

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 1



**SISTEMA DE GENERACIÓN DE REPORTES ESTADÍSTICOS PARA
ADMINISTRADORES DE REDES**

***Trabajo presentado en opción al título de
Ingeniero en Ciencias Informáticas***

Autores:

Alibech Nuñez Padilla

Isabelys Luisa Suárez Sevilla

Tutores:

Ing. Carlos Rafael Galán Cabello

Ing. Miguel Angel Chávez Alfonso

Consultantes:

Ing. Jorge Luis Hernández Cruz

Ing. Javier Menéndez Rizo

Ciudad de La Habana, Cuba

Junio, 2012

“Año 54 de la Revolución”

“El mundo no está en peligro por las malas personas sino por aquellas que permiten la maldad.”

Albert Einstein

Declaración de autoría

Declaramos ser los autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firman la presente a los _____ días del mes de _____ de 2012.

Alibech Nuñez Padilla

Isabelys Luisa Suárez Sevilla

Ing. Carlos Rafael Galán Cabello

Ing. Miguel Angel Chávez Alfonso

Agradecimientos

Le agradezco de forma muy especial a mi abuela Josefa, por ser una madre y un padre para mí, por estar cada momento a mi lado, por sus consejos, por su amor incondicional, por inculcarme su fuerza y su espíritu de lucha, por ser la persona más importante en mi vida.

A mis padres, por el apoyo que me han brindado, por creer en mí, por el amor y la confianza que me han tenido siempre.

A mi hermana, por ser parte de mi alegría e inspiración.

A mi hermano, por quererme tanto y por existir.

A mi abuelo Tito, por todo el amor y cariño que me brindó.

A mi novio Javier, por ser mi compañero, mi profesor, por guiarme en el buen camino del conocimiento, por su amor absoluto y por estar a mi lado en cada momento difícil de mi vida.

A mis tíos Adolfo y Alejandro, por la alegría y la confianza que siempre han depositado en mí.

A mis primas y a mi tía Belkis, por su preocupación y cariño.

A mis abuelos, Padilla, Barbara y Eneida, por ayudarme, quererme y preocuparse tanto.

A Rafael por el apoyo familiar que nos ha brindado incondicionalmente y a Damari por todo su afecto.

A mi amiga Milena, por ser como una hermana para mí, por estar presente en los momentos felices y difíciles de mi vida.

A mi otra familia, Yunet, Yamilia, Tilsia, Taymi, Yairon, Dayán, por apoyarme y apoyar a mi familia en todo momento.

A todos mis amigos y amigas del técnico.

A mi Grupo 4, por ser los "Inseparables" y por todo lo que vivimos juntos estos 5 años.

A mis amigas y hermanas de la Universidad, Mairetsy, Inalbis, Lisbet, Liuba, Yanet, Ilda, las mellizas (Lianet y Yanet), Leanet y Yanet Casas, que aunque no nos volvamos a ver por un buen tiempo, que sepan que nunca las olvidaré, por todo lo que aprendimos y crecimos juntas.

A mi compañera de tesis por toda su ayuda y por su gran espíritu alegre.

A mis tutores y en especial al Chino por brindarme la posibilidad de realizar esta tesis.

En fin a todos los que de una forma u otra contribuyeron tanto en mi formación profesional como en el desarrollo de este trabajo. A todos muchas Gracias!!!!

Alibech Nuñez Padilla

Un agradecimiento especial a mis padres y hermanos por darme fuerza desde lejos y por su preocupación constante. Gracias por todo el amor y confianza que siempre me han otorgado; sin ustedes, este sueño no significaría nada.

A mi novio y a la familia de mi novio, en especial a mi suegra por apoyarme y ayudarme durante los dos últimos años de mi carrera.

A mis tutores y co-tutores, a mi compañera de tesis por su ayuda, colaboración incondicional y apoyo en todo momento, por su preocupación constante en la realización de este trabajo, sin sus esfuerzos no hubiera llegado a la meta.

Les agradezco de corazón a todas aquellas personas que estaban a mi lado y estuvieron presentes desde los momentos más fáciles hasta los difíciles, en especial a mi familia y aquellas otras que no estaban presentes, pero que también brindaron su apoyo.

A mis amistades y compañeros, especialmente a los cercanos, por su preocupación y compañía y por darme ánimo para seguir adelante.

De todo corazón muchas Gracias

Isabelys Luisa Suárez Sevilla

Dedicatoria

De Alibech:

A mi abuela Josefa, por ser mi "mami" querida, mi razón de ser, de crecer y de vivir.

A mi mamá Zoraida, por el amor, la dedicación y la entrega que tiene con sus hijos.

A mi papá René, por todo su esfuerzo incondicional y por quererme tanto.

A mi hermana, la mejor del mundo y que pronto me dará el mismo regalo de ser Universitaria.

A mi hermano, por ser un regalo de la vida.

A mi abuelo Tito, que no se encuentra entre nosotros, pero me crío como a una hija.

A mi novio Javier, por ser una persona especial en mi vida.

De Isabelys:

A mi familia, a mis hermanitos y de forma especial a mis padres que han permitido con su esfuerzo y apoyo incondicional concluir mis estudios universitarios.

Resumen

La Universidad de las Ciencias Informáticas (UCI) además de contar con los privilegios y beneficios de Internet, rechaza la violencia y los riesgos a la cual se encuentran sometidos los internautas que navegan en ella. Debido a esto los Administradores de redes de la universidad necesitan el uso de herramientas que permitan mantener el control y seguimiento de las actividades que realizan los usuarios en la red.

El centro de Ideoinformática en conjunto con la Dirección de Redes y Seguridad Informática de la UCI, desarrolló el sistema Analizador Inteligente de Registros Proxy (AiresProxy) como una herramienta de análisis de la navegación que realizan los usuarios en Internet. Este sistema no permite extraer información estadística sobre datos generales de la navegación de los usuarios por Internet, por lo que se decide desarrollar un sistema informático que genere reportes estadísticos sobre la navegación de los usuarios, a través de un servidor proxy, para los Administradores de redes.

Para el desarrollo de este sistema se definen como tecnologías para su implementación el uso de MongoDB como Sistema Gestor de Base de Datos, C++ como lenguaje de programación, SXP como metodología que guía el proceso de desarrollo y Qt como biblioteca para desarrollar aplicaciones con interfaz gráfica de usuario. Con el desarrollo del sistema para la generación de reportes generales de las conexiones a Internet de los usuarios y el uso del servidor proxy, los Administradores de redes podrán contar con elementos de gran importancia para la toma de decisiones y el empleo de normas de seguridad.

Palabras clave: Analizadores de registros proxy, Archivo log, Estadísticas, Internet, Reportes, Servidor proxy.

ÍNDICE GENERAL

Introducción	1
1. Fundamentación teórica	5
1.1. Introducción	5
1.2. Principales definiciones	5
1.2.1. Archivos logs	5
1.2.2. Servidor proxy	5
1.2.3. Logs basados en un servidor proxy	6
1.2.4. Servidor proxy-caché	7
1.3. Sistemas generadores de reportes estadísticos	8
1.3.1. Ámbito Internacional	9
1.3.2. Ámbito Nacional	11
1.4. Determinación del tipo de aplicación	12
1.5. Metodología, lenguajes y herramientas a utilizar	14
1.5.1. Metodología para el desarrollo del software	14
1.5.2. Lenguaje de modelado	16
1.5.3. Lenguaje de programación	17
1.5.4. Entorno de Desarrollo Integrado	19
1.5.5. Herramientas de modelado de software	20
1.5.6. Sistema Gestor de Base de Datos	22
1.5.7. Biblioteca	24
1.5.8. Formato XML	24
1.6. Conclusiones	25
2. Descripción y características del sistema AiresProxy para Administradores de redes	26
2.1. Introducción	26
2.2. Descripción del sistema a desarrollar	26
2.2.1. Reportes asociados a las visitas y conexiones a Internet	27

2.2.2. Reportes asociados al funcionamiento y rendimiento del servidor proxy	27
2.2.3. Opciones de interactividad	28
2.3. Requerimientos del sistema	28
2.4. Historias de usuario	31
2.5. Descripción de la arquitectura	38
2.5.1. Especificación de la arquitectura del sistema	39
2.6. Patrones de diseño	41
2.6.1. Patrones de diseño GRASP	42
2.6.2. Patrones de diseño GoF	43
2.7. Diseño del sistema	44
2.7.1. Modelo de dominio	44
2.7.2. Diagrama de clases del diseño	46
2.7.3. Diagrama de paquetes del sistema	47
2.8. Diseño de la Base de Datos	49
2.8.1. Diagrama de la Base de Datos	49
2.9. Conclusiones	51
3. Implementación del sistema generador de reportes estadísticos	52
3.1. Introducción	52
3.2. Diagrama de componentes	52
3.3. Diagrama de despliegue	56
3.4. Conclusiones	58
4. Validación de la solución propuesta	59
4.1. Introducción	59
4.2. Pruebas de funcionalidad	59
4.2.1. Resultado de las pruebas	66
4.3. Conclusiones	67
Conclusiones Generales	68
Recomendaciones	69
Referencias	72

Bibliografía	73
Glosario de Términos	74
Anexos	76
Anexo # 1. Jerarquía de caché.	76
Anexo # 2. Interfaz de AiresProxy para Usuarios.	76
Anexo # 3. Interfaz de AiresProxy para Administradores de redes.	77
Anexo # 4. Interfaz de la funcionalidad “Dominios más visitados”.	78
Anexo # 5. Interfaz de la funcionalidad “Dominios por usuario”.	79
Anexo # 6. Interfaz de la funcionalidad “Usuarios por dominio”.	80
Anexo # 7. Interfaz de la funcionalidad “Direcciones IP de peticiones”.	81
Anexo # 8. Interfaz de la funcionalidad “Resumen general”.	82
Anexo # 9. Interfaz de la funcionalidad “Clasificación de protocolos”.	83
Anexo # 10. Interfaz de la funcionalidad “Respuesta de servidores”.	84
Anexo # 11. Interfaz de la funcionalidad “Gestionar usuario”.	85
Anexo # 12. Documento en formato PDF generado por el sistema.	86
Anexo # 13. Historias de Usuario.	86
Anexo # 14. Descripción de la clase VentanaPrincipal.	98
Anexo # 15. Descripción de la clase Generador.	99
Anexo # 16. Descripción de la clase Reporte.	101
Anexo # 17. Descripción de la clase Conexión.	103
Anexo # 18. Descripción de la clase Configuración.	105
Anexo # 19. Descripción de la colección Domains.	106
Anexo # 20. Descripción de la colección Logs.	107
Anexo # 21. Descripción de la colección Usuarios.	108
Anexo # 22. Casos de pruebas de funcionalidad.	108

ÍNDICE DE TABLAS

1.1. Analizadores de logs	9
2.1. Historia de Usuario SGREAR-01. Visualizar los dominios más visitados	32
2.2. Historia de Usuario SGREAR-02. Visualizar las mayores descargas	33
2.3. Historia de Usuario SGREAR-03. Mostrar las respuestas de los servidores	34
2.4. Historia de Usuario SGREAR-04. Visualizar los dominios por usuarios	35
2.5. Historia de Usuario SGREAR-05. Visualizar los usuarios por dominio	36
2.6. Historia de Usuario SGREAR-06. Configuración de la conexión de acceso a datos	37
2.7. Historia de Usuario SGREAR-07. Generar reportes en formato PDF	38
3.1. Descripción de componentes incorporados en la VentanaPrincipal	54
3.2. Descripción de componentes incorporados en el Generador	54
3.3. Descripción de componentes incorporados en el Reporte	55
3.4. Descripción de componentes incorporados en Configuración	56
4.1. Caso de prueba de aceptación SGREAR-01	60
4.2. Caso de prueba de aceptación SGREAR-02	61
4.3. Caso de prueba de aceptación SGREAR-03	62
4.4. Caso de prueba de aceptación SGREAR-04	63
4.5. Caso de prueba de aceptación SGREAR-05	64
4.6. Caso de prueba de aceptación SGREAR-06	65
4.7. Caso de prueba de aceptación SGREAR-07	66
4.8. No conformidades identificadas en el sistema	67
4.9. Historia de Usuario SGREAR-08. Mostrar resumen general del uso servidor proxy	87
4.10. Historia de Usuario SGREAR-09. Visualizar las direcciones IP por peticiones	88
4.11. Historia de Usuario SGREAR-10. Mostrar los diferentes métodos de conexiones	89
4.12. Historia de Usuario SGREAR-11. Visualizar la clasificación de los protocolos	89
4.13. Historia de Usuario SGREAR-12. Mostrar los códigos de jerarquía	90
4.14. Historia de Usuario SGREAR-13. Mayores consumos por usuarios	91

4.15. Historia de Usuario SGREAR-14. Mostrar las conexiones realizadas a los diferentes dominios	92
4.16. Historia de Usuario SGREAR-15. Visualizar las conexiones por tipo de contenido	93
4.17. Historia de Usuario SGREAR-16. Visualizar las direcciones IP de los servidores	94
4.18. Historia de Usuario SGREAR-17. Permitir la autenticación con el sistema	95
4.19. Historia de Usuario SGREAR-18.1. Adicionar usuario	96
4.20. Historia de Usuario SGREAR-18.2. Eliminar usuario	97
4.21. Historia de Usuario SGREAR-18.3. Cambiar contraseña de usuario	98
4.22. Descripción de la clase VentanaPrincipal	99
4.23. Descripción de la clase Generador	101
4.24. Descripción de la clase Reporte	103
4.25. Descripción de la clase Conexión	105
4.26. Descripción de la clase Configuración	106
4.27. Descripción de la colección Domains	106
4.28. Descripción de la colección Logs	108
4.29. Descripción de la colección Usuarios	108
4.30. Caso de prueba de aceptación SGREAR-08	109
4.31. Caso de prueba de aceptación SGREAR-09	110
4.32. Caso de prueba de aceptación SGREAR-10	111
4.33. Caso de prueba de aceptación SGREAR-11	111
4.34. Caso de prueba de aceptación SGREAR-12	112
4.35. Caso de prueba de aceptación SGREAR-13	113
4.36. Caso de prueba de aceptación SGREAR-14	114
4.37. Caso de prueba de aceptación SGREAR-15	115
4.38. Caso de prueba de aceptación SGREAR-16	116
4.39. Caso de prueba de aceptación SGREAR-17	117
4.40. Caso de prueba de aceptación SGREAR-18.1	117
4.41. Caso de prueba de aceptación SGREAR-18.2	118
4.42. Caso de prueba de aceptación SGREAR-18.3	119

Introducción

Internet es una inmensa red que conecta computadoras a nivel mundial, la cual le permite a millones de personas tener la posibilidad de comunicarse, compartir información y servicios de forma directa. Pese a las posibilidades que ofrece, para facilitar muchas de las actividades humanas y contribuir a una mejor satisfacción de las necesidades y del desarrollo personal, su uso también conlleva riesgos, especialmente para los niños, los adolescentes y las personas en general. Muchos de estos riesgos son contenidos ilegales tales como pornografía infantil, informaciones racistas, violentas, difamatorias, contenidos que violan la intimidad o la reputación de una persona, las violaciones de la propiedad intelectual, el tráfico de personas, entre otras.

La lucha contra el acceso a estos sitios anti-éticos es de gran importancia a nivel mundial. La Administración de redes se encarga de instalar, configurar y supervisar sistemas que garantizan el funcionamiento de diferentes servicios, así como de velar el cumplimiento de políticas de seguridad relacionadas con el nivel de acceso a determinados sitios. Esta administración es un conjunto de técnicas pendientes a mantener una red operativa, eficiente, segura, constantemente monitorizada, con una planificación adecuada y propiamente documentada [1]. Los Administradores de redes tienen que incrementar todo lo concerniente a la seguridad de sus sistemas, los cuales tienen a su favor el uso de servidores proxy para gestionar el tráfico en la red y controlar el acceso a Internet, manteniendo así el nivel de protección requerida.

