

CAPITULO IV

Desarrollo Técnico de la Investigación

4. DESARROLLO TÉCNICO DE LA INVESTIGACIÓN.

4.1. INTRODUCCIÓN.

La aplicación que realizaremos estará basado en un paquete de software que será adaptados a un sistema GPL (General Public License) conformado por un Sistema Operativo Linux y el Servidor Proxy Squid.

El uso de esta tecnología nos permitirá abaratar costos tanto en licencias de software como dispositivos de hardware, sin que esto signifique una menor calidad en el desempeño.

El sistema nos dará un acceso transparente a internet a través de cualquier programa y a cualquier servicio. Nos ayudara a tener una mayor velocidad en la navegación, las páginas web que hayan sido visitadas serán almacenadas en el servidor para que no haya que solicitarlas de internet.

También nos ayudara a controlar las páginas web prohibidas para los estudiantes o cualquier tipo de usuarios. La aplicación optimizara el servicio de internet a las computadoras que integran la intranet de la Biblioteca Virtual de la F.A.F.I. El administrador de este servidor podrá configurar y manipular el servidor localmente desde el servidor u otro ordenador que esta enlazado a la intranet.

Esta aplicación dará una mejor administración y control del servicio de internet. Entre los cuales tenemos un control de anchos de banda, identificación de usuarios mediante las ip's y Mac address (identificación física de las tarjetas de red), también se optimizara el tiempo de enlace a la web, esto se debe a la cache con la que cuenta el Servidor Squid, también podemos separar a los usuarios por grupos o individuales a los que se aplicara perfiles para su control, controlar el tráfico de información que circula en la red.

La aplicación dará como resultado la optimización del ancho de banda que dispone la Biblioteca Virtual de la Facultad de Administración, Finanzas e Informática con estándares de calidad QoS, para que la navegación en internet sea más segura y para los fines con que cuenta dicho departamento de una conexión a internet.

4.2. OBJETIVO DE LA PROPUESTA.

4.2.1. Objetivo General.

Realizar una aplicación que permita la configuración de un Servidor Proxy de manera fácil y rápida para controlar el acceso a internet y el ancho de banda en la Biblioteca Virtual de la Facultad de Administración, Finanzas e Informática.

4.2.2. Objetivos específicos.

- Gestionar el ancho de banda que dispone la Biblioteca Virtual de la Facultad de Administración, Finanzas e Informática.
- Controlar el tráfico de información que circula en la intranet.
- Restringir contenido web o HTTP es decir páginas prohibidas.
- Optimizar el tiempo de acceso de los recursos tecnológicos con la asignación de perfiles controlados.
- Abaratar costos tanto en licencias de Software como dispositivos de Hardware.
- Mejorar la administración y control del servicio de internet.
- Implementar la configuración del servidor proxy Squid y el control del ancho de banda para brindar QoS al uso del internet en la Biblioteca Virtual.
- Acceder de manera transparente a internet por medio de cualquier programa y a cualquier servicio.
- Probar el proyecto y verificar la aceptación del usuario en la Biblioteca Virtual de la Facultad de Administración, Finanzas e Informática.

4.3. METODOLOGÍA DE DESARROLLO UTILIZADA.

Una metodología es una guía que se sigue a fin de realizar las acciones propias de una investigación. En términos más sencillos se trata de la guía que nos va indicando qué hacer y cómo actuar cuando se quiere desarrollar algún tipo de investigación. Es posible definir una metodología como aquel enfoque que permite observar un problema de una forma total, sistemática y disciplinada.

La metodología de desarrollo que vamos a utilizar esta basada en el análisis de necesidades y se la va a desarrollar con técnicas de programación web orientadas a objetos en un entorno distribuido. El desarrollo del sistema se realizara bajo la metodología Rational Unified Process (RUP). Esta parte corresponde a los pasos que se emplearon para dar respuestas a los objetivos planteados. Se especifican las herramientas usadas en cada fase de la investigación, y se presentan los pasos que amerita el desarrollo con la metodología RUP, cuyo objetivo es asegurar la producción de software de alta calidad que satisfaga la necesidad del usuario final dentro de un tiempo y presupuesto previsible. El resultado, una aplicación que responde a las actuales necesidades de la administración de los recursos de red.

El proceso RUP se repite en una serie de ciclos. Cada ciclo concluye con una versión del producto (release) y cada ciclo está dividido en 4 fases:

- Concepción o inicio: El objetivo es determinar la visión del proyecto y definir lo que se desea realizar.
- Elaboración: Se determina la arquitectura óptima del proyecto.
- Construcción: Se obtiene la capacidad operacional inicial.
- Transición: Obtener el producto acabado y definido.

Cada una de las fases está a su vez dividida en iteraciones, y en cada iteración se realizan 5 procesos o etapas principales: requerimientos, análisis, diseño, implementación, pruebas y entrega.

4.4. ANÁLISIS PREVIO.

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

Nuestra investigación se enfoca en examinar y determinar qué políticas de administración de red son usadas en la Biblioteca Virtual de la F.A.F.I. El análisis a efectuarse permite conocer que recursos y/o servicios, herramientas, etc. son utilizadas para la adecuada administración del tráfico de red.

Para la recopilación de dicha información, se efectuó una entrevista a uno de los administradores, el cual nos dio a conocer los recursos existentes y que explicaremos a continuación.

4.4.1. Infraestructura Tecnológica.

Para tener un mejor conocimiento de la infraestructura vigente en la Biblioteca Virtual de la Facultad de Administración, Finanzas e Informática, analizaremos los componentes que la conforman.

Servidores

En la actualidad la Biblioteca de la Facultad de Administración, Finanzas e Informática, no consta con ningún equipo Servidor para el control del acceso a internet.

Entorno de Red

Actualmente los equipos de red que existen en la Biblioteca de la Facultad de Administración, Finanzas e Informática están orientados hacia los protocolos IPv4, por ejemplo el switch principal es un D-Link DES-1024S y es el encargado de administrar las peticiones de los usuarios de la Biblioteca Virtual de la Facultad de

Administración, Finanzas e Informática, hacia el servidor principal ubicado en el Departamento de Sistemas ya que ellos son los encargados de distribuir los recursos hacia cada dependencia existente en la Universidad Técnica de Babahoyo.



Grafico N° 4.1: Infraestructura de red de la Biblioteca de la Facultad de Administración, Finanzas e Informática.

Administración de Red

Los contenidos del tráfico de red de todos los laboratorios y de la Biblioteca Virtual son filtrados mediante la utilización del servidor Proxy principal de la U.T.B., es importante mencionar que no está implantado un filtrador de paquetes que permita administrar de una manera adecuada el tráfico generado en la red, todo lo referente a políticas para filtrado de paquetes lo realiza directamente el firewall de la U.T.B. ubicado en el Departamento de Sistemas.

Infraestructura Física

La Biblioteca Virtual de la F.A.F.I. cuenta con una infraestructura basada en el protocolo IPv4, es decir cableado estructurado categoría 5. Además la Biblioteca Virtual de la F.A.F.I. posee un switch local que se conecta con el switch principal ubicado en el Departamento de Sistemas de la U.T.B. para de ésta manera proveer los servicios a las peticiones efectuadas por los usuarios de la red.

4.4.2. Resultados Obtenidos.

Una vez recopilada la información necesaria tras el análisis a la Biblioteca Virtual F.A.F.I., se obtuvo los siguientes resultados:

- En cuanto al filtrado de contenidos al ser el servidor principal de la U.T.B. el encargado de esta tarea todos los cambios y limitaciones que se apliquen son reflejadas en todos los departamentos de la institución, generando molestias y quejas por parte del personal administrativo y docentes que utilizan cotidianamente el internet, por este motivo con la implementación interna de un servidor proxy en la Biblioteca Virtual F.A.F.I. solo se controla la navegación y el tráfico generado por los estudiantes que utilizan el internet de la Biblioteca para realizar sus consultas y tareas.
- En lo referente a mecanismos que permitan el filtrado de paquetes de red, no existe una herramienta que cumpla con esta finalidad; es decir, un Firewall. Por lo que sería conveniente la utilización de un Firewall que cumpla con este propósito.
- Con la información recopilada tras examinar las condiciones actuales en la Biblioteca Virtual F.A.F.I., podemos delinear los módulos y los parámetros de configuración para administrar el tráfico de red usando Proxy y Firewall, administrables mediante un entorno Web.

4.4.3. Lista de requerimientos y funcionalidades.

- Disponibilidad del internet para una navegación fluida.
- No existan retrasos en la comunicación, ni lentitud al navegar por el internet.
- Velar por el buen uso del ancho de banda, y del tránsito que circula en la red.

- Proporcionar mecanismos seguros que restrinjan y limiten el uso de Internet a solo material pertinente, que vaya orientado al cumplimiento de los objetivos de la institución.
- Administrar los recursos de la red de forma organizada y centralizada.
- Conexiones de red para cada estación que se desee conectar.
- Concentrador Fast Ethernet con los puertos suficientes para las estaciones de trabajo y servidor que se conecten al sistema.
- La biblioteca de la Facultad de Administración, Finanzas e Informática dispone de 19 puntos de red para la conexión de las computadoras.
- Definir reglas para restringir el contenido disponible para los usuarios de la red interna.

4.5. DISEÑO.

4.5.1. Diseño de red.

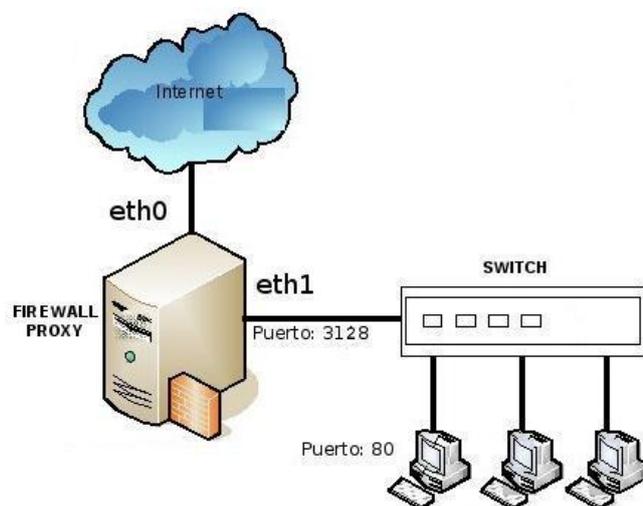


Grafico N° 4.2: Infraestructura de una red con Servidor Proxy.

Entenderemos fácilmente como funciona un proxy si tenemos claro cómo funciona Internet. Internet está basado en el modelo Cliente-Servidor. A grandes rasgos, un usuario (cliente) hace una solicitud (petición de un archivo) al Servidor, que devolverá una respuesta (el archivo). Para ello, el servidor web requiere una

información adicional. Esta información es transmitida al servidor por un navegador o un servidor proxy.

Generalmente, lo que solicita el servidor es:

- Nombre y versión del SO.
- Nombre y versión del navegador.
- Configuración del navegador (resolución de pantalla, profundidad de color, si existe soporte para java / Javascript).
- Dirección IP del cliente.
- Otra información.

Así, el término proxy hace referencia a cualquier dispositivo o aplicación que hace de intermediario entre un ordenador conectado a Internet y el servidor al que se conecta. Su uso más común es permitir la conexión a Internet de varios equipos conectados en red a través de uno que es el que está conectado realmente a Internet.

4.5.2. Diseño de Base de datos.

La aplicación propuesta utiliza el sistema de gestión de base de datos SQLite, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo.

El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host, en este caso la aplicación utiliza dos ficheros para almacenar los datos, “proxynet.db” es el script de base de datos donde se almacena información general referente al servidor y “squid.db” es el script de base de datos donde se almacena la información de la configuración del servidor proxy.

4.5.2.1. Modelo conceptual.

Los modelos de datos conceptuales están orientados a la descripción de estructuras de datos y restricciones de integridad.

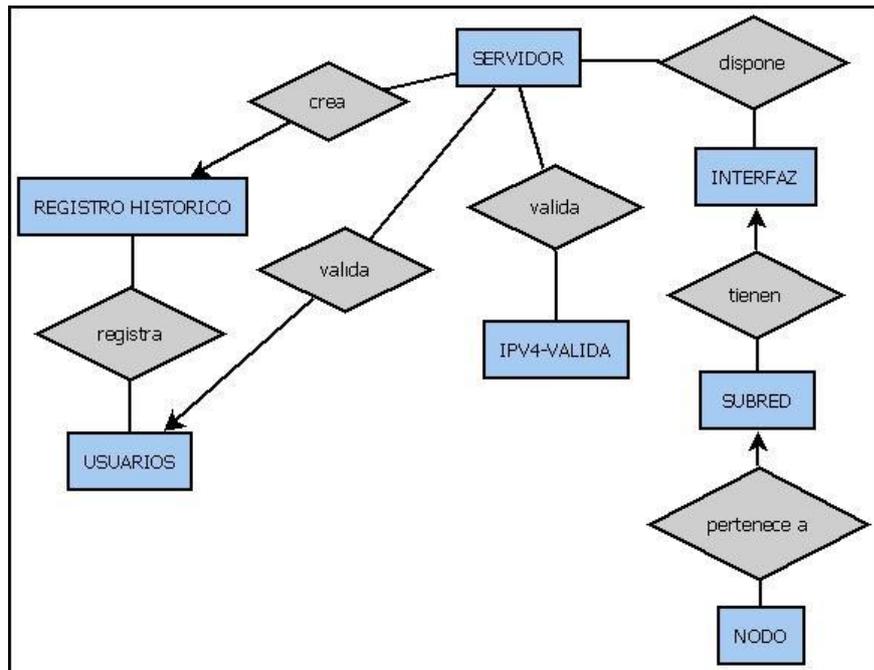


Grafico N° 4.3: Modelo conceptual de la base de datos “proxynet.db”.

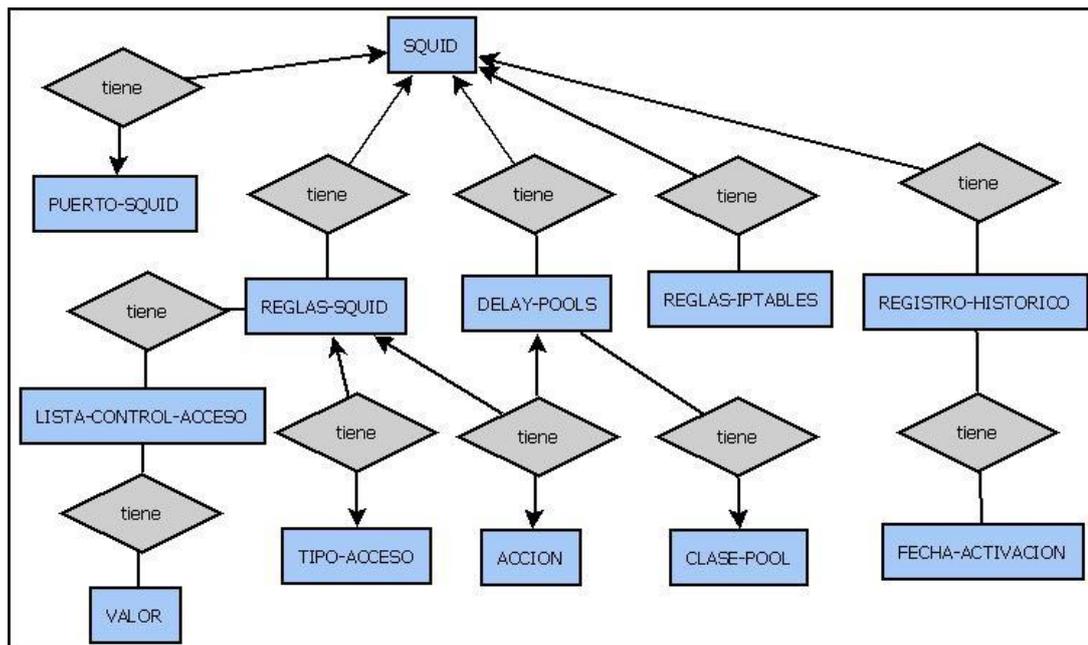


Grafico N° 4.4: Modelo conceptual de la base de datos “squid.db”.

4.5.2.2. Modelo Físico – Modelo Entidad Relación.

Los modelos de datos físicos son estructuras de datos a bajo nivel implementadas dentro del propio manejador.

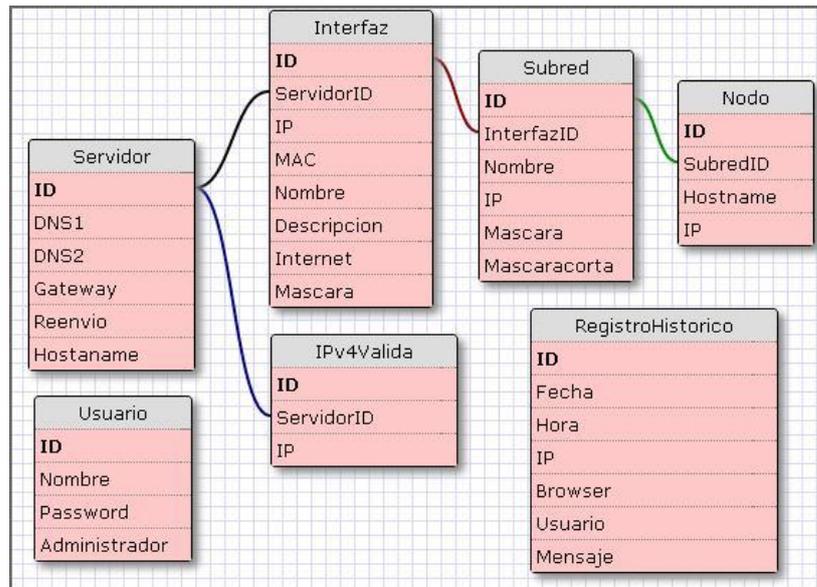


Grafico N° 4.5: Modelo físico de la base de datos “proxynet.db”.

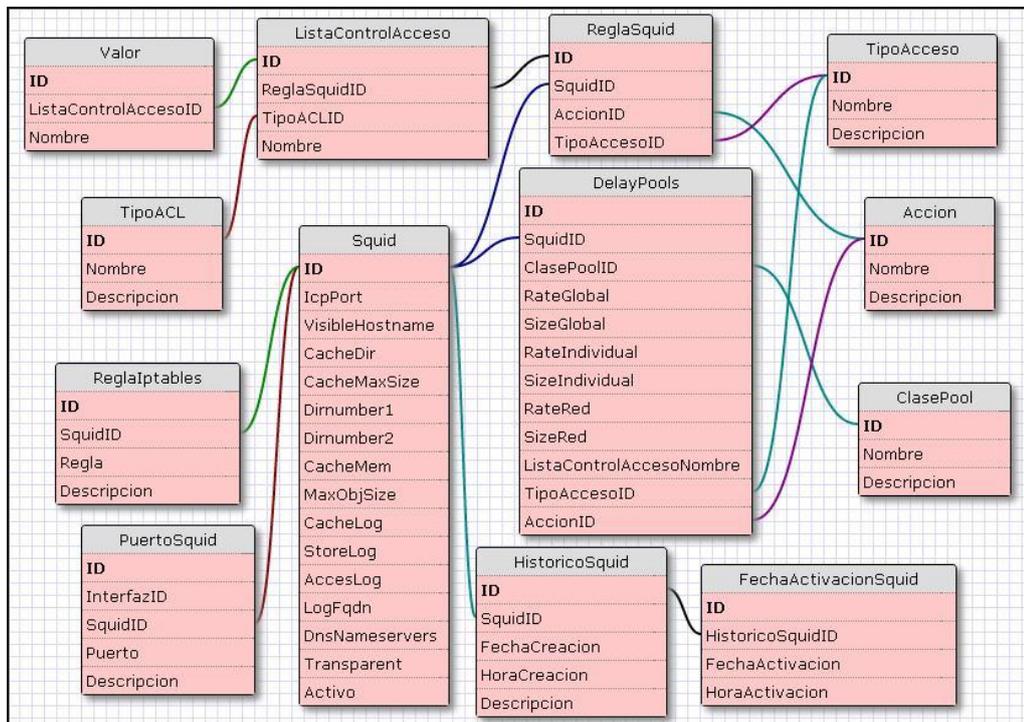


Grafico N° 4.6: Modelo físico de la base de datos “squid.db”.

4.5.2.3. Diccionario de datos.

Diccionarios de datos de la base de datos “proxynet.db”.

Entidad SERVIDOR		
Información relevante del servidor		
Atributo	Descripción del atributo	
ID	Código del servidor	
DNS1	Dirección ip del servidor DNS registrado	
DNS2	Dirección ip del servidor DNS registrado	
Gateway	Puerta de enlace del ISP	
Reenvio	Bit de reenvío del servidor	
Hostname	Nombre del host	
Relaciones		Campo clave
Servidor con Interfaz		ID
Servidor con IPv4Valida		ID

Tabla N° 4.1: Diccionario de datos de la entidad Servidor de la BD “proxynet”.

Entidad INTERFAZ		
Información de las interfaces existentes		
Atributo	Descripción del atributo	
ID	Código de la interfaz	
ServidorID	Código del servidor	
IP	Dirección IP de la interfaz	
MAC	Dirección MAC de la interfaz	
Nombre	Nombre de la interfaz	
Descripcion	Descripción de la interfaz	
Internet	Bit que indica si la interfaz está conectada a internet	
Mascara	Mascara de red de la interfaz	
Relaciones		Campo clave
Interfaz con Servidor		ServidorID
Interfaz con Subred		ID

Tabla N° 4.2: Diccionario de datos de la entidad Interfaz de la BD “proxynet”.

Entidad IPV4VALIDA		
Direcciones ip validas		
Atributo	Descripción del atributo	
ID	Código de la dirección ip	
ServidorID	Código del servidor	
IP	Dirección ip valida	
Relaciones		Campo clave
IPv4Valida con Servidor		ServidorID

Tabla N° 4.3: Diccionario de datos de la entidad IPv4Valida de la BD “proxynet”.

Entidad SUBRED		
Información de las subredes registradas		
Atributo	Descripción del atributo	
ID	Código de la subred	
InterfazID	Código de la interfaz	
Nombre	Nombre de la subred	
IP	Dirección IP de la subred	
Mascara	Mascara de red	
MascaraCorta	Mascara corta según clase de red	
Relaciones		Campo clave
Subred con Interfaz		InterfazID
Subred con Nodo		ID

Tabla N° 4.4: Diccionario de datos de la entidad Subred de la BD “proxynet”.

Entidad NODO		
Información de los nodos de red		
Atributo	Descripción del atributo	
ID	Código del nodo	
SubredID	Código de la subred	
Hostname	Nombre del nodo	
IP	Dirección ip del nodo	
Relaciones		Campo clave
Nodo con Subred		ID

Tabla N° 4.5: Diccionario de datos de la entidad Nodo de la BD “proxynet”.

