



UNIVERSIDAD JOSÉ ANTONIO PÁEZ

DESARROLLO DE UNA APLICACIÓN
WEB PARA LA ADMINISTRACIÓN DE
SERVIDORES PROXY EN SISTEMAS
OPERATIVOS GNU/LINUX. CASO:
SQUID PROXY

Montesino, Pedro
C.I. 17.594.295
Morales, José
C.I. 14.465.814

Urb. Yuma II, calle N° 3. Municipio San Diego
Teléfono: (0241) 8714240 (master) – Fax: (0241) 8712394

REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN
CARRERA INGENIERIA EN COMPUTACIÓN

**DESARROLLO DE UNA APLICACIÓN WEB PARA LA
ADMINISTRACIÓN DE SERVIDORES PROXY EN SISTEMAS
OPERATIVOS GNU/LINUX. CASO: SQUID PROXY**

Trabajo de Grado para optar al título de Ingeniero de Computación

Autor(a): Pedro Javier Montesino Delgado
C.I. 17.594.295
José Francisco Morales Luces
C.I. 14.465.814

Tutor(a): Ing. Ifigenia Requena
Ing. Alicia de Pizzella

San Diego, octubre de 2010

Carta de aceptación de aprobación del proyecto

REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN COMPUTACION

ACEPTACIÓN DEL TUTOR

Quien suscribe, **Ing. Ifigenia A. E. Requena N**, portador(a) de la cédula de identidad N° **10.543.541**, en mi carácter de tutor del trabajo de grado presentado por los ciudadanos: **Pedro Javier Montesino Delgado**, portador de la cédula de identidad N° **17.594.295** y **José Francisco Morales Luces**, portador de la cédula de identidad N° **14.465.814** titulado: **DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE SERVIDORES PROXY EN SISTEMAS OPERATIVOS GNU/LINUX. CASO: SQUID PROXY**, presentado como requisito parcial para optar al título de **Ingeniero de Computación**, considero que dicho trabajo de grado reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del jurado examinador que se designe.

En San Diego, a los trece (13) días del mes de diciembre del año dos mil diez (2010).

Ing. Ifigenia A. E. Requena N.

C.I.: 10.543.541

DEDICATORIA

A nuestros padres quienes han estado presentes en cada etapa de nuestras vidas, apoyándonos a lo largo del camino, haciendo suyas nuestras metas, convirtiendo nuestras dificultades en propias.

A Dios, sin él lo imposible no sería posible. Gloria y honor a quien merece lo mejor, al mismo que cuelga la tierra sobre la nada y quien nos da el aliento para vivir cada día y es la fuente de la vida eterna, Jesús Rey de Reyes a quien debemos la vida.

A nuestros amigos por sus palabras de ánimo.

A Alisabel por siempre estar a mi lado. José Morales.

A nuestros profesores gracias y que Dios les bendiga por su valiosa ayuda.

RECONOCIMIENTO

A nuestros profesores por su invaluable colaboración en nuestra formación.

A la profesora Ing. Ifigenia Requena Negrón por su continua colaboración en el desarrollo de la investigación, su apoyo incondicional y por su profesionalismo más allá del deber.

A la profesora Ing. Alicia De Pizella por su colaboración en el desarrollo del presente trabajo.

ÍNDICE GENERAL

CONTENIDO	p.p.
LISTA DE CUADROS	ix
LISTA DE FIGURAS	xi
RESUMEN INFORMATIVO	xiii
INTRODUCCIÓN	1
CAPÍTULO	
I EL PROBLEMA	
1.1. Planteamiento del Problema.....	4
1.2. Formulación del Problema.....	5
1.3. Objetivos de la Investigación.....	5
1.3.1. Objetivo General.....	5
1.3.2. Objetivos Específicos.....	5
1.4. Justificación.....	6
1.5. Alcance de la Investigación.....	6
II MARCO TEÓRICO	
2.1. Antecedentes.....	7
2.2. Bases Teóricas.....	9
2.2.1. Squid.....	12
2.2.2. Web Cache.....	13
2.2.3. Linux.....	14
2.2.4. La Licencia Publica General (GNU GPL).....	14
2.2.5. Debian GNU/LINUX.....	16
2.3 Definición de Términos Básicos.....	18
III MARCO METODOLÓGICO	
3.1. Tipo de Investigación.....	22
3.2. Diseño.....	22
3.3. Nivel de la Investigación.....	23
3.4. Fases Metodológicas.....	23
IV RESULTADOS	
4.1 Fase I: Alcance, Requerimientos Básicos y Factores De Riesgo.....	26
4.2 Fase II: Análisis de Requerimientos.....	33
4.3 Fase III: Construcción de Interfaz y Módulos.....	66
4.4 Fase IV: Evaluación de los Resultados del Sistema Propuesto.....	83

V CONCLUSIONES Y RECOMENDACIONES	
5.1 CONCLUSIONES.....	89
5.2 RECOMENDACIONES.....	90
REFERENCIAS.....	91
ANEXOS	
ANEXO A.....	92

LISTA DE CUADROS

CUADRO	CONTENIDO	p.p.
1	Plan de Desarrollo del producto.....	27
2	Cuadro de Análisis de la Aplicación Webmin.....	28
3	Cuadro de Análisis de la Aplicación Sarg.....	29
4	Cuadro de Análisis de la Aplicación Pysquila.....	29
5	Cuadro de Riesgos.....	30
6	Modelo del Dominio.....	31
7	Diagrama Inicial de la Arquitectura	33
8	Diagrama de Secuencia de Configuración.....	60
9	Diagrama de Secuencia de Servicio.....	60
10	Diagrama de Secuencia de Servicio Auxiliar Archivos Log...	61
11	Diagrama de Secuencia de Servicio Auxiliar Archivo de Configuración.....	61
12	Diagrama de Secuencia de Configuración.....	62
13	Diagrama de Colaboración de Servicio.....	63
14	Diagrama de Colaboración Archivo de Configuración.....	64
15	Diagrama de Colaboración Servicio Auxiliar Archivos Log.....	64
16	Diagrama de Colaboración Administrador Archivo de Configuración.....	65
17	Diagrama de Arquitectura Final.....	82
18	Diagrama de Despliegue.....	83
19	Casos de Prueba Caja Negra N° 1.....	84
20	Casos de Prueba Caja Negra N° 2.....	85
21	Casos de Prueba Caja Negra N° 3.....	85

22	Casos de Prueba Caja Negra N° 4.....	86
23	Casos de Prueba Caja Negra N° 5.....	86
25	Casos de Prueba Caja Blanca N° 1.....	87
25	Casos de Prueba Caja Blanca N° 2.....	87
26	Casos de Prueba Caja Blanca N° 3.....	88
27	Casos de Prueba Caja Blanca N° 4.....	88
28	Casos de Prueba Caja Blanca N° 5.....	89

LISTA DE FIGURAS

FIGURA	CONTENIDO	p.p.
1	Caso de Uso Inicial	32
2	Diagrama de Actividad de Configuración.....	34
3	Diagrama de Actividad del Servicio.....	35
4	Modelo de Caso de Uso Administrador.....	36
5	Modelo de Caso de Uso del Servicio Auxiliar.....	37
6	Especificación Caso de Uso Detener Servicio SQUID Proxy.....	38
7	Especificación Caso de Uso Reiniciar Servicio SQUID Proxy.....	39
8	Especificación Caso de Uso Recargar Servicio SQUID Proxy.....	40
9	Especificación Caso de Uso Borrar Cache SQUID Proxy.....	41
10	Especificación Caso de Uso Generar Reportes.....	42
11	Especificación Caso de Uso Visualizar el archivo de configuración.....	43
12	Especificación Caso de Uso Configurar Parámetros de Memoria Secundaria.....	44
13	Especificación Caso de Uso Configurar Parámetros de Red.....	45
14	Especificación Caso de Uso Crear Listas de Control de Acceso.....	46
15	Especificación Caso de Uso Definir Usuarios Del Sistema.....	47
16	Especificación Caso de Uso Definir Datos de la Organización.....	48
17	Especificación Caso de Uso Definir Directivas, Parámetros y Categorías.	49
18	Especificación Caso de Uso Iniciar Servicio SQUID Proxy.	50
19	Especificación Caso de Uso Generar Archivo de Configuración por Defecto.	51

20	Especificación Caso de Uso Consultar Parámetros de la Base de Datos	52
21	Especificación Caso de Uso Reemplazar Archivo de Configuración	53
22	Especificación Caso de Uso Almacenar Contenido de Archivos Log	54
23	Especificación Caso de Uso Iniciar Servicio SQUID Proxy	55
24	Especificación Caso de Uso Recargar Servicio SQUID Proxy.....	56
25	Especificación Caso de Uso Reiniciar Servicio SQUID Proxy	57
26	Especificación Caso de Uso Detener Servicio SQUID Proxy	58
27	Especificación Caso de Uso Borrar Cache del SQUID Proxy	59
28	Modelo de Datos	65
29	Pantalla Autenticación.	66
30	Pantalla de Bienvenida	67
31	Pantalla de Servicios	67
32	Selección de Configuración.....	68
33	Pantalla de Configuración Avanzada.....	68
34	Configuración de Puertos.....	69
35	Parámetros de Cache.....	69
36	Configuración de Directorios Log.....	70
37	Lista de Grupos.....	70
38	Reglas de Lista de Acceso.....	71
39	Lista de Control de Acceso.....	71
40	Pantalla Visualización Archivo de Configuración.....	72
41	Pantalla de Monitoreo.....	72
42	Sitio Más Visitado por clientes.....	73
43	Sitios Más Visitados.....	73
44	Sitios No Permitidos.....	74
45	Administración del Sistema.....	74

46	Pantalla Configuración General.....	75
47	Datos de Ejecución.....	75
48	Pantalla de Rutas Generales.....	76
49	Menú Datos de la Organización.....	76
50	Datos de la Organización.....	77
51	Creación de Departamentos.....	77
52	Creación de Personal.....	78
53	Creación de Usuarios del Sistema.....	78
54	Tablas Auxiliares.....	79
55	Creación de Categorías.....	79
56	Creación de Directivas.....	80
57	Creación de Parámetros.....	80

REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN
CARRERA INGENIERIA EN COMPUTACIÓN

**DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN
DE SERVIDORES PROXY EN SISTEMAS OPERATIVOS GNU/LINUX.
CASO: SQUID PROXY**

Autores: Pedro Javier Montesino Delgado

José Francisco Morales Luces

Tutoras: Ing. Ifigenia Requena

Ing. Alicia de Pizzella

Fecha: diciembre de 2010

RESUMEN INFORMATIVO

La Internet se ha convertido en una necesidad dentro del ámbito empresarial, lo que ha originado que los departamentos de Tecnología de Información (TI) recurran a herramientas que respalden las políticas de seguridad de sus organizaciones. Dentro de esas herramientas, se encuentra a Squid Proxy, un software libre que ha adquirido gran importancia a nivel mundial siendo indispensable para todo departamento de TI. Sin embargo, la carencia de una interfaz de configuración amigable, hace que el Squid pierda valor frente al software privativo equivalente. Ante esta situación, se desarrolló una herramienta bajo software libre que permite la configuración y administración básica de Squid Proxy. Este trabajo se enmarca dentro de la modalidad de proyecto factible apoyado en una investigación documental y es de tipo descriptivo. El desarrollo del sistema fue realizado bajo la metodología Rational Unified Process (RUP). El resultado, una aplicación que responde a las actuales necesidades de la administración de los recursos de red.

Descriptores: Proxy, GNU, Linux, RUP, Servidor, Software Libre

INTRODUCCIÓN

El acelerado desarrollo de nuevas tecnologías ha servido de plataforma para la aparición de nuevas formas de comunicación, Internet ha sido y sigue siendo una autopista sin fronteras por donde usuarios, sin importar su ubicación geográfica, pueden tener acceso a una ilimitada fuente de información. Es incontable la gran cantidad de operaciones que viajan a través de la gran telaraña mundial y las ventajas que estas han traído consigo, las empresas han crecido y han estrechado sus lazos comerciales a través de este medio, sin hablar de la automatización de tareas que suprimen la presencia física de un individuo.

Pero a medida que crecen las comunicaciones en internet cobra mayor importancia para toda organización que las transacciones sean precisas y seguras tanto dentro como fuera de sus redes. El personal de tecnología de la información, tiene la gran responsabilidad de velar por el buen uso tanto del ancho de banda, como del tránsito que circula en la red, pero en muchas ocasiones se escapan detalles referentes al verdadero uso por parte de los usuarios de Internet. Las aplicaciones de mensajería instantánea se han convertido en programas de uso masivo, al igual que las redes sociales, y los innumerables sitios de ocio disponible en la red de redes. Sin embargo, esto no se considera un problema, hasta el momento en que este comportamiento es emulado dentro de las organizaciones, donde el uso incorrecto del internet como recurso se convierte en un mal que atenta contra los objetivos y metas definidas por las empresas u organizaciones.

Proporcionar mecanismos seguros que restrinjan este tipo de accesos y limiten el uso de Internet a solo material pertinente, que vaya orientado al cumplimiento de los objetivos de la empresa puede ser una tarea costosa. Sin embargo, existen herramientas en el mercado que permiten administrar los recursos de la red de forma organizada y centralizada. Algunas de estas herramientas son de software propietario, y otras tantas son software libre. La diferencia entre ellas, es el costo de la licencia, que en el caso del software

propietario tiende a ser elevado, en contraposición a las de software libre donde no existe costo por licenciamiento del producto.

Dentro de las herramientas que permiten la administración y el control del recurso red, encontramos a Squid Proxy, que es una aplicación que tiene como objetivo fungir como mediador entre un cliente y un servidor, recibiendo peticiones web de un usuario, procesándolas y brindándole respuestas a sus solicitudes. Esto permite que se aligere el tráfico en la red, y disminuya la carga de trabajo de los servidores destino, aumentando la velocidad de respuesta. El servidor Proxy crea una caché que evita transferencias idénticas de la información entre servidores durante un tiempo preestablecido por el usuario administrador. Adicionalmente el Squid Proxy permite definir reglas para restringir el contenido disponible para los usuarios de la red interna, por tanto es posible disminuir el uso indebido de los recursos, aumentando el ancho de banda disponible para transacciones permitidas y disminuyendo los elementos distractores para el recurso humano.

El Squid Proxy es una herramienta de gran importancia dentro de las organizaciones, y su uso extensivo a nivel mundial lo ha convertido en una herramienta indispensable de todo departamento de tecnología de la información, sin embargo, este software carece de una interfaz de control, administración y monitoreo, que se ajuste a las necesidades propias de usuarios cada vez más exigentes. Algunas herramientas como el SARG (Squid Analysis Report Generator), permiten obtener información detallada acerca del uso que hacen los usuarios de la red, sin embargo carece de herramientas para la configuración del Proxy Squid, y su interfaz grafica no es la más idónea para la presentación de datos.

La carencia de un mecanismo de configuración amigable, y la falta de un generador de reportes adecuado, hace que el Squid pierda valor frente al software privativo equivalente. Ante esta situación, se propone el desarrollo de una herramienta bajo software libre que permita la configuración, administración y el control estadístico de la herramienta Squid Proxy, con una interfaz amigable a usuarios de Tecnología de la Información, disponible a la hora de configurar y

administrar la herramienta y proporcionando de manera inmediata estadísticas graficas que revelen el integro uso de las demandas Web, orientado a todo tipo de empresas u organizaciones.

El presente trabajo de grado está estructurado en capítulos.

En el Capítulo I se introduce al lector en el contexto del tema de investigación, describiendo el problema, los objetivos que se desean alcanzar con el desarrollo del proyecto, el alcance y las limitaciones del mismo.

En el Capítulo II se exponen los fundamentos teóricos del contexto de la investigación, se introduce al lector en las definiciones pertinentes al tema, como Squid Proxy, Web Cache, Linux, GNU, Debian, y otros conceptos importantes dentro del marco de desarrollo del proyecto.

En el Capítulo III se describe la metodología utilizada para el desarrollo del sistema, así como sus fases y los resultados esperados en cada una de ellas.

En el Capítulo IV, se describen los resultados de la aplicación de la metodología Rational Unified Process (RUP), y se muestran los objetos y artefactos obtenidos en cada una de las fases.

CAPÍTULO I

EL PROBLEMA

1.1. Planteamiento del Problema

El uso intensivo de Internet como fuente de información es una necesidad que se hace presente cada vez más en cada una de las áreas del ámbito empresarial. Esto ha originado que los departamentos de tecnología de información establezcan políticas de seguridad y de uso, e implementen herramientas orientadas a optimizar el uso de los recursos tecnológicos presentes en las organizaciones. Una de estas herramientas es el servidor proxy cuya función es recopilar las peticiones de los usuarios de la red interna de la organización y transmitirlos a la Internet, logrando con esto tener control sobre las páginas visitadas y los recursos solicitados. El servidor proxy también colabora con el incremento de la velocidad de respuesta a las solicitudes mediante el almacenamiento de los recursos más solicitados en forma local y permite establecer filtros de contenido, así como proteger la privacidad de los usuarios del servicio.

Actualmente existen varias herramientas disponibles en el mercado que permiten la administración y configuración de servidores proxies. En plataformas basadas en Microsoft Windows usualmente se emplea Microsoft Forefront. En Sistemas Operativos Linux se emplea por lo general Squid, que también funciona bajo Sistemas Operativos Windows. A diferencia de Microsoft Forefront, Squid es de distribución libre bajo licencia GPL. El hecho de ser gratuito hace que Squid sea una opción muy atractiva para los departamentos de tecnología de la información, y por tanto se ha convertido en una de las herramientas más utilizadas en el entorno empresarial.

Squid no cuenta con una interfaz gráfica que permita administrar y configurar las distintas opciones de las que dispone, lo que hace mucho más difícil su mantenimiento. Aunado a esto, no es posible visualizar las estadísticas de forma adecuada, debido a que la información es almacenada en archivos de texto plano en forma de registros no organizados. Existen algunas herramientas en el mercado que complementan al Squid, como es el caso de Squid Analysis Report Generator, pero muchos de ellos sólo permiten obtener información limitada y difícil de emplear para la toma de decisiones en cuanto a la asignación de recursos, la optimización del sistema y los costos de operación.

La carencia de una interfaz gráfica y base de datos hace que el Squid represente un costo de horas hombre elevado para su configuración y mantenimiento y disminuye la capacidad de obtener información para la toma de decisiones.

1.2. Formulación del Problema

¿Cómo mejorar la administración y configuración del Servidor proxy Squid mediante una aplicación de software libre que facilite la interacción y la gestión de la información?

1.3. Objetivos de la Investigación

1.3.1. Objetivo General

Desarrollar una aplicación web para la administración de servidores proxy en sistemas operativos GNU/Linux. Caso: Squid Proxy.

1.3.2. Objetivos específicos

- Determinar el alcance, requerimientos básicos y factores de riesgo del sistema propuesto.
- Analizar los requerimientos detallados de los procedimientos propuestos.
- Construir los módulos e interfaces de la aplicación basado en los diseños propuestos.
- Evaluar los resultados obtenidos del sistema propuesto.

1.4. Justificación

En el campo organizacional, obtener información detallada y ordenada del uso del Internet es de suma importancia para la toma de decisiones y la buena administración de los recursos tecnológicos, es por ello que la implementación de herramientas de control como Squid Proxy resultan necesarias.

El desarrollo de una aplicación que permita integrar procesos referentes a la configuración, administración y control estadístico de la herramienta Squid Proxy resulta de gran importancia para las organizaciones ya que proporciona información que facilita la toma de decisiones referentes al uso de recursos tecnológicos y la seguridad de la información.

La herramienta de administración y configuración de servidor proxy disminuirá el número de horas hombres empleados para la implementación y mantenimiento del servidor proxy lo que reducirá costos y optimizara la ocupación del personal.

Desde una perspectiva más general se plantea, entonces, el reto de desarrollar una aplicación que permita administrar y configurar de forma eficiente los recursos que brinda el Squid, mejorando así los reportes generados por el mismo, haciendo más fácil la toma de decisiones y disminuyendo el costo en horas hombres.

1.5. Alcance de la Investigación

El sistema se desarrollará bajo código abierto y estará dirigido al personal de tecnología de la información de habla hispana de organizaciones empresariales y educativas. Brindará una interfaz para la configuración y administración básica del proxy Squid: inicio, finalización y reinicio del servicio Squid, definición de las listas de usuarios para el control de acceso, creación y modificación de las reglas de control de acceso, definición de los parámetros de la memoria cache y de los puertos empleados para establecer las conexiones. También contará con una interfaz que mostrará algunas estadísticas: páginas visitadas por usuarios, la página más visitada, horas del día con más consumo de ancho de banda y cantidad de peticiones denegadas.

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes

Bitar J. (2004) en su trabajo de grado **“Implementación de una herramienta basada en software libre para la generación estandarizada de reportes no estructurados en línea”** desarrollado en la Universidad José Antonio Páez, propone la implementación de una herramienta basado en software libre para la generación de informes de diversas áreas de RedUC (Unidad de Redes Telemáticas de la Universidad de Carabobo), como complemento de un sistema ya existente, que carecía de un componente capaz de generar reportes a partir de la información almacenada. La falta del componente de reportes disminuía en gran medida la funcionalidad del sistema.

El objetivo de este trabajo fue implementar una herramienta basada en software libre para la generación estandarizada de reportes no estructurados en línea que facilitara el uso y ordenamiento de los datos almacenados y administrados a través del software de administración de RedUC.

Este trabajo muestra la ventaja que supone realizar integraciones con sistemas preexistentes con el objetivo de mejorar sus cualidades y en particular obtener mayor información de los mismos mediante el tratamiento de los datos almacenados.

Así mismo, Rojas C. (2008) en su estudio **“Propuesta de esquema de implantación de una herramienta de monitoreo y seguridad para la administración de servidores”** desarrollado en la Universidad José Antonio Páez, propone un esquema de implantación de una herramienta de monitoreo y seguridad de redes, que soporte las plataformas ISeries, Windows y Linux, para la administración de las operaciones de la plataforma de TI (Tecnología de la Información) permitiendo el control de la seguridad de las mismas desde una única consola utilizando una metodología de comparación y administración de software.

Este estudio resalta la importancia del monitoreo de recursos de forma centralizada, y su impacto positivo sobre el rendimiento y los resultados en los departamentos de Tecnología de Información, así como también en los costos asociados a la administración de las redes organizacionales y en la toma de decisiones referentes a inversión en tecnología.

En otro entorno de ideas, Moreno L. (2008) en su trabajo de grado **“Desarrollo de una herramienta de monitoreo para la verificación y control de políticas de seguridad en servidores mediante el uso de Tivoli Security Compliance Manager”**, se plantea una automatización para el chequeo de seguridad de servidores, como un servicio adicional prestado por IBM como parte de sus servicios postventa.

Se emplea una herramienta Tivoli Security Compliance Manager (TSCM) para el monitoreo automático de la seguridad, brindando información tanto a la empresa IBM, prestador del servicio, como a los clientes, quienes pueden a partir de dicha información tomar decisiones más acertadas en cuanto a la seguridad y el uso de los recursos de su empresa.

Igualmente, Moreno M. (2009) en su investigación **“Propuesta de implementación y actualización de la plataforma de infraestructura de servidores, seguridad informática y telecomunicaciones”**, desarrollado en la Universidad José Antonio Páez, plantea un método para la actualización de plataformas de tecnología de la información como respuesta al crecimiento sostenido en Tecnología de la Información en el Grupo Intershipping C.A. y a sus necesidades de control de los recursos disponibles a través de la red, de la seguridad de la información contenida en el correo electrónico alojado en servidores externos y de la infraestructura para el soporte técnico remoto.

El objetivo de esta investigación fue actualizar la plataforma de tecnología de información para la mejora de los sistemas de correo electrónico, administración y control de las conexiones tanto internas como externas, mejoramiento de las políticas de seguridad informática e interconexión de oficinas geográficamente distantes, para

atender a las necesidades de seguridad, buen uso de los recursos de red, reducción de costos de personal de sistemas de información y disponibilidad de la información en cualquier lugar donde opere la organización.

En este trabajo se plantea el uso de un servidor proxy que gestiona todo el tráfico web de los clientes teniendo control y seguimiento sobre los accesos deseados e indeseados, de igual forma funge como gestor de caché, lo cual permite un tiempo de respuesta más rápido al momento de realizar cualquier tipo de petición web, dado que el servidor determinará si dicha información se encuentra almacenada localmente, o si hay que buscarla en la web.

2.2. Bases Teóricas

2.2.1. Squid

El trabajo de Squid es el de ser un Proxy y un Cache a la vez, como un Proxy, Squid es un intermediario en una transacción Web. Es decir acepta una solicitud de un cliente, procesa esa petición, y luego envía dicha solicitud al servidor de origen, la solicitud se puede registrar, rechazar, e incluso modificar antes de su remisión; como en una Cache, se almacenan los contenidos Web leídos recientemente para su posible reutilización posterior las próximas solicitudes de contenido Web ya visitado se pueden servir desde la Cache, en lugar de conectarse con el servidor de origen nuevamente, aunque se puede deshabilitar la función de almacenamiento en cache no es recomendable ya que esta parte del Proxy es fundamental.

Su funcionamiento es simple, almacena los objetos más solicitados en memoria RAM y los menos en una base de datos en el disco. Los servidores Squid, además, pueden configurarse de forma jerárquica para formar un árbol de proxies dependiendo de las necesidades. Existen dos configuraciones posibles:

1. Como acelerador de http para lograr más prestaciones al servicio de web.
2. Como proxy-caching server para permitir a los usuarios de una corporación utilizar el Proxy Server (PS) para salir hacia Internet.

En el primer modo, actúa como proxy inverso, es decir, acepta una petición del cliente, sirve el objeto si lo tiene y si no, lo solicita y se lo pasa al cliente cuando lo

tiene, almacenándolo para la vez siguiente. En la segunda opción se puede utilizar como control y para restringir los sitios donde se puede conectar en Internet o autorizar el acceso a determinadas horas del día.