El servidor proxy actúa como intermediario entre un usuario de una estación de trabajo e Internet. Este puede controlar el acceso a los sitios web, así como almacenar páginas descargadas con frecuencia, en un servidor de caché¹ local. El servicio de caché mejora los tiempos de respuesta y reduce el tráfico en Internet [2]. Las conexiones al ser utilizado el servidor proxy se pueden rastrear al crear registros de actividad, conocidos como logs, que guardan metódicamente las peticiones de los usuarios cuando solicitan conexiones a Internet.

Los generadores de reportes estadísticos permiten el manejo de los registros de actividades de la navegación de los usuarios a través de un servidor proxy. Estos registros contienen la información relacionada con cada petición realizada por los usuarios, cuentan con el Identificador Uniforme de Recursos (URI, siglas del inglés *Uniform Resource Identifier*) de los sitios web, el día y la hora del

¹Memoria de acceso rápido de una computadora, que guarda temporalmente las últimas informaciones procesadas.

acceso, la cantidad de bytes² que han sido descargados, la relación de sitios denegados y los errores de autenticación entre otras informaciones que pueden ser de gran significado para los Administradores de redes.

En la Universidad de las Ciencias Informáticas (UCI) fue desarrollado el sistema Análisis Inteligente de Registros Proxy (AiresProxy) con una interfaz para la generación de reportes para usuarios, el cual permite consultar reportes estadísticos de la navegación de los trabajadores y estudiantes de forma individual. La dirección de Redes y Seguridad Informática utiliza varias herramientas para la generación de reportes generales sobre las conexiones y el funcionamiento del servidor proxy, sin embargo estas herramientas no utilizan los datos procesados por AiresProxy. El empleo de varias herramientas trae consigo la necesidad de guardar la misma información de forma diferente y un mayor trabajo para los Administradores de redes. El sistema AiresProxy actualmente no permite el procesamiento de información sobre las conexiones de los usuarios y de las actividades del servidor proxy para la generación de reportes estadísticos para los Administradores de redes. Debido a esta situación se plantea el siguiente **problema a resolver**: ¿Cómo posibilitar la usabilidad de la información de la navegación en Internet, a partir de los registros de AiresProxy, para Administradores de redes?

El **objeto de estudio** de la presente investigación se basa en los sistemas generadores de reportes estadísticos de la navegación de los usuarios por Internet y el **campo de acción** en los sistemas generadores de reportes estadísticos de la navegación por Internet a través de un servidor proxy.

Para dar respuesta a la situación problemática planteada anteriormente se trazó el siguiente **objetivo general**: Desarrollar un sistema informático que genere reportes estadísticos sobre la navegación de los usuarios a través de un servidor proxy para los Administradores de redes.

Para dar cumplimiento al objetivo planteado se definen los siguientes **objetivos específicos**:

- ▶ Definir las tecnologías y herramientas para la implementación del sistema de generación de reportes estadísticos para Administradores de redes.
- ▶ Diseñar el sistema informático de generación de reportes estadísticos sobre la navegación de los usuarios y del funcionamiento del servidor proxy, para Administradores de redes.
- ▶ Implementar las funcionalidades que permitan la visualización de reportes sobre la navegación de los usuarios por un servidor proxy.

² Unidad de información compuesta por una secuencia de bits, tiene diversos múltiplos, como kilobyte (1.000 bytes), megabyte (1.000.000 bytes), gibabyte (1.000.000.000 bytes), terabyte (1.000.000.000.000 bytes).

- ▶ Realizar las pruebas de funcionalidad al sistema informático desarrollado.

Métodos científicos

Para el arribo al resultado final se utilizan los siguientes **métodos teóricos**:

- ▶ El **Analítico-Sintético** se utilizó en la identificación de las características de diversas aplicaciones relacionadas con el objeto de estudio, con el objetivo de extraer los elementos más importantes para el desarrollo de un sistema generador de reportes estadísticos de la navegación de los usuarios en Internet.
- ▶ El **Histórico-Lógico** permitió una mayor comprensión del estado actual y de la evolución que se ha llevado a cabo en el desarrollo de herramientas similares.

Como resultado de la investigación se espera obtener un sistema informático que genere reportes estadísticos para que los Administradores de redes mantengan el control y seguimiento de la navegación de los usuarios por Internet. Logrando como relevancia social un nuevo aporte de control y de análisis en las áreas administrativas, ayudando en la toma de decisiones sobre el posicionamiento de los sitios web y de las acciones de los usuarios en la red.

Estructura del documento

El contenido del presente trabajo está estructurado en cuatro capítulos distribuidos de la siguiente manera:

Capítulo 1. “Fundamentación teórica”: Contiene la fundamentación teórica de la investigación, se exponen los lenguajes, herramientas, y metodologías en los que se apoya la solución propuesta para el problema. Se realiza un análisis de las tendencias actuales y el estado del arte de los diferentes sistemas que generan reportes estadísticos de los registros de navegación de los usuarios.

Capítulo 2. “Descripción y características del sistema AiresProxy para Administradores de redes”: Se efectúa la descripción de la posible solución. Se especifica el diseño y la arquitectura que formará parte de la misma, se detalla el posible funcionamiento del sistema a desarrollar, así como los elementos que lo integran.

Capítulo 3. “Implementación del sistema generador de reportes estadísticos”: En este capítulo se exponen los elementos fundamentales del desarrollo ágil del software. Se representa la vista estática y

la distribución física del sistema.

Capítulo 4. “Validación de la solución propuesta”: En este capítulo se presentan las pruebas realizadas a la aplicación para comprobar el correcto funcionamiento de las principales funcionalidades del sistema, así como un análisis de los resultados obtenidos a partir de las pruebas aplicadas.

Capítulo 1

Fundamentación teórica

1.1. Introducción

En este capítulo se realiza un estudio del arte referente a los analizadores de logs tanto nacionales como internacionales, sus características principales y su funcionamiento. Se analizan las ventajas de los sistemas generadores de reportes estadísticos de la navegación de los usuarios por Internet, de los servidores proxy, de las aplicaciones de escritorios, así como de las herramientas, lenguajes y metodologías a utilizarse en el desarrollo del software.

1.2. Principales definiciones

1.2.1. Archivos logs

Un log es un registro oficial de eventos durante un rango de tiempo en particular. En informática, principalmente seguridad informática, es utilizado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué un evento ocurre para un dispositivo en particular o aplicación [3]. Los archivos logs registran todos los accesos a su alojamiento, guardando información acerca del Protocolo de Internet (IP, siglas del inglés *Internet Protocol*) donde se ha realizado la conexión, la fecha, la hora, la página a la que se accede, entre otras. A partir del análisis de la información de los registros logs, existen herramientas que generan reportes estadísticos.

1.2.2. Servidor proxy

Un servidor proxy actúa de intermediario entre un explorador web e Internet. Puede utilizarse para registrar el uso de Internet y también para bloquear el acceso a un sitio web. Los servidores proxy ayudan a mejorar el rendimiento en Internet ya que almacenan una copia de las páginas web más utilizadas. La finalidad más habitual del servidor proxy, es permitir el acceso a Internet a todos los equipos de una organización cuando sólo se puede disponer de un único equipo conectado, es decir, una única dirección IP. Cuando un explorador solicita una página web almacenada en la colección (su caché) del servidor

proxy, este la proporciona, lo que resulta más rápido que consultar la web. Los servidores proxy también mejoran la seguridad, ya que filtran algunos contenidos web y software malintencionado [4].

El principio operativo básico de un servidor proxy es bastante sencillo, se trata de un servidor que actúa como “representante” de una aplicación efectuando solicitudes en Internet en su lugar. De esta manera, cuando un usuario se conecta a Internet con una aplicación del cliente configurada para utilizar un servidor proxy, la aplicación primero se conectará con el servidor proxy que le dará la solicitud. El servidor proxy se conecta entonces al servidor de la aplicación que el cliente desea conectarse y le envía la solicitud. Seguidamente, el servidor le envía la respuesta al proxy, el cual a su vez la envía a la aplicación del cliente [4].

Al utilizar un servidor proxy, las conexiones pueden rastrearse al crear registros de actividad (logs) para guardar sistemáticamente las peticiones de los usuarios cuando solicitan conexiones a Internet. Gracias a esto, las conexiones de Internet pueden filtrarse al analizar tanto las solicitudes del cliente como las respuestas del servidor. El filtrado que se realiza comparando la solicitud del cliente con una lista de solicitudes autorizadas se denomina lista blanca; y el filtrado que se realiza con una lista de sitios prohibidos se denomina lista negra. Finalmente, el análisis de las respuestas del servidor que cumplen con una lista de criterios (como palabras clave) se denomina filtrado de contenido [5].

1.2.3. Logs basados en un servidor proxy

El empleo de servidores proxy permite una serie de mejoras en la calidad de los servicios ofrecidos a los clientes de un servidor web, el conjunto de solicitudes de los clientes cuyos buscadores están configurados para utilizar el proxy, son resueltas por este de manera similar a como un servidor web resuelve esas solicitudes, es posible por tanto capturar los eventos generados a partir de las solicitudes de los “clientes del proxy” y almacenarlas en un archivo log [5].

En general, los proxy hacen posible [6]:

- ▶ **Control:** Sólo el intermediario hace el trabajo real, por tanto se pueden limitar y restringir los derechos de los usuarios y dar permisos sólo al proxy.
- ▶ **Ahorro:** Sólo uno de los usuarios (el proxy) ha de estar equipado para hacer el trabajo real.
- ▶ **Velocidad:** Si varios clientes van a pedir iguales recursos, el proxy puede hacer caché: guardar la respuesta de una petición para darla directamente cuando otro usuario la pida. Así no tiene que volver a contactar con el destino y finaliza más rápido.

- ▶ **Filtrado:** El proxy puede negarse a responder algunas peticiones si detecta que están prohibidas.
- ▶ **Modificación:** Como intermediario que es, un proxy puede falsificar información, o modificarla siguiendo un algoritmo.
- ▶ **Anonimato:** Si todos los usuarios se identifican como uno sólo, es difícil que el recurso accedido pueda diferenciarlos.

1.2.4. Servidor proxy-caché

Un servidor proxy-caché permite incrementar la velocidad de acceso a Internet al mantener localmente las páginas más consultadas por los usuarios, evitando las conexiones directas con los servidores remotos.

Cuando un usuario, que tiene configurado el uso del proxy-caché, solicita una petición al servidor web, el navegador irá primeramente al proxy-caché y este verá en su base de datos si tiene dicha página; en caso afirmativo el proxy verifica que no ha expirado, que el objeto se corresponde con el actual, si es correcto la envía directamente al usuario; en caso de que no se encuentre la petición o no este actualizada, el proxy descarga el elemento solicitado y lo sirve al cliente [7].

Algunos paquetes que pueden ser usados como proxy-caché son Squid, Microsoft ISA Server y Blue Coat.

Squid

Es un proxy-caché web para el apoyo a HTTP, HTTPS, FTP y muchos más. Reduce el ancho de banda y mejora los tiempos de respuesta de almacenamiento en caché. Reutiliza las páginas web solicitadas con frecuencia. Tiene amplios controles de acceso y es un gran servidor del acelerador. Es utilizado por cientos de proveedores de Internet para proporcionar a sus usuarios el acceso a la web lo más factible posible, se encuentra licenciado bajo los términos de la Licencia Pública General GNU (GNU/GPL, siglas del inglés *GNU General Public License*). Squid permite especificar otros servidores intermediarios, utilizando la caché en una jerarquía como padres o como hermanos, dependiendo de la topología de la red estos pueden operar en cascada (padres) o en paralelo (hermanos) [8]. (Ver Anexo 1)

Microsoft ISA Server

Servidor de Seguridad y Aceleración de Internet (ISA Server, siglas del inglés *Internet Security and Acceleration Server*) permite a los administradores crear políticas para el uso de la regulación en función del usuario, grupo, aplicación, criterios de tipo destino, horario y contenido. ISA Server está diseñada para funcionar con Windows 2000 y sistemas operativos posteriores y para tomar ventaja de la seguridad

de Windows Kerberos (sistema de autenticación utilizado para comprobar la identidad de un usuario o máquina). El servidor ISA Proxy Web se puede configurar para mantener una caché de objetos web y tratar de cumplir con las solicitudes web de la caché [9].

Blue Coat

Blue Coat además de acelerar la respuesta de su enlace con Internet permite incrementar la seguridad de la red bloqueando sitios no autorizados, controlando el tipo de archivos que se bajan de Internet y hasta el contenido de los mensajeros instantáneos. El filtrado de contenido que contiene Blue Coat, permite autorizar o prohibir el acceso de los usuarios a los sitios de Internet a través de categorías [10].

1.3. Sistemas generadores de reportes estadísticos

La navegación por Internet no es más que el camino de cada servidor a cada petición, los distintos sitios web visitados, el registro de la estancia y de cada acción del internauta. Cada vez que se participa en un foro de debate o se rellena un formulario web, los servidores registran automáticamente la visita. Esta información contiene la dirección IP, el navegador que se utiliza, sistema operativo y tiempo de permanencia, entre otros datos que generalmente se emplean con fines estadísticos [11]. Sin embargo, en algunas ocasiones, es necesario recurrir a estos registros para revelar la identidad de personas que se creen impunes en la red.

Los registros oficiales de eventos son utilizados por Administradores de redes para registrar datos sobre quién, qué, cuándo, dónde y por qué un evento ocurre para un dispositivo o aplicación. Cada servidor de Internet cuenta con un log que registra todas las visitas de los usuarios, a razón de una línea de texto por cada acceso al servidor. En el caso de sitios web con gran tráfico, los archivos logs pueden llegar a superar los 100 megabytes diarios. Estos archivos son utilizados con fines estadísticos y en los casos de delitos cibernéticos [11].

Los sistemas generadores de reportes estadísticos, son aplicaciones comerciales o de libre disposición que generan diversas estadísticas a partir de la información rescatada de estos archivos, estadísticas que se basan principalmente en la información factible que se obtiene a partir de los datos almacenados [12]. A partir de la información almacenada en los archivos logs estos sistemas pueden generar reportes tales como visualizar las páginas más visitadas, las mayores descargas realizadas, las fallas de autenticación, entre otros muchos reportes estadísticos.

Los generadores de reportes estadísticos son programas que analizan los logs de los servidores

proporcionando grandes ventajas como por ejemplo [12]:

- ▶ Los servidores siempre producen ficheros de logs, por lo que la información está siempre disponible.
- ▶ Los servidores capturan la totalidad de los accesos a los sitios.
- ▶ La información normalmente reside en los propios servidores y tiene un formato estandarizado. Esto facilita la migración de unas herramientas a otras.
- ▶ Los ficheros de logs almacenan información sobre las peticiones fallidas, mientras que con otras técnicas, ésta se pierde.

1.3.1. Ámbito Internacional

En el mundo existen potentes analizadores de logs, por lo que se hace necesario caracterizar los analizadores más utilizados para tener una idea del principio de funcionamiento de un analizador de logs. De esta forma se podrán tomar elementos positivos característicos de los programas analizados e incluirlos como funcionalidad en la nueva propuesta. Entre los principales se encuentran: Calamaris, Sarg, Logrep y Sawmill. A continuación se hace una breve descripción de los sistemas antes mencionados.