Entidad USUARIO	
Información de los usuarios	
Atributo	Descripción del atributo
ID	Código del usuario
Nombre	Nombre del usuario
Password	Contraseña del usuario
Administrador	Bit que indica si el usuario es administrador o normal

Tabla N° 4.6: Diccionario de datos de la entidad Usuario de la BD “proxynet”.

Entidad REGISTROHISTORICO	
Información de registro de inicio de sesión de los usuarios	
Atributo	Descripción del atributo
ID	Código del registro histórico
Fecha	Fecha en la que se produce el registro
Hora	Hora en la que se produce el registro
IP	Dirección ip desde la que se accede
Browser	Información del navegador
Usuario	Nombre del usuario
Mensaje	Mensaje generado por la aplicación

Tabla N° 4.7: Diccionario de datos de la entidad RegistroHistorico de la BD “proxynet”.

Diccionario de dato de la base de datos “squid.db”.

Entidad SQUID	
Información de las configuración Squid generadas por la aplicación	
Atributo	Descripción del atributo
ID	Código de la configuración squid
IcpPort	Puerto ICP
VisibleHostname	Nombre del Servidor
CacheDir	Directorio de la cache
CacheMaxSize	Tamaño máximo de la cache
DirNumber1	Directorio de nivel 1
DirNumber2	Directorio de nivel 2
CacheMem	Tamaño de la memoria cache
MaxObjSize	Tamaño máximo de objetos almacenables
CacheLog	Directorio del archivo cache.log
StoreLog	Directorio del archivo store.log
AccesLog	Directorio del archivo access.log
LogFqdn	Almacenamiento por dominio o ip
DnsNameServer	Nombre del servidor DNS
Transparent	Configuración Squid convencional o transparente
Activo	Si esta la configuración activa
ValorInicialCubo	Especifica el valor inicial del cubo del ancho de banda
Relaciones	Campo clave
Squid con ReglaSquid	ID
Squid con ReglaIptables	ID
Squid con DelayPools	ID
Squid con PuertoSquid	ID
Squid con HistoricoSquid	ID

Tabla N° 4.8: Diccionario de datos de la entidad Squid de la BD “squid”.

Entidad REGLASQUID	
Información de las reglas de acceso	
Atributo	Descripción del atributo
ID	Código de la regla Squid
SquidID	Código de la configuración Squid
AccionID	Código de la acción
TipoAccesoID	Código del tipo de acceso
Relaciones	Campo clave
RglaSquid con ListaControlAcceso	ID
ReglaSquid con Squid	SquidID
ReglaSquid con Accion	AccionID
ReglaSquid con TipoAcceso	TipoAccesoID

Tabla N° 4.9: Diccionario de datos de la entidad ReglaSquid de la BD “squid”.

Entidad LISTACONTROLACCESO	
Información de las listas de control de acceso	
Atributo	Descripción del atributo
ID	Código de la lista de control de acceso
ReglaSquidID	Código de la regla Squid
TipoACLID	Tipo del tipo de acl
Nombre	Nombre de la lista de control de acceso
Relaciones	Campo clave
ListaControlAcceso con Valor	ID
ListaControlAcceso con ReglaSquid	ReglaSquidID
ListaControlAcceso con TipoACL	TipoACLID

Tabla N° 4.10: Diccionario de datos de la entidad ListaControlAcceso de la BD “squid”.

Entidad VALOR	
Información de los valores asociados a las listas de control de acceso	
Atributo	Descripción del atributo
ID	Código del valor
ListaControlAccesoID	Código de la lista de control de acceso
Nombre	Nombre del valor
Relaciones	Campo clave
Valor con ListaControlAcceso	ListaControlAccesoID

Tabla N° 4.11: Diccionario de datos de la entidad Valor de la BD “squid”.

Entidad TIPOACL	
Información de los tipos de lista de control de acceso	
Atributo	Descripción del atributo
ID	Código del tipo de acl
Nombre	Nombre del tipo de acl
Descripcion	Descripción del tipo de acl
Relaciones	Campo clave
TipoACL con ListaControlAcceso	ID

Tabla N° 4.12: Diccionario de datos de la entidad TipoACL de la BD “squid”.

Entidad ACCION	
Información de las acciones que toman las reglas	
Atributo	Descripción del atributo
ID	Código de la acción
Nombre	Nombre de la acción
Descripcion	Descripción de la acción
Relaciones	Campo clave
Accion con ReglaSquid	ID
Accion con DelayPools	ID

Tabla N° 4.13: Diccionario de datos de la entidad Accion de la BD “squid”.

Entidad TIPOACCESO		
Información de los tipos de acceso		
Atributo	Descripción del atributo	
ID	Código del tipo de acceso	
Nombre	Nombre del tipo de acceso	
Descripcion	Descripción del tipo de acceso	
Relaciones		Campo clave
TipoAcceso con ReglaSquid		ID

Tabla N° 4.14: Diccionario de datos de la entidad TipoAcceso de la BD “squid”.

Entidad REGLAIPABLES		
Información de las regla Iptables registradas en la configuración		
Atributo	Descripción del atributo	
ID	Código de la regla	
SquidID	Código Squid de las reglas	
Regla	Regla Iptables	
Descripcion	Descripción de la regla Iptables	
Relaciones		Campo clave
ReglaIptables con Squid		SquidID

Tabla N° 4.15: Diccionario de datos de la entidad ReglaIptables de la BD “squid”.

Entidad DELAYPOOLS		
Información de los Delay pools (control del ancho de banda)		
Atributo	Descripción del atributo	
ID	Código de la Delay pool	
SquidID	Código de la configuración Squid	
ClasePoolID	Código de la clase de pool	
RateGlobal	Cantidad de bits por segundo	
SizeGlobal	Cantidad máxima de bits	
RateIndividual	Cantidad de bits por segundo	
SizeIndividual	Cantidad máxima de bits	
RateRed	Cantidad de bits por segundo	
SizeRed	Cantidad máxima de bits	
ListaControlAccesoNombre	Nombre de lista de control de acceso	
TipoAccesoID	Código del tipo de acceso	
AccionID	Código de la acción	
Relaciones		Campo clave
DelayPools con Squid		SquidID
DelayPools con ClasePool		ClasePoolID
DelayPools con TipoAcceso		TipoAccesoID
DelayPools con AccionID		AccionID

Tabla N° 4.16: Diccionario de datos de la entidad DelayPools de la BD “squid”.

Entidad CLASEPOOL	
Información de las clases de pools existentes	
Atributo	Descripción del atributo
ID	Código de la clase de pool
Nombre	Nombre de la clase pool
Descripcion	Descripción de la clase de pool
Relaciones	
ClasePool con DelayPools	Campo clave ID

Tabla N° 4.17: Diccionario de datos de la entidad ClasePool de la BD "squid".

Entidad PUERTOSQUID	
Información de puertos por donde escuchara el servidor	
Atributo	Descripción del atributo
ID	Código del puerto
InterfazID	Código de la interfaz del puerto registrado
SquidID	Código de la configuración Squid
Puerto	Puerto del Squid
Descripcion	Descripción del puerto
Relaciones	
Puertosquid con Squid	Campo clave SquidID

Tabla N° 4.18: Diccionario de datos de la entidad PuertosSquid de la BD "squid".

Entidad HISTORICOSQUID	
Información de las bitácoras de activación de la configuración	
Atributo	Descripción del atributo
ID	Código del registro histórico
SquidID	Código de la configuración Squid
FechaCreacion	Fecha de creación de la configuración
HoraCreacion	Hora de creación de la configuración
Descripcion	Descripción del registro histórico
Relaciones	
HistoricoSquid con Squid	Campo clave SquidID
HistoricoSquid con FechaActivacionSquid	ID

Tabla N° 4.19: Diccionario de datos de la entidad HistoricoSquid de la BD "squid".

Entidad FECHAActivACIONESQUID	
Información de las fechas en que aplico la configuración	
Atributo	Descripción del atributo
ID	Código del puerto
HistoricoSquidID	Código del registro histórico
FechaActivacion	Fecha en la que se aplico la configuración
HoraActivacion	Hora en la que se aplico la configuración
Relaciones	
FechaActivacionSquid con HistoricoSquid	Campo clave HistoricoSquidID

Tabla N° 4.20: Diccionario de datos de la entidad FechaActivacionde la BD "squid".

4.5.2.4. Script de base de datos.

Script de la base de datos “proxynet”

```
Creación de la tabla USUARIO  
CREATE TABLE Usuario (  
  ID integer primary key,  
  Nombre varchar(20),  
  Password varchar(255),  
  Administrador boolean  
);  
  
Creación de la tabla SERVIDOR  
CREATE TABLE Servidor (  
  ID integer primary key,  
  DNS1 varchar(15),  
  DNS2 varchar(15),  
  Gateway varchar(15),  
  Reenvio boolean,  
  Hostname varchar(50)  
);  
  
Creación de la Tabla IPV4VALIDA  
CREATE TABLE IPv4Valida (  
  ID integer primary key,  
  ServidorID integer,  
  IP varchar(15)  
);
```

Script de la base de datos “squid”

```
Creación de la tabla SQUID  
CREATE TABLE Squid (  
  ID integer PRIMARY KEY,  
  IcpPort integer,  
  VisibleHostname varchar(100),  
  CacheDir varchar(100),  
  CacheMaxSize integer,  
  DirNumber1 integer,  
  DirNumber2 integer,  
  CacheMem integer,  
  MaxObjSize integer,  
  CacheLog varchar(100),  
  StoreLog varchar(100),  
  AccessLog varchar(100),  
  LogFqdn boolean,  
  DnsNameservers varchar(100),  
  Transparent boolean,  
  Activo boolean,  
  ValorInicialCubo varchar(3)  
);
```

```

Creación de la tabla REGLASQUID
CREATE TABLE ReglaSquid (
  ID integer PRIMARY KEY,
  SquidID integer,
  AccionID integer,
  TipoAccesoID integer
);

Creación de la tabla REGLAIPTABLES
CREATE TABLE ReglaIptables (
  ID integer PRIMARY KEY,
  SquidID integer,
  Regla varchar(150),
  Descripcion varchar(150)
);

Creación de la tabla DELAYPOOLS
CREATE TABLE DelayPools (
  ID integer PRIMARY KEY,
  SquidID integer,
  ClasePoolID integer,
  RateGlobal varchar(100),
  SizeGlobal varchar(100),
  RateIndividual varchar(100),
  SizeIndividual varchar(100),
  RateRed varchar(100),
  SizeRed varchar(100),
  ListaControlAccesoNombre varchar(20),
  TipoAccesoID integer,
  AccionID integer
);

```

4.5.3. Casos de uso.

Los Casos de Uso no son parte del diseño, sino parte del análisis. De forma que al ser parte del análisis nos ayuda a describir qué es lo que es sistema debe hacer. Los Casos de Uso son qué hace el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario.

Caso de uso 1: Validación de datos con el sistema.

Caso de Uso:	Validación de datos con el sistema.
Actores:	Usuario, Sistema
Tipo:	Primario
Propósito:	“Proxynet” permite un ingreso seguro, mediante la validación de la ip autorizada y los datos registrados en el sistema, con los que se registra el usuario, utilizando sesiones.

Tabla N° 4.21: Caso de uso 1 validación de datos con el sistema.

Curso típico de eventos

Actor	Sistema
1. El usuario ingresa la dirección IP del servidor.	2. Procede a verificar la dirección IP que solicita la página. 2.1. Si la dirección es válida le muestra la página de Inicio de sesión. 2.2. Si la IP no es válida se re direcciona a una página de equipo no autorizado. 3. Se crea un registro histórico
4. El usuario ingresa sus datos (usuario y contraseña).	5. Verifica los datos ingresados. 5.1. Si los datos son correctos se le re direcciona a la página principal de la aplicación. 5.2. Si los datos son incorrectos, se le re direcciona a una página de usuario no autorizado. 6. Se crea un registro histórico

Tabla N° 4.22: Curso típico de eventos del caso de uso 1.

Caso de Uso 2: Generar Configuración Squid.

Caso de Uso:	Generar Configuración Squid.
Actores:	Administrador, Sistema
Tipo:	Primario
Propósito:	“Proxynet” permite generar configuraciones que son almacenadas en la base de datos con los parámetros básicos para el funcionamiento del proxy en modo convencional o transparente.

Tabla N° 4.23: Caso de uso 2 Generar configuración Squid.

Curso típico de eventos

Actor	Sistema
1. El administrador inicia el asistente para crear una nueva configuración Squid.	2. Muestra la ventana de inicio del asistente de configuración, donde se solicita los siguientes datos: Puertos del Squid Valores iniciales Directorios de log Subred principal - Tipo proxy
	2.1. En la pagina Puerto del Squid debe agregar los puertos por donde escuchara el servidor proxy, por defecto es 3128
	2.2. En la pagina Valores iniciales debe modificar si lo desea, los valores por defecto presentados.

	2.3. En la página Directorios de log debe modificar si lo desea las rutas a los directorios de log del Squid, además especificar si se almacena por ip o dominio.
	2.4. En la página Subred principal – tipo proxy debe especificar primero la subred principal para la configuración del firewall, y especificar si la configuración del Squid es convencional o transparente.
3. El administrador avanza entre las páginas del asistente.	4. Captura los datos ingresados y los mantiene serializados.
5. El administrador finaliza el asistente.	6. Almacena los datos en la base de datos, y presenta la página de registro de reglas de control.

Tabla N° 4.24: Curso típico de eventos del caso de uso 2.

Caso de uso 3: Crear Reglas de Control de Acceso.

Caso de Uso:	Crear Reglas de Control de Acceso.
Actores:	Usuario, Sistema
Tipo:	Primario
Propósito:	“Proxynet” permite a los usuarios del sistema crear reglas de control de acceso y reglas de control de la cache, con sus respectivas listas de control de acceso.

Tabla N° 4.25: Caso de uso 3 Crear reglas de control de acceso.

Curso típico de eventos

Actor	Sistema
1. El usuario selecciona del menú Squid (Proxy) la opción reglas y control de acceso.	2. Presenta la página de registro de nuevas reglas de control de acceso. 2.2. Despliega las reglas creadas. 2.3. Muestra las reglas aprobadas.
3. El usuario crea nueva regla de control de acceso. 3.1. Establece el nombre de la regla. 3.2. Selecciona el tipo de ACL. 3.3. Especifica los valores. 3.4. Presiona en Agregar.	4. Guarda la regla de control de acceso en la base de datos y la agrega a las reglas sin acción definida.
5. El usuario especifica la acción de la regla (Aprobar o denegar).	6. Guarda la acción escogida y traslada la regla a la sección de reglas aprobadas.

Tabla N° 4.26: Curso típico de eventos del caso de uso 3.

Caso de Uso 4: Crear Pools de Retardo.

Caso de Uso:	Crear Pools de Retardo.
Actores:	Usuario, Sistema
Tipo:	Primario
Propósito:	“Proxynet” permite a los usuarios del sistema crear 3 clases de Delay pools para el control del ancho de banda, también especificar a qué regla de control aplicar.

Tabla N° 4.27: Caso de uso 4 Crear Pools de Retardo.

Curso típico de eventos

Actor	Sistema
1. El usuario selecciona del menú Squid (Proxy) la opción reglas y control de acceso, y luego el apartado DelayPools.	2. Presenta la página de registro de nuevas Delay pools. 2.2. Despliega las reglas creadas y asignada la ACL. 2.3. Muestra las reglas creadas y el listado de reglas a aprobar.
3. El usuario crea nueva pool de retardo, especifica los parámetros. 3.1. Selecciona la Clase de Pool a crear. 3.2. Dependiendo la clase de pool se establecen los parámetros. 3.3. Presiona en Agregar.	4. Guarda la Delay pool en la base de datos.
	5. Muestra en la Delay Pool nueva una lista de ACL a cual aplicar.
6. Selecciona de la lista a cual ACL aplicar la Delay Pool y especifica la acción de la regla (Aprobar o denegar).	7. Actualiza la información de la Delay Pools.

Tabla N° 4.28: Curso típico de eventos del caso de uso 4.

Caso de uso 5: Aplicar Configuración.

Caso de Uso:	Aplicar Configuración.
Actores:	Usuario, Servidor, Sistema
Tipo:	Primario
Propósito:	“Proxynet” permite a los usuarios del sistema aplicar las configuraciones creadas a los archivos de configuración de Squid e Iptables generando un registro histórico de configuraciones aplicadas.

Tabla N° 4.29: Caso de uso 5 Aplicar Configuración.

Curso típico de eventos

Actor	Sistema
1. El usuario accede a la opción Registro Histórico del menú Squid (Proxy).	2. Despliega las configuraciones generadas, muestra la configuración que está actualmente activa identificada con el icono de círculo verde.
3. Escoge la opción de aplicar la configuración, ya sea la activa o una no activa para activarla.	4. Genera los archivos de configuración. 4.1. Escribe toda la configuración, las reglas proxy y las Delay pools en el archivo "squid.conf". 4.2. Escribe todas las reglas de filtrado en Iptables.
	5. Reinicia los servicios Squid e Iptables.
	6. Activa la configuración cargando nuevamente los valores.
	7. Crea un registro de histórico

Tabla N° 4.30: Curso típico de eventos del caso de uso 5.

Caso de uso 6: Información del servidor.

Caso de Uso:	Información del servidor.
Actores:	Usuario, Sistema
Tipo:	Primario
Propósito:	Obtener información específica referente al equipo donde se ejecuta "Proxynet".

Tabla N° 4.31: Caso de uso 6 Información del servidor.

Curso típico de eventos

Actor	Sistema
1. El administrador selecciona la opción Datos del servidor de la página principal.	2. Despliega la página con la información del servidor referente a las interfaces de red, las direcciones ip, direcciones mac y direcciones ip validas para acceder al sistema.
	3. Pide al usuario realizar una acción.
4. Al administrador puede agregar una dirección ip valida la cual tendrá acceso a la aplicación.	5. Guarda la dirección ip en la base de datos.
6. El administrador puede recargar los valores presionando Recargar si existiera algún cambio en el servidor.	7. Ejecuta comandos de detección de información del servidor, actualiza la información actual y la almacena en la base de datos.

Tabla N° 4.32: Curso típico de eventos del caso de uso 6.

Caso de uso 7: Módulos registrados.

Caso de Uso:	Módulos registrados.
Actores:	Usuario, Sistema
Tipo:	Primario
Propósito:	Obtener información del estado de los módulos Squid e Iptables registrados en la aplicación “Proxynet”.

Tabla N° 4.33: Caso de uso 7 Módulos registrados.

Curso típico de eventos

Actor	Sistema
1. El usuario selecciona Módulos Registrados de la página principal de la aplicación.	2. Muestra la pagina de módulos registrados.
	2.1. Describe la forma de agregar nuevos módulos a la aplicación.

Tabla N° 4.34: Curso típico de eventos del caso de uso 7.

Caso de uso 8: Inicio y detención de servicios.

Caso de Uso:	Inicio y detención de servicios.
Actores:	Usuario, Sistema
Tipo:	Primario
Propósito:	“Proxynet” provee la posibilidad de iniciar y/o detener servicios, dependiendo del tipo de configuración que el administrador efectué sobre el sistema.

Tabla N° 4.35: Caso de uso 8 Inicio y detención de servicios.

Curso típico de eventos

Actor	Sistema
1. Si el usuario una vez que ha configurado un archivo del sistema, puede establecer la manera en que el servicio opere.	2. Si la configuración de un fichero depende o no de la inicialización de servicios, el sistema será capaz de realizar estas tareas de forma automática.
	3. Visualizará los resultados obtenidos al usuario para que compruebe su correcto desempeño.

Tabla N° 4.36: Curso típico de eventos del caso de uso 8.

Caso de uso 9: Administración de usuarios.

Caso de Uso:	Administración de usuarios.
Actores:	Administrador, Sistema
Tipo:	Primario
Propósito:	El administrador es el único autorizado a crear nuevos usuarios y gestionar su propia cuenta y la de los demás usuarios.

Tabla N° 4.37: Caso de uso 9 Administración de usuarios.

Curso típico de eventos

Actor	Sistema
1. El administrador selecciona la opción Administrar usuarios del la pagina principal.	2. Muestra la lista de los usuarios registrados en el sistema con las opciones de agregar nuevo usuario, modificar usuario y eliminar usuario.
	3. Solicita al administrador elegir el tipo de acción a tomar.
4. El administrador escoge la opción que necesite.	5. Muestra los resultados tras la acción seleccionada.

Tabla N° 4.38: Curso típico de eventos del caso de uso 9.

Caso de uso 10: Administrar Subredes.

Caso de Uso:	Administración de usuarios.
Actores:	Usuario, Sistema
Tipo:	Primario
Propósito:	“Proxynet” puede gestionar las interfaces de red existentes en el servidor, así como tener referencia de las subredes existente.

Tabla N° 4.39: Caso de uso 10 Administrar Subred.

Curso típico de eventos

Actor	Sistema
1. El usuario selecciona Administrar Subred de la página principal.	2. Muestra la pagina de administrar subredes, poniendo como encabezado las interfaces existentes en el equipo servidor.
	3. Solicita al usuario el tipo de acción a realizar. Listar Nodos Nueva Subred Editar interfaz
	3.1. En la pagina Listar nodos muestra

	las subredes creadas con sus respectivos nodos.
	3.2. En la página Nueva subred muestra la interfaz para poder registrar una nueva subred y poder ingresar sus respectivos nodos.
	3.3. En la pagina Editar interfaz puede modificar las descripciones de las interfaces mas no las direcciones ip de dichas interfaces.
4. El usuario escoge la opción que necesite.	5. Realiza las acciones tras la acción seleccionada y muestra los resultados.

Tabla N° 4.40: Curso típico de eventos del caso de uso 10.

Caso de Uso 11: Verificación de Log.

Caso de Uso:	Administración de usuarios.
Actores:	Usuario, Sistema
Tipo:	Primario
Propósito:	“Proxynet” tiene acceso a los archivos de log generados por el servidor proxy, para su consulta clasificados en tres tipos: cache, store y access.

Tabla N° 4.41: Caso de uso 11 Verificación de Log.

Curso típico de eventos

Actor	Sistema
1. El usuario selecciona el tipo de Reporte que desea visualizar del menú Reportes Squid .	2. Muestra el reporte log dependiendo de la opción seleccionada por el usuario. Log de la cache Log de Acceso Log store
	2.1. En la pagina Log de la cache puede visualizar el contenido del archivo cahe.log del servidor squid, detallada en una tabla.
	2.2. En la pagina Log de acceso puede visualizar el contenido del archivo access.log del servidor squid, detallada en una tabla.
	2.3. En la pagina Log store puede visualizar el contenido del archivo store.log del servidor squid, detallada en una tabla.

Tabla N° 4.42: Curso típico de eventos del caso de uso 11.

Caso de uso 12: Monitoreo de usuarios.

Caso de Uso:	Administración de usuarios.
Actores:	Usuario, Sistema
Tipo:	Primario
Propósito:	“Proxynet” incorpora una sección de monitoreo de usuarios activos en la red, para tener un seguimiento de las páginas visitadas.

Tabla N° 4.43: Caso de uso 12 Monitoreo de usuarios.

