Universidad de Cienfuegos "Carlos Rafael Rodríguez"

Facultad de Informática

Carrera de Ingeniería Informática

Sistema para la centralización, análisis y procesamiento de trazas de servicios telemáticos para la Red UCf: Nuevos módulos y servicios.

Trabajo de diploma para optar por el título de Ingeniería en Informática

Autor:

Iris Castillo Quintana.

Tutores:

MSc. Alexis Gómez Domínguez. PDVCupetSA "Refinería de Petróleo de Cienfuegos".

Ing. Jorge Luis Rivero Pérez. Universidad de Cienfuegos "Carlos Rafael Rodríguez".

Ing. Lissette Montero. Universidad de Cienfuegos "Carlos Rafael Rodríguez".

Cienfuegos, Cuba
Curso 2009 – 2010

Agradecimientos

Gracias a Dios por ser hoy lo que soy.

Gracias a mis padres y hermano por ayudarme a cumplir todas mis metas.

Gracias a mi esposo por pasar conmigo todas las pruebas difíciles y apoyarme siempre.

Gracias en especial a Raúl por su ayuda y apoyo incondicional.

Gracias a mi tía Elba y mis primas por estar pendientes de mi a pesar de la distancia.

Gracias a mis tías Coca y Marta por tratar siempre de ayudarme.

Gracias a mis tutores Alexis, Jorge y Lissette por brindarme su ayuda y dedicación.

Gracias a mis compañeros de curso, en especial a mis amigos: Janny, Boris, Ary, Yeidy y Mailé.

Gracias a mis profesores por contribuir a mi preparación profesional.

Dedicatoria

Dedico este trabajo especialmente a mis padres, mi esposo y mi hermano por formar parte imprescindible de la realización de todos mis sueños.

Resumen

El presente trabajo surge como continuidad del Sistema para la centralización, análisis y procesamiento de trazas de servicios telemáticos para la Red UCf, atendiendo a las recomendaciones asociadas a su desarrollo inicial e incorporándole nuevas funcionalidades, como son: un módulo para la gestión de alarmas, mejoras a los módulos existentes, nuevas opciones de visualización de resultados y nuevas trazas de servicios telemáticos, específicamente Squid y Radius, elevando así las potencialidades del sistema como herramienta de trabajo de los administradores de redes en lo relacionado con las tareas de supervisión y control de todos los servicios telemáticos existentes en la red.

Índice

Introducción.....1

Capítulo I. Fundamentación teórica.....6

1.1Conceptos asociados al dominio del problema.....6

1.2Descripción del objeto de estudio.....8

1.2.1Flujo actual de los procesos y análisis crítico de la ejecución de estos 8

1.3 Sistemas automatizados existentes vinculados al campo de acción.....9

1.4 Tendencias, metodologías y tecnologías actuales.....10

1.4.1 Tecnologías Web.....10

1.4.2 Fundamentación del lenguaje, gestor de bases de datos y de otros software utilizados.....11

1.4.3 Herramientas para el desarrollo.....16

1.4.4 Fundamentación de la metodología utilizada.....16

1.5 Conclusiones.....18

Capítulo II. Modelo del dominio y requisitos.....19

2.1 Descripción del modelo del dominio.....19

2.2 Definición de las entidades y los conceptos principales.....19

2.3 Reglas del negocio a considerar.....20

2.4 Representación del modelo del dominio.....21

2.5 Descripción del sistema propuesto.....21

2.5.1 Concepción general del sistema.....21

2.5.2 Instalación y configuración del sistema.....22

2.5.2.1 Centralización de las trazas.....22

2.5.2.2 Módulo de captura de la información almacenada en las trazas. 26

2.5.2.3 Módulo de pos procesado.....26

2.5.3 Requerimientos funcionales.....26

2.5.4 Requerimientos no funcionales.....28

2.6 Modelo de casos de uso del sistema....31
2.6.1 Actores del modelo del sistema....31
2.6.2 Paquetes y sus relaciones....32
2.6.3 Casos de uso del sistema....32
2.6.4 Diagramas de casos de uso del sistema....33
2.6.5 Descripción de los casos de uso del sistema....34
2.6.5.1 Descripción de los casos de uso del módulo de pos procesado....34
2.6.5.2 Descripción de los casos de uso del módulo de captura de la información contenida en las trazas....39
2.7 Conclusiones....40
Capítulo III: Construcción y validación de la solución propuesta....41
3.1 Diagrama de clases del diseño....41
3.2 Diseño de la base de datos....42
3.2.1 Modelo lógico de datos....42
3.2.2 Modelo físico de datos....42
3.3 Diagrama de implementación....45
3.4 Principios de diseño del sistema....45
3.4.1 Estándares en la interfaz de la aplicación....45
3.4.2 Tratamiento de errores....46
3.4.3 Concepción general de la ayuda....46
3.5 Análisis de prueba y estudio de factibilidad....47
3.5.1 Pruebas y validaciones realizadas....47
3.5.1.1 Modificación de las trazas en los equipos que las generan....47
3.5.1.2 Simulación de propagación de virus conocidos....47
3.5.1.3 Simulación de accesos no autorizados por RAS....48
3.5.1.4 Simulación de accesos a sitios peligrosos o url peligrosas....48
3.5.1.5 Simulación de falsos positivos....48

3.5.1.6 Simulación de intentos de accesos no autorizados 48

3.5.1.7 Simulación de ataque de DDoS 49

3.5.1.8 Simulación de intentos de Spoofing 49

3.5.1.9 Simulación de monopolización del canal o el Backbone por un Host 49

3.5.1.10 Simulación de empleo de herramientas de Scanner de vulnerabilidades desde un Host no autorizado 49

3.5.1.11 Simulación de violación de privilegios en el servicio Web 49

3.5.1.12 Herramientas de Software utilizadas para validar los resultados 50

3.6 Estudio de factibilidad 50

3.6.1 Planificación por casos de usos 51

3.6.1.1 Cálculo de Puntos de Casos de Uso sin Ajustar 51

3.6.1.2 Cálculo de los Puntos de Casos de Uso Ajustados 52

3.6.2 Estimación del esfuerzo 56

3.6.3 Cálculo de costos 57

3.6.4 Beneficios tangibles e intangibles 57

3.7 Conclusiones 57

Conclusiones 59

Recomendaciones 60

Referencias Bibliográficas 61

Bibliografía 62

Glosario de Términos 64

Anexos 68

Índice de figuras

Figura 1. Diagrama de clases del modelo de objetos del dominio. 21

Figura 2. Paquetes y sus relaciones 32

Figura 3. Diagrama de casos de uso del paquete de captura de la información contenida en las trazas. 34

Figura 4. Diagrama de casos de uso del paquete de pos procesado. 34

Figura 5. Diagrama de clases persistentes para la base de datos de servicios. 42

Figura 6. Diagrama del modelo físico de datos para la base de datos de servicios. 45

Figura 7. Diagrama de implementación. 45

Índice de tablas

Tabla 1. Actores del Sistema. 32

Tabla 2. Descripción del caso de uso de sistema: Gestionar reglas por servicio. 35

Tabla 3. Descripción del caso de uso de sistema: Gestionar alarmas. 36

Tabla 4. Descripción del caso de uso de sistema: Gestionar envío de alarmas. 37

Tabla 5. Descripción del caso de uso de sistema: Gestionar criterios de configuración. 38

Tabla 6. Descripción del caso de uso de sistema: Procesar los datos almacenados de las trazas del servicio Radius. 38

Tabla 7. Descripción del caso de uso de sistema: Procesar los datos almacenados de las trazas del servicio Squid. 39

Tabla 8. Descripción del caso de uso de sistema: Filtrar y Almacenar la información contenida en las trazas. 40

Tabla 9. Diagramas de clases Web. Paquete de captura de la información contenida en las trazas. 41

Tabla 10. Diagramas de clases Web. Paquete de pos procesado. 42

Tabla 11. Asignación de peso a los actores. 52

Tabla 12. Factor de peso de los casos de uso sin ajustar. 52

Tabla 13. Cálculo del factor de complejidad técnica. 54

Tabla 14. Cálculo del factor ambiente. 56

Tabla 15. Criterios de distribución de esfuerzos. 57

Introducción

Las actuales redes de telecomunicación se caracterizan por un constante incremento del número, complejidad y heterogeneidad de los recursos que la componen por lo cual el análisis y monitoreo de las mismas se ha convertido en una labor cada vez más importante ante la ocurrencia de cualquier fallo. Es por ello que se hace necesario el uso de herramientas que permitan realizar el análisis de la información contenida en las trazas que generan los diferentes servicios y dispositivos de red de forma centralizada permitiendo que el administrador no tenga que realizar dicho análisis de forma manual y pueda obtener reportes que le informen sobre el estado de los mismos. Por otra parte resulta engorroso y tedioso localizar manualmente problemas que ocurren en la red por lo que contar con un sistema capaz de notificar mediante alarmas la ocurrencia de situaciones anómalas y poder gestionar dichas alarmas desempeña una útil función ya que se brinda la posibilidad al administrador de definir reglas por servicios con el objetivo de que ante determinadas ocurrencias se puedan generar alarmas que brinden información relacionada con el fallo, ofreciendo especificidades de las trazas donde se produjo el mismo. En la red de la Universidad de Cienfuegos existe un sistema de análisis, centralización y procesamiento de las trazas que generan los diferentes servicios telemáticos de la red. Dicho sistema permite la generación de reportes que contienen información acerca del estado de los mismo pero este se encuentra limitado al análisis de determinados servicios, por lo que no todos los servicios pueden ser procesados por el mismo y algunos se analizan de forma manual o con el uso de otras herramientas que son útiles para la generación de reportes pero no lo suficientemente adecuadas para la resolución de problemas ocurridos en la red.

Situación del problema

La información generada por los servicios telemáticos de red de la Universidad de Cienfuegos es almacenada en un servidor común con el objetivo de facilitar el proceso de análisis de trazas; pero existen servicios como Proxy (Squid) y Radius que no almacenan la información en dicho servidor por lo que el análisis de eventos ocurridos en la red por parte de estos servicios continúa

realizándose de forma local con herramientas independientes lo cual requiere de tiempo. Además el sistema utilizado no brinda la posibilidad de generar alarmas de acuerdo a necesidades que se vayan presentando al administrador ya sea en uno o varios servicios, ya que una vez que ocurre un fallo el administrador debe revisar cierto volumen de información para conocer qué trazas específicamente produjeron el mismo.

Problema

Por la situación antes expuesta se plantea como problema de la investigación, la carencia por parte del "Sistema para la centralización, análisis y procesamiento de trazas de servicios telemáticos para la Red UCf" del módulo de gestión de alarmas, así como la centralización, procesamiento y análisis de las trazas de los servicios telemáticos Squid y Radius.

Objeto de estudio y campo de acción

Se considera como objeto de estudio de la presente investigación los mensajes generados por los servicios de red Radius y Squid así como la gestión de alarmas para los servicios Cisco, Apache2, Squid y Radius. Dicha investigación se desarrolla en la red de la Universidad de Cienfuegos.

Idea a defender

Con el desarrollo del módulo de gestión de alarmas y la incorporación del análisis y procesamiento de trazas de los servicios telemáticos Squid y Radius, mejorarán las tareas de supervisión y control de los servicios telemáticos de la Red UCf.

Justificación de la investigación

El desarrollo de esta investigación permite aumentar las opciones de monitoreo ya que brinda la posibilidad de incorporar al sistema de análisis y procesamiento de trazas existente en la red UCf, la centralización de trazas de los servicios telemáticos Squid y Radius. Además permitirá la generación de alarmas que informen la ocurrencia de situaciones críticas de acuerdo a reglas definidas previamente por el administrador, el cual no tendrá que ingresar al

sistema para saber si ocurrió un fallo ya que en caso de que la prioridad de la alarma sea alta o media recibirá un correo y/o un SMS[1] informándole de la cantidad de alarmas de estos tipos detectadas. Con esto queda evidenciada la posibilidad de emprender acciones proactivas, encaminadas a garantizar la estabilidad, disponibilidad y seguridad de la red.

Objetivo general

Desarrollar el módulo de gestión de alarmas para los servicios Cisco, Apache2, Squid y Radius e incorporar el módulo para el análisis y procesamiento de trazas de los servicios telemáticos Squid y Radius al Sistema de centralización, análisis y procesamiento de trazas de servicios telemáticos desarrollado para la red UCf.

Objetivos específicos

- Desarrollar para el sistema de análisis y procesamiento de trazas de la red UCf; el módulo de centralización y almacenamiento de trazas de los servicios telemáticos Squid y Radius.

- Desarrollar el módulo de gestión de alarmas para los servicios telemáticos Cisco, Apache2, Squid y Radius.

- Validar la implementación final, mediante el empleo de herramientas de Software y el análisis del comportamiento del sistema.

Tareas científicas

Para el desarrollo de la investigación se utilizarán diferentes métodos y técnicas que permitirán enfrentar el problema y favorecerán el cumplimiento de las siguientes tareas:

- Realización de un estudio del funcionamiento del sistema actual con el fin de encontrar deficiencias en su implementación.

[1] SMS: Sistema para el envío de mensajes de texto en las redes de telefonía móvil.

- Revisión de la bibliografía técnico-especializada para la construcción del marco teórico que permita conocer el estado del arte referente a las tecnologías empleadas en los sistemas de análisis y procesamiento de trazas, principio de funcionamiento, ventajas y desventajas asociadas.

- Entrevistas al personal capacitado para conocer más a fondo conceptos que se deben tener en cuenta para la investigación.

- Realización de pruebas prácticas y utilización de aplicaciones que permitan reunir elementos para la definición de las técnicas y mecanismos de seguridad a emplear, así como la simulación de situaciones reales en escenarios de pruebas.

- Validación de la implementación final mediante el empleo de herramientas de Software y el análisis del comportamiento del sistema.

Aporte Práctico

El desarrollo del sistema propuesto permitirá centralizar, analizar y procesar las trazas de los servicios telemáticos Radius y Squid. Además el mismo dará la posibilidad de que el administrador configure reglas para generar alarmas ante eventos que ocurran en la red de la universidad de Cienfuegos.

Estructura del trabajo

El presente trabajo seguirá la siguiente estructura:

Capítulo I.- "Fundamentación teórica": Este capítulo recoge el contenido relacionado con la fundamentación teórica del tema a desarrollar. Se aborda todo lo relacionado con el estado del arte, se describe el objeto de estudio, se hace un análisis de los procesos que dan origen a la situación del problema y se analizan las tendencias, metodologías y tecnologías actuales.

Capítulo II.- "Modelo del dominio y requisitos": En el presente capítulo se hace una descripción detallada del modelo del dominio, se definen conceptos y entidades así como las reglas del negocio a tener en cuenta. Además se describe el sistema propuesto identificando los requerimientos funcionales y no funcionales, los casos de uso del sistema, actores, paquetes y sus relaciones, la descripción de casos de uso y los diagramas de los mismos.

Capítulo III.- "Construcción y validación de la solución propuesta": En este capítulo se describe la solución propuesta mediante los diagramas de clases de diseño, el diseño de la base de datos, los modelos lógico y físico de los datos, el diagrama de implementación, así como principios de diseño del sistema. Posteriormente se hace una descripción de los resultados obtenidos luego de probar el sistema y las validaciones asociadas. Se realiza un estudio de factibilidad del sistema mediante el análisis de puntos de casos de uso, teniendo en cuenta para ello la estimación del esfuerzo, el cálculo de costos y el análisis de los beneficios.

Capítulo I. Fundamentación teórica

El presente capítulo expone todo el contenido relacionado con la fundamentación teórica del tema a desarrollar. En el mismo se abordan conceptos relacionados con el dominio del problema, se describe el objeto de estudio, se trata todo lo relacionado con el estado del arte, se analiza la forma en que se realizan los procesos que dan lugar a dicha investigación, se hace una descripción de sistemas existentes así como metodologías, tecnologías y tendencias actuales.

1.1Conceptos asociados al dominio del problema

Servidor de red:

"Un servidor de red es una computadora que maneja peticiones de datos, correo electrónico, transferencia de archivos de otras computadoras (clientes) y variados servicios telemáticos. También puede referirse a un Software específico, como lo es el servidor WWW. Una computadora puede tener distintas aplicaciones de Software del tipo servidor, proporcionando muchos servidores a clientes en la red.

Por ejemplo, las computadoras que contienen sitios Web se llaman servidores ya que "sirven" recursos de Web para aplicaciones cliente como los navegadores o Browsers." (1)

Dispositivo de red:

Los dispositivos de red son aquellos que permiten la interconexión de los Hosts que participan de una red, los mismos se describen generalmente en función de las operaciones que realizan según el modelo de referencia OSI o la arquitectura TCP/IP como dispositivos de nivel 1, 2, 3, 4 o superior, también suelen llamarse como dispositivos de nivel físico, de nivel de enlace, de nivel de red, de nivel de transporte o de nivel de aplicación haciendo alusión al nombre de cada uno de dichos niveles. (1)

Servicios telemáticos:

Los servicios telemáticos están muy relacionados con los usuarios finales, son el elemento fundamental de una red de computadoras pues los enlaces,

servidores de red y equipos de interconexión no tienen utilidad práctica si no existen servicios telemáticos soportados en la red. Ejemplos clásicos de ellos son:

Correo electrónico

Acceso a páginas Web.

Bases de Datos de información.

Mensajería instantánea

Video Conferencias.

Foros de discusión.

Laboratorios virtuales.

Noticias (1)

Trazas:

"Las trazas son el primer registro de la actividad en los sistemas y redes. Ellos contienen el registro de acciones y eventos que tienen lugar en un sistema. Las trazas de un sistema son una parte primaria de la seguridad y pueden ser usados en la detección de ataques e intrusiones, así como en el análisis de fallas de Hardware y Software." (1)

Syslog centralizado:

"El programa Syslog, es una interface que provee un Framework estándar para que tanto programas como el mismo sistema operativo puedan generar mensajes que pueden ser almacenados localmente o ser enviados a un Host remoto. Originalmente escrito para Unix, se convirtió en un estándar que se usa en muchos sistemas operativos y dispositivos de red. En un sistema de Syslog centralizado, un servidor común recibe todos los mensajes de Syslog de todos los sistemas de la red. Esto incluye trazas de los servidores Unix/Linux/Windows etc., cortafuegos, y dispositivos de red (Routers, Switches, etc.)." (1)

Syslog-ng

"Si bien el demonio de Syslog estándar de Linux (*syslogd*) es adecuado para concentrar información de otros Syslog, se decidió utilizar *syslogng (syslog new*

generation) que es un reemplazo directo de Syslog. El syslog original permite que los mensajes sean organizados basándose en los pares *priority/facility*; *syslogng* agrega la posibilidad de filtrar basándose en el contenido de los mensajes usando expresiones regulares.

Syslogng permite guardar los mensajes en una base de datos y también usar TCP en lugar de UDP para enviar los mensajes a otro servidor." (1)

Script:

Un Script es un conjunto de instrucciones que permiten la automatización de tareas, creando pequeñas utilidades. Es muy utilizado para la administración de sistemas UNIX. Son ejecutados por un intérprete de línea de órdenes y usualmente son archivos de texto. (1)

1.2Descripción del objeto de estudio

1.2.1Flujo actual de los procesos y análisis crítico de la ejecución de estos

La red de la Universidad de Cienfuegos (UCf) cuenta aproximadamente con 14 servidores y varios equipos de red, los cuales brindan una gran variedad de servicios telemáticos que se encuentran distribuidos en dichos servidores. Cada uno de estos servicios genera una o varias trazas (registros de informaciones). De estos servicios se benefician aproximadamente 3000 usuarios entre estudiantes, trabajadores docentes y no docentes. La red interconecta cerca de 500 computadoras entre laboratorios de estudiantes, locales de profesores o trabajadores, la residencia estudiantil y sedes universitarias municipales (SUMs).

Las tareas relacionadas con el procesamiento y análisis de las trazas de los servicios telemáticos en la red de la Universidad de Cienfuegos se realiza con el uso de un sistema centralizado para el análisis y procesamiento de trazas pero el mismo está limitado, como se explicó anteriormente, a los servicios Cisco y Apache2 por lo que existen servicios como Squid y Radius que se analizan de forma manual (inspección visual) o de manera local y en los formatos propios de cada uno de ellos y con las configuraciones generadas por defecto, además en las trazas almacenadas existen referencias continuas a números de códigos y de errores estandarizados y publicados en los RFC pero

que por la gran cantidad de códigos existentes no se pueden memorizar completamente y tener que estar revisando un listado constantemente se convierte en un trabajo lento y tedioso. Tampoco existen algoritmos de análisis para determinar situaciones anómalas en la red ni mecanismos de generación de alarmas ante estas situaciones. Si bien es cierto que existe un elevado control de los eventos que ocurren y es posible determinar los detalles de la ocurrencia de los mismos, su realización se vuelve muy compleja requiriendo largas jornadas de trabajo y una participación total del administrador y de su nivel de preparación y experiencia para saber qué trazas revisar e interpretar correctamente los resultados, existiendo por consiguiente un margen de error elevado.

1.3 Sistemas automatizados existentes vinculados al campo de acción

Existe una gran variedad de sistemas informáticos desarrollados para realizar el análisis y procesamiento de las trazas que generan los servicios telemáticos en una red de computadoras, cada uno de ellos tiene características específicas y permiten controlar uno o varios servicios, algunos de ellos fueron analizados en la investigación asociada al desarrollo del "Sistema para la centralización, análisis y procesamiento de las trazas de servicios telemáticos de la red UCf". Entre los analizados se pueden mencionar los siguientes (1):

- PyFlag
- Bitácora 4.0
- IPtables Log Analyzer
- OSSEC
- Php-Syslog-ng
- Observer
- Logrep

SCAPT

El sistema SCAPT (Sistema centralizado para el análisis y procesamiento de trazas de servicios telemáticos para la red UCf) brinda la posibilidad y facilidad

de trabajar con las trazas generadas de forma que con los conocimientos que tengan los administradores en materia de seguridad informática y protocolos TCP/IP, pueda detectar anomalías y errores que no estaban previstos en la configuración del sistema, además permite evitar o comprobar en mayor medida los falsos positivos, satisfaciendo así las necesidades de los administradores de la Red UCf. El sistema desarrollado permite centralizar, analizar y procesar las trazas asociadas a los servicios telemáticos Apache2 y Equipamiento de redes con tecnología Cisco, el mismo brinda la posibilidad de revisión de grandes volúmenes de trazas de manera rápida y enfocada a criterios de búsquedas deseados por los administradores de redes, además permite graficar el comportamiento del tráfico en la red y generar alarmas a partir de reglas previamente configuradas e informaciones almacenadas en la base de datos.