Nombre del Sistema	Sistemas Operativos	Licencia
Calamaris	GNU/Linux	GPL
Sarg	GNU/Linux	GPL
Logrep	GNU/Linux, Windows	OpenSource, Freeware
Sawmill	Windows, GNU/Linux, FreeBSD, OpenBSD, Mac OS, Solaris y Unix	Propietaria

Tabla 1.1: Analizadores de logs

Calamaris

Está implementado para analizar los logs generados por el servidor proxy Squid. Es capaz de generar tanto informes en texto plano como en páginas HTML³, con estadísticas de intereses ordenadas por tablas. Existen versiones que generan gráficos de estas estadísticas. Esta herramienta está construida bajo licencia GPL. Escrito y desarrollado en el lenguaje Perl [13].

Este analizador es muy útil para especialistas de redes, por la gran cantidad de información que

³ Lenguaje de Marcado de Hipertexto (siglas del inglés *HyperText Markup Language*, versión 5)

puede mostrar, representa las estadísticas de una forma muy atractiva visualmente, contiene muchas funcionalidades factibles, que pueden servir de guía para la aplicación propuesta, de igual forma sus reportes son muy generales, enfocados a la administración de la navegación sin realizar un análisis detallado del mismo. En general el sistema es un buen ejemplo y puede servir como punto de partida para el desarrollo de la aplicación propuesta.

Sarg

Generador de reportes de análisis de Squid, es una herramienta que permite llevar un control de la navegación que realizan los usuarios por Internet. Este producto proporciona información acerca de las actividades de los usuarios del proxy: tiempo, bytes, sitios, entre otras. Dicho software, construye una estadística específica de usuario, se encuentra construida bajo licencia GPL y tiene soporte para generar reportes en diferentes idiomas [14].

Sarg está configurado para generar reportes web de los accesos a Internet de forma periódica, además de poder ejecutarlo manualmente para generar reportes de fechas, usuarios o dominios en específico. El sistema genera informes en HTML, con muchos campos, como por ejemplo: usuarios, direcciones IP, sitios web, entre otros. Sarg contiene un buen ambiente de trabajo, de manera sencilla, recomendado por el alto nivel de usabilidad que presenta y debido a la facilidad con que se puede operar. Una desventaja que lo dificulta es la lentitud con la que puede llegar a funcionar, a pesar de esto se puede tomar como referencia para la solución propuesta.

Logrep

Es una solución de código libre para centralizar los logs de los servidores, tanto para Linux como para Windows. Los logs se presentan en formato HTML a través del componente cliente de esta utilidad que es totalmente multiplataforma. El cliente es capaz de recolectar logs de 30 sistemas diferentes entre los que están: Snort, Squid, Postfix, Apache, Syslog, Ipchains, Qmail, Sendmail, Iptables, Servidores Windows, Firewall-1, Wtmp, Xferlog, Oracle y Pix [15]. Logrep presenta un protocolo de comunicación segura, permite guardar copias de todos los logs en un sistema central, consume pocos recursos y sus logs son representados gráficamente.

Logrep es una herramienta ideal para administradores de sistemas que necesiten centralizar la gestión de logs y tener un sistema de acceso rápido a todos ellos, es importante también a la hora de analizar una posible intrusión en la red. Soporta varios lenguajes, su contenido se encuentra bien estructurado y sencillo. La mayor restricción que posee este sistema es la instalación, debido a la pobre

y escasa documentación con que cuenta, haciendo difícil su estudio y análisis para la nueva propuesta a desarrollar.

Sawmill

Potente herramienta de análisis jerárquico optimizado para reportes Web, en su proceso es capaz de soportar 818 formatos de logs. Además de procesar logs de formatos combinados y ampliados, genera estadísticas de ellos, presentación de informes y análisis de acontecimientos. Tiene la capacidad de exportar los datos hacia una base de datos MySQL, Microsoft SQL Server, Oracle o a la suya propia. Genera y agrega informes de filtrado de forma dinámica, todos a través de una interfaz web. Puede realizar análisis de registros en cualquier plataforma, incluyendo Windows, Linux, FreeBSD, OpenBSD, Mac OS, Solaris y Unix [16].

El hecho de ser un software altamente configurable, de gran facilidad de uso y que permite procesar gran variedad de reportes mostrando las estadísticas en forma de gráficas, lo hace uno de los sistemas más utilizados de su tipo. Este sistema es muy dinámico a la hora de filtrar la información procesada de los registros, lo cual permite extraer estadísticas que no pueden obtenerse con otros analizadores de logs. Hay que señalar que el sistema posee una estructura y características que son de interés para la aplicación que se desea desarrollar, aunque tiene como deficiencia que está construida bajo licencia propietaria haciendo costoso su mantenimiento y actualización, de igual manera sirve de ayuda para el sistema que se desea implementar.

1.3.2. Ámbito Nacional

En Cuba existen sistemas similares los cuales tienen elementos positivos para el desarrollo del sistema propuesto. En la Universidad de las Ciencias Informáticas y en el Instituto Central de Investigaciones Digitales (ICID) se han creado este tipo de aplicaciones.

Isaweb

El Instituto Central de Investigaciones Digitales (ICID) es una experimentada y moderna entidad cubana, que trabaja en los campos de Automatización y Sistemas Integrados, en el 2006 crearon un software denominado Isaweb, el cual permite monitorizar el tráfico en Internet. Esta aplicación permite que un usuario autenticado pueda filtrar información relevante sobre el acceso a Internet. Esta información se extrae del servidor "ISA Server", se transforman los ficheros de las trazas de la navegación y se

almacenan en una base de datos SQL Server [17].

Esta es una aplicación web y su interfaz es sencilla, pero la mayor restricción que posee es que está creada especialmente para la entidad ICID por lo tanto dificulta su aprovechamiento a la hora de crear el sistema propuesto.

AiresProxy

La Universidad de las Ciencias Informáticas cuenta con un sistema que permite el reporte de datos estadísticos a partir del análisis de los logs de un servidor proxy. Con el uso de técnicas de minería de datos, inteligencia artificial y estadísticas, AiresProxy posibilita tener datos reales sobre las actividades de los usuarios en la red tanto para trabajadores como para estudiantes de forma individual. Procesa y almacena en una base de datos la información relacionada con los logs del servidor proxy Squid.

AiresProxy visualiza de forma dinámica la información estadística generada. Este sistema muestra reportes tales como: reporte general, dominios por IP, dominios por fecha, localizador de recursos uniforme (URL, sigla del inglés *Uniform Resource Locator*), total por fecha y total por IP, mostrando además como datos característicos el consumo real y el consumo UCI.

El uso de esta herramienta es de gran beneficio ya que permite a los usuarios mantener el conocimiento de sus trazas por Internet pero no cuenta actualmente con reportes especiales para Administradores de redes. La desventaja que tiene el sistema, es que ocupa gran espacio en disco según la cantidad de usuarios que estén navegando por el proxy.

El estudio de esta herramienta servirá de guía y de ayuda a la hora de desarrollar el sistema propuesto, tomando para ello las mejores prácticas y funcionalidades, ya sea a la hora de generar reportes estadísticos o a la hora de implementar una interfaz amigable y usable.

1.4. Determinación del tipo de aplicación

Para el desarrollo de un sistema es necesario ajustar las necesidades y definir el tipo de aplicación que se desea implementar, analizando para ello las aplicaciones de escritorio y las aplicaciones web.

Aplicaciones de escritorio

Las aplicaciones de escritorio están diseñadas para cubrir las necesidades del negocio y administrar la información interna. Con las aplicaciones de escritorio se puede contar con altos niveles de rendimiento, velocidad de respuesta y seguridad [18]. Estas aplicaciones han sido desarrolladas para ser ejecutadas

en una plataforma específica, ya sea Windows, Linux o Mac. El desarrollo sobre una plataforma, normalmente, implica que la aplicación “no” pueda ser ejecutada en otras [19]. Por otro lado se puede señalar que existen aplicaciones de escritorio que son multiplataforma como por ejemplo la plataforma Java, la cual provee una máquina virtual que ejecuta cualquier código que haya sido escrito en dicho lenguaje, siendo esta máquina virtual, ejecutada por el sistema operativo.

Entre las ventajas de las aplicaciones de escritorio, se pueden destacar:

- ▶ Mayor capacidad gráfica visual.
- ▶ Menor tiempo de respuesta (aplicación más rápida).
- ▶ Mayor personalización cubriendo características específicas de la empresa.

Aplicaciones web

Actualmente las aplicaciones web son muy populares, las mismas pueden ser operadas desde cualquier ordenador con conexión a Internet, consumen muy pocos recursos del equipo en el que se encuentran y su funcionamiento es independiente del sistema operativo. Poder mostrar y acceder a la información a través de Internet ofrece una gran ventaja competitiva. Con una aplicación web, se pueden publicar catálogos electrónicos de productos, manejar inventarios, facturación, mantenimiento y actualización de sitios web [18].

Entre las ventajas de las aplicaciones web, se pueden destacar:

- ▶ No hay problemas de compatibilidad, basta tener un navegador actualizado para poder utilizarlas.
- ▶ No ocupan espacio en el disco duro.
- ▶ Multiplataforma.

Especificando el tipo de aplicación

Debido a que el sistema generador de reportes estadísticos sobre la navegación de los usuarios por Internet, no requiere comunicación con el exterior sino que su ejecución debe realizarse de forma local para los Administradores de redes, se decide desarrollar una aplicación de escritorio. La misma tendrá buena velocidad de procesamiento, tiempo de respuesta y seguridad.

Esta aplicación se instalará en cada cliente (computadora) aprovechando los recursos del sistema operativo y la información contenida en ellas no estará expuesta a personas no autorizadas a través de Internet. Se debe agregar que las herramientas anteriormente analizadas son aplicaciones web donde

la mayoría de ellas analizan un log y no un lote de ellos, por lo que su rendimiento, muchas veces, no se afecta con esta característica; sin embargo la aplicación que se desea implementar debe trabajar con lotes de log donde el rendimiento y la rapidez de los reportes es de vital importancia.

1.5. Metodología, lenguajes y herramientas a utilizar

1.5.1. Metodología para el desarrollo del software

La selección de la metodología de desarrollo de software a utilizar, es una labor importante, ya que la calidad del producto final estará dada en gran medida por este aspecto. Las metodologías ágiles surgen como una alternativa a las metodologías tradicionales. Se centran en el individuo, la colaboración con el cliente y se ajustan a proyectos de corta duración y cuya planificación no debe ser estricta. Ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva [20]. Entre las metodologías más notables se encuentran: SCRUM, Programación Extrema (XP, siglas del inglés *eXtreme Programming*) y SXP (SCRUM + XP). A continuación se hace un estudio de estas metodologías.

SCRUM

Más que una metodología de desarrollo de software, es una forma de auto-gestión de los equipos de programadores. Un grupo de programadores deciden cómo hacer sus tareas y cuánto van a tardar en ello. SCRUM ayuda a que trabajen todos juntos, en una sola dirección, con un objetivo claro. Permite además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo. Es una de las metodologías ágiles más conocidas, y se basa en un enfoque iterativo, donde cada iteración se denomina *Sprint*. El principio básico es que es muy difícil contar desde el principio con un catálogo completo de funcionalidades, ya que los requisitos van surgiendo conforme el propietario de la aplicación y los usuarios van haciendo sucesivas aportaciones [20]. SCRUM plantea el desarrollo de sucesivas versiones ampliadas, todas ellas plenamente usables y evaluables por el usuario. SCRUM es, además, una metodología especialmente indicada para pequeños equipos de desarrollo y se orienta a una entrega rápida de resultados y una alta flexibilidad.

XP

Programación Extrema (XP), utilizada para proyectos de corto plazo y equipos pequeños, ha generado gran interés por su creciente número de casos de éxito en la industria. Metodología ágil que requiere gran disciplina. Un proceso ligero de bajo riesgo, flexible, predecible, científico y divertido de desarrollar

software. Su utilidad se mide en valores como la simplicidad, comunicación, realimentación y coraje. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Las entregas son frecuentes, y existe una refactorización continua, lo que permite mejorar el diseño cada vez que se le añade una nueva funcionalidad [21]. Sustenta la propiedad colectiva, la programación en parejas y pequeños encuentros diarios donde cada uno de los integrantes del equipo cuenta que ha hecho, que problemas tiene, y que hará más adelante.

SXP

SXP (SCRUM + Programación Extrema) es una metodología ágil desarrollada como un híbrido de SCRUM y XP, para coleccionar documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. Garantizando la eficacia y la eficiencia en su proceso de generación. Con el uso de SCRUM, para la gestión del trabajo de forma eficiente y de tener medidos los procesos, SXP logra resultados satisfactorios y toma de XP las mejores prácticas que guían el desarrollo del software. SXP es ideal para proyectos de corta duración con requisitos cambiantes o no bien definidos, donde prevalezca la retroalimentación entre el cliente y el equipo de trabajo [20]. El desarrollo con SXP se realiza en iteraciones cortas (*sprints*) a lo largo de 3 fases, Planificación-Definición, Desarrollo y Entrega-Mantenimiento, para cumplir con un grupo de actividades, de las que se generan una serie de artefactos que documentan el proceso de desarrollo, obteniendo un *release* (liberación) del producto con nuevas funcionalidades.

Argumentos de la metodología seleccionada

Luego de analizar las características de las metodologías de desarrollo de software antes planteadas, se pudo apreciar que están diseñadas para elevar la calidad de la producción de software y la labor de los desarrolladores. SCRUM se basa principalmente en la gestión del trabajo de los programadores, XP en la programación rápida pero disciplinada y SXP en la combinación de las dos anteriormente mencionadas, por lo que se decide utilizar como metodología de desarrollo SXP.

Esta metodología seleccionada fue creada para equipos pequeños, con iteraciones igualmente pequeñas y utiliza las mejores estrategias de las metodologías SCRUM y XP. Se usará SCRUM para la correcta planificación del proyecto ya que no es una metodología de análisis, ni de diseño, es una metodología de gestión del trabajo; mientras que XP respalda con sus prácticas todo el proceso de desarrollo, obteniéndose de esta forma un proceso de software completo. SXP cuenta además con una buena documentación y existe en la universidad personal altamente calificado de conocimientos sobre esta

metodología.

1.5.2. Lenguaje de modelado

Los lenguajes de modelado de objetos son símbolos estandarizados que permiten modelar parte de un diseño de software. Gracias a estos modelos se puede trabajar con sistemas complejos y abstraerse de las diferentes situaciones permitiendo así una mejor implementación del software a desarrollar. Estos lenguajes de modelado son independientes del lenguaje de programación que se quiera utilizar. La Notación para el Modelado de Procesos de Negocio (BPMN, siglas del inglés *Business Process Modeling Notation*) y el Lenguaje de Modelado Unificado (UML, siglas del inglés *Unified Modeling Language*) son dos de los lenguajes más utilizados en la Industria de Software, a continuación se plantean algunas de sus características:

BPMN

La Notación para el Modelado de Procesos de Negocio brinda la capacidad de comprensión de los procesos de negocio internos en una notación gráfica dándole a las organizaciones la capacidad de comunicar estos procedimientos de una manera estándar. Por otra parte, la notación gráfica facilitará la comprensión de las colaboraciones de rendimiento y las transacciones comerciales entre las organizaciones. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes. Esto asegura que las empresas se entiendan ellas mismas y de los participantes en sus negocios [22]. BPMN está planeada para dar soporte únicamente a aquellos procesos que sean aplicables a procesos de negocios, lo cual significa que cualquier otro tipo de modelado, con fines distintos a los del negocio, no estará en el ámbito de BPMN.