Curso típico de eventos

Actor	Sistema
1. El usuario selecciona la opción Monitoreo de usuarios del menú Reportes o de la página principal.	2. Muestra la pagina de monitoreo de usuarios con la información de los usuarios actualmente activos en la red.
	3. Solicita al usuario realizar una acción. Actualizar Parar
4. El usuario especifica el tiempo en segundos y presiona en actualizar.	5. Actualiza la pagina de monitoreo dependiendo del tiempo dado por el usuario.
6. El usuario presiona en parar para detener la actualización.	7. Detiene la actualización de la página.

Tabla N° 4.44: Curso típico de eventos del caso de uso 12.

4.5.4. Diagramas de casos de usos.

Caso de uso 1: Validación de datos con el sistema.

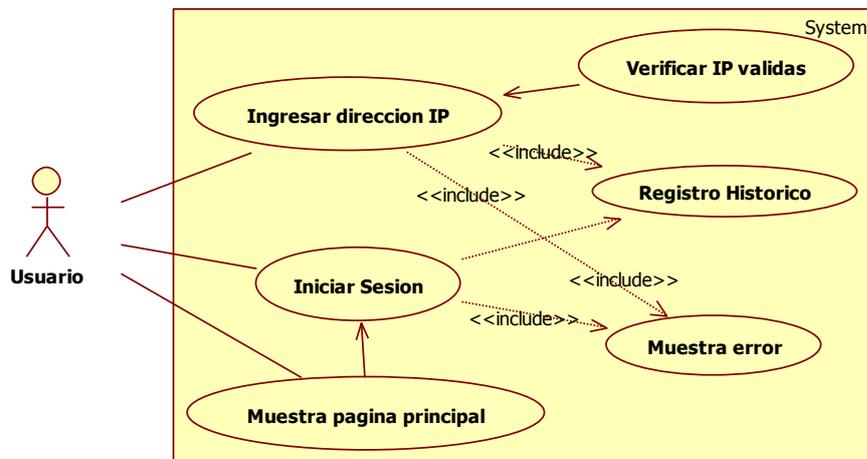


Grafico N° 4.7: Diagrama de caso de uso 1.

Caso de uso 2: Generar Configuración Squid.

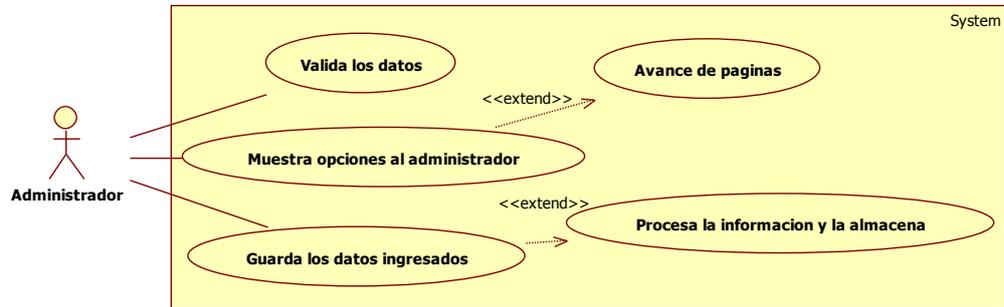


Grafico N° 4.8: Diagrama de caso de uso 2.

Caso de uso 3: Crear Reglas de Control de Acceso.

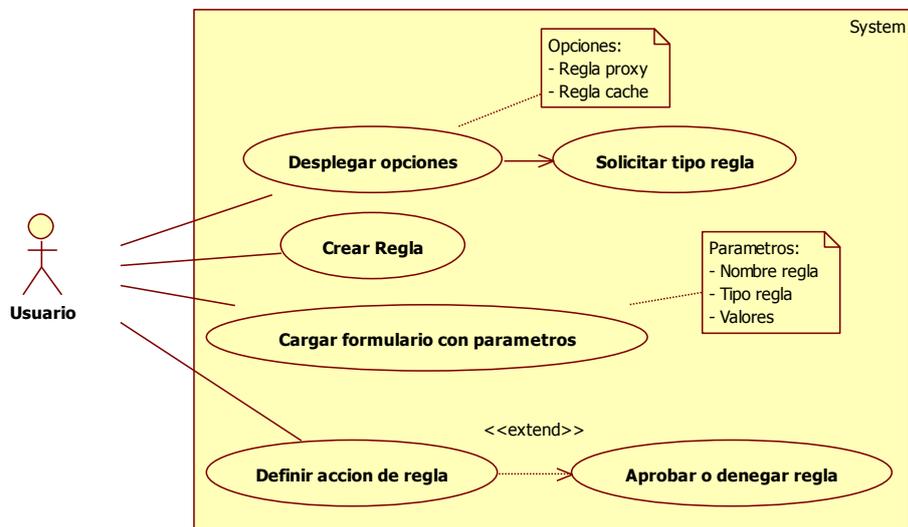


Grafico N° 4.9: Diagrama de caso de uso 3.

Caso de uso 4: Crear Pools de Retardo.

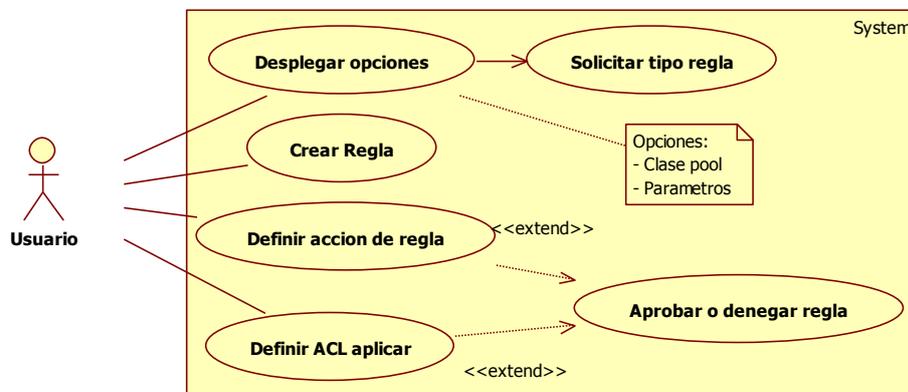


Grafico N° 4.10: Diagrama de caso de uso 4.

Caso de uso 5: Aplicar Configuración.

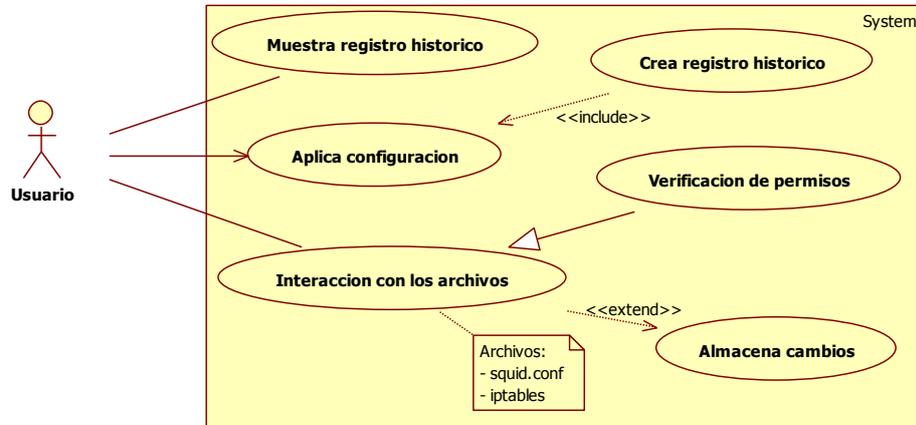


Grafico N° 4.11: Diagrama de caso de uso 5.

Caso de uso 6: Información del servidor.

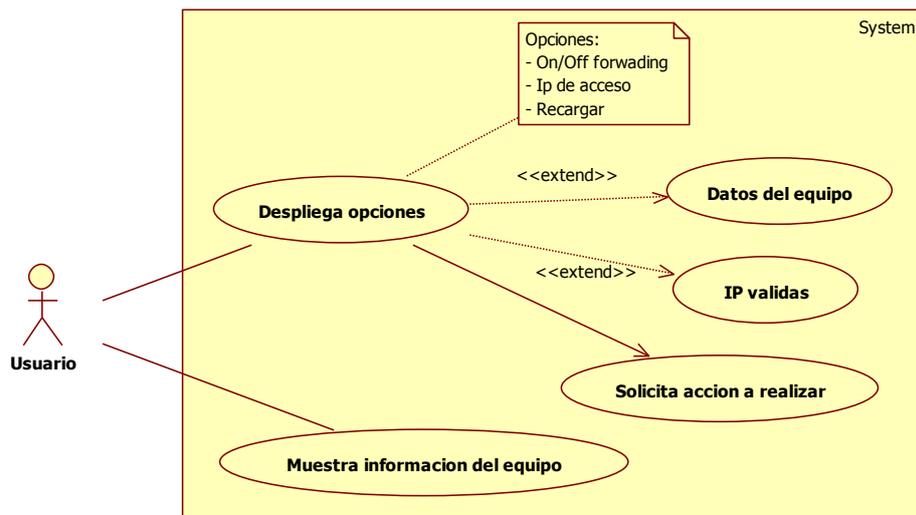


Grafico N° 4.12: Diagrama de caso de uso 6.

Caso de uso 7: Módulos registrados.

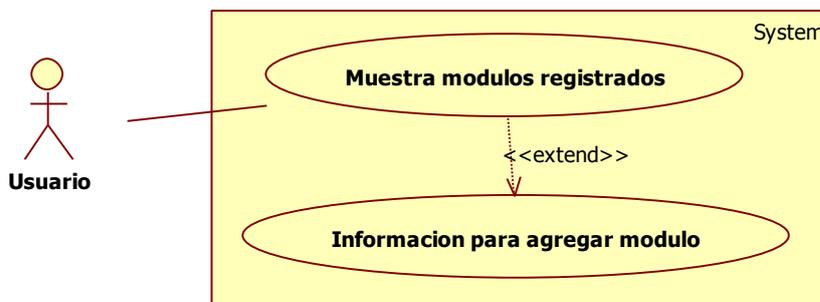


Grafico N° 4.13: Diagrama de caso de uso 7.

Caso de uso 8: Inicio y detención de servicios.

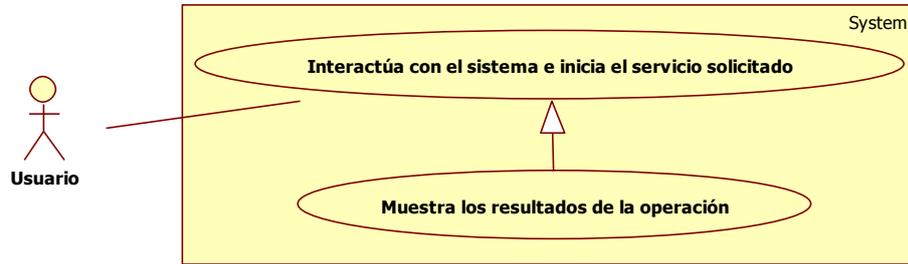


Grafico N° 4.14: Diagrama de caso de uso 8.

Caso de uso 9: Administración de usuarios.



Grafico N° 4.15: Diagrama de caso de uso 9.

Caso de uso 10: Administrar Subredes.

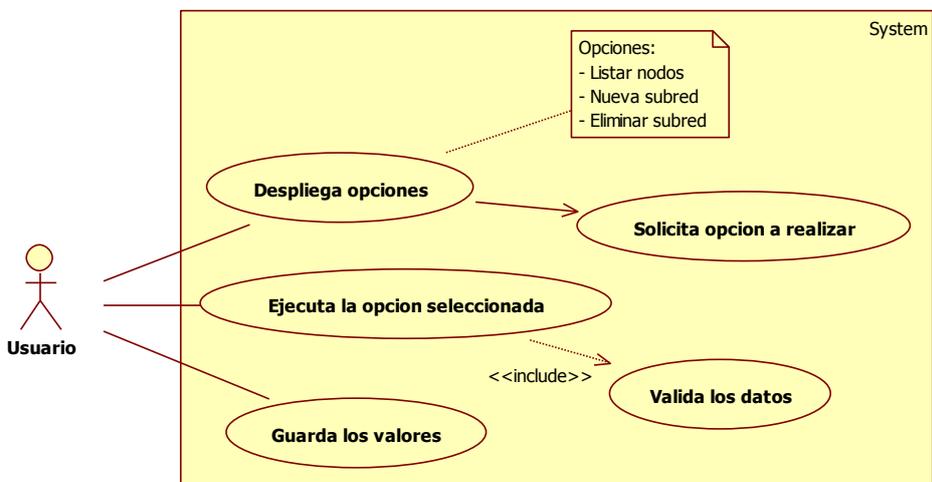


Grafico N° 4.16: Diagrama de caso de uso 10.

Caso de uso 11: Verificación de Log.

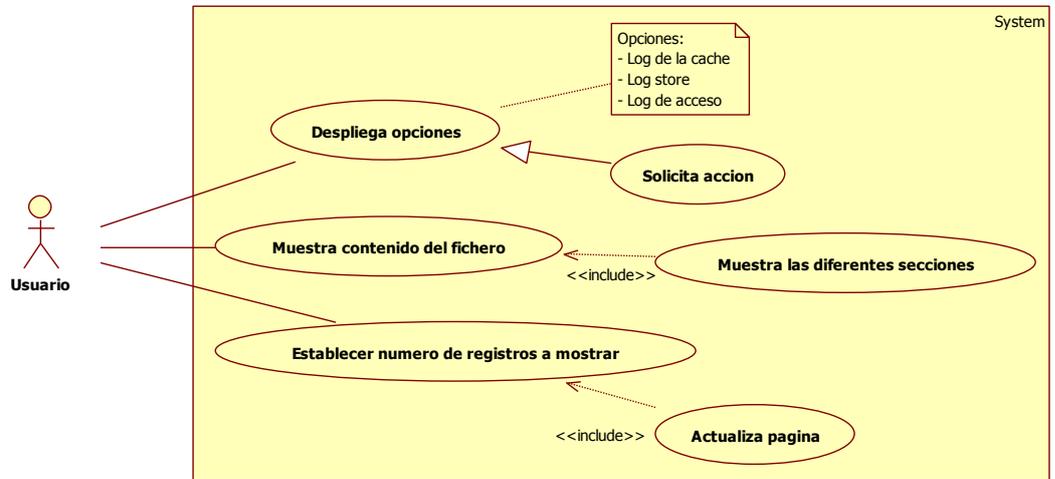


Grafico N° 4.17: Diagrama de caso de uso 11.

Caso de uso 12: Monitoreo de usuarios.

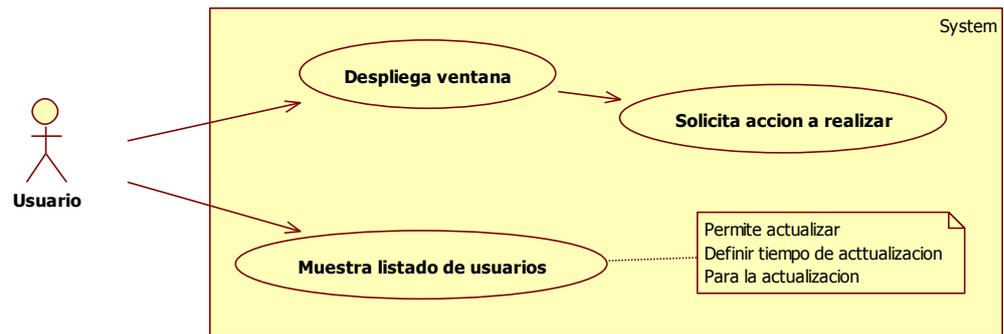


Grafico N° 4.18: Diagrama de caso de uso 12.

4.5.5. Diagramas de secuencia.

El diagrama de secuencia muestra la forma en que se comunican los objetos al transcurrir el tiempo en el orden de los eventos del sistema. El evento del sistema es una entrada externa que origina una operación del sistema como respuesta al evento, representados en secuencias, el detalle del diagrama depende de la fase en la que estemos, lo que pretendamos contar con el diagrama y a quien.

Los diagramas de secuencia desarrollados para “Proxynet” son presentados en los siguientes gráficos.

Caso de uso 1: Validación de datos con el sistema.

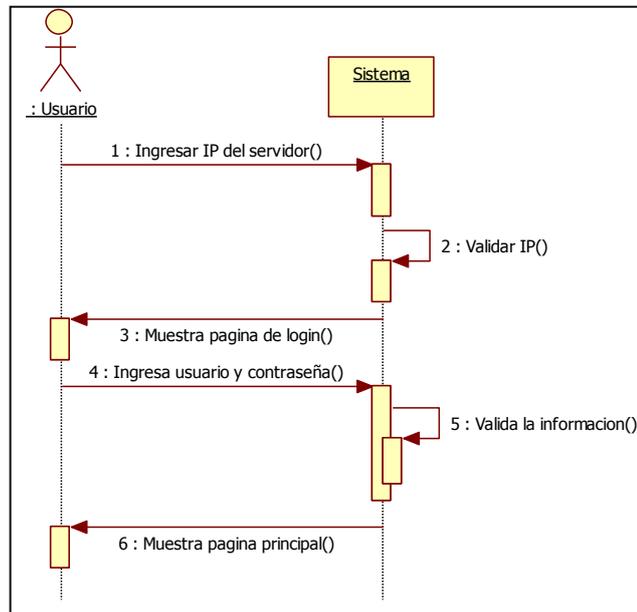


Grafico N° 4.19: Diagrama de secuencia del caso de uso 1.

Caso de uso 2: Generar Configuración Squid.

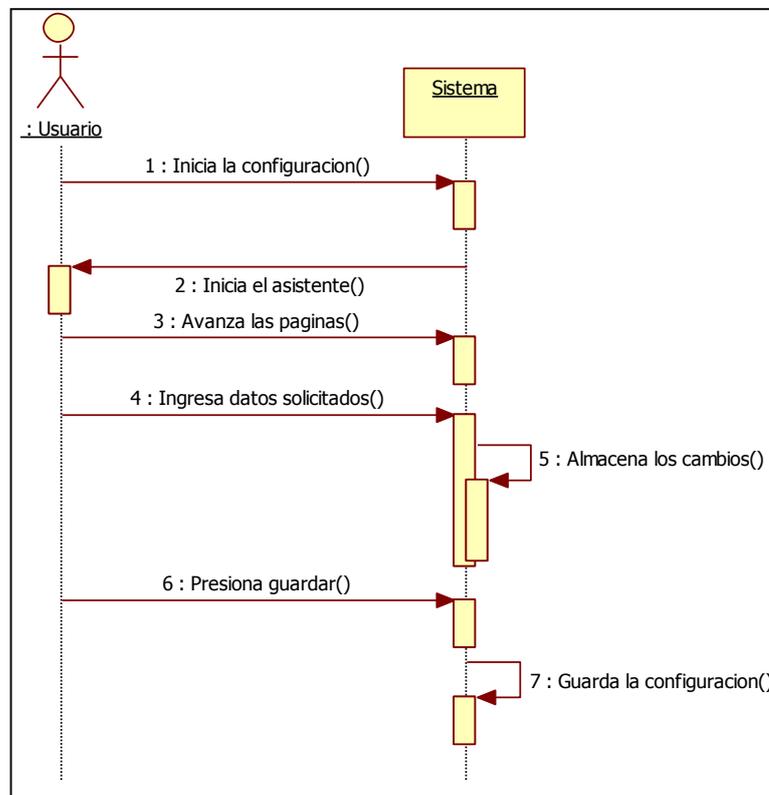


Grafico N° 4.20: Diagrama de secuencia del caso de uso 2.

Caso de uso 3: Crear Reglas de Control de Acceso.

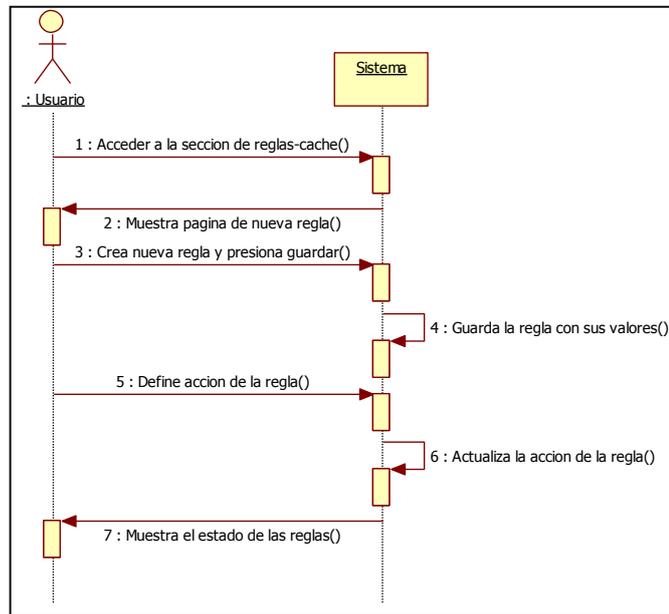


Grafico N° 4.21: Diagrama de secuencia del caso de uso 3.

Caso de uso 4: Crear Pools de Retardo.

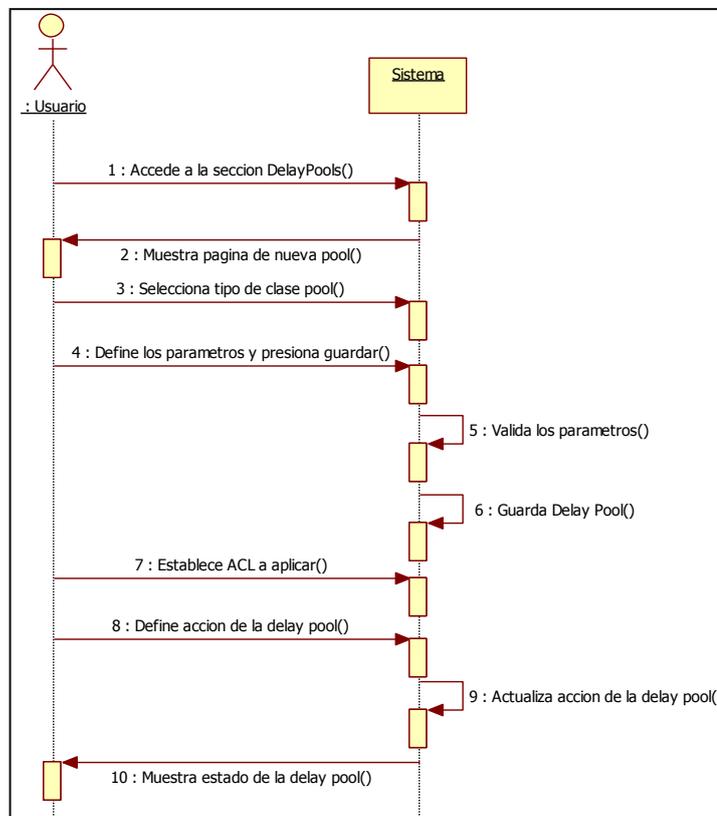


Grafico N° 4.22: Diagrama de secuencia del caso de uso 4.