Conclusiones sobre SCAPT:

A pesar de facilitar al administrador el trabajo con la información contenida en las trazas, se encuentra limitado a los servicios Apache2 y Radius. Además las reglas configuradas para la detección de alarmas están contenidas en el código de los script que realizan dicha función por lo que no es posible la modificación de las mismas desde el sistema.

1.4 Tendencias, metodologías y tecnologías actuales

Luego de realizar un estudio de las tecnologías actuales para seleccionar la más conveniente a utilizar en el proceso de desarrollo de la herramienta de software propuesta con el fin de lograr la calidad de dicha herramienta se eligió como metodología a utilizar el Proceso Unificado de Desarrollo; metodología implementada en el sistema actual. A continuación se presentan las principales características que justifican la selección de dicha metodología.

1.4.1 Tecnologías Web

Tecnologías Cliente

Navegador Web: Mozilla Firefox.

Tecnologías de programación:

HTML.

Hojas de estilo en cascada (CSS).

JavaScript.

Tecnologías Servidor

Servidor Web: Apache.

Servidor de Trazas: Syslog-ng

Servidor de Base de Datos: PostgreSQL 8.3

Tecnologías de Programación:

PHP

Python

1.4.2 Fundamentación del lenguaje, gestor de bases de datos y de otros software utilizados

Python

Se utilizó este lenguaje ya que resulta muy adecuado para manipular archivos de texto, siendo esta característica útil para el trabajo con las trazas. Python permite mantener de forma sencilla interacción con el sistema operativo y está disponible en MS-Windows, GNU/Linux, Mac y cualquier entorno *IX. Contiene librerías, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan en el trabajo con redes. Además posee otras características como:

- Propósito general: Se pueden crear todo tipo de programas.
- Multiplataforma: Hay versiones disponibles de Python en muchos sistemas informáticos distintos.
- Interpretado: Quiere decir que no se debe compilar el código antes de su ejecución.
- Interactivo: Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias.
- Orientado a Objetos: La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.

- Sintaxis clara: Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación identada (con márgenes) de obligado cumplimiento.

Python está en movimiento y en pleno desarrollo, pero ya es una realidad y una interesante opción para realizar todo tipo de programas que se ejecuten en cualquier máquina. (2)

HTML

Por ser HTML el lenguaje de marcado predominante para la construcción de páginas Web se decidió su utilización ya que el sistema cuenta con una interfaz web. Además dicho lenguaje se emplea para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML es un lenguaje muy sencillo que permite describir *hipertexto*, es decir, texto presentado de forma estructurada y agradable, con enlaces (*hyperlinks*) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia como gráficos y sonidos. (3)

CSS

Se hizo uso de CSS ya que especifica la forma del diseño de los documentos (tanto XHTML como HTML). Una misma página Web puede ser vista de diferentes formas, gracias a las hojas de estilo CSS. Por otra parte, con CSS se logra aumentar la densidad de las palabras claves dentro de los contenidos, ya que muchas de las etiquetas ocuparán muchísimo menos espacio. Esto también supone un menor peso para las páginas Web. El modo de funcionamiento de las CSS consiste en definir, mediante una sintaxis especial, la forma de presentación que se le aplicará a:

- Un documento web entero, de modo que se puede definir la forma de este de una sola vez.
- Un documento HTML o página, se puede definir la forma, en un pequeño trozo de código en la cabecera, a toda la página.

- Una porción del documento, aplicando estilos visibles en un trozo de la página. (4)

PHP

Fue elegido PHP por ser un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Posee ciertas características como:

- Es un lenguaje multiplataforma.
- Completamente orientado a la web.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
  Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones.
- Rapidez. PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Esta completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.

- Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros. (5)

JavaScript

JavaScript es un lenguaje de programación que sirve para mejorar el aspecto y la funcionalidad de una página web. En el caso del sistema implementado se utilizó para realizar validaciones de formularios. La característica principal de Javascript, de hecho, es la de ser un *lenguaje de scripting* por excelencia y, sin lugar a dudas, el más usado. Los script tienen capacidades limitadas, por razones de seguridad, por lo cual no es posible hacer todo con Javascript, sino que es necesario usarlo conjuntamente con otros lenguajes evolucionados, posiblemente más seguros, como Java. (6)

PostgreSQL

PostgreSQL es un avanzado gestor de bases de datos relacionales, es por ello que se decidió utilizarlo ya que se hace necesario tener almacenados los datos de las trazas de los diferentes servicios que serán procesadas. Además brinda una serie de facilidades como:

DBMS Objeto-Relacional: aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, optimización de consultas, herencia, y arreglos.

Altamente Extensible: soporta operadores, métodos de acceso y tipos de datos definidos por el usuario.

Soporte SQL Comprensivo: soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

Integridad Referencial: soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

API Flexible: ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

Lenguajes Procedurales: tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

Cliente/Servidor: usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL. (7)

Apache2

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows y Macintosh. Es usado primariamente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web. Apache es usado para tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable.
Entre sus principales características se encuentran:

- Modular
- Código abierto
- Multi-plataforma
- Extensible
- Popular (fácil conseguir ayuda/soporte) (8)

Syslog-ng

Syslog-ng, como su nombre indica, es una aplicación cuya utilidad principal es la de ser un reemplazo de syslog, pero con nuevas funcionalidades que se adapten a la nueva generación.
Desempeña todas las funciones del Syslog estándar, además de incorporar nuevas características que dejan atrás, algunas de las ineficiencias ya mencionadas con que contaba su predecesor. Algunas de estas ventajas se muestran a continuación.

La aplicación syslog original permite que los mensajes sean únicamente clasificados basándose en criterios de prioridad y facilidad, mientras que syslog-ng añade también, la posibilidad de clasificar basándose en los contenidos del mensaje haciendo uso de expresiones regulares.

El nuevo tema de configuración es potente e intuitivo, adelantando registros a través de TCP y recordándolos todos adelantando saltos; haciéndolo ideal para lugares con *firewall.*

El protocolo Syslog-ng posee numerosas ventajas con relación a su antecesor, el antiguo Syslog tradicional. Este nuevo protocolo ha demostrado mayor eficiencia desde el punto de vista de comunicación, seguridad e implementación, permitiendo que la tarea de recopilación, almacenamiento y análisis de archivos log sea cada vez menos compleja, facilitándose así la gestión de redes, por parte de los administradores de los sistemas. (9)

1.4.3 Herramientas para el desarrollo

Zend Studio

Es un IDE por sus siglas en inglés (Integrated Development Environment) destinado a desarrolladores profesionales. Es un IDE propietario, compatible con las plataformas Linux,,MAC y Windows. Incluye todos los componentes necesarios durante el ciclo de vida de una aplicación en PHP. Incluye editor, análisis, depuración, optimizadores de código y herramientas de base de datos. Zend Studio nos permite agiliza el desarrollo web y permite simplificar proyectos complejos.

- Características: excelente completamiento de código, coloreado en la sintaxis del código, administración avanzada de proyectos, múltiples lenguajes, incorpora el Framework de Zend, PHP Documentor, manual de PHP. Integración con subversión, los navegadores, integración avanzada con FTP. Soporte para Web Services, PHP4, PHP5 y SQL.
- Ventajas: agiliza el trabajo, cuenta con un buen Depurador e infinitas opciones que permiten un desarrollo profesional de aplicaciones. (10)

1.4.4 Fundamentación de la metodología utilizada

Lenguaje de Modelación Unificado (UML)

El Lenguaje de Modelado Unificado (UML:Unified Modeling Language) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80's y principios de los 90s. Es llamado un lenguaje de modelado, no un método. El mismo fusiona los conceptos de la orientación a objetos aportados por Booch, OMT y OOSE (Booch, G. et al., 1999). UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos. Los autores de UML apuntaron también al modelado de sistemas distribuidos y concurrentes para asegurar que el lenguaje maneje adecuadamente estos dominios.

UML cuenta con una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos.
Bloques de construcción del lenguaje:
Elementos: «estructurales, comportamiento, agrupación, anotación»
Relaciones: «dependencia, asociación, generalización, realización»
Diagramas: «clases, objetos, casos de uso, secuencia, colaboración, estados, actividades, componentes, despliegue» (11)

Proceso Unificado de Desarrollo (RUP)

El proceso unificado conocido como RUP, es un modelo de software que permite el desarrollo de software a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad. Esta metodología concibió desde sus inicios el uso de UML como lenguaje de modelado.

El proceso de desarrollo constituye un marco metodológico que define en términos de metas estratégicas, objetivos, actividades y artefactos (documentación) requerido en cada fase de desarrollo. Esto permite enfocar esfuerzo de los recursos humanos en términos de habilidades, competencias y capacidades a asumir roles específicos con responsabilidades bien definidas. Dicho proceso metodológico cuenta con tres fases fundamentales:

Fase de elaboración: En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en

esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

Fase de construcción: El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

Fase de transición: El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto. (12)

1.5 Conclusiones

En este capítulo se expone la importancia de incorporar al sistema de centralización, análisis y procesamiento las trazas de los servicios telemáticos en la Red UCf, módulos que permitan centralizar, procesar y analizar otros servicios y generar alarmas a partir de reglas definidas por el administrador demostrando las deficiencias asociadas a los métodos tradicionales utilizados para este fin. Además se analizan otras formas de procesamiento utilizadas en la Red UCf, determinándose la necesidad de desarrollar e implementar nuevos módulos para el sistema actual exponiendo las metodologías, herramientas y tecnologías a emplear en el desarrollo de la solución propuesta.

Capítulo II. Modelo del dominio y requisitos

En el presente capítulo se describe el análisis de la solución propuesta, utilizando UML como lenguaje de modelado y RUP como metodología, empleando el Modelo del Dominio para la captura de objetos; de esta forma se logra un mejor entendimiento del problema en cuestión. Además, se exponen los requerimientos funcionales y no funcionales del sistema, la descripción general del producto que se propone, así como el modelo de casos de uso del sistema y la descripción de cada uno de estos.

2.1 Descripción del modelo del dominio

"El modelo del domino captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos o mediante la entrevista con los expertos del dominio. El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema." (12)

Para una mayor comprensión del contexto en que se desarrolla el sistema se definen los principales conceptos relacionados con el entorno del problema.

2.2 Definición de las entidades y los conceptos principales

En el modelo de dominio se definen las siguientes clases: Servidor, Servicio, Traza, Script, Datos Almacenados, Criterios de Configuración, Administrador, Resultado.

Servidor: computadora que maneja peticiones de datos, correo electrónico, servicios de redes y transferencia de archivos brindando así, servicios a otras computadoras (clientes).

Servicio: Utilidad brindada por los servidores (servicio Web, servicio Proxy, etc.). Sus eventos y acciones quedan almacenados en las trazas.

Traza: Son el primer registro de la actividad en los sistemas y redes. Contienen el registro de acciones y eventos que tienen lugar en un sistema. Son una parte

primaria de la seguridad y pueden ser usados en la detección de ataques e intrusiones, así como en el análisis de fallas de hardware y software.

Script: Utilidad o programa ejecutado por un intérprete que permite la automatización de tareas. Es el encargado de analizar las trazas y emitir los datos almacenados.

Datos Almacenados: Es toda la información que el Script extrae de las trazas y almacena en una base de datos.

Criterios de Configuración: Conjunto de patrones definidos por el Administrador para analizar los datos.

Administrador: Es quien analiza los Datos Almacenados y según los Criterios de Configuración definidos con anterioridad por el mismo emite los resultados.

Resultado: muestra las acciones y el estado de los Servicios que tienen lugar en la red LAN.

2.3 Reglas del negocio a considerar

Un Servidor puede ofrecer uno o más Servicios, según sus características. Atendiendo a las especificidades de la red puede existir más de un Servidor brindando Servicios del mismo tipo.

- En una red LAN puede haber tantos Servicios como requieran los usuarios de la misma, los Servicios serán ofrecidos por los Servidores.
- Las trazas sólo serán emitidos por los Servicios y sólo serán analizados por los Scripts.
- Los Scripts sólo analizarán el contenido de las trazas generadas por los Servicios sin alterarlo; para de esta forma emitir los Datos Almacenados.
- Los Datos Almacenados no pueden ser alterados.
- El Administrador sólo analizará la información contenida en los Datos Almacenados.
- Los Criterios de Configuración son patrones definidos únicamente por el Administrador.
- El Resultado sólo será emitido por el Administrador.

2.4 Representación del modelo del dominio

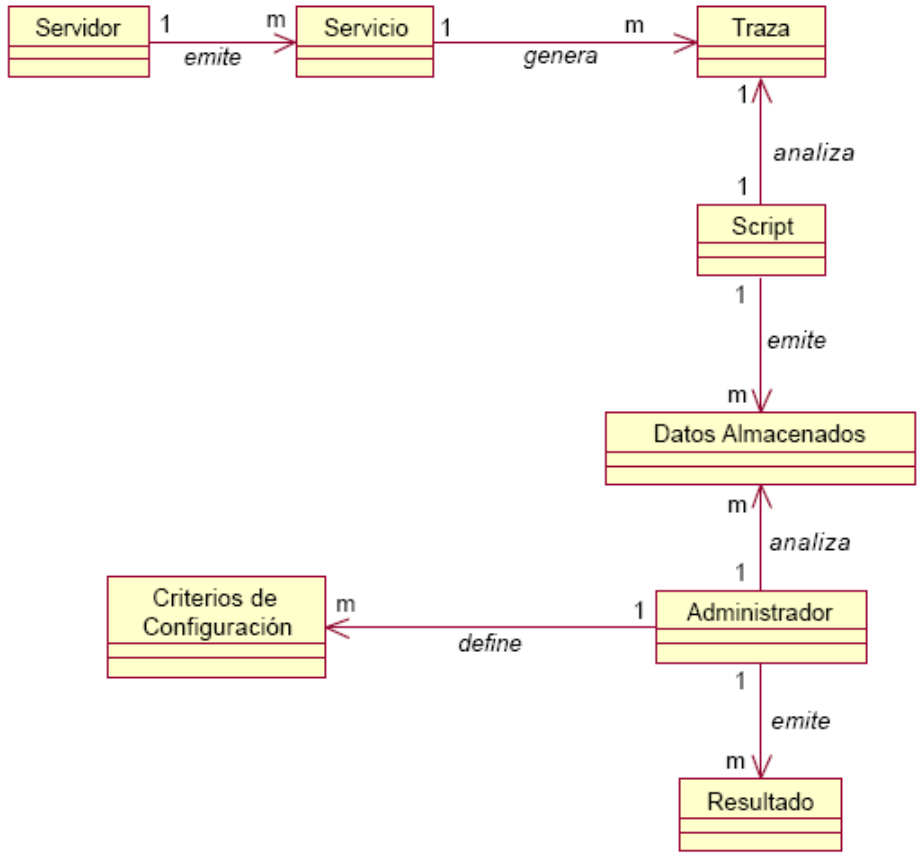


Figura 1. Diagrama de clases del modelo de objetos del dominio.

2.5 Descripción del sistema propuesto

En este epígrafe será detallado el sistema, describiendo su concepción, así como la instalación y configuración del mismo.

2.5.1 Concepción general del sistema

La incorporación de nuevos módulos al sistema para la centralización, análisis y procesamiento de trazas de servicios telemáticos para la Red UCf, surge por la necesidad de contar con un sistema que además de analizar y procesar las trazas de los servicios Cisco y Apache2, analice y procese las trazas de los servicios Squid y Radius mediante el uso de reportes y que brinde la posibilidad al administrador de configurar reglas que permitan la generación de

alarmas con determinado nivel de prioridad, ayudando así a garantizar los niveles de funcionalidad, operatividad y seguridad esperados.

El sistema estará compuesto por un módulo de captura de la información contenida en las trazas, que se encargará de filtrar las trazas de los servicios Squid y Radius y de almacenar su contenido en una base de datos y por un módulo de pos procesado que se encargará de la generación de reportes, a partir de los datos almacenados en la base de datos, y de la generación de alarmas a partir de reglas definidas por el administrador, mostrando así el funcionamiento de cada servicio de la red y de la red en general, así como posibles anomalías que pueden suceder de acuerdo a las reglas definidas y que según la gravedad del evento el administrador será informado mediante un correo electrónico o un SMS. Con este sistema serán eliminadas las deficiencias existentes en el sistema de análisis y procesamiento de trazas que se utiliza actualmente.

2.5.2 Instalación y configuración del sistema.

2.5.2.1 Centralización de las trazas.

Esta etapa constituye el pilar del sistema, de ahí la importancia de una correcta instalación y configuración de la misma. Para esto se debe:

1. Instalar servidor para centralizar las trazas de los servicios telemáticos.

En el caso de Ubuntu GNU/Linux distribución Hardy se emplea el siguiente comando:

apt-get install syslog-ng

Una vez instalado el Syslog-ng, se procede a la configuración del mismo.

2. Configuración del servidor de trazas centralizadas.

Para configurar el Syslog-ng se debe editar el fichero /etc/syslog-ng.conf, esta configuración cuenta con varios parámetros como son:

i) Options: Define las propiedades globales del fichero de configuración.

ii) Source: Define la fuente de donde provienen los mensajes, syslog-ng acepta mensajes a través de la red, utilizando los protocolos TCP o UPD.

El formato es:

source <sourcename> { sourcedriver params; sourcedriver params; ... };

donde *sourcename* es un identificador y *sourcedriver* es el método usado para recibir estos mensajes.

iii) Destination: Define donde serán escritos los mensajes.

El formato es:

destination <destname> { destdriver params; destdriver params; ... ; }; donde *destname* es un identificador y *destdriver* es el método usado para guardar los mensajes.

iv) filtering rules: reglas de filtrado (palabra clave: *filter*)

El formato es:

filter <filtername> { expression; };

Donde *filtername* es el nombre del filtro y *expression* es una expresión lógica simple. Se puede usar "and", "or" y "not" para conectar las funciones incorporadas.

Las funciones pueden ser:

facility: una lista separada por comas de nombres de facilities.

level: una lista separada por comas o un rango separado por "." de nombres de priority.

program: una expresión regular para buscar coincidencias con nombres de programas.

host: una expresión regular para buscar coincidencias con nombres de Host.

match: una expresión regular que busca coincidencias en el mensaje de Syslog en sí.

v) Log: Conecta los campos de Source y Destination. Une la fuente y el destino de las trazas. (Log = Source + Destination).

El formato es:

log { source S1; source S2; ... filter F1; filter F2; ... destination D1; destination D2;

... };

Donde Sx se refiere a uno de los orígenes de log declarados, Fx a uno de los filtros y Dx a uno de los destinos.

Ejemplo:

source net { udp(ip(0.0.0.0) port(514)); }; Escucha en todos las interfaces del servidor.

destination d_switch01 { file("/var/log/cisco/cisco3550.log"); }; Indica el camino del fichero destino para almacenar las trazas.

filter f_switch01 { host("192.168.0.1"); }; Filtra todos los paquetes que en el campo host tienen la dirección IP 192.168.0.1

log { source(net); filter(f_switch01); destination(d_switch01); }; Crea la traza que se forma con los paquetes recibidos por cualquier interface y el puerto 514 del servidor de Syslog, que además en el campo Host del paquete Syslog tienen la IP 192.168.0.1 y ellos se almacenarán en el fichero cisco3550.log indicado en destination.

Una vez configurado el servidor syslog-ng, se procede a configurar los clientes.

3. Configuración de los clientes Syslog-ng.

Se denomina clientes de Syslog a los Hosts y dispositivos de red que utilizando este protocolo enviarán los mensajes al servidor central de trazas. En este caso fueron configurados los servicios Squid y Radius.

Ejemplo de configuración para Squid

Configurar el Servidor Squid para que envíe las trazas a un servidor Syslog resulta relativamente fácil. Generalmente Squid emplea en su fichero de configuración denominado squid.conf la directiva access_log para especificar dónde almacenar las trazas, siendo comúnmente la configuración por defecto la siguiente:

access_log /var/log/squid/squid.log squid

Para permitir que squid envíe las trazas a syslog se necesita modificar esta directiva, sin embargo en este caso se incorpora una nueva directiva a fin de mantener ambas posibilidades, o sea, el envío a un fichero local y al servicio syslog. Para ello se incorpora la siguiente línea en el fichero de configuración:

access_log syslog squid

Una vez modificado el squid.conf se requiere editar el syslog-ng.conf en el servidor local y adicionar las siguientes líneas.

 # Filtro para eliminar todas las entradas que lleguen desde el programa
#squid para evitar que se registren dos veces.

```
filter f_remove { program("squid"); };

filter f_user { facility(user); };

# La traza se registrará en un fichero local
destination df_user { file("/var/log/user.log"); };

# La traza también se enviará por la red vía UDP
destination logserver { udp("servidor-de-syslog-de-la-red"); };

# Directiva
log {
   # registrar todas las fuentes
   source(s_all);
   # aplicar el filtro 'f_user'
   filter(f_user);
   # aplicar el filtro 'f_remove'
   filter(f_remove);
   #enviar la traza a un fichero local
   destination(df_user);
   #enviar la traza a un servidor central
   destination(logserver);
};
```

Ejemplo de configuración para Radius

En el caso del servicio Radius resulta similar, se requiere configurar la directiva *log_dir* en el fichero de configuración radiusd.conf, declarando que envíe las trazas al servicio syslog, para luego configurar, de manera similar a Squid, que el syslog envíe las trazas a un servidor central de trazas que emplea igualmente syslog.

2.5.2.2 Módulo de captura de la información almacenada en las trazas.

1. Implementación de los Scripts de captura

Los Scripts de captura fueron desarrollados utilizando Python como lenguaje de programación. Los mismos establecen una conexión con el servidor de base de datos y leen línea a línea el contenido de las trazas y van realizando operaciones como: el filtrado y almacenamiento de la información en la base de datos.

2. Configuración del Cron

Al fichero /etc/crontab2 se le adicionan los Scripts encargados de filtrar y almacenar el contenido de las trazas. Para ello es empleada la siguiente configuración:

Minuto Hora DiaDelMes Mes DiaDeLaSemana Intérprete Script

2.5.2.3 Módulo de pos procesado.

1. Copiar la herramienta Web al directorio /var/www del servidor Web y configurar el fichero globalvars.php.

2. Entrar al sistema a través de un navegador Web y añadir los servidores y dispositivos de red, así como los servicios telemáticos cuyas trazas van a ser analizadas. Empleando para ello los criterios de configuración.

3. Agregar los criterios de configuración, como son: puertos, direcciones IP, patrones, URL, números de teléfono, etc.

2.5.3 Requerimientos funcionales

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. Por lo general se describen mejor a través del modelo de Casos de uso y los Casos de uso como tal. Por lo tanto los requerimientos funcionales especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto.