UML

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Grupo de Gestión de Objeto (OMG, siglas del inglés *Object Management Group*). El UML es un grupo de especificaciones de notación orientadas a objeto, las cuales están compuestas por distintos diagramas, que representan las diferentes etapas del desarrollo de un software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Propone un estándar para el intercambio técnico de modelos y diseños de código reutilizable [23]. Es una consolidación de muchas

de las notaciones y conceptos más usados.

Argumentos del lenguaje de modelado seleccionado

Tras haber analizado los lenguajes de modelado antes expuestos, se puede comprender que permiten modelar procesos de negocio altamente flexibles, BPMN toma un perfil orientado a los procesos del negocio en el modelado de aplicaciones, mientras que UML toma un perfil orientado a objetos, enfocado en el diseño del sistema. Es por eso que se selecciona como lenguaje de modelado el UML en su versión 2.1, ya que es capaz de definir, detallar, documentar y construir un sistema de software, permite igualmente modelar los procesos de negocio siendo un lenguaje más abarcador con relación al sistema que se desea implementar. Es un lenguaje fácil de aprender, adaptable a cambios y ofrece gran variedad de diagramas para visualizar el software desde varias vías, facilitando así la relación entre clientes y desarrolladores.

1.5.3. Lenguaje de programación

Un lenguaje de programación es un conjunto de símbolos y reglas que permiten crear software. Existen muchos lenguajes de programación entre los que se destacan como líderes de popularidad C++ y Java, por lo que a la hora de elegir un lenguaje de programación para el desarrollo de una aplicación hay que definir bien el que se va a utilizar, así como la experiencia de los desarrolladores.

C++

El lenguaje de programación C++ es uno de los más empleados en la actualidad. Se puede decir que C++ es un lenguaje versátil, potente, híbrido, ya que permite programar tanto en estilo procedimental (como si fuese C), como en estilo orientado a objetos, como en ambos a la vez. Además, también se puede emplear mediante programación basada en eventos para crear programas que usen interfaz gráfica de usuario [24]. Su éxito entre los programadores profesionales le ha llevado a ocupar el primer puesto como herramienta de desarrollo de aplicaciones. Abarca tres paradigmas de la programación estructurada, la programación genérica y la programación orientada a objetos. Es un lenguaje de alto nivel, con buena velocidad de respuesta y puede ser compilado en muchos sistemas operativos.

Entre sus ventajas se encuentran [25]:

- ▶ **Difusión:** Al ser uno de los lenguajes más empleados en la actualidad, posee un gran número de usuarios y existe una gran cantidad de libros, cursos y páginas web, dedicados a él.
- ▶ **Versatilidad:** C++ es un lenguaje de propósito general, por lo que se puede emplear para resolver

cualquier tipo de problema.

- ▶ Portabilidad: El lenguaje está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.
- ▶ Eficiencia: C++ es uno de los lenguajes más rápidos en cuanto ejecución.
- ▶ Herramientas: Existe una gran cantidad de compiladores, depuradores y librerías.

Java

Java es un lenguaje de programación orientado a objetos, toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen persuadir muchos errores, como la manipulación directa de punteros o memoria. Se utiliza en los principales sectores de la industria de todo el mundo y está presente en un gran número de dispositivos, equipos y redes [25]. La versatilidad y eficiencia de la tecnología Java, la portabilidad de su plataforma y la seguridad que aporta, la han convertido en una gran tecnología.

Java ha sido probado, mejorado y ampliado por una comunidad especializada de más de 6,5 millones de desarrolladores. Java se ha convertido en un recurso inestimable ya que permite a los desarrolladores [25]:

- ▶ Desarrollar software en una plataforma y ejecutarlo en cualquier otra plataforma.
- ▶ Crear programas para que funcionen en un navegador web y en servicios web.
- ▶ Desarrollar aplicaciones para servidores como por ejemplo, foros en línea, tiendas, encuestas, procesamiento de formularios HTML.
- ▶ Combinar aplicaciones o servicios que usan el lenguaje Java para crear servicios o aplicaciones totalmente personalizados.
- ▶ Desarrollar potentes y eficientes aplicaciones para teléfonos móviles, procesadores remotos, productos de consumo de bajo costo y prácticamente cualquier tipo de dispositivo digital.

Argumentos del lenguaje de programación seleccionado

De los lenguajes de programación anteriormente expuestos se puede concluir que ambos son lenguajes significativos para el desarrollo de software, pero, para la implementación del sistema propuesto se seleccionó C++ ya que se necesita de un lenguaje capaz de soportar una aplicación de escritorio con

una carga de datos significativa, es un lenguaje con abundante documentación y bien rápido en cuanto a ejecución.

1.5.4. Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado (IDE, siglas del inglés *Integrated Development Environment*) es un programa informático compuesto por un conjunto de herramientas de programación. Un IDE consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI) [26]. Existen variedades de IDEs entre ellos Eclipse, Qt Creator y NetBeans.

Eclipse

Eclipse cuenta con una atractiva interfaz que lo hace fácil y agradable de usar. Es una completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java. Contiene un editor de texto donde se puede ver el contenido del fichero en el que se está trabajando, una lista de tareas, y otros módulos similares. Eclipse trabaja con lenguajes para procesadores de texto, aplicaciones en red y sistema de gestión de base de datos. La arquitectura en forma de *plugin* permite escribir cualquier extensión deseada en el ambiente, como sería gestión de la configuración [27].

Qt Creator

Qt Creator es un excelente IDE multiplataforma para desarrollar aplicaciones en C++ de forma sencilla y rápida. Está basado y contiene toda la documentación de la biblioteca Qt, disponible para Linux, Mac OS X y las plataformas de Windows. Permite a los desarrolladores crear aplicaciones para escritorio y plataformas de dispositivos móviles. Qt Creator contiene un editor de código diseñado para crear, editar y navegar código. Está totalmente equipado con la comprobación de sintaxis, completado de código, ayuda sensible en línea y los indicadores de error al teclear. Posee un avanzado editor de código C++, además soportar los lenguajes: C#, Python, Ada, Pascal, Perl, PHP y Ruby. Posee también la Interfaz Gráfica de Usuario (GUI, siglas del inglés *Graphical User Interface*) integrada y diseñador de formularios. Qt Creator contiene herramientas para la administración y construcción de proyectos, ayuda sensible al contexto integrado, así como un depurador visual [28]. Este IDE permite el auto-completado de código y da soporte para refactorización de código.

NetBeans

NetBeans es un entorno de desarrollo integrado, modular, de base estándar (normalizado), escrito

en el lenguaje de programación Java. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general (*framework*) para compilar cualquier tipo de aplicación. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. Posee todas las herramientas necesarias para crear aplicaciones web, de escritorio y para teléfonos móviles con los lenguajes de programación Java, C, C++ e incluso lenguajes dinámicos como PHP, JavaScript, Groovy y Ruby. Es fácil de usar, instalar y puede ser ejecutado en múltiples plataformas como Windows, Linux, Mac OS X y Solaris [29]. Además posee un número importante de módulos para extender las funcionalidades del IDE.

Argumentos del IDE seleccionado

El Eclipse y el NetBeans son excelentes IDEs para el desarrollo de aplicaciones, pero para la programación en C++ con la biblioteca Qt requieren la instalación de las dependencias necesarias para que funcione Qt, como son los compiladores, biblioteca Qt, ambientes de trabajo; mientras que Qt Creator viene con la instalación estándar de Qt, el cual se encuentra muy bien integrado con C++. Por lo que luego de analizar estas características de los IDEs antes mencionados y de comparar las ventajas que tienen cada uno, se selecciona como el entorno de desarrollo integrado para la solución propuesta el Qt Creator.

Este IDE elegido permite la implementación de aplicaciones de escritorio de forma rápida, es sumamente completo, moderno, potente, fácil de manejar, con gran eficiencia, es de código abierto y de licencia gratuita. Permite realizar programación visual donde se presta atención al diseño en el aspecto gráfico de la aplicación, a la escritura del código que se ejecutará y a la programación dirigida por eventos. La versión con la que se pretende implementar el sistema es la de Qt Creator 2.1.0, la cual contiene mejoras en todas las áreas, ya sea en el editor de código sofisticado, en el control de versiones, así como en la biblioteca Qt.

1.5.5. Herramientas de modelado de software

En la actualidad existen varias herramientas de modelado, que permiten la automatización del desarrollo de un software para mejorar la calidad y refinamiento de las aplicaciones, facilitar la reutilización de componentes de software además de estandarizar la documentación de los procesos; entre estas herramientas se encuentran: Visual Paradigm y Rational Rose Enterprise Edition.

Visual Paradigm

El Visual Paradigm proporciona un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de

los programas y la documentación; también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta licencia gratuita y comercial. Esta es una de las herramientas más potentes que existe en la actualidad para asistir el proceso de construcción del software. Útil para desarrollar software que utiliza la programación orientada a objetos [30]. Es una herramienta multiplataforma que se puede utilizar tanto en Linux como en Windows, la interfaz es intuitiva y de fácil aprendizaje para los desarrolladores, es capaz de generar diagramas automáticamente. Contiene un modelo y código que permanece sincronizado en todo el ciclo de desarrollo.

Rational Rose Enterprise Edition

Rational Rose es una herramienta de desarrollo basada en modelos que se integra con las bases de datos y los IDE de las principales plataformas del sector. Todos los productos de Rational Rose dan soporte a UML, pero no son compatibles con las mismas tecnologías de implementación. Rational Rose Enterprise es un entorno de modelado que permite generar código a partir de modelos Ada, ANSI C++, C++, Java/J2EE, Visual C++ y Visual Basic. Ofrece un lenguaje de modelado común que agiliza la creación del software. Permite especificar, analizar y diseñar el sistema antes de codificarlo y que se mantenga la consistencia de los modelos del sistema de software, el chequeo de la sintaxis UML, la generación de documentación automática, generación de código a partir de los modelos, crear modelos a partir del código (ingeniería inversa) y trabajo en grupo [31].

Argumentos de la herramienta seleccionada

Se selecciona como herramienta de modelado el Visual Paradigm (versión 8.0) proporcionando calidad y rapidez a la hora de implementar y desarrollar una aplicación. No se escoge, a pesar de ser muy buena herramienta, Rational Rose ya que no contiene el ciclo de vida completo del desarrollo del software y se basa en sistemas operativos Windows.

Visual Paradigm por otro lado, contiene todos los tipos de diagramas de clases, permite la generación de códigos, reportes y cuenta además con abundante documentación. Soporta el ciclo de vida completo del desarrollo del software ya sea análisis y diseño, construcción, pruebas y despliegue. Hay que destacar que es un sistema multiplataforma y su interfaz es sencilla y agradable. Cuenta con una adecuada organización, es un sistema bien completo y fácil de integrar con otras herramientas. Proporciona soporte a varios lenguajes en generación de código e ingeniería inversa, es sin duda la herramienta para modelar la propuesta realizada.

1.5.6. Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Por lo tanto debe permitir la especificación de estructuras, restricción, construcción y manipulación de las bases de datos, para así realizar consultas, modificar y actualizar los datos. Ejemplos de SGBD son Oracle, MongoDB, MySQL entre otros.

Oracle

Oracle es un sistema de gestión de base de datos objeto-relacional de licencia propietaria. El propósito general es almacenar y recuperar información relacionada. Se considera como uno de los sistemas de bases de datos más completos, destacando su soporte de transacciones, su estabilidad y su soporte multiplataforma. Presenta un gran dominio en el mercado, con la competencia del SQL Server de Microsoft y de los gestores de ofertas libres. Una base de datos Oracle es una colección de datos tratada como una unidad. Una instancia Oracle consta de una estructura de memoria, llamada Área Global del Sistema (SGA, siglas del inglés *System Global Area*), y de unos procesos de fondos utilizados por el servidor Oracle para manejar una base de datos [32]. Cada instancia Oracle puede abrir y utilizar sólo una base de datos en cualquier punto y momento. Permite la gestión de grandes bases de datos, con un alto rendimiento en transacciones.

MongoDB

MongoDB es un sistema de base de datos de alto rendimiento, multiplataforma orientado a documentos, de esquema libre, no relacional. Está escrito en C++ y cuenta con gran velocidad a la hora de ejecutar sus tareas. Además, está licenciado como GNU AGPL 3.0 con código abierto, de modo que se trata de un software de licencia libre. Funciona en sistemas operativos Windows, Linux, OS X y Solaris. MongoDB es otra de las muchas bases de datos NoSQL (no solo SQL) existentes, cuya particularidad es que intenta unir las ventajas de los almacenes clave-valor y de las bases de datos relacionales clásicas (RBDMS, siglas del inglés *Relational Database Management System*). MongoDB soporta consultas dinámicas (comportamiento análogo al de SQL), es decir, se pueden formular sobre cualquier valor de los documentos y no solamente en los indexados. Ofrece gran facilidad de almacenamiento de objetos/datos y consultas dinámicas [33]. Es un sistema sencillo, con las funcionalidades ofrecidas se está ganando la

atención de muchas personas.

MySQL

MySQL es un sistema manejador de bases de datos, multihilo y multiusuario. Es un sistema escrito en C y en C++, multiplataforma y de licencia libre. Es muy rápido en la lectura lo que la hace ideal en aplicaciones web con baja concurrencia en la modificación de datos pero intensivo en la lectura. Su popularidad como aplicación web está muy ligada al lenguaje de programación PHP, que a menudo aparece en combinación con MySQL. Es una base de datos muy rápida en la lectura, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones [34]. Es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación cuando se desea utilizar este gestor de base de datos.

Argumentos del SGBD escogido

Una vez analizados los sistemas antes expuestos y de comprender el entorno en que se desarrollan cada uno de ellos, se selecciona como sistema gestor de base de datos, para el desarrollo de la propuesta realizada, el MongoDB en su versión estable 1.6, la cual incluye la replicación y la auto-fragmentación. Los restantes sistemas son excelentes gestores de base de datos pero no contienen las características suficientes, Oracle es un sistema de licencia propietaria y MySQL es ideal para aplicaciones web con baja concurrencia, ambos son bases de datos relacionales dificultando la velocidad con la que se desea que la aplicación llegue a funcionar, debido al gran volumen de registros que se pretende procesar.

El MongoDB por su parte, es un gestor de bases de datos relativamente nuevo, con licencia libre, orientado a documentos por lo que no implementa el concepto de tablas, esquemas SQL o columnas, no tiene transacciones, joins (uniones), claves foráneas o cualquier otra característica que dificulte la comodidad y la rapidez con que se desea trabajar. Es un sistema que se destaca por su velocidad y facilidad de uso, considerada una base de datos muy diferente a las que se está acostumbrada pero sin dudas alcanza un balance perfecto entre rendimiento y funcionalidad.

1.5.7. Biblioteca

Una biblioteca, en términos de informática, es un conjunto de algoritmos prefabricados, que pueden ser utilizados por el programador para realizar determinadas operaciones.

Qt es utilizada en el Entorno de Escritorio K (KDE, siglas del inglés *K Desktop Environment*), un entorno de escritorio para sistemas como GNU/Linux o Windows, entre otros. Es producido por la división de software Qt de Nokia, que entró en vigor después de la adquisición por parte de Nokia de la empresa noruega Trolltech, su productor original. Qt utiliza el lenguaje de programación C++ de forma nativa, adicionalmente puede ser utilizado en otros lenguajes de programación a través de enlaces. Funciona en todas las principales plataformas, y tiene un amplio apoyo. La Aplicación de Interfaz de Programación (API, siglas del inglés *Application Programming Interface*) de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML, gestión de hilos, soporte de red, una API multiplataforma unificada para la manipulación de archivos y una multitud de otros para el manejo de ficheros, además de estructuras de datos tradicionales. Distribuida bajo los términos de GNU Licencia Pública General Reducida (del inglés *Lesser General Public License*) y otras, Qt es software libre y de código abierto [35].