Entre sus funciones destacan:

1. Centralizar el tráfico de red.
2. Acelerar el acceso a contenidos Web mediante la cache.
3. Restringir cierto tipo de tráfico según una serie de reglas establecidas en su fichero de configuración.
4. Permitir acceso a otras redes, como Internet, a servidores ubicados en una red Privada.
5. Guardar un registro del tráfico de red.

Squid proporciona un servicio de Proxy que soporta peticiones HTTP, HTTPS y FTP a equipos que necesitan acceder a Internet y a su vez provee la funcionalidad de caché especializado en el cual almacena de forma local las páginas consultadas recientemente por los usuarios. De esta forma, incrementa la rapidez de acceso a los servidores de información Web y FTP que se encuentra fuera de la red interna.

Proxy para SSL

Squid también es compatible con SSL (Secure Socket Layer) con lo que también acelera las transacciones cifradas, y es capaz de ser configurado con amplios controles de acceso sobre las peticiones de usuarios.

Jerarquías de Caché

Squid puede formar parte de una jerarquía de caches. Diversos proxies trabajan conjuntamente sirviendo las peticiones de las páginas. Un navegador solicita siempre las páginas a un sólo proxy, si este no tiene la página en la caché hace peticiones a sus hermanos, que si tampoco las tienen las hacen a sus padres. Estas peticiones se pueden hacer mediante dos protocolos: HTTP e ICMP.

ICP, HTCP, CARP, caché digests

Squid sigue los protocolos ICP, HTCP, CARP y caché digests que tienen como objetivo permitir a un proxy preguntarle a otros proxies caché si poseen almacenado un recurso determinado.

Caché Transparente

Squid puede ser configurado para ser usado como proxy transparente de manera que las conexiones son enrutadas dentro del proxy sin configuración por parte del cliente, y habitualmente sin que el propio cliente conozca de su existencia. De modo predefinido Squid utiliza el puerto 3128 para atender peticiones, sin embargo se puede especificar que lo haga en cualquier otro puerto disponible o bien que lo haga en varios puertos disponibles a la vez.

WCCP

A partir de la versión 2.3 Squid implementa WCCP (Web Cache Control Protocol). Permite interceptar y redirigir el tráfico que recibe un router hacia uno o más proxies caché, haciendo control de la conectividad de los mismos. Además permite que uno de los proxies caché designado pueda determinar como distribuir el tráfico redirigido a lo largo de todo el array de proxies caché.

Control de Acceso

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

Aceleración de servidores HTTP

Cuando un usuario hace petición hacia un objeto en Internet, este es almacenado en el Caché, si otro usuario hace petición hacia el mismo objeto, y este no ha sufrido modificación alguna desde que lo accedió el usuario anterior, Squid mostrará el que ya se encuentra en el Caché en lugar de volver a descargarlo desde Internet. Esta función permite navegar rápidamente cuando los objetos ya están en el caché y además optimiza enormemente la utilización del ancho de banda.

SNMP

Squid permite activar el protocolo SNMP, este proporciona un método simple de administración de red, que permite supervisar, analizar y comunicar información de estado entre una gran variedad de máquinas, pudiendo detectar problemas y proporcionar mensajes de estados.

Caché de resolución DNS

Squid está compuesto también por el programa dnsserver, que se encarga de la búsqueda de nombres de dominio. Cuando Squid se ejecuta, produce un número configurable de procesos dnsserver, y cada uno de ellos realiza su propia búsqueda en DNS. De este modo, se reduce la cantidad de tiempo que la caché debe esperar a estas búsquedas DNS.

2.2.2. Web Cache

El origen del término Cache proviene del Francés y significa literalmente almacenar. Como término en el procesamiento de datos Cache se refiere al almacenamiento de reciente información de computadora para luego a futuro ser recuperada y utilizada. La información almacenada puede o no, ser usada nuevamente, así que la cache puede ser beneficioso solo cuando el costo del almacenamiento de la información es menor que el costo de recuperación o calculo nuevamente de la información.

El concepto de Cache ha encontrado su camino en casi todos los aspectos de la computación y sistemas de redes. Los procesadores de computadoras tienen datos e instrucciones Caches. Los sistemas operativos poseen buffer de almacenamiento cache para los dispositivos de discos y los sistemas de archivos distribuidos (en red). Los sistemas de archivos tanto como NFS y AFS dependen en gran medida del almacenamiento en cache para un buen desempeño. Los routers de Internet almacenan en cache las rutas usadas recientemente. Los DNS Servidores de nombre de Dominios almacenan en cache las direcciones de nombres.

Caches funciona bien debido a un principio conocido como localidad referencial. Hay dos tipos de localidad: Temporal y espacial. La localidad temporal significa que algunos datos son más populares que otros. Como por ejemplo la página de CNN es más popular que la pagina personal de algún individuo, dentro de un determinado periodo de tiempo, alguien tiene más probabilidad a solicitar la pagina de CNN que la pagina personal. La localidad espacial significa que las solicitudes de ciertos datos son probables que se presenten juntos, ejemplo de ello es una solicitud a la página de CNN que está generalmente seguida por la solicitud de todas las páginas de gráficos incrustadas en ella. Cache usa la localidad de referencia para predecir accesos futuro basados en consultas anteriores. Cuando la predicción es correcta hay una mejora significativa en el rendimiento. En la práctica, esta técnica funciona tan bien que podemos encontrar sistemas insoportablemente lentos. Casi todas las tareas de procesamiento de datos muestran la localidad de referencia y por lo tanto se benefician de la Cache.

Cualquier sistema que utiliza el almacenamiento en caché debe tener mecanismos para mantener la coherencia de caché, éste es el proceso mediante el cual las copias en caché se mantienen al día con los originales. Podemos decir que los datos en cache están actualizados o desactualizados. Las Caches pueden reutilizar copias frescas inmediatamente, pero los datos obsoletos usualmente requieren validación. Los algoritmos que mantienen la consistencia pueden ser débiles o fuertes, una débil consistencia significa que la cache a veces devuelve información obsoleta, una fuerte consistencia por otro lado significa que la data almacenada esta siempre validada antes de ser usada. La cache en la CPU y los sistemas de archivos requieren de una fuerte consistencia, sin embargo algunos tipos de cache tanto como en routers y servidores DNS son efectivos incluso si retornan información obsoleta.

2.2.3. Linux

El kernel (núcleo en castellano) Linux es, sin duda, la aplicación estrella del software libre, hasta el punto de que, aun siendo una pequeña parte del sistema, ha dado nombre a todo él. Es más, se podría incluso afirmar que el propio software libre se confunde en multitud de ocasiones con Linux, lo que no deja de ser un gran desacierto, ya que existe software libre que corre sobre sistemas que no se basan en Linux (de hecho, una de las grandes metas del movimiento y de muchos proyectos de software libre es que las aplicaciones puedan ejecutarse en multitud de entornos). Por otro lado, también existen aplicaciones que funcionan en Linux y que no son software libre (Acrobat Reader, el lector de documentos PDF, también cuenta con su versión para Linux).

Para evitar la asociación del software libre únicamente con sistemas Linux existen varios proyectos que se dedican a integrar y distribuir aplicaciones libres que se corren en sistemas Windows. Uno de estos proyectos, probablemente el más conocido y completo, es GNUWin, que distribuye en un CD autoarrancable más de un centenar de aplicaciones libres para sistemas Win32. La mayoría de estas aplicaciones también se encuentran disponibles en las distribuciones GNU/Linux comunes, por lo que son una buena herramienta para ir preparando el cambio de sistemas Windows a GNU/Linux de manera más sosegada.

2.2.4. La Licencia Pública General de GNU (GNU GPL)

La Licencia Pública General del proyecto GNU (más conocida por su acrónimo en inglés GPL) es con diferencia la licencia más popular y conocida de todas las licencias del mundo del software libre. Su autoría corresponde a la Free Software Foundation (promotora del proyecto GNU) y en un principio fue creada para ser la licencia de todo el software generado por la Free Software Foundation (FSF). Sin embargo, su utilización ha ido más allá hasta convertirse en la licencia más utilizada, incluso por proyectos bandera del mundo del software libre, como es el caso del núcleo Linux.

La licencia GPL es interesante desde el punto de vista legal porque hace un uso tan curioso de la legislación de copyright que haciendo estricto uso del término llega a una solución totalmente contraria a la original, hecho por el que también se ha venido a llamar una licencia copyleft. En líneas básicas, la licencia GPL permite la redistribución binaria y la de las fuentes, aunque, en el caso de que redistribuya de manera binaria, obliga a que también se pueda acceder a las fuentes. Asimismo, está permitido realizar modificaciones sin restricciones, aunque sólo se pueda integrar código licenciado bajo GPL con otro código que se encuentre bajo una licencia idéntica o compatible, lo que ha venido a llamarse el efecto viral de la GPL, ya que el código publicado una vez con esas condiciones nunca puede cambiar de condiciones.

La licencia GPL está pensada para asegurar la libertad del código en todo momento, ya que un programa publicado y licenciado bajo sus condiciones nunca podrá ser hecho propietario. Es más, ni ese programa ni modificaciones al mismo pueden ser publicados con una licencia diferente a la propia GPL. Los partidarios de las licencias tipo Berkeley Software Distribution (BSD) ven en esta cláusula un recorte de la libertad, mientras que sus seguidores ven en ello una forma de asegurarse que ese software siempre va a ser libre. Por otro lado, se puede considerar que la licencia GPL maximiza las libertades de los usuarios, mientras que las de tipo BSD lo hacen para los desarrolladores. Nótese, sin embargo, que en el segundo caso estamos hablando de los desarrolladores en general y no de los autores, ya que muchos autores consideran que la licencia GPL es más beneficiosa para sus intereses, ya que obliga a sus competidores a publicar sus modificaciones (mejoras, correcciones, entre otras) en caso de redistribuir el software, mientras que con una licencia tipo BSD éste no tiene por qué ser el caso. De cualquier modo, podemos ver que la elección de licencia no es una tarea fácil y que hay que tener multitud de factores en cuenta.

En cuanto a la naturaleza contraria al copyright, esto se debe a que la filosofía que hay detrás de esta licencia (y detrás de la Free Software Foundation) es que el software no debe tener propietarios. Aunque es cierto que el software licenciado con

la GPL tiene un autor, que es el que a fin de cuentas permite la aplicación de la legislación de copyright sobre su obra, las condiciones bajo las que publica su obra confieren a la misma tal carácter que podemos considerar que la propiedad del software corresponde a quien lo tiene y no a quien lo ha creado.

2.2.5. Debian GNU/Linux

Debian es un sistema operativo libre que en la actualidad utiliza el núcleo de Linux para llevar a cabo su distribución (aunque se espera que existan distribuciones Debian basadas en otros núcleos, como es el caso con The HURD, en el futuro). Actualmente está disponible para varias arquitecturas diferentes, incluyendo Intel x86, ARM, Motorola, 680x0, PowerPC, Alpha y SPARC. Debian no es sólo la distribución GNU/Linux más grande en la actualidad, también es una de las más estables y disfruta de varios premios en cuanto a la preferencia de los usuarios. Aunque su base de usuarios sea difícil de estimar, ya que el proyecto Debian no vende CD u otros medios con su software, y el software que contiene puede ser redistribuido por cualquiera que así lo desee, podemos suponer sin faltar mucho a la verdad que se trata de una distribución importante dentro del mercado de GNU/Linux. En Debian existe una categorización según la licencia y los requisitos de distribución de los paquetes. El núcleo de la distribución Debian (la sección llamada main, que aglutina una gran variedad de paquetes) está compuesto sólo por software libre de acuerdo con las (Debian Free Software Guidelines). Está disponible en Internet para ser descargado y muchos redistribuidores lo venden en CD u otros medios.

Las distribuciones de Debian son creadas por cerca de un millar de voluntarios (generalmente profesionales de la informática). La labor de estos voluntarios radica en tomar los programas fuente –en la mayoría de los casos de sus autores originales–, configurarlos, compilarlos y empaquetarlos, de manera que un usuario típico de una distribución Debian sólo tenga que seleccionar el paquete para que el sistema lo añada sin mayores problemas. Esto que a simple vista puede parecer simple, se torna complejo en cuanto se introducen factores como las dependencias entre los diferentes

paquetes (el paquete A necesita, para poder funcionar, del paquete B) y las diferentes versiones de todos estos paquetes. La labor de los integrantes del proyecto Debian es la misma que la que se realiza en cualquier otra distribución: la integración de software para su correcto funcionamiento conjunto. Además del trabajo de adaptación y empaquetamiento, los desarrolladores Debian se encargan de mantener una infraestructura de servicios basados en Internet (sitio web, archivos en línea, sistema de gestión de errores, listas de correo de ayuda, soporte y desarrollo, entre otros), de varios proyectos de traducción e internacionalización, del desarrollo de varias herramientas específicas de Debian y, en general, de cualquier elemento que hace que la distribución Debian sea posible. Aparte de su naturaleza voluntaria, el proyecto Debian tiene una característica que lo hace especialmente singular: el contrato social de Debian. Este documento contiene no sólo los objetivos principales del proyecto Debian, sino también los medios que se utilizarán para llevarlos a cabo.

Debian también es conocida por tener una política de paquetes y de versionado muy estricta con el fin de conseguir una mayor calidad del producto. Así, en todo momento existen tres sabores diferentes de Debian: una versión estable, una inestable y otra en pruebas. Como su propio nombre indica, la versión estable es la versión indicada para sistemas y personas no aptas para sobresaltos. Su software ha de pasar un periodo de congelación en el que sólo se corrigen erratas. La norma es que en la versión estable de Debian no ha de haber ningún error crítico conocido. Por contra, la versión estable de Debian no suele tener las últimas versiones del software (lo más novedoso). Para los que deseen tener una versión con el software más actual existen otras dos versiones de Debian coetáneas con la estable. La versión inestable incluye paquetes en vía de estabilización, mientras que la versión en pruebas, como su propio nombre indica, es la más proclive a fallar y contiene lo último de lo último en lo que a novedades de software se refiere.

2.3. Definición de Términos Básicos

ACL: es una lista de control de acceso que almacena información de permisos y privilegios de un objeto de la red.

Cache Web: Se puede definir como la disponibilidad en caché para el almacenamiento de documentos Web (textos, imágenes, etc.), permitiendo mejorar el ancho de banda consumido, la carga de los servidores y el retardo en la descarga. Ya que un caché Web intercepta y almacena copias de todo el contenido Web que pasa por él, de manera que las subsiguientes peticiones puedan ser respondidas por el propio caché.

Copyleft: es la forma general de hacer un programa software libre y requerir que todas las modificaciones y versiones extendidas del programa sean también software libre.

Daemon: aplicación UNIX que está alerta permanentemente en un servidor de Internet para realizar determinadas tareas como, por ejemplo, enviar un mensaje de correo electrónico o servir una página Web.

Debian: es un sistema operativo libre que en la actualidad utiliza el núcleo de Linux para llevar a cabo su distribución.

Dirección IP: son 4 números decimales comprendidos entre 0 y 255 separados por puntos (.) que identifican un sistema en una red basada en los protocolos TCP/IP.

FSF (Free Software Foundation, Fundación de software libre): organización que promueve el desarrollo de software libre.

FTP: (File Transfer Protocol): protocolo empleado para transmitir archivos entre dos sistemas conectados a una red TCP/IP.

GNU (GNU's not UNIX, GNU no es UNIX): un proyecto de la FSF dedicado a crear un sistema UNIX completo independiente de UNIX sistema V.

GPL: (General Public License, licencia pública general): licencia que se otorga a los programas del proyecto GNU y que garantiza a los usuarios la posibilidad de hacer copias de éstos de forma gratuita. La licencia permite realizar modificaciones al código fuente del programa siempre y cuando se preserve el copyright.

GUI (Graphical User Interface, Interfaz grafica de usuario): interfaz de acceso a un sistema basado en el uso de elementos gráficos, como íconos, ventanas, botones, entre otros.

HTML (Hyper Text Markup Language, lenguaje de definición de hipertexto): lenguaje empleado en el diseño de documentos para la WWW. Se denomina hipertexto porque contiene enlaces que apuntan a otros documentos.

HTTP (Hyper Text Transfer Protocol, protocolo de transferencia hipertexto): protocolo empleado en la WWW para transferir documentos HTML.

ICMP (Internet Control Message Protocol, protocolo de mensajes de control de Internet): protocolo empleado en Internet para transmitir información de control. Se usa para realizar diagnósticos sobre el estado de la red.

Internet: Red mundial de computadoras basadas en el conjunto de protocolos TCP/IP.

Kernel: es el núcleo o centro de gestión de un sistema operativo, y está encargado de gestionar recursos, a través de servicios de llamada al sistema.

LAN (Local Area Network): Red de computadoras de pequeñas dimensiones.

Linux: sistema operativo UNIX de gran aceptación en entornos Intel acogidos a la licencia GPL y creado por Linus Torvalds a principios de los años noventa.

Llamadas al Sistema: llamadas efectuadas desde un proceso UNIX al núcleo de sistema operativo.

Log: es una bitácora o un registro oficial de eventos ocurridos en un software durante un rango de tiempo en particular.

LPD (Linux Documentation Project, proyecto de documentación de Linux): proyecto coordinado por los desarrolladores de linux en cuyo objeto es proporcionar toda documentación relativa a este sistema operativo. El LDP comprende varios formatos para la documentación: desde páginas de manual a documentos para Texinfo pasando por manuales impresos.

Proceso: Programa en ejecución en un sistema UNIX.

Programa Shell: conjunto de ordenes de un shell situadas en un archivo para realizar una cierta operación compleja.

Protocolo: es una convención o estándar que controla o permite la conexión, comunicación, y transferencia de datos entre dos puntos finales.

RFC (Request For Comments): serie de documentos que detalla prácticamente todo lo relacionado con la tecnología de la que se sirve Internet: protocolos, recomendaciones, comunicaciones.

Script: conjunto de instrucciones que serán ejecutadas mediante un intérprete de órdenes, como por ejemplo el shell.

Servidor: aplicación que proporciona un cierto servicio a las aplicaciones clientes que lo solicitan. La solicitud puede hacerse a través de una red de computadoras. A menudo se denomina servidor al sistema de una red que contiene una aplicación servidora.

Servidor Proxy: Es un equipo de computación que intercepta las conexiones de red que un cliente hace a un servidor de destino. Siendo intermediario entre ambos ya sea por motivos de seguridad, rendimiento y anonimato.

Squid: es un software bajo licencia GPL cuya finalidad es brindar servicios de Proxy y cache, es un intermediario en una transacción Web, se posiciona entre un servidor y un cliente aceptando peticiones del cliente, procesándola y luego enviando las peticiones al servidor de origen. Las solicitudes pueden ser registradas, rechazadas e incluso modificadas antes de ser reenviadas. como cache almacena la información recientemente visitadas por los usuarios en una red para su posible reutilización futura, evitando la carga en servidor nuevamente de la información ya visitada, mejorando así el rendimiento y ancho de banda y navegación Web.

TCL (Tool Command Language): es un muy poderoso pero fácil de aprender lenguaje de programación dinámico, adecuado para una amplia gama de aplicaciones, incluyendo aplicaciones Web y de escritorio, redes, administración, ensayos y muchos más. El código es abierto y amigable, TCL es un lenguaje maduro aún en evolución es una verdadera plataforma cruzada y de fácil implementación y altamente extensible.

TCP/IP (Transport Control Protocol/Internet Protocol): nombre con el que se denomina un conjunto de protocolos empleados para la comunicación de sistemas en redes de área extensa (WANs). Es la base de la red Internet.

UNIX: sistema Operativo creado en 1969 en los laboratorios BELL por Ken Thompson, Dennis Ritchie y otros investigadores. Actualmente el término engloba a cualquier sistema operativo que cumple con los estándares de UNIX. Entre las características más importantes cabe resaltar que es un sistema operativo de red, es multitarea y multiusuario.

URL (Universal Resource Locator, localizador universal de recurso): formato de direcciones empleado en los navegadores de red. Permite indicar la dirección de un documento junto al protocolo que se debe emplear para acceder a él.

WWW (World Wide Web, tela de araña mundial): nombre con el que se conoce al extenso conjunto de documentos hipertexto HTML situados en Internet.

CAPÍTULO III

MARCO METODOLÓGICO

3.1. Tipo de Investigación

La metodología de esta investigación se enmarca dentro de la modalidad de proyecto factible, apoyada en una investigación de campo, en el manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales de la universidad Pedagógica Experimental Libertador (UPEL, 2006) se define como:

El proyecto factible consiste en la investigación, elaboración y desarrollo de una propuesta, de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos; puede referirse a la formulación de políticas, programas, tecnologías, métodos, o procesos. El proyecto debe tener apoyo en una investigación tipo documental de campo o un diseño que incluya ambas modalidades. (página 16).

3.2. Diseño

Como anteriormente se mencionó las investigaciones de tipo Proyecto Factible deben de tener un apoyo ya sea de tipo documental, de campo o una modalidad que los incluya a ambos, para la UPEL (2006), la investigación de campo es:

El análisis sistemático de problemas en la realidad, con el propósito bien sea de describirlos, interpretarlos, entender su naturaleza y factores constituyentes, explicar sus causas y efectos o predecir su ocurrencia, haciendo uso de métodos característicos de cualquiera de los paradigmas o enfoques de investigación conocidos o en desarrollo. La fuente principal de datos es el sitio donde se presenta el problema, los datos de interés son recogidos en forma directa de la realidad, en este sentido se trata de investigaciones a partir de datos originarios o primarios. (página 8).

Así mismo, la investigación documental según la UPEL (2006) “es el estudio de un problema con el propósito de ampliar y profundizar el conocimiento de su

naturaleza principal, en trabajos previos, así como información y datos divulgables por medios impresos” (página 6)

Dicho esto, se considera que este Proyecto Factible apoya su investigación en un diseño de campo y documental. Ya que los datos para su realización fueron obtenidos directamente del lugar donde se basó la investigación y de otros muchos de material teórico ya existente.

El presente trabajo se enmarca en la modalidad de proyecto factible, pues se plantea la solución de un problema real, que presenta mejoras a proyectos ya existentes y componentes nuevos para dar solución a dicho problema.

3.3. Nivel de la Investigación

La presente investigación es de tipo descriptiva, ya que se miden y evalúan diversos aspectos, dimensiones y componentes de un sistema. La investigación descriptiva según la UPEL (2006) "es el tipo de investigación concluyente que tiene como objetivo principal la descripción de algo, generalmente las características o funciones del problema en cuestión" (página 9)

3.4. Fases Metodológicas

Esta parte corresponde a los pasos que se emplearon para dar repuestas 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, llamada así por sus siglas en inglés Rational Unified Process, y 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 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, elaboración, construcción y transición. 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 o preparación del release (versión).

Fase I: Determinación de alcance, requerimientos básicos y factores de riesgo

En esta fase, se definió el ámbito y los objetivos del proyecto, así como la funcionalidad y capacidades del producto. Se examinaron algunas herramientas de administración de Squid Proxy Server disponibles en el mercado actual para determinar cuáles son sus fortalezas y debilidades, con la intención de mejorar las funcionalidades de los mismos e incluir en el desarrollo del nuevo sistema aquellos elementos que tengan relevancia para la solución del problema planteado. Se revisó la interfaz gráfica, el sistema de almacenamiento de datos empleados, la usabilidad y la integración con el Proxy Squid. Una vez examinados se determinarán cuales son los elementos relevantes para el desarrollo del sistema propuesto, con la finalidad de incluirlos o mejorarlos según el alcance de la investigación.

Fase II: Análisis de Requerimientos

Seguidamente, en la fase de elaboración se determinó tanto la funcionalidad como el dominio del problema, se definió una arquitectura básica y se planificó el proyecto basado en los recursos disponibles. Se determinaron los requerimientos de forma detallada de cada uno de los componentes del sistema propuesto, con la finalidad de hacer explícito el proceso de construcción de los mismos y definir la forma de interacción entre ellos, lo que permitirá conocer que datos intercambiarán y la forma de hacerlo.

Fase III: Construcción de Interfaz y Módulos.

En esta etapa se construyeron cada uno de los módulos y la interfaz de la aplicación fundamentando su creación a partir de los diseños propuestos y tomando en cuenta el alcance previamente analizado en etapas anteriores de la investigación.

Fase IV: Evaluación de los Resultados del Sistema Propuesto.

En esta fase se crearon los medios para la comunicación entre cada uno de los módulos, fundamentando su desarrollo en el diseño planteado para ello en fases anteriores. Esta etapa se enfocó en conformar las partes del sistema en un todo.

En esta etapa de la investigación se analizaron los resultados obtenidos del desarrollo del sistema, se evaluó si el sistema cumplía los requisitos inicialmente propuestos y si las expectativas en cuanto a su desarrollo fueron cubiertas.

CAPÍTULO IV

RESULTADOS

El presente trabajo tuvo como objetivo desarrollar una aplicación web para la administración de servidores proxy en sistemas operativos gnu/Linux cuya plataforma de servicio fuese SQUID Proxy, que permitiera mejorar la administración y configuración de dichos servidores con una interfaz intuitiva y con una alta usabilidad para usuarios especializados en Tecnología de la Información, mejorando también el repositorio de datos empleado y la capacidad de usar la información provista por SQUID Proxy.

Los resultados arrojados en cada una de las cuatro fases, obtenidos de la aplicación de la metodología RUP (Rational Unified Process), se presentan a continuación:

4.1. Fase I: Alcance, Requerimientos Básicos y Factores De Riesgo

En esta fase, se definió el ámbito y los objetivos del proyecto, así como la funcionalidad y capacidades del producto. Se examinaron algunas herramientas de administración de Squid Proxy Server disponibles en el mercado actual y se determinaron sus fortalezas y debilidades en cuanto a: la interfaz gráfica, el sistema de almacenamiento de datos, la usabilidad, las capacidades de configuración y la integración con el Proxy Squid. Una vez examinados se determinaron cuales eran los elementos relevantes para el desarrollo del sistema propuesto.