Para la determinación de varios de los requerimientos funcionales que aquí se plantean, se realizaron entrevistas a los administradores de red de la Universidad de Cienfuegos y de otros centros, a fin de tener una visión, lo más cercana posible a la realidad con respecto a los principales problemas e

insuficiencias existentes en el análisis y procesamiento de las trazas. Además de las sugerencias aportadas por los mismos en cuanto a posibles soluciones desde el punto de vista teórico, las que se tuvieron en cuenta a la hora automatizar las funciones.

Los requerimientos funcionales del sistema propuesto son los siguientes:

1. Filtrar y almacenar el contenido de las trazas del servicio Squid.
2. Filtrar y almacenar el contenido de las trazas del servicio Radius.
3. Importar puertos por lote.
4. Importar direcciones ips por lote.
5. Agregar números telefónicos.
6. Listar números telefónicos.
7. Modificar números telefónicos.
8. Eliminar números telefónicos.
9. Agregar direcciones de correo.
10. Listar direcciones de correo.
11. Modificar direcciones de correo.
12. Eliminar direcciones de correo.
13. Determinar accesos a sitios específicos.
14. Determinar accesos a sitios peligrosos.
15. Determinar accesos a direcciones IP peligrosas.
16. Determinar URL con patrones peligrosos.
17. Determinar URL peligrosas.
18. Determinar accesos de usuarios específicos.
19. Determinar accesos desde direcciones IP específicas.
20. Determinar accesos según usuarios, método y URL.
21. Determinar accesos por usuarios, palabras claves, IPs, dominios.
22. Acceso desde un número telefónico específico.
23. Acceso a una IP específica desde un número telefónico.
24. Accesos desde números telefónicos específicos.
25. Accesos desde un puerto específico.
26. Determinar paquetes con direcciones IP de origen prohibidas o peligrosas.
27. Determinar paquetes con direcciones de IP de destino prohibidas o

Peligrosas.

28. Determinar paquetes con puertos de origen prohibido o peligroso.
29. Determinar paquetes con puertos de destino prohibido o peligroso.
30. Determinar accesos mediante un método específico.
31. Determinar accesos mediante un código de estado específico.
32. Determinar cantidad de megas transferidos.
33. Crear reglas para configurar alarmas por servicios
34. Eliminar reglas por servicio.
35. Crear reglas para uno o todos los servicios cisco.
36. Almacenar alarmas por servicio.
37. Eliminar alarmas por servicio.
38. Listar alarmas.
39. Listar alarmas por categoría.
40. Determinar cantidad de alarmas por categoría.
41. Determinar cantidad total de alarmas.
42. Validar falsos positivos.
43. Mostrar detalles de las alarmas.

2.5.4 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener (entiéndase por propiedades las características que hacen al producto atractivo, usable, rápido o confiable) como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad.

Los requerimientos no funcionales del sistema propuesto son los siguientes:

Requerimientos de apariencia o interfaz externa

El sistema debe tener una interfaz sencilla, legible y simple de usar, el producto debe ser profesional, autoritario e interactivo para que el usuario se sienta confiado. Se deben brindar opciones de navegabilidad al usuario, como información sobre dónde está y todo lo que puede hacer desde ese lugar, a través de señales de navegación. El contenido debe ser mostrado de manera comprensible para el usuario.

Requerimientos de Usabilidad

El sistema estará dirigido a administradores y a operadores de redes para facilitarles el estudio de las características del tráfico cursado por sus redes, ayudándoles a conocer el uso que se está dando a sus infraestructuras e identificando las fuentes de los problemas de rendimiento que aparecen. Por lo tanto el sistema está diseñado para ser utilizado por personas con conocimientos en el manejo de redes de computadora y el ambiente Web en sentido general.

Requerimientos de Rendimiento

El sistema deberá responder de forma eficaz y eficiente ante las solicitudes de los usuarios. Deberá ser preciso y exacto a la hora de realizar los cálculos. La velocidad de respuesta a las peticiones de reportes debe ser normal. La eficiencia de la aplicación estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor, y la velocidad de las consultas a la base de datos. Se realizará la validación de los datos y la manipulación de eventos en el cliente y en el servidor aquellas que por cuestiones de seguridad, o de acceso a los datos lo requieran. Lográndose así un tiempo de respuesta más rápido, una mayor velocidad de procesamiento y un mayor aprovechamiento de los recursos.

Requerimientos de Soporte

Una vez terminado el desarrollo del sistema se realizarán pruebas en distintas redes Lan-IP para observar su comportamiento. La aplicación será documentada para garantizar el soporte de la herramienta. Se realizará mantenimiento al sistema a fin de aumentar las funcionalidades del mismo a través de versiones posteriores. Los servicios de instalación y mantenimiento del mismo serán responsabilidad del administrador del sistema en la entidad que sea utilizado.

Requerimientos de Portabilidad

El producto podrá ser usado bajo los sistemas operativos *GNU/Linux y Windows 95* o superior.

Requerimientos Políticos-culturales

La aplicación debe cumplir con las regulaciones políticas de la entidad que utilice el sistema.

Requerimientos de Confiabilidad

El sistema deberá ser robusto, no presentar fallos frecuentes y/o severos, y en caso de ocurrir deberá recuperarse rápidamente. Las operaciones a realizar deben ser transaccionales.

Requerimientos de Ayudas y Documentación en línea.

El sistema tendrá una ayuda donde se realizarán aclaraciones sobre las opciones del sistema, que garantizarán el buen desempeño de los usuarios a la hora de interactuar con el mismo.

Requerimientos de Software

La aplicación debe poderse ejecutar en entornos *Unix y GNU/Linux.* (Multiplataforma), para su ejecución del lado del servidor necesita preferentemente GNU/Linux como sistema operativo, PostgreSQL como sistema gestor de base de datos, Apache como servidor Web y Python para ejecutar los Scripts del módulo de captura de la información contenida en las trazas. Los servicios telemáticos deben estar configurados para garantizar la centralización de las trazas a través del Syslog-ng. Del lado del cliente cualquiera de los navegadores Web existentes en el mercado con la opción de Java Script activada.

Requerimientos de Hardware

Se requiere de una computadora como servidor de base de datos con PostgreSQL como gestor de base de datos, con una capacidad de memoria RAM no menor de 384 Mbyte, capacidad de almacenamiento proporcional a la cantidad de trazas que se generan y el tiempo de almacenamiento exigido (se

recomienda como mínimo 80 GByte de almacenamiento), microprocesador Pentium III a 700 MHz como mínimo. Las terminales clientes sólo requerirán de una computadora conectada a la red, para poder ejecutar los navegadores Web, las mismas deben cumplir los requisitos mínimos de hardware del sistema operativo que se emplee.

Requerimientos de Seguridad

La información estará protegida contra accesos no autorizados utilizando mecanismos de autenticación y autorización que puedan garantizar el cumplimiento de esto: cuenta, contraseña; de manera que sólo el administrador pueda acceder a la información brindada por el sistema, garantizando así la confidencialidad. Además para reforzar la seguridad en el acceso se configurarán reglas en el cortafuego (Firewall) del servidor donde se instale el sistema, permitiéndole acceder al mismo solo a las estaciones de trabajo de los administradores de la red que lo utilizarán. Se usarán mecanismos de encriptación de los datos que por cuestiones de seguridad no deben viajar al servidor en texto claro, para ello se empleará SSL en la comunicación cliente servidor. Los datos sensibles se almacenarán de forma encriptada en la base de datos utilizando para ello MD5 como algoritmo de encriptación. Se implementarán sesiones para evitar el acceso al sistema sin pasar por el mecanismo de validación o autenticación. Se harán validaciones de la información tanto en el cliente como en el servidor. No obstante, el administrador accederá de manera rápida y operativa al sistema sin que los requerimientos de seguridad se conviertan en un retardo para él.

2.6 Modelo de casos de uso del sistema

2.6.1 Actores del modelo del sistema

Un actor no es más que un conjunto de roles que los usuarios de Casos de Uso desempeñan cuando interaccionan con estos Casos de Uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Una vez identificado los actores del sistema, queda identificado el entorno externo del sistema. (12)

Nombre del	Descripción

Actor	
Administrador	Es quien utiliza el sistema para conocer parámetros como estabilidad, disponibilidad y seguridad de la red. Realiza tareas como la gestión de servidores y/o dispositivos de red, de los servicios telemáticos, del formato de salida de los reportes; además procesa el contenido de las trazas de los servicios, en busca de situaciones críticas. Define los criterios de configuración y genera los reportes.
Syslog_ng	Es el Software encargado de centralizar en un servidor, los mensajes generados por los servicios telemáticos Radius y Squid.
Cron del Sistema Operativo	Es la utilidad del Sistema Operativo, encargada de ejecutar los Scripts que almacenan el contenido de las trazas, en la base de datos del sistema

Tabla 1. Actores del Sistema.

2.6.2 Paquetes y sus relaciones

Figura 3. Diagrama de casos de uso del paquete de captura de la información contenida en las trazas.

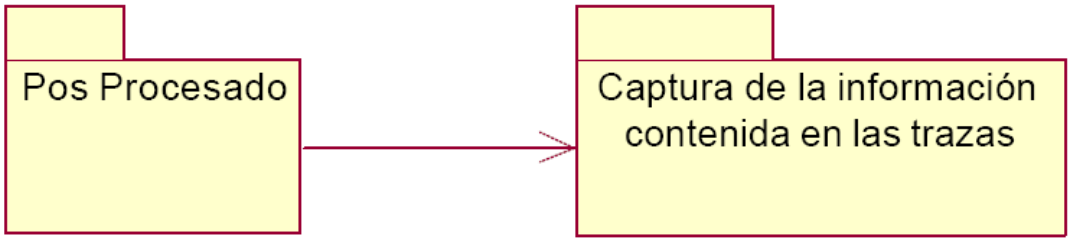


Figura 2. Paquetes y sus relaciones

2.6.3 Casos de uso del sistema

"La forma en que los actores usan el sistema se representa con un caso de uso. Los casos de uso son "fragmentos" de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia." (12)

Casos de uso del paquete de captura de la información contenida en las trazas.

1. Filtrar y almacenar la información de las trazas.

Casos de uso del paquete de pos procesado

1. Gestionar reglas por servicio.
2. Gestionar alarmas.
3. Gestionar envío de alarmas.
4. Gestionar criterios de configuración.
5. Procesar los datos almacenados de las trazas del servicio Radius.
6. Procesar los datos almacenados de las trazas del servicio Squid.

2.6.4 Diagramas de casos de uso del sistema

"EL modelo de casos de uso permite que los desarrolladores de Software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores, y proporciona la entrada fundamental para el análisis, el diseño y las pruebas." (12)

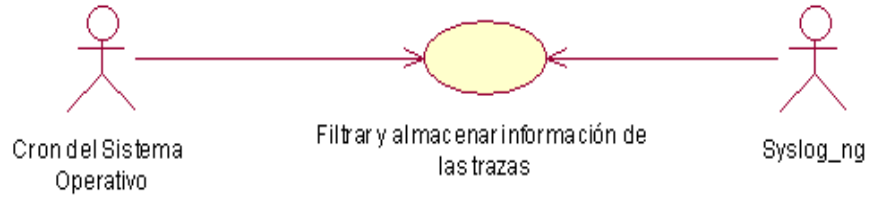


Figura 3. Diagrama de casos de uso del paquete de captura de la información contenida en las trazas.

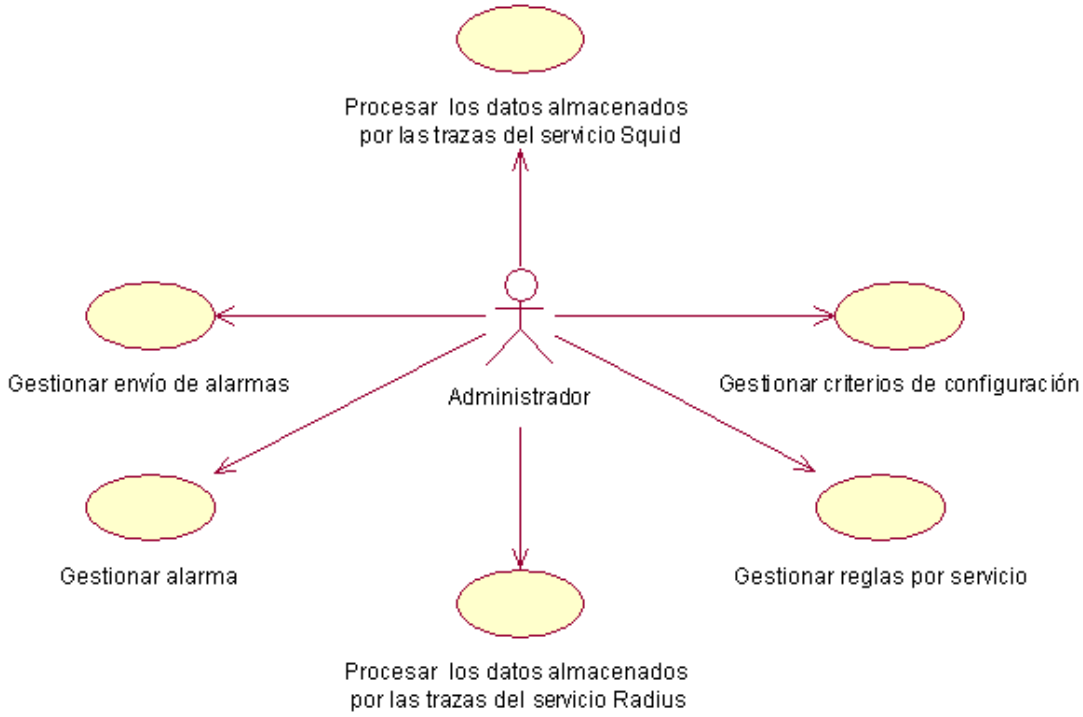


Figura 4. Diagrama de casos de uso del paquete de pos procesado.

2.6.5 Descripción de los casos de uso del sistema

2.6.5.1 Descripción de los casos de uso del módulo de pos procesado

Caso de Uso:	Gestionar reglas por servicio.
Actores:	Administrador (inicia).
Propósito:	Definir reglas por servicio que permitan informar acerca de situaciones alarmantes.
Resumen:	

El caso de uso se inicia cuando el administrador define o elimina reglas que permitan analizar los datos almacenados de los servicios telemáticos e informar en caso que ocurra un evento alarmante culminando así el caso de uso. Para la configuración de reglas el administrador seleccionará el servicio para el cual confeccionará la regla (en el caso del servicio cisco podrá definir una regla para un cisco en específico o indicar que la regla creada se aplicará a todos los cisco) y para cada regla deberá definir una prioridad que puede ser alta, media o baja, una breve descripción de la regla y el número de falsos positivos permisibles para dicha regla. Una vez definida la regla el sistema realiza la búsqueda en los datos almacenados para generar alarmas en caso de alguna irregularidad, si la prioridad de la alarma es alta o media además de mostrarla en pantalla se envía un correo y/o un SMS al administrador indicando la cantidad de alarmas detectadas.	
Referencias:	R13, R14, R15, R16, R17, R20, R21, R23, R24, R26, R27, R28, R29, R33, R34, R35.
Precondiciones:	Deben existir reglas definidas para poder detectar alarmas.
Poscondiciones:	Quedan definidas las reglas mediante las cuales se va a procesar la información obtenida de las trazas.
Requisitos especiales:	-
Prototipo:	Anexo

Tabla 2. Descripción del caso de uso de sistema: Gestionar reglas por servicio.

Caso de Uso:	Gestionar alarmas.
Actores:	Administrador (inicia).
Propósito:	Conocer detalles de situaciones alarmantes que puedan surgir.
Resumen: El caso de uso se inicia cuando el administrador analiza o elimina alarmas que sean generadas a partir de las reglas definidas previamente culminando así el caso de uso. Para realizar el análisis de las alarmas el administrador debe	

seleccionar un servicio, además puede elegir qué tipo de alarmas desea chequear: las de prioridad alta, media o baja, o puede seleccionar todas las alarmas de un determinado servicio. Por cada servicio se muestra la cantidad de alarmas que existen según la prioridad. De cada alarma se puede conocer los detalles de la traza y marcar dicha alarma si el administrador considera que la misma es un falso positivo.	
Referencias:	R36, R37, R38, R39, R40, R41, R42, R43.
Precondiciones:	Deben estar almacenados los datos de los diferentes servicios, así como las reglas a partir de las cuales se generarán las alarmas en caso de existir anomalías.
Poscondiciones:	Son mostradas las alarmas según el servicio.
Requisitos especiales:	-
Prototipo:	Anexo

Tabla 3. Descripción del caso de uso de sistema: Gestionar alarmas.

Caso de Uso:	Gestionar envío de alarmas.
Actores:	Administrador (inicia).
Propósito:	Configurar las direcciones de correo para el envío de alarmas.
Resumen: El caso de uso se inicia cuando el administrador añade, modifica o elimina las direcciones de correo para el envío de alarmas; culminando así el caso de uso. Cuando se añade o modifica un usuario se debe especificar qué tipo de alarmas serán enviadas: las altas o las altas y las medias.	
Referencias:	R9, R10, R11, R12.
Precondiciones:	Deben definirse usuarios relacionados con el procesamiento de la información generada por los diferentes servicios.
Poscondiciones:	Quedan definidos los usuarios a los cuales se les

	informará la ocurrencia de alguna alarma.
Requisitos especiales:	-
Prototipo:	Anexo

Tabla 4. Descripción del caso de uso de sistema: Gestionar envío de alarmas.

Caso de Uso:	Gestionar criterios de configuración.
Actores:	Administrador (inicia).
Propósito:	Definir los criterios que le van a indicar al sistema cómo tiene que clasificar y procesar la información capturada de la red.
Resumen: El caso de uso se inicia cuando el administrador define, modifica o elimina los criterios por los cuales se procesan y clasifican los datos almacenados en la base de datos del sistema sobre el estado de los servicios telemáticos, dispositivos de red y/o servidores; culminando así el caso de uso. Los criterios a gestionar son: puerto, dirección ip, url, extensión, patrón, números de teléfonos, importar puertos por lote, importar direcciones ip por lote. Para el caso de los números telefónicos, se introduce al sistema el usuario y el o los números de teléfono, en el caso de importar puertos o ip por lote se debe indicar la dirección del fichero que contiene los valores; dicho fichero debe tener como estructura el puerto o la dirección ip y una descripción que estará separada del valor por una coma. El sistema realiza la búsqueda de estos patrones en los datos almacenados para generar reportes y alarmas en caso de encontrar alguna irregularidad.	
Referencias:	R3, R4, R5, R6, R7, R8.
Precondiciones:	Deben existir criterios mediante los cuales se pueda procesar el contenido de las trazas.
Poscondiciones:	Quedan definidos los criterios por los cuales se

	va a procesar la información obtenida de las trazas.
Requisitos especiales:	-
Prototipo:	Anexo

Tabla 5. Descripción del caso de uso de sistema: Gestionar criterios de configuración.

Caso de Uso:	Procesar los datos almacenados de las trazas del servicio Radius.
Actores:	Administrador (inicia).
Propósito:	Procesar los datos almacenados provenientes de las trazas del servicio Radius, en busca de información de interés.
Resumen: El caso de uso se inicia cuando el administrador analiza los datos almacenados provenientes de las trazas del servicio Radius, en busca de información como: accesos desde direcciones ip peligrosas, puertos peligrosos, un usuario o un número de teléfono específico. Para la realización de alguna de estas tareas, es seleccionada la opción correspondiente y se acota la búsqueda por fechas. El caso de uso culmina cuando es mostrado el reporte con los resultados obtenidos.	
Referencias:	R18, R19, R22, R25.
Precondiciones:	Deben estar almacenados los datos provenientes del servicio Radius, en el sistema, así como los criterios de configuración necesarios para el análisis de la información referente al servicio.
Poscondiciones:	Son mostrados los valores antes mencionados.
Requisitos especiales:	-
Prototipo:	Anexo

Tabla 6. Descripción del caso de uso de sistema: Procesar los datos almacenados de las trazas del servicio Radius.

Caso de Uso:	Procesar los datos almacenados de las trazas del servicio Squid.
Actores:	Administrador (inicia).
Propósito:	Procesar los datos almacenados provenientes de las trazas del servicio Squid.
Resumen: El caso de uso se inicia cuando el administrador analiza los datos almacenados provenientes de las trazas del servicio Squid, en busca de información como: acceso desde direcciones ip peligrosas, accesos mediante un método o un código de estado específico, cantidad de megas transferidos. Para la realización de alguna de estas tareas, es seleccionada la opción correspondiente y se acota la búsqueda por fechas. El caso de uso culmina cuando es mostrado el reporte con los resultados obtenidos.	
Referencias:	R19, R30, R31, R32.
Precondiciones:	Deben estar almacenados los datos provenientes del servicio Squid, en el sistema, así como los criterios de configuración necesarios para el análisis de la información referente al servicio.
Poscondiciones: **Requisitos especiales:** **Prototipo:**	Son mostrados los valores antes mencionados. - Anexo

Tabla 7. Descripción del caso de uso de sistema: Procesar los datos almacenados de las trazas del servicio Squid.

2.6.5.2 Descripción de los casos de uso del módulo de captura de la información contenida en las trazas.

Caso de Uso:	Filtrar y Almacenar la información contenida en las trazas.
Actores:	Cron del Sistema Operativo (inicia).

Propósito:	Capturar la información contenida en las trazas de los servicios Radius y Squid.
Resumen: El caso de uso se inicia, cuando el Cron del sistema operativo ejecuta, con la periodicidad establecida por el administrador, los Scripts encargados de filtrar la información contenida en las trazas de los servicios telemáticos, como son las de los servicios Radius y Squid. Para esto primero se abre la traza, se identifica a que servicio pertenece, y se lee línea a línea su contenido, separando el mismo según su tipo; (Ej: fecha, puerto, usuario, teléfono, etc.). Esta información es almacenada en la base de datos del sistema. Culminando así el caso de uso.	
Referencias:	R1, R2.
Precondiciones:	Deben estar ejecutándose los servicios telemáticos, y centralizadas sus trazas en el servidor de trazas.
Poscondiciones:	Queda filtrada y almacenada en la base de datos del sistema, la información que estaba contenida en las trazas de los diferentes servicios telemáticos.
Requisitos especiales:	-
Prototipo:	Anexo

Tabla 8. Descripción del caso de uso de sistema: Filtrar y Almacenar la información contenida en las trazas.

2.7 Conclusiones

Con el desarrollo de este capítulo queda descrita la solución propuesta y el sistema con sus requerimientos funcionales y no funcionales, utilizando UML como lenguaje de modelado y RUP como metodología. Además se realizó una descripción detallada del modelo de sistema a través de la definición de sus actores, paquetes y casos de usos, mostrándose para estos los diagramas de casos de uso.