Es seleccionada la biblioteca Qt en su versión 4.7.2 ya que es una biblioteca multiplataforma ampliamente usada para desarrollar aplicaciones con una interfaz gráfica de usuario, así como para el desarrollo de programas sin interfaz gráfica como por ejemplo, herramientas para la línea de comandos y consolas para servidores. Una de las diferencias fundamentales por la que se selecciona Qt, a parte de las características antes mencionadas, es que le agrega al C++ los conceptos de señales y puertos, definiendo como señal a un aviso que un objeto puede emitir cuando le ocurre un cambio de estado importante y un puerto se define como un método de un objeto que puede ser llamado cuando se genera una señal en particular. Las clases de Qt poseen señales y puertos predefinidos, sin embargo se pueden crear nuevos tipos.

1.5.8. Formato XML

El lenguaje de marcado extensible (XML, siglas del inglés *eXtensible Markup Language*) es un lenguaje de marcado para documentos que contengan información estructurada. XML proporciona una facilidad para definir etiquetas y las relaciones estructurales entre ellas. Como no hay un conjunto de etiquetas predefinidas, no puede haber ninguna semántica preconcebida. Todas las semánticas de un documento XML serán definidas por las aplicaciones que los procesan o las hojas de estilo. Un lenguaje de marcas

es un mecanismo para identificar estructuras en un documento. La especificación XML define una manera estándar de añadir etiquetas a los documentos [36].

XML posee una estructura más abierta y extensible, permite gestionar los datos aunque procedan de diversas fuentes y también permite el intercambio de documentos entre distintas aplicaciones ya estén en un único ordenador o en una red. Como la información en XML está mucho más estructurada, esto facilitará enormemente la labor a los buscadores que accederán a los datos de manera más precisa. Una de las características más importantes de XML en relación al hipertexto es la inclusión de enlaces multidireccionales (que apuntan a varios documentos). Además, hay que tener en cuenta la exportabilidad de XML a otros formatos de edición, ya que el documento maestro de la edición de un documento XML podría ser exportado de forma directa tanto a la web como para su edición en papel [36].

Las ventajas de adoptar XML para una posible solución, se basa principalmente en la libertad que tienen los autores de poder diseñar y crear sus propios documentos y etiquetas, dependiendo solamente de las funciones que le quieran dar a los datos. La información contenida en un documento XML puede ser fácil de usar, más accesible y reutilizable, ya que la flexibilidad de las etiquetas de XML así lo permiten.

1.6. Conclusiones

Una vez finalizado el presente capítulo se puede concluir que se señalaron los aspectos más importantes de diferentes generadores de reportes estadísticos de la navegación por Internet, los cuales por sí solos no satisfacen completamente las necesidades de los Administradores de redes de la universidad, evidenciando la necesidad de desarrollar una aplicación que permita cumplir con las características específicas para la generación de reportes estadísticos de la navegación por Internet. Con el objetivo de tomar las mejores prácticas de las aplicaciones estudiadas, se seleccionaron una serie de tecnologías las cuales ayudarán al desarrollo de una nueva aplicación.

Capítulo 2

Descripción y características del sistema AiresProxy para

Administradores de redes

2.1. Introducción

EN el presente capítulo se realiza la descripción de la aplicación, se presentan los requisitos funcionales y no funcionales. Se explican todas las funcionalidades del sistema en forma de historias de usuario. Se plantea la arquitectura del sistema y los patrones de diseño utilizados, así como el diagrama de clases del diseño, el diagrama de paquetes, el diagrama de dominio y la modelación de la base de datos.

2.2. Descripción del sistema a desarrollar

El departamento de Soluciones Informáticas para Internet (SINI), que pertenece al centro Centro de Ideo-Infornática (CIDI), en convenio con la Dirección de Redes y Seguridad Informática de la UCI, desarrolla un sistema informático denominado AiresProxy. Este sistema pretende la integración de tres subsistemas especializados en la representación de reportes personalizados, AiresProxy para Usuarios, AiresProxy para Auditores y AiresProxy para Administradores de redes.

Actualmente se encuentra liberado el subsistema de reportes estadísticos AiresProxy para Usuarios (Ver Anexo 2), este permite la generación de varios reportes asociados a las visitas que realizan los usuarios de una red interna a Internet a través de un proxy. Esta aplicación informática visualiza diferentes reportes para auditar y comparar los consumos asociados a los dominios, dominios por IP, direcciones y fechas en que se realizaron conexiones a Internet. La auditoría se realiza de forma individual por cada usuario dando la posibilidad a las personas que cuentan con acceso a Internet de comprobar su navegación. En el caso de los Administradores de redes pueden visualizar reportes de otros usuarios.

El segundo subsistema AiresProxy para Auditores se encuentra en la fase de planificación. El cual permitirá la creación y actualización de una estructura en forma de árbol para la consulta de datos según los departamentos identificados en una empresa o entidad. Mediante una autenticación segura con el

uso de roles, cada auditor podrá realizar un reporte de los departamentos a los que tiene acceso. El sistema dará la posibilidad de mostrar diferentes reportes a través de tablas y gráficos de forma directa, así como la exportación de estos en diferentes formatos.

El subsistema AiresProxy para Administradores de redes (Ver Anexo 3) que se desea implementar debe visualizar diferentes reportes agrupados en dos categorías, los reportes asociados a las visitas y conexiones a Internet y los asociados al funcionamiento y rendimiento del servidor proxy.

2.2.1. Reportes asociados a las visitas y conexiones a Internet

Los reportes que están relacionados con las conexiones a Internet se enfocan en las visitas realizadas por los usuarios, los accesos a los sitios web, en los tipos de contenidos y extensiones accedidos y en las descargas realizadas (Ver Anexos 4, 5, 6 y 7). Estas funcionalidades son las siguientes:

- ▶ Visualizar los dominios más visitados.
- ▶ Visualizar las mayores descargas.
- ▶ Visualizar los dominios que han sido visitados por un usuario.
- ▶ Visualizar todos los usuarios que visitaron un dominio.
- ▶ Mostrar un resumen de las conexiones realizadas a los diferentes tipos de dominios.
- ▶ Visualizar las conexiones realizadas por cada tipo de contenido.
- ▶ Visualizar los mayores consumos por usuarios.
- ▶ Visualizar las direcciones IP donde se realizan las peticiones.

2.2.2. Reportes asociados al funcionamiento y rendimiento del servidor proxy

Las funcionalidades con que debe contar el sistema a desarrollar, en relación con el rendimiento del servidor, están dirigidas a las estadísticas del servidor proxy, a los diferentes protocolos y métodos de conexión, así como a las direcciones IP de servidores que brindan la información (Ver Anexos 8, 9 y 10). Estos reportes desarrollan las funcionalidades siguientes:

- ▶ Mostrar un resumen general sobre el uso del servidor proxy.

- ▶ Mostrar un resumen estadístico sobre los diferentes métodos de conexiones (ejemplos: POST, GET, CONNECT).
- ▶ Visualizar la clasificación de los protocolos del resultado de las transacciones.
- ▶ Mostrar las diferentes respuestas de los servidores.
- ▶ Visualizar las direcciones IP de los servidores de origen.
- ▶ Mostrar la información de los códigos de jerarquía.

2.2.3. Opciones de interactividad

De forma adicional a las funcionalidades planteadas anteriormente, el sistema permite la exportación en formato de documento portátil (PDF, siglas del inglés *Portable Document Format*) de los reportes estadísticos visualizados, establece la configuración de la conexión a la base de datos mediante formularios de autenticación y permite la gestión de usuarios que accederán al sistema. La aplicación desarrollada permite el uso de bases de datos en las que hayan sido insertados diferentes campos (de acuerdo a las especificaciones de los administradores del sistema para la inserción de datos) de forma tal que la ausencia de un dato necesario para un reporte no implique un mal funcionamiento del sistema. La configuración de la conexión con la base de datos, en caso que se desee guardar, será almacenada en un fichero XML, lenguaje de marcado de extensión de archivos, en el cual se puede guardar la información estructurada de la manera más abstracta y reutilizable posible. En este caso la información de la conexión que se almacenará será, la dirección del servidor, el usuario y la contraseña codificada de la base de datos.

2.3. Requerimientos del sistema

La Lista de Reserva del Producto (LRP) es utilizada por la metodología de desarrollo SXP para definir y organizar el trabajo que se va a desarrollar, es donde los requerimientos del sistema pueden encontrarse de forma clara y legible. Esta lista puede modificarse o incrementarse a medida que avanza el producto y su principal objetivo es pretender que el producto al finalizar sea el más completo y seguro, apoyándose en los requisitos funcionales y no funcionales del sistema [20].

Se debe tener en cuenta que los requisitos funcionales son necesidades, funciones y opciones que debe

cumplir el sistema, de manera que éste reaccione a determinadas situaciones en particular.

Para que un producto de software tenga un buen grado de aceptación debe establecer ciertas características que se denominan requisitos no funcionales. Estas características son cualidades o propiedades con las que debe contar el producto, ya sea a través de la usabilidad, la rapidez y lo atractivo que pueda llegar a ser. A continuación se plantean los requisitos funcionales y no funcionales del sistema.

Requisitos funcionales:

RF1. Visualizar los dominios más visitados.

RF2. Visualizar las mayores descargas.

RF3. Mostrar las diferentes respuestas de los servidores (ejemplos: 200-ok, 304-Aceptado).

RF4. Visualizar los dominios que han sido visitados por un usuario.

RF5. Visualizar todos los usuarios que visitaron un dominio.

RF6. Permitir la configuración de la conexión a la base de datos mediante una ventana emergente.

RF7. Exportar reportes en formato PDF.

RF8. Mostrar un resumen general sobre el uso del servidor proxy.

RF9. Visualizar las direcciones IP donde se realizan las peticiones.

RF10. Mostrar un resumen estadístico sobre los diferentes métodos de conexiones (ejemplos: POST, GET, CONNECT).

RF11. Visualizar la clasificación de los protocolos del resultado de las transacciones (ejemplos: TCP_HIT, TCP_DENIED).

RF12. Mostrar la información de los códigos de jerarquía (NONE, DIRECT, etc).

RF13. Visualizar los mayores consumos por usuarios.

RF14. Mostrar un resumen de las conexiones realizadas a los diferentes dominios (ejemplos: *.com,

*.cu).

RF15. Visualizar las conexiones realizadas por cada tipo de contenido (ejemplos: imagen/jpg, text/html).

RF16. Visualizar las direcciones IP de los servidores de origen.

RF17. Permitir la autenticación con el sistema.

RF18. Gestionar usuario.

18.1. Adicionar usuario.

18.3. Eliminar usuario.

18.3. Cambiar contraseña de usuario.

Requisitos no funcionales:

1. **Apariencia o interfaz externa:** La interfaz del sistema tiene que ser sencilla y clara para su utilización, debe ser un diseño formal, serio e intuitivo, teniendo en cuenta el fin con el que se desarrolla la aplicación.
2. **Usabilidad:** El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora, de las aplicaciones de escritorio y de la Administración de redes. Con pocos pasos se podrá acceder a la información deseada, logrando que el usuario encuentre lo que busca en el menor tiempo posible y con la facilidad requerida.
3. **Rendimiento:** El sistema tiene que ser lo más estable y confiable posible. La disponibilidad de trabajo en red contra el servidor es constante. Se garantiza que la respuesta a solicitudes de los usuarios del sistema sea en un período de tiempo breve, en correspondencia con el volumen de datos, para evitar inconformidades con los usuarios.
4. **Soporte:** Se requiere que el producto reciba mantenimiento ante cualquier fallo que pueda ocurrir.
5. **Portabilidad:** El producto está destinado para sistemas operativos GNU/Linux, los administradores deben contar con un servidor de base de datos MongoDB.
6. **Seguridad:** El sistema está destinado para un entorno de red local que solo tendrán acceso los Administradores de redes, para acceder es necesario proporcionar los datos de la conexión al servidor MongoDB, en caso de que se deseen guardar estos datos, serán almacenados en un fichero XML, con las configuraciones de la aplicación. Además se necesita introducir las

credenciales del usuario a nivel de aplicación que accederá al sistema. Todo el acceso se realizará a través de un formulario de autenticación.

7. **Políticos-culturales:** El diseño cumple con las pautas trazadas para el conjunto de subsistemas de AiresProxy y que a su vez cumplen con las de la Universidad.
8. **Legales:** Las tecnologías escogidas para el desarrollo de la aplicación, están basadas principalmente en licencias libres.
9. **Confiabilidad:** Toda la información deberá estar protegida del acceso no autorizado, los administradores del sistema son los únicos que podrán gestionar cualquier información.
10. **Ayuda y documentación en línea:** El producto brindará a los usuarios una documentación apropiada, principalmente para los usuarios que administrarán el sistema. El principal elemento de documentación lo constituye el manual de usuario.
11. **Software:** Sistema Operativo GNU/Linux Ubuntu 11.04. Librería compilada de MongoDB (se recomienda mongod 1.6).
12. **Hardware:** Como requerimientos mínimos de hardware se sugiere 1 G de memoria RAM. Procesador familia INTEL u otro con una velocidad mínima de 1.60 GHz.

2.4. Historias de usuario

Las historias de usuario son utilizadas por la metodología SXP para especificar los requisitos funcionales del software. Constituyen las tareas y comportamientos que el sistema debe cumplir, además de especificar la respuesta que deberá dar a determinados problemas. A continuación se muestra una selección de algunas de las historias de usuario más significativas, el resto de las mismas se pueden encontrar en el Anexo 13.

Historia de Usuario	
Número: SGREAR ⁴ -01	Nombre historia: Visualizar los dominios más visitados
Modificación de Historia de Usuario Número: 1	
Usuario: Alibech Nuñez Padilla	Iteración asignada: Primera Iteración

⁴Sistema Generador de Reportes Estadísticos para Administradores de Redes (SGREAR).

