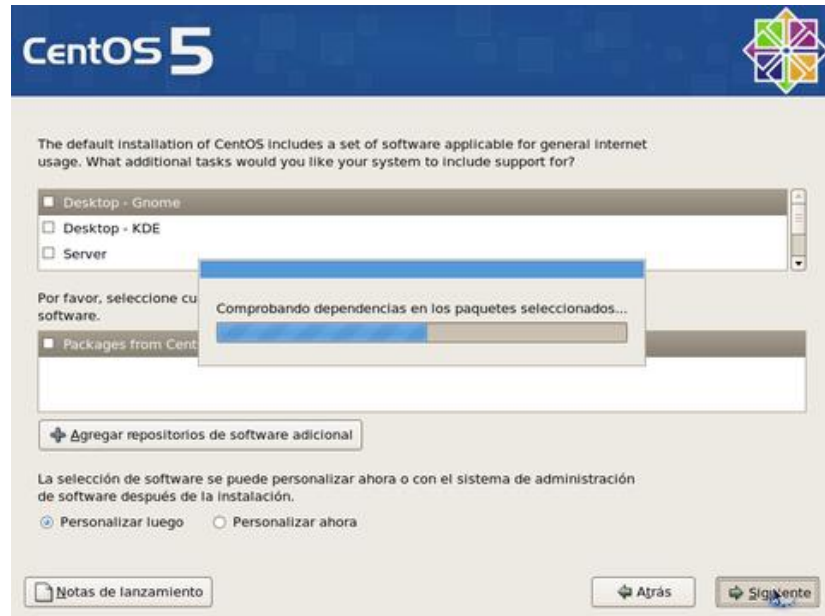
Activar Soporte para IPV6.

Activar al Inicio.

Nombre del Host: Configuración manual el nombre de la máquina.

**Selección de Paquetes**





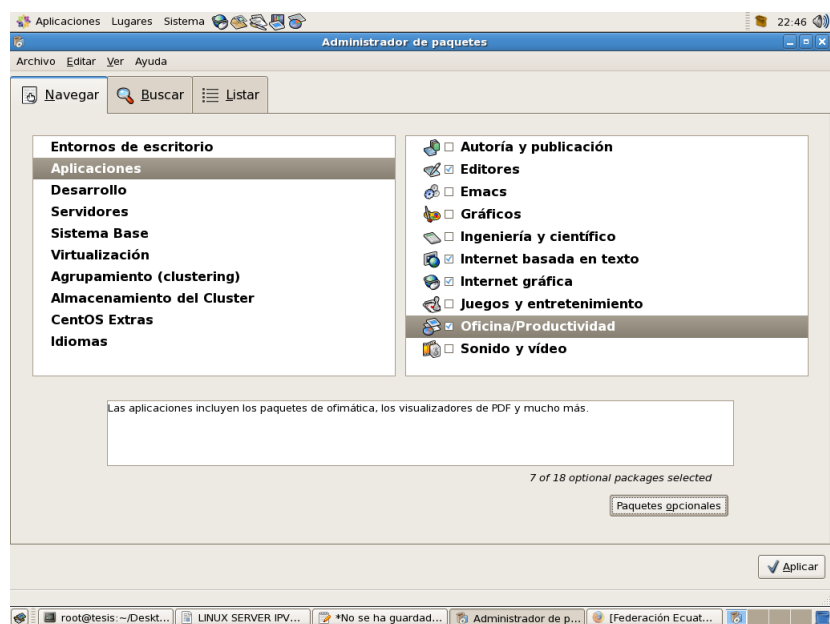
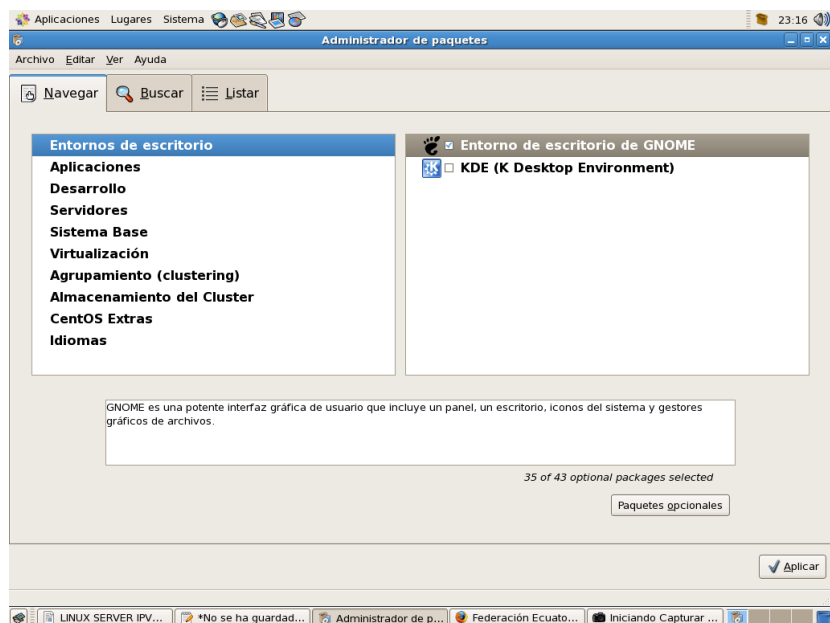
Seleccionar las siguientes categorías:

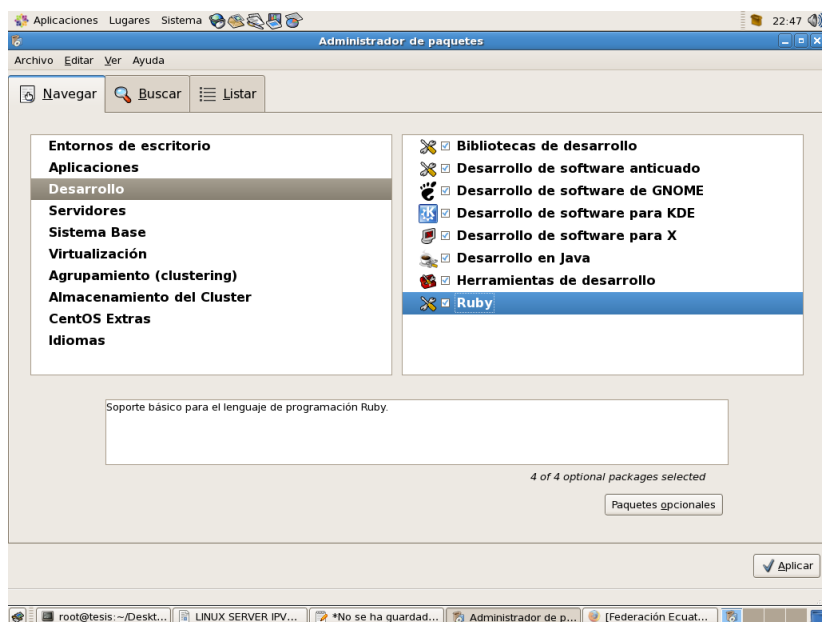
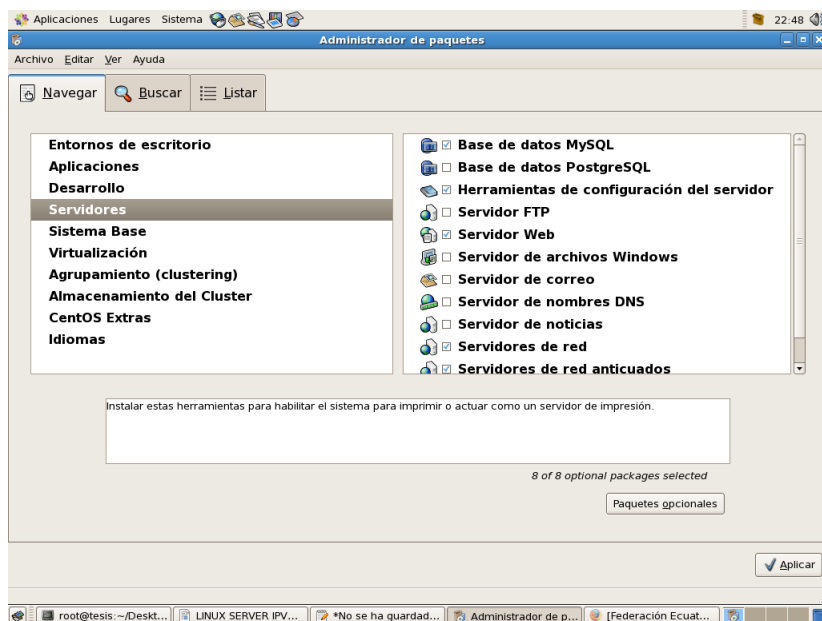
Desktop – Gnome