Los artefactos obtenidos como resultado en la fase I fueron:

- a) Plan de Desarrollo del Proyecto
- b) Cuadros de Análisis de Herramientas de Administración de Squid Proxy Server.
- c) Cuadro de Riesgos
- d) Modelo del Dominio

e) Modelo Inicial de Caso de Uso

f) Diagrama Inicial de la Arquitectura

El plan de desarrollo es la planificación de las actividades a llevar a cabo, y la definición de entregables de cada fase de la metodología RUP.

Cuadro 1. Plan de Desarrollo del Producto

Nombre de tarea	Duración
<input type="checkbox"/> Desarrollar una aplicación web para la administración de servidores proxy en sistemas operativos GNU/Linux. Caso de Uso	180 días
<input type="checkbox"/> Determinar el alcance, requerimientos básicos y factores de riesgo del sistema propuesto.	45 días
Cuadro Comparativo de Herramientas de Administración de Squid Proxy Server.	5 días
Cuadro de Riesgos	5 días
Modelo del Dominio	5 días
Modelo Inicial de Caso de Uso	20 días
Diagrama Inicial de la Arquitectura	10 días
<input type="checkbox"/> Analizar los requerimientos detallados de los procedimientos propuestos.	45 días
Diagrama de Actividades.	5 días
Modelos de Caso de Uso del Sistema.	15 días
Especificaciones de Casos de Uso.	5 días
Diagramas de Secuencia.	5 días
Diagramas de Colaboración.	5 días
Modelo de Datos.	10 días
<input type="checkbox"/> Construir los módulos e interfaces de la aplicación basado en los diseños propuestos.	45 días
Pantallas del Sistema	30 días
Diagrama de Arquitectura Final	10 días
Diagrama de Despliegue	5 días
<input type="checkbox"/> Evaluar los resultados obtenidos del sistema propuesto.	50 días
Casos de Prueba	30 días
Manual del Usuario	20 días

Fuente: Montesino, P y Morales, J (2010)

Los siguientes cuadros muestran el análisis realizado a tres herramientas de administración de Squid Proxy Server existentes en la actualidad, donde se destacan las fortalezas y debilidades de cada una de ellas.

Cuadro 2. Cuadro de Análisis de la Aplicación Webmin

Webmin	Fortalezas	Debilidades
Interfaz Gráfica	Esta basada en una arquitectura web que permite administrar de manera grafica sistemas Unix. Posee una gran variedad de módulos de administración para servicios tales como Apache, DNS, Usuarios, y en nuestro caso de estudio Squid Proxy.	Manejo inadecuado de los elementos de configuración dispuestos en pantalla, causando en muchos casos saturación visual y confusión por parte del usuario.
Usabilidad	Es una Herramienta de gran utilidad en la configuración tanto básica como avanzada por parte de administradores de red con experiencia.	Es una herramienta orientada a administradores de red con gran experiencia y no para usuarios noveles que aunque no es limitativo, la gran cantidad de parámetros por módulo de configuración hace engorrosa y tediosa la misma.
Integración con el Proxy Squid	Trabaja bajo su propio servicio web, accesible vía TCP desde el puerto 10000. lo que le concede privilegios dentro del sistema Linux para hacer escritura y lectura del archivo de configuración squid.conf.	Debido a sus características de funcionar como su propio proceso y servicio web, la seguridad ha sido en varias ocasiones un punto frágil para la aplicación donde usuarios han conseguido saltarse reglas de control de accesos, obteniendo así información y privilegios sobre módulos de configuración del sistema sin la debida autorización, como lo podría ser el caso del Squid Proxy, ocasionando sabotajes y delitos contra las políticas de seguridad presentes en la organización dependiente de este servicios.
	Webmin genera un completo informe estadístico relacionado al consumo y tráfico en la red para esto hace uso de una herramienta desarrollada en su mismo lenguaje Perl llamada Calamaris.	Presenta la información en formato HTML y ASCII lo que dificulta su entendimiento para usuarios noveles.
Sistema de almacenamiento de datos	No requiere de un servidor o manejador de Base de Datos para almacenar la información relacionada con el análisis de log.	Almacena la información en archivos planos, lo que no permite una óptima manipulación de la Información.

Fuente: Montesino, P y Morales, J (2010)

Cuadro 3. Cuadro de Análisis de la Aplicación Sarg

Sarg	Fortalezas	Debilidades
Interfaz Gráficas	Resultados estadísticos basados en la web	La configuración del mismo no se hace sobre si interfaz web, posee archivos de configuración los cuales se modifican desde consola o mediante el uso de algún editor de texto.
Usabilidad	Interfaz de los resultados estadísticos amigables, sencillos e intuitivos para todo tipo de usuario.	Instalación y configuración requieren de conocimientos medios o avanzados en sistemas basados en Unix.
Integración con el Proxy Squid	El alcance de Sarg con respecto al Squid proxy es a nivel de los archivos log, donde sólo consulta los datos registrados en el access.log generando los reportes estadísticos	Su filosofía es sólo gestionar el access.log del Squid proxy sin tener nada que ver con módulos que permitan la configuración del Squid Proxy
Sistema de almacenamiento de datos	Almacenamiento de la información sin necesidad de servicios de base de datos	Archivos planos

Fuente: Montesino, P y Morales, J (2010)

Cuadro 4. Cuadro de Análisis de la Aplicación Pysquila

Pysquila	Fortalezas	Debilidades
Interfaz Gráficas	Sencilla basada en la web.	No cuenta con interfaz gráfica para su instalación ni configuración. Requiere de librerías especiales para la puesta en pantalla de los gráficos.
Usabilidad	Intuitiva fácil de manipular	Su instalación requiere de conocimientos medios avanzados en administración de equipos
Integración con el Proxy Squid	Esta desarrollado para solo obtener resultados usados en la generación de estadísticas de consumo y tráfico de red.	
Sistema de almacenamiento de datos	Almacena los resultados obtenidos en una base de datos MySQL.	No almacena una bitácora de análisis previos lo que puede ocasionar incongruencias en la información, al momento de un nuevo análisis.

Fuente: Montesino, P y Morales, J (2010)

El cuadro de riesgos muestra todos aquellos elementos a lo que el sistema podría estar expuesto.

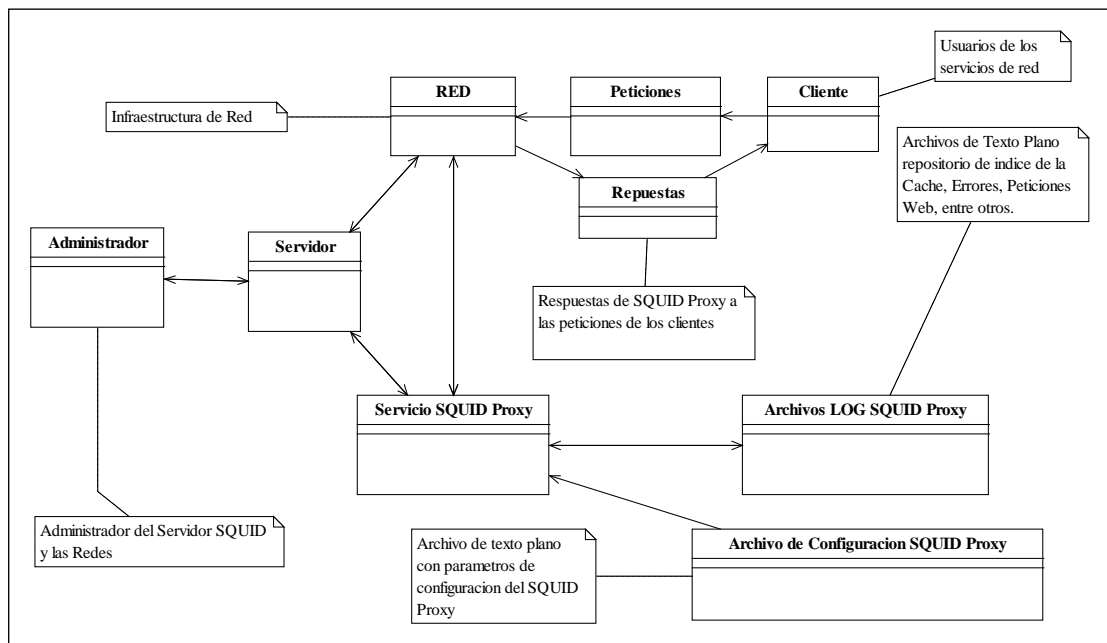
Cuadro 5. Cuadro de Riesgos

Riesgo	Descripción
Seguridad	Carencia de políticas de uso de los recursos. Seguridad física de los equipos. La falta de políticas de seguridad.
Hardware	La carencia de políticas de mantenimiento y actualización de los recursos de hardware.
Software	La no disponibilidad de las herramientas requeridas para el funcionamiento del software. La carencia de políticas de mantenimiento y actualización de los recursos de software. Uso inapropiado del sistema
Squid Proxy	Abandono del proyecto GNU Squid Proxy. Modificaciones estructurales profundas sobre el funcionamiento y los componentes del servidor Squid.

Fuente: Montesino, P y Morales, J (2010)

En el modelo del dominio se capturaron los tipos más importantes de objetos en el contexto del sistema. El modelo de dominio se muestra en el siguiente cuadro:

Cuadro 6. Modelo del Dominio



Fuente: Montesino, P y Morales, J (2010)

El modelo inicial de caso de uso muestra el escenario de interacción del Administrador con el sistema actual.

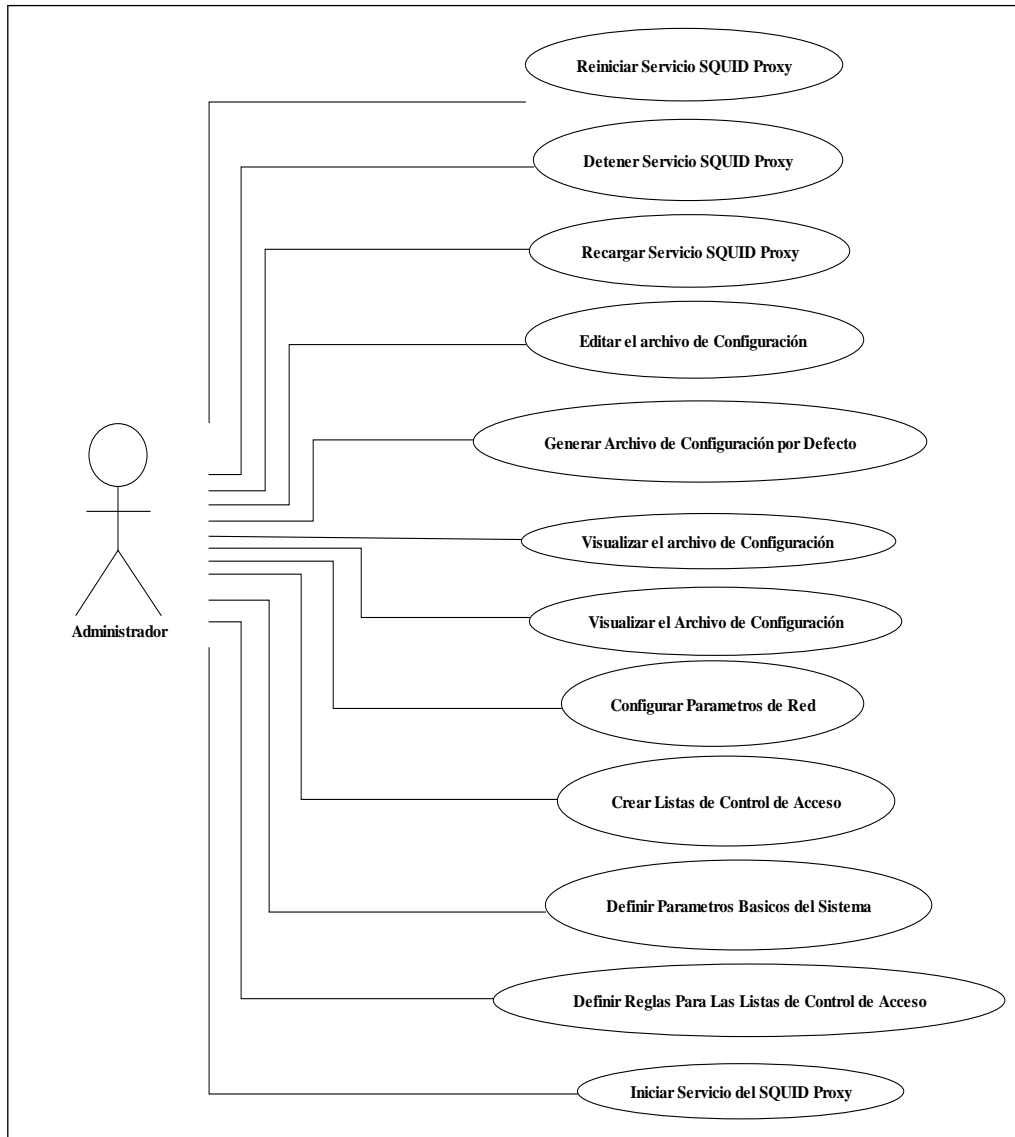
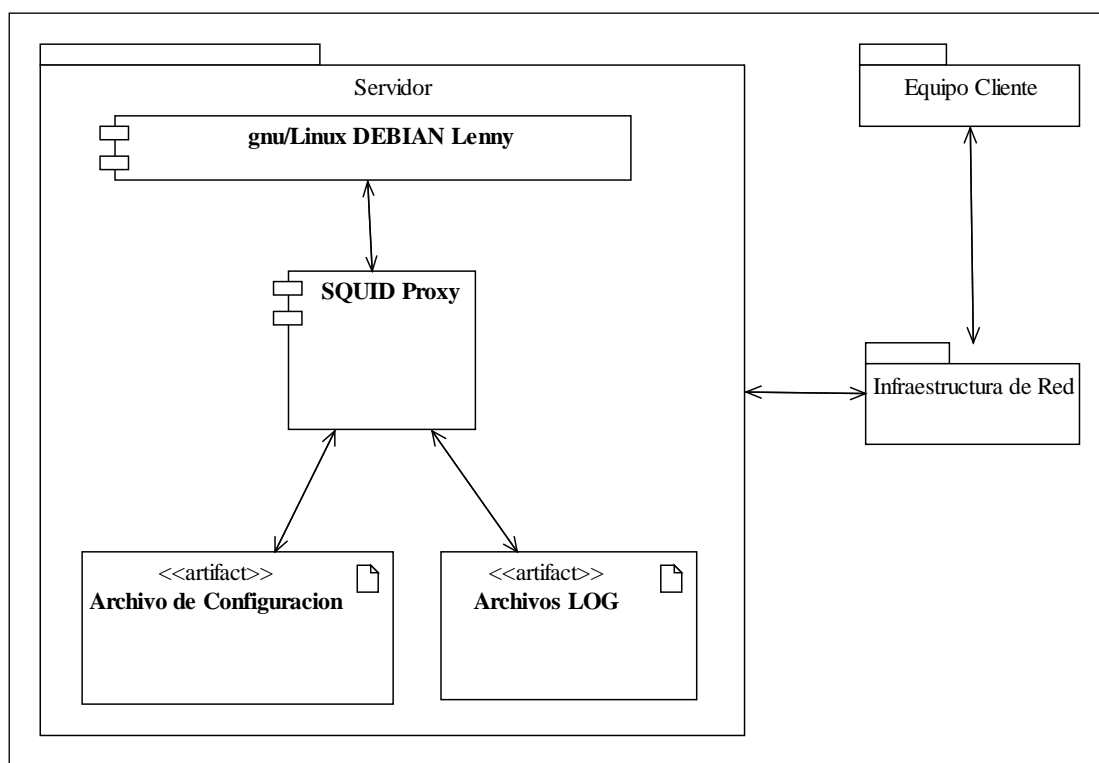


Figura 1. Caso de Uso Inicial
Fuente: Montesino, P y Morales, J (2010)

El diagrama inicial de la arquitectura nos muestra los objetos representativos y sus interrelaciones en el sistema.

Cuadro 7. Diagrama Inicial de la Arquitectura



Fuente: Montesino, P y Morales, J (2010)

4.2 Fase II: Análisis de Requerimientos

En esta fase se determinó la funcionalidad y el dominio del problema, se definió una arquitectura básica y se planificó el proyecto basado en los recursos disponibles. Se determinó los requerimientos de forma detallada de cada uno de los componentes del sistema propuesto y se obtuvieron los siguientes modelos y diagramas:

- a) Diagrama de Actividades.
- b) Modelos de Caso de Uso del Sistema.
- c) Especificaciones de Casos de Uso.
- d) Diagramas de Secuencia.
- e) Diagramas de Colaboración.
- f) Modelo de Datos.

Las siguientes figuras muestran los diagramas de actividades. Estos representan el flujo de trabajos de las actividades que fueron consideradas más críticas para el sistema

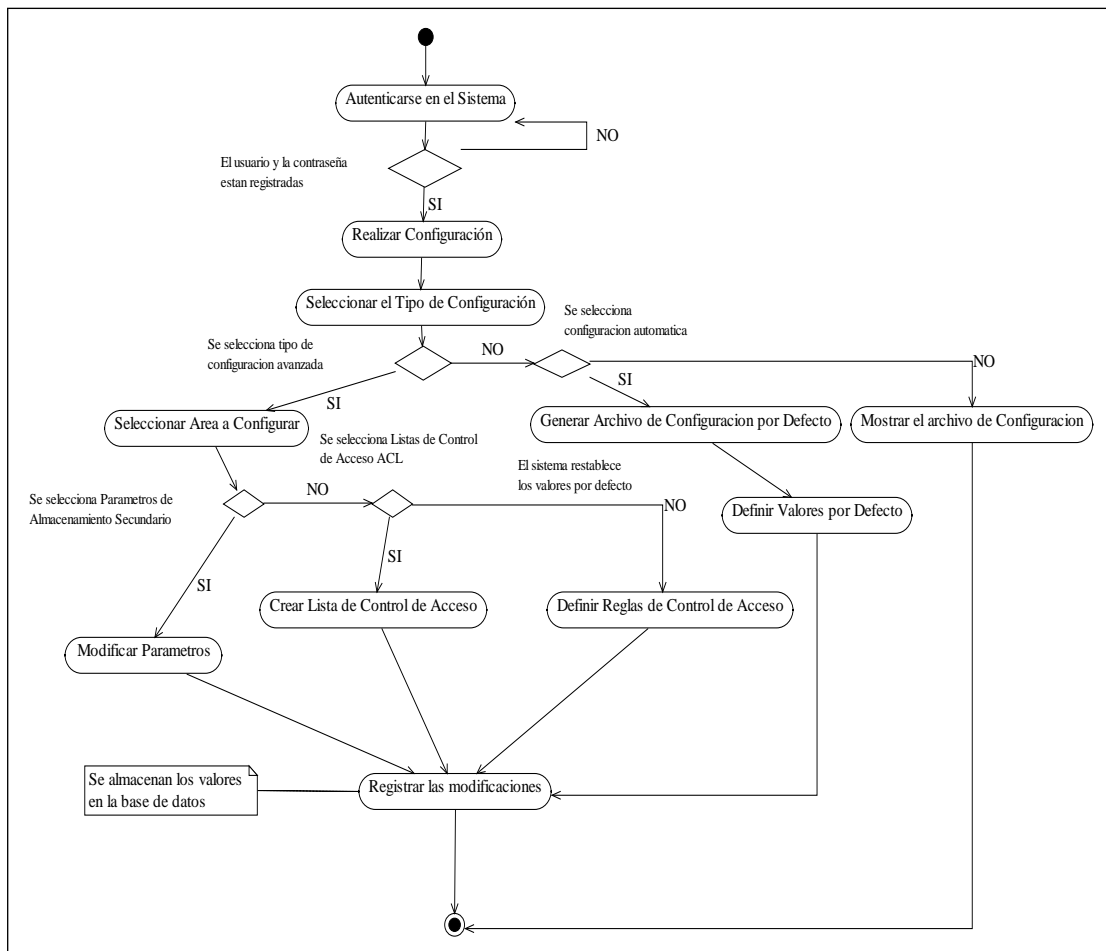


Figura 2. Diagrama de Actividad de Configuración

Fuente: Montesino, P y Morales, J (2010)

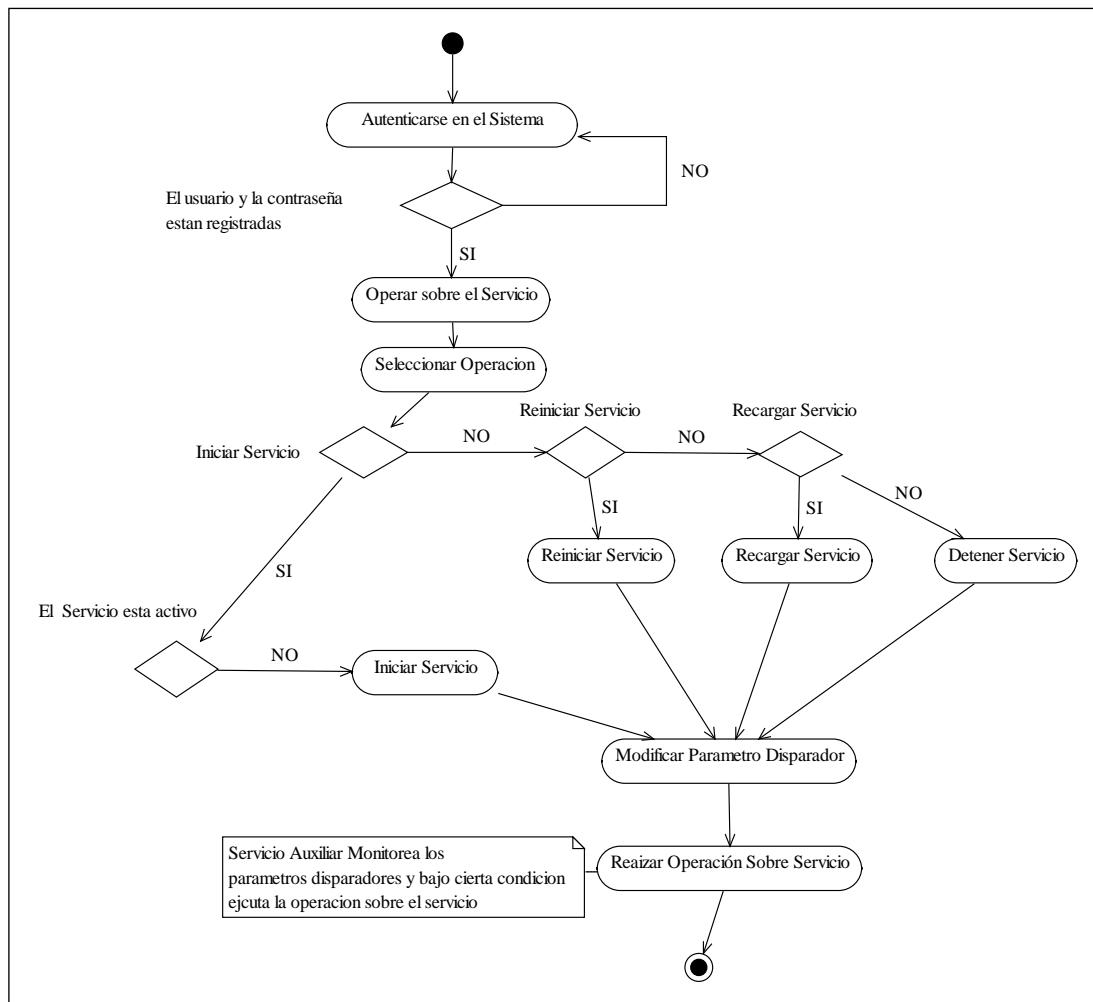


Figura 3. Diagrama de Actividad del Servicio

Fuente: Montesino, P y Morales, J (2010)

El modelo de caso de uso del sistema muestra el escenario de interacción del Administrador con el sistema desarrollado.