Caso de uso 5: Aplicar Configuración.

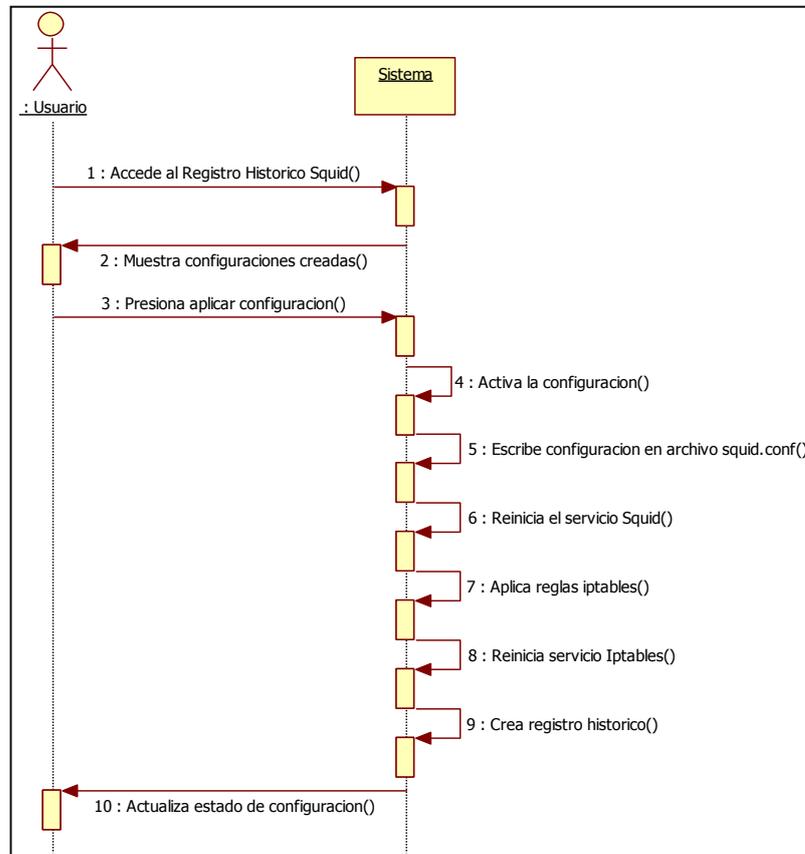


Grafico N° 4.23: Diagrama de secuencia del caso de uso 5.

Caso de uso 6: Información del servidor.

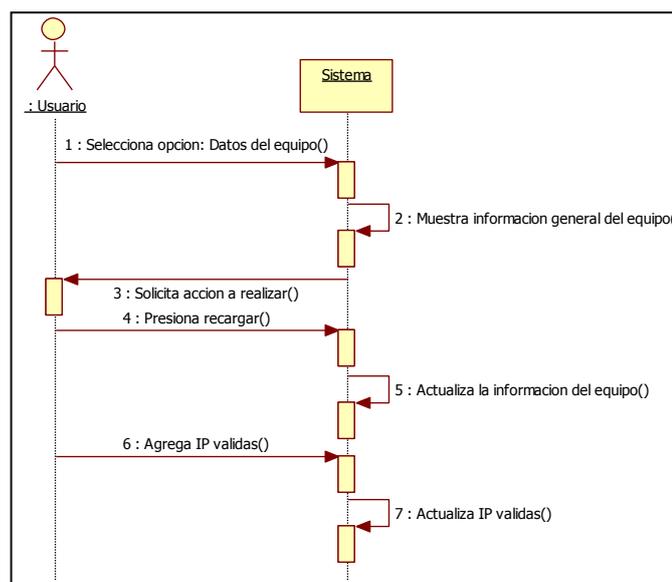


Grafico N° 4.24: Diagrama de secuencia del caso de uso 6.

Caso de uso 7: Módulos registrados.

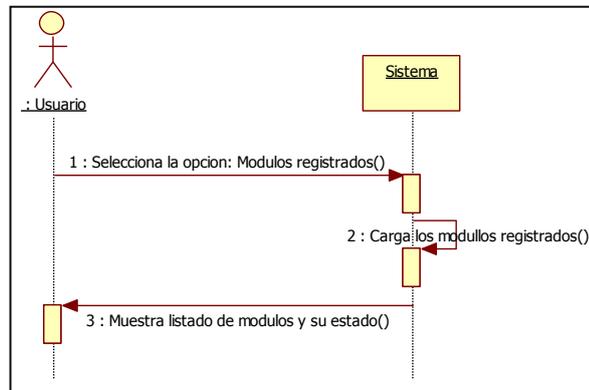


Grafico N° 4.25: Diagrama de secuencia del caso de uso 7.

Caso de uso 8: Inicio y detención de servicios.

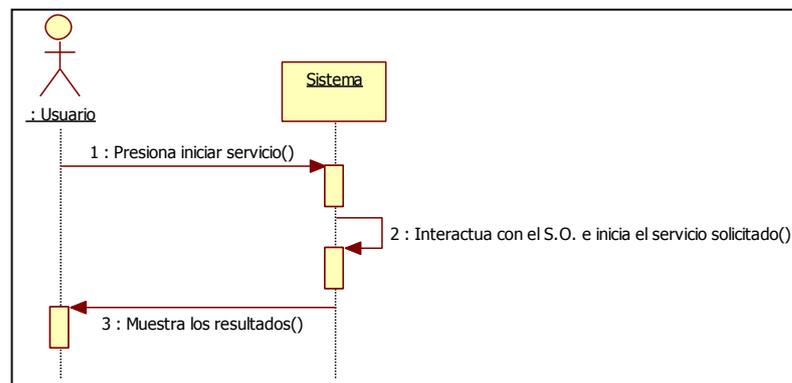


Grafico N° 4.26: Diagrama de secuencia del caso de uso 8.

Caso de uso 9: Administración de usuarios.

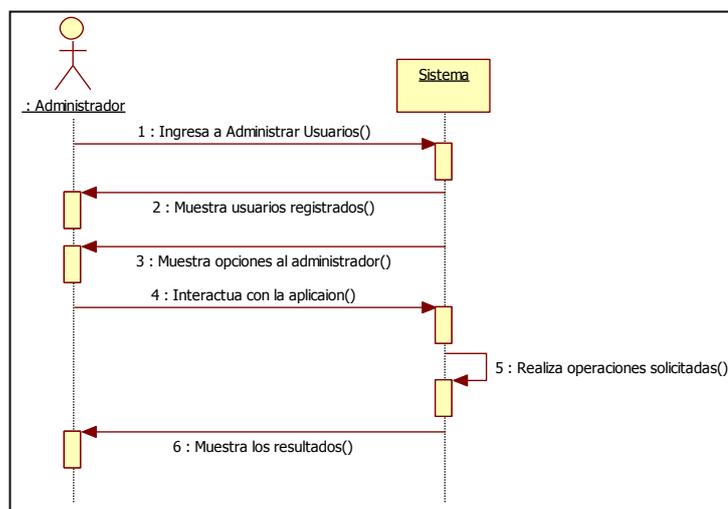


Grafico N° 4.27: Diagrama de secuencia del caso de uso 9.

Caso de uso 10: Administrar Subredes.

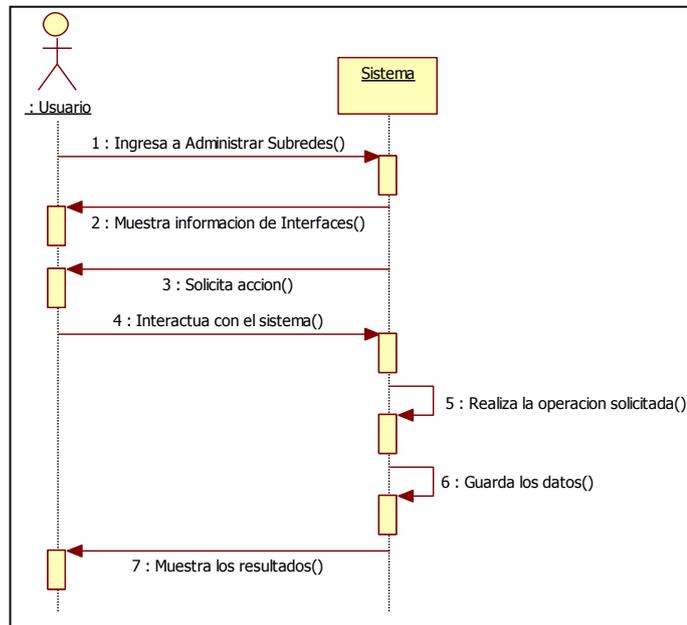


Grafico N° 4.28: Diagrama de secuencia del caso de uso 10.

Caso de uso 11: Verificación de Log.

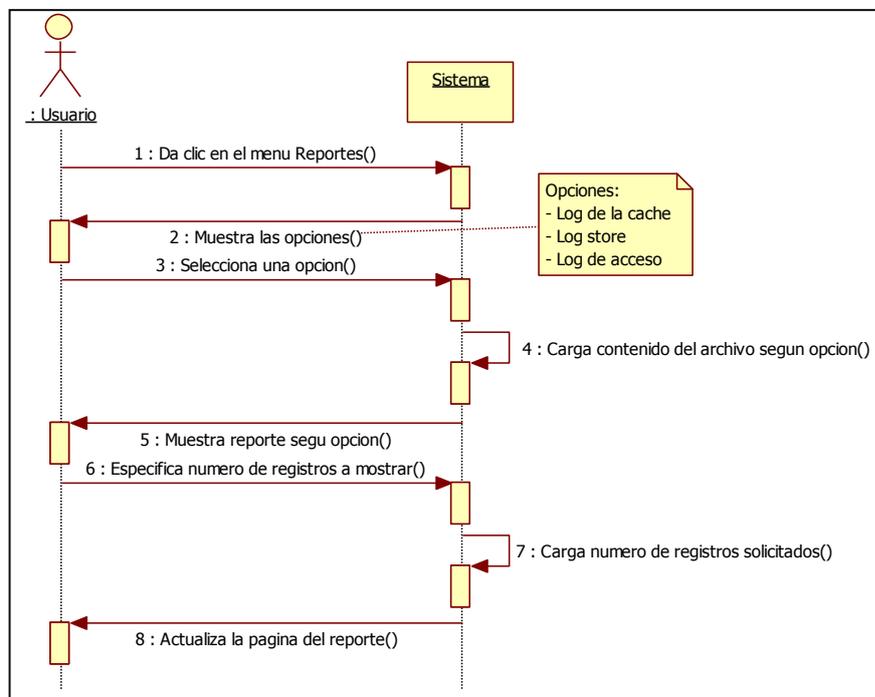


Grafico N° 4.29: Diagrama de secuencia del caso de uso 11.

Caso de uso 12: Monitoreo de usuarios.

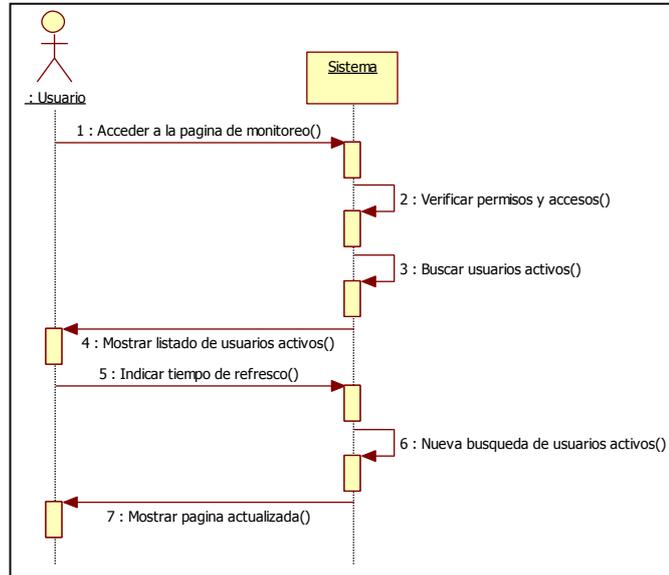


Grafico N° 4.30: Diagrama de secuencia del caso de uso 12.

4.5.6. Diagramas de actividad.

Un diagrama de actividad representa los flujos de trabajo paso a paso de los componentes en un sistema. Muestra el flujo de control general.

Caso de uso 1: Validación de datos con el sistema.

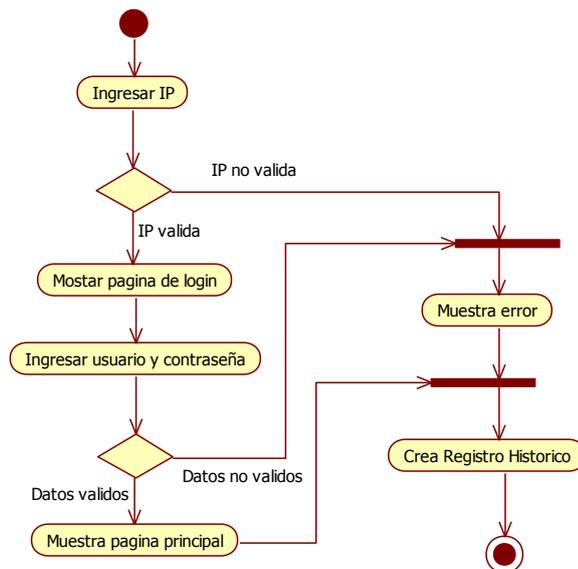


Grafico N° 4.31: Diagrama de actividad del caso de uso 1.

Caso de uso 2: Generar Configuración Squid.

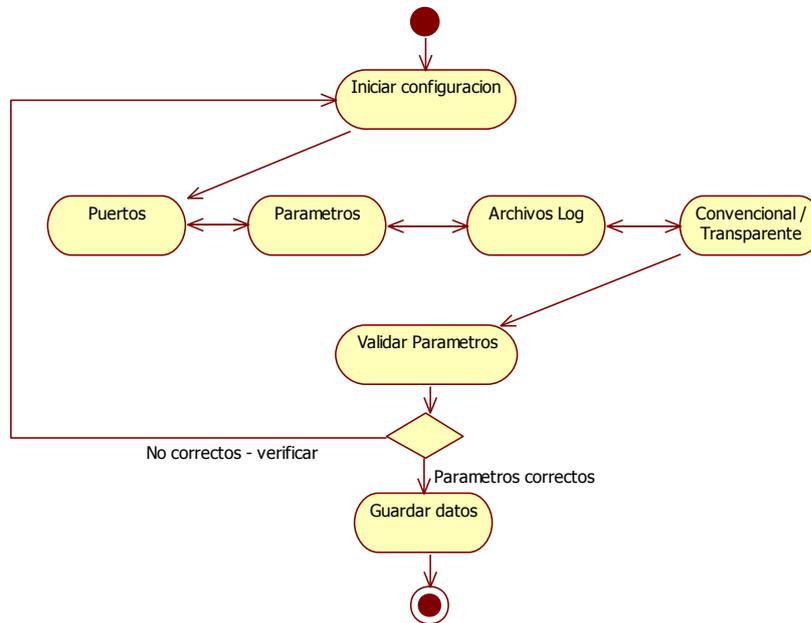


Grafico N° 4.32: Diagrama de actividad del caso de uso 2.

Caso de uso 3: Crear Reglas de Control de Acceso.

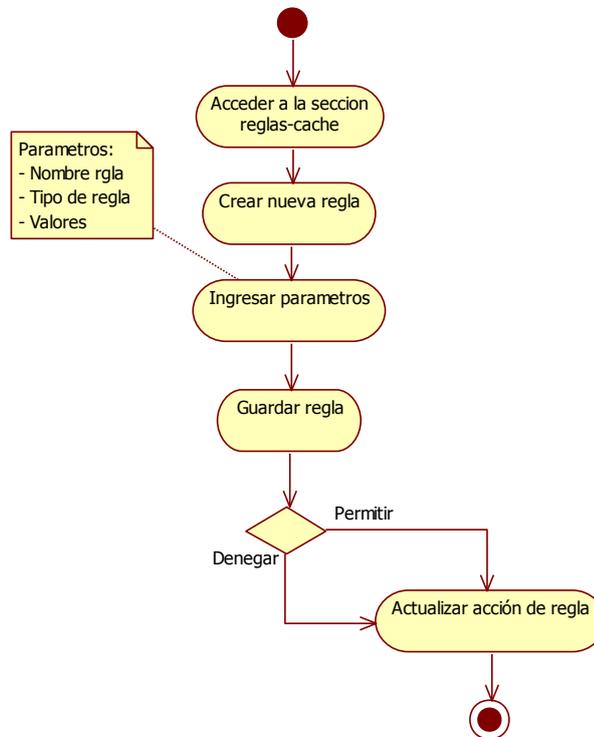


Grafico N° 4.33: Diagrama de actividad del caso de uso 3.

Caso de uso 4: Crear Pools de Retardo.

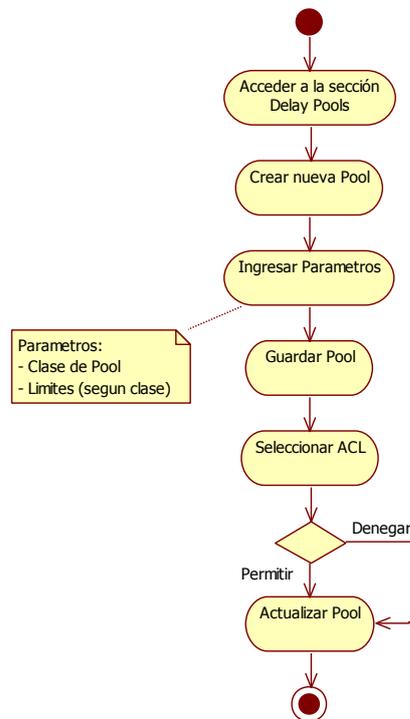


Grafico N° 4.34: Diagrama de actividad del caso de uso 4.

Caso de uso 5: Aplicar Configuración.

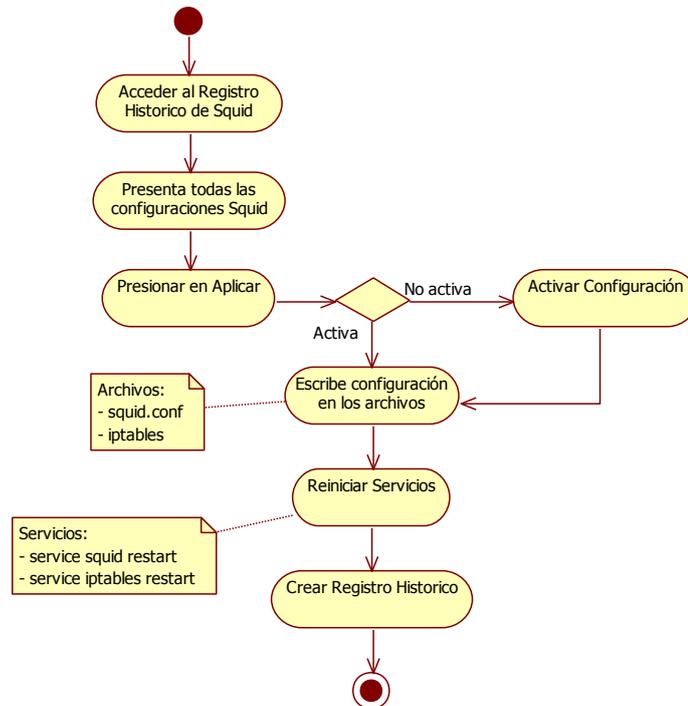


Grafico N° 4.35: Diagrama de actividad del caso de uso 5.

Caso de uso 6: Información del servidor.

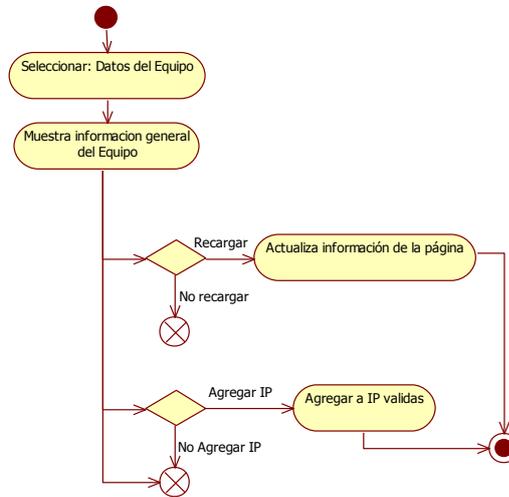


Grafico N° 4.36: Diagrama de actividad del caso de uso 6.

Caso de uso 7: Módulos registrados.

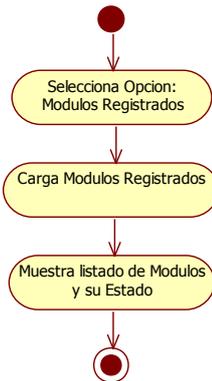


Grafico N° 4.37: Diagrama de actividad del caso de uso 7.

Caso de uso 8: Inicio y detención de servicios.

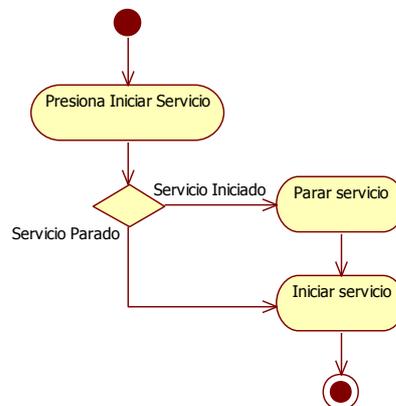


Grafico N° 4.38: Diagrama de actividad del caso de uso 8.

Caso de uso 9: Administración de usuarios.

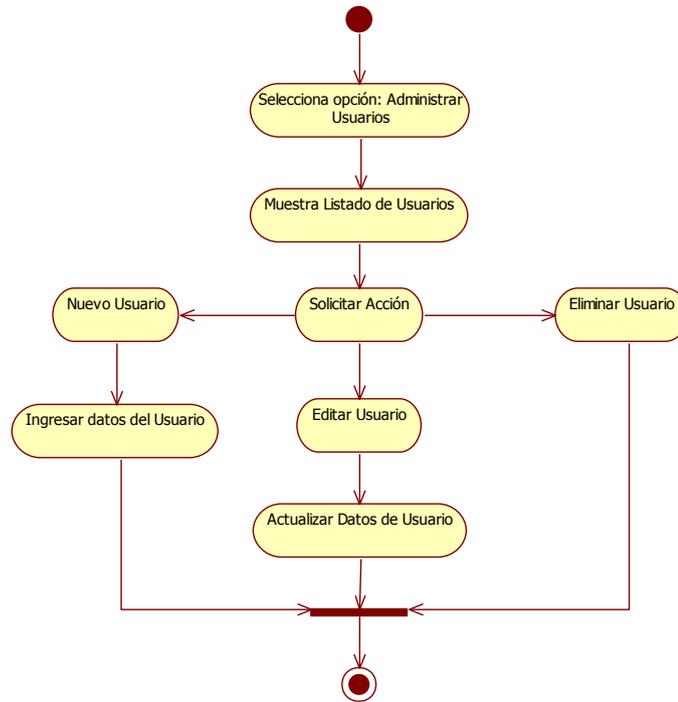


Grafico N° 4.39: Diagrama de actividad del caso de uso 9.