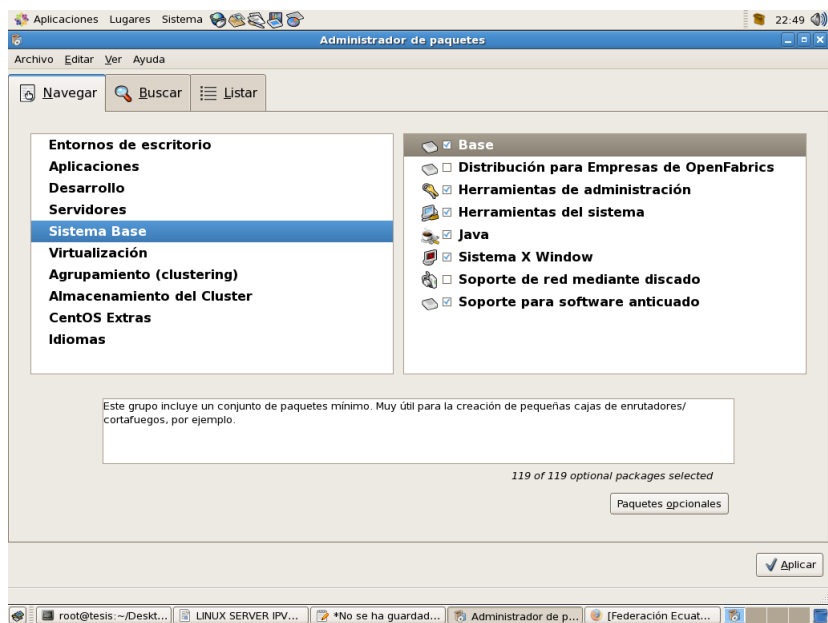
Server (Seleccionar todos excepto FTP Sever, Servidor de Impresora).