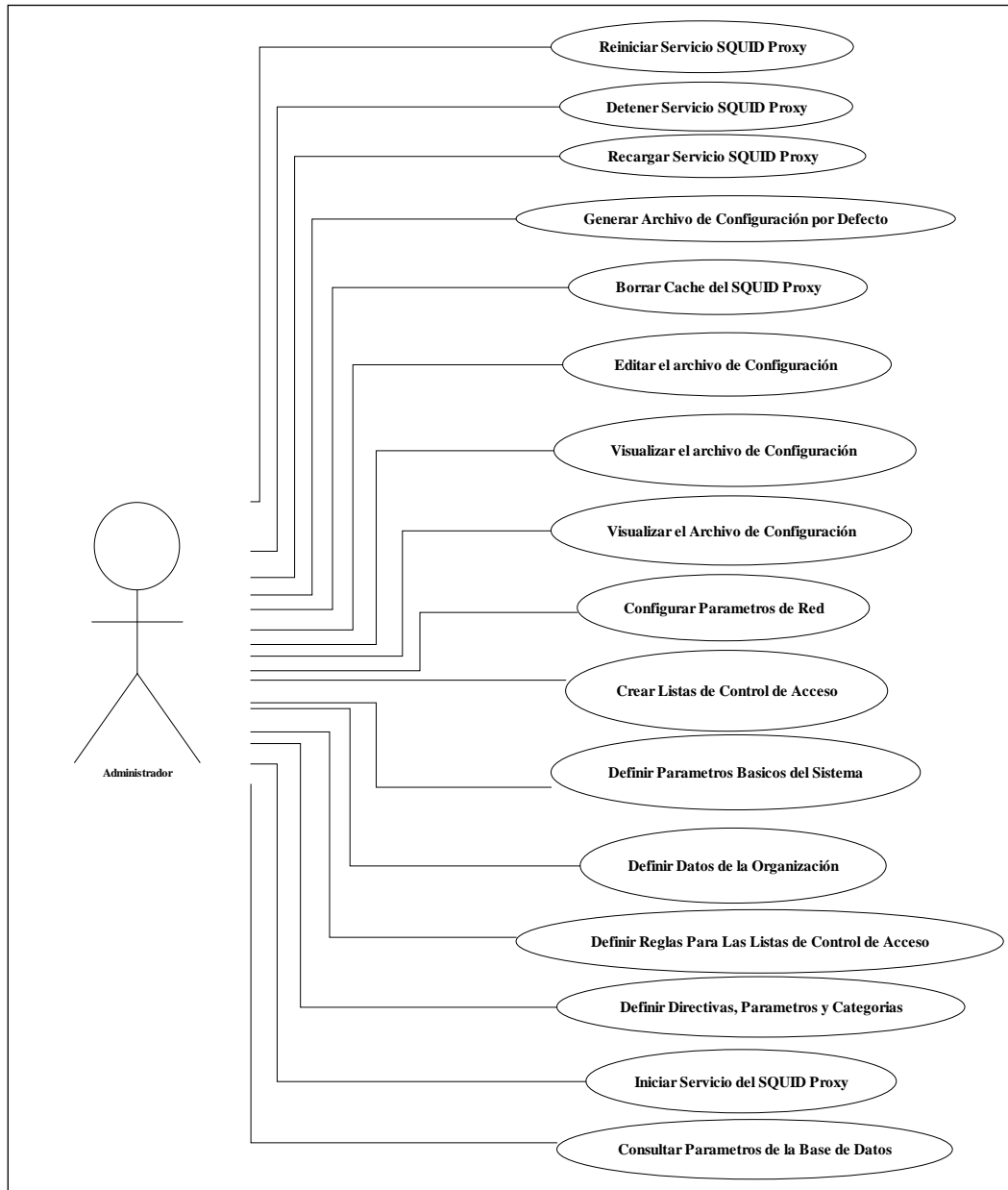


Figura 4. Modelo de Caso de Uso Administrador

Fuente: Montesino, P y Morales, J (2010)

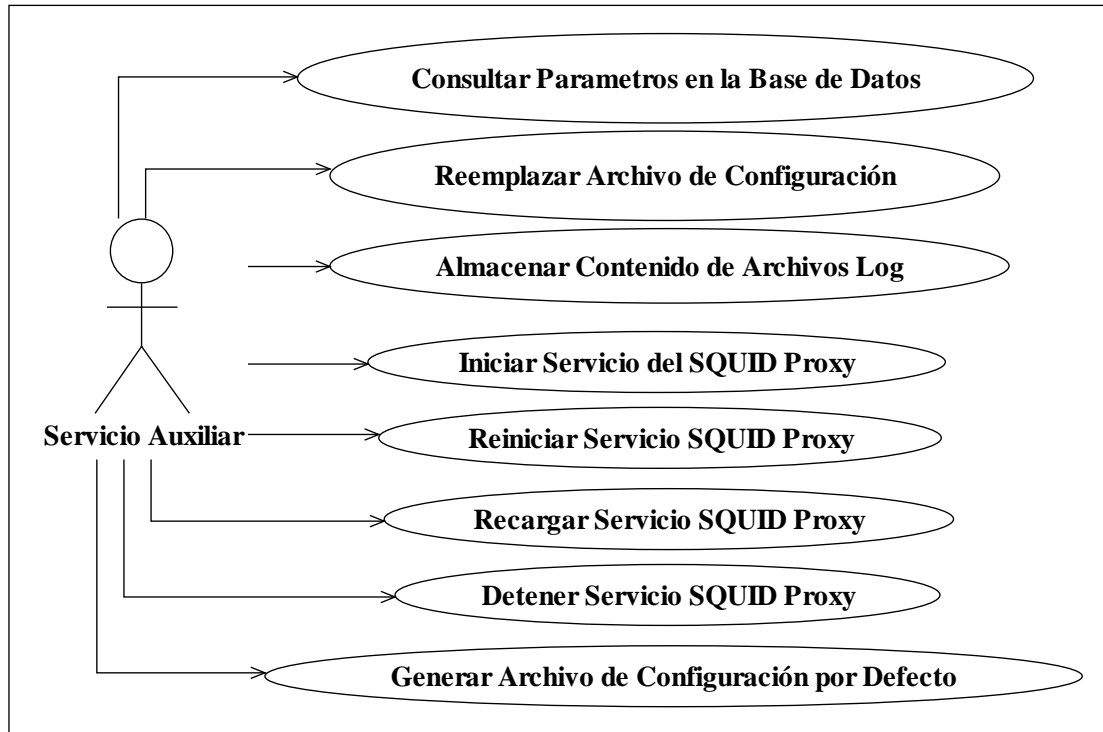


Figura 5. Modelo de Caso de Uso del Servicio Auxiliar

Fuente: Montesino, P y Morales, J (2010)

Las siguientes figuras muestran las especificaciones de caso de prueba. En ellas se describe la funcionalidad y el procedimiento asociado a las acciones del actor sobre los objetos del sistema

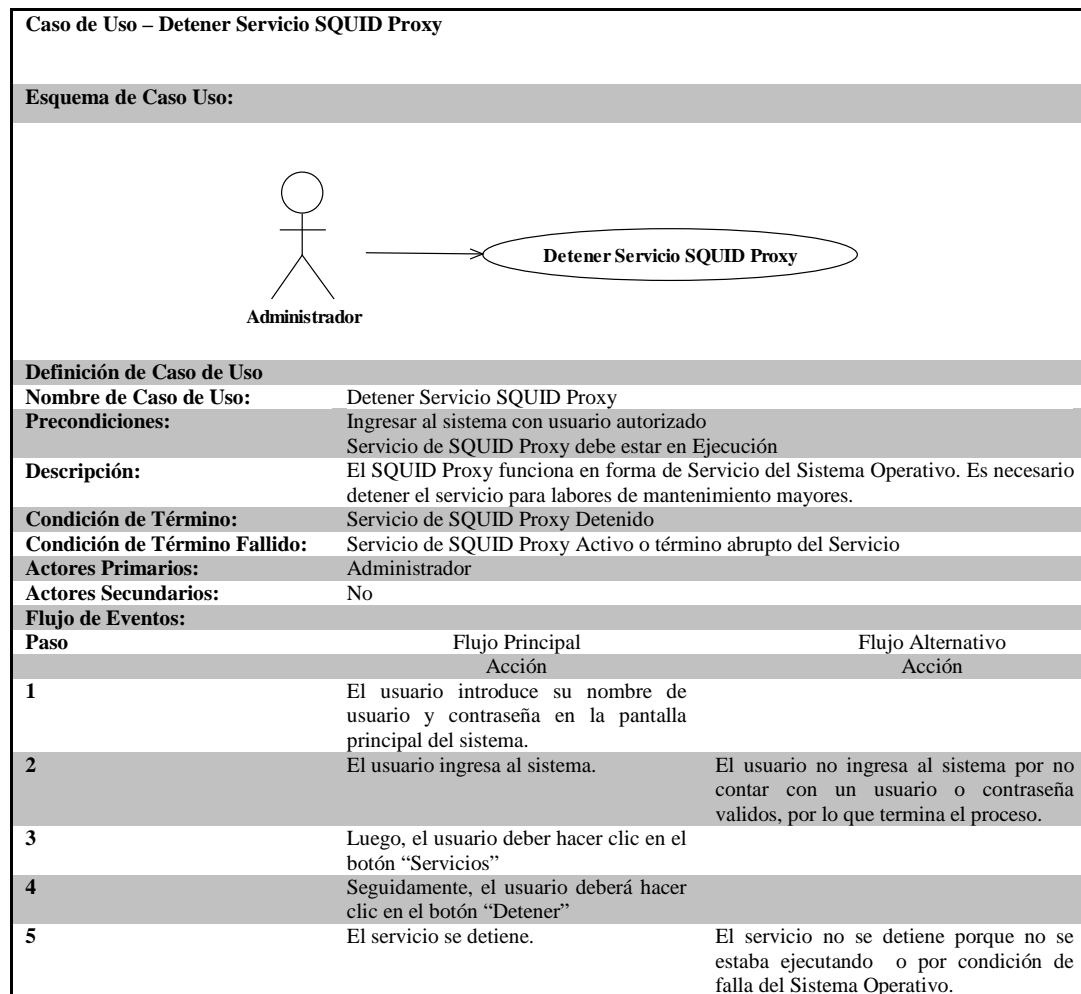


Figura 6. Especificación Caso de Uso Detener Servicio SQUID Proxy

Fuente: Montesino, P y Morales, J (2010)

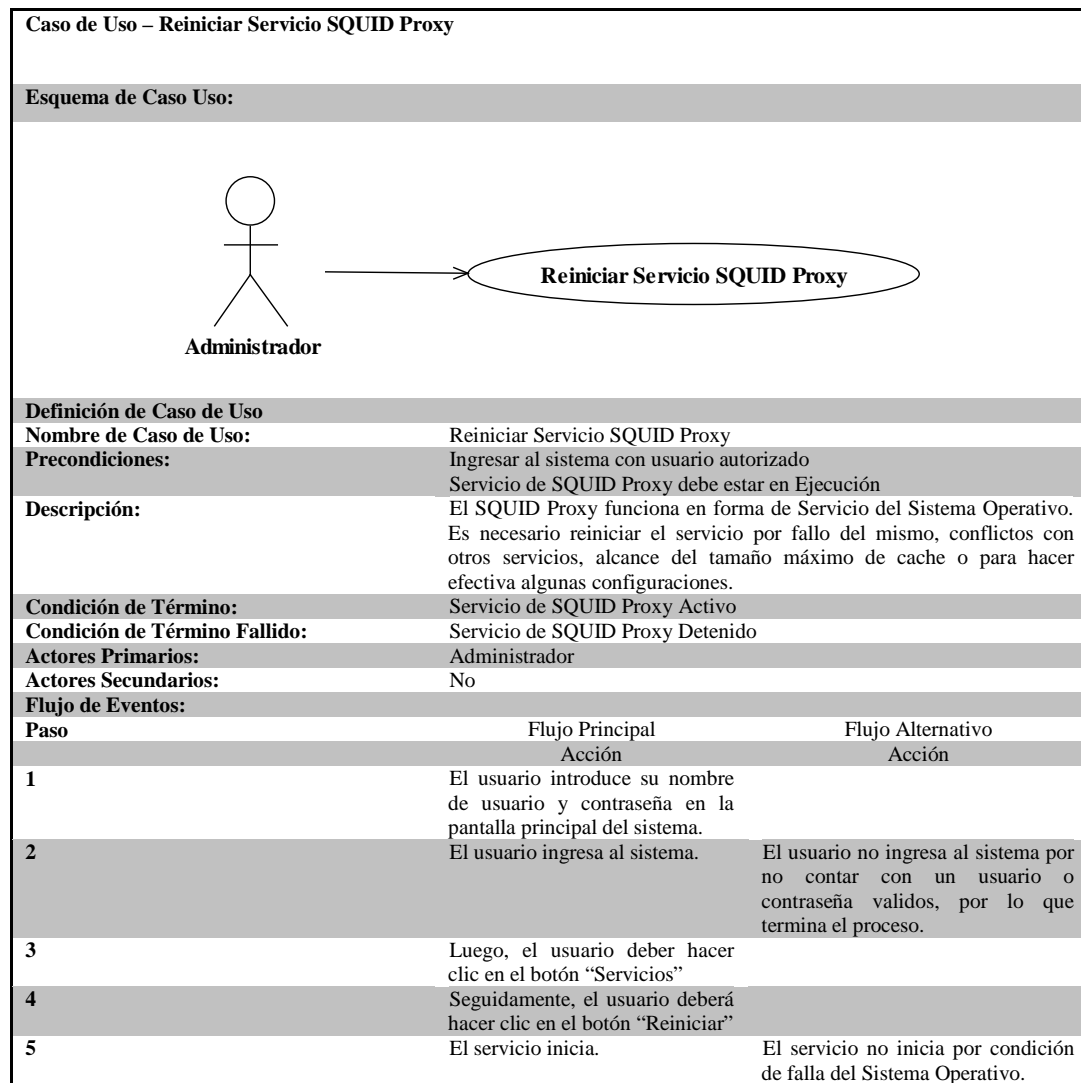


Figura 7. Especificación Caso de Uso Reiniciar Servicio SQUID Proxy

Fuente: Montesino, P y Morales, J (2010)

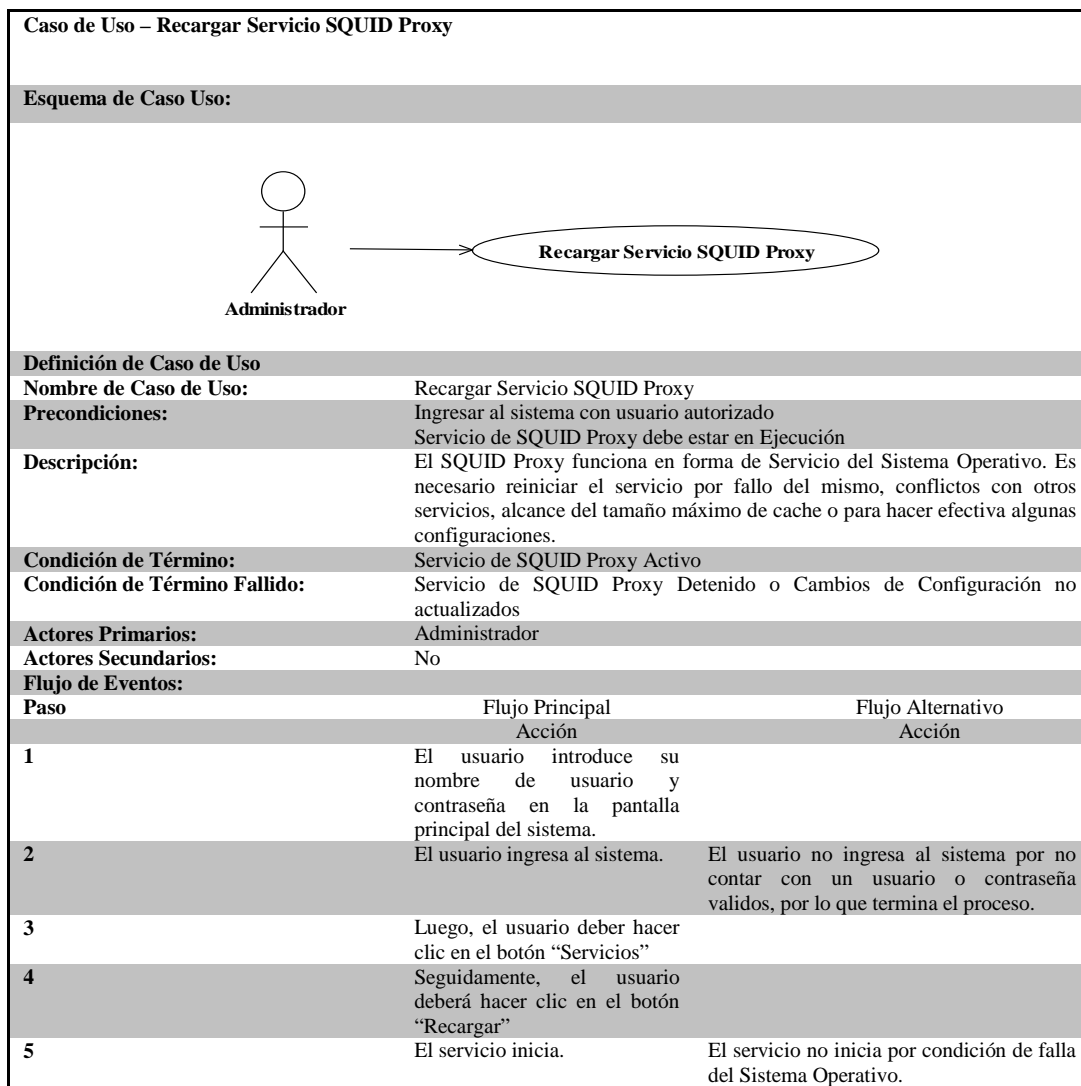


Figura 8. Especificación Caso de Uso Recargar Servicio SQUID Proxy

Fuente: Montesino, P y Morales, J (2010)

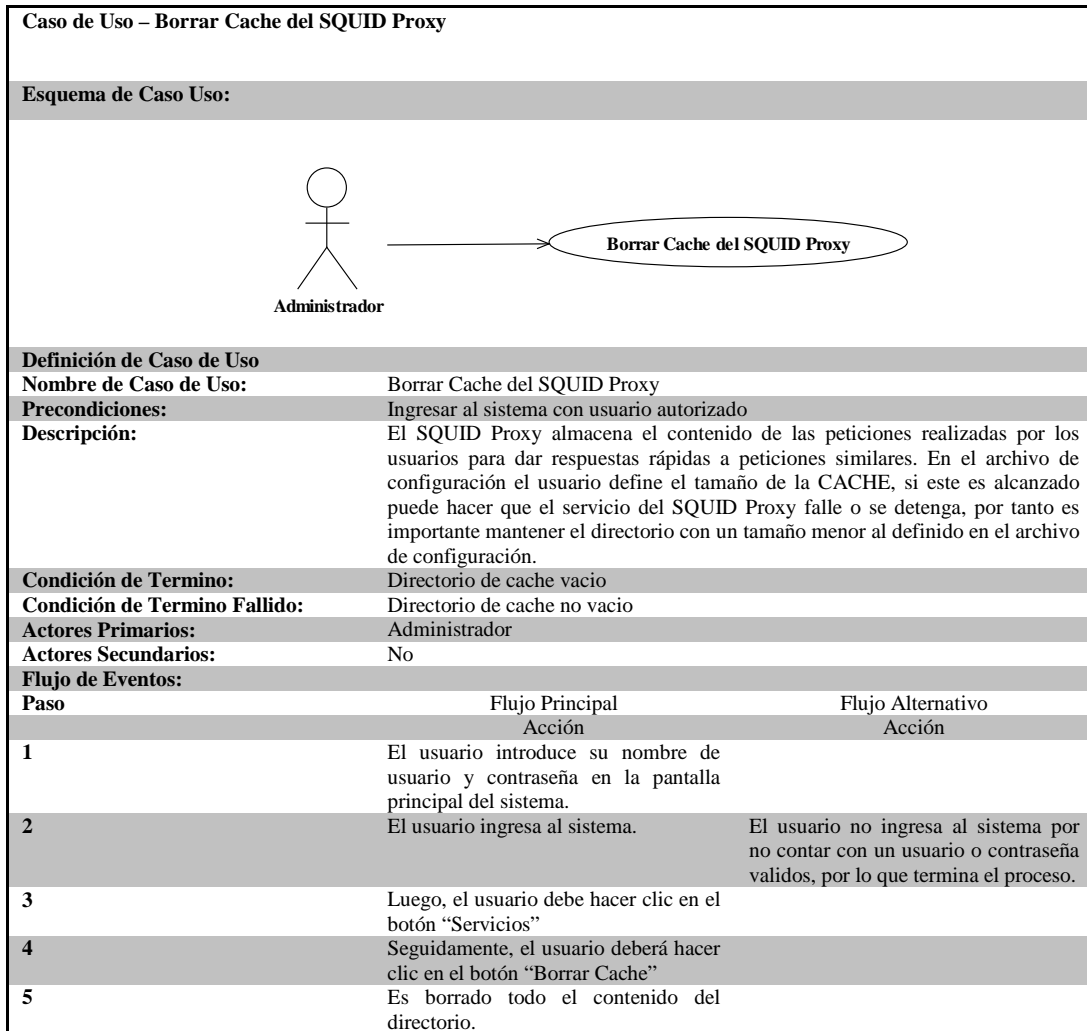


Figura 9. Especificación Caso de Uso Borrar Cache SQUID Proxy

Fuente: Montesino, P y Morales, J (2010)

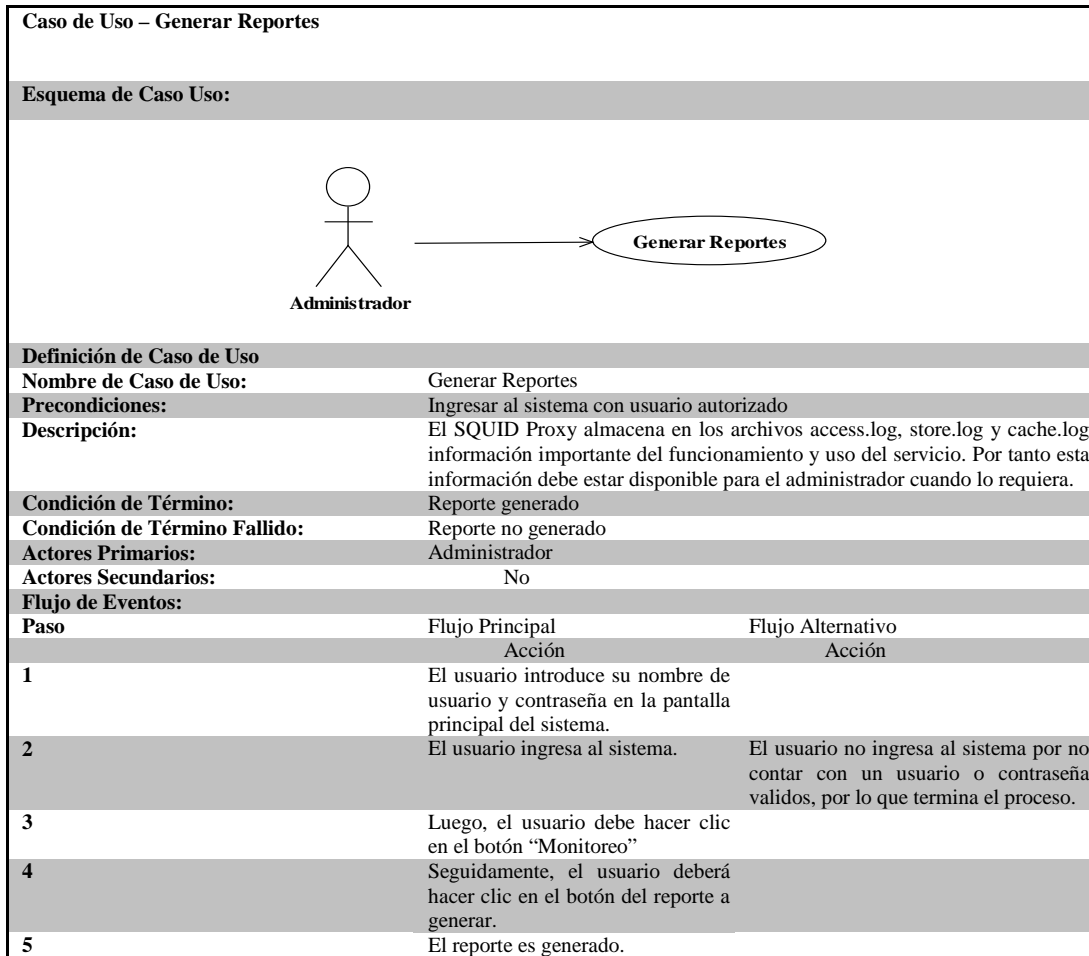


Figura 10. Especificación Caso de Uso Generar Reportes

Fuente: Montesino, P y Morales, J (2010)

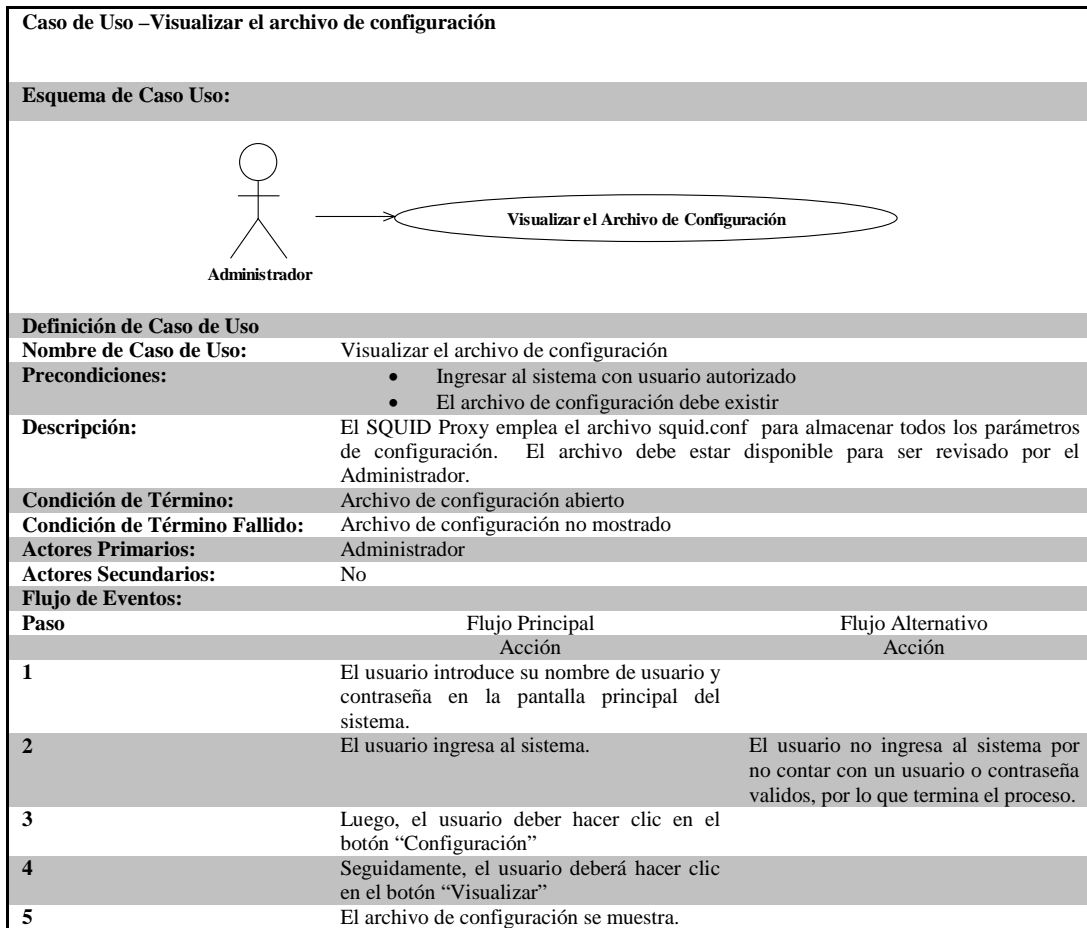


Figura 11. Especificación Caso de Uso Visualizar el archivo de configuración

Fuente: Montesino, P y Morales, J (2010)