Prioridad en negocio: Muy Alta	Puntos estimados: 1.5																																																																		
Riesgo en desarrollo: Bajo	Puntos reales: 1																																																																		
Descripción: La presente historia de usuario tiene como objetivo visualizar, dado un rango de fechas, los dominios más visitados organizados de forma descendente por la cantidad de conexiones. Mostrará una tabla con las columnas: dominio, conexiones (cantidad de conexiones realizadas), por ciento de conexiones, bytes (mostrará la suma total de los bytes consumidos por las visitas por dominio) y por ciento de bytes.																																																																			
Observaciones:																																																																			
Prototipo de interfaces: <table border="1"> <thead> <tr> <th colspan="6">Monitorizando del 18-06-2009 al 08-06-2012</th> </tr> <tr> <th></th> <th>Dominio</th> <th>Conexiones</th> <th>% Conexiones</th> <th>Bytes</th> <th>% Bytes</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>foro.jovenclub.cu</td> <td>21478</td> <td>23.23%</td> <td>48.93M</td> <td>9.536%</td> </tr> <tr> <td>2</td> <td>ak.fbcdn.net</td> <td>13899</td> <td>15.03%</td> <td>83.38M</td> <td>16.25%</td> </tr> <tr> <td>3</td> <td>www.facebook.com</td> <td>5462</td> <td>5.907%</td> <td>21.25M</td> <td>4.141%</td> </tr> <tr> <td>4</td> <td>google.com.cu</td> <td>3578</td> <td>3.869%</td> <td>14.37M</td> <td>2.8%</td> </tr> <tr> <td>5</td> <td>channel.facebook.com</td> <td>3448</td> <td>3.729%</td> <td>1.584M</td> <td>0.3086%</td> </tr> <tr> <td>6</td> <td>vanguardia.co.cu</td> <td>2877</td> <td>3.111%</td> <td>7.063M</td> <td>1.376%</td> </tr> <tr> <td>7</td> <td>oscars.hola.com</td> <td>1938</td> <td>2.096%</td> <td>5.71M</td> <td>1.113%</td> </tr> <tr> <td>8</td> <td>1hi.info</td> <td>1536</td> <td>1.661%</td> <td>25.53M</td> <td>4.975%</td> </tr> <tr> <td>9</td> <td>www.talksms.com</td> <td>1508</td> <td>1.631%</td> <td>10.55M</td> <td>2.056%</td> </tr> </tbody> </table>		Monitorizando del 18-06-2009 al 08-06-2012							Dominio	Conexiones	% Conexiones	Bytes	% Bytes	1	foro.jovenclub.cu	21478	23.23%	48.93M	9.536%	2	ak.fbcdn.net	13899	15.03%	83.38M	16.25%	3	www.facebook.com	5462	5.907%	21.25M	4.141%	4	google.com.cu	3578	3.869%	14.37M	2.8%	5	channel.facebook.com	3448	3.729%	1.584M	0.3086%	6	vanguardia.co.cu	2877	3.111%	7.063M	1.376%	7	oscars.hola.com	1938	2.096%	5.71M	1.113%	8	1hi.info	1536	1.661%	25.53M	4.975%	9	www.talksms.com	1508	1.631%	10.55M	2.056%
Monitorizando del 18-06-2009 al 08-06-2012																																																																			
	Dominio	Conexiones	% Conexiones	Bytes	% Bytes																																																														
1	foro.jovenclub.cu	21478	23.23%	48.93M	9.536%																																																														
2	ak.fbcdn.net	13899	15.03%	83.38M	16.25%																																																														
3	www.facebook.com	5462	5.907%	21.25M	4.141%																																																														
4	google.com.cu	3578	3.869%	14.37M	2.8%																																																														
5	channel.facebook.com	3448	3.729%	1.584M	0.3086%																																																														
6	vanguardia.co.cu	2877	3.111%	7.063M	1.376%																																																														
7	oscars.hola.com	1938	2.096%	5.71M	1.113%																																																														
8	1hi.info	1536	1.661%	25.53M	4.975%																																																														
9	www.talksms.com	1508	1.631%	10.55M	2.056%																																																														

Tabla 2.1: Historia de Usuario SGREAR-01. Visualizar los dominios más visitados

Historia de Usuario	
Número: SGREAR-02	Nombre historia: Visualizar las mayores descargas
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alibech Nuñez Padilla	Iteración asignada: Primera Iteración
Prioridad en negocio: Muy Alta	Puntos estimados: 1.5
Riesgo en desarrollo: Bajo	Puntos reales: 1.5
Descripción: El sistema mostrará, dado un rango de fechas y la cantidad de descargas que se desea mostrar, las URL con las descargas más frecuentes, a través de una tabla que contiene: URL, IP, fecha, usuario y la cantidad de bytes.	

Observaciones:					
Prototipo de interfaces:					
<i>Monitorizando del 18-06-2009 al 08-06-2012</i>					
	<i>Url</i>	<i>Ip</i>	<i>Fecha</i>	<i>Usuario</i>	<i>Bytes</i>
1	http://www.centrocuba.com/img/header.png	10.33.13.44	28-02-2011	anonymous_user2395115	432.6K
2	http://2010.encuentrolinux.cl/static/style/images/5.png	10.33.13.44	01-03-2011	anonymous_user2395115	337.7K
3	http://2010.encuentrolinux.cl/static/style/images/4.png	10.33.13.44	01-03-2011	anonymous_user2395115	333.8K
4	http://2010.encuentrolinux.cl/static/style/images/3.png	10.33.13.44	01-03-2011	anonymous_user2395115	322K
5	http://2010.encuentrolinux.cl/static/style/images/1.png	10.33.13.44	01-03-2011	anonymous_user2395115	307.7K
6	http://1hi.info/browse.php?	10.8.8.113	01-03-2011	anonymous_user1343432	303.6K
7	http://www.psicofxp.com/forums/images/contenidos/20080406/02_casi...	10.33.13.44	28-02-2011	anonymous_user2395115	253.2K
8	http://2010.encuentrolinux.cl/static/style/images/2.png	10.33.13.44	01-03-2011	anonymous_user2395115	251.2K
9	http://www.amd.com/Style%20Library/driver_08_us.xml	10.8.128.123	28-02-2011	anonymous_user1533776	221.6K

Tabla 2.2: Historia de Usuario SGREAR-02. Visualizar las mayores descargas

Historia de Usuario	
Número: SGREAR-03	Nombre historia: Mostrar las diferentes respuestas de los servidores
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alibech Nuñez Padilla	Iteración asignada: Primera Iteración
Prioridad en negocio: Muy Alta	Puntos estimados: 1.5
Riesgo en desarrollo: Bajo	Puntos reales: 1
Descripción: Dado un rango de fechas, el sistema mostrará los códigos de estados de HTTP, es decir, si fue una respuesta de acceso prohibido (403), de no disponibilidad (000), entre otras. La aplicación mostrará una tabla con las columnas: respuesta del servidor, cantidad de conexiones realizadas, el por ciento de esas conexiones, los bytes consumidos y el por ciento de esos bytes.	
Observaciones:	
Prototipo de interfaces:	

Capítulo 2: Descripción y características del sistema AiresProxy para Administradores de redes

Monitorizando del 18-06-2009 al 08-06-2012					
	Respuesta de Servidores	Conexiones	% Conexiones	Bytes	% Bytes
1	200	33010	61.9%	290.9M	96.31%
2	304	17650	33.1%	5.469M	1.811%
3	302	1113	2.087%	635.9K	0.2056%
4	204	706	1.324%	249K	0.0805%
5	404	424	0.7951%	1.166M	0.3861%
6	400	119	0.2232%	127.6K	0.04124%
7	000	115	0.2157%	0	0%
8	301	90	0.1688%	53.17K	0.01719%
9	503	45	0.08439%	76.53K	0.02474%

Tabla 2.3: Historia de Usuario SGREAR-03. Mostrar las respuestas de los servidores

Historia de Usuario	
Número: SGREAR-04	Nombre historia: Visualizar los dominios que han sido visitados por un usuario
Modificación de Historia de Usuario Número: 1	
Usuario: Isabelys Luisa Suarez Sevilla	Iteración asignada: Primera Iteración
Prioridad en negocio: Muy Alta	Puntos estimados: 1.5
Riesgo en desarrollo: Bajo	Puntos reales: 1.5
Descripción: El sistema permitirá visualizar los dominios que han sido visitados por un usuario. Dado el nombre de un usuario y un rango de fechas, el sistema devolverá en una tabla todos los dominios a los que tuvo acceso, la cantidad de conexiones, el por ciento de las conexiones, la cantidad de bytes consumidos y el por ciento de los bytes.	
Observaciones:	
Prototipo de interfaces:	

Monitorizando del 18-06-2009 al 08-06-2012						
	Dominio	Conexiones	% Conexiones	Bytes	% Bytes	Usuario
1	ak.fbcdn.net	796	12.48%	3.668M	8.457%	anonymous_user232236
2	sony.com.mx	280	4.391%	1.961M	4.523%	anonymous_user232236
3	masoportunidades.com.ar	266	4.172%	1.571M	3.623%	anonymous_user232236
4	google.com.cu	266	4.172%	1.236M	2.849%	anonymous_user232236
5	www.facebook.com	262	4.109%	978.6K	2.203%	anonymous_user232236
6	urs.microsoft.com	260	4.078%	1024K	2.305%	anonymous_user232236
7	webmail.sld.cu	244	3.827%	852.3K	1.919%	anonymous_user232236
8	s5.scribdassets.com	214	3.356%	381.4K	0.8589%	anonymous_user232236
9	www.ojolink.com	210	3.294%	624.8K	1.407%	anonymous_user232236

Tabla 2.4: Historia de Usuario SGREAR-04. Visualizar los dominios por usuarios

Historia de Usuario	
Número: SGREAR-05	Nombre historia: Visualizar todos los usuarios que visitaron un dominio
Modificación de Historia de Usuario Número: 1	
Usuario: Isabelys Luisa Suarez Sevilla	Iteración asignada: Primera Iteración
Prioridad en negocio: Muy Alta	Puntos estimados: 1.5
Riesgo en desarrollo: Bajo	Puntos reales: 1
Descripción: La aplicación permitirá visualizar todos los usuarios que visitaron un dominio. Dado el nombre de un dominio y un rango de fechas, el sistema mostrará una tabla con los usuarios que se han conectado a ese dominio, la cantidad de conexiones, el por ciento que representan las conexiones, los bytes consumidos por cada uno de ellos y el por ciento que representan los bytes.	
Observaciones:	
Prototipo de interfaces:	

Monitorizando del 18-06-2009 al 08-06-2012						
	Usuario	Conexiones	% Conexiones	Bytes	% Bytes	Dominio
1	anonymous_user232236	202	17.73%	1.308M	26.8%	www.google.com
2	anonymous_user1565745	152	13.35%	545.6K	10.92%	www.google.com
3	anonymous_user1574812	130	11.41%	478.7K	9.583%	www.google.com
4	anonymous_user1383481	78	6.848%	20.64K	0.4132%	www.google.com
5	anonymous_user2395115	74	6.497%	158.1K	3.164%	www.google.com
6	anonymous_user185248	68	5.97%	115.2K	2.307%	www.google.com
7	anonymous_user691791	66	5.795%	449.3K	8.994%	www.google.com
8	anonymous_user1131717	42	3.687%	156.7K	3.137%	www.google.com
9	anonymous_user29973	36	3.161%	287.9K	5.763%	www.google.com

Tabla 2.5: Historia de Usuario SGREAR-05. Visualizar los usuarios por dominio

Historia de Usuario	
Número: SGREAR-06	Nombre historia: Permitir la configuración de la conexión a la base de datos mediante una ventana emergente
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Isabelys Luisa Suarez Sevilla	Iteración asignada: Primera Iteración
Prioridad en negocio: Muy Alta	Puntos estimados: 1.5
Riesgo en desarrollo: Bajo	Puntos reales: 1.5
Descripción: La aplicación permitirá la configuración de la conexión a la base de datos mediante una ventana emergente. Mediante esta ventana se introducirán los datos de conexión a MongoDB (servidor, usuario y contraseña).	
Observaciones: La configuración de la conexión a la base de datos se guarda en un fichero XML, una vez autenticado el usuario en el sistema podrá cambiar los datos de la configuración, los cuales serán usados en la próxima entrada al sistema.	
Prototipo de interfaces:	



Tabla 2.6: Historia de Usuario SGREAR-06. Configuración de la conexión de acceso a datos

Historia de Usuario	
Número: SGREAR-07	Nombre historia: Exportar reportes en formato PDF
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alibech Nuñez Padilla	Iteración asignada: Segunda Iteración
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en desarrollo: Bajo	Puntos reales: 1
Descripción: La aplicación permitirá exportar en formato PDF el reporte que el usuario este consultando en ese momento.	
Observaciones:	
Prototipo de interfaces:	

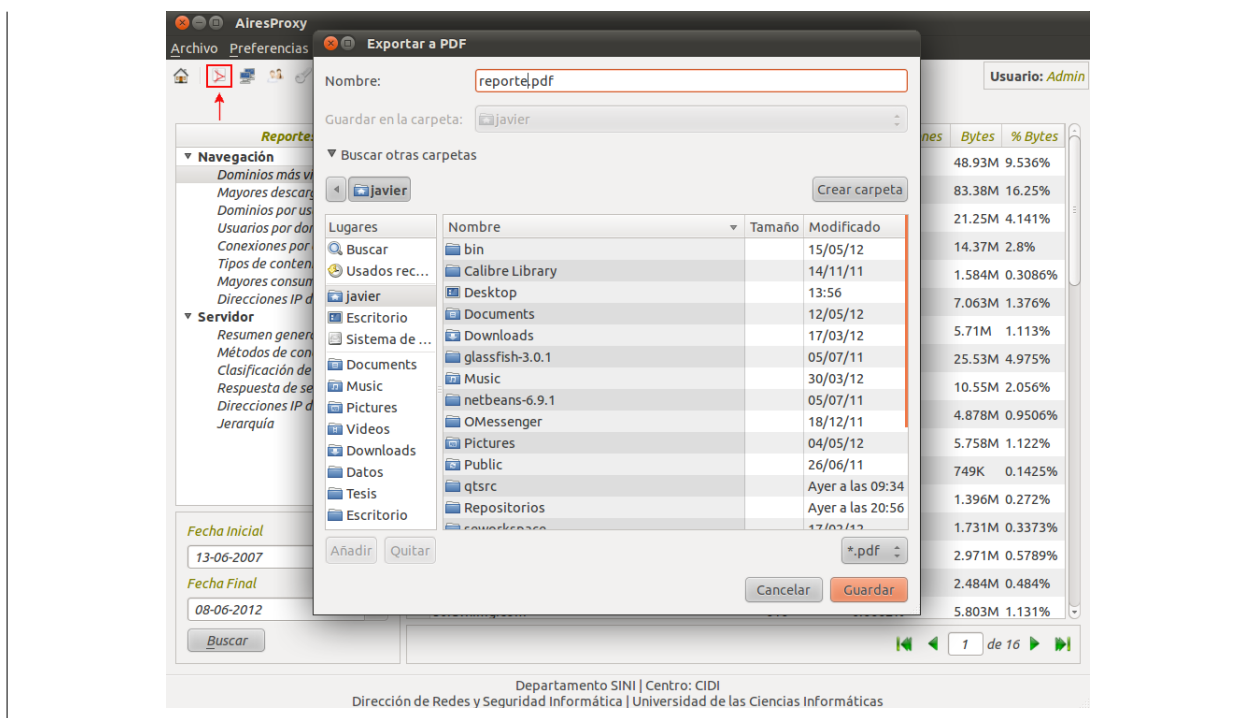


Tabla 2.7: Historia de Usuario SGREAR-07. Generar reportes en formato PDF

2.5. Descripción de la arquitectura

La arquitectura de software o arquitectura lógica es una de las técnicas metodológicas desarrolladas con el fin de facilitar la programación. Se refiere a un grupo de abstracciones y patrones que brindan un esquema de referencia útil para guiar el desarrollo de software dentro de un sistema informático. Una arquitectura de software es diseñada y seleccionada con relación a los objetivos y requerimientos necesarios, es capaz de definir, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. La arquitectura de software es la organización fundamental de un sistema, encarnada en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución [37].

Uno de los patrones de arquitectura es el Modelo Vista Controlador (MVC, siglas del inglés *Model View Controller*), un patrón de llamada y retorno de la arquitectura de software que separa los datos de una

aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

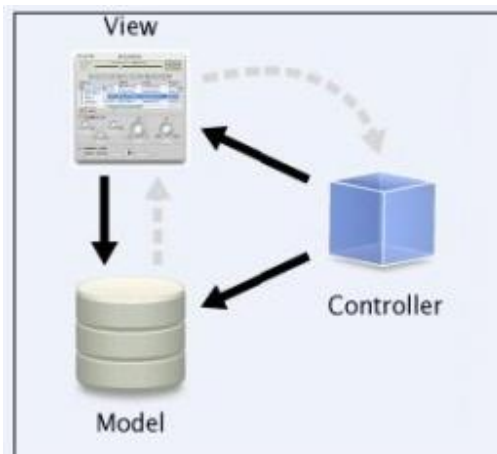


Figura 2.1: Modelo-Vista-Controlador

La finalidad del patrón es mejorar la reusabilidad por medio del desacople entre la vista y el modelo. Los elementos del patrón son los siguientes [37]:

Modelo: El modelo administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambio de estado (habitualmente desde el controlador). Es el encargado de acceder de forma directa a los datos actuando como intermediario con la base de datos.