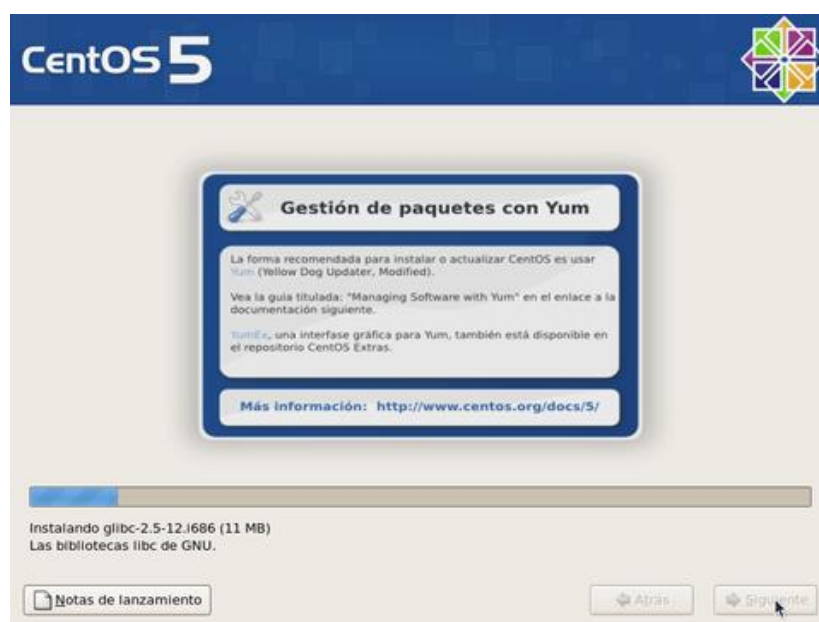
Server – GUI

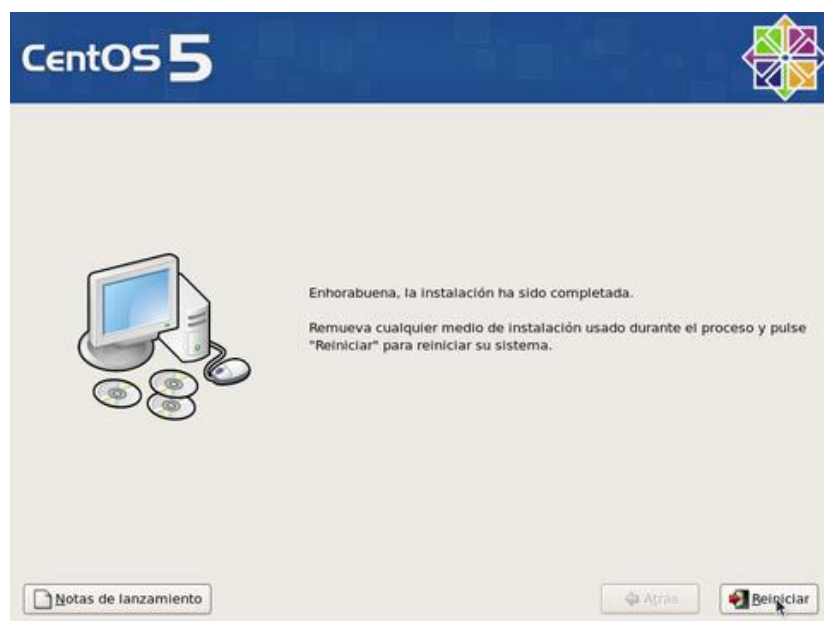
Desarrollo











Una vez instalado el Sistema Operativo al reiniciar van aparecer algunas ventanas de configuración en este punto **NO** habilitar ni el firewall ni el SELinux.

## Actualización del Sistema Operativo

Una vez ingresado al sistema ejecutar el siguiente comando:

```
#yum -y upgrade
```

## Instalación del SQUID

```
#tar -zxof squid-3.1.0.13.tar.gz
#cd squid-3.1.0.13
```

```
#!/configure --prefix=/opt/squid --exec-prefix=/opt/squid --enable-delay-pools --enable-cache-digests --enable-poll --disable-ident-lookups --enable-truncate --enable-removal-policies --enable-follow-x-forwarded-for --enable-snmp --enable-arp-acl --enable-linux-netfilter --enable-epoll --with-ipv6-split-stack
```

```
#make all
#make install
```

```
#nano /opt/squid/etc/squid.conf
```

```
http_port 3128
icp_port 3130
#cgi-bins will not be cached.
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
#Memory the Squid will use. Well, Squid will use far more than that.
cache_mem 64 MB
#250 means that Squid will use 250 megabytes of disk space.
cache_dir ufs /cache 10240 16 256
cache_effective_user squid
```

```

#Places where Squid's logs will go to.
#cache_log /var/log/squid/cache.log
cache_access_log /var/log/squid/access.log
#cache_store_log /var/log/squid/store.log
#cache_access_log none
cache_store_log none
cache_swap_log /var/log/squid/swap.log
#How many times to rotate the logs before deleting them.
#See the FAQ for more info.
logfile_rotate 10

acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8
acl localnet src 10.0.0.0/8 # RFC1918 possible internal network
acl localnet src 172.16.0.0/12 # RFC1918 possible internal network
acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
acl prueba src fe80::e437:fa5e:4baa:8e38
acl prueba src ::1
acl prueba src fe80::fd05:b010:b174:9811

acl SSL_ports port 443
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http

acl CONNECT method CONNECT
http_access allow prueba
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports

```



```

http_access deny CONNECT !SSL_ports
http_access allow localnet
http_access allow localhost
http_access deny all

hierarchy_stoplist cgi-bin ?
coredump_dir /opt/squid/var/cache

refresh_pattern ^ftp:      1440  20%  10080
refresh_pattern ^gopher:   1440  0%   1440
refresh_pattern -i (/cgi-bin/|\?) 0    0%   0
refresh_pattern .          0      20%  4320

visible_hostname tesis.universidad.com

```

### **Configuración del Firewall para IPV6**

```

#yum -y install iptables-ipv6
# cd /etc/rc.d/
#nano firewallipv6

#!/bin/bash
#DEFINICION DEL COMANDO IPTABLES
IPT6="/sbin/ip6tables"

#DEFINICION DE LA VARIABLE INTERFAZ ETH1
PUBIF="eth1"

echo "Starting IPv6 firewall..."

# BORRADO DE LAS CADENAS
$IPT6 -F
$IPT6 -X
$IPT6 -t mangle -F
$IPT6 -t mangle -X

#SIN LIMITES EL ACCESO DEL LOOPBACK