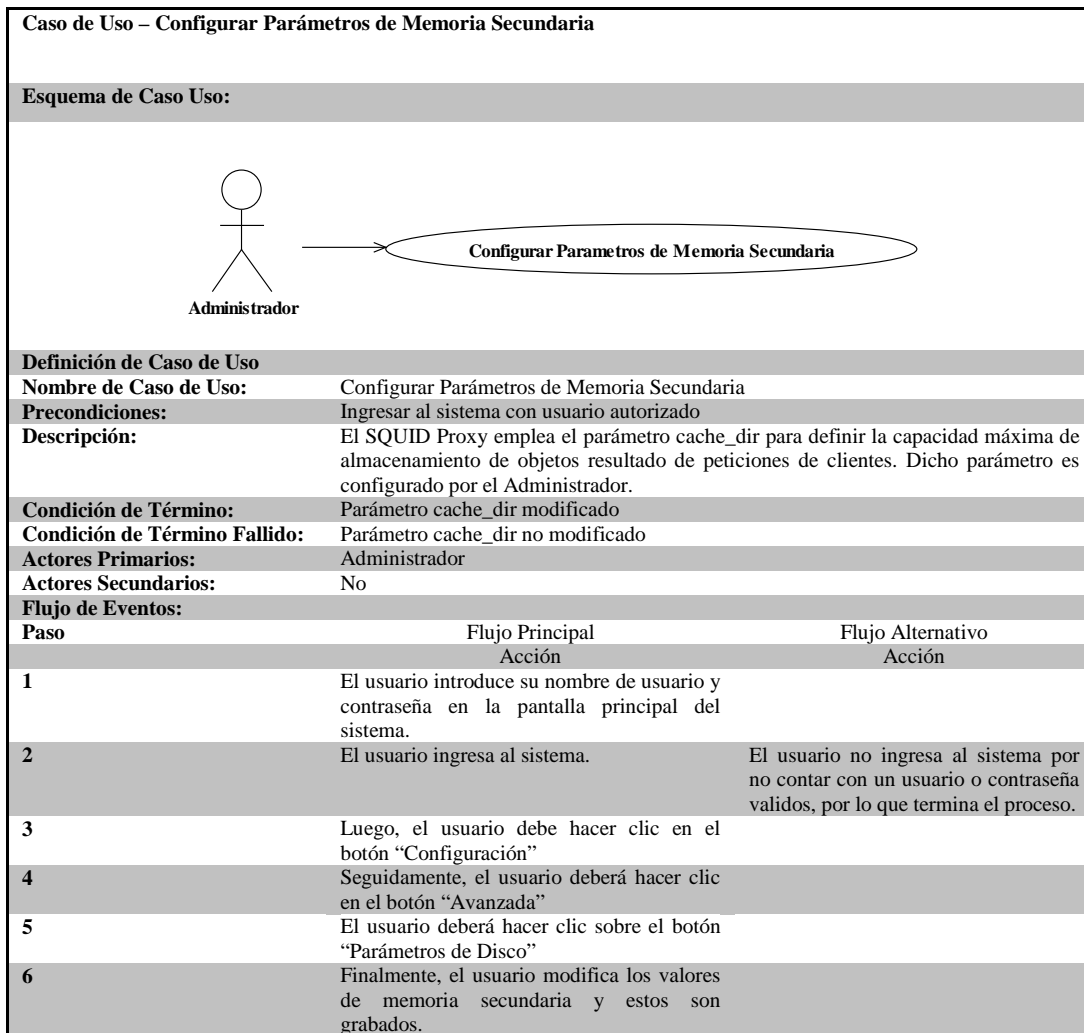


Figura 12. Especificación Caso de Uso Configurar Parámetros de Memoria Secundaria

Fuente: Montesino, P y Morales, J (2010)

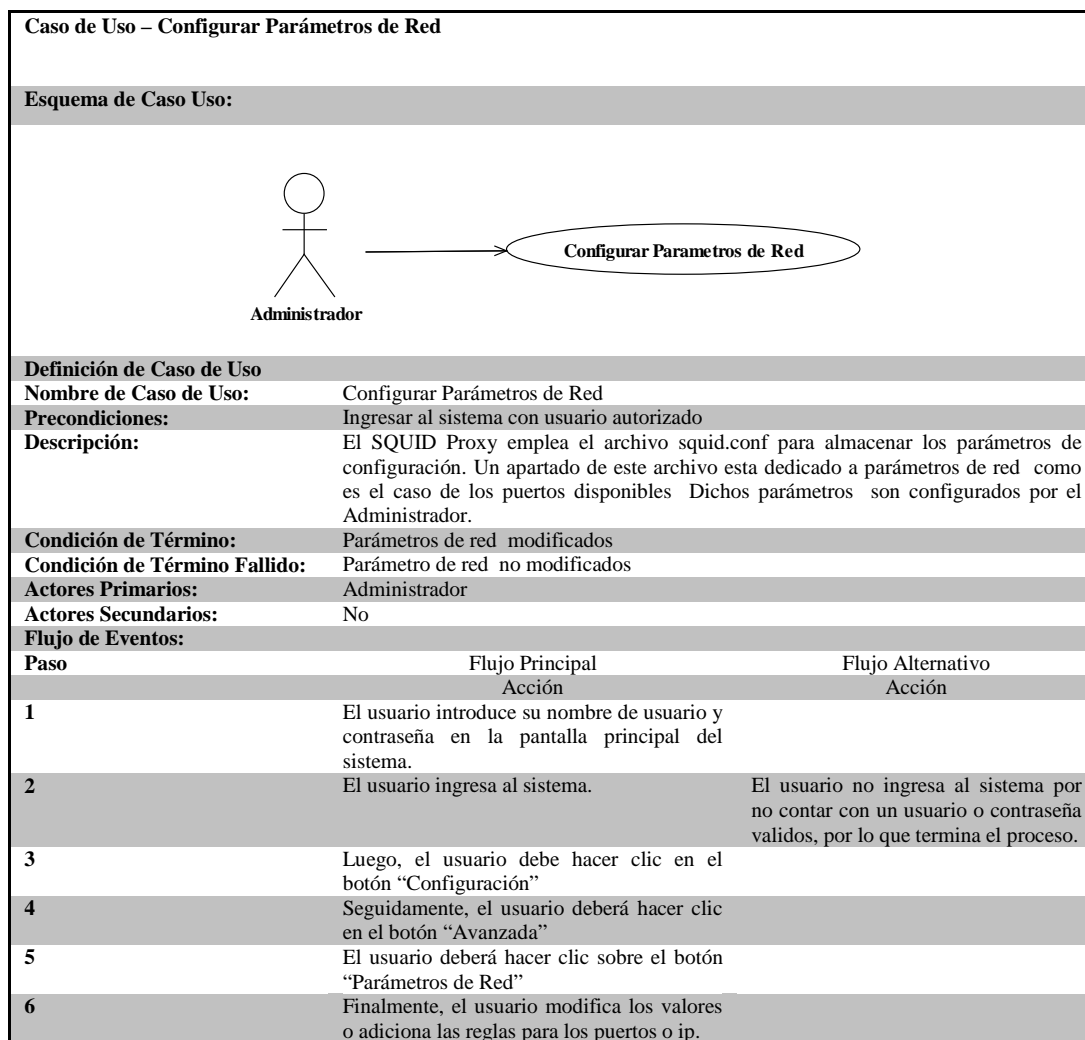


Figura 13. Especificación Caso de Uso Configurar Parámetros de Red

Fuente: Montesino, P y Morales, J (2010)

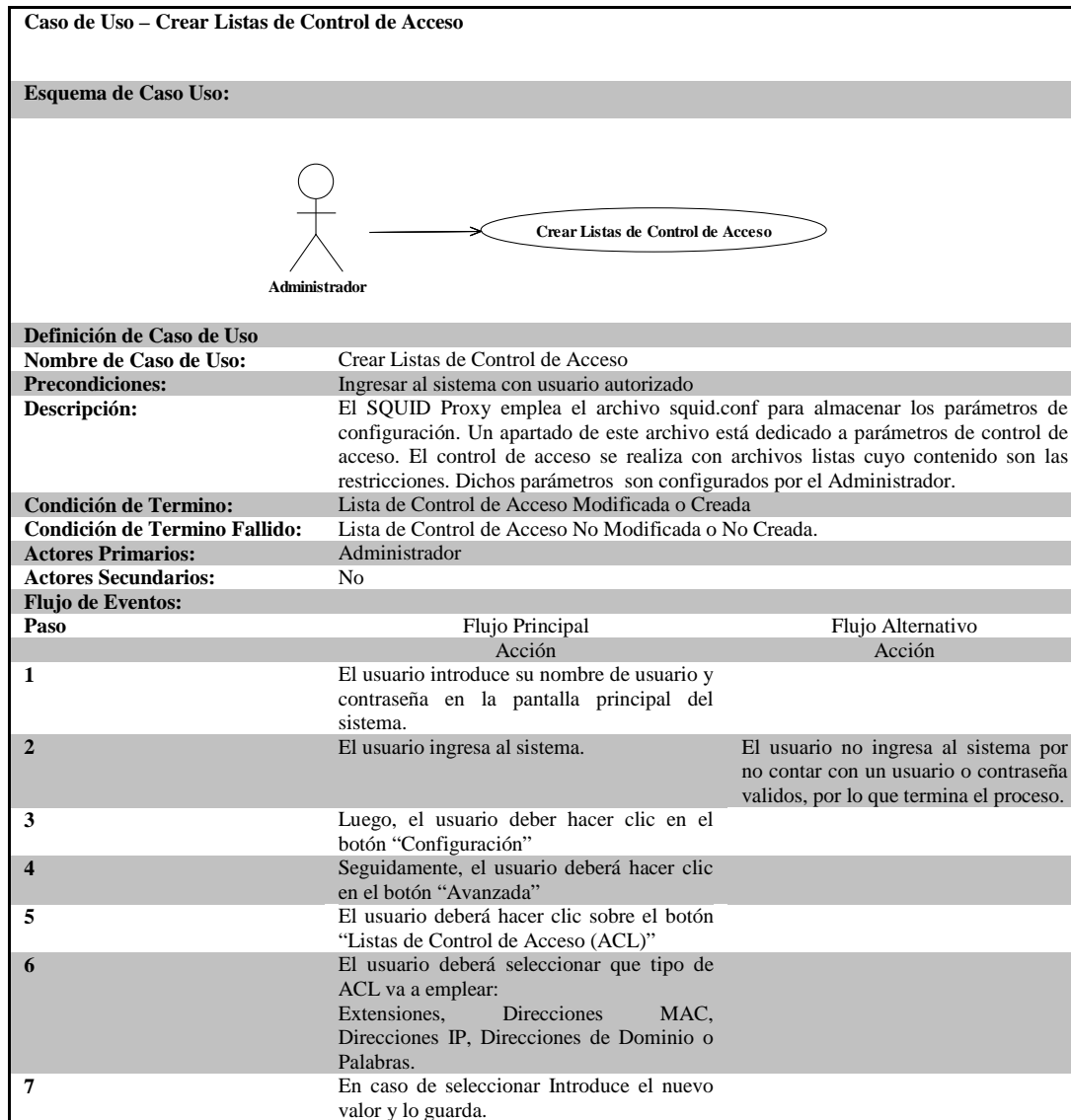


Figura 14. Especificación Caso de Uso Crear Listas de Control de Acceso

Fuente: Montesino, P y Morales, J (2010)

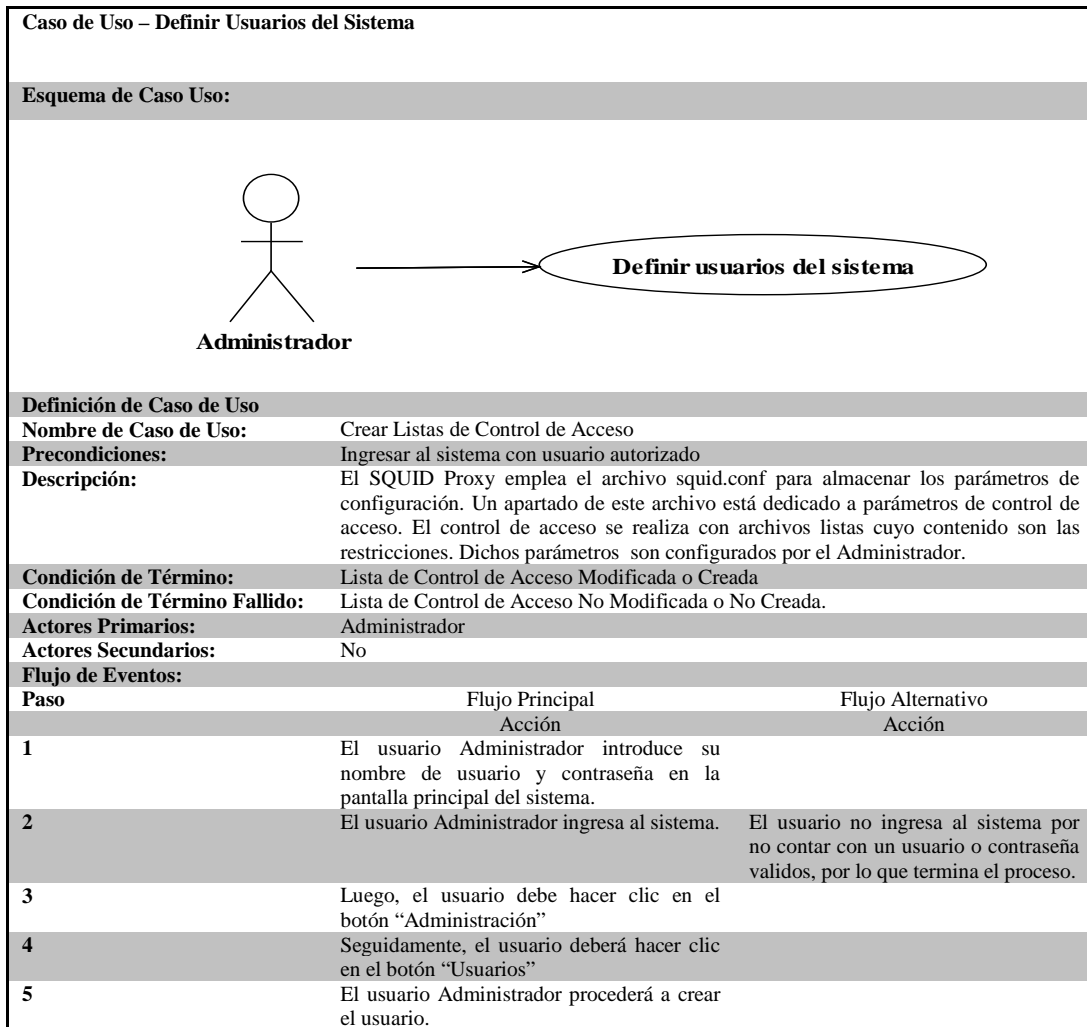


Figura 15. Especificación Caso de Uso Definir Usuarios del Sistema

Fuente: Montesino, P y Morales, J (2010)

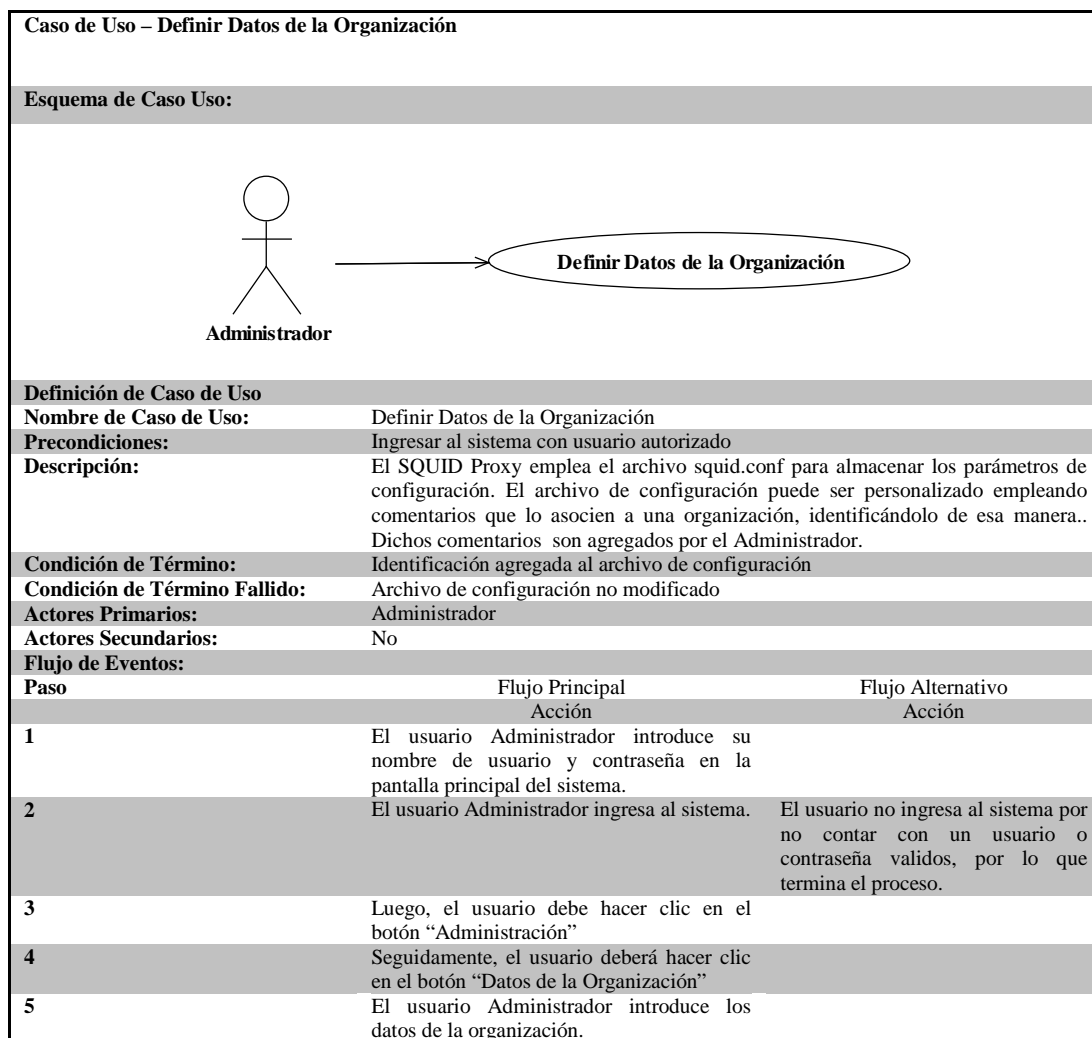


Figura 16. Especificación Caso de Uso Definir Datos de la Organización

Fuente: Montesino, P y Morales, J (2010)

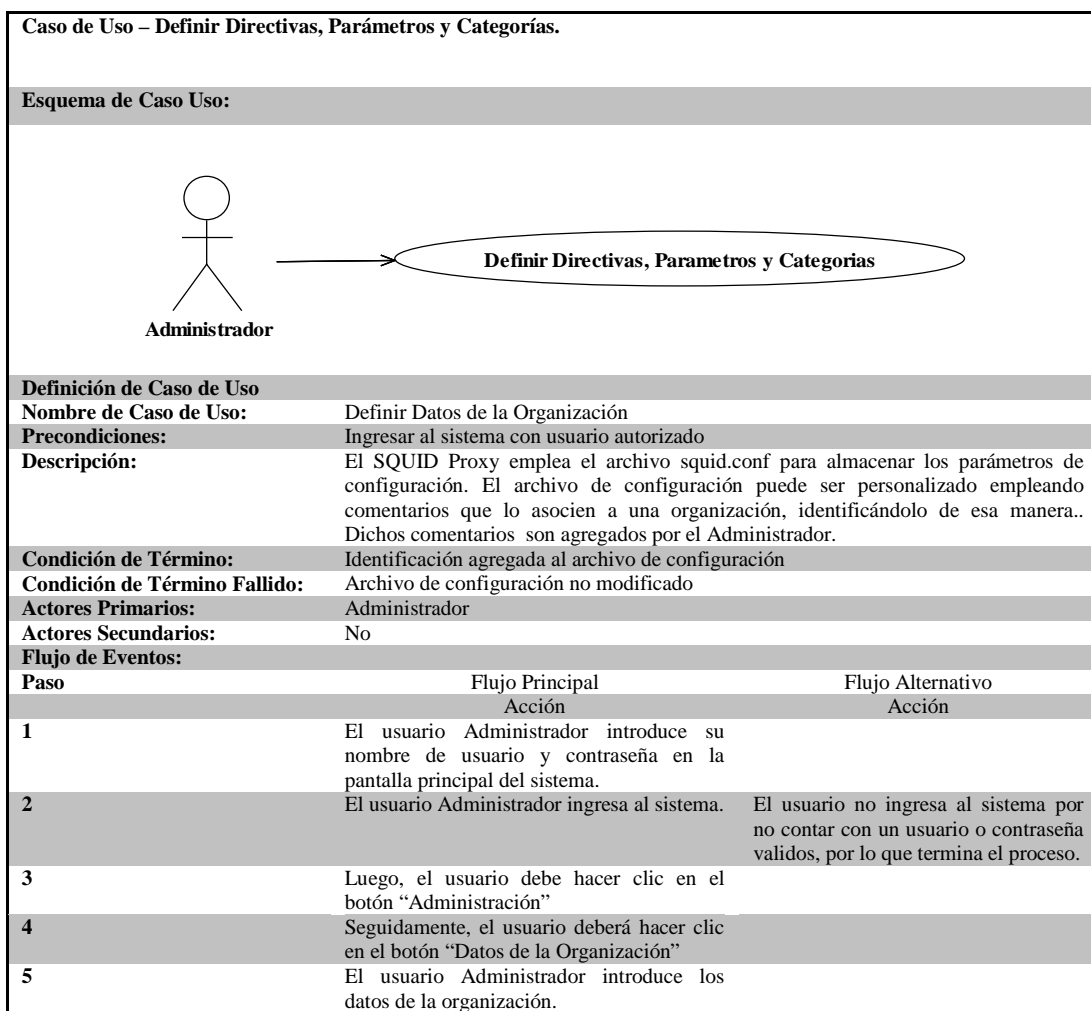


Figura 17. Especificación Caso de Uso Definir Directivas, Parámetros y Categorías.

Fuente: Montesino, P y Morales, J (2010)

Caso de Uso – Iniciar Servicio SQUID Proxy
Esquema de Caso Uso:

Figura 18. Especificación Caso de Uso Iniciar Servicio SQUID Proxy.

Fuente: Montesino, P y Morales, J (2010)

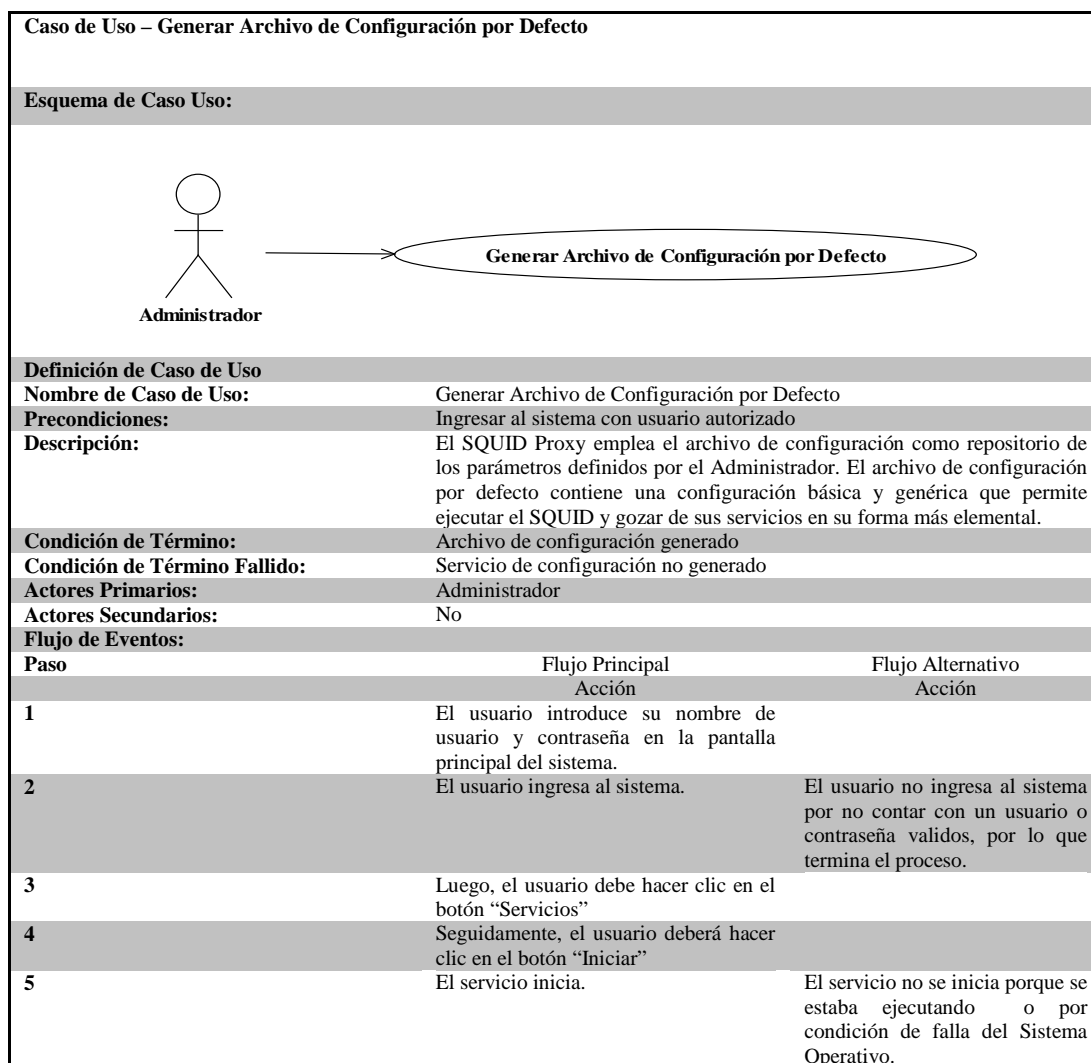


Figura 19. Especificación Caso de Uso Generar Archivo de Configuración por Defecto.