Capítulo III: Construcción y validación de la solución propuesta

En el presente capítulo se realiza una descripción de la construcción de la solución propuesta, para ello se han utilizado los diagramas de clases Web del diseño y el modelo lógico y físico de la base de datos. Se describen los principios de diseño utilizados, mostrando ejemplos de cómo son presentados al usuario y la concepción general de la ayuda. Para describir los elementos fundamentales de la implementación se muestra el diagrama de implementación. Posteriormente se describen las pruebas y validaciones realizadas al sistema así como la realización del análisis de la factibilidad para la aplicación desarrollada, utilizando como metodología COCOMO II.

3.1 Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de Software, de las interfaces, así como sus relaciones en una aplicación. En el caso de las aplicaciones Web, el diagrama de clases representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase.

A continuación se presentan los diagramas de clases Web que describen los casos de uso del sistema.

Caso de Uso	Diagrama de Clases
Filtrar y almacenar la información de las trazas.	Anexo 7

Tabla 9. Diagramas de clases Web. Paquete de captura de la información contenida en las trazas.

Casos de Uso	Diagrama de Clases
Gestionar reglas por servicio	Anexo 9
Gestionar alarmas	Anexo 10
Gestionar envío de alarmas	Anexo 13
Gestionar criterios de configuración	Anexo 8
Procesar datos almacenados de las trazas del servicio Radius	Anexo 11

Procesar datos almacenados de las trazas del servicio Squid	Anexo 12

Tabla 10. Diagramas de clases Web. Paquete de pos procesado.

3.2 Diseño de la base de datos

3.2.1 Modelo lógico de datos

El objetivo fundamental del modelo lógico de datos es brindar una vista de las entidades lógicas de datos y sus relaciones con independencia de la plataforma de base de datos a utilizar. Por lo general el modelo se construye en 3FN y es una representación gráfica del diagrama de clases persistentes.

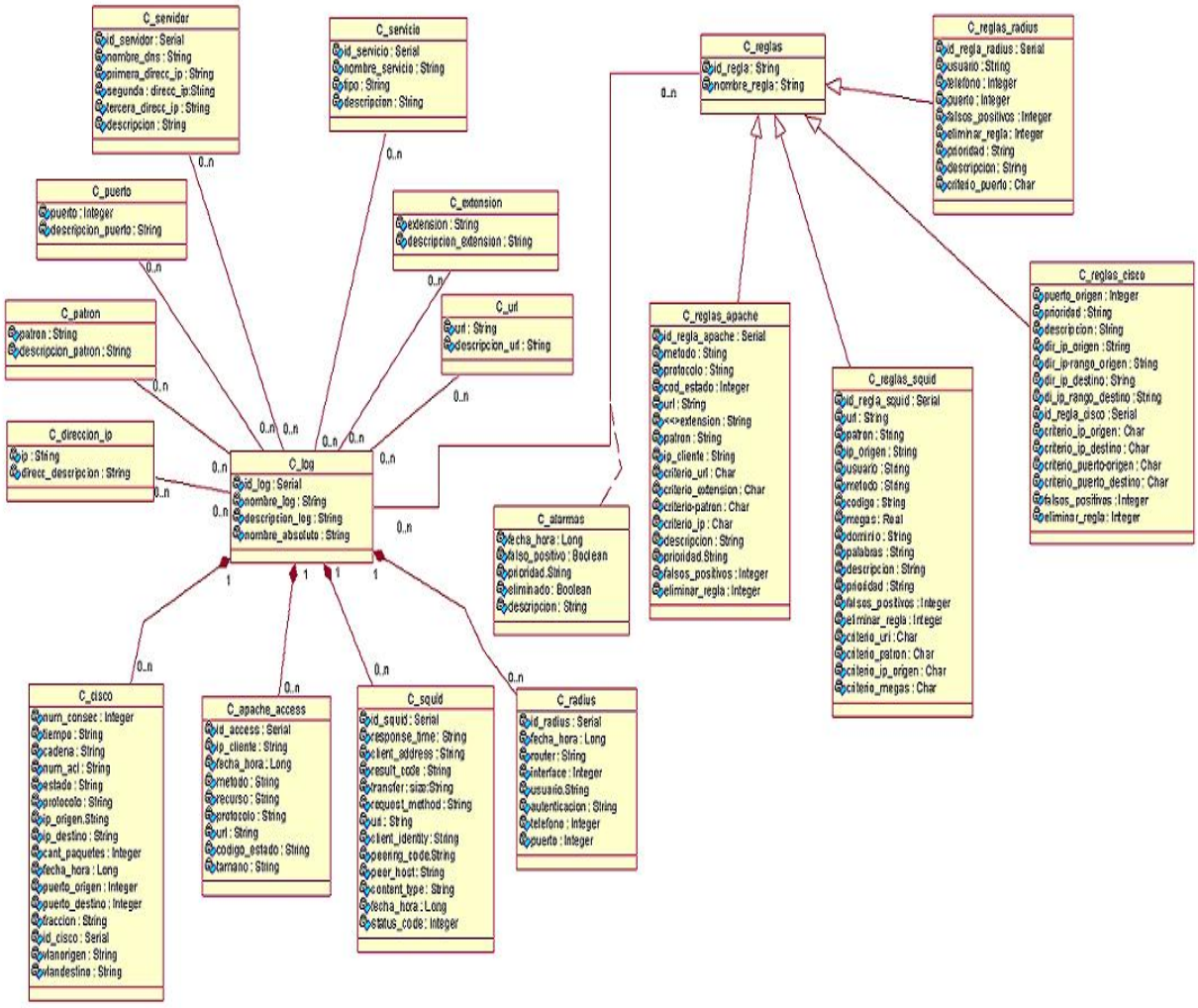


Figura 5. Diagrama de clases persistentes para la base de datos de servicios.

3.2.2 Modelo físico de datos

Describe la representación lógica y física de los datos persistentes (la representación de las tablas que conforman la base de datos del sistema, obtenido a partir del diagrama de clases persistentes). Es el artefacto resultante de la actividad de diseño de base de datos.

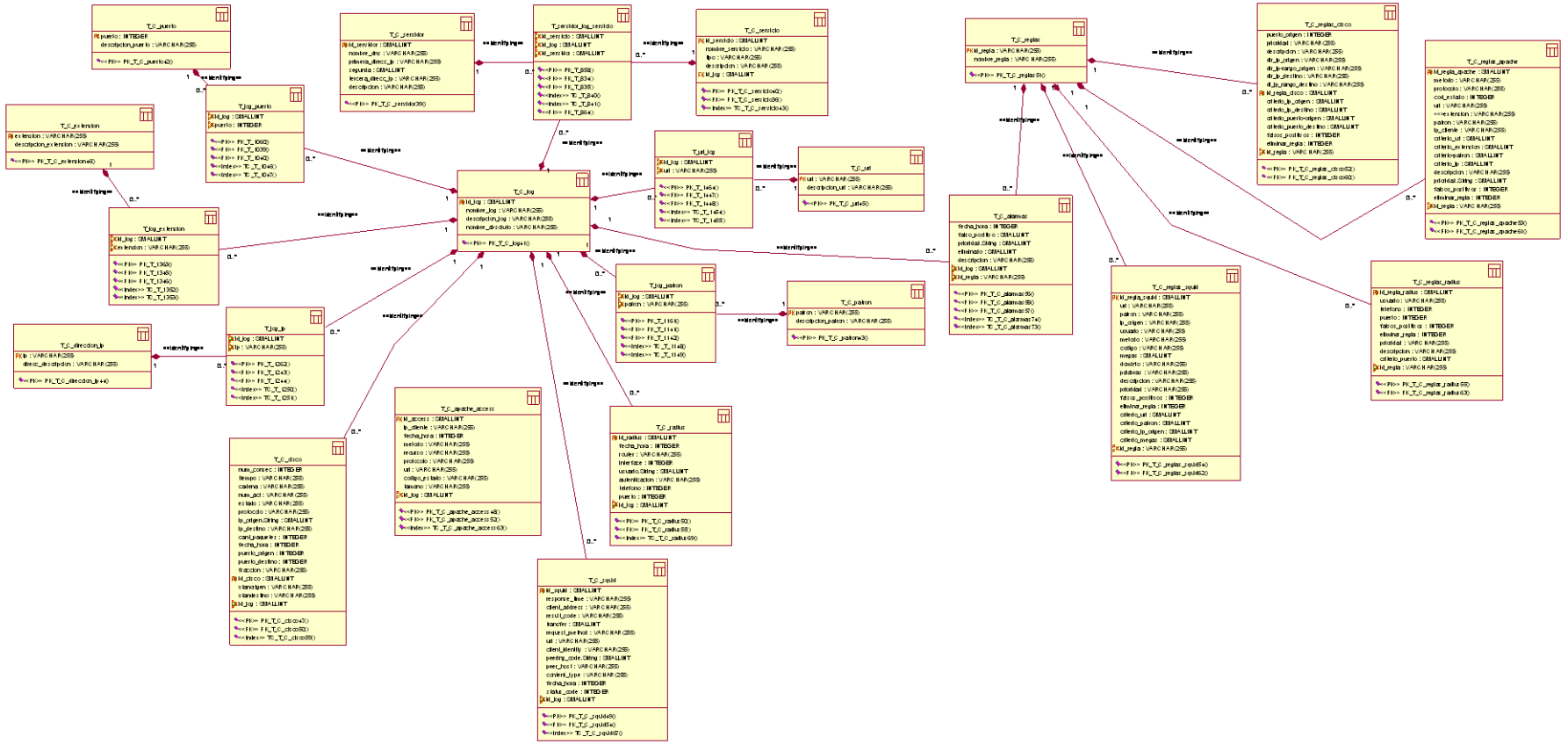


Figura 6. Diagrama del modelo físico de datos para la base de datos de servicios.

3.3 Diagrama de implementación

Con el diagrama de implementación se describen los componentes y la organización de acuerdo a los nodos, así como las dependencias entre ellos, por lo que en una sola vista se pueden apreciar los Diagramas de Componente y de Despliegue. Aunque se pueden construir por separado estos diagramas, en ellos no queda clara la relación directa que puede darse entre componentes ubicados en nodos diferentes.

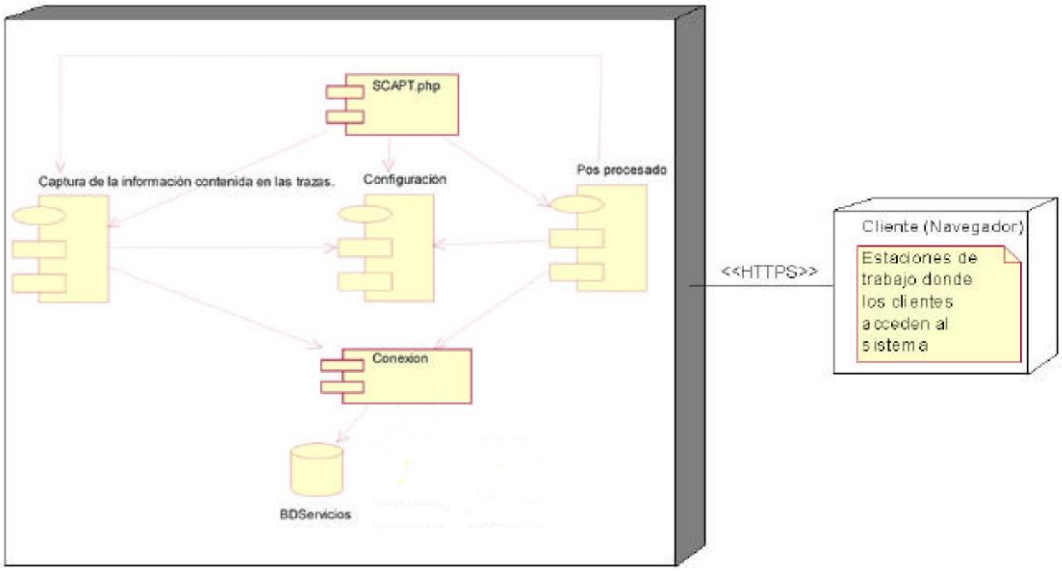


Figura 7. Diagrama de implementación.

3.4 Principios de diseño del sistema

A continuación se describen los principios de diseño seguidos para el desarrollo del sistema. Es importante mencionar que el mismo está orientado, a facilitar la rapidez y eficiencia en su utilización y en el caso de los módulos administrativos a garantizar su funcionalidad y comodidad en la gestión y administración descentralizada y remota.

3.4.1 Estándares en la interfaz de la aplicación

Para el caso del módulo de pos procesado, en el sistema se contrastan colores con tonalidad gris oscuro, gris claro y negro (negro y gris oscuro para la imagen empleada como Banner, las líneas y las letras utilizadas en todo el sistema, son de color azul oscuro, y para el fondo se utiliza el gris claro), además la letra debe ser legible y sencilla (se utiliza el tipo de letra Arial) para que unido al color utilizado contrasten con el fondo y sean refrescantes a la vista del usuario, se evita el uso excesivo de imágenes, animaciones y efectos visuales, en beneficio de una interfaz sencilla y rápida. Se mantuvo un diseño único en todo el sistema que proporciona un menú en la parte izquierda del sitio el cual está presente en todas las páginas del sistema, permitiendo que los administradores naveguen por el sistema manteniendo la posibilidad de acceso a todas las opciones en todo momento.

3.4.2 Tratamiento de errores

El sistema será programado de manera tal que se minimicen los errores aplicando técnicas de validación de datos y mediante la cuidadosa confección de la interfaz de entrada salida. Los mensajes de error que emite el sistema se mostrarán en un lenguaje claro y de fácil comprensión a personas sin conocimientos avanzados de computación. Además de esto se pedirá la confirmación por parte del usuario de operaciones que impliquen riesgos de pérdida de datos, tales como abrir o crear nuevas presentaciones y plantillas.

3.4.3 Concepción general de la ayuda

El sistema cuenta con una ayuda general la cual se encuentra disponible al administrador en cada momento, permitiéndole navegar por sus tópicos y consultar cualquier duda sobre el sistema, lo cual permite que el administrador pueda hacer un uso eficiente de la aplicación, la ayuda mantiene los mismos principios de diseño pero se muestra en una página independiente. Además existen ayudas específicas para diferentes opciones del sistema, a la que se puede acceder desde un enlace indicado por un signo de interrogación ubicado a continuación de la opción que lo requiera y mostrándose la ayuda en una pequeña ventana independiente.

3.5 Análisis de prueba y estudio de factibilidad

3.5.1 Pruebas y validaciones realizadas

A fin de comprobar el funcionamiento del sistema desarrollado se realizaron pruebas prácticas simulando posibles situaciones que afectarían la estabilidad, operatividad y seguridad en la red telemática. Además se emplearon herramientas de Software tanto para las pruebas antes mencionadas como para validar algunos resultados, según la experiencia de las pruebas empleadas en el SCAPT (Sistema para la centralización, análisis y procesamiento de trazas de los servicios telemáticos para la red UCf).

3.5.1.1 Modificación de las trazas en los equipos que las generan

Esta prueba consistió en eliminar las evidencias locales de acceso desde una IP restringida al servidor Web, de forma que quedara registrado el acceso a otra hora y desde una IP autorizada.

En este caso se comprobó que como el sistema almacena una copia de las trazas que se le envían directamente según se generan, se pueden detectar con prontitud situaciones similares y en la medida de la existencia de alarmas previamente configuradas se puede conocer rápidamente la ocurrencia de una violación. Para esta prueba se modificaron las trazas de servicios Squid y Radius.

3.5.1.2 Simulación de propagación de virus conocidos

Esta prueba que consistió en 2 acciones diferentes se había realizado en la variante inicial del SCAPT, volviendo a emplearse ahora a fin de validar diferentes opciones empleando el módulo de gestión de alarmas, las acciones fueron las siguientes:

Primero. Desde un Host fue empleado un Shell Script para enviar paquetes desde ese Host a diferentes Hosts pero al mismo puerto, empleando un puerto que está registrado como peligroso por ser empleado en la propagación de virus, según los informes de los CERT y sitios antivirus. El Script se basó en una modificación del uso del protocolo Telnet y se empleó TCPDUMP para validar el comportamiento esperado.

Segundo. Fue simulada la posible contaminación del servidor Web, para ello se ubicaron en el mismo archivos con patrones registrados en la base de datos

como peligrosos, a partir de los informes de los CERT. Posteriormente se procede al acceso a través del protocolo HTTP a los archivos contaminados.
Como resultado en ambos casos se obtuvo una alarma informando la ocurrencia del evento.

3.5.1.3 Simulación de accesos no autorizados por RAS

Para esta prueba se definieron reglas y se realizaron accesos RAS desde números telefónicos no registrados, comprobándose que el sistema genera una alarma ante la detección de un número no registrado para un determinado usuario en la base de datos.

3.5.1.4 Simulación de accesos a sitios peligrosos o url peligrosas

Para esta prueba se intentó el acceso a sitios permitidos por los filtros de contenidos como Dans-Guardian pero que se registraron en la base de datos de url peligrosas, además se utilizaron otros criterios aprovechando las posibilidades que brinda el módulo de gestión de alarmas para crear diferentes reglas basadas en la información de las trazas de Squid. Comprobándose que se generaban las alarmas de forma satisfactoria.

3.5.1.5 Simulación de falsos positivos

Para la realización de esta prueba se configuraron reglas que generaran fácilmente un buen número de alarmas, al ser generadas las mismas, se comprobó que cuando el número de falsos positivos en las alarmas de una regla determinada fuese igual que el número de falsos positivos definidos en su configuración, dicha regla se eliminara.

3.5.1.6 Simulación de intentos de accesos no autorizados

Para esta prueba se intentó acceder desde varios Hosts no autorizados y de forma consecutiva a un servidor empleando el protocolo SSH, lo que generó una alarma producto de la detección de un número de peticiones denegadas de forma consecutiva mayor o igual que 5 (incluye desde un solo Host) y de 10 de forma no consecutiva.
También se pudo comprobar el empleo de las opciones de inspección visual de las trazas, demostrando ser una herramienta efectiva que agiliza este trabajo.

3.5.1.7 Simulación de ataque de DDoS

Para esta prueba fueron simulados accesos desde varios Hosts de forma concurrente a un mismo servidor y un mismo servicio (Web puerto 80).

Obteniéndose el resultado esperado al generarse una alarma al sobrepasar 50 peticiones en total.

3.5.1.8 Simulación de intentos de Spoofing

Se utilizó una herramienta de construcción de paquetes, incluida en la distribución de seguridad (GNU/Linux) Backtrack, con la cual se modificaron los datagramas IP incorporándole en la IP de origen una dirección de la VLAN administrativa procediéndose a enviarlos desde diferentes VLAN. Empleándose la herramienta para determinar esta anomalía mediante la comprobación de la alarma generada por el sistema.

3.5.1.9 Simulación de monopolización del canal o el Backbone por un Host

Empleando los reportes de tráfico y gráficas de VLAN se pudo definir primeramente qué VLAN tiene más tráfico y después con los reportes de tráfico de la herramienta se pudo definir cual o cuales Hosts de esa VLAN estaban generando o recibiendo el tráfico.

3.5.1.10 Simulación de empleo de herramientas de Scanner de vulnerabilidades desde un Host no autorizado

Para esta prueba fueron verificadas a través de los reportes de tráfico, las peticiones desde un mismo Host a otro Host y diferentes puertos, fundamentalmente los puertos que se encuentran en el rango de 1 a 1023. Los resultados obtenidos se lograron mediante el empleo de herramientas como Retina y Nmap.

3.5.1.11 Simulación de violación de privilegios en el servicio Web

Para esta prueba se simuló el acceso al servidor Web desde un Host con dirección IP registrada como peligrosa, lo cual provocó una alarma informando la ocurrencia del evento. También fueron chequeados los mensajes con códigos de estado 401 el cual indica un acceso no autorizado al servidor.

3.5.1.12 Herramientas de Software utilizadas para validar los resultados

Para validar los resultados obtenidos se utilizaron varias herramientas ya empleadas con anterioridad en el SCAPT, entre las que se destacan:

Retina es una herramienta que ofrece la posibilidad de identificar eficazmente las vulnerabilidades que dejan a las redes expuestas a ataques maliciosos, revisando los servicios existentes en una máquina y proponiendo soluciones.

Nmap es un programa de código abierto que sirve para efectuar rastreo de puertos. Se usa para evaluar la seguridad de sistemas informáticos, así como para descubrir servicios o servidores en una red informática.

Tcpdump es una herramienta en línea de comandos, del tipo Sniffer, cuya utilidad principal es analizar el tráfico que circula por la red. Permite al usuario capturar y mostrar a tiempo real los paquetes transmitidos y recibidos en la red a la cual el ordenador está conectado.

Backtrack es una distribución GNU/Linux en formato LiveCD pensada y diseñada para la auditoría de seguridad y relacionada con la seguridad informática en general. Incluye una larga lista de herramientas de seguridad listas para usar, entre las que destacan numerosos scanners de puertos y vulnerabilidades, archivos de exploits, sniffers, herramientas de análisis forense y herramientas para la auditoría Wireless.

Telnet y SSH son protocolos utilizados para acceder a máquinas remotas a través de una red. Pudiendo ser empleados para comprobar algunos servicios.

3.6 Estudio de factibilidad

Para llevar a cabo un buen proyecto de desarrollo de Software, se debe comprender el ámbito del trabajo a realizar, los recursos requeridos, las tareas a ejecutar, las referencias a tener en cuenta, el esfuerzo (COSTE) a emplear y la agenda a seguir.

Para determinar si la construcción de un Software es factible o no se han introducido en la Ingeniería del Software una serie de técnicas utilizadas dentro de las tareas de planificación, que ayudan a planificar y controlar el esfuerzo y el tiempo necesario de desarrollo:

- Técnicas de estimación del esfuerzo (costo) de desarrollo. Dentro de las cuales se sitúa el Modelo Constructivo de Costes (Cons tructive Cost Model (COCOMO)).
- Técnicas de planificación y seguimiento de proyectos.

La estimación es una de las primeras actividades de la gestión de proyectos informáticos. Se le define como la predicción del personal, del esfuerzo, de los costos y del tiempo que se requerirán para realizar todas las actividades y construir todos los productos asociados con el proyecto. Su objetivo es conocer en etapas tempranas y de manera aproximada, el costo, la duración y los recursos necesarios para el desarrollo de proyectos de Software. Se trata de una apreciación del futuro y la exactitud con la que ésta se realice, depende la mayoría de las veces de una buena herramienta de estimación, de la experiencia del estimador y del acceso a una base de información histórica de los proyectos. (13)

3.6.1 Planificación por casos de usos

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner, y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

3.6.1.1 Cálculo de Puntos de Casos de Uso sin Ajustar

UUCP = UAW + UUCW

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

Cálculo de UAW:

Actor	Factor de Peso
Administrador	3
Syslog_ng	2

Cron	2

Tabla 11. Asignación de peso a los actores.