```

```
$IPT6 -A INPUT -i lo -j ACCEPT
$IPT6 -A OUTPUT -o lo -j ACCEPT
```

```
# DENEGAR TODO EL TRAFICO ENTRANTE
```

```
$IPT6 -P INPUT DROP
$IPT6 -P OUTPUT DROP
$IPT6 -P FORWARD DROP
```

```
# HABILITAR LAS CONEXIONES SALIENTES YA ESTABLECIDAS
```

```
$IPT6 -A INPUT -i $PUBIF -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT6 -A OUTPUT -o $PUBIF -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

```
# HABILITAR EL PROTOCOLO ICMP VERSION 6 (PING IPV6)
```

```
$IPT6 -A INPUT -i $PUBIF -p ipv6-icmp -j ACCEPT
$IPT6 -A OUTPUT -o $PUBIF -p ipv6-icmp -j ACCEPT
#ACEPTAR LAS CONEXIONES ENTRANTES IPV6 sobre IPV4 en LA ETH0
```

```
iptables -A INPUT -i eth0 -p ipv6 -j ACCEPT
```

```
#ACEPTAR LAS CONEXIONES SALIENTES IPV6 SOBRE IPV4 EN LA ETH0
```

```
iptables -A OUTPUT -o eth0 -p ipv6 -j ACCEPT
```

```
### HABILITAR PUERTO 80 SOBRE IPV6 NAVEGACION
```

```
$IPT6 -A INPUT -i $PUBIF -p tcp --destination-port 80 -j ACCEPT
$IPT6 -A INPUT -i $PUBIF -p tcp --destination-port 3128 -j ACCEPT
### HABILITAR PUERTO 22 sobre IPV6 CONEXIONES REMOTAS SSH
```

```
$IPT6 -A INPUT -i $PUBIF -p tcp --destination-port 22 -j ACCEPT
```

```
### HABILITAR PUERTO 25 SOBRE IPV6 MAILS
```

```
$IPT6 -A INPUT -i $PUBIF -p tcp --destination-port 25 -j ACCEPT
```

### ESTA ES LA LINEA PARA EL PROXY TRANSPARENTE LA CUAL NO ESTA SOPORTADA

```
# $IPT6 -t nat -A PREROUTING -i $PUBIF -p tcp --dport 80 -j REDIRECT --port 8080
```

# REGISTROS LAS CONEXIONES ENTRANTES Y DENEGAR POR DEFECTO

```
$IPT6 -A INPUT -i $PUBIF -j LOG
```

```
$IPT6 -A INPUT -i $PUBIF -j DROP
```

#Aquí fin

```
# chmod ugo+x firewallip6
```

```
# ./firewallip6
```

### Configuración del Firewall para IPV4

```
# cd /etc/rc.d/
```

```
# nano firewallip4
```

```
#!/bin/sh
```

```
PROG=/sbin/iptables
```

```
EXTIF=eth0
```

```
INTIF=eth1
```

```
RED="red privada del cliente"
```

```
INTNET="192.168.0.0/24"
```

```
UNIVERSE="0.0.0.0/0"
```

```
#IP="IP DEL SERVIDOR"
```

##LINEA QUE CAPTURA LA IP DE LA TARJETA DE RED ###

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

```
$PROG -F
```

```
$PROG -t nat -F
```

#####LINEA DE PROXY TRANSPARENTE#####

```
$PROG -t nat -A PREROUTING -i $INTIF -p tcp --dport 80 -j REDIRECT --to-port 3128
```

####NAT PARA LOS PUERTOS 25, 110, 443, 53####

```
$PROG -t nat -A POSTROUTING -o $EXTIF -p TCP --dport 25 -j SNAT --to-source $IP
```

```
$PROG -t nat -A POSTROUTING -o $EXTIF -p TCP --dport 110 -j SNAT --to-source $IP
```

```
$PROG -t nat -A POSTROUTING -o $EXTIF -p TCP --dport 443 -j SNAT --to-source $IP
```

```
$PROG -t nat -A POSTROUTING -o $EXTIF -p TCP --dport 53 -j SNAT --to-source $IP
```

```
$PROG -t nat -A POSTROUTING -o $EXTIF -p UDP --dport 53 -j SNAT --to-source $IP
```

#####NAT PARA VIP#####