Fuente: Montesino, P y Morales, J (2010)

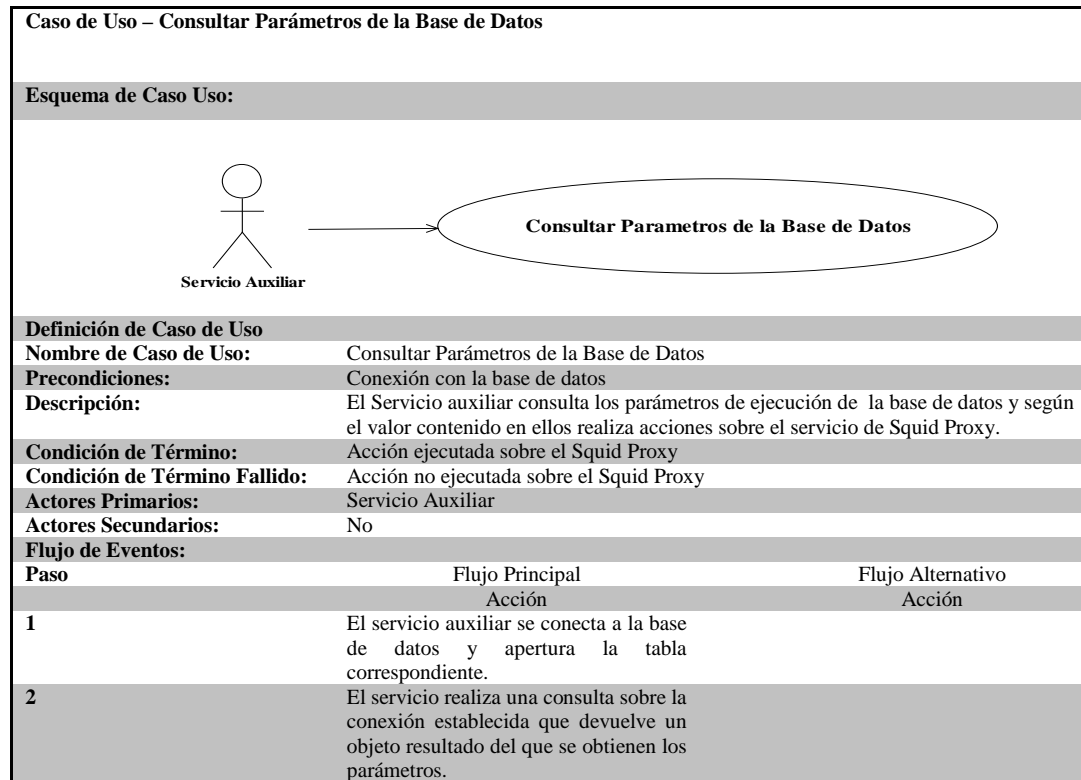


Figura 20. Especificación Caso de Uso Consultar Parámetros de la Base de Datos

Fuente: Montesino, P y Morales, J (2010)

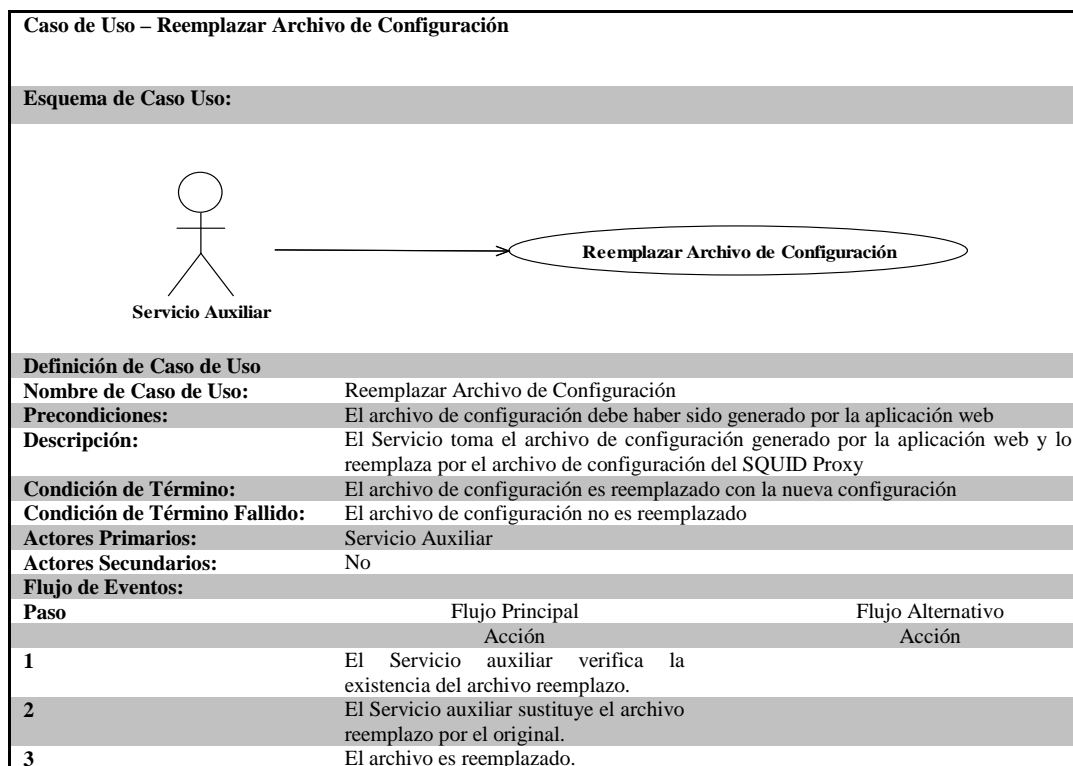


Figura 21. Especificación Caso de Uso Reemplazar Archivo de Configuración

Fuente: Montesino, P y Morales, J (2010)

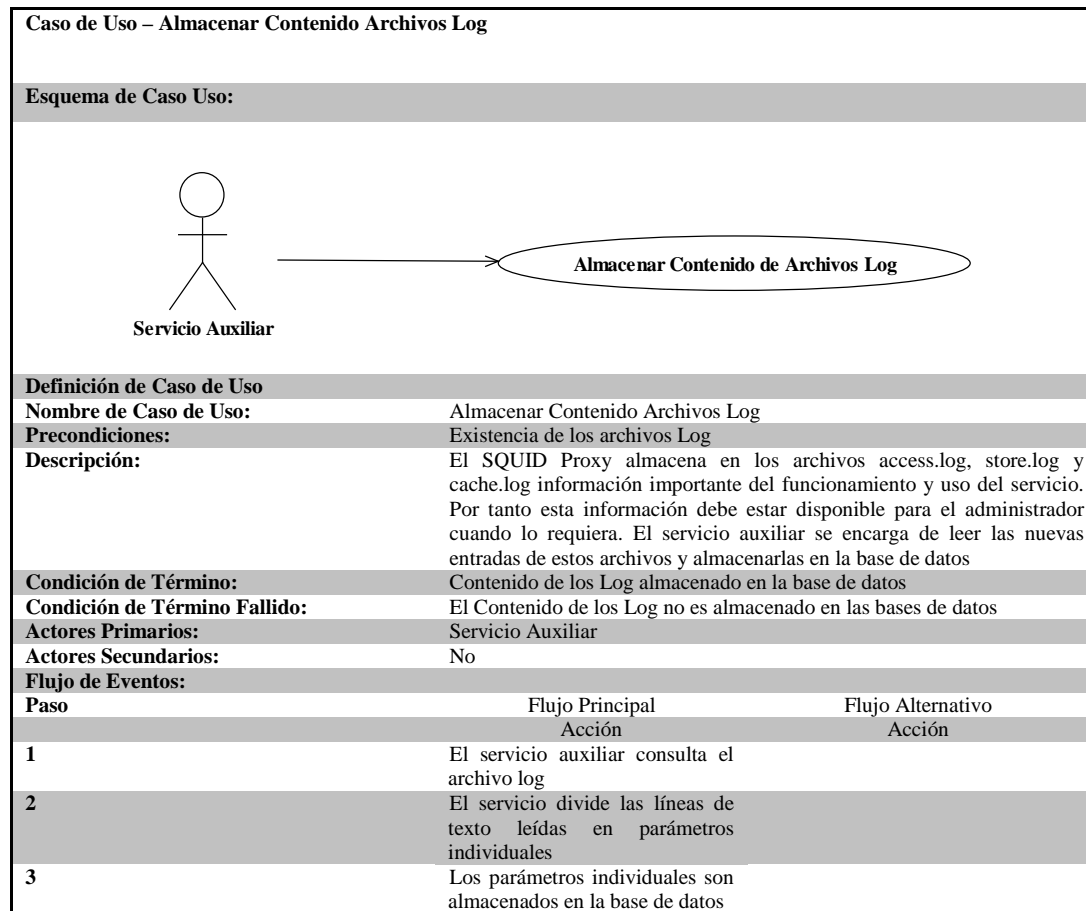


Figura 22. Especificación Caso de Uso Almacenar Contenido de Archivos Log

Fuente: Montesino, P y Morales, J (2010)

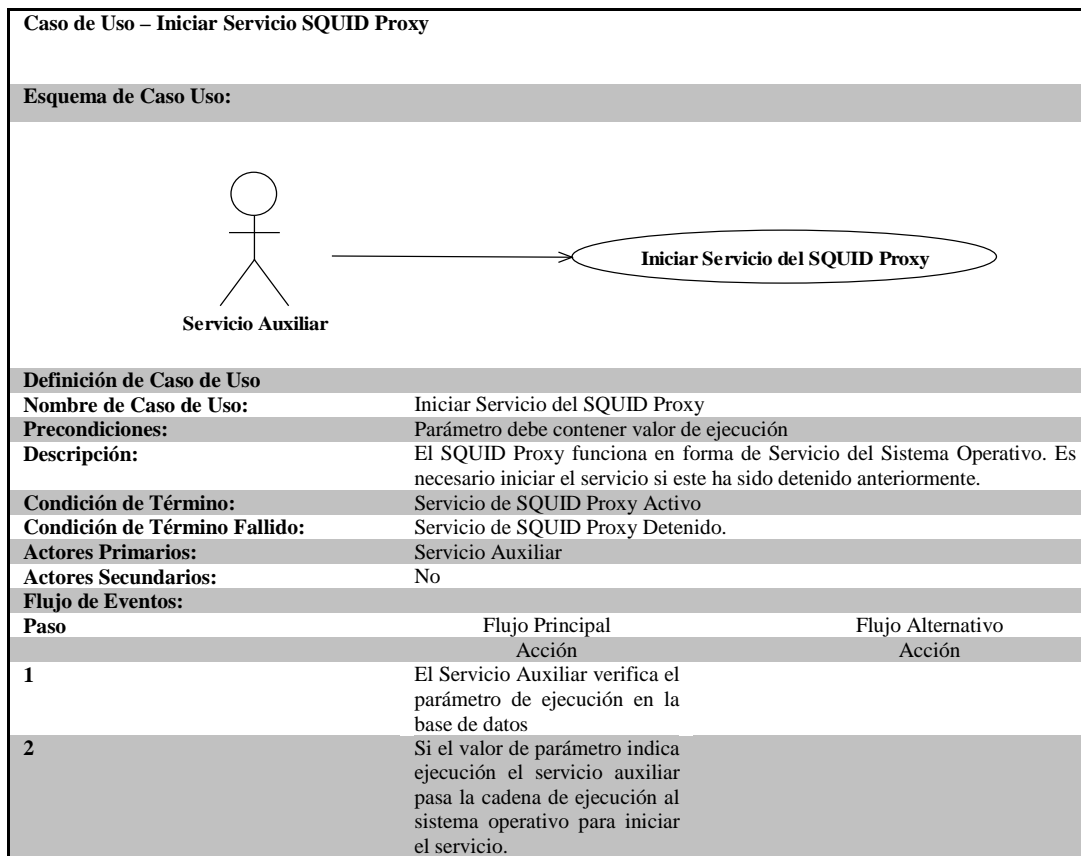


Figura 23. Especificación Caso de Uso Iniciar Servicio SQUID Proxy

Fuente: Montesino, P y Morales, J (2010)

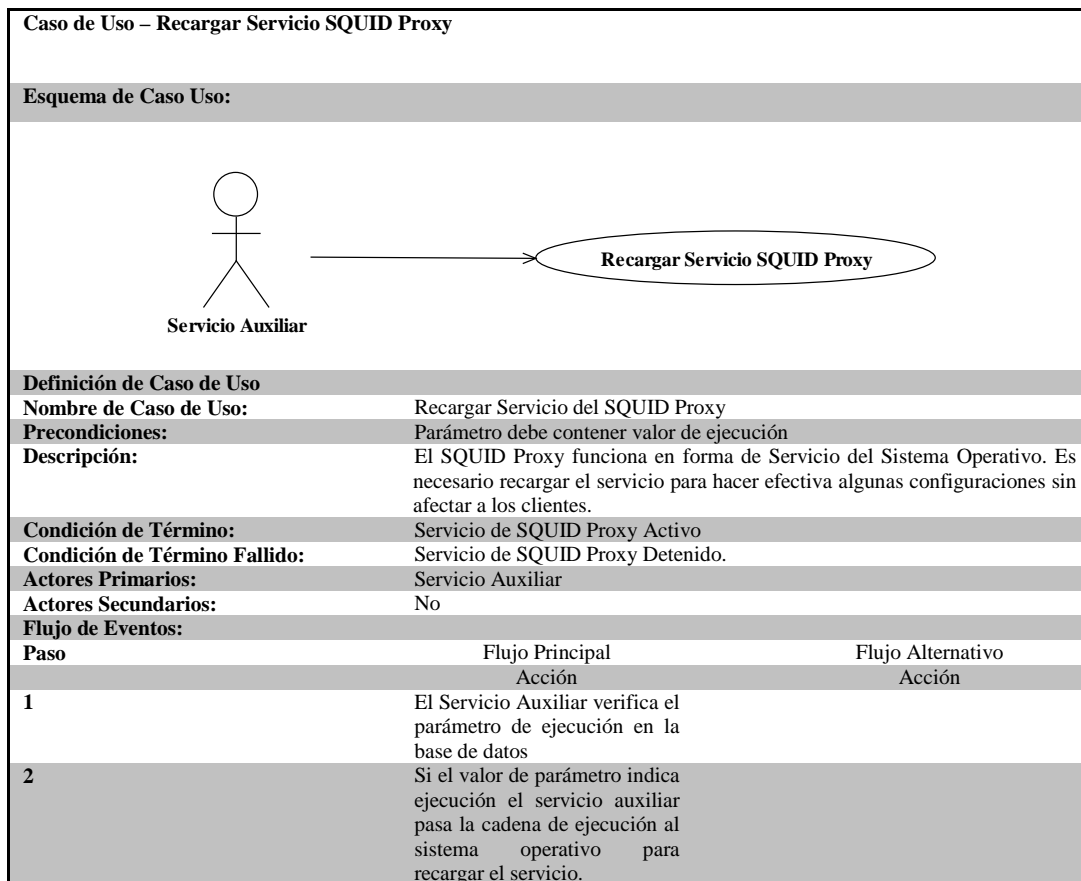


Figura 24. Especificación Caso de Uso Recargar Servicio SQUID Proxy

Fuente: Montesino, P y Morales, J (2010)

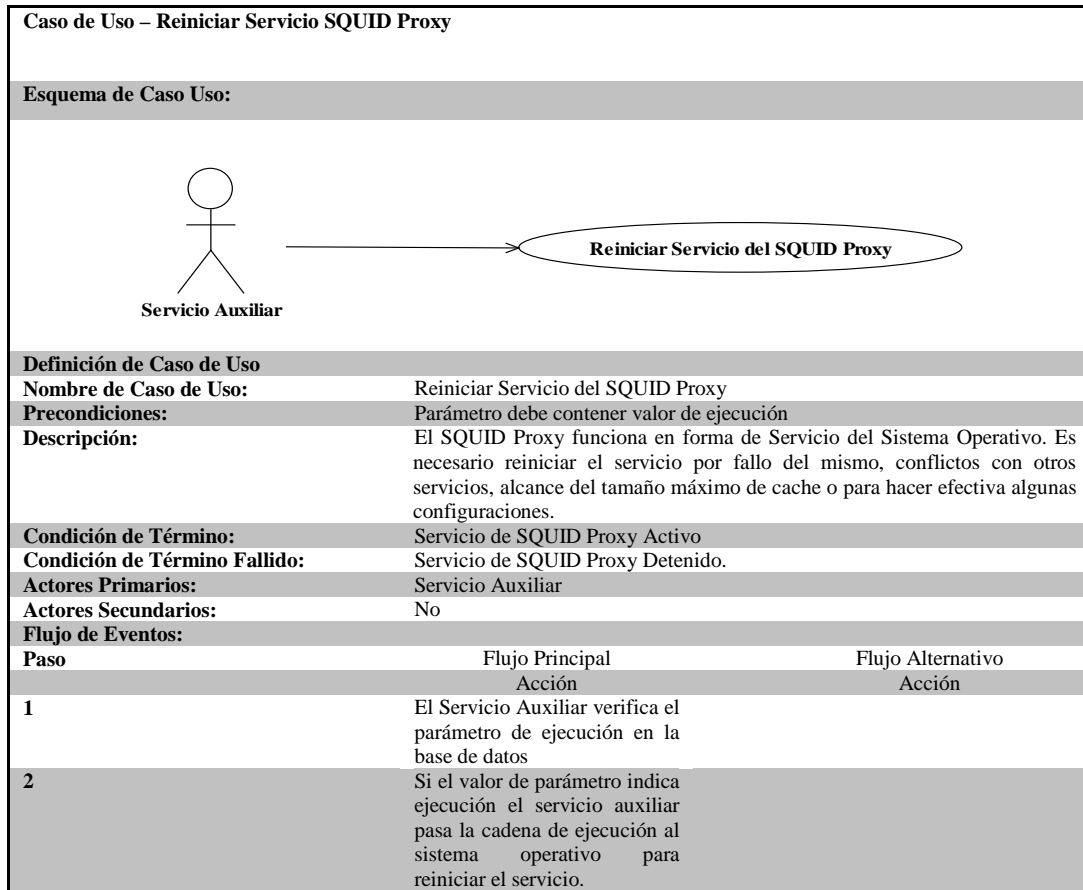


Figura 25. Especificación Caso de Uso Reiniciar Servicio SQUID Proxy

Fuente: Montesino, P y Morales, J (2010)

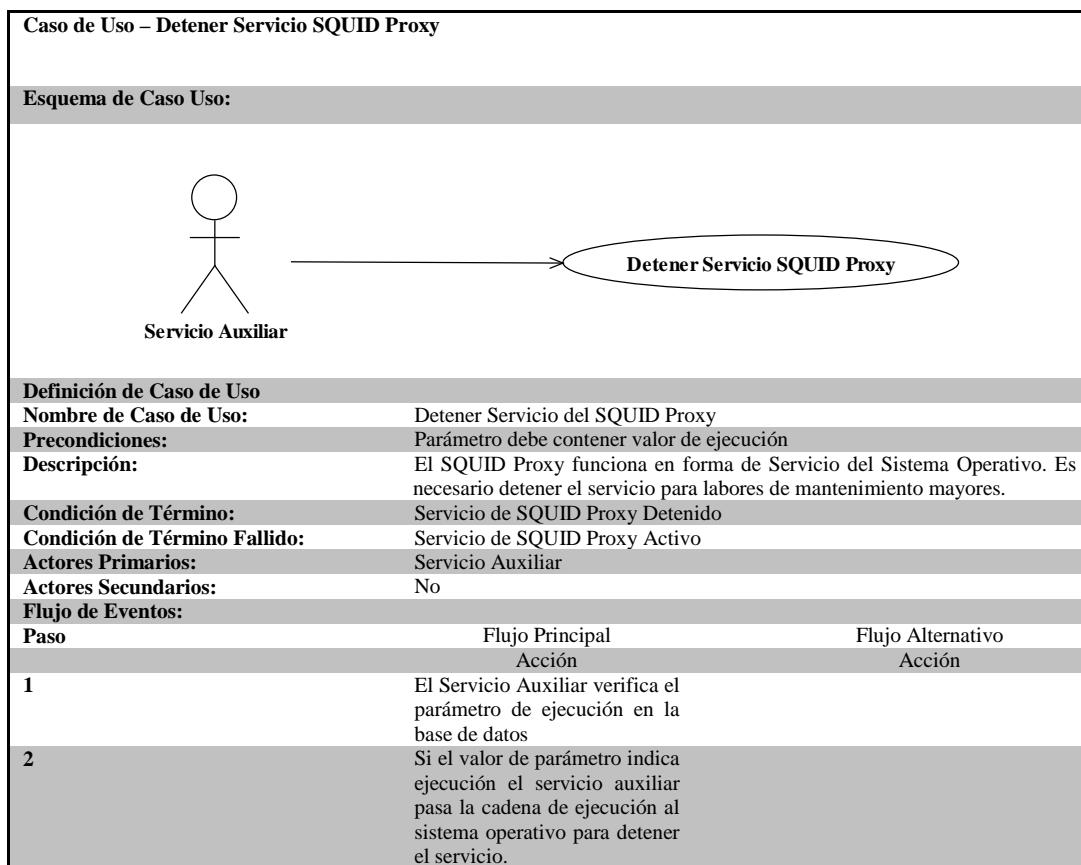


Figura 26. Especificación Caso de Uso Detener Servicio SQUID Proxy

Fuente: Montesino, P y Morales, J (2010)

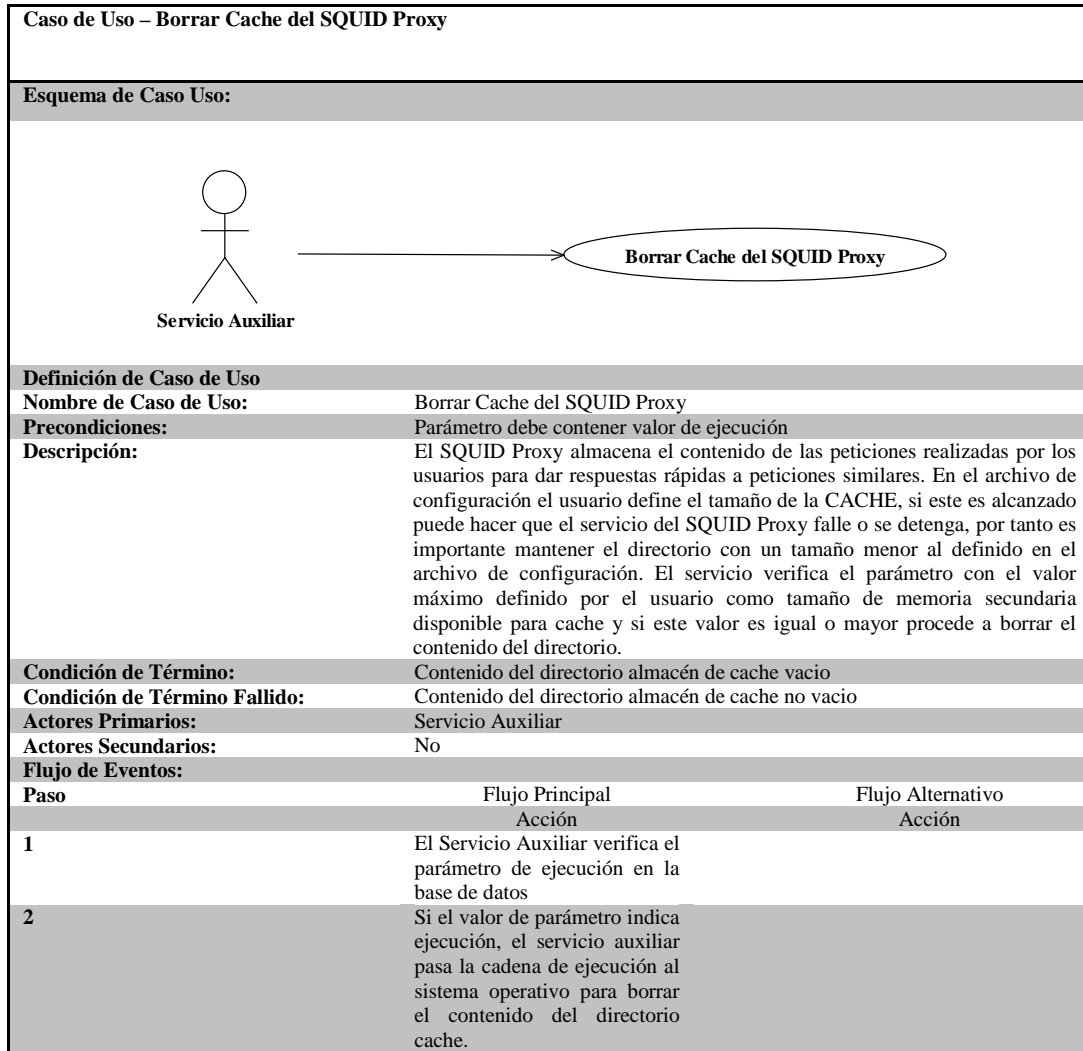
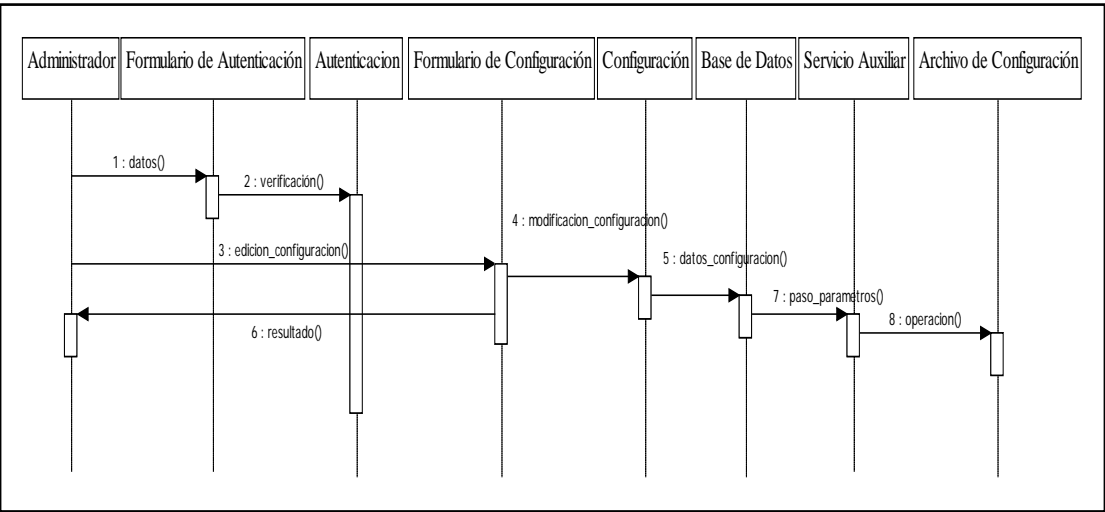