Caso de uso 10: Administrar Subredes.

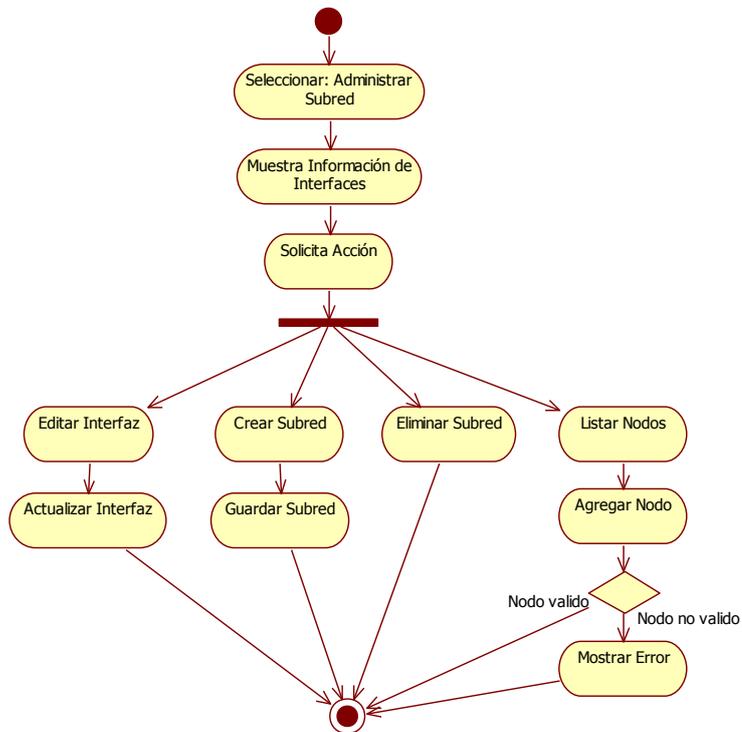


Grafico N° 4.40: Diagrama de actividad del caso de uso 10.

Caso de uso 11: Verificación de Log.

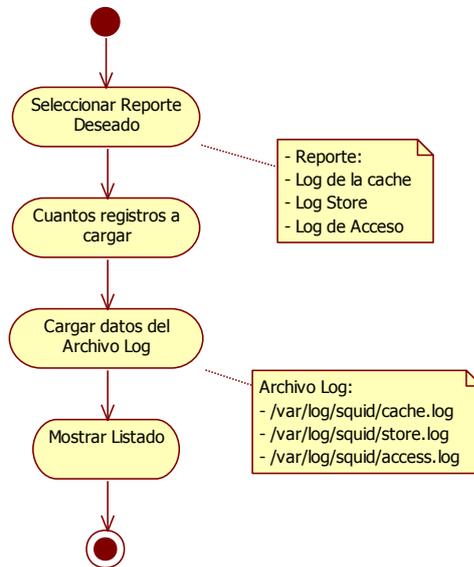


Grafico N° 4.41: Diagrama de actividad del caso de uso 11.

Caso de uso 12: Monitoreo de usuarios.

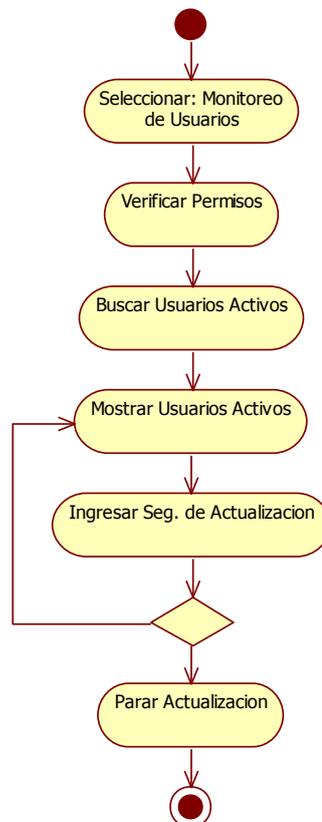


Grafico N° 4.42: Diagrama de actividad del caso de uso 12.

4.5.7. Diagrama de despliegue.

El diagrama de despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

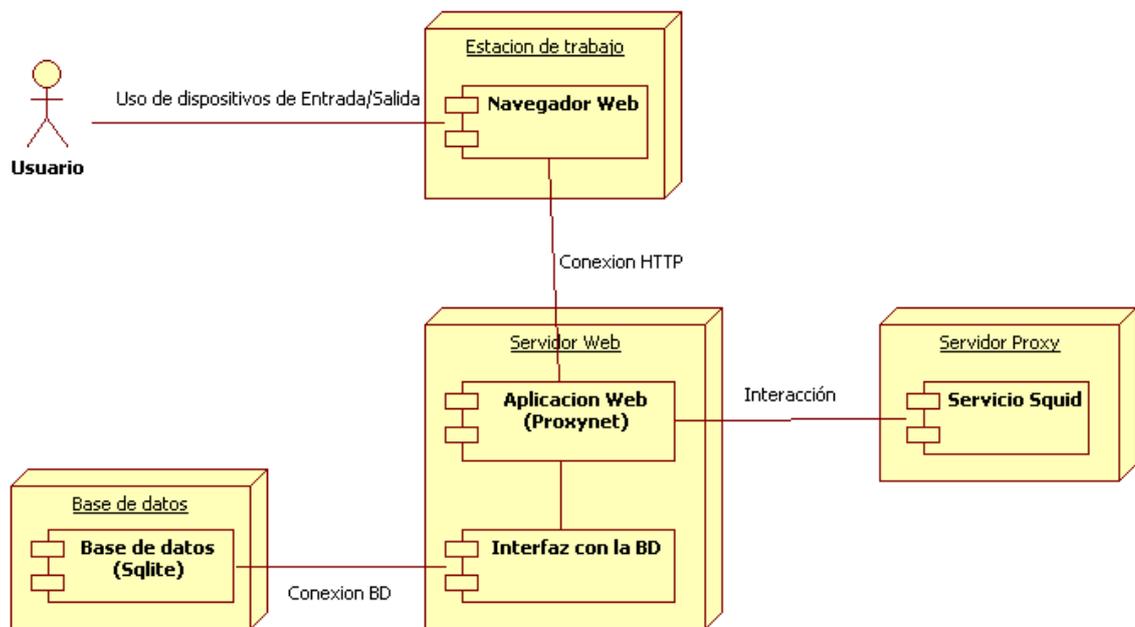


Grafico N° 4.43: Diagrama de despliegue.

4.5.8. Diseño de interfaces.

Esta fase presenta el entorno gráfico que “Proxynet” tiene:

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

A continuación se pueden ver dos ejemplos gráficos de Proxynet, en las Figuras que a continuación siguen.



Grafico N° 4.44: Diseño de la interfaz “Proxynet”.

4.5.9. Diseño de Salidas.

En esta fase presentamos los reportes generados por el servidor proxy y desplegados de una manera comprensible por la aplicación Proxynet.

Registro de la caché

Filtros de Búsqueda

Número de registros:

Registros

Fecha	Registro
12-07-2012 17:29:52	CACHEMGR: @192.168.1.1 requesting 'active_requests'
12-07-2012 17:29:53	CACHEMGR: @192.168.1.1 requesting 'active_requests'
12-07-2012 17:29:54	CACHEMGR: @192.168.1.1 requesting 'active_requests'
12-07-2012 17:29:55	CACHEMGR: @192.168.1.1 requesting 'active_requests'
12-07-2012 17:29:56	CACHEMGR: @192.168.1.1 requesting 'active_requests'
12-07-2012 17:29:57	CACHEMGR: @192.168.1.1 requesting 'active_requests'
12-07-2012 17:29:58	CACHEMGR: @192.168.1.1 requesting 'active_requests'
12-07-2012 17:29:59	CACHEMGR: @192.168.1.1 requesting 'active_requests'
12-07-2012 17:30:00	CACHEMGR: @192.168.1.1 requesting 'active_requests'
12-07-2012 17:30:02	CACHEMGR: @192.168.1.1 requesting 'active_requests'

Grafico N° 4.45: Diseño de salida: Reporte del archivo cache.log.

Registro de almacenamiento

Filtros de Búsqueda

Número de registros:

Registros

Fecha	Acción	Estado	Tipo	MIME	Método
12-07-2012 17:29:52	RELEASE	unparsable	200	text/plain	GET
12-07-2012 17:29:53	RELEASE	unparsable	200	text/plain	GET
12-07-2012 17:29:54	RELEASE	unparsable	200	text/plain	GET
12-07-2012 17:29:55	RELEASE	unparsable	200	text/plain	GET
12-07-2012 17:29:56	RELEASE	unparsable	200	text/plain	GET
12-07-2012 17:29:57	RELEASE	unparsable	200	text/plain	GET
12-07-2012 17:29:58	RELEASE	unparsable	200	text/plain	GET
12-07-2012 17:29:59	RELEASE	unparsable	200	text/plain	GET
12-07-2012 17:30:00	RELEASE	unparsable	200	text/plain	GET
12-07-2012 17:30:02	RELEASE	unparsable	200	text/plain	GET

Grafico N° 4.46: Diseño de salida: Reporte del archivo store.log.

Registro de acceso

Filtros de Búsqueda

Número de registros:

Registros

Fecha	Respuesta Tiempo/Tamaño	IP Origen	Squid Request/ Http Status	URL	MIME
12-07-2012 17:33:06	0.294/2432	192.168.1.2	TCP_MISS/ 200	Enlace	text/javascript
12-07-2012 17:33:06	0.306/2245	192.168.1.2	TCP_MISS/ 200	Enlace	text/javascript
12-07-2012 17:33:06	0.405/2245	192.168.1.2	TCP_MISS/ 200	Enlace	text/javascript
12-07-2012 17:33:06	0.196/359	192.168.1.2	TCP_REFRESH_HIT/ 304	Enlace	.
12-07-2012 17:33:30	0.485/968	192.168.1.2	TCP_MISS/ 200	Enlace	text/html
12-07-2012 17:33:31	0.581/7856	192.168.1.2	TCP_MISS/ 200	Enlace	text/javascript
12-07-2012 17:33:35	3.421/533	192.168.1.2	TCP_MISS/ 404	Enlace	text/html

Grafico N° 4.47: Diseño de salida: Reporte del archivo access.log.

Registro Histórico de Accesos al Sistema

[Ultimos Registros](#)
[Buscar por Fecha](#)
[Buscar por Usuario](#)

Buscar por usuario

En la presente página puede realizar consultas sobre los intentos de acceso efectuados por un usuario.

Criterio de Búsqueda

Nombre de Usuario:

Listado de Registros

Nº	Fecha / Hora	Usuario	IP	Mensaje
1	09-07-2012 20:55	admin	127.0.0.1	El usuario admin ha iniciado sesión satisfactoriamente
2	09-07-2012 09:18	admin	127.0.0.1	El usuario admin ha iniciado sesión satisfactoriamente
3	09-07-2012 08:29	admin	127.0.0.1	El usuario admin ha iniciado sesión satisfactoriamente

Grafico N° 4.48: Diseño de salida: Registro histórico de acceso al sistema.

4.6. DESARROLLO.

La aplicación Web “Proxynet” fue colocada en un servidor cuya plataforma es Linux CentOS 5.4, la misma que contiene los servicios básicos de funcionamiento como son: Servidor Web Apache y PHP. Vale mencionar que todas las pruebas de la aplicación Web se las hicieron en el entorno descrito en ésta investigación; hecho por el que no se puede asegurar su eficiencia en ambientes Linux distintos.

4.6.1. Pruebas.

Se efectuaron pruebas básicas de configuración para el módulo Squid (Proxy), que “Proxynet” incorpora con la finalidad de demostrar la administración de determinados parámetros mediante la utilización de un entorno Web.

El escenario planteado para esta prueba es el siguiente: 3 equipos, un equipo cumple con la función de un Equipo Remoto (Internet), el segundo equipo cumple con la función de enrutador, es decir, permite la comunicación entre un equipo local y un equipo remoto, este equipo tiene incorporado 2 tarjetas de red, la primera tarjeta permite la conexión hacia la intranet y la segunda tarjeta permite la conexión hacia nuestro equipo remoto (Internet), además este equipo tiene incorporado el servicio Squid para administrar el acceso hacia recursos Web y el firewall Iptables para realizar el filtrado de paquetes.

Finalmente el tercer equipo es el equipo cliente perteneciente a la Intranet y cuya Puerta de enlace es el equipo que cumple con las funciones de enrutador. A continuación, en el Grafico N° 4.49 se presenta el esquema gráfico de este escenario:

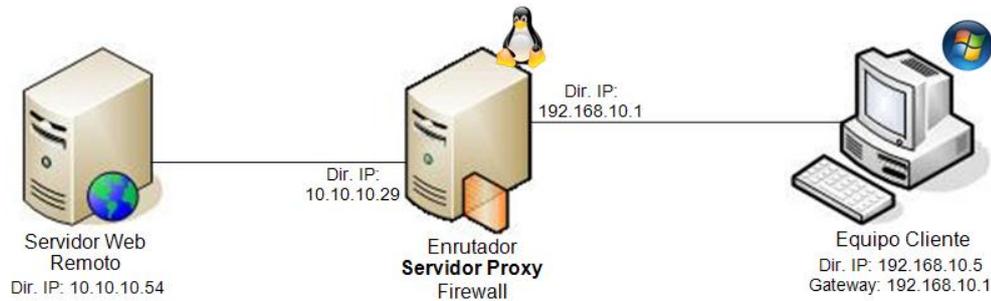


Grafico N° 4.49: Escenario de pruebas para el servidor proxy.

Una vez especificado el escenario, procedemos a las pruebas respectivas. Desde el equipo Servidor accedemos a la aplicación “Proxynet” mediante el navegador web con el usuario admin y procedemos a ingresar a la sección Datos del equipo para cargar los datos del servidor a la aplicación para ello presionamos en el botón **Recargar...**, luego ingresamos la dirección IP del equipo cliente en la sección de IP validas, con ello la IP de este equipo tiene acceso a la aplicación, creamos un nuevo usuario para que tenga acceso a la aplicación.



Grafico N° 4.50: Pagina con datos del servidor y las ip validas para acceder a la aplicación.

Una vez completado lo anterior podremos configurar el módulo Squid, el cual permite administrar el contenido hacia recursos Web. Para este fin, podemos ver en el Grafico N° 4.51, varias de las opciones que se pueden manipular con el sistema.

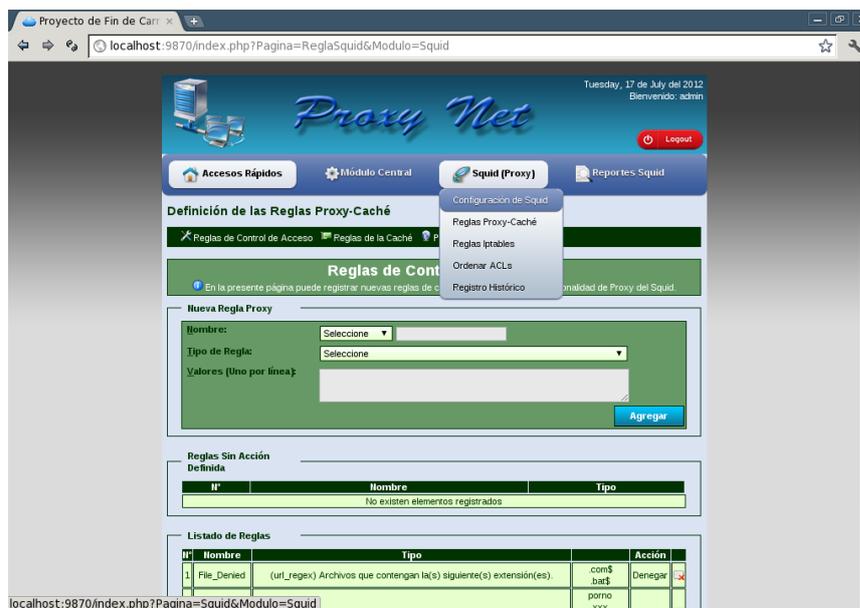


Grafico N° 4.51: Configuración de reglas Squid y menú de Squid.

Para tener un mejor soporte respecto al tráfico de paquetes, al módulo para configurar Firewall IPv4 se especifica ciertas reglas que permitan redireccionar todo el tráfico HTTP (ver Grafico N° 4.52) proveniente de la intranet apunte hacia el servidor Proxy, el cual será el encargado de administrar el contenido existente en dicho tráfico.

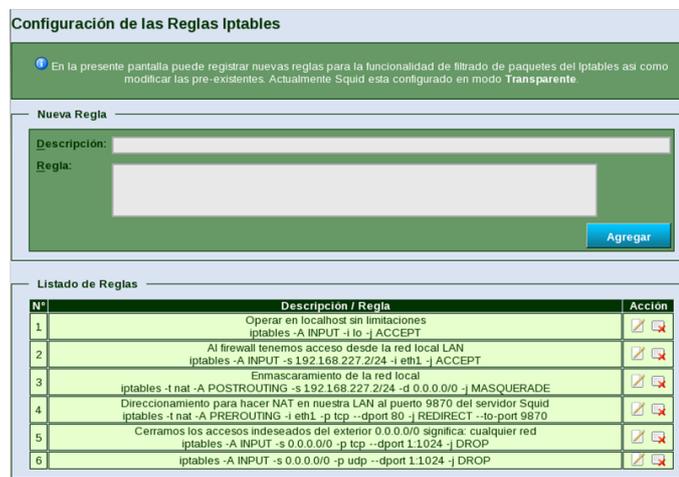


Grafico N° 4.52: Modulo de configuración del firewall.

Para que la configuración del Squid funcione en los equipos clientes, debemos configurar en el navegador web, la dirección IP y el puerto de nuestro servidor Proxy, tal como vemos en el Grafico N° 4.53.

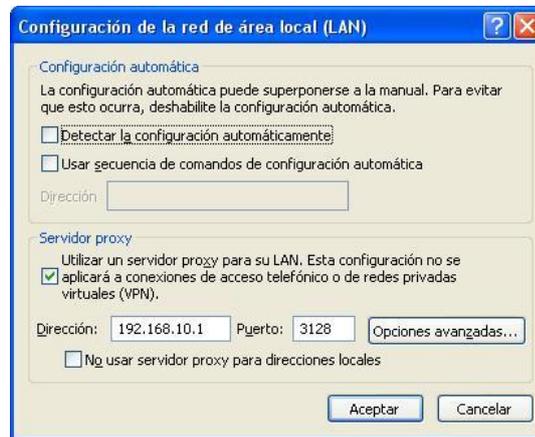


Grafico N° 4.53: Redireccionamiento del navegador hacia el proxy.

Por el momento se tiene creado una regla, la cual permite el acceso por parte de los clientes hacia sitios web remotos. Esto se verifica accediendo desde el cliente en el que se configuró el proxy, (en nuestro ejemplo) solicitando la página de inicio del servidor web, como se ve en el Grafico N° 4.54.

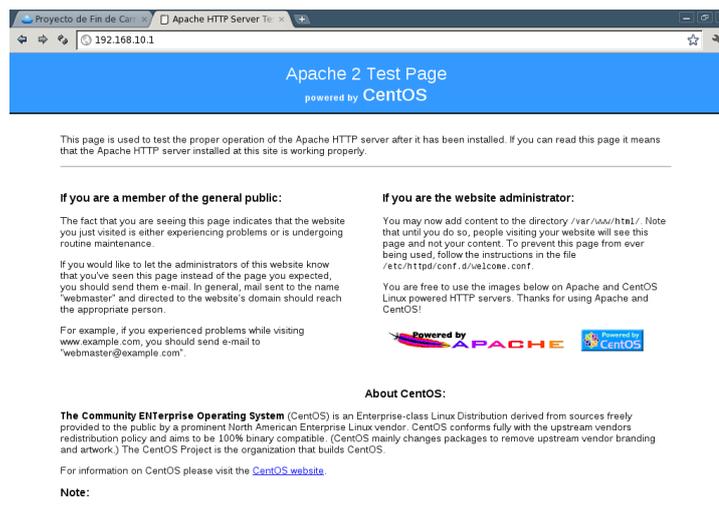


Grafico N° 4.54: Acceso a equipo remoto desde equipo cliente usando proxy.

Si deseamos restringir el acceso hacia una determinada dirección IP por parte de nuestros clientes, configuramos una regla que cumpla este propósito,

mediante nuestra aplicación usando el formulario de creación de ACL's mostrada en el Grafico N° 4.55. Creamos una ACL cuyo prefijo será IP_negada en la cual especificaremos que dirección IP deseamos bloquear el acceso, para nuestro caso la dirección IP será la 10.10.10.54.

Listado de Reglas					
N°	Nombre	Tipo		Acción	
1	red_local	(src) Acceder a Internet desde la siguiente IP o segmento de red local.	192.168.10.0/24	Permitir	
2	lp_negados	(dstdomain) Acceder desde Internet al siguiente dominio de la red.	10.10.10.54	Denegar	

Grafico N° 4.55: Dos ACL creadas para la prueba realizada.

Verificamos la creación de nuestra acl IP_negadas, accediendo a la sección Ordenar Restricciones Proxy-Caché, mostrada en el Grafico N° 4.56.

Ordenar Restricciones Proxy-Caché			
Restricciones Proxy			
N°	Acción	Nombre ACLs	Mover
1	Denegar	lp_negados	
2	Permitir	red_local	

Grafico N° 4.56: Listado de reglas ACL registradas.

Debemos tomar en consideración que las reglas ACL se ejecutan desde la más específica hacia la más general, por ello es importante ordenarlas, esto se consigue al dar clic sobre las fechas que indican el orden de las ACL.

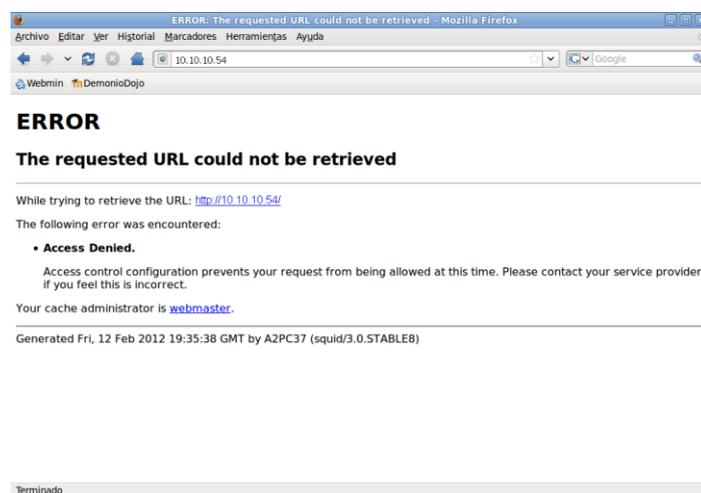


Grafico N° 4.57: Bloqueo a dirección remota por regla ACL registrada.