Vista: Maneja la visualización de la información, recibe los datos del modelo y los muestra al usuario, manteniendo la información constantemente actualizada.

Controlador: Interpreta las acciones de los eventos de entrada, informando al modelo y/o a la vista para que cambien según resulte apropiado. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista.

2.5.1. Especificación de la arquitectura del sistema

La arquitectura del sistema AiresProxy para Administradores de redes se basa en una arquitectura Modelo-Vista-Controlador, separando los conceptos para que el desarrollo quede lo mejor estructurado posible, dividiendo el sistema en tres capas, modelo, vista y controlador. La idea del MVC se encuentra

asociado a la idea de 3 capas, con la diferencia de que el MVC permite la comunicación entre la capa Vista y la capa Modelo, de manera que la relación entre capas se realiza de forma triangular, mientras que la arquitectura 3 capas se basa en una distribución lineal donde cada capa proporciona servicios a la capa superior y se sirve de las prestaciones que le brinda la capa inferior. A continuación se representa la arquitectura del sistema.

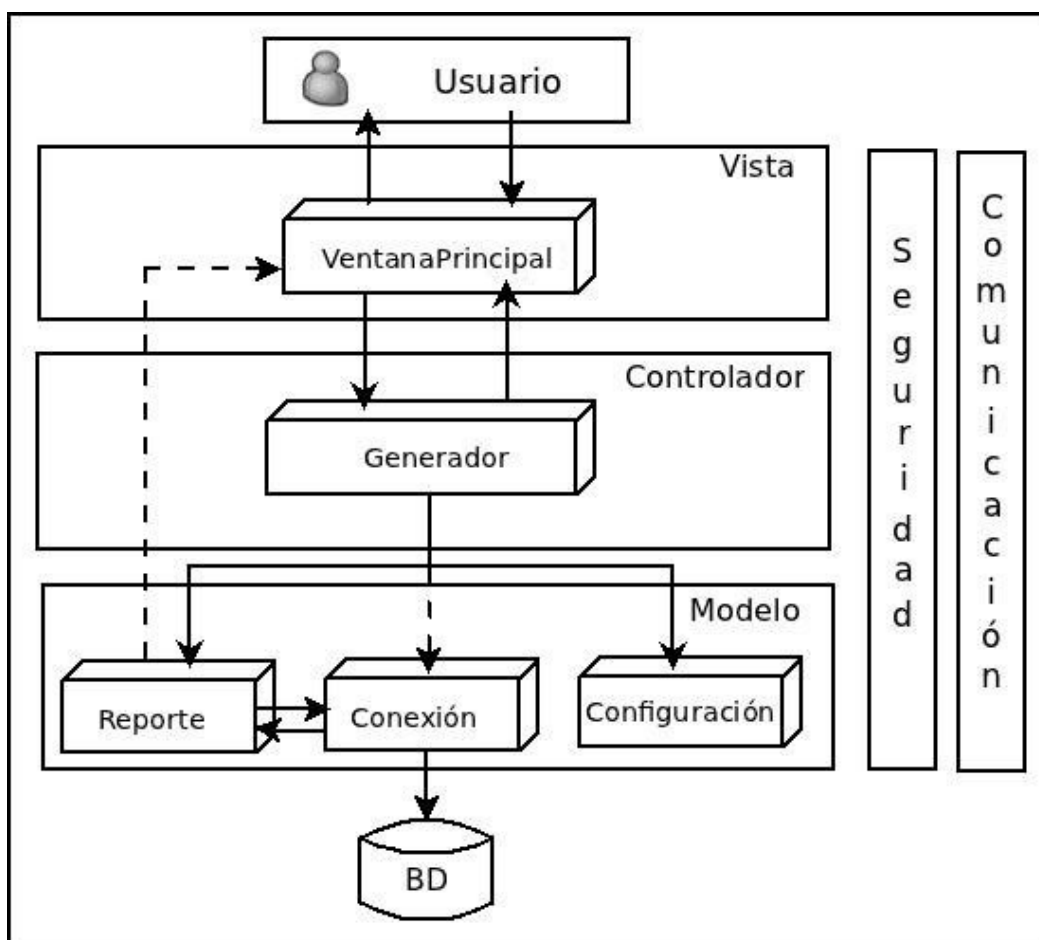


Figura 2.2: Arquitectura del AiresProxy para Administradores de redes

Vista: En la capa vista, del sistema para Administradores de redes, se localiza la VentanaPrincipal, interfaz del sistema que contiene todos los componentes visuales, encargada de interactuar con el usuario y enviar las solicitudes a la controladora (Ver Anexo 3).

Controlador: En la capa controlador se visualiza el componente Generador, que es el encargado de interpretar todos los eventos, es donde se reciben y se analizan todas las acciones de la VentanaPrincipal y es el responsable de ordenarle a la capa modelo, principalmente al componente Configuración y

Reporte, las configuraciones que se deben verificar y los algoritmos de reportes que se deben ejecutar respectivamente. Se encarga de igual manera de actualizar la conexión con la base de datos.

Modelo: En la capa modelo, se ven declarados los componentes Reporte, Conexión y Configuración los cuales son los responsables del comportamiento y de los datos del sistema. Reporte representa la abstracción de un reporte y contiene todos los algoritmos relacionados con los mismos y con el funcionamiento del sistema, es el encargado de solicitar la información a la Conexión, procesarla y enviarla a la vista. Conexión es el responsable de realizar la conexión con la base de datos y una vez conectado, de proveer toda la información requerida, respondiendo así a las solicitudes del Reporte. Configuración se encarga de leer y escribir todos los datos referentes a las configuraciones del sistema.

Seguridad: Para la seguridad del sistema es necesario introducir los datos relacionados con la conexión al servidor de base de datos MongoDB, lo cual incluye dirección de servidor, usuario y contraseña, además es necesario introducir el usuario y la contraseña con que se iniciará el sistema perteneciente a cada administrador, todo esto mediante un formulario de autenticación.

Comunicación: El sistema debe contar con una adecuada comunicación entre todos sus componentes, permitiendo facilitar su programación y lograr el buen funcionamiento del patrón seleccionado.

2.6. Patrones de diseño

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son descripciones de clases cuyas instancias colaboran entre sí. Cada patrón es adecuado para ser adaptado a un cierto tipo de problema. La clave para la reutilización es anticiparse a los nuevos requisitos y cambios, de modo que los sistemas evolucionen de forma adecuada. Cada patrón permite que algunos aspectos de la estructura del sistema puedan cambiar independientemente de otros aspectos. Facilitan la reusabilidad, extensibilidad y mantenimiento. Estos patrones pueden ser clasificados en [38]:

- ▶ Creacionales: solucionan problemas de creación de instancias, ayudan a encapsular y abstraer dicha creación.
- ▶ Estructurales: solucionan problemas de composición (agregación) de clases y objetos.
- ▶ De Comportamiento: son soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan.

2.6.1. Patrones de diseño GRASP

Los GRASP⁵ son patrones generales de software para la asignación de responsabilidades. En el diseño orientado a objetos, a pesar de utilizar metodologías ágiles como SXP, es inevitable elegir cuidadosamente las responsabilidades de cada clase, desde la primera codificación así como en la refactorización constante del programa. Los patrones que a continuación se mencionan son los utilizados en el sistema.

Experto: Es el principio básico de asignación de responsabilidades. Indica, que la responsabilidad de realizar cualquier tarea, debe recaer sobre la clase que conoce toda la información necesaria para ello.

Con el uso de este patrón se garantiza que las clases del sistema puedan cumplir con los métodos encomendados, ya que cada una de ellas tiene incluidos todos los paquetes y atributos necesarios. En el caso de la clase Configuración tiene bien definida su responsabilidad de manejar las configuraciones del sistema, ya sea leyendo o escribiendo en el fichero de configuración de tipo XML. Asimismo la clase Reporte tiene la total responsabilidad de generar los reportes y almacenar todos sus datos. Igualmente la clase Conexión es la responsable de realizar la conexión a la base de datos y de leer y escribir información en esta.

Creador: La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. El patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases.

El sistema tiene bien definida la responsabilidad relacionada con la creación de las instancias. La clase VentanaPrincipal contiene a la clase Generador expresando de esta forma su relación directa como indica el patrón MVC y por lo tanto es la responsable de instanciar a esta última. El controlador de eventos que es el Generador incluye a las clases Reporte, Conexión y Configuración por lo que es evidente que en él recae la responsabilidad de proveer los datos necesarios para la creación de los objetos de dichas clases.

Controlador: El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que recibe los datos del usuario y los envía a las distintas clases según el método llamado.

El uso de este patrón es de vital importancia en el sistema ya que es el encargado de manejar todos los

⁵Siglas del inglés *General Responsibility Assignment Software Patterns*, "Patrones generales de software para la asignación de responsabilidades".

sucesos que ocurran, la clase Generador es la que hace la función de controlador, es la que maneja los eventos ocurridos en la clase Ventana Principal y es la responsable de decidir y responder según cada evento, ya sea ordenando a la clase Reporte que ejecute un algoritmo o requerir información de la clase Configuración.

Alta cohesión: Plantea que la información que almacena una clase debe ser coherente y debe estar, en la medida de lo posible, relacionada con la clase. Al realizar un cambio en una clase con alta cohesión, todos los métodos que pueden verse afectados, estarán a la vista, en el mismo fichero. Se incrementa claridad, reutilización y facilita la comprensión del diseño.

Con el uso de este patrón se logra que en el sistema cada clase se enfoque exactamente a una responsabilidad determinada, realizando una sola labor, no desempeñada por otra clase. Por ejemplo la clase VentanaPrincipal se encarga solamente de los elementos de la interfaz, la clase Generador de los eventos, la clase Reporte de los algoritmos para los reportes estadísticos, la clase Conexión del acceso a la base de datos y la clase Configuración de gestionar las configuraciones del sistema.

Bajo acoplamiento: Es la idea de tener las clases lo menos ligadas entre sí. De tal forma que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre las clases.

El patrón MVC se basa en la separación de las características de un sistema en capas donde se intercomunican en forma triangular. AiresProxy para Administradores de redes sigue dicha filosofía, las clases Reporte, VentanaPrincipal y Generador son las principales representantes de las tres capas: modelo, vista y controlador respectivamente. Reporte representa la abstracción de los reportes que manipula la aplicación, por lo tanto puede fácilmente y sin muchas modificaciones relacionarse con otro controlador y asociarse a otras vistas. Generador es el controlador de eventos y tiene definidas las decisiones según el evento que le envíe su vista asociada, gracias a esto también cumple con el bajo acoplamiento ya que de asociarle una vista diferente sólo sería necesario ajustar los nuevos eventos.

2.6.2. Patrones de diseño GoF

Los patrones de diseño GoF⁶, publicados en el libro *"Design Patterns"* (Patrones de Diseño), son básicamente modelos que se pueden utilizar para resolver con éxito problemas que se presentan una y otra vez en la realización de un sistema.

Observer (Observador): El patrón de comportamiento observador define una dependencia de uno-

⁶Siglas del inglés *Gang of Four*, "Banda de los cuatros".

a-muchos entre objetos, de forma que cuando un objeto cambia de estado se notifica y actualizan automáticamente todos los objetos relacionados.

Se utiliza el patrón Observador para el desarrollo del sistema ya que la propia arquitectura MVC se basa en este patrón para representar las relaciones más débiles entre el modelo y la vista. Los principales participantes de dicho patrón son el sujeto y los observadores, donde un sujeto puede tener asociados varios observadores y cuando este cambia su estado se les notifica del cambio a sus objetos relacionados. En el caso de AiresProxy para Administradores de redes el modelo Reporte funciona como sujeto y de él va a depender la clase VentanaPrincipal con sus subcomponentes encargados de mostrar la tabla con el reporte actual y los datos relacionados con el paginado que actúan como observadores. Cuando se genera un reporte a partir de uno de los algoritmos, se notifica a VentanaPrincipal y se actualizan automáticamente la tabla de reporte actual y el paginado según el nuevo estado del reporte.

2.7. Diseño del sistema

El diseño del sistema debe proporcionar una idea clara de lo que es el software, enfocándose en el funcionamiento desde el punto de vista de la implementación. Para la realización del diseño y facilitar el entendimiento de la aplicación en cuestión, se tuvo en cuenta la plantilla modelo de diseño, documento generado por la metodología SXP.

2.7.1. Modelo de dominio

Debido a que el negocio no se encuentra bien definido, se ve la necesidad de describir el funcionamiento de la herramienta mediante conceptos y relaciones, mostrándolos en un modelo de dominio. El modelo de dominio se centra en una parte del negocio, la relacionada con el ámbito del sistema, donde los procesos del negocio no están claramente definidos.

Se ha decidido realizar esta modelación ya que es una alternativa apropiada dada la situación del problema, permitiendo comprender el contexto exacto de la aplicación. A continuación se plantean los conceptos y eventos que se determinaron:

Conceptos:

- ▶ **Usuario:** Se le denomina a los Administradores de redes que son los encargados de trabajar con la aplicación.
- ▶ **Reporte:** Se le denomina a los datos estadísticos generados por el sistema, que se encuentran

agrupados de una manera estructurada.

- ▶ **Exportar Reporte:** Se le denomina a los reportes que son exportados en formato PDF.
- ▶ **Sistema:** Es la aplicación encargada de generar los reportes.
- ▶ **Configuración:** Se le denomina a las configuraciones con las que debe contar el sistema.
- ▶ **Servidor:** Es el encargado de suministrarle los datos al sistema para la generación de los reportes.

Eventos:

- ▶ **Selecciona:** El usuario puede seleccionar varios reportes y varias exportaciones de reportes.
- ▶ **Tiene:** Cada reporte puede ser exportado a formato PDF.
- ▶ **Composición:** El sistema está compuesto por varios reportes, diversas configuraciones y un servidor de base de datos (MongoDB).

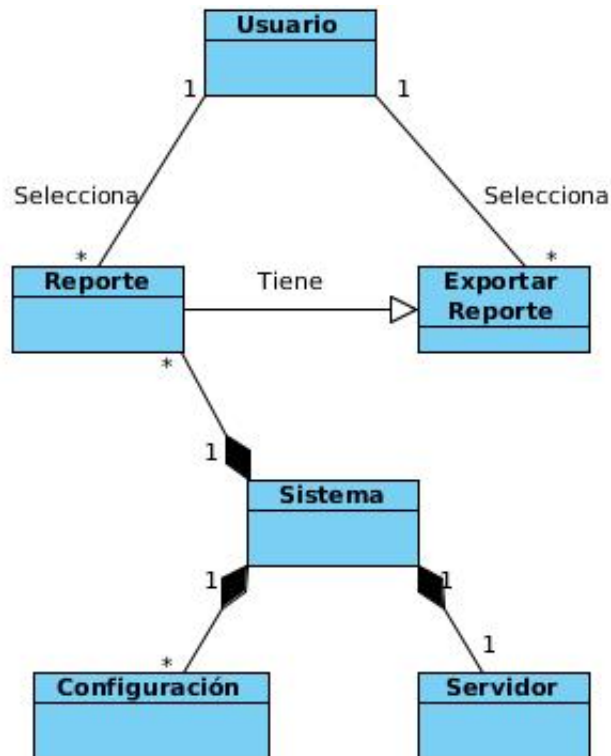


Figura 2.3: Modelo de dominio

2.7.2. Diagrama de clases del diseño

Uno de los diagramas principales para el análisis y diseño de una aplicación es el diagrama de clases, el cual representa las clases del sistema mostrando las relaciones existentes entre ellas. Estas clases contienen atributos, operaciones y relaciones, estas últimas pueden ser de herencia, composición, agregación, asociación y dependencia. En el diagrama que se representa a continuación se muestran las clases del sistema, con su estructura y comportamiento.