```
#$PROG -t nat -A POSTROUTING -o eth0 -s $IP_DE_CLIENTE -j SNAT --to-source $IP
```

```
$PROG -A INPUT -s localhost -j ACCEPT
```

```
$PROG -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
$PROG -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

####ACCESO DENEGADOS####

```
$PROG -A INPUT -i $EXTIF -p tcp --dport 8080 -j DROP
```

```
$PROG -A INPUT -i $EXTIF -p tcp --dport 3128 -j DROP
```

```
$PROG -A INPUT -i $EXTIF -p tcp --dport 53 -j DROP
```

```
$PROG -A INPUT -i $EXTIF -p udp --dport 53 -j DROP
```

```
# chmod ugo+x firewallipv4
```

```
# ./firewallipv4
```

### **Activación del Forward para IPV4**

```
#nano /etc/sysctl.conf
```

```
# Kernel sysctl configuration file for Red Hat Linux
```

```
#
```

```
# For binary values, 0 is disabled, 1 is enabled. See sysctl(8) and
```

```
# sysctl.conf(5) for more details.
```

```
net.ipv6.conf.all.forwarding = 1
```

```
# Controls IP packet forwarding
net.ipv4.ip_forward = 1

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

# Controls the use of TCP syncookies
net.ipv4.tcp_syncookies = 1
```

### **Instalación del paquete SARG para la creación de reportes de navegación**

```
#tar -zxof sarg-2.2.2.tar.gz
#cd sarg-2.2.2
#./configure
#make all
#make install

#nano /usr/local/sarg/sarg.conf

access_log /var/log/squid/access.log
output_dir /var/www/html/squid-reports

# mkdir /var/www/html/squid-reports
```

```
# ln -s /usr/local/etc/httpd/htdocs/squid-reports/ squid-reports
#sarg
```

```
#nano /etc/crontab
```

```
59 23 * * 3,5,7 root /usr/bin/sarg
```

### **Configuración del Servicio httpd para levantar el servicio web**

```
SecRule REQUEST_HEADERS:Host "^[\d\.]+$" "deny,log,auditlog,status:400,msg:'Host
header is a numeric IP address', severity:'2',id:'960017',
```

Comentar la línea anterior en estos dos archivos:

```
#nano /etc/httpd/modsecurity.d/modsecurity_crs_21_protocol_anomalies.conf
#nano
/etc/httpd/modsecurity.d/blocking/modsecurity_crs_21_protocol_anomalies.conf

#/etc/init.d/httpd restart
```

### **Instalación del paquete JNETTOP para monitoreo del trafico de red**

```
#yum install libpcap-devel
#tar -xzf jnettop-0.13.0.tar.gz
#cd jnettop-0.13.0
#./configure
#make all
#make install
```

### **Archivos de Configuración necesarios**

```
#nano /etc/sysconfig/network

NETWORKING=yes
HOSTNAME=tesis.universidad.com
NETWORKING_IPV6=yes
```

```
IPV6FORWARDING=yes
IPV6_ROUTER=yes
IPV6_DEFAULTDEV=tun6to4
```

```
#nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
# Intel Corporation 82557/8/9 Ethernet Pro 100
DEVICE=eth0
BOOTPROTO=dhcp
HWADDR=00:03:47:77:c1:e2
IPV6INIT=yes
IPV6_AUTOCONF=yes
ONBOOT=yes
TYPE=Ethernet
PEERDNS=yes
USERCTL=no
```

```
#### LINEAS TUNEL 6TO4 ####
```

```
IPV6TO4INIT=yes
IPV6_CONTROL_RADVD=yes
IPV6TO4_ROUTING="eth1-:1::1/64 wlan0-:2::1/64"
```

```
#nano /etc/sysconfig/network-scripts/ifcfg-eth1
```

```
DEVICE=eth1
BOOTPROTO=none
BROADCAST=192.168.1.255
HWADDR=00:16:76:42:48:a3
IPADDR=192.168.1.1
IPV6INIT=yes
IPV6_AUTOCONF=yes
NETMASK=255.255.255.0
NETWORK=192.168.1.0
ONBOOT=yes
TYPE=Ethernet
PEERDNS=yes
USERCTL=yes
```

```
# nano /etc/rc.d/rc.local
```

```
#!/bin/sh
```

```
#
```

```
# This script will be executed *after* all the other init scripts.
```

```
# You can put your own initialization stuff in here if you don't
```

```
# want to do the full Sys V style init stuff.
```

```
touch /var/lock/subsys/local
```

```
/opt/squid/sbin/squid
```

```
/etc/rc.d/firewallip6
```

```
/etc/rc.d/firewallip4
```

```
/etc/init.d/named start
```