4.6.2. Implementación del sistema.

Tras haber definido como desarrollar la aplicación web “Proxynet”, procedemos a implementar la misma.

4.6.2.1. Codificación.

La codificación de la aplicación se utilizó el modelo MVC (Modelo Vista Controlador) como patrón o modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

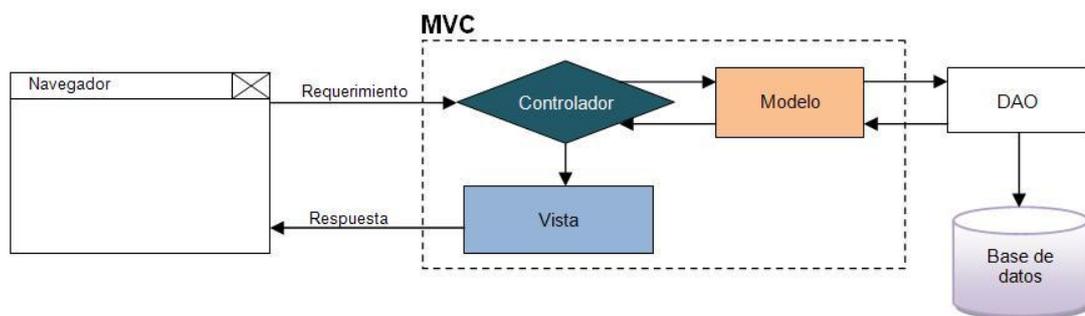


Grafico N° 4.58: Patrón de diseño MVC.

El cual nos permite dividir la aplicación en tres partes:

- **Modelo**
Encapsula los datos y la funcionalidad de la aplicación (lógica que da solución al problema).
- **Vista**
Interfaz de usuario para el ingreso y presentación de datos generados por el modelo.
- **Controlador**
Es el que controla el flujo de la aplicación y administra quien es el responsable de dar solución a un problema; se comunica con el modelo y la vista.

La codificación del Sistema se realizó en el editor de texto gedit integrado en el S.O. Linux CentOS, utilizando el lenguaje PHP, parte del código se puede ver en las imágenes mostradas a continuación.

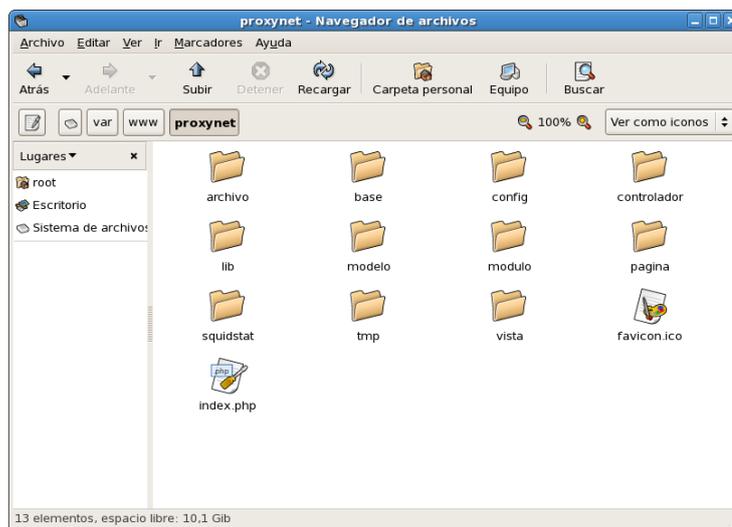


Grafico N° 4.59: Organización de los datos en la raíz del servidor web apache.

Se puede observar que se hace una clase entrada la cual corresponde a la Página de Entrada al Sistema, esta es una página conocida como manejadora, la cual hace uso de clases para poder realizar las operaciones debidas.

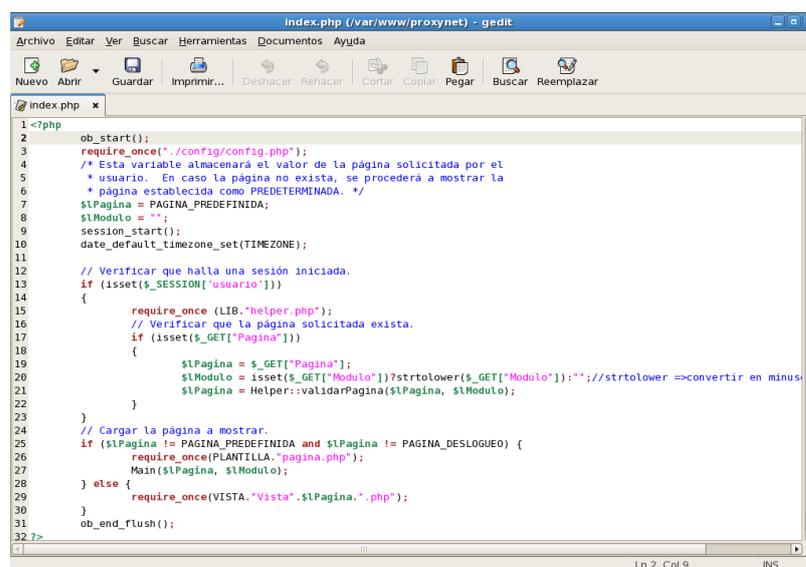


Grafico N° 4.60: Editor de textos gedit con parte del código de la aplicación.

La página contiene referencias de las clases que esta manejadora hace uso, para poder mostrarnos la página index.php, que es la página en la que un usuario ingresa sus datos para ingresar a la aplicación. Ahora veremos varios ejemplos de programación en PHP.

Procedimiento que permite leer las líneas de un archivo.

```
public function leerLineas()
{
    try {
        $lLineas = array();
        $this->abrirArchivo();
        while (!feof($this->aHandler))
        {
            $lLineas[] = $this->leerLinea();
        }
        $this->cerrarArchivo();
        return $lLineas;
    } catch (exception $e) {
        throw $e;
    }
}
```

La pagina principal utilizada para mostrar la interfaz.

```
<?php
require once(PLANTILLA."cabecera.php");
require_once(PLANTILLA."pie.php");
require once(PLANTILLA."menu.php");

function cargarVista($pPagina, $pModulo)
{
    $lVista = "";
    if ($pModulo == "")
    {
        $lVista = VISTA."Vista".$pPagina.".php";
    } else {
        $lVista = MODULO.$pModulo."/vista/Vista".$pPagina.".php";
    }
    require_once($lVista);
}

function cargarControlador($pPagina, $pModulo)
{
    $lControlador = "";
    if ($pModulo == "")
    {
        $lControlador = CONTROLADOR."Controlador".$pPagina.".php";
    } else {
        $lControlador =
MODULO.$pModulo."/controlador/Controlador".$pPagina.".php";
    }
}
```

```

        require_once ($lControlador);
    }

function Main($pPagina, $lModulo = "")
{
    $lVistaPagina = "Vista".$pPagina;
    $lControladorPagina = "Controlador".$pPagina;
    cargarControlador($pPagina,$lModulo);
    cargarVista($pPagina,$lModulo);
    eval('$lControlador = '.$lControladorPagina.'.::obtenerInstancia();');
    $lVista = new $lVistaPagina($lControlador);
    $lPath = "JS".(trim($lModulo) != ""?"_".strtoupper(trim($lModulo)):"");
    eval('$lEvento = '.$lPath.'.');
    $lEvento .= "evento".$pPagina.".js";
?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <title>Proyecto de Fin de Carrera: Proxy Net</title>
    <meta name="GENERATOR" content="Quanta Plus">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <link rel="stylesheet" type="text/css" href="<?CSS?>menu.css" >
    <link rel="stylesheet" type="text/css" href="<?CSS?>estilo.css" >
    <link rel="stylesheet" type="text/css" href="<?CSS?>icon_menu.css" >
    <link rel="stylesheet" type="text/css" href="<?CSS?>domtab.css" >
    <link rel="stylesheet" type="text/css" href="<?CSS?>tabla.css" >
    <script language="javascript" type="text/javascript" src="<?=$lEvento?>">
    </script>
    <script type="text/javascript" src="/vista/js/menu.js"></script>
    <script type="text/javascript" src="/vista/js/domtab.js"></script>
    <script type="text/javascript" src="/vista/js/popup.js"></script>
</head>
<body>
    <table border="0" cellpadding="0" cellspacing="0" width="100%"
bgcolor="#dae3f2">
        <tr>
            <td colspan="2" style="background-
image:url('/pagina/imagenes/header back.jpg'); width:100%; height:113px; padding-
left:5px; padding-right:5px;"><?php Cabecera();?>
            </td>
        </tr>
        <tr>
            <td>
                <?php
                $lUsuario = unserialize($_SESSION["usuario"]);
                Menu($lUsuario->Administrador);
                ?>
            </td>
        </tr>
        <tr>
            <td colspan="2">
                <?php if (isset($_SESSION["advertencia"])) {
                if ($_SESSION["advertencia"] != "") { ?>
                <div class="advertencia">
                
                &nbsp;<?=$_SESSION["advertencia"];?> </div>
                <?php unset($_SESSION["advertencia"]);

```

```

    }
    } else if (isset($_SESSION["info"])) {
    if ($_SESSION["info"] != "") { ?>
    <div class="info">
    
    &nbsp;<?=$_SESSION["info"];?> </div>
    <?php unset($_SESSION["info"]);
    }
    } else if (isset($_SESSION["error"])) {
    if ($_SESSION["error"] != "") { ?>
    <div class="error">
    
    &nbsp;<?=$_SESSION["error"];?> </div>
    <?php unset($_SESSION["error"]);
    }
    }?>
</td>
</tr>
<tr>
<td width="650" valign="top">
<!-- Contenido -->
<?php
if ($lVista->restringido())
{
    if ($lUsuario->Administrador)
    {
        $lVista->contenido();
    } else {
        include PLANTILLA.'error403.php';
    }
} else {
    $lVista->contenido();
}
?>
</td>
</tr>
<tr>
<td colspan="2" style="background-
image:url('/pagina/imagenes/footer_back.jpg'); width:100%; height:40px;">
<?php Pie();?>
</td>
</tr>
</table>
</body>
</html>
<?php
}
?>

```

4.6.2.2. Ejecución de la aplicación.

Para ejecutar la aplicación se lo puede hacer tipiendo en el browser la ruta donde se encuentra “Proxynet” para de esta manera cargar el contenido de la aplicación, teniendo configurado el servidor web correctamente.

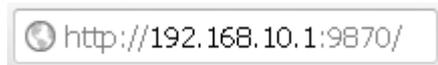
Por ejemplo:

Acceso local:



http://localhost:9870/

Acceso remoto:



http://192.168.10.1:9870/

Una vez realizado el paso anterior se cargará la pantalla inicial de Proxynet.

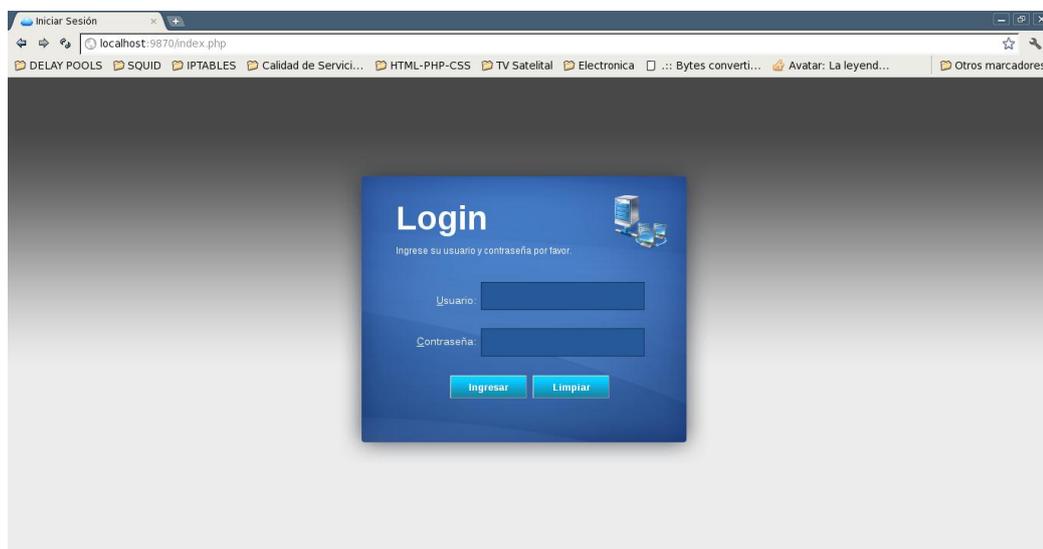


Grafico N° 4.61: Página de login de la aplicación Proxynet.

4.6.2.3. Requerimientos de Hardware.

Lo más importante es cuantificar la carga que va a tener que soportar nuestro sistema. Para esto es importante fijarse más en los picos de carga del sistema que en la media total, ya que los picos pueden llegar a ser varias veces la media del día. En caso de duda siempre es mucho mejor sobreestimar los requerimientos del sistema, ya que un Squid trabajando al límite de su capacidad puede repercutir negativamente en el funcionamiento de los servicios.

En las siguientes secciones se explican en orden de importancia los distintos factores del sistema.

Discos Duros:

Cuando se trata de cachés, la velocidad es un parámetro importantísimo. En los discos duros este parámetro se mide mediante su “tiempo medio de acceso” en milisegundos, que debe ser lo más bajo posible. Para lograr una velocidad elevada se recomienda utilizar discos duros rápidos.

Debido a que en la mayoría de los casos Squid lee o escribe pequeños bloques del disco duro, el tiempo de acceso del disco duro es más importante que su capacidad de transferencia de datos. Precisamente en este contexto muestran su valía los discos duros con una alta velocidad de rotación, ya que permiten un posicionamiento más rápido de la cabeza de lectura.

Tamaño del Caché de Disco:

El método más fácil para determinar el tamaño del caché es en función del tráfico máximo que pase por el mismo. Si se dispone de una conexión de 1 Mb/s, como mucho se transferirán 125 KB por segundo. Si todo este tráfico va a parar al caché, en una hora será 450 MB, y suponiendo que este tráfico se genera durante las 8 horas de trabajo, tendremos en total 3,6 GB diarios. Como la línea no suele trabajar al máximo, la cantidad total de datos procesada por el caché es de unos 2 GB. Así pues, para guardar todos los datos navegados por la WWW en un día, necesitamos en este ejemplo 2 GB de memoria RAM para Squid.

Memoria RAM:

La cantidad de memoria requerida por Squid está relacionada directamente con la cantidad de objetos que se encuentran en el caché. La memoria RAM es muchísimo más rápida que el disco duro. Squid también guarda muchos otros datos en la memoria, como por ejemplo una tabla con todas las direcciones IP utilizadas,

un caché para los nombres de dominio totalmente cualificados, objetos “calientes” (los que más se solicitan), buffers, listas de control de acceso, etc.

Es muy importante tener memoria más que suficiente para el proceso de Squid, ya que en el caso de tener que pasar el proceso al disco duro, las prestaciones del sistema se reducirán drásticamente.

Potencia del Procesador:

Squid no es un programa que consuma mucho CPU. Solamente al arrancar y comprobar el contenido del caché es cuando se trabaja más intensamente con el procesador.

El uso de máquinas con multiprocesador tampoco incrementa el rendimiento del sistema. Para obtener una mayor efectividad, es preferible aumentar la cantidad de memoria RAM o bien utilizar discos más rápidos antes que cambiar el procesador por otro más potente.

Características de los Equipos

A continuación los requerimientos principales del servidor.

Dispositivo	Capacidad
Procesador	Intel Pentium Core 2 Duo 2.8 GHz
Almacenamiento	SCSI 2 HD 120 GB
Memoria RAM	2 GB DDR3
2 Tarjetas de red	Fast Ethernet 10/100/1000 Mbps

A continuación las características de las computadoras disponibles.

Dispositivo	Capacidad
Procesador	Intel Core I3 3Ghz
Almacenamiento	500 GB
Memoria RAM	4 GB
Tarjeta de red	1 Fast Ethernet 10/100/1000 Mbps

A continuación las características de switch para la LAN.

Nombre:	D-Link Power Switch
Modelo:	DES-1024D
Tecnología:	10Base-T
Puertos:	24
Memoria:	2 MB
Descripción:	Este dispositivo provee conectividad a las computadoras de la Biblioteca Virtual F.A.F.I.

4.6.2.4. Requerimientos de Software.

“Proxynet” es una aplicación Web diseñada para ejecutarse en entornos Linux, por ello los requerimientos necesarios para su correcto funcionamiento son los siguientes:

Para el servidor:

- **Sistema Operativo:** Linux CentOS 5.
- **Servidor Web:** Apache 2.4.
- **Entorno de desarrollo:** PHP 5.3.
- **Servidor Proxy:** Squid 2.6 STABLE.
- **Firewall:** Iptables.

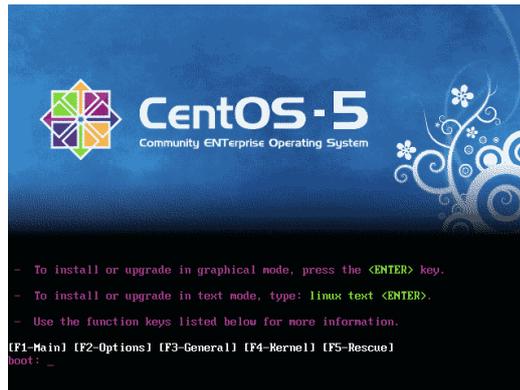
Para los computadores:

- Sistema operativo Windows XP Profesional Service Pack 3 o Superior.
- Internet Explorer 9, Google Chrome o Mozilla Firefox.

4.6.2.5. Proceso de instalación.

Instalación de CentOS 5.

Esta es la ventana de inicio de instalación de CentOS 5, presionamos Enter para iniciar el asistente de instalación.



En esta ventana tenemos que escoger la opción Skip para seguir el proceso de instalación de CentOS 5.



Inicio del proceso de instalación damos clic en Next.



Seleccionamos el idioma con el cual se presentara la interfaz de Centos 5, en este caso es Español.



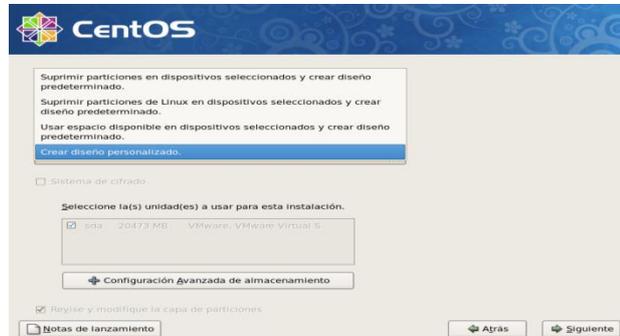
Seleccionamos el idioma del teclado en este caso Español.



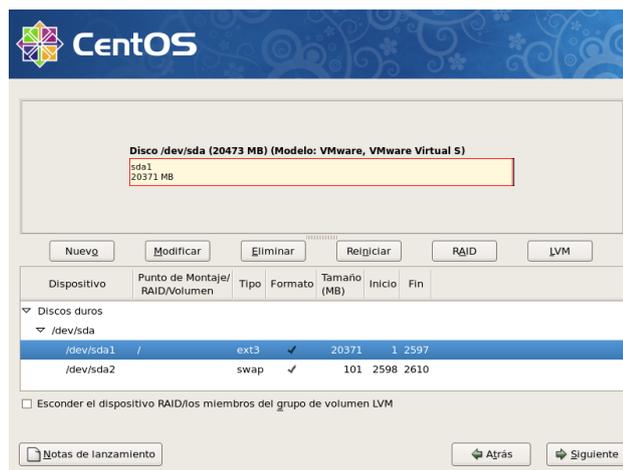
En esta ventana nos indica que tenemos que Particionar el disco duro, damos clic Si.



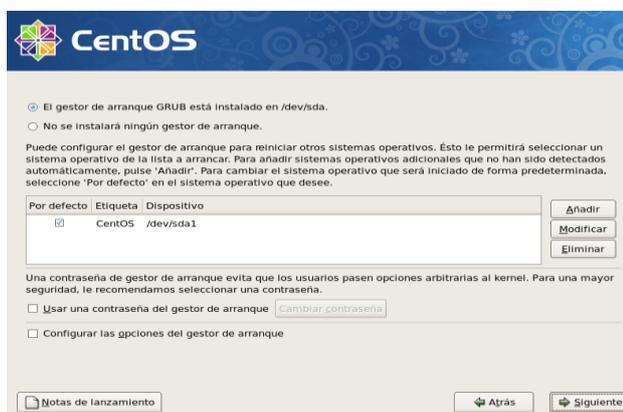
En la siguiente ventana tenemos que escoger la opción crear diseño personalizado y damos clic en Siguiente.



Creamos las particiones necesarias para que CentOS funcione, una de tipo swap y una con extensión ext3 e instalamos CentOS en ext3 y damos clic en Siguiente.



Seleccionamos con que sistema operativo iniciara por defeco el servidor para este caso solo existe CentOS, damos clic en Siguiente.



Configuramos los parámetros de la tarjeta de red y damos clic en Siguiente.



Seleccionamos la zona horaria América Guayaquil y damos clic en Siguiente.



Establecemos la contraseña para el root y damos clic en Siguiente.



Seleccionamos el tipo de entorno grafico y los repositorios para la actualización, para este caso seleccionamos el tipo de entorno grafico Gnome, Clic en Siguiente.



Comienzo de la instalación d CentOS, comprobando las dependencias.



Una vez finalizada la configuración inicia el proceso de instalación, damos clic en Siguiente.



En esta ventana se puede observar el proceso de instalación de CentOS.



Una vez completada la instalación de CentOS, se reiniciara el servidor.



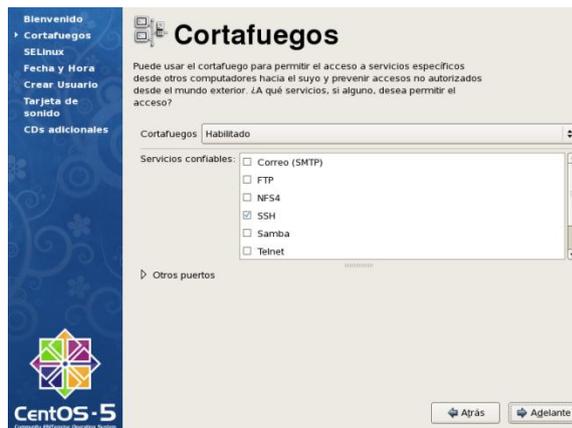
Una vez reiniciada se cargara el Sistema Operativo CentOS, detectando todos los componentes de la maquina servidor.



Se iniciara por primera y única vez un asistente de configuración que terminara de configurar el Sistema Operativo CentOS, damos clic en Adelante.



Primero la configuración del Cortafuegos de CentOS, el cual deberá estar habilitado con los parámetros por defecto. Presionamos en Adelante.