UAW = (Cantidad de actores) * Peso

UAW = 2*2+1*3

UAW = 7

Cálculo de UUCW:

Caso de Uso	Tipo
Filtrar y almacenar la información de las trazas	Simple
Gestionar criterios de configuración	Complejo
Procesar los datos almacenados de las trazas del servicio Radius	Complejo
Procesar los datos almacenados de las trazas del servicio Squid	complejo
Gestionar reglas por servicio	complejo
Gestionar Alarmas	complejo
Gestionar envío de alarmas	simple

Tabla 12. Factor de peso de los casos de uso sin ajustar.

UUCW = 2*5 + 0*10 + 5*15

UUCW = 85

Los puntos de casos de uso sin ajustar resultan:

UUCP = UAW + UUCW

UUCP = 7 + 85

UUCP = 92

3.6.1.2 Cálculo de los Puntos de Casos de Uso Ajustados

UCP = UUCP x TCF x EF

Donde:

UCP: Puntos de Casos de Uso ajustados.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Cálculo de TCF:

Factor	Descripción	Peso	Val. Asig.	Porque	Total
T1	Sistema Distribuido	2	5	Cuenta con un módulo que es una aplicación Web, y el otro módulo está conformado por Scripts que son ejecutados por el Cron, además cuenta con un servidor de trazas.	10
T2	Obj. de performance o tiempo de respuesta	1	4	La velocidad de respuesta es directamente proporcional a las entradas provistas por el administrador.	4
T3	Eficiencia del usuario final	1	5	No existen restricciones de eficiencia para el administrador.	5
T4	Procesamiento interno complejo	1	5	Se realizan operaciones complejas en ambos módulos.	5
T5	El código debe ser reutilizable	1	5	Cada módulo es implementado pensando en la incorporación de nuevos servicios, lo que fomentó que el código escrito fuera cien por ciento reutilizable.	5
T6	Facilidad de instalación	0.5	2	Al existir dos módulos, diferentes en cuanto al uso	1

				de tecnologías de programación, y al necesitar de otros sistemas como el Syslog-ng, con su respectiva configuración, se hace un poco difícil la instalación.	
T7	Facilidad de uso	0.5	4	El sistema una vez instalado es fácil de usar.	2
T8	Portabilidad	2	4	El sistema es muy portable.	8
T9	Facilidad de cambio	1	4	El sistema ha sido concebido pensando en la incorporación futura de nuevos servicios, y en cambios en las operaciones sobre los ya existentes.	4
T10	Concurrencia	1	4	Buena concurrencia	4
T11	Incluye objetivos especiales de seguridad	1	4	Debido a la importancia que poseen los datos que maneja el sistema, se hacen necesarios objetivos especiales de seguridad.	4
T12	Provee acceso directo a terceras partes	1	3	Cuenta con accesos a partes más importantes.	3
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	3	Para administradores y operadores de red, no se requieren muchos entrenamientos para el uso del sistema.	3

Tabla 13. Cálculo del factor de complejidad técnica.

TCF =0.6 + 0.01* S(Pesoi *Valor asignadoi)

TCF = 0.6 + 0.01 * (10 + 4 + 5 + 5 + 5 + 1 + 2 + 8 + 4 + 4 + 4 + 3 + 3)

TCF = 0.6 + 0.01 * 58

TCF = 0.6 + 0.58

TCF = 1.18

Cálculo de EF

Factor	Descripción	Peso	Val. Asig.	Porque	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	4	Existe familiarización con este tipo de proyectos.	6
E2	Experiencia con la aplicación	0.5	4	Se han realizados aplicaciones de este tipo.	2
E3	Experiencia en orientación a objetos	1	4	Los paradigmas de la programación orientada a objetos han sido aplicados en los sistemas implementados anteriormente.	4
E4	Capacidad del analista líder	0.5	4	Experiencia media	2
E5	Motivación	1	5	Alta motivación	5
E6	Estabilidad de los requerimientos	2	4	Abierto a cambios y mejoras	8
E7	Personal part-time	-1	0	El proyecto lo realiza una sola persona.	0
E8	Dificultad del lenguaje de programación	-1	2	Se utilizan varios lenguajes de programación, pero el	-2

				que más se usa es PHP.	

Tabla 14. Cálculo del factor ambiente.

EF = 1.4 - 0.03 * S (Pesoi *Valor asignadoi)

EF = 1.4 - 0.03 * (6+2+4+2+5+8+0-2)

EF = 1.4 - 0.03 * 25

EF = 0.65

Los puntos de casos de uso ajustados resultan:

UCP= UUCP * TCF * EF

UCP=92 * 1.18 * 0.65

UCP=70.56

3.6.2 Estimación del esfuerzo

CF: Factor de Conversión

CF= 20 Horas-Hombre

El esfuerzo en horas /hombre está dado por:

E= UCP * CF

E= 70.56 * 20 = 1411.2 horas-hombre

Duración:

Trabajando 25 días al mes y 12 horas diarias como promedio, se tiene que:

Duración (días)= Total de Horas /Hombre entre 12 horas al día = 3528/12 =294 días

Duración (meses)=Total de días / 25 días por mes = 294 / 25 = 11.76 ˜ 12 Meses

Actividad	Porcentaje	Valor
Análisis	10%	352.8
Diseño	20%	705.6
Programación	40%	1411.2
Pruebas	15%	529.2
Sobrecarga(otras actividades)	15%	529.2
Total de horas	100%	3528

Tabla 15. Criterios de distribución de esfuerzos.

3.6.3 Cálculo de costos

Tomando como salario promedio mensual $275.00

Costo = 12 meses * $275.00 = $3300.00

3.6.4 Beneficios tangibles e intangibles

El desarrollo de este sistema posibilita la obtención datos que permiten tomar acciones proactivas encaminadas a garantizar la estabilidad, disponibilidad y seguridad de la red, garantizando así el correcto funcionamiento de la misma. Además se logra que las tareas de supervisión y control de la información contenida en las trazas de los servicios telemáticos no resulten tediosas y sean realizadas con la periodicidad necesaria. De esta manera se logra que los esfuerzos empleados en el desarrollo del sistema estén encaminados al cumplimiento de los objetivos planteados.

3.6.5 Análisis de costos y beneficios

Anteriormente las tareas de supervisión y control de la información contenida en las trazas de los diferentes servicios telemáticos de la Red UCf, se realizaban de la siguiente manera: mediante el uso de herramientas que por ser muy específicas para un tipo de servicio, dejaban de ser una buena solución; manualmente lo que provocaba un consumo elevado de tiempo.

Con la explotación de los módulos incorporados al sistema antes descrito se logran mejorar estas tareas, con la ayuda de reportes para los servicios Squid y Radius y la generación de alarmas por prioridad a partir de reglas definidas por el administrador.

Con el análisis del costo del sistema, los beneficios que brinda el mismo y dando cumplimiento los objetivos planteados, se concluye que la aplicación es factible.

3.7 Conclusiones

En el presente capítulo se ha descrito la solución propuesta a través de la representación gráfica de los diagramas de clases del modelo de sistema, de los modelos lógico y físico de datos, y del diagrama de implementación. Fueron descritos además, los principios de diseño del sistema, específicamente, los

temas de estándares de la interfaz, concepción del tratamiento de errores, sistema de ayuda.

A lo largo de este capítulo se describen también los resultados obtenidos al someter el sistema a pruebas después de haber incorporado al mismo nuevos módulos, validando estos con herramientas a fines, obteniéndose en cada prueba los resultados deseados. Además se realizó el estudio de factibilidad económica del desarrollo del sistema, llegando a la conclusión de que con un hombre que desarrolle todas las etapas requeridas como son análisis, diseño e implementación, trabajando 12 horas diarias, 25 días al mes durante 12 meses, es suficiente. El desarrollo del sistema tendrá un costo de $3300.00.

Conclusiones

Como resultado de la presente investigación se puede concluir que con el desarrollo de los módulos de gestión de alarmas de los servicios Cisco, Apache2, Squid y Radius y de centralización para el análisis y procesamiento de trazas de los servicios telemáticos Squid y Radius, se logra mejorar las tareas de supervisión y control de los servicios telemáticos de la Red UCf permitiendo al administrador tener un conocimiento más rápido y detallado de eventos que ocurren en la red lo cual contribuye a elevar los niveles de seguridad existentes.

De igual forma se puede afirmar que:

Se incorporó al sistema de análisis y procesamiento de trazas de la red UCf; la centralización y almacenamiento de trazas de los servicios telemáticos Squid y Radius.

Se desarrolló el módulo de gestión de alarmas para los servicios telemáticos Cisco, Apache2, Squid y Radius que permite la validacion de las reglas a fin de poder disminuir los posibles falsos positivos que se generen.

Se incorporó al sistema la posibilidad de importar ficheros por lotes a fin de facilitar la actualización de la base dedatos con los datos obtenidos en los CERTs y otras instituciones similares.

Se validó la implementación final, mediante el empleo de herramientas de Software y el análisis del comportamiento del sistema.

Recomendaciones

Realizar estudio que permita la incorporación de las trazas de otros servicios telemáticos al sistema, con el fin de lograr la integración del análisis de trazas de todos los servicios de la Red UCf.

Continuar trabajando para culminar los módulos de correlación de eventos, análisis estadístico y procesamiento en línea.

Implementar el algoritmo de inteligencia artificial seleccionado en la investigación titulada “Propuesta de técnica de Inteligencia Artificial para la detección de anomalías en la red de datos de la Universidad de Cienfuegos” que tributa a esta investigación, a fin de mejorar la detección de situaciones anómalas ante eventos no conocidos.

Todas estas acciones en las que se trabajan actualmente como continuidad de la investigación realizada permitirán la realización de análisis proactivos, en busca de lograr una estrategia de red autodefensiva.

Referencias Bibliográficas

1. **Rivero Pérez, Jorge Luis.** *Sistema para la centralización, análisis y procesamiento de trazas de los servicios telemáticos para la red UCf.*

2. **Alvarez, Miguel Angel.** Qué es Python. [En línea] www.python.org.

3. A vocabulary and associated APIs for HTML and XHTML. [En línea] http://html.programacion.net.

4. Introducción a CSS. [En línea] http://www.desarrolloweb.com.

5. PHP. [En línea] http://www.maestrosdelweb.com/editorial/desarrollo-web/php/.

6. Introducción a JavaScript. [En línea] http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/.

7. Gestor de Base de Datos: MySQL, PostgreSQL, SQLite. [En línea] http://www.eaprende.com/gestor-de-basededatos-mysql-postresql-sqlite.html.

8. Servidor HTTP Apache. [En línea] es.wikipedia.org/wiki/Servidor_HTTP_Apache.

9. Como usar Syslog . [En línea] www.linux-magazine.es/issue/01/Syslog.pdf.

10. **Damián Pérez Valdés.** Editores web que facilitan tu trabajo. [En línea] http://www.laisla.cult.cu/index2.php.

11. **Mora, Francisco.** UML: Lenguaje Unificado de Modelado. [En línea] http://www.dccia.ua.es/dccia/inf/asignaturas/GPS/archivos/Uml.PDF.

12. **Jacobson.** El Proceso Unificado de Desarrollo de Software. [En línea]

13. **Peralta, Mario.** *Estimación del esfuerzo basada en casos de uso.* s.l. : Instituto Tecnológico de Buenos Aires.

Bibliografía

1. **Rivero Pérez, Jorge Luis.** *Sistema para la centralización, análisis y procesamiento de trazas de los servicios telemáticos para la red UCf.*

2. **Alvarez, Miguel Angel.** Qué es Python. [En línea] www.python.org.

3. A vocabulary and associated APIs for HTML and XHTML. [En línea] http://html.programacion.net.

4. Introducción a CSS. [En línea] http://www.desarrolloweb.com.

5. PHP. [En línea] http://www.maestrosdelweb.com/editorial/desarrollo-web/php/.

6. Introducción a JavaScript. [En línea] http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/..

7. Gestor de Base de Datos: MySQL, PostgreSQL, SQLite. [En línea] http://www.eaprende.com/gestor-de-basededatos-mysql-postresql-sqlite.html.

8. Servidor HTTP Apache. [En línea] es.wikipedia.org/wiki/Servidor_HTTP_Apache.

9. Como usar Syslog . [En línea] www.linux-magazine.es/issue/01/Syslog.pdf.

10. **Damián Pérez Valdés.** Editores web que facilitan tu trabajo. [En línea] http://www.laisla.cult.cu/index2.php.

11. **Mora, Francisco.** UML: Lenguaje Unificado de Modelado. [En línea] http://www.dccia.ua.es/dccia/inf/asignaturas/GPS/archivos/Uml.PDF.

12. **Jacobson.** El Proceso Unificado de Desarrollo de Software. [En línea]

13. **Peralta, Mario.** *Estimación del esfuerzo basada en casos de uso.* s.l. : Instituto Tecnológico de Buenos Aires.

14. Apache2. [En línea] http://www.apache2.com.

15. CERT de Cuba. [En línea]

16. Como usar el cron. [En línea] http://blogdrake.net/node/2171.

17. Configuración del crontab. [En línea] http://www.fabrega.info/documentos/cron/x32.html.

18. El Proceso Unificado de Desarrollo de Software (RUP). [En línea] http://www.utvm.edu.mx/OrganoInformativo/orgJul07/RUP.htm.

19. Herramientas de análisis para ficheros de registros de cortafuegos. [En línea] http:// www.linux-magazine.es/issue/03/Analyzer.pdf.

20. OSSEC. [En línea] http://www.ossec.net/main/about/.

21. PostgreSQL. [En línea] http://www.postgresql.org.

22. Protocolos de comunicaciones. [En línea] http://www.telefonicaonline.com/on/io/es/atencion/tutoriales_articulos/firewall/A1.htm.

23. Radius. [En línea] http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/COMUNW01/Tema_5_Desarrollado.htm.

24. Referencia sobre Squid. [En línea] http://diccionario.sensagent.com/squid/es-es/.

25. Retina. [En línea] http://www.amodio.biz/jorge/inetar/docs/Pagina12-Retina-940316.pdf.

26. UNAM-CERT. [En línea] http://www.cert.org.mx/boletin/?vulne=4906.

27. Weird Solutions. [En línea] http://corporate.weird-solutions.com/products/syslogturbo.

28. What's new in GFI EventsManager 8. [En línea] http://kbase.gfi.com/showarticle.asp?id=KBID003285.

29. **Altalef, N.** Syslog centralizado con detección de eventos. [En línea] http://www.redklee.com.ar.

30. Syslog Centralizado con detección de eventos. [En línea] http://www.redklee.com.ar.

Glosario de Términos

Squid: es el software para servidor Proxy**,** más popular y extendido entre los sistemas operativos basados sobre UNIX®. Es muy confiable, robusto y versátil. Al ser software libre, además de estar disponible el código fuente, está libre del pago de costosas licencias por uso o con restricción a un uso con determinado número de usuarios.

Radius: el Servicio de usuario de acceso telefónico de autenticación remota es un protocolo cliente-servidor de autenticación de seguridad, ampliamente utilizado por los proveedores de servicios Internet en otros servidores de acceso remoto. RADIUS es el medio más frecuente de autenticación y autorización de usuarios de acceso telefónico y de redes de túnel.

Backbone: Se refiere a las principales conexiones troncales de una red.

CERT: El término CERT proviene de las siglas en inglés *Computer Emergency Response Team* y viene a definir a un equipo de personas dedicado a la implantación y gestión de medidas tecnológicas con el objetivo de mitigar el riesgo de ataques contra los sistemas de la comunidad a la que se proporciona el servicio. También es conocido por las siglas CSIRT (Computer Security and Incidente Response Team) y ofrece servicios de respuesta ante incidentes y de gestión de seguridad.

Cron: En el sistema operativo Unix, cron es un administrador regular de procesos en segundo plano (*demonio*) que ejecuta procesos o Scripts a intervalos regulares (por ejemplo, cada minuto, día, semana o mes).

DDOS: Denegación distribuida de servicios.

GNU/Linux: es el término empleado para referirse al sistema operativo similar a Unix que utiliza como base las herramientas de sistema de GNU y el núcleo Linux.

IDS: Sistema detector de intrusos.

IP: Internet Protocol (Protocolo de Internet). Uno de los protocolos más representativos del estándar TCP/IP, es el responsable del esquema de direccionamiento utilizado en Internet y define la estructura y el intercambio de los datagramas IP entre redes distantes, definido en el RFC 791.

LAN: Local Area Network (Red de área local). Red de datos para dar servicio a un área geográfica máxima de unos pocos kilómetros cuadrados, por lo cual pueden optimizarse los protocolos de señal de la red para llegar a velocidades de transmisión de Gbps (gigabits por segundo).

MD5: En criptografía, MD5 (abreviatura de Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado.

NMap: Network Mapper.

Open Source: En la actualidad Open Source es utilizado para definir un movimiento nuevo de Software, diferente al movimiento del Software Libre, aunque no completamente incompatible con este, de modo que es posible (como de hecho ocurre) que ambos movimientos trabajen juntos en el desarrollo práctico de proyectos.

RFC: Acrónimo inglés de Request For Comments "Petición de comentarios".Conjunto de archivos de carácter técnico donde se describen los estándares o recomendaciones. Resultado y proceso de creación de un estándar en Internet. Los nuevos estándares se proponen y publican en Internet como RFC.
El grupo de trabajo de ingeniería de Internet (IETF) es un cuerpo de opinión que admite discusión a través de comentarios, en los que se establece un nuevo estándar.

SANS: SysAdmin Audit, Networking and Security Institute.

Script: En informática, un Script es un guión o conjunto de instrucciones. Permiten la automatización de tareas creando pequeñas utilidades. Es muy utilizado para la administración de sistemas UNIX. Son ejecutados por un intérprete de línea de órdenes y usualmente son archivos de texto.

Shell: Intérprete de comandos.

Spoofing: En términos de seguridad de redes hace referencia al uso de técnicas de suplantación de identidad generalmente con usos maliciosos.

SSH: Acrónimo en inglés de "**S**ecure **SH**ell" en español: intérprete de órdenes seguro. Es un protocolo que sirve para acceder a máquinas remotas a través de una red.

SSL: Protocolo de Capa de Conexión Segura, son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

Syslog-ng: es una interfaz que provee un Framework estándar para que tanto programas como el mismo sistema operativo puedan generar mensajes que pueden ser almacenados localmente o ser enviados a un Host remoto.

Telnet: Nombre de un protocolo de red que sirve para acceder mediante una red a otra máquina.

TCP: Transmission Control Protocol (Protocolo de Control de Transmisión). Protocolo de nivel de transporte estándar de la internet definido en STD 7, RFC 793.

TCP/IP: Transmission Control Protocol/Internet Protocol (Protocolo de Control de Transmisión/Protocolo Internet). Convenio de los protocolos de aplicación y transporte que se utiliza en el IP. Incluye FTP, TELNET, SMTP y UDP.

UCf: Universidad de Cienfuegos.

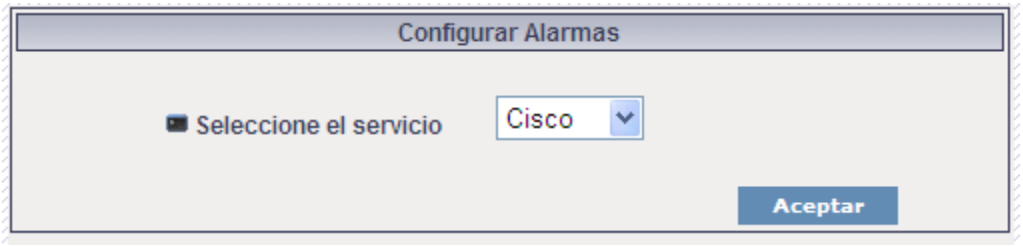
URL: localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

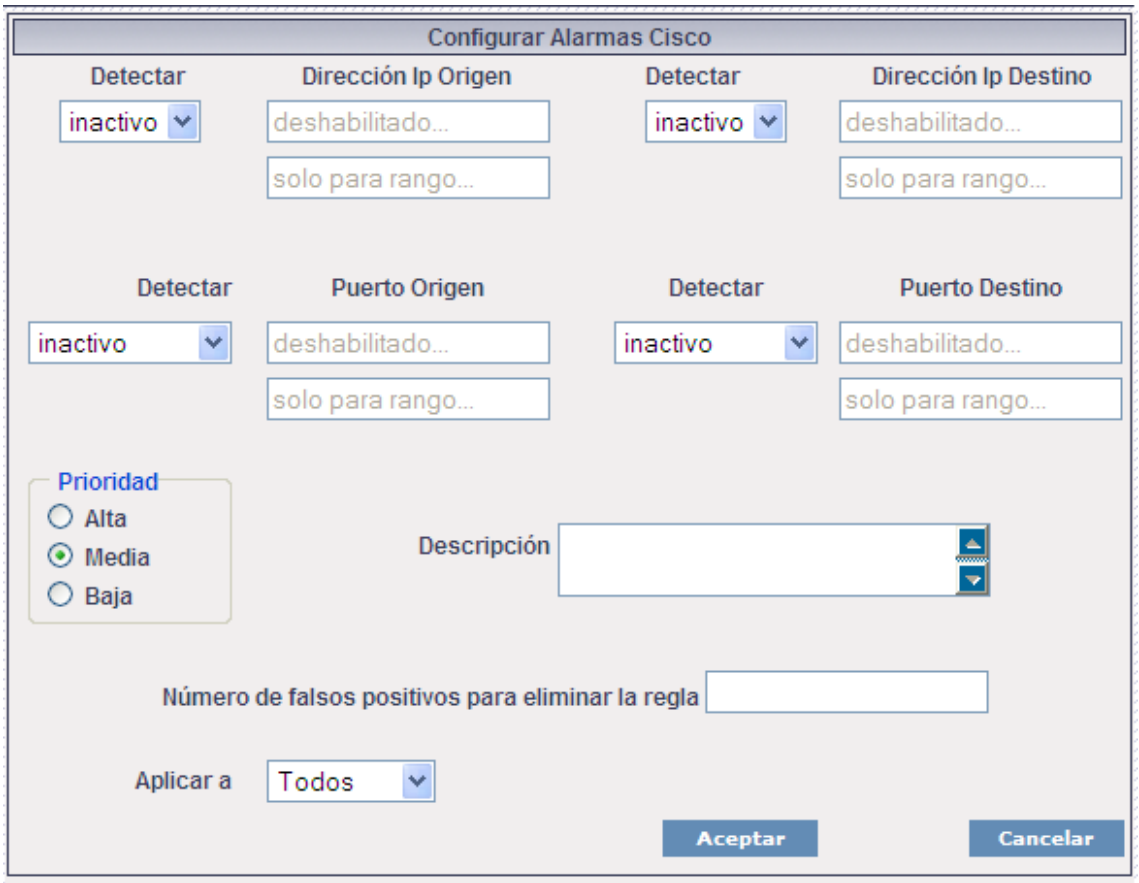
VLAN: (acrónimo de 'red de área local virtual') es un método de crear redes lógicamente independientes dentro de una misma red física.