Figura 27. Especificación Caso de Uso Borrar Cache del SQUID Proxy

Fuente: Montesino, P y Morales, J (2010)

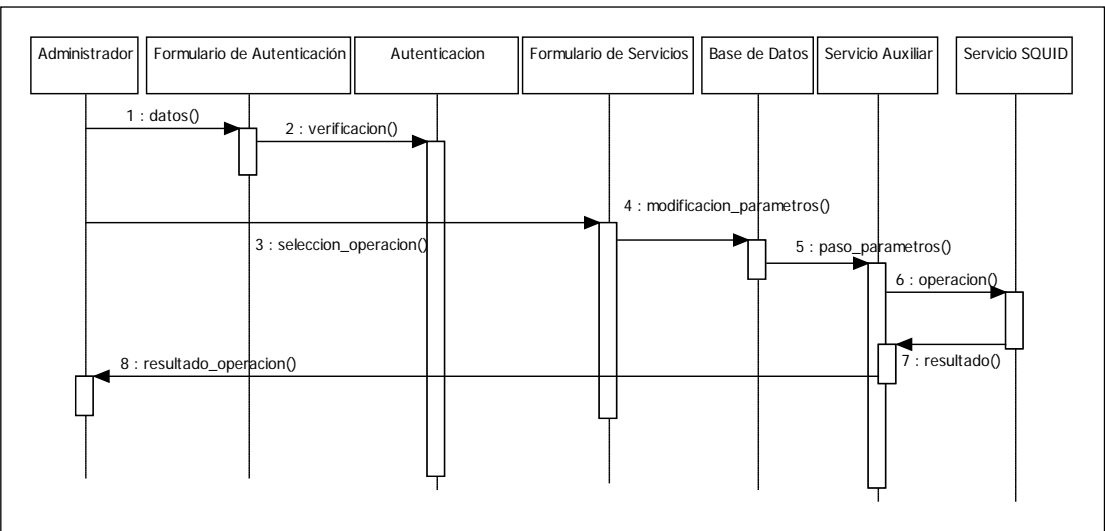
Los siguientes diagramas muestran el orden en que los objetos intercambian mensajes en los escenarios de acción del sistema.

Cuadro 8. Diagrama de Secuencia de Configuración



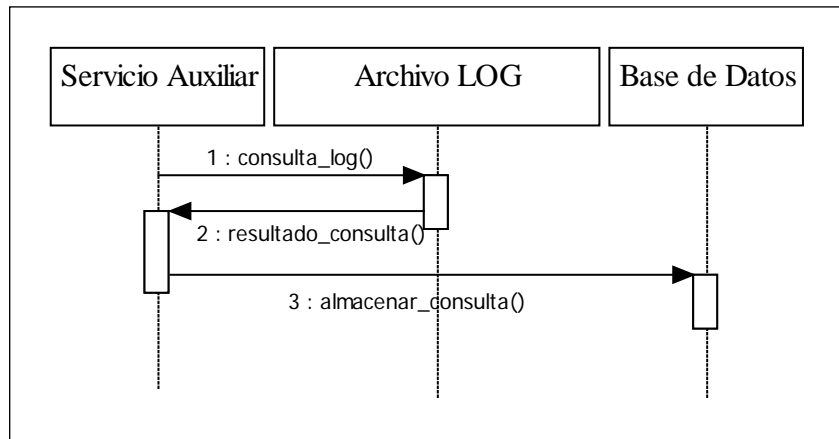
Fuente: Montesino, P y Morales, J (2010)

Cuadro 9. Diagrama de Secuencia de Servicio



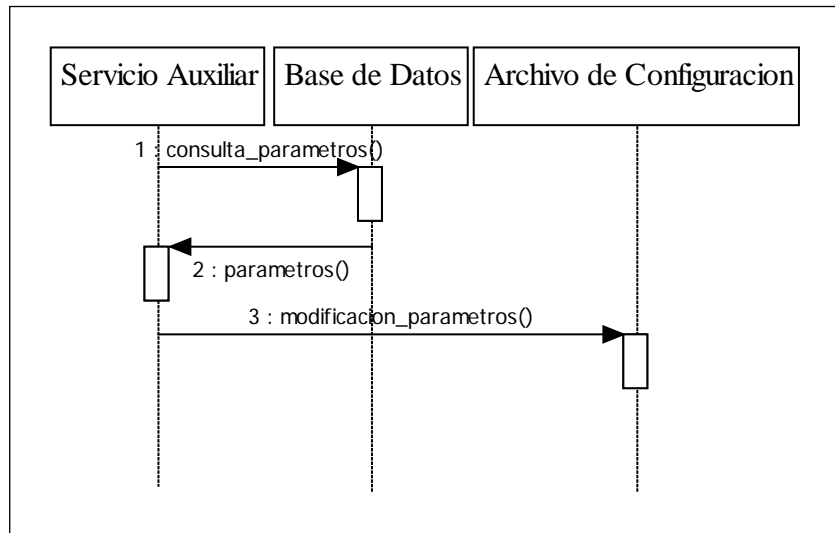
Fuente: Montesino, P y Morales, J (2010)

Cuadro 10. Diagrama de Secuencia de Servicio Auxiliar Archivos Log



Fuente: Montesino, P y Morales, J (2010)

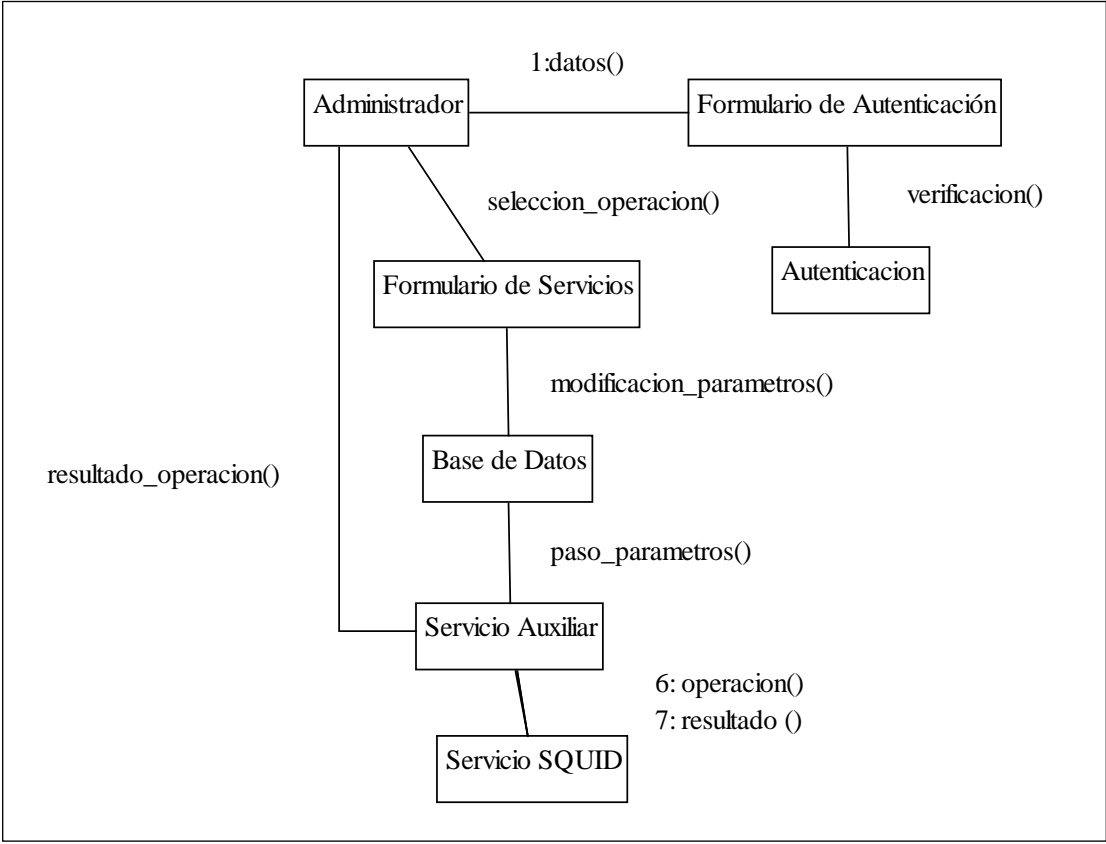
Cuadro 11. Diagrama de Secuencia de Servicio Auxiliar Archivo de Configuración



Fuente: Montesino, P y Morales, J (2010)

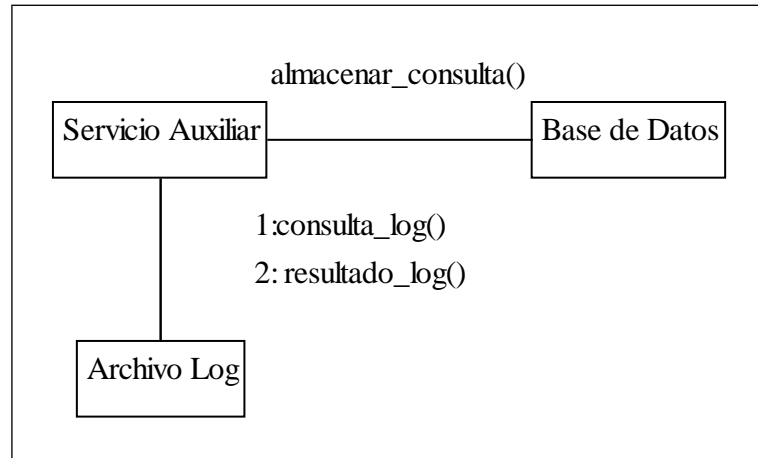
Los siguientes diagramas muestran los mensajes que intercambian los objetos en los diferentes escenarios de acción del sistema.

Cuadro 12. Diagrama de Colaboración de Servicio



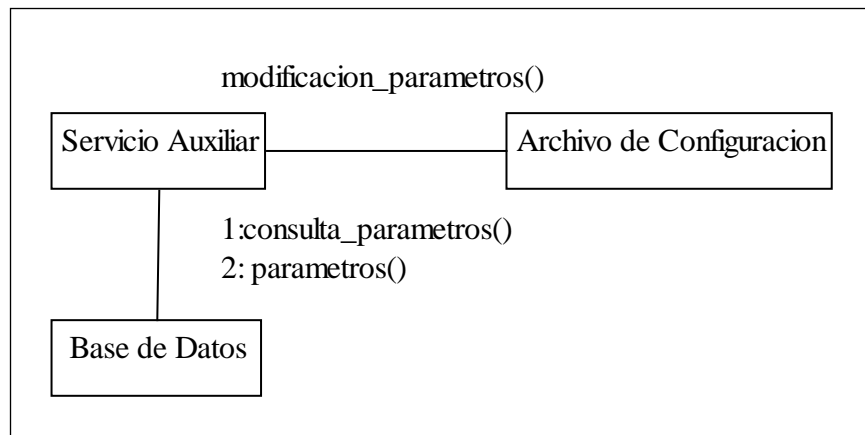
Fuente: Montesino, P y Morales, J (2010)

Cuadro 13. Diagrama de Colaboración Archivo de Configuración



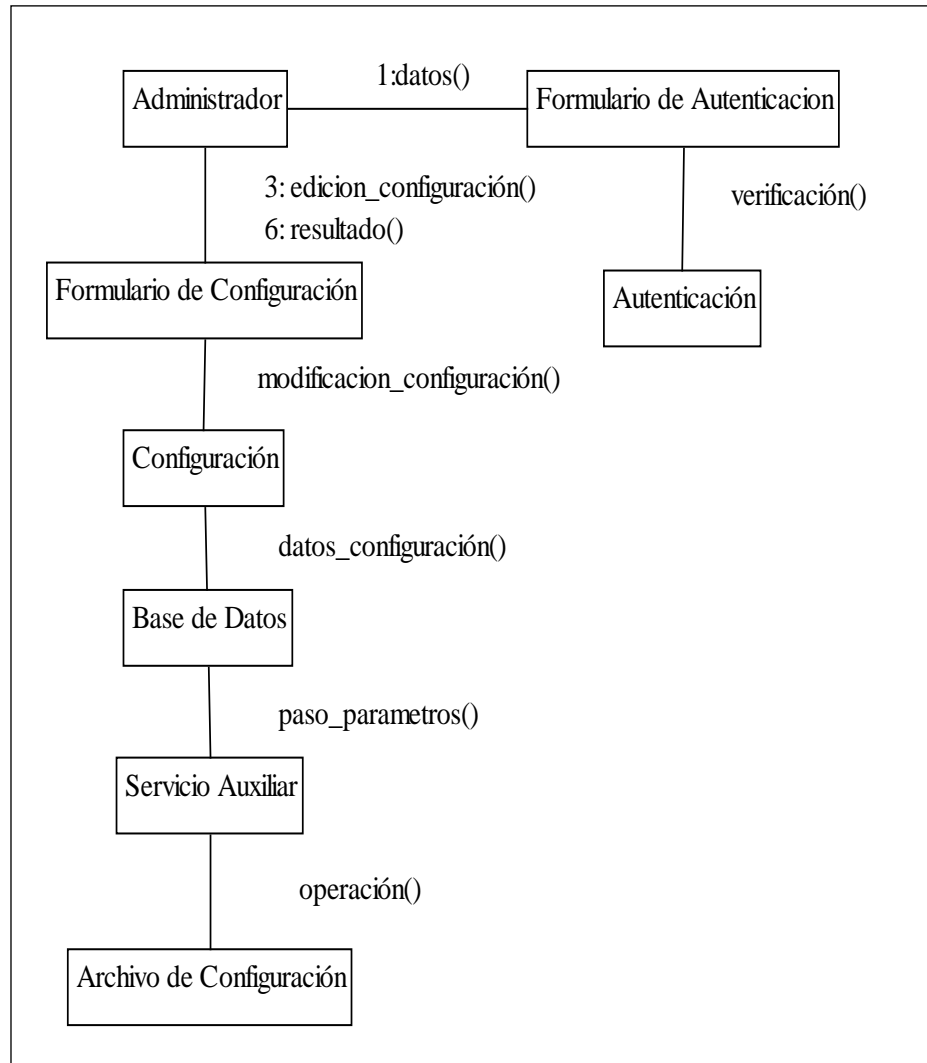
Fuente: Montesino, P y Morales, J (2010)

Cuadro 14. Diagrama de Colaboración Servicio Auxiliar Archivos Log



Fuente: Montesino, P y Morales, J (2010)

Cuadro 15. Diagrama de Colaboración Administrador Archivo de Configuración



Fuente: Montesino, P y Morales, J (2010)

La siguiente figura muestra el modelo de datos del sistema. La base de datos y la relación entre las tablas que la componen.

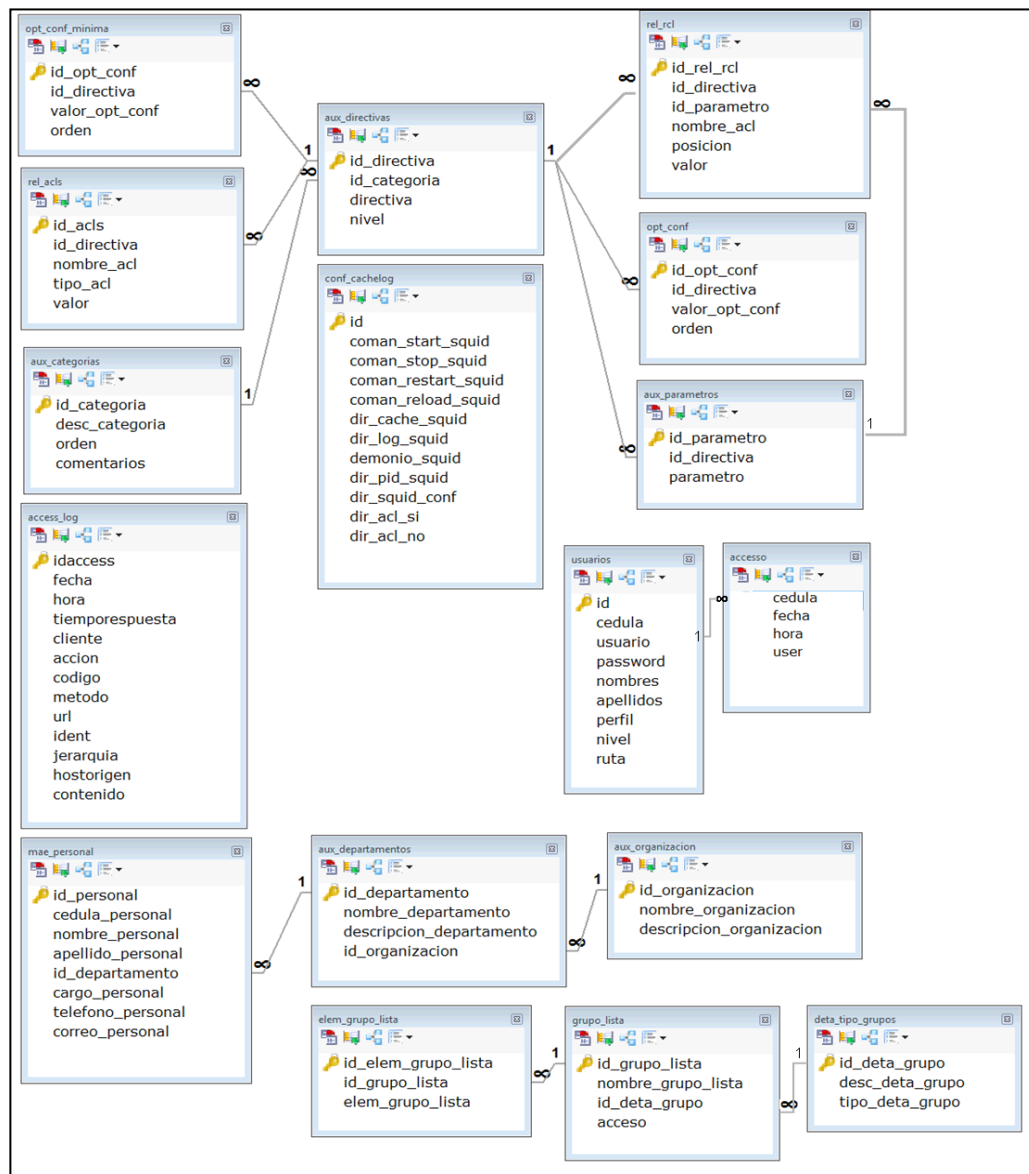


Figura 28. Modelo de Datos

Fuente: Montesino, P y Morales, J (2010)

4.3. Fase III: Construcción de Interfaz y Módulos

En esta etapa se construyeron cada uno de los módulos y la interfaz de la aplicación fundamentando su creación a partir de los diseños propuestos y tomando en cuenta el alcance previamente analizado en etapas anteriores de la investigación y se obtuvieron los siguientes modelos y diagramas:

- a) Pantallas del Sistema
- b) Diagrama de Arquitectura Final
- c) Diagrama de Despliegue

Las siguientes figuras muestran las diferentes pantallas del sistema.



Figura 29. Pantalla Autenticación.

Fuente: Montesino, P y Morales, J (2010)



Figura 30. Pantalla de Bienvenida
Fuente: Montesino, P y Morales, J (2010)



Figura 31. Pantalla de Servicios
Fuente: Montesino, P y Morales, J (2010)



Figura 32. Selección de Configuración
Fuente: Montesino, P y Morales, J (2010)



Figura 33. Pantalla de Configuración Avanzada
Fuente: Montesino, P y Morales, J (2010)



Figura 34. Configuración de Puertos
Fuente: Montesino, P y Morales, J (2010)

Cuadro 50. Parámetros de Cache



Figura 35. Parámetros de Cache
Fuente: Montesino, P y Morales, J (2010)



Figura 36. Configuración de Directorios Log

Fuente: Montesino, P y Morales, J (2010)



Figura 37. Lista de Grupos

Fuente: Montesino, P y Morales, J (2010)



Figura 38. Reglas de Lista de Acceso
Fuente: Montesino, P y Morales, J (2010)



Figura 39. Lista de Control de Acceso
Fuente: Montesino, P y Morales, J (2010)

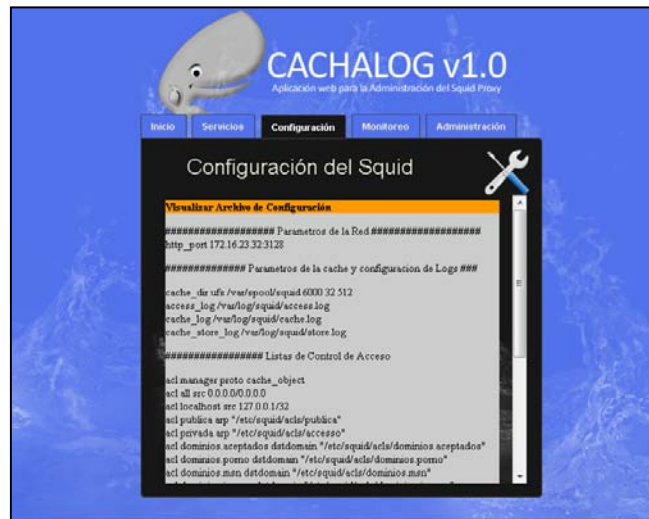


Figura 40. Pantalla Visualización Archivo de Configuración

Fuente: Montesino, P y Morales, J (2010)



Figura 41. Pantalla de Monitoreo

Fuente: Montesino, P y Morales, J (2010)



Figura 42. Sitio Más Visitado por Clientes
Fuente: Montesino, P y Morales, J (2010)



Figura 43. Sitios Más Visitados
Fuente: Montesino, P y Morales, J (2010)



Figura 44. Sitios No Permitidos

Fuente: Montesino, P y Morales, J (2010)



Figura 45. Administración del Sistema

Fuente: Montesino, P y Morales, J (2010)



Figura 46. Pantalla Configuración General

Fuente: Montesino, P y Morales, J (2010)



Figura 47. Datos de Ejecución

Fuente: Montesino, P y Morales, J (2010)



Figura 48. Pantalla de Rutas Generales
Fuente: Montesino, P y Morales, J (2010)



Figura 49. Menú Datos de la Organización
Fuente: Montesino, P y Morales, J (2010)



Figura 50. Datos de la Organización

Fuente: Montesino, P y Morales, J (2010)



Figura 51. Creación de Departamentos

Fuente: Montesino, P y Morales, J (2010)



Figura 52. Creación de Personal
Fuente: Montesino, P y Morales, J (2010)



Figura 53. Creación de Usuarios del Sistema
Fuente: Montesino, P y Morales, J (2010)



Figura 54. Tablas Auxiliares
Fuente: Montesino, P y Morales, J (2010)



Figura 55. Creación de Categorías
Fuente: Montesino, P y Morales, J (2010)



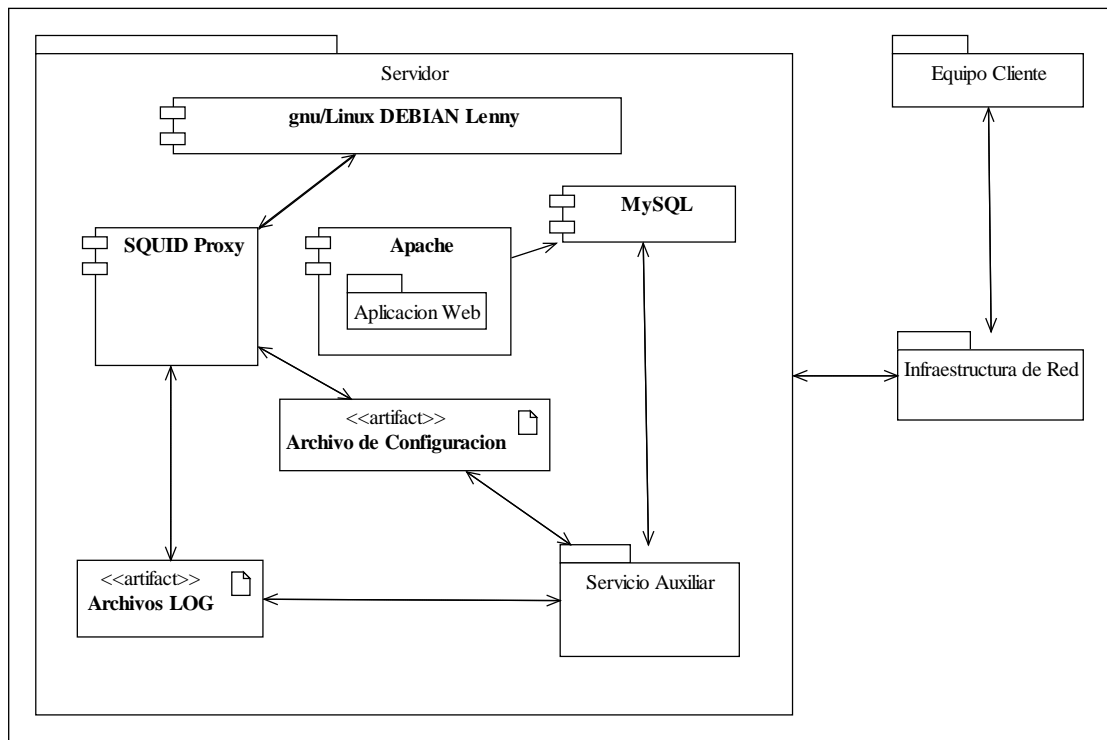
Figura 56. Creación de Directivas
Fuente: Montesino, P y Morales, J (2010)



Figura 57. Creación de Parámetros
Fuente: Montesino, P y Morales, J (2010)

El diagrama de la arquitectura final nos muestra los objetos representativos y sus interrelaciones en el sistema desarrollado.