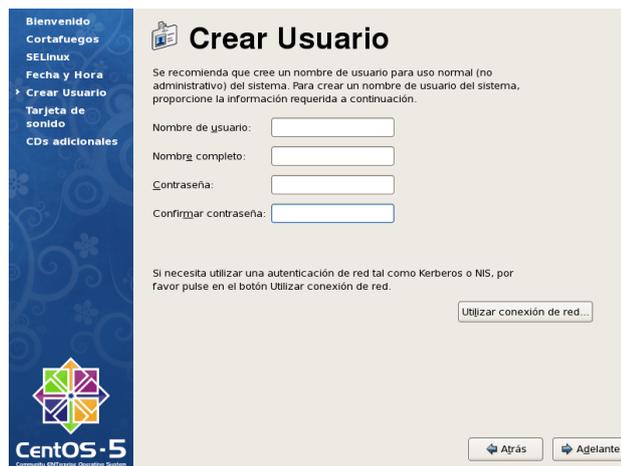
Damos clic en Adelante, dejando deshabilitado SELinux.



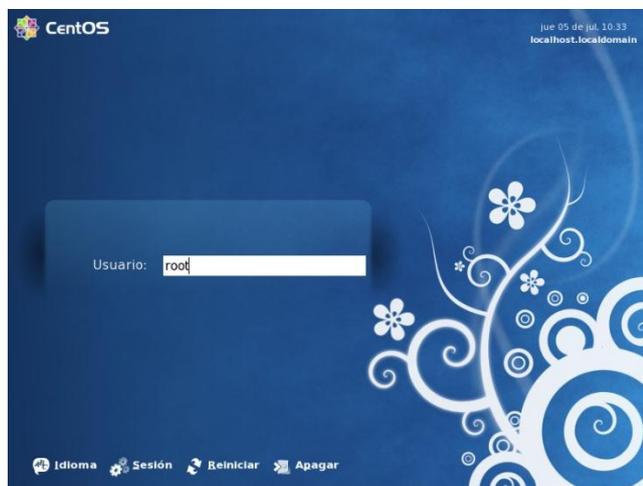
Configuramos la fecha y hora del sistema, damos clic en Adelante.



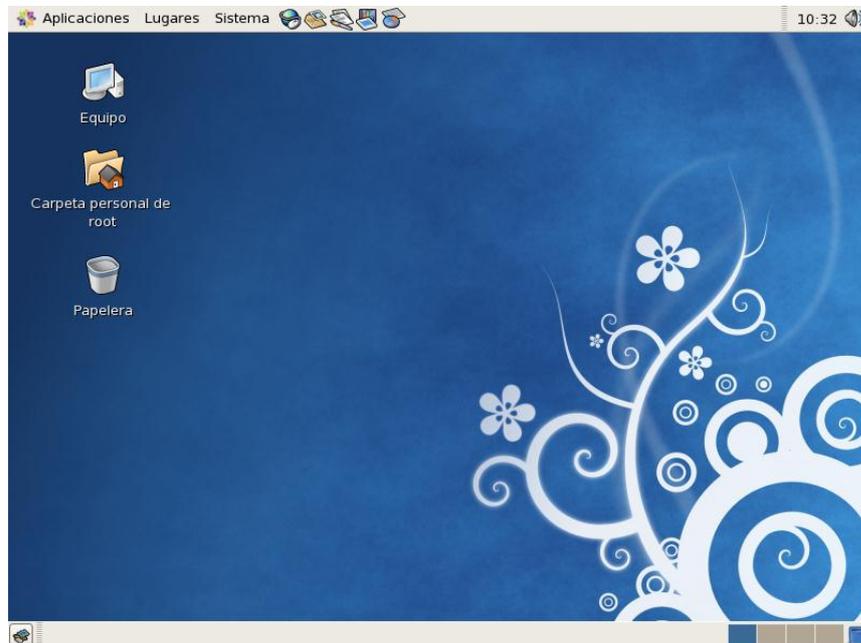
Si deseamos creamos otros usuarios, damos clic en adelante



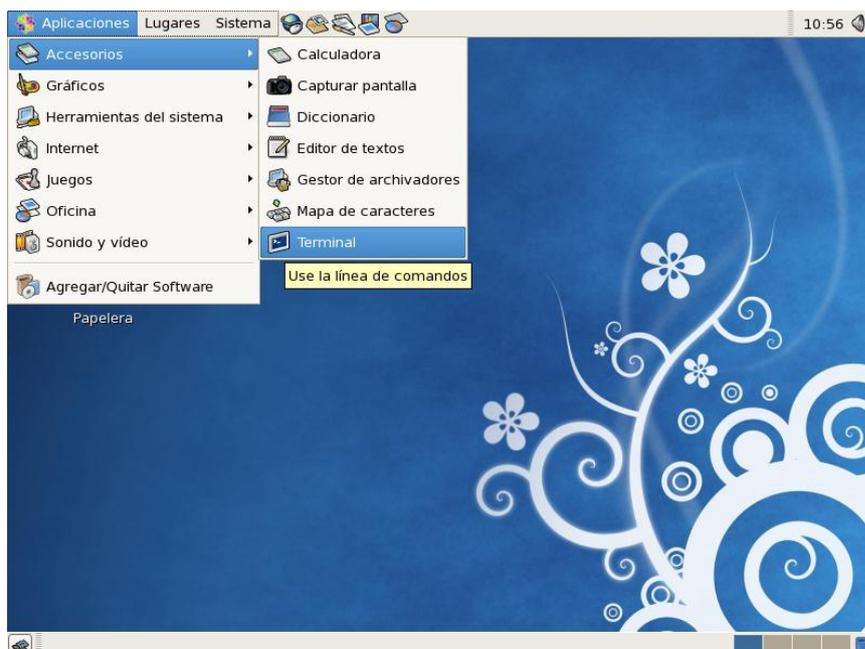
Se presenta la ventana de login e iniciamos sesión como usuario root.



Escritorio Gnome del Sistema operativo CentOS seleccionado en el proceso de instalación, tenemos los accesos al equipo con todas las funcionalidades del usuario root, este es el escritorio principal de CentOS.

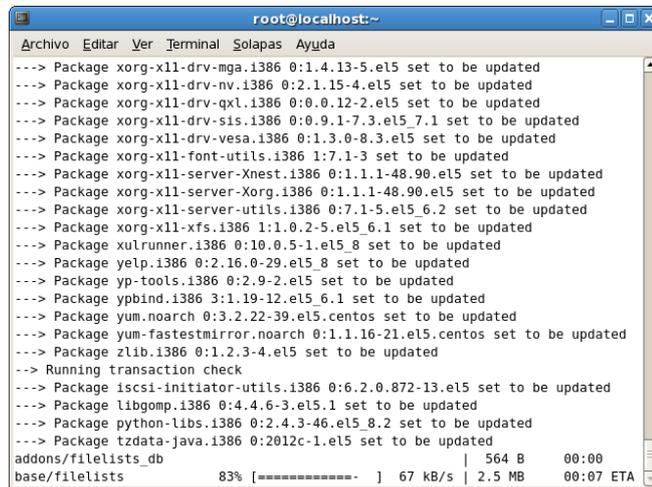


Para proceder a instalar y configurar la aplicación web Proxynet, abrimos una ventana Terminal o de comandos desde el menú Aplicaciones en Accesorios.



Actualización del sistema operativo.

Con la ventana terminal abierta procedemos a actualizamos el sistema operativo CentOS con el comando yum `–y update`.



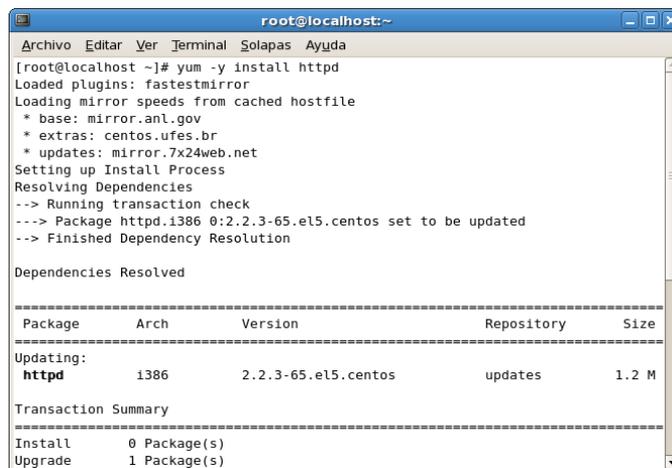
```
root@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda
--> Package xorg-x11-drv-mga.i386 0:1.4.13-5.el5 set to be updated
--> Package xorg-x11-drv-nv.i386 0:2.1.15-4.el5 set to be updated
--> Package xorg-x11-drv-qxl.i386 0:0.0.12-2.el5 set to be updated
--> Package xorg-x11-drv-sis.i386 0:0.9.1-7.3.el5 7.1 set to be updated
--> Package xorg-x11-drv-vesa.i386 0:1.3.0-8.3.el5 set to be updated
--> Package xorg-x11-font-utils.i386 1:7.1-3 set to be updated
--> Package xorg-x11-server-Xnest.i386 0:1.1.1-48.90.el5 set to be updated
--> Package xorg-x11-server-Xorg.i386 0:1.1.1-48.90.el5 set to be updated
--> Package xorg-x11-server-utils.i386 0:7.1-5.el5_6.2 set to be updated
--> Package xorg-x11-xfs.i386 1:1.0.2-5.el5_6.1 set to be updated
--> Package xulrunner.i386 0:10.0.5-1.el5_8 set to be updated
--> Package yelp.i386 0:2.16.0-29.el5_8 set to be updated
--> Package yp-tools.i386 0:2.9-2.el5 set to be updated
--> Package ypbind.i386 3:1.19-12.el5_6.1 set to be updated
--> Package yum.noarch 0:3.2.22-39.el5.centos set to be updated
--> Package yum-fastestmirror.noarch 0:1.1.16-21.el5.centos set to be updated
--> Package zlib.i386 0:1.2.3-4.el5 set to be updated
--> Running transaction check
--> Package iscsi-initiator-utils.i386 0:6.2.0.872-13.el5 set to be updated
--> Package libgomp.i386 0:4.4.6-3.el5.1 set to be updated
--> Package python-libs.i386 0:2.4.3-46.el5_8.2 set to be updated
--> Package tzdata-java.i386 0:2012c-1.el5 set to be updated
addons/filelists_db | 564 B | 00:00
base/filelists      83% [=====] | 67 kB/s | 2.5 MB | 00:07 ETA
```

Instalación de las herramientas utilizadas.

Para que Proxynet Funcione correctamente debemos instalación algunos servicios y realizar algunas configuraciones básicas pero necesarias.

Instalación del Servidor Web Apache.

Instalar el servidor web apache con el comando yum `–y install httpd`.



```
root@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost ~]# yum -y install httpd
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: mirror.anl.gov
* extras: centos.ufes.br
* updates: mirror.7x24web.net
Setting up Install Process
Resolving Dependencies
--> Running transaction check
--> Package httpd.i386 0:2.2.3-65.el5.centos set to be updated
--> Finished Dependency Resolution

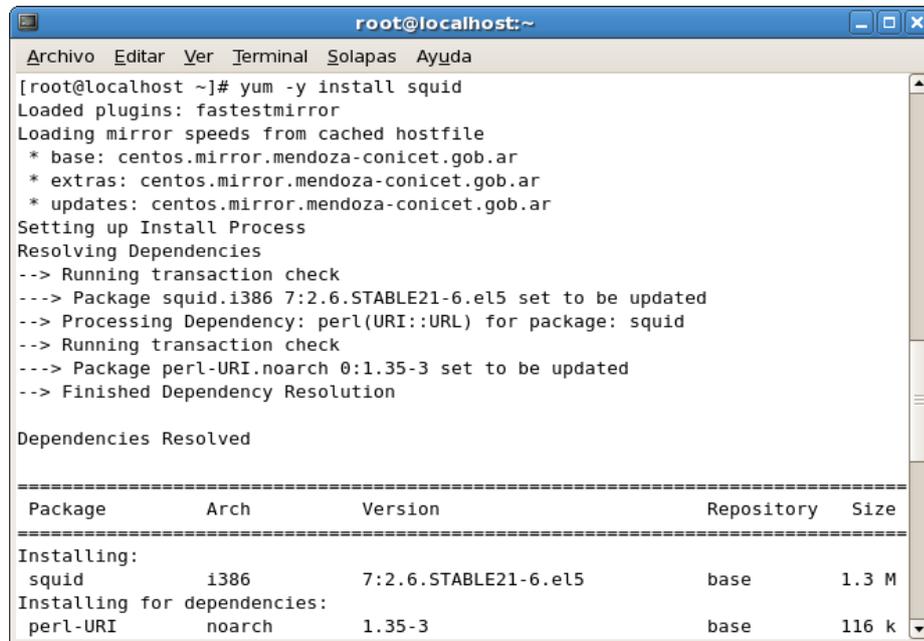
Dependencies Resolved

=====
Package      Arch      Version                Repository      Size
=====
Updating:
httpd        i386      2.2.3-65.el5.centos    updates        1.2 M
=====

Transaction Summary
-----
Install      0 Package(s)
Upgrade     1 Package(s)
```

Instalación del Servidor Proxy Squid.

Instalar el servidor proxy squid con el comando yum `-y install squid`.

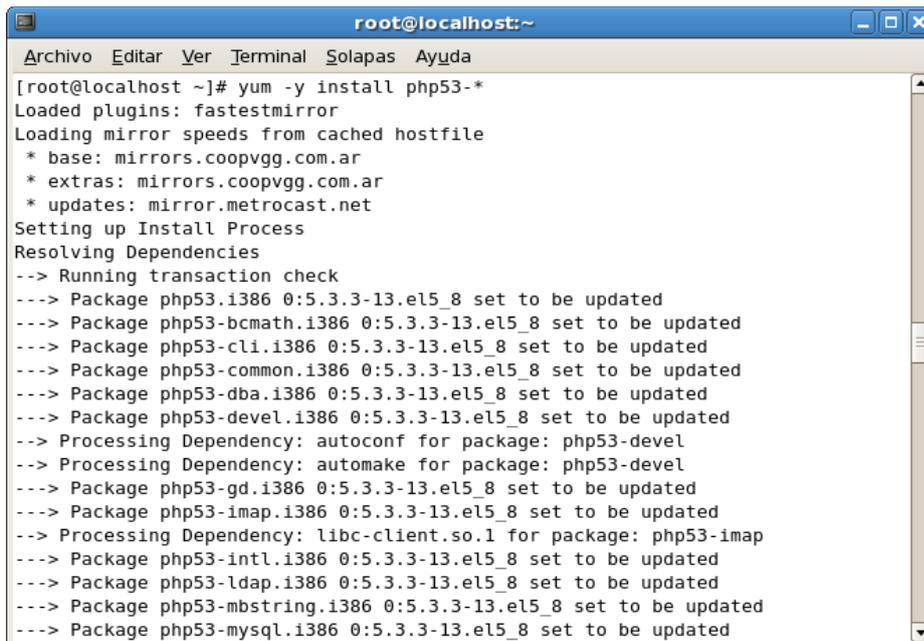


```
root@localhost:~  
Archivo Editar Ver Terminal Solapas Ayuda  
[root@localhost ~]# yum -y install squid  
Loaded plugins: fastestmirror  
Loading mirror speeds from cached hostfile  
* base: centos.mirror.mendoza-conicet.gob.ar  
* extras: centos.mirror.mendoza-conicet.gob.ar  
* updates: centos.mirror.mendoza-conicet.gob.ar  
Setting up Install Process  
Resolving Dependencies  
--> Running transaction check  
--> Package squid.i386 7:2.6.STABLE21-6.el5 set to be updated  
--> Processing Dependency: perl(URL::URL) for package: squid  
--> Running transaction check  
--> Package perl-URI.noarch 0:1.35-3 set to be updated  
--> Finished Dependency Resolution  
  
Dependencies Resolved  
  
=====
```

Package	Arch	Version	Repository	Size
Installing:				
squid	i386	7:2.6.STABLE21-6.el5	base	1.3 M
Installing for dependencies:				
perl-URI	noarch	1.35-3	base	116 k

Instalación del lenguaje de programación PHP.

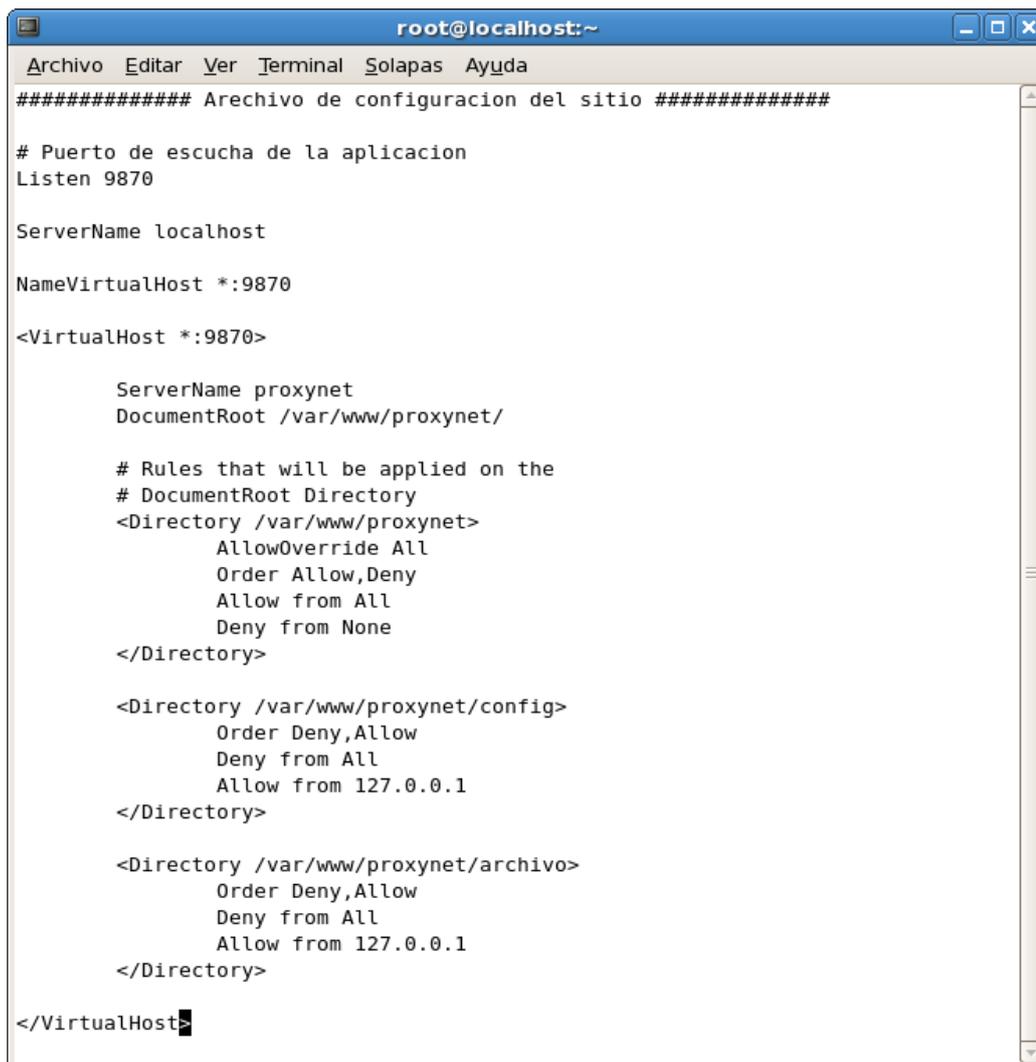
Instalar la versión PHP 5.3 para CentOS con el comando yum `-y install php53-*`



```
root@localhost:~  
Archivo Editar Ver Terminal Solapas Ayuda  
[root@localhost ~]# yum -y install php53-*  
Loaded plugins: fastestmirror  
Loading mirror speeds from cached hostfile  
* base: mirrors.coopvvg.com.ar  
* extras: mirrors.coopvvg.com.ar  
* updates: mirror.metrocast.net  
Setting up Install Process  
Resolving Dependencies  
--> Running transaction check  
--> Package php53.i386 0:5.3.3-13.el5_8 set to be updated  
--> Package php53-bcmath.i386 0:5.3.3-13.el5_8 set to be updated  
--> Package php53-cli.i386 0:5.3.3-13.el5_8 set to be updated  
--> Package php53-common.i386 0:5.3.3-13.el5_8 set to be updated  
--> Package php53-dba.i386 0:5.3.3-13.el5_8 set to be updated  
--> Package php53-devel.i386 0:5.3.3-13.el5_8 set to be updated  
--> Processing Dependency: autoconf for package: php53-devel  
--> Processing Dependency: automake for package: php53-devel  
--> Package php53-gd.i386 0:5.3.3-13.el5_8 set to be updated  
--> Package php53-imap.i386 0:5.3.3-13.el5_8 set to be updated  
--> Processing Dependency: libc-client.so.1 for package: php53-imap  
--> Package php53-intl.i386 0:5.3.3-13.el5_8 set to be updated  
--> Package php53-ldap.i386 0:5.3.3-13.el5_8 set to be updated  
--> Package php53-mbstring.i386 0:5.3.3-13.el5_8 set to be updated  
--> Package php53-mysql.i386 0:5.3.3-13.el5_8 set to be updated
```

Configuraciones necesarias para que la aplicación “Proxynet” funcione.

1. Crear el archivo de configuración del host virtual con el comando vi de la siguiente forma “*vi /etc/httpd/conf.d/proxynet.conf*”, con el fin de administrar el sitio de una manera sencilla y eficiente. El archivo contendrá lo siguiente:



```
root@localhost:~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
##### Arechivo de configuracion del sitio #####

# Puerto de escucha de la aplicacion
Listen 9870

ServerName localhost

NameVirtualHost *:9870

<VirtualHost *:9870>

    ServerName proxynet
    DocumentRoot /var/www/proxynet/

    # Rules that will be applied on the
    # DocumentRoot Directory
    <Directory /var/www/proxynet>
        AllowOverride All
        Order Allow,Deny
        Allow from All
        Deny from None
    </Directory>

    <Directory /var/www/proxynet/config>
        Order Deny,Allow
        Deny from All
        Allow from 127.0.0.1
    </Directory>

    <Directory /var/www/proxynet/archivo>
        Order Deny,Allow
        Deny from All
        Allow from 127.0.0.1
    </Directory>

</VirtualHost>
```

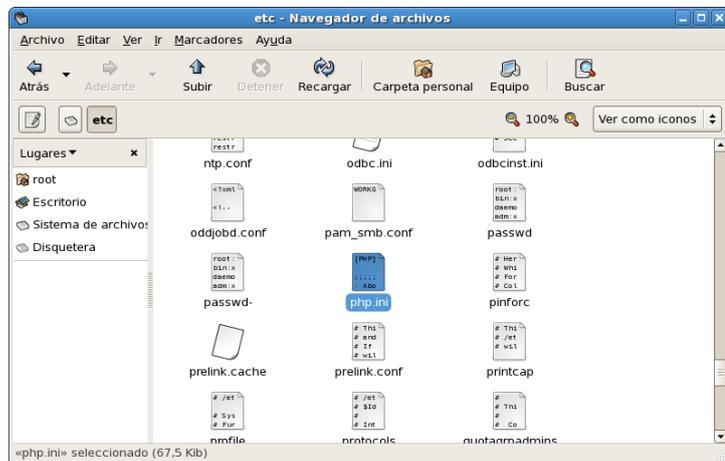
Podemos observar que la aplicación va a escuchar peticiones desde cualquier dirección Ip por el puerto 9870 y que los archivos de la aplicación están alojados en la ruta especificada en DocumentRoot “*/var/www/proxynet*”, a la que le otorgamos ciertos permisos.