## ANEXO 2

### PASOS PARA CONFIGURAR AUTOMATICAMENTE EL PROXY EN EL EXPLORADOR

Vamos a necesitar varias cosas:

- Proxy “propio”, un esnifer y un servidor web.
- Si usamos Linux, como es el caso, también SAMBA

Se van a plantear dos escenarios, uno en el que los clientes reciben una IP asignada por un servidor (DHCP) y otro en el que la IP de los clientes de red es estática.

Centrémonos en DHCP, cuando un cliente se “engancha” a una red y solicita una Ip, intercambian una serie de mensajes con el servidor... DHCP Discover – DHCP Offer – DHCP ACK son los habituales... pero también existen otros, como DHCP Inform que contiene alguna otra información relevante, en nuestro caso el famoso (a partir de hoy) WPAD (**Web Proxy Auto-Discovery Protocol**).

Ese servidor DHCP contiene en su configuración la información típica y necesaria para entregar a los clientes:

Dirección IP

Máscara de Subred

Puerta de Enlace

Servidores DNS

Si ese servidor DHCP se configura, además de con todo eso, para informar al/los clientes de que existe un “Proxy para su detección automática” también se lo dirá a los clientes que soliciten esa IP.

Realmente, lo que les entrega es una dirección URL que incluye la configuración del Proxy, el puerto de conexión y las reglas de quien puede usar el Proxy.

Por esto mismo necesitaremos un servidor WEB, puesto que lo primero que hará el navegador del cliente es “conectarse” a esa URL para cargar el archivo de configuración.

Hay que hacer notar que el servidor Web y el Proxy no tienen por qué ser la misma máquina, ni tan siquiera estar en la misma red. Usaremos un dhcpd en Linux y Apache para solventar este asunto.

Si queremos informar de un “posible” Proxy para ser “auto-descubierto” tenemos que añadir (al menos) dos líneas en el archivo de configuración dhcpd.conf, estas son:

Código:

```
option wpad-url code 252 = text; # en la configuración global
```

```
option wpad-url = "http://dirección_del_proxy/archivo.pac" # en la configuración  
de la/las subredes que asigna.
```

También hay que reseñar otra cosa, ese archivo lo llevaremos al directorio donde el webserver guarda las páginas web, vamos al document root, si es un apache pues por ejemplo en /usr/local/apache/htdocs

Y mas cosas, hay que “enseñar” a apache o al webserver que se use para que interprete correctamente las extensiones .pac y el autodiscovery. Eso lo conseguimos añadiendo una línea del tipo AddType y/o en la configuración del Virtual Host, este es un ejemplo:

En la sección Addtype de Apache en el archivo httpd.conf añadimos:

Código:

```
AddType application/x-ns-proxy-autoconfig .pac
```

Y en VirtualHost del mismo archivo httpd.conf, añadimos:

Código:

```
<VirtualHost *:80>
```

```
    ServerName 192.168.4.66
```

```
    ServerAlias 192.168.4.66
```

```
    AddType application/x-ns-proxy-autoconfig .pac
```

```
</VirtualHost>
```

Si lo dejamos “tal cual” no funcionará lo del Proxy, porque ese equipo “no sabe” que existe un Proxy automático de configuración... nadie se lo ha dicho, antes fue dhcp, pero al ponerle una ip “a mano”, nada de nada, se lo tendría que decir. un DNS.

En estos casos se debe utilizar el servicio DNS, pero ni podemos, ni esa Máquina lo hará puesto que los DNS son públicos...

En un entorno “normal” sería el mismo DNS quien ha de guardar la dirección del servidor web y la del Proxy... y aquí es donde caen muchos.... nadie se acuerda de esto ni de quitar lo del auto descubrimiento del Proxy en los navegadores....

Y entonces? Que? Ponemos un DNS?

NetBios es otro protocolo, propietario de los Windows, un engendro que idearon hace años Microsoft e IBM, el caso es que como el DNS no les resolverá el nombre o la IP del Proxy, lo buscarán por NetBIOS.