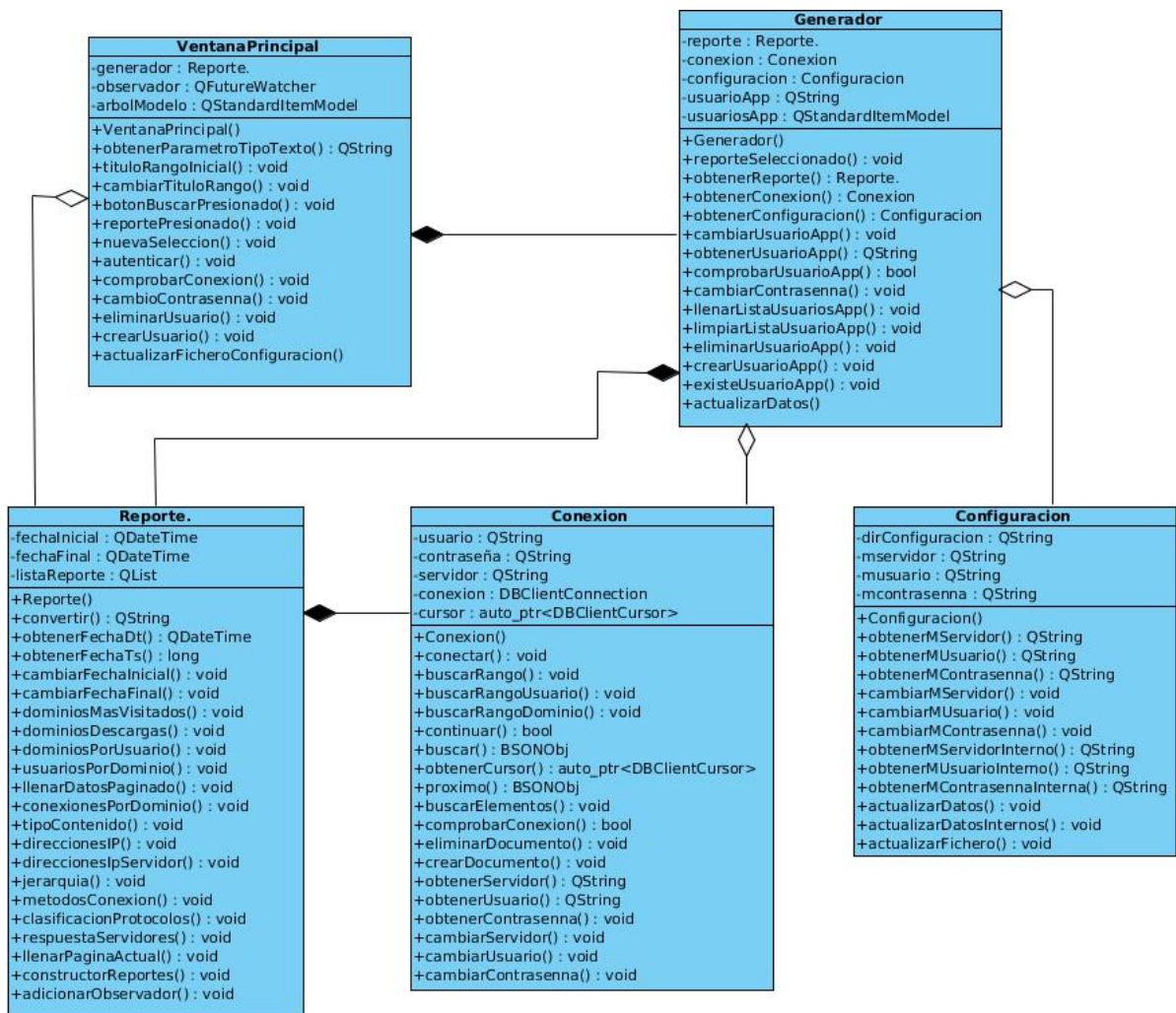


Figura 2.4: Diagrama de clases del diseño

La **VentanaPrincipal** es la clase encargada de mostrar los componentes visuales de la aplicación, así como obtener los diferentes parámetros que son seleccionados. Permite levantar ventanas de dialogo para introducir datos, establecer textos, modificar mensajes y notificar que reporte ha sido seleccionado.

Mantiene una relación de composición con la clase **Generador** y de agregación con la clase **Reporte**.

La clase **Generador** es la encargada, una vez que la **VentanaPrincipal** le transmita la señal del reporte seleccionado, de notificarle el cambio de la selección a la clase **Reporte**. Mantiene una relación de composición con la clase **Reporte** y de agregación con **Conexión** y **Configuración**.

La clase **Reporte** es la responsable de las funcionalidades relacionadas con los reportes estadísticos de la navegación de los usuarios en la red. Es la encargada de ordenarle a la clase **Conexión** que se conecte a la base de datos, llenando, a través de los algoritmos implementados, la lista de datos de los reportes que serán mostrados seguidamente en la **VentanaPrincipal**. Mantiene una relación de composición con la clase **Conexión**.

Conexión es la clase encargada de conectarse y de realizar las consultas a la base de datos.

La clase **Configuración** se encarga de las configuraciones de todo el sistema, principalmente las relacionadas con la conexión a la base de datos, almacenando los datos en un fichero XML.

Para obtener más detalles de las clases, con sus métodos y atributos ver los Anexos del 14 al 18.

2.7.3. Diagrama de paquetes del sistema

El objetivo principal del diagrama de paquetes es obtener una visión más clara del sistema, organizándolo en subsistemas, agrupando los elementos del análisis, diseño o construcción y detallando las relaciones de dependencia entre ellos [11]. El diagrama de paquetes muestra como un sistema está dividido en partes lógicas, las cuales muestran las dependencias entre esas partes. Los paquetes pueden tener a su vez otros paquetes anidados.

A continuación se muestra el diagrama de paquetes, donde el criterio de empaquetamiento fue a nivel de aplicación, representando el sistema generador de reportes estadísticos para Administradores de redes en un solo paquete denominado **AiresProxy**, el mantiene una dependencia con el paquete **AiresCore**, aplicación encargada de procesar los logs generados por el servidor proxy **Squid**.

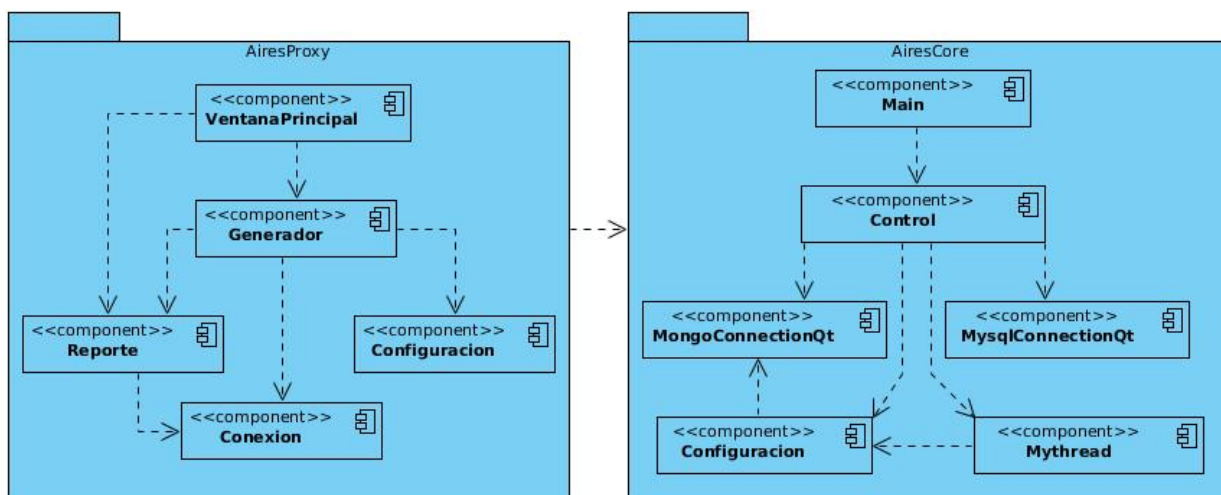


Figura 2.5: Diagrama de paquetes del AiresProxy para Administradores de redes

Este diagrama representa la relación entre dos paquetes el AiresProxy y el AiresCore.

El **paquete AiresProxy** representa el sistema para Administradores de redes, contiene los componentes del sistema VentanaPrincipal, Generador, Reporte, Conexión y Configuración, con sus respectivas relaciones, donde VentanaPrincipal representa la interfaz de la aplicación, el Generador es el componente donde se interpretan todas las acciones del sistema, Reporte es donde se encuentran todas las funcionalidades de los reportes a visualizar, Conexión es el componente de acceso a datos y Configuración es el responsable de la correcta configuración del sistema. El paquete AiresProxy para Administradores de redes mantiene una relación de dependencia con el paquete AiresCore, lo cual indica que utiliza sus servicios.

El **paquete AiresCore** es el encargado de leer los archivos logs generados por el servidor proxy Squid, permitiendo crear la estructura de la base de datos, de manera que el sistema AiresProxy para Administradores de redes pueda leer e interpretar los datos de forma apropiada. El paquete AiresCore contiene igualmente una representación de todos sus componentes, donde Main se encarga de la ejecución del programa y de la inserción de los datos en MongoDB, Control maneja las conexiones a las base de datos y la configuración de todo el sistema, MysqlConnectionQt se encarga de la conexión a una base de datos MySQL, MongoConnectionQt se encarga de la conexión a una base de datos MongoDB, Configuración contiene toda la configuración de la aplicación y Mythread es el componente que ejecuta el programa en hilos.

2.8. Diseño de la Base de Datos

Para el diseño de la base de datos se tuvo en cuenta una serie de consideraciones que son de vital importancia, como por ejemplo, la velocidad de acceso, los tipos de datos, así como la facilidad con que se debe acceder y extraer la información requerida.

Los sistemas de bases de datos no relacionales han crecido en los últimos años debido a las grandes compañías de Internet que necesitan procesar grandes volúmenes de datos. Es por eso que surge el MongoDB la cual permite optimizar y mejorar el rendimiento del almacenamiento de los mismos. El mecanismo para guardar la información en MongoDB es mucho más simple ya que no hay que realizar relaciones ni varias actualizaciones por cada una de las tablas involucradas. Una vez que se entiende su funcionamiento, adaptarse a este tipo de base de datos resulta sencillo.

2.8.1. Diagrama de la Base de Datos

Como bien se planteó anteriormente MongoDB no es una base de datos relacional, ya que es basada en documentos, almacena la información como clave-valor en forma de documento dentro de una colección. Para un mejor entendimiento de este sistema con los relacionales se puede encontrar una similitud entre clave-valor como campo-valor, un documento como una fila y una colección como una tabla. En este caso los documentos pueden tener su propia estructura sin necesidad de ser común entre todos los documentos como es el caso de las filas en una base de datos relacional. A continuación se representa el diagrama de la base de datos.

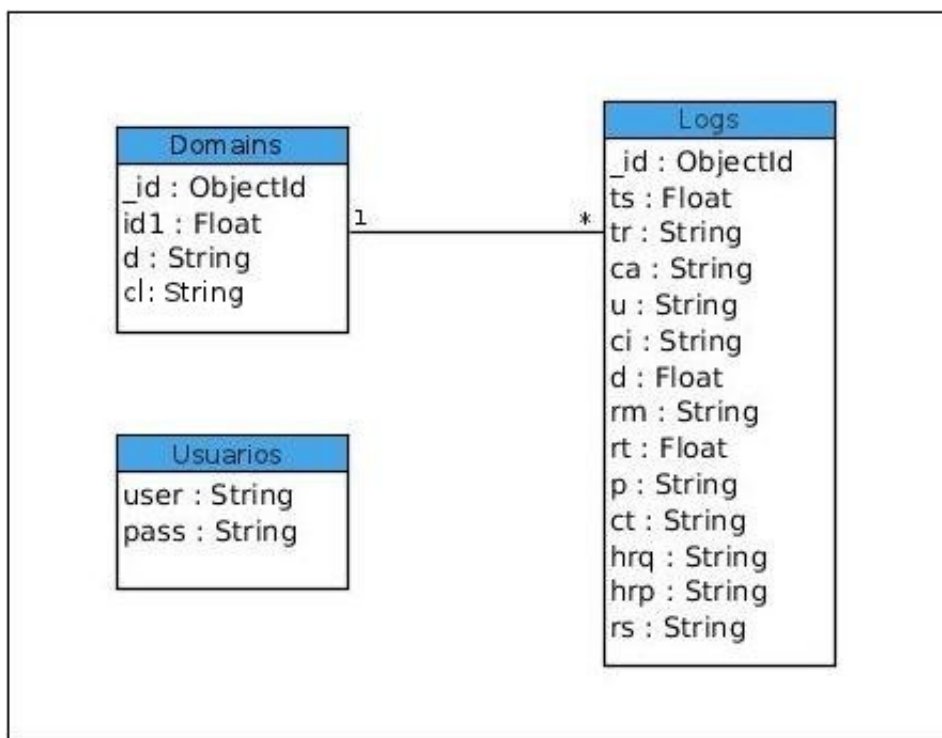


Figura 2.6: Base de datos del AiresProxy para Administradores de redes

En la colección **Domains** (Dominios) es donde se almacenan todos los dominios que se encuentran en los logs y la categoría que representan cada uno de ellos, es decir, si son dominios de interés, irrelevantes, de ocio, nacionales o bibliotecas, además la colección Domains contiene un identificador propio para cada dominio logrando así una relación con la colección Logs. Un dominio puede referenciar a varios Logs.

En la colección **Logs** es donde se guardan todos los datos generados por los logs del servidor proxy Squid, las direcciones IP de peticiones y de servidores, los usuarios, los distintos tipos de métodos y protocolos, los tipos de contenidos de las respuestas a las solicitudes, entre otras informaciones de interés que permiten la generación de reportes estadísticos.

En la colección **Usuarios** es donde se registran los datos con los que se iniciará el sistema, guardando para ello, el usuario y la contraseña codificada, perteneciente a cada Administrador de red.

Para obtener una mayor información sobre los atributos de la base de datos Ver Anexos del 19 al 21.

2.9. Conclusiones

En el desarrollo de este capítulo se definieron los aspectos que constituyen la base para el desarrollo del sistema AiresProxy para Administradores de redes. Se identificaron las características y responsabilidades con las que debe cumplir el sistema propuesto, lo que permitió la definición de los requerimientos de software, así como la descripción de cada uno de los escenarios a desarrollar. Se ha obtenido una visión sobre lo que debe hacer el sistema, posibilitando conformar una propuesta de solución adecuada a las necesidades existentes. Una vez finalizado este capítulo se da paso a la implementación de la aplicación propuesta.

Capítulo 3

Implementación del sistema generador de reportes estadísticos

3.1. Introducción

EL principal objetivo de este capítulo es realizar la implementación del sistema generador de reportes estadísticos sobre la navegación de los usuarios por Internet, AiresProxy para Administradores de redes. A partir de todos los análisis realizados se pretende implementar las clases y funcionalidades del sistema. Mediante el diagrama de componentes y el diagrama de despliegue se representa la vista estática del sistema y su distribución física respectivamente.

3.2. Diagrama de componentes

Un diagrama