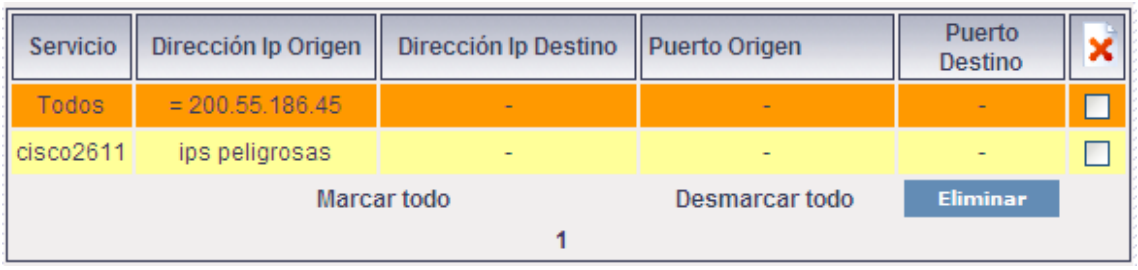
SMS: Sistema para el envío de mensajes de texto en las redes de telefonía móvil.

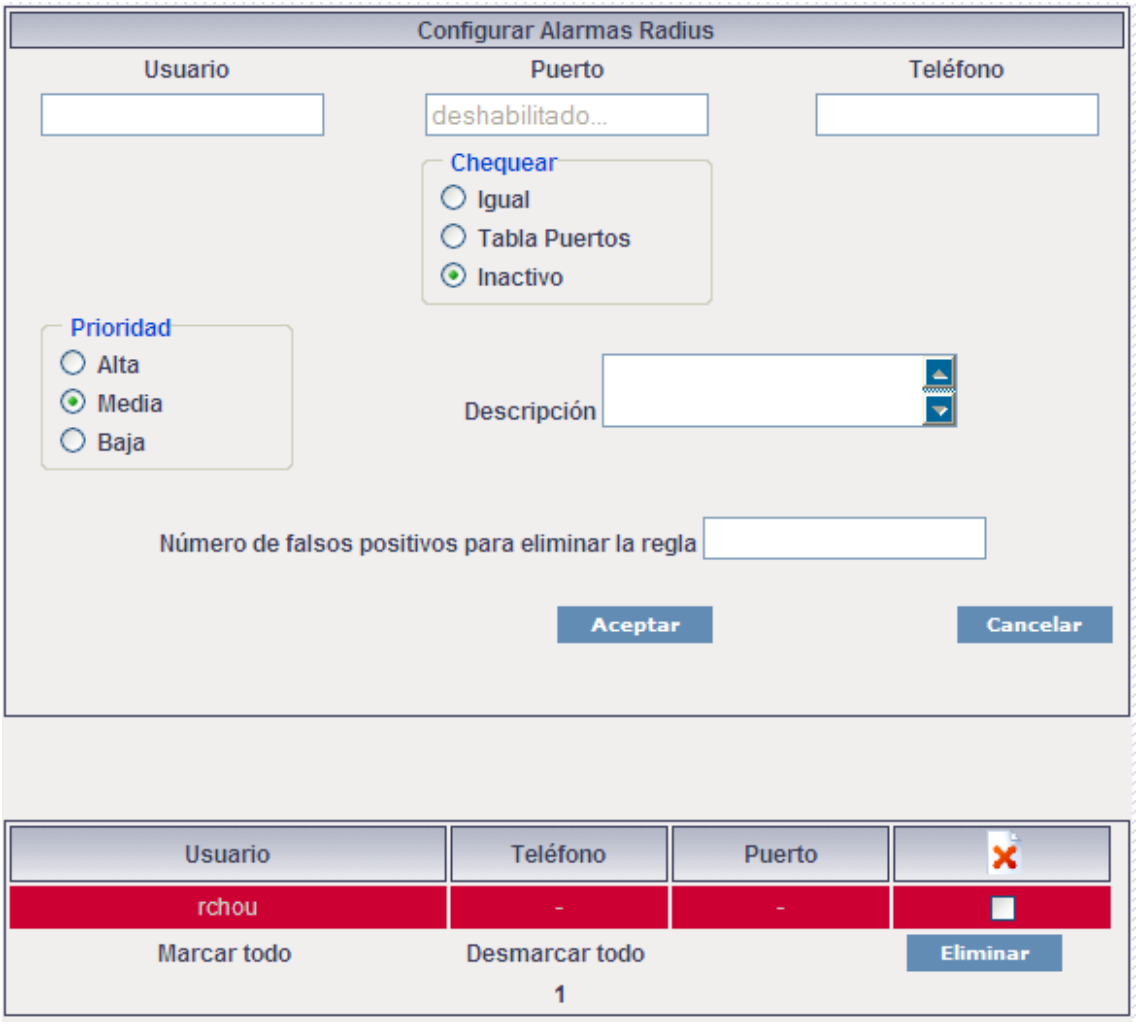
Anexos

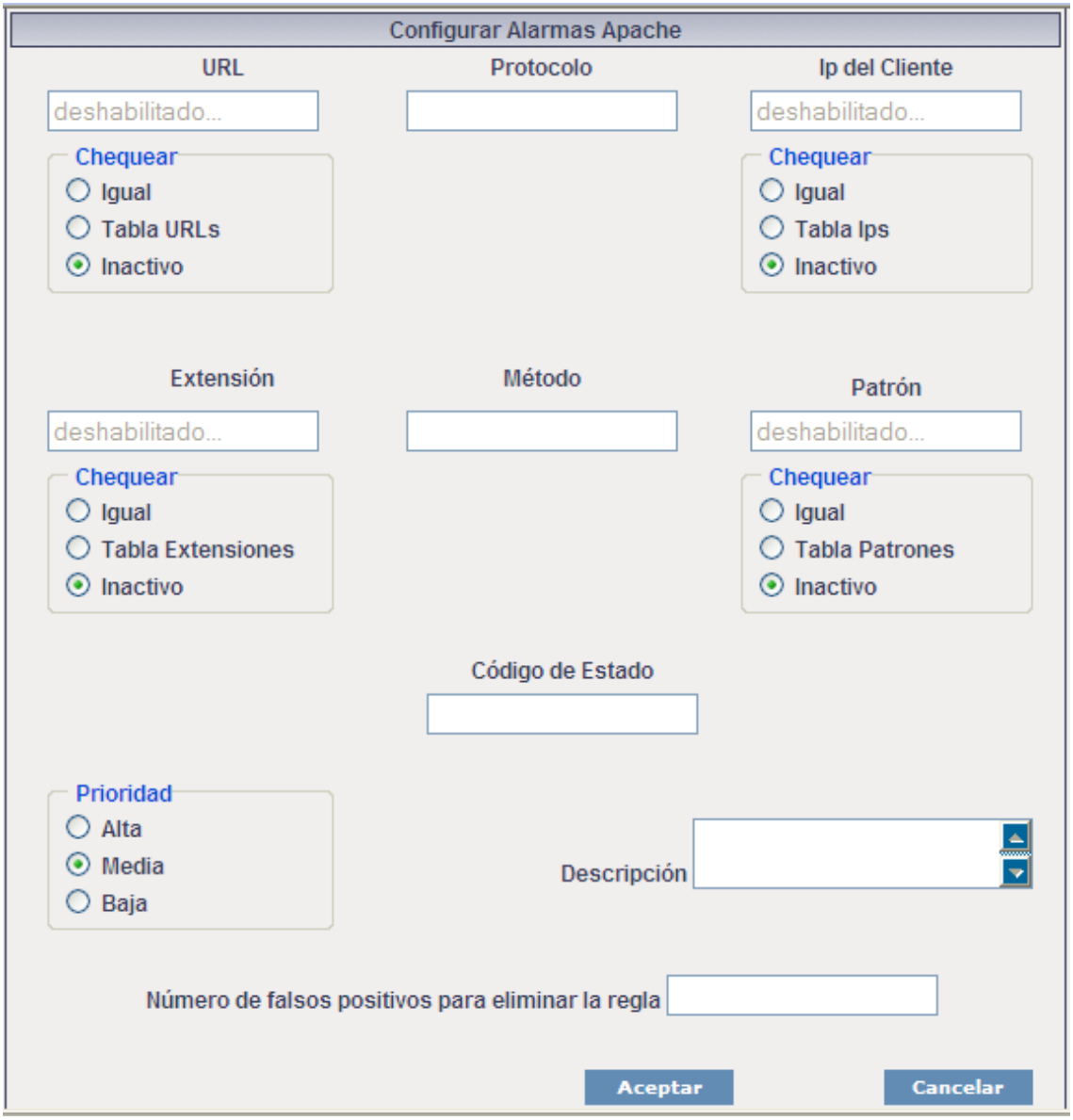
Anexo1: Prototipo del caso de uso Gestionar reglas por servicio.

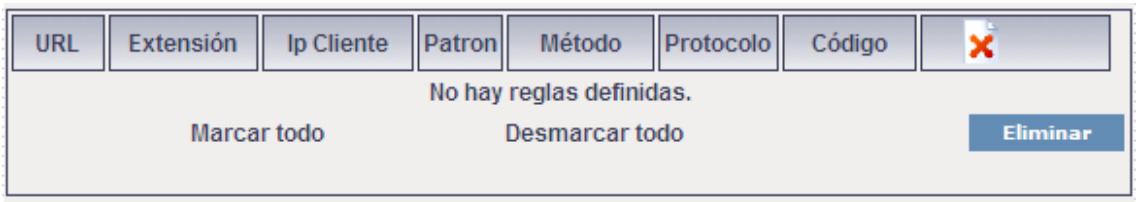




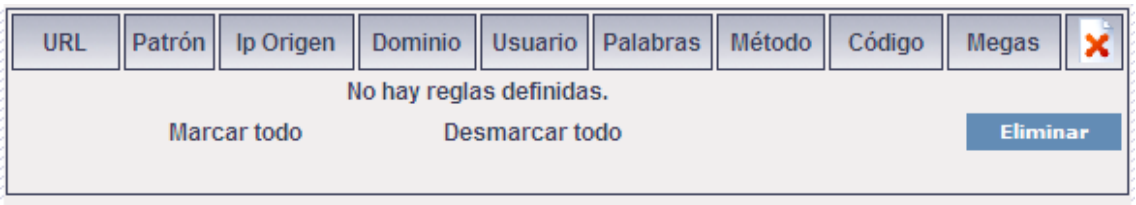



Configurar Alarmas Radius
Usuario
Puerto
Teléfono
deshabilitado...
Chequear
Igual
Tabla Puertos
Inactivo
Prioridad
Alta
Media
Baja
Descripción
Número de falsos positivos para eliminar la regla
Aceptar
Cancelar
Usuario
Teléfono
Puerto
rchou
-
-
Marcar todo
Desmarcar todo
Eliminar
1

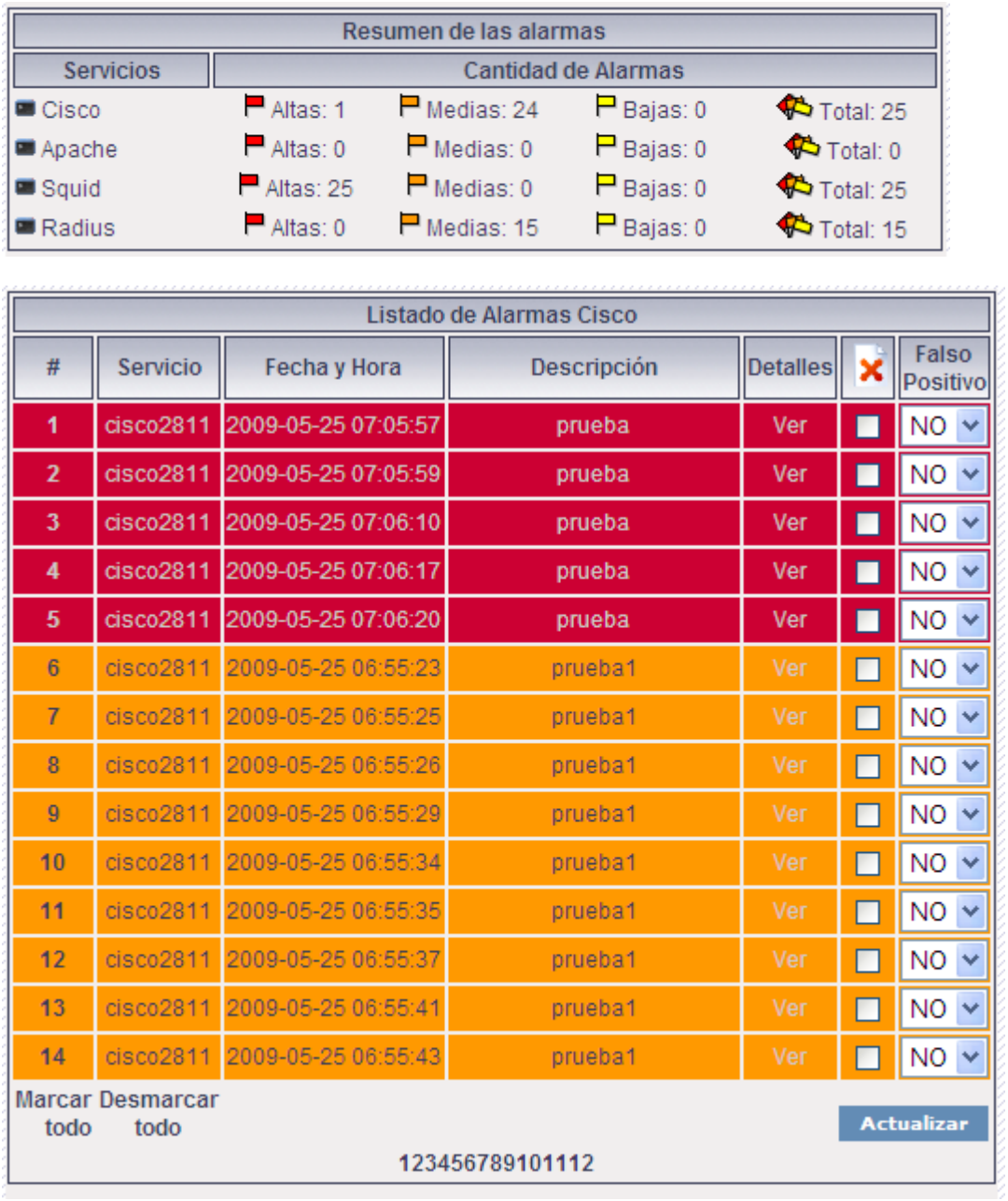

Configurar Alarmas Apache
URL
deshabilitado...
Chequear
Igual
Tabla URLs
Inactivo
Protocolo
Ip del Cliente
deshabilitado...
Chequear
Igual
Tabla Ips
Inactivo
Extensión
deshabilitado...
Chequear
Igual
Tabla Extensiones
Inactivo
Método
Patrón
deshabilitado...
Chequear
Igual
Tabla Patrones
Inactivo
Código de Estado
Prioridad
Alta
Media
Baja
Descripción
Número de falsos positivos para eliminar la regla
Aceptar
Cancelar


URL
Extensión
Ip Cliente
Patron
Método
Protocolo
Código
No hay reglas definidas.
Marcar todo
Desmarcar todo
Eliminar

Configurar Alarmas Squid
URL
deshabilitado...
Chequear
Igual
Tabla URLs
Inactivo
Patrón
deshabilitado...
Chequear
Igual
Tabla Patrones
Inactivo
Ip Origen
deshabilitado...
Chequear
Igual
Tabla Ips
Inactivo
Dominio
Usuario
Palabras Claves
Método
Cantidad de Megas
deshabilitado...
Código
inactivo
Prioridad
Alta
Media
Baja
Descripción
Número de falsos positivos para eliminar la regla
Aceptar
Cancelar


URL
Patrón
Ip Origen
Dominio
Usuario
Palabras
Método
Código
Megas
No hay reglas definidas.
Marcar todo
Desmarcar todo
Eliminar

Anexo2: Prototipo del caso de uso Gestionar alarmas.



Resumen de las alarmas

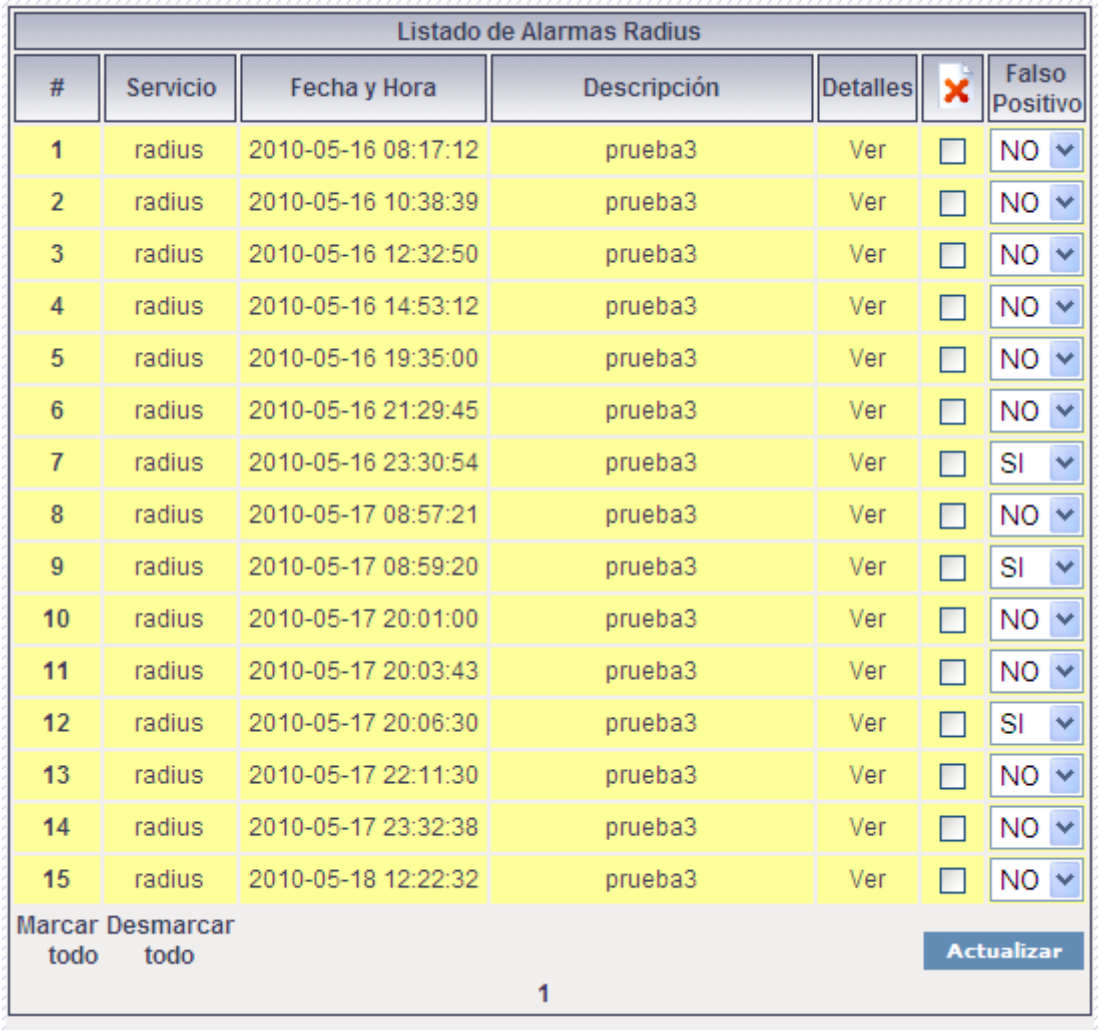
Servicios	Cantidad de Alarmas			
Cisco	Altas: 1	Medias: 24	Bajas: 0	Total: 25
Apache	Altas: 0	Medias: 0	Bajas: 0	Total: 0
Squid	Altas: 25	Medias: 0	Bajas: 0	Total: 25
Radius	Altas: 0	Medias: 15	Bajas: 0	Total: 15

Listado de Alarmas Cisco

#	Servicio	Fecha y Hora	Descripción	Detalles		Falso Positivo
1	cisco2811	2009-05-25 07:05:57	prueba	Ver	☐	NO
2	cisco2811	2009-05-25 07:05:59	prueba	Ver	☐	NO
3	cisco2811	2009-05-25 07:06:10	prueba	Ver	☐	NO
4	cisco2811	2009-05-25 07:06:17	prueba	Ver	☐	NO
5	cisco2811	2009-05-25 07:06:20	prueba	Ver	☐	NO
6	cisco2811	2009-05-25 06:55:23	prueba1	Ver	☐	NO
7	cisco2811	2009-05-25 06:55:25	prueba1	Ver	☐	NO
8	cisco2811	2009-05-25 06:55:26	prueba1	Ver	☐	NO
9	cisco2811	2009-05-25 06:55:29	prueba1	Ver	☐	NO
10	cisco2811	2009-05-25 06:55:34	prueba1	Ver	☐	NO
11	cisco2811	2009-05-25 06:55:35	prueba1	Ver	☐	NO
12	cisco2811	2009-05-25 06:55:37	prueba1	Ver	☐	NO
13	cisco2811	2009-05-25 06:55:41	prueba1	Ver	☐	NO
14	cisco2811	2009-05-25 06:55:43	prueba1	Ver	☐	NO