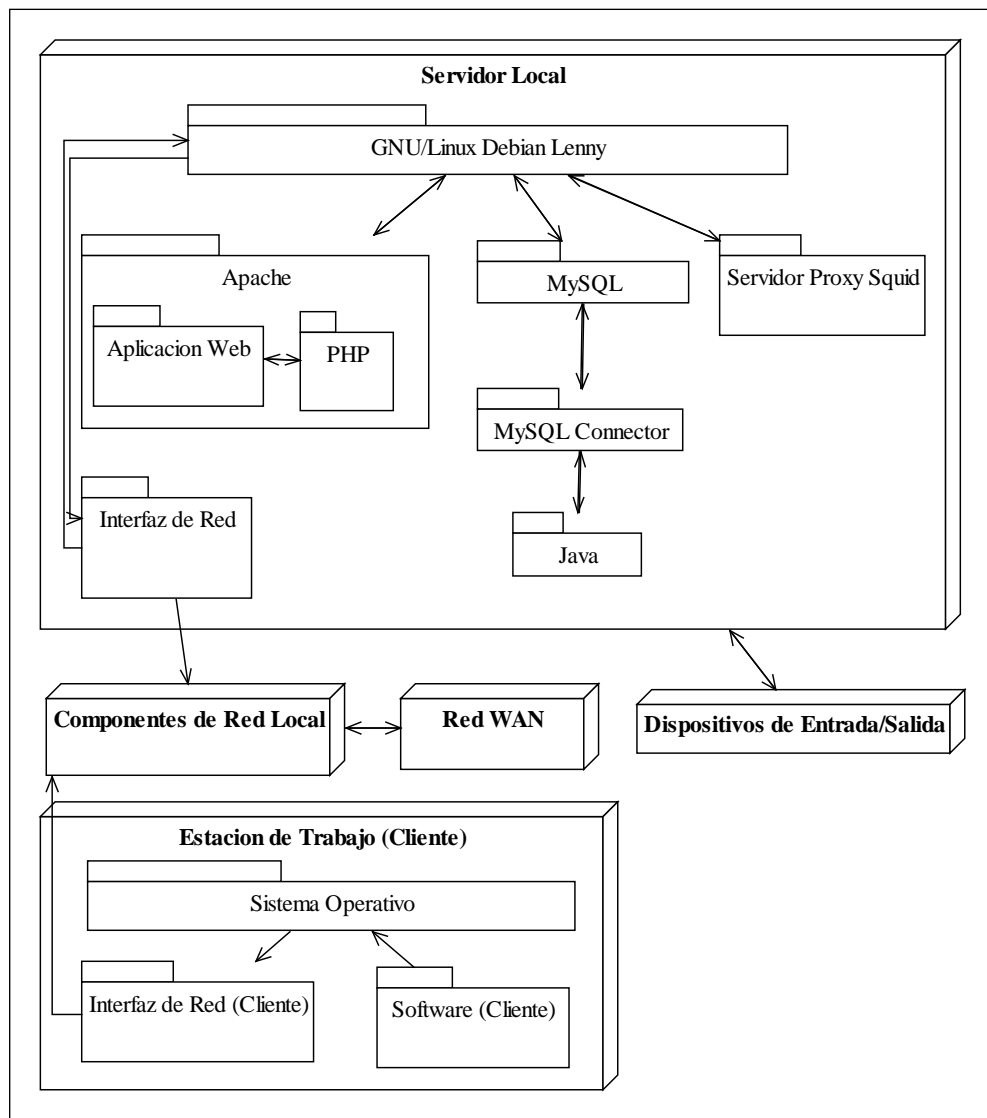
Cuadro 16. Diagrama de Arquitectura Final



Fuente: Montesino, P y Morales, J (2010)

El diagrama de despliegue detalla los componentes de software y hardware, y la interrelación que existe entre ellos para dar soporte al sistema. Este diagrama muestra también la infraestructura usual de implantación.

Cuadro 17. Diagrama de Despliegue



Fuente: Montesino, P y Morales, J (2010)



4.4. Fase IV: Evaluación de los Resultados del Sistema Propuesto.

En esta etapa de la investigación se analizaron los resultados obtenidos del desarrollo del sistema, se evaluó si el sistema cumplía los requisitos inicialmente propuestos y si las expectativas en cuanto a su desarrollo fueron cubiertas. Se obtuvieron los siguientes modelos y diagramas:

- a) Casos de Prueba
- b) Manual del Usuario (Ver Anexo A)


Los siguientes cuadros muestran las pruebas realizadas al sistema, y los resultados obtenidos de la aplicación de las mismas.

Cuadro 18. Casos de Prueba Caja Negra N° 1

	DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE SERVIDORES PROXY EN SISTEMAS OPERATIVOS GNU/LINUX. CASO: SQUID PROXY			 CACHALOG v1.0
Casos de Prueba N 1				
Estrategia de Prueba	Caja Blanca			
	Caja Negra			
Tipo de Prueba Aplicada				
Unidad		Modular		
Integración		Validación		
Técnica de Prueba	Carga de Datos			
Resultados	Al ingresar la dirección IP en el formulario de opciones de red se pudo apreciar que aceptó valores por encima de 255 generando así errores de configuración.			
Decisión	Revisar validaciones en el ingreso de los valores IP			
Probado por	Morales y Montesino			


Fuente: Montesino, P y Morales, J (2010)

Cuadro 19. Casos de Prueba Caja Negra N° 2

	DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE SERVIDORES PROXY EN SISTEMAS OPERATIVOS GNU/LINUX. CASO: SQUID PROXY		 CACHALOG v1.0
Casos de Prueba N 2			
Estrategia de Prueba	Caja Blanca		
	Caja Negra		
Tipo de Prueba Aplicada			
Unidad		Modular	
Integración		Validación	
Técnica de Prueba	Carga de Datos		
Resultados	Luego de Guardar una Regla de Acceso se verificó en pantalla que no se había guardado en el orden correcto y valido lo que generó un error en el servicio del squid proxy.		
Decisión	Verificar el archivo donde se hace el algoritmo de ordenación validando que se almacene de forma correcta.		
Probado por	Morales y Montesino		



Fuente: Montesino, P y Morales, J (2010)

Cuadro 20. Casos de Prueba Caja Negra N° 3

	DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE SERVIDORES PROXY EN SISTEMAS OPERATIVOS GNU/LINUX. CASO: SQUID PROXY		 CACHALOG v1.0
Casos de Prueba N 3			
Estrategia de Prueba	Caja Blanca		
	Caja Negra		
Tipo de Prueba Aplicada			
Unidad		Modular	
Integración		Validación	
Técnica de Prueba	Usabilidad		
Resultados	Permite el ingreso desde la barra de direcciones al menú principal del sistema.		
Decisión	Revisar el archivo del menú principal para verificar que el inicio de sesiones de usuarios esté activado.		
Probado por	Morales y Montesino		



Fuente: Montesino, P y Morales, J (2010)

Cuadro 21. Casos de Prueba Caja Negra N° 4

	DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE SERVIDORES PROXY EN SISTEMAS OPERATIVOS GNU/LINUX. CASO: SQUID PROXY			 CACHALOG v1.0
Casos de Prueba N 4				
Estrategia de Prueba	Caja Blanca			
	Caja Negra			
Tipo de Prueba Aplicada				
Unidad		Modular		
Integración		Validación		
Técnica de Prueba	Carga de Datos			
Resultados	Se registró las reglas de control de acceso y se genero un error al ejecutar la configuración.			
Decisión	Revisar los archivos de registro de rcl			
Probado por	Morales y Montesino			



Fuente: Montesino, P y Morales, J (2010)

Cuadro 22. Casos de Prueba Caja Negra N° 5

	DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE SERVIDORES PROXY EN SISTEMAS OPERATIVOS GNU/LINUX. CASO: SQUID PROXY		 CACHALOG v1.0
Casos de Prueba N 5			
Estrategia de Prueba	Caja Blanca		
	Caja Negra		
Tipo de Prueba Aplicada			
Unidad		Modular	
Integración		Validación	
Técnica de Prueba	Funcionalidad		
Resultados	No se presentaron evidencias de que el servicio iniciar levantara el Proxy		
Decisión	Revisión del servicio auxiliar.		
Probado por	Morales y Montesino		


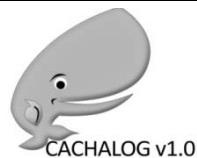
Fuente: Montesino, P y Morales, J (2010)

Cuadro 23. Casos de Prueba Caja Blanca N° 1

	DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE SERVIDORES PROXY EN SISTEMAS OPERATIVOS GNU/LINUX. CASO: SQUID PROXY			
Casos de Prueba N 1				
Estrategia de Prueba	Caja Blanca			
	Caja Negra			
Tipo de Prueba Aplicada				
Unidad		Modular		
Integración		Validación		
Técnica de Prueba	Validación de Datos			
Resultados	Al ingresar valores IP se pudo apreciar que aceptaban valores por encima de 255			
Decisión	Se realizó la debida validación en el ingreso de las ternas que conforman un dirección ip restringiendo el ingreso de valores por encima de los 255.			
Probado por	Morales y Montesino			



Fuente: Montesino, P y Morales, J (2010)

Cuadro 24. Casos de Prueba Caja Blanca N° 2

	DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE SERVIDORES PROXY EN SISTEMAS OPERATIVOS GNU/LINUX. CASO: SQUID PROXY			
Casos de Prueba N 2				
Estrategia de Prueba	Caja Blanca			
	Caja Negra			
Tipo de Prueba Aplicada				
Unidad		Modular		
Integración		Validación		
Técnica de Prueba	Validación de Datos			
Resultados	Se pudo apreciar que el orden de las reglas ACL no era el correcto lo que genera problemas en la configuración del squid proxy.			
Decisión	Se verificó el algoritmo de ordenación de las Reglas ACL depurando su debido funcionamiento, logrando la solución del problema.			
Probado por	Morales y Montesino			



Fuente: Montesino, P y Morales, J (2010)

Cuadro 25. Casos de Prueba Caja Blanca N° 3

	DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE SERVIDORES PROXY EN SISTEMAS OPERATIVOS GNU/LINUX. CASO: SQUID PROXY		
Casos de Prueba N 3			
Estrategia de Prueba	Caja Blanca		
	Caja Negra		
Tipo de Prueba Aplicada			
Unidad		Modular	
Integración		Validación	
Técnica de Prueba	Funcionalidad		
Resultados	Al hacer clic en el botón de iniciar squid no se encontraron resultados que indicarán que el servicio se había levantado.		
Decisión	Se verificó el campo correspondiente en la base de datos y el algoritmo del servicio auxiliar.		
Probado por	Morales y Montesino		

Fuente: Montesino, P y Morales, J (2010)

Cuadro 26. Casos de Prueba Caja Blanca N° 4

	DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE SERVIDORES PROXY EN SISTEMAS OPERATIVOS GNU/LINUX. CASO: SQUID PROXY			
Casos de Prueba N 4				
Estrategia de Prueba	Caja Blanca			
	Caja Negra			
Tipo de Prueba Aplicada				
Unidad		Modular		
Integración		Validación		
Técnica de Prueba	Interfaz			
Resultados	Los textos en varios formularios variaban en formato y estilos.			
Decisión	Se solucionó mediante la aplicación de hojas de estilos unificando todos los formatos y adjuntándolos a cada archivo de la aplicación.			
Probado por	Morales y Montesino			

Fuente: Montesino, P y Morales, J (2010)

Cuadro 27. Casos de Prueba Caja Blanca N° 5

	DESARROLLO DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE SERVIDORES PROXY EN SISTEMAS OPERATIVOS GNU/LINUX. CASO: SQUID PROXY		 CACHALOG v1.0
Casos de Prueba N 5			
Estrategia de Prueba	Caja Blanca		
	Caja Negra		
Tipo de Prueba Aplicada			
Unidad		Modular	
Integración		Validación	
Técnica de Prueba	Carga de Datos		
Resultados	Se generó un error de conexión con la base de datos al ingresar un valor en la tabla categoría.		
Decisión	Se verificó el archivo de conexión con la base de datos no encontrando ninguna anomalía los datos eran correctos por lo que se detectó que el servicio de base de datos no estaba levantado.		
Probado por	Morales y Montesino		

Fuente: Montesino, P y Morales, J (2010)

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusión

El creciente uso de los servicios de red, y la presencia cada vez mayor de la Web en los ambientes organizacionales, ha conllevado a que el Personal de Tecnología de la Información emplee 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 Proxy 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.

El presente trabajo de grado 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 el personal de tecnología de la información 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, lo que hace posible llevar este sistema de control a las pequeñas y medianas empresas donde los recursos para la inversión en tecnología son limitados. En conclusión, el resultado es una aplicación que responde a las necesidades de la administración de los recursos de red en los nuevos tiempos.

5.2. Recomendaciones

Para investigaciones futuras se considera importante tomar en cuenta los siguientes tópicos:

- Analizar la arquitectura de Squid Proxy bajo plataforma Microsoft Windows y elaborar componentes para emplear la aplicación web desarrollada en este trabajo de grado en ese Sistema Operativo.
- Agregar módulos para la generación de reportes más especializados.
- Adicionar componentes para administración avanzada.
- Aumentar la funcionalidad, permitiendo al sistema configurar múltiples servidores Squid Proxy desde una misma interfaz.
- Promover el uso de Squid Proxy como herramienta de los Departamentos de TI.
- Diseñar un sistema para la instalación automatizada del sistema, que disminuya el tiempo de instalación y reduzca los posibles errores en el proceso.
- Diseñar un manual de instalación del sistema y de todos los paquetes asociados.

REFERENCIAS

Bibliográficas

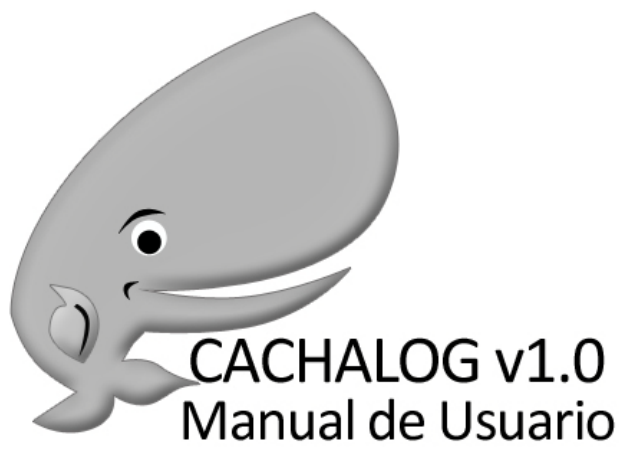
- Balestrini, Miriam, (1998) “Cómo se Elabora El Proyecto De Investigación” Caracas, VENEZUELA, 2 Edición.
- Bautts Tony, Dawson Terry y Purdy Gregor (2005) Linux. Guía para administradores de redes, Primera Edición, Anaya multimedia/O’reilly
- Cázares, Chisten y Jaramillo, (2000) “Técnicas actuales de investigación documental”. México.
- Gallego, Alfredo y Gallego, Miguel, (2004) UNIX/Linux, Segunda Edicion, España, McGraw Hill.
- Gift, Noah y Jones Jeremy (2009) Python para administración de sistemas Unix y Linux, Primera Edición, España, Anaya Multimedia.
- Hagen, Bill (2006). Linux Servers Los Mejores Trucos. 1ra edición. Editorial O’Reilly.
- Perez, Justo y Matas, Abel. (2004), Guia Practica para Usuarios Debian. Primera Edición. Madrid. Anaya Multimedia, S.A.
- Thompson, Ed y Myer Thomas (2009) PHP 6 Profesional, Primera Edición, España, Anaya Multimedia/Wrox
- Timothy Boronczyk y Psinas, Martin (2009) Php y Mysql Crea-Modifica-Reutilizar, Primera Edición, España, Anaya Multimedia/Wrox.
- Universidad José Antonio Páez (2007). Normas para la elaboración y presentación de los anteproyectos, proyectos y trabajos de grado. San Diego: Mijares, Hector.
- Universidad Pedagógica Experimental Libertador (2006). Manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales de la universidad Pedagógica Experimental Libertador.
- Wessels, Duane (2001) Web Caching. First Edition. United States of America.O'Reilly & Associates, Inc.
- Wessels, Duane (2004) Squid: The Definitive Guide. First Edition. United States of America.O'Reilly & Associates, Inc.

Electrónicas

- Universidad San Martin de Porres - Peru. Metodologia RUP Vs. XP. Consultado en Octubre 2010 en <http://www.usmp.edu.pe/publicaciones/boletin>
- Welcome to the Tcl Developer Xchange! (2010). consultado el dia 15 de julio del 2010 de la World Wide Web: <http://www.tcl.tk>
- Wessels, Duane y Nordstrom, Henrik (2010) Why should I deploy Squid? Consultado el dia 15 de junio del 2010 de la World Wide Web:<http://www.squid-cache.org/Intro/why.dyn>

ANEXOS

ANEXO A



Ingresando a la Aplicación web para la administración del Squid Proxy.



Para entrar a la aplicación es necesario el ingreso de los datos de usuario y contraseña para la debida autenticación como administrador del servidor proxy Squid.

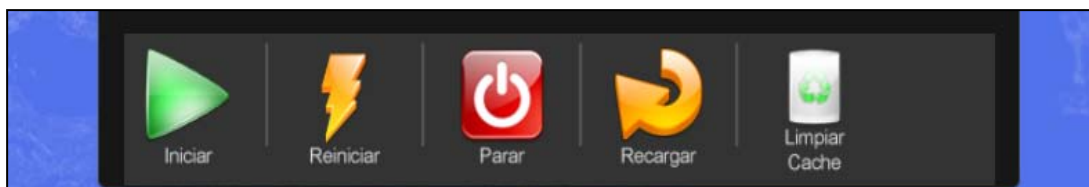
Ingresando al Menú Principal

Una vez verificado que los datos ingresados por el administrador sean correctos la aplicación le dará acceso al menú principal de la aplicación en donde encontrará las pestañas concernientes a los distintos módulos relacionados con los servicios, configuración, monitoreo y administración del servidor squid proxy.



Menú Servicios

Procedemos a hacer clic en la pestaña de Servicios y seguido nos aparecerá la ventana referente a la Administración de servicios del Squid Proxy tales como: iniciar, parar, reiniciar, recargar y limpiar la cache del servidor.





A continuación se procede a explicar detalladamente cada una de las opciones dispuestas en el módulo de servicios:

Iniciar: esta opción le permitirá al administrador ejecutar una orden de inicializar el demonio squid en el servidor.

Reiniciar: esta opción le permite al administrador ejecutar la orden de reiniciar el demonio squid en el servidor.

Parar: esta opción le permite al administrador ejecutar la orden para detener el demonio squid en el servidor.

Recargar: esta opción le permite al administrador ejecutar la orden al servidor squid proxy para que vuelva a leer el archivo de configuración sin necesidad de reiniciar el demonio squid.

Limpiar Cache: esta opción le permite al administrador enviar la orden al servidor proxy de limpiar los archivos dispuestos en cache.

Menú Configuración



Luego de hacer clic en la pestaña de configuración observamos el menú principal de este módulo donde podemos encontrar las opciones de configuración por defecto o avanzada que le permitirán al administrador el establecimiento de parámetros básicos o avanzados para poner en marcha el servidor squid proxy.



Procederemos a explicar cada uno de los elementos del menú de configuración:

Configuración por defecto: el hacer clic en este botón le permitirá al administrador levantar el servidor proxy squid con los datos mínimos para su correcto funcionamiento.

Configuración avanzada: al hacer clic en este botón se mostrará el menú de configuración avanzada donde se encuentran los elementos necesarios para la configuración del squid proxy.





Al hacer clic en el botón de parámetros de red ingresaremos al formulario de configuración de opciones de Red. Que se muestra a continuación.

Formulario de Opciones de Red:

En el siguiente formulario el administrador deberá ingresar la dirección ip así como el puerto donde

Configuracion de Puertos

Directiva
http_port

Incrementa la Seguridad vinculando la Direccion IP al Puerto

Direccion IP: 0 0 0 0
Recomendaciones Generales
Puertos 3128 y 8080 para TCP

Registrar

ID	PUERTO	VALOR	ELIMINAR
14	HTTP_PORT	172.16.23.1:3128	
73	HTTP_PORT	4000	

Formulario de Parámetros de Cache:



Al hacer clic en el botón de parámetros de Cache ingresaremos al formulario donde el administrador deberá ingresar los parámetros concernientes a los datos de la cache tales como ubicación de la cache en disco tamaño de la cache y niveles de memoria.

Parametros de Memoria Cache					
Directorio Cache					
Directiva	Tipo	Directorio de Almacenamiento en Cache	Tamaño en disco	L1	L2
Cache_dir	ufs ▼	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
					<input type="button" value="Registrar"/>



Al hacer clic en el botón de parámetros de Archivos Log ingresaremos al formulario donde se administrarán las rutas de los archivos log del squid proxy.

Formulario de Listas de Control de Acceso (ACL)



Cuando hacemos clic en el botón de Listas de control de Acceso se nos mostrará un formulario donde tendremos q ingresar el nombre de la nueva lista ACL así como el tipo de la misma.

LISTA DE GRUPOS	
Ingrese un nombre para su Grupo:	<input type="text"/>
Seleccione el Tipo de Acl:	Seleccione el Tipo del Acl ▼
<input type="button" value="Guardar"/>	

Al guardar la Lista de Control de Acceso se presentara el formulario para ingresar los elementos de la Lista de Acceso en el caso de que sean grupos de elementos.

LISTA DE CONTROL DE ACCESO				
Nombre del ACL: CARABOBO				
Elementos de su Lista:	0	0	0	0 /
Ingresar				
Listas de Valores				
ELEMENTO		ELIMINAR		

Formulario de Reglas de Control de Acceso (RCL):



El siguiente botón nos presentará el formulario de Reglas de Lista de Control de Acceso.

En el cual se definirán las reglas que tendrán cada una de las listas de control de Acceso.

REGLAS DE LISTAS DE ACCESO			
Num	Directiva	Acceso	Acceso
5	http_access	deny	Seleccione el Tipo del Grupo

Agregar

Listas de control de acceso



Al hacer clic en la opción de visualizar configuración podremos observar el archivo squid.conf donde reposarán las directivas y parámetros seleccionados en el menú de configuración.

```
Visualizar Archivo de Configuración

##### Parametros de la Red #####
http_port 172.16.23.32:3128

##### Parametros de la cache y configuracion de Logs ###

cache_dir ufs /var/spool/squid 6000 32 512
access_log /var/log/squid/access.log
cache_log /var/log/squid/cache.log
cache_store_log /var/log/squid/store.log

##### Listas de Control de Acceso

acl manager proto cache_object
acl all src 0.0.0.0/0.0.0.0
acl localhost src 127.0.0.1/32
acl publica arp "/etc/squid/acls/publica"
acl privada arp "/etc/squid/acls/acceso"
acl dominios.aceptados dstdomain "/etc/squid/acls/dominios.aceptados"
acl dominios.porno dstdomain "/etc/squid/acls/dominios.porno"
acl dominios.msn dstdomain "/etc/squid/acls/dominios.msn"
```

Menú Monitoreo

Al hacer clic en esta pestaña observaremos las distintas opciones de resultados estadísticos sobre el tráfico consumido en la red.



Menú Administración



El botón de Configuración del Cachalog le permitirá al administrador poder ingresar datos generales de configuración del Squid proxy. Tales como las rutas del log y comandos de servicios.



El botón de datos de la Organización le permitirá al administrador ingresar la información relacionados con la organización donde se está ejecutando el Cachalog para la administración del Squid proxy.



Usuarios del Cachalog le permiten al administrador gestionar los usuarios que podrán administrar la aplicación



Al hacer clic sobre este botón el administrador tendrá acceso al mantenimiento de tablas auxiliares de la aplicación tales como categorías, directivas, parámetros entre otros.