2. En la programación de la aplicación Proxynet utilizamos métodos abreviados editamos el archivo php.ini para modificando los siguientes valores:

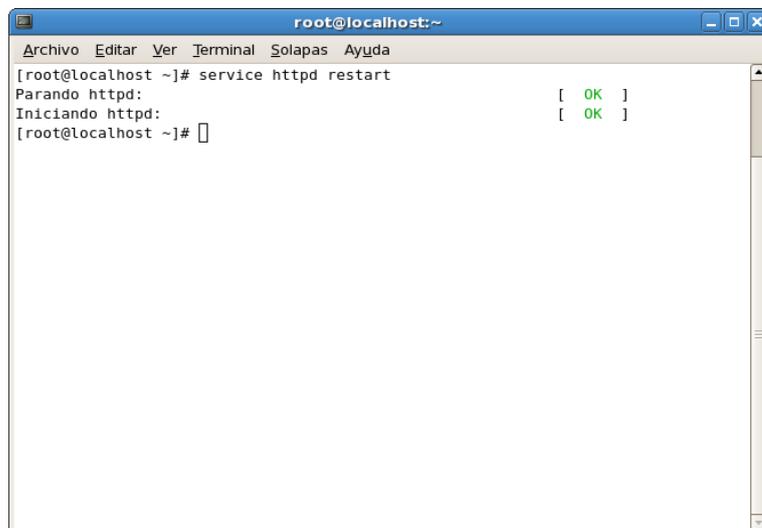
```
short_open_tag = off  
date.timezone = '-'
```

Por estos valores:

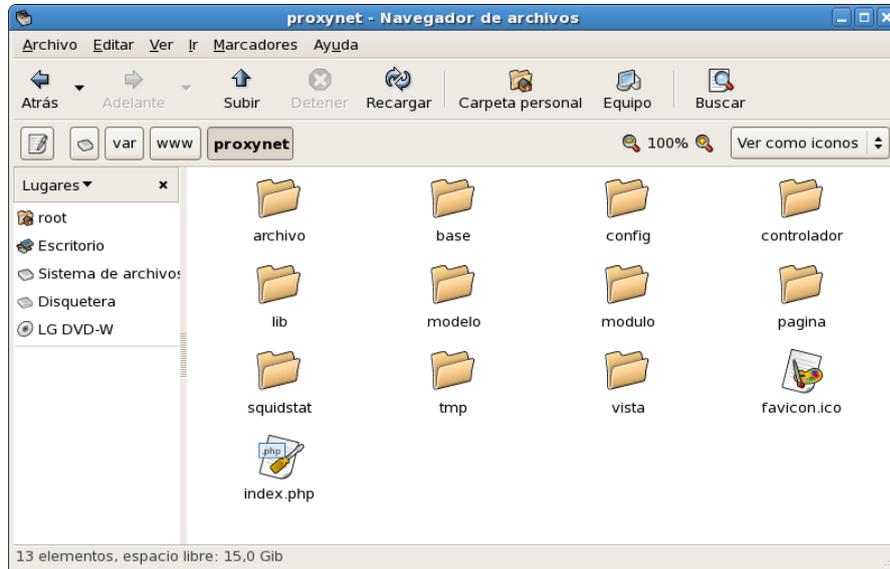
```
short_open_tag = on  
date.timezone = 'America/Guayaquil'
```



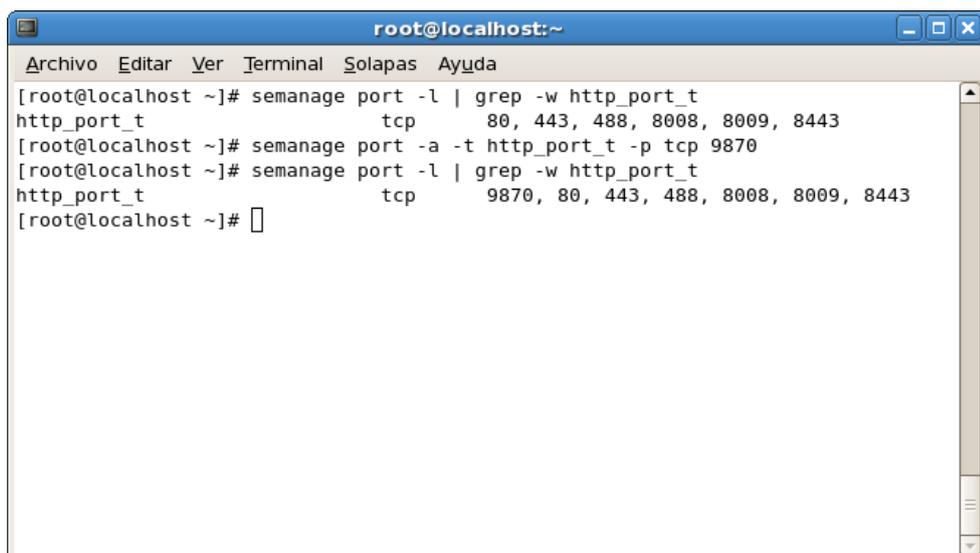
Para que los cambios en el archivo php.ini surtan efectos debemos reiniciar el servidor web apache utilizando el siguiente comando: *service httpd restart*



3. Creamos una carpeta denominada “proxynet” en el directorio del servidor web apache “/var/www/”, y colocamos los archivos de la aplicación.

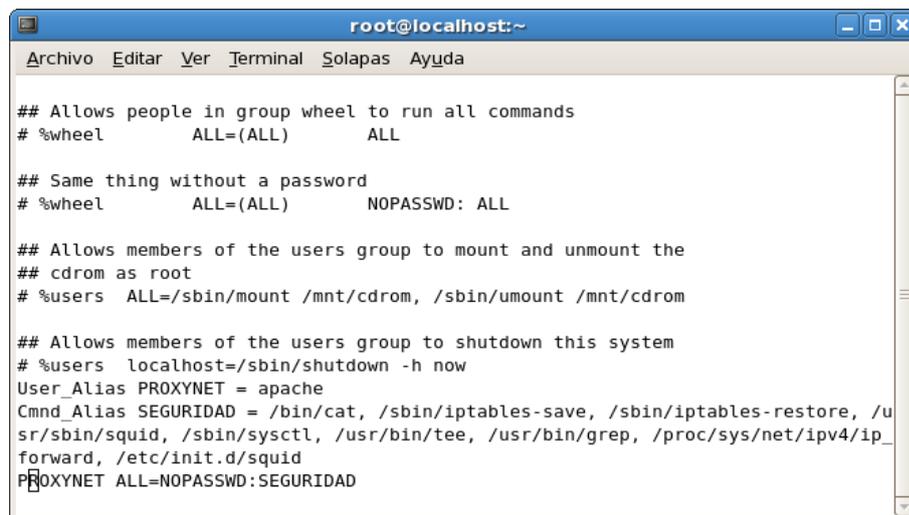


En el archivo de configuración del Servidor de la aplicación colocamos como puerto de escucha el 9870 debemos verificar si este puerto está habilitado para recibir peticiones, para verificar introducimos en el terminal el siguiente comando: “*semanage port -l | grep -w http_port_t*”, en el listado que aparece deberíamos ver los puertos disponibles. Para agregar el puerto 9870 ejecutamos el comando “*semanage port -a -t http_port_t -p tcp 9870*”.



4. Debido a que el usuario root es el único con acceso de ejecutar ciertos comandos necesarios para la aplicación Proxynet creamos un alias del usuario apache otorgándoles permisos para ejecutar dichos comandos. Para ello editamos el archivo “*sudoers*” con el comando “*sudo visudo*”. Agregamos las siguientes líneas:

```
Editar la línea: Defaults    requiretty
Agregar lo siguiente al final:
User_Alias PROXYNET = apache
Cmdnd_Alias SEGURIDAD = /bin/cat, /sbin/iptables-save,
/sbin/iptables-restore, /usr/sbin/squid, /sbin/sysctl, /usr/bin/tee,
/usr/bin/grep, /proc/sys/net/ipv4/ip_forward, /etc/init.d/squid
PROXYNET ALL=NOPASSWD:SEGURIDAD
```



```
root@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda

## Allows people in group wheel to run all commands
# %wheel    ALL=(ALL)    ALL

## Same thing without a password
# %wheel    ALL=(ALL)    NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users    ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users    localhost=/sbin/shutdown -h now
User_Alias PROXYNET = apache
Cmdnd_Alias SEGURIDAD = /bin/cat, /sbin/iptables-save, /sbin/iptables-restore, /u
sr/sbin/squid, /sbin/sysctl, /usr/bin/tee, /usr/bin/grep, /proc/sys/net/ipv4/ip_
forward, /etc/init.d/squid
PROXYNET ALL=NOPASSWD:SEGURIDAD
```

Ejecutamos el comando “*chown*” para otorgarle como propietario de la carpeta y los archivos de la aplicación el usuario apache que al que le otorgamos los permisos en el archivo “*sudoers*”.



```
root@localhost:~
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost ~]# chown -R apache /var/www/proxynet
[root@localhost ~]#
```

4.6.2.6. Seguridades.

Red:

- Contar con un Firewall con el puerto 8080 abierto.
- Servidor proxy propuesto.

Servidor:

- Listas de control de acceso.
- Reglas de control de acceso.
- Encriptación de cuentas de usuarios.
- Control de acceso por IP y por usuario.
- Identificación física de las tarjetas de red.
- Perfiles de control de acceso a usuarios individuales.

4.7. CONCLUSIONES Y RECOMENDACIONES PARA UNA EFICIENTE IMPLEMENTACIÓN DEL SOFTWARE.

4.7.1. CONCLUSIONES.

El creciente uso de los servicios de red, y la presencia cada vez mayor de la Web en los ambientes organizacionales, ha conllevado al empleo herramientas que garanticen el buen uso de los recursos y su disponibilidad para los usuarios. La necesidad de control sobre el tráfico de la red es cada vez mayor ante la importancia que cobra en el mundo actual la seguridad de la información. Los servidores Proxy brindan utilidades para establecer normativas, y obtener esa información tan vital que se traduce en decisiones que llevan al mejoramiento en el uso de recursos y a la aplicación de estrategias de seguridad mucho más adecuadas al ambiente organizacional. Squid es el software libre que brinda a los usuarios todas las ventajas de un servidor Proxy. Al ser un proyecto libre es gratuito, y está siendo mejorado por la comunidad constantemente.

Este proyecto recopila todo el proceso de investigación y posterior construcción de un software que facilita la administración del Servidor Proxy Squid. Mediante la aplicación web desarrollada se contara con una herramienta que reducirá el tiempo de configuración, los riesgos de equivocación o daños por manipulación y la capacidad de respuesta ante eventualidades.

A lo largo del desarrollo del proceso de investigación y construcción se profundizó el conocimiento en la estructura del sistema operativo Linux, el uso de aplicaciones GNU relacionados a Servidores Proxy y el desarrollo de aplicaciones de servicio. Se enfrentaron retos de desarrollo, y se aplicaron soluciones novedosas, sustentadas en la observación y análisis de los procesos y componentes del software Squid Proxy y de la plataforma sobre el que éste se sustenta. Se alcanzaron los objetivos planteados en la investigación. El resultado, una aplicación que incrementa la usabilidad, disminuye el riesgo de error, reduce la brecha entre el administrador,

aumenta la seguridad y actualiza la tecnología de repositorio de datos abriendo un universo de posibilidades en la generación de reportes y la obtención de información en forma eficaz y oportuna. Adicionalmente reduce los costos de implementación y mantenimiento. En conclusión, el resultado es una aplicación que responde a las necesidades de la administración de los recursos de red en los nuevos tiempos.

4.7.2. RECOMENDACIONES.

- Efectuado el análisis de las Políticas de red se constató que no se cuenta con un plan de administración para tráfico de red adecuado. Es conveniente que se tomen las consideraciones necesarias, que permitan obtener el máximo rendimiento a la infraestructura existente en dicho punto de red.
- Es conveniente la actualización de los sistemas encargados de administrar el tráfico de red, ya que la administración del tráfico de red no solo involucra el tener uno o varios servicios activos, sino realizar un constante seguimiento (monitoreo) de dichos servicios que permitan determinar si el rendimiento de la red es el óptimo para las necesidades de los usuarios que hacen uso de ella.
- Para que los docentes y el personal administrativo de la Biblioteca Virtual de la Facultad de Administración, Finanzas e Informática dispongan de una herramienta de comunicación y mensajería interna se recomienda implementar un servidor de correo institucional para facilitar el envío de correo electrónico de manera segura y estable.
- Aprovechando el nuevo rendimiento de la red y las potentes características del servidor se debe tomar en consideración ampliar la topología física de la red, colocar puntos de red en cada aula y distribuir puntos de acceso inalámbricos estratégicamente en la institución, y así mejorar las labores académicas en la institución.

BIBLIOGRAFÍA

- BALESTRINI ACUÑA Miriam.
Cómo se Elabora el Proyecto de Investigación.
BL Consultores Asociados; Sexta Edición. 2002.

- CORONEL CASTILLO Eric Gustavo.
PHP Profesional.
Editorial Macro. 2010.

- DE LA CRUZ VILLAR Joel.
Para Todos LINUX.
Megabyte; Primera Edición. Junio 2008.

- HERNÁNDEZ SAMPIERI Roberto.
Metodología de la investigación.
Mc Graw Hill; Cuarta Edición. 2006.

- KENDALL Kenneth; KENDALL Julie.
Análisis y Diseño de Sistemas de Información.
Mc Graw Hill; Sexta Edición. 2005.

- VASWANI Vikram.
Fundamentos de PHP; Cubre PHP 5.3.
Mc Graw Hill. 2011.

LINKOGRAFÍA

Apache (Servidor web).

1. Documentación del Servidor de HTTP Apache.
[<http://httpd.apache.org/docs/2.4/>]
2. SLaYeR y Lechon. Administración y Gestión de un Servidor Web Apache.
[http://www.odiss.org/manuales/apache/FAQ_Servidor_Apache.pdf]

Iptables.

3. ALTADILL IZURA Pello Xabier.
Iptables Manual Practico.
[<http://www.pello.info/filez/firewall/iptables.html>]
4. BARRIOS DUEÑAS Joel; Alcancelibre.org.
Introducción a IPTABLES.
[<http://www.alcancelibre.org/staticpages/index.php/introduccion-iptables>]
5. Iptables; doc.ubuntu-es
[<http://doc.ubuntu-es.org/Iptables>]
6. Netfilter.org; Firewalling, NAT, and packet mangling for linux.
[<http://www.netfilter.org>]
7. Netfilter/Iptables; Wikipedia, la enciclopedia libre.
[<http://es.wikipedia.org/wiki/Netfilter/iptables>]
8. Netfilter (IpTables); Wikilearning, Comunidad de wikis libres para aprender.
[http://www.wikilearning.com/monografia/netfilter_iptables-que_es_netfilter/9944-1]

9. TORRES FARÍAS Daniela A.

IPTABLES y Squid.

[<http://es.scribd.com/doc/50714632/iptables>]

Proxy (Squid).

10. BARRIOS DUEÑAS Joel; Alcancelibre.org.

Configuración de Squid: Parámetros básicos.

[<http://www.alcancelibre.org/staticpages/index.php/19-0-como-squid-general>]

Configuración de Squid: Acceso por Autenticación.

[<http://www.alcancelibre.org/staticpages/index.php/19-1-como-squid-autenticacion>]

Configuración de Squid: Restricción de acceso a Sitios de Internet

[<http://www.alcancelibre.org/staticpages/index.php/19-2-como-squid-restriccion-web>]

Configuración de Squid: Restricción de acceso a contenido por extensión.

[<http://www.alcancelibre.org/staticpages/index.php/19-3-como-squid-restriccion-extensiones>]

Configuración de Squid: Restricción de acceso por horarios.

[<http://www.alcancelibre.org/staticpages/index.php/19-4-como-squid-tiempo>]

Apéndice: Listas y reglas de control de acceso para Squid.

[<http://www.alcancelibre.org/staticpages/index.php/20-como-squid-reglas>]

Cómo configurar squid con soporte para direcciones MAC.

[<http://www.alcancelibre.org/staticpages/index.php/como-squid-arp>]

11. Ecuallug.org; Ecuador GNU/Linux User Group.

[<http://www.ecuallug.org/>]

12. MORENO Alejandro.

Configuración de Proxy transparente.

[<http://s218907966.mialojamiento.es/doc/blogspot/20080128/200801282157.pdf>]

13. PROXY Squid; Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado.

[http://www.ite.educacion.es/formacion/materiales/85/cd/REDES_LINUX/enrutamiento/Proxy_squid.html]

14. The Squid Guide; Squid User's guide site!.

[<http://www.deckle.co.za/squid-users-guide/index.html>]

15. Squid-cache.org; Optimising Web Delivery.

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

16. Squid (programa) ; Wikipedia, la enciclopedia libre.

[[http://es.wikipedia.org/wiki/Squid_\(programa\)](http://es.wikipedia.org/wiki/Squid_(programa))]

PHP.

17. Manual de PHP.

[<http://www.php.net/manual/es/index.php>]

Qos (Delay Pools).

18. BALLIACHE Leonardo.

SERVICIOS QoS.

[<http://opalsoft.net/qos/Spanish-QOS.htm>]

19. Calidad de servicio; Wikipedia la enciclopedia libre.
[http://es.wikipedia.org/wiki/Calidad_de_servicio]
20. Squid - Limitar ancho de banda.
[<http://linuxsix.blogspot.com/2009/01/squid-limitar-ancho-de-banda.html>]
21. TAN NOZAWA Jaime M.
Control de Ancho de banda con Squid: Delay Pools.
[<http://www.alcancelibre.org/staticpages/index.php/Squid-delay-pools>]
22. MARTIN MALDONADO Daniel.
Implementando QoS en las Redes.
[<http://www.aplicacionesempresariales.com/implementando-qos-en-las-redes.html>]

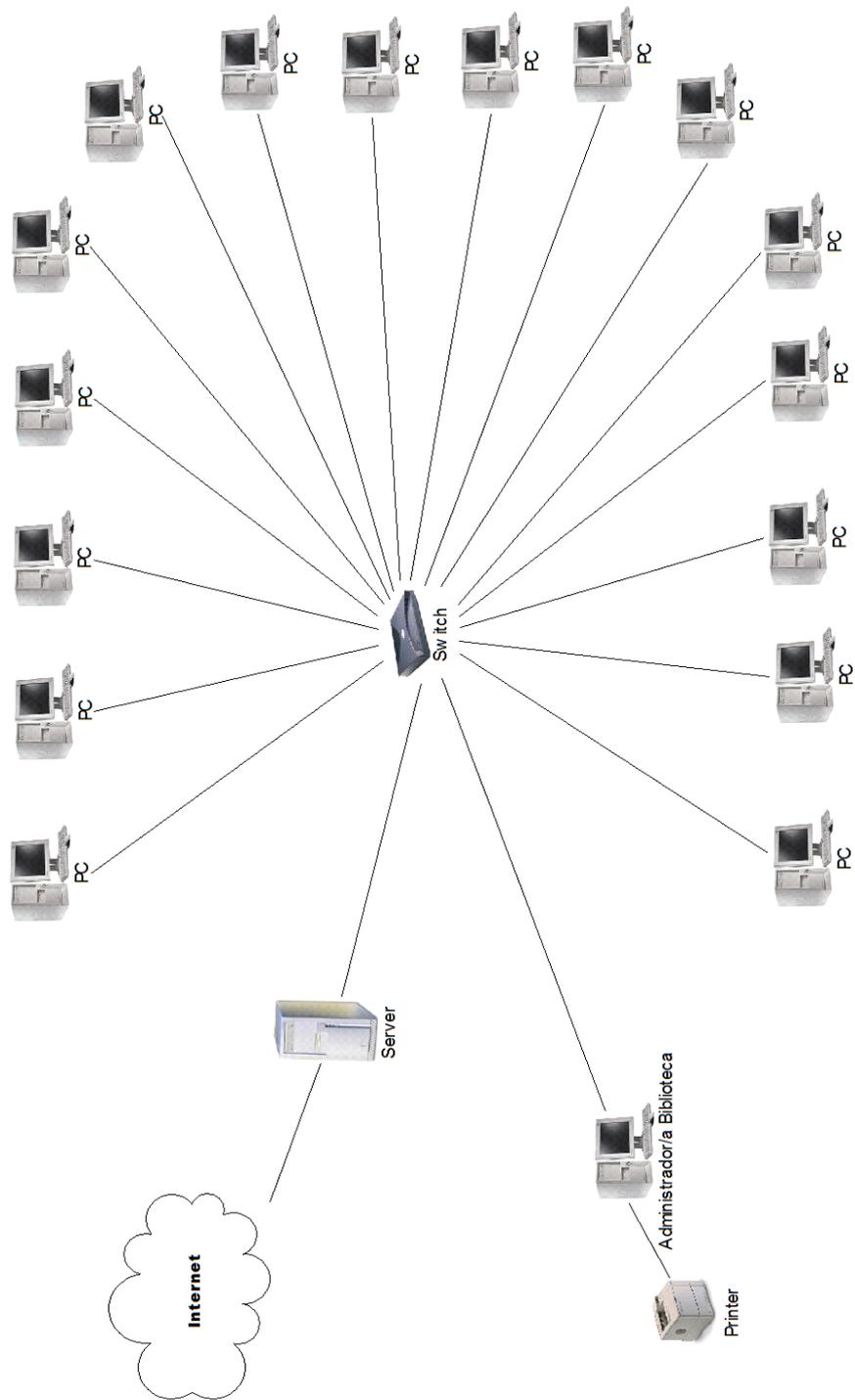
Varios.

23. [http://www.aprendaredes.com/downloads/Como_Administrar_Red.es.pdf]
24. [http://www.ecualug.org/2011/05/29/blog/iknaxio/squid_un_error_com%C3%BA_n_al_combinar_acls]
25. [<http://www.forosdelweb.com/f20/squid-router-887058/>]
26. [<http://dns.bdat.net/documentos/squid/x30.html>]
27. [<http://recursostic.educacion.es/observatorio/web/es/equipamiento-tecnologico/redes/694-administrar-la-red-en-un-ies>]
28. [<http://www.peruhardware.net/foros/showthread.php?t=132986>]

ANEXOS

Anexo A

Infraestructura de Red de la Biblioteca Virtual de la Facultad de Administración, Finanzas e Informática



Anexo B

“Proxynet” Manual de Usuario