Marcar todo Desmarcar todo Actualizar

123456789101112



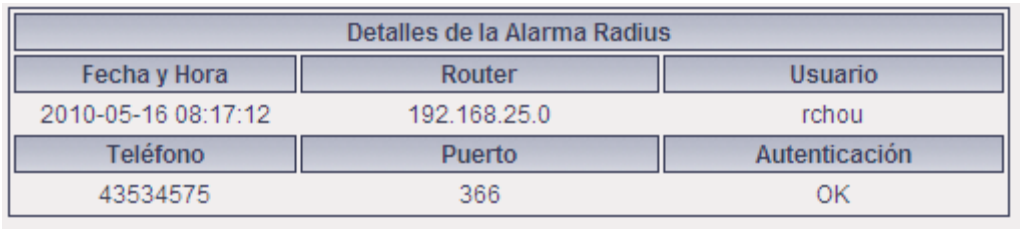
Listado de Alarmas Radius

#	Servicio	Fecha y Hora	Descripción	Detalles		Falso Positivo
1	radius	2010-05-16 08:17:12	prueba3	Ver	☐	NO
2	radius	2010-05-16 10:38:39	prueba3	Ver	☐	NO
3	radius	2010-05-16 12:32:50	prueba3	Ver	☐	NO
4	radius	2010-05-16 14:53:12	prueba3	Ver	☐	NO
5	radius	2010-05-16 19:35:00	prueba3	Ver	☐	NO
6	radius	2010-05-16 21:29:45	prueba3	Ver	☐	NO
7	radius	2010-05-16 23:30:54	prueba3	Ver	☐	SI
8	radius	2010-05-17 08:57:21	prueba3	Ver	☐	NO
9	radius	2010-05-17 08:59:20	prueba3	Ver	☐	SI
10	radius	2010-05-17 20:01:00	prueba3	Ver	☐	NO
11	radius	2010-05-17 20:03:43	prueba3	Ver	☐	NO
12	radius	2010-05-17 20:06:30	prueba3	Ver	☐	SI
13	radius	2010-05-17 22:11:30	prueba3	Ver	☐	NO
14	radius	2010-05-17 23:32:38	prueba3	Ver	☐	NO
15	radius	2010-05-18 12:22:32	prueba3	Ver	☐	NO

Marcar todo Desmarcar todo

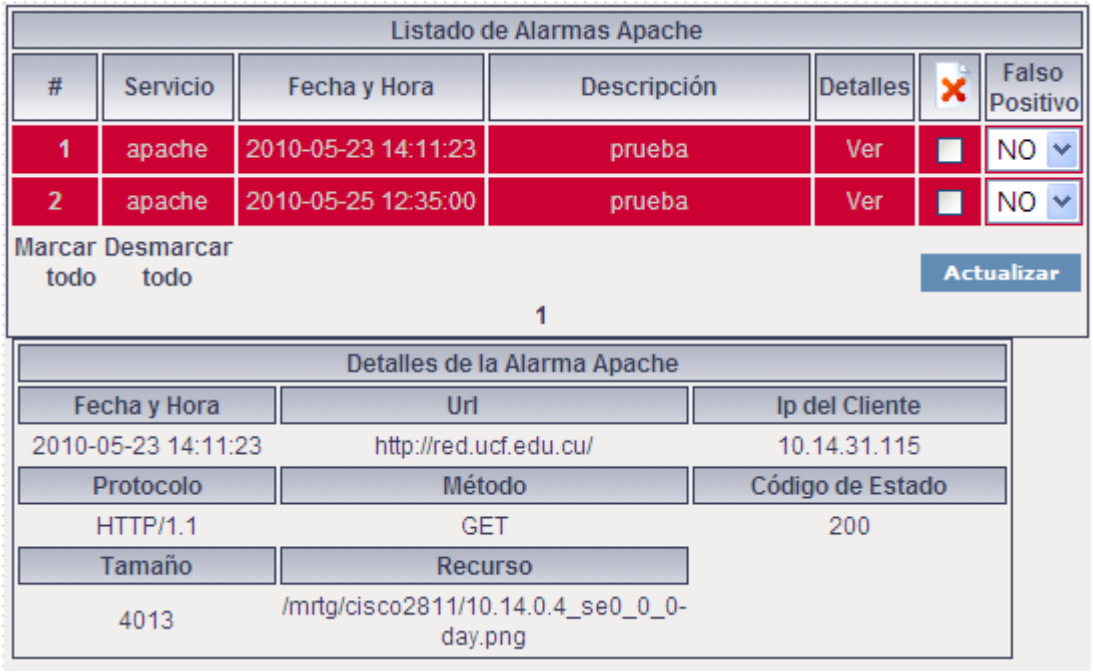
Actualizar

1

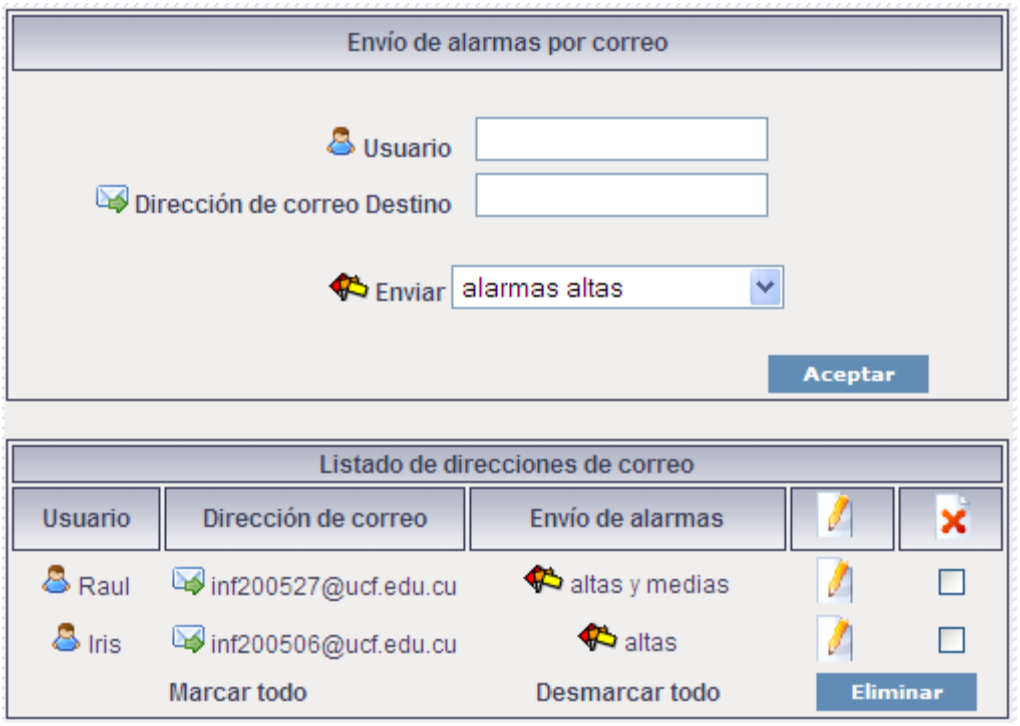


Detalles de la Alarma Radius

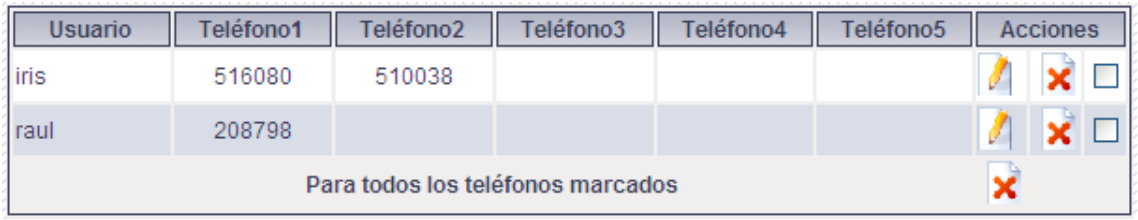
Fecha y Hora	Router	Usuario
2010-05-16 08:17:12	192.168.25.0	rchou
Teléfono	**Puerto**	**Autenticación**
43534575	366	OK

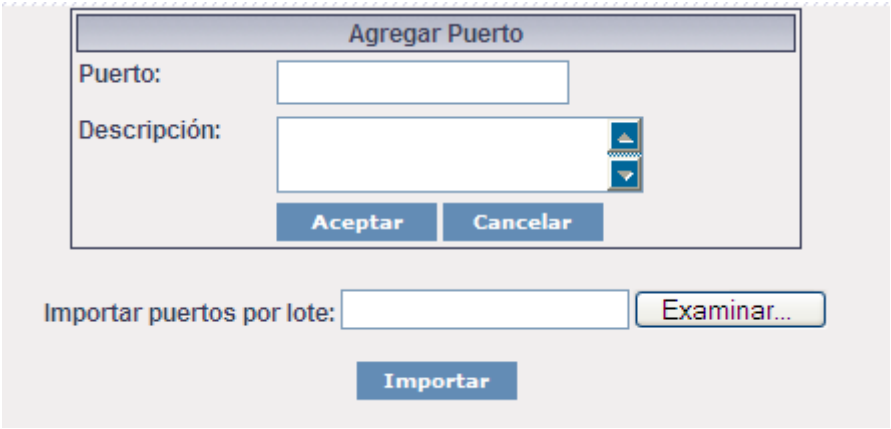


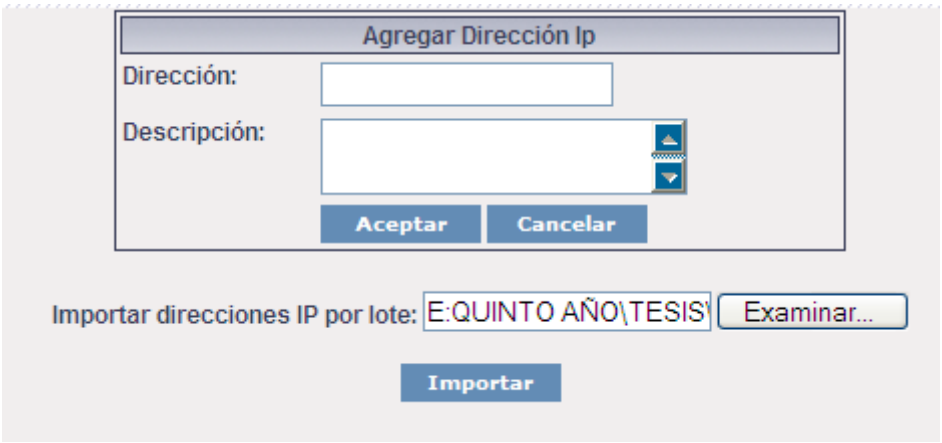
Anexo3: Prototipo del caso de uso Gestionar envío de alarmas.



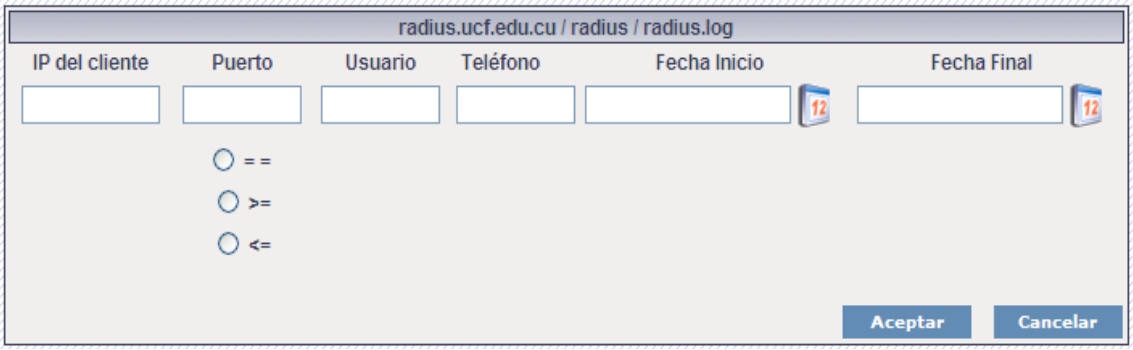
Anexo4: Prototipo del caso de uso Gestionar criterios de configuración.





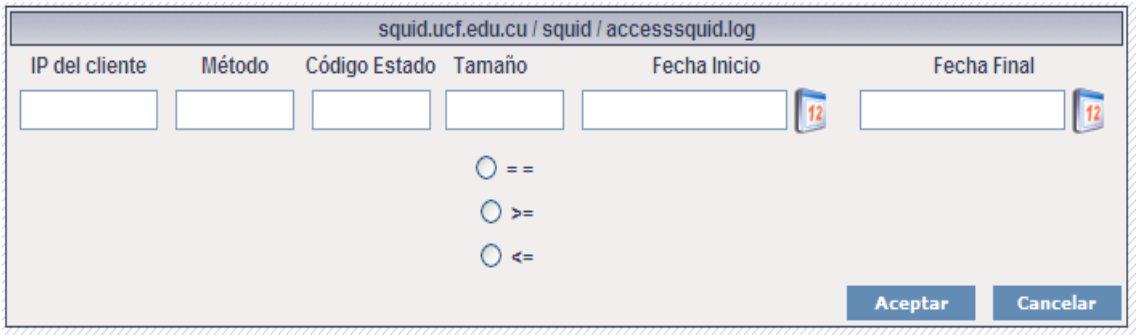


Anexo5: Prototipo del caso de uso Procesar los datos almacenados de las trazas del servicio Radius.



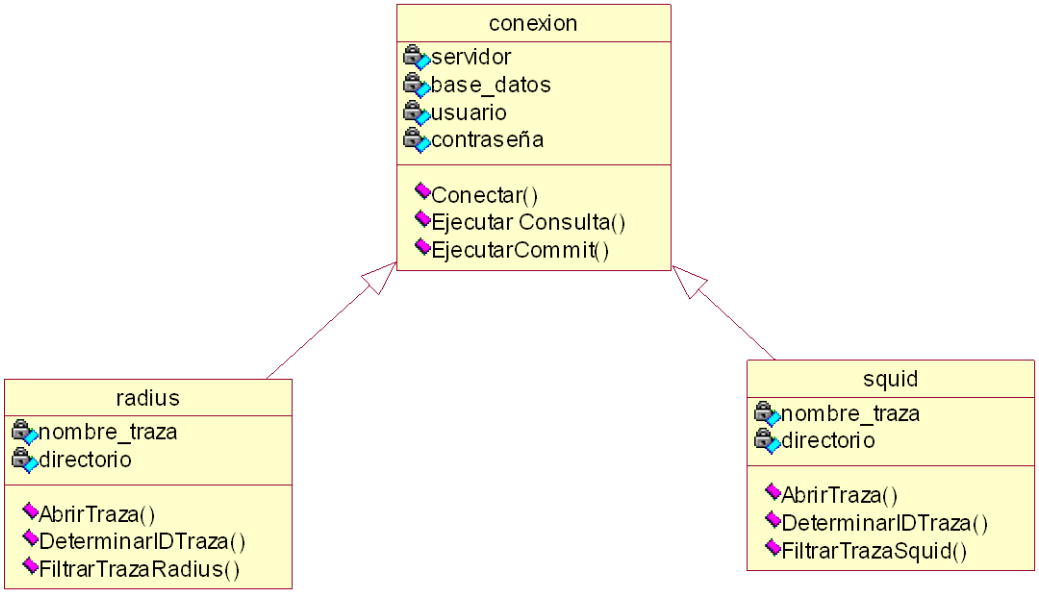
Fecha	Router	Usuario	Autenticación	Teléfono	Puerto
2010-05-16 08:17:12	192.168.25.0	rchou	OK	43534575	366
2010-05-16 10:38:39	192.168.25.0	rchou	OK	43534575	443
2010-05-16 12:32:50	192.168.25.0	rchou	OK	43534575	461
2010-05-16 14:53:12	192.168.25.0	rchou	OK	43534575	277
2010-05-16 19:35:00	192.168.25.0	rchou	OK	43534575	256
2010-05-16 21:29:45	192.168.25.0	rchou	OK	43534575	449
2010-05-16 23:30:54	192.168.25.0	rchou	OK	43534575	390
2010-05-17 08:57:21	192.168.25.0	rchou	OK	43534575	242
2010-05-17 08:59:20	192.168.25.0	rchou	OK	43534575	469
2010-05-17 20:01:00	192.168.25.0	rchou	OK	43534575	364
2010-05-17 20:03:43	192.168.25.0	rchou	OK	43534575	475
2010-05-17 20:06:30	192.168.25.0	rchou	OK	43534575	256
2010-05-17 22:11:30	192.168.25.0	rchou	OK	43534575	203
2010-05-17 23:32:38	192.168.25.0	rchou	OK	43534575	198
2010-05-18 12:22:32	192.168.25.0	rchou	OK	43534575	199

Anexo6: Prototipo del caso de uso Procesar los datos almacenados de las trazas del servicio Squid.

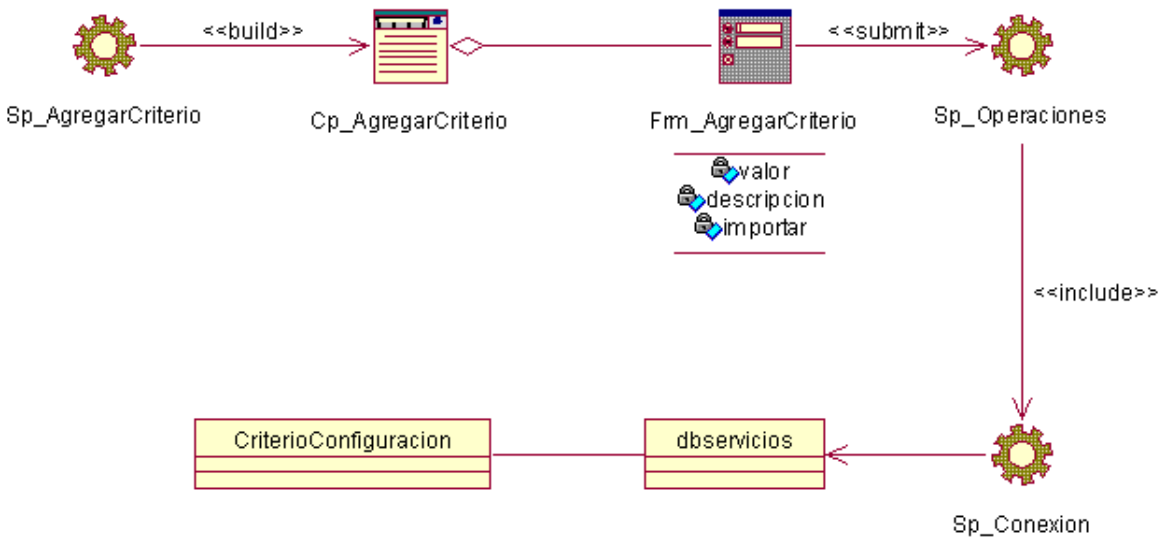


Fecha	Ip	Método	Tamaño	Url	Código
2010-05-23 14:44:22	10.71.38.102	GET	1954	http://www.microsoft.com/isapi/redir.dll?	TCP_DENIED
2010-05-23 14:45:13	10.71.38.102	GET	840	http://www.microsoft.com/isapi/redir.dll?	TCP_MISS
2010-05-23 14:45:14	10.71.38.102	GET	1954	http://www.microsoft.com/isapi/redir.dll?	TCP_DENIED
2010-05-23 14:46:04	10.71.38.102	GET	934	http://www.microsoft.com/isapi/redir.dll?	TCP_MISS
2010-05-23 14:46:04	10.71.38.102	GET	1927	http://go.microsoft.com/fwlink/?	TCP_DENIED
2010-05-23 14:46:45	10.71.38.102	GET	773	http://go.microsoft.com/fwlink/?	TCP_MISS
2010-05-23 14:46:45	10.71.38.102	GET	1888	http://es.msn.com/?	TCP_DENIED
2010-05-23 14:47:49	10.71.38.102	GET	1933	http://ads1.msn.com/library/dap.js	TCP_DENIED
2010-05-23 14:47:50	10.71.38.102	GET	1993	http://db2.stj.s-msn.com/br/hp/v12/es-es/js/15/hptr.js	TCP_DENIED
2010-05-23 14:47:50	10.71.38.102	GET	1987	http://db2.stj.s-msn.com/br/hp/v12/es-es/js/15/hp.js	TCP_DENIED
2010-05-23 14:47:54	10.71.38.102	GET	74025	http://es.msn.com/?	TCP_MISS

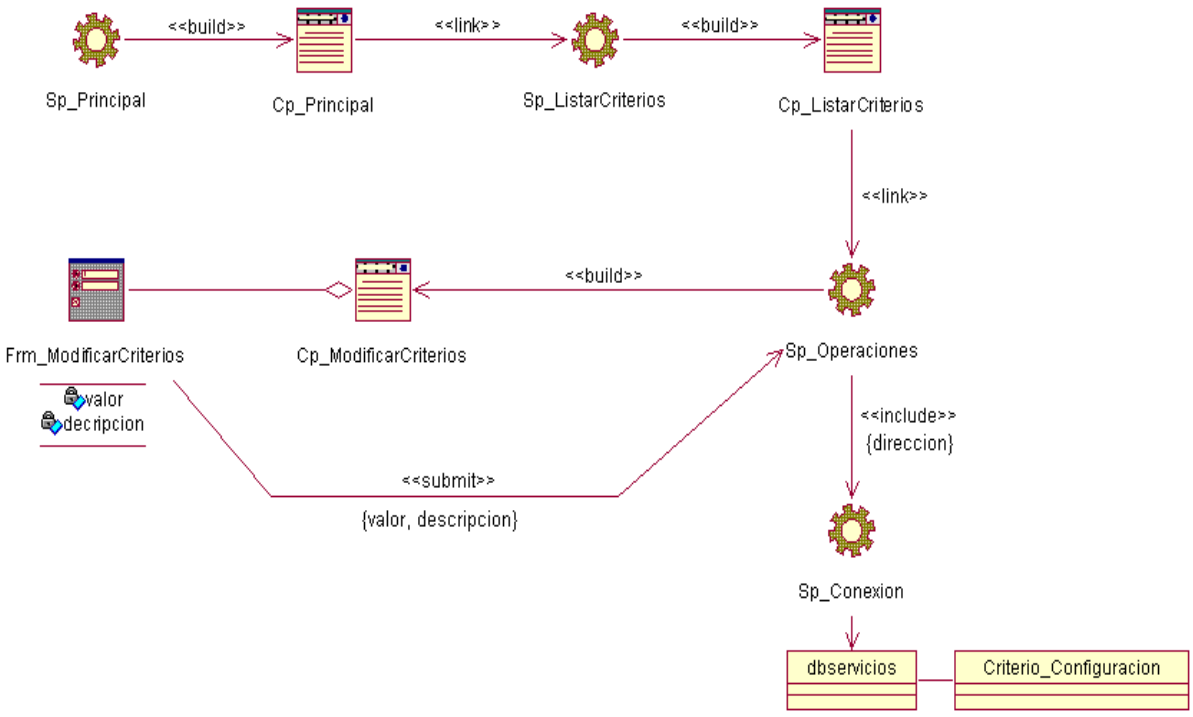
Anexo7: Diagrama de clases del módulo de captura de la información contenida en las trazas.



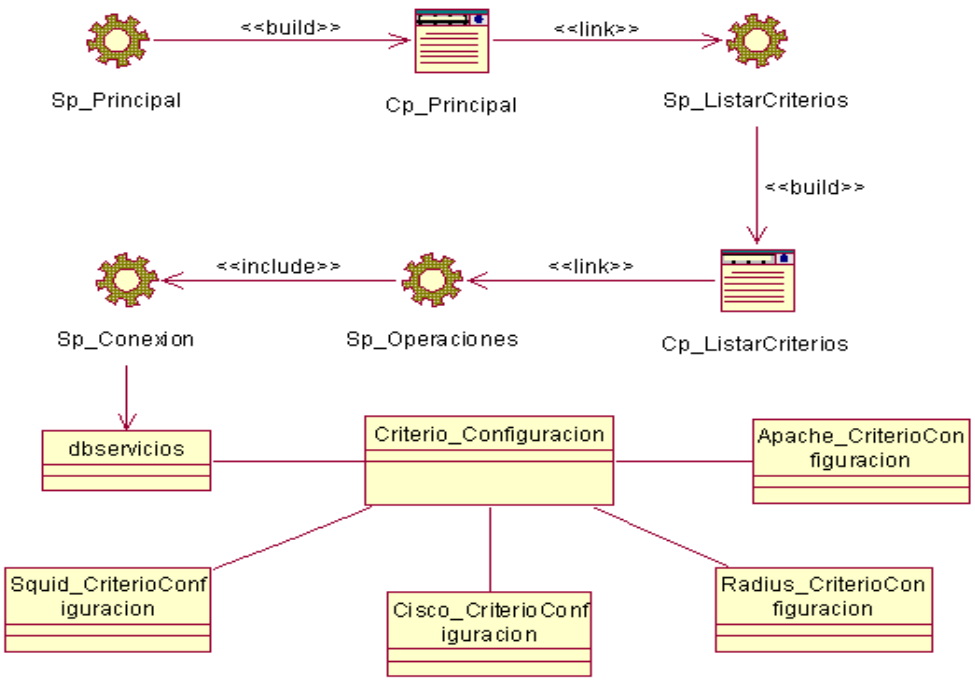
Anexo8: Diagramas de clases Web. Gestionar criterios de configuración



Agregar Criterio de Configuración

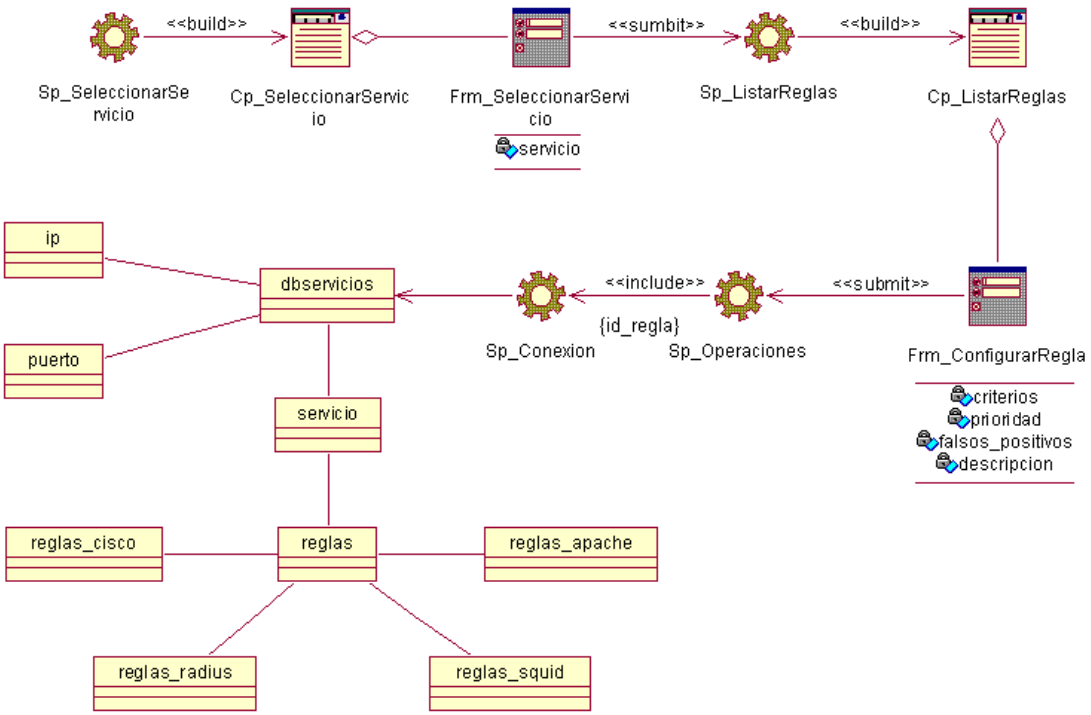


Modificar Criterio de Configuración

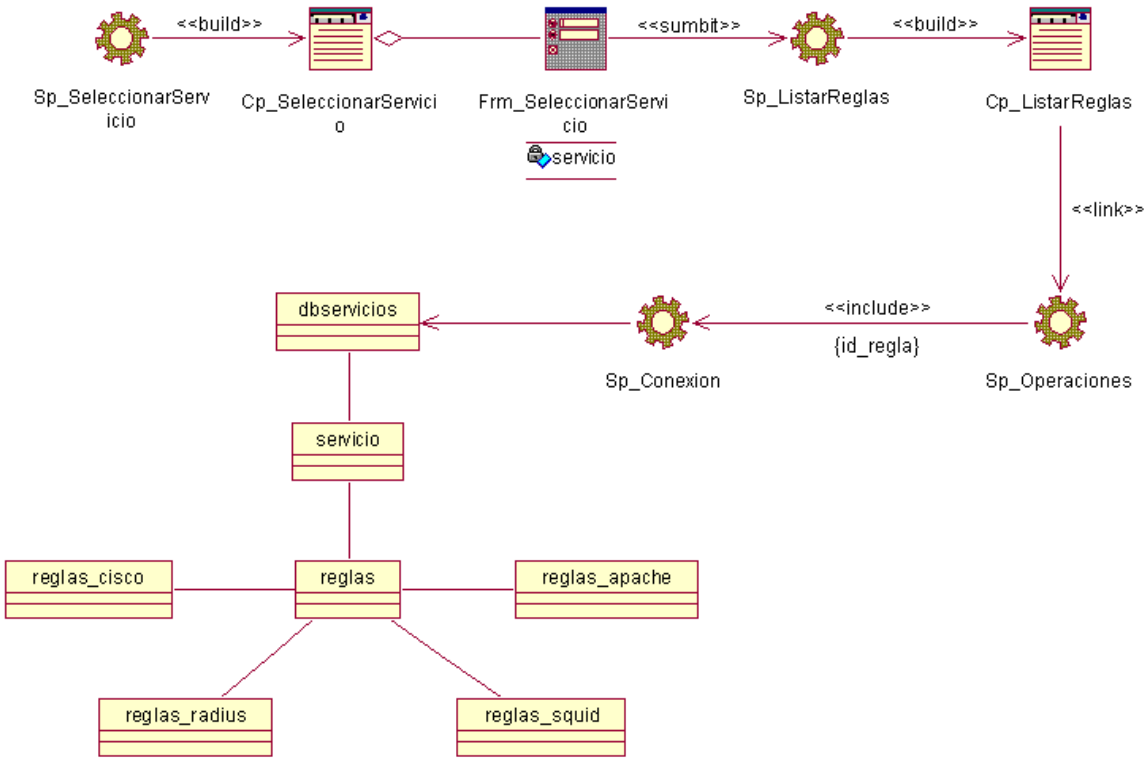


Eliminar Criterio de Configuración

Anexo9: Diagramas de clases Web. Gestionar reglas por servicio.

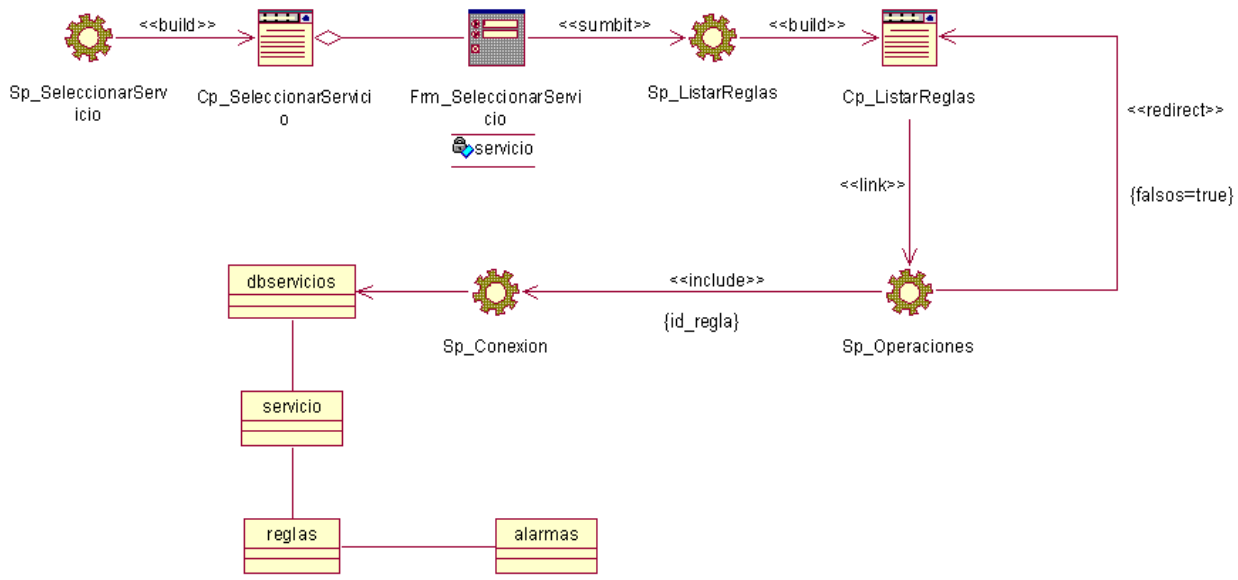


Agregar reglas por servicio



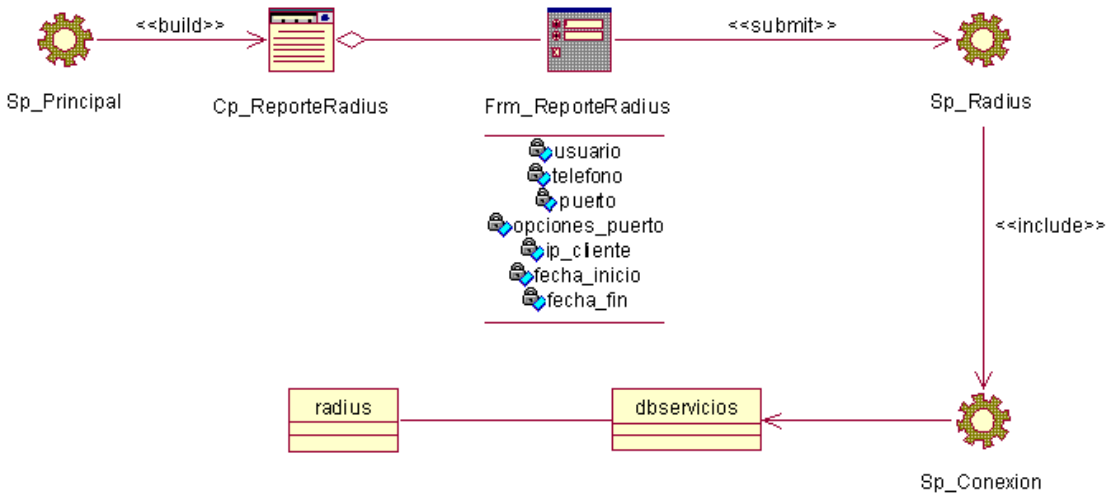
Eliminar reglas por servicio

Anexo10: Diagramas de clases Web. Gestionar alarmas.

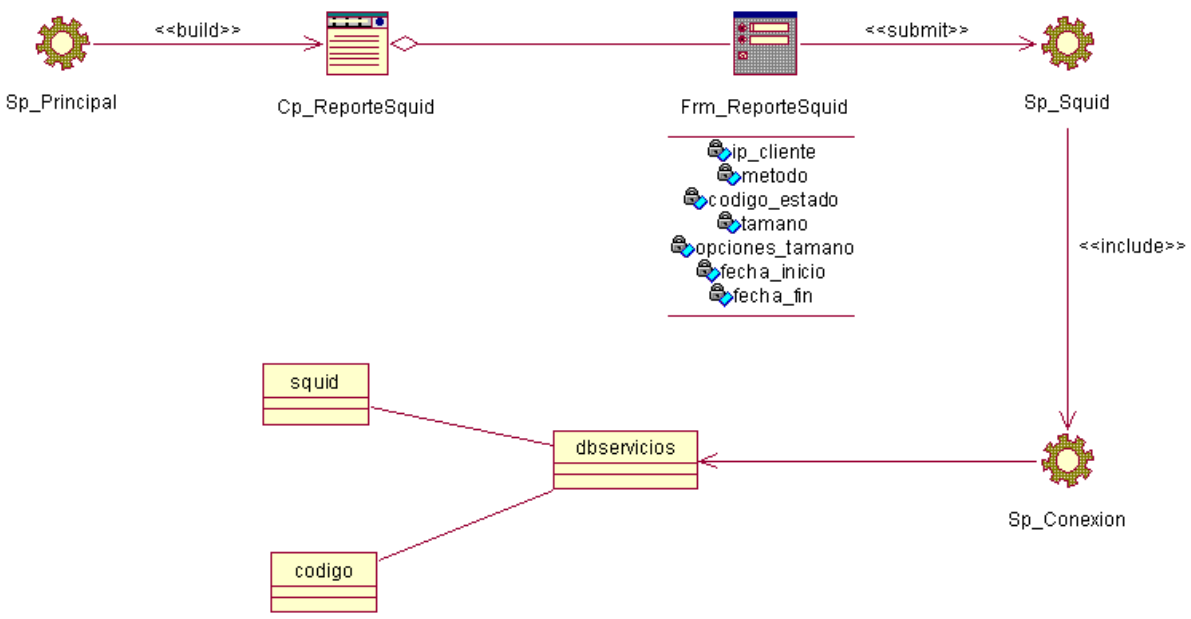


Eliminar Alarmas

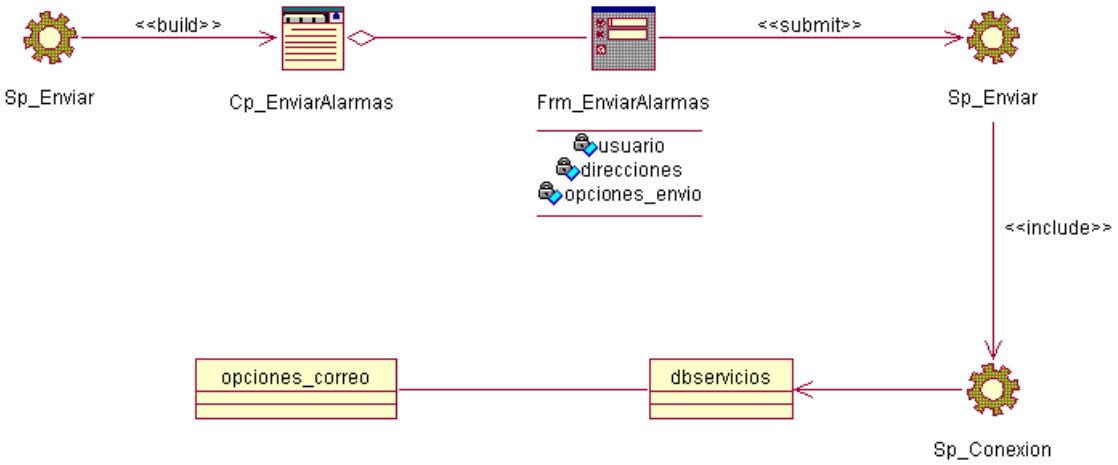
Anexo11: Diagramas de clases Web. Procesar los datos almacenados de las trazas del servicio Radius.



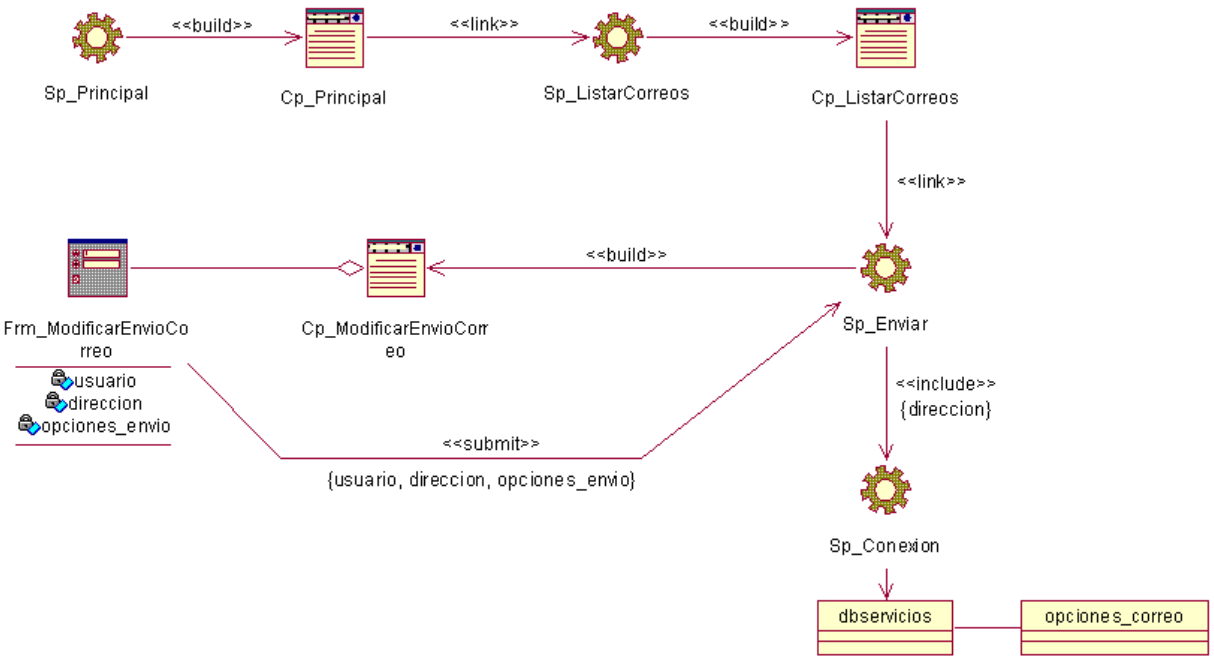
Anexo12: Diagramas de clases Web. Procesar los datos almacenados de las trazas del servicio Squid.



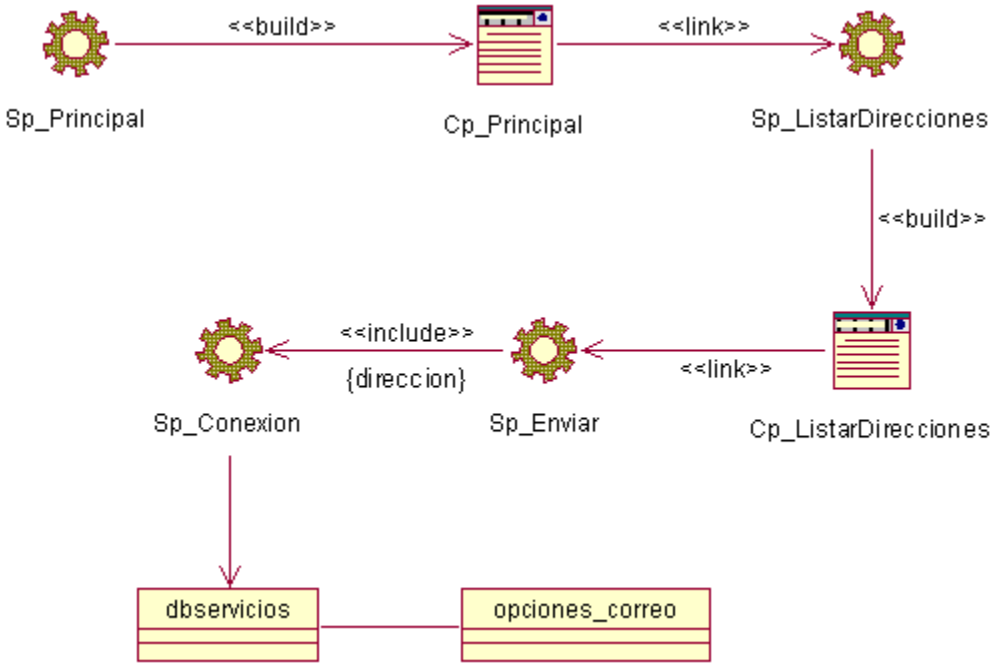
Anexo13: Diagramas de clases Web. Gestionar envío de alarmas.

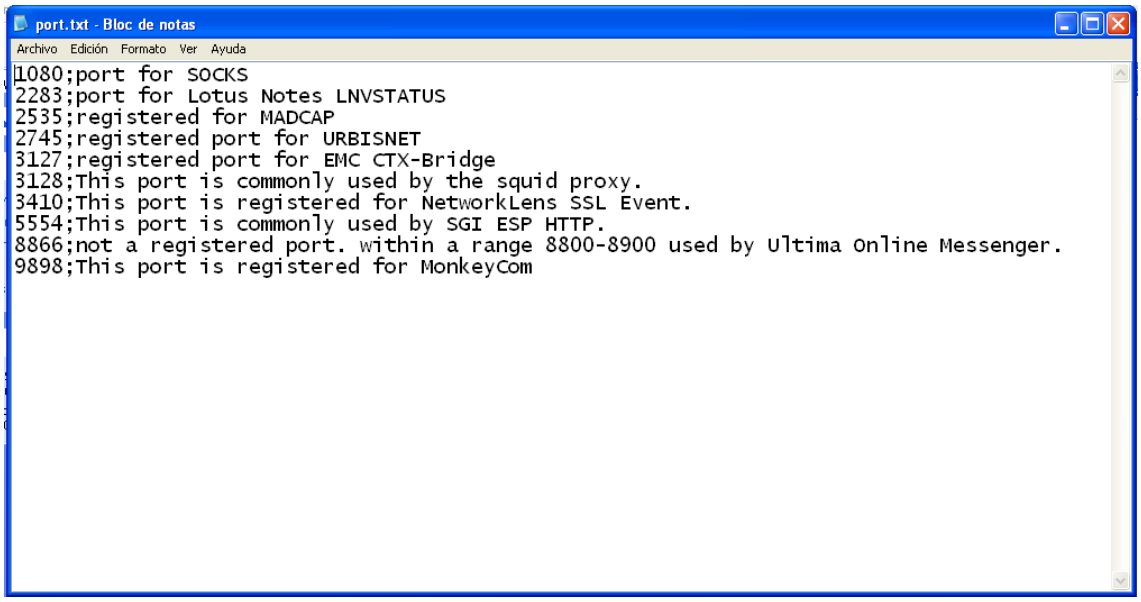


Agregar dirección de correo



Modificar direcciones de correo



Eliminar direcciones de correo

Anexo14: Estructura de los ficheros a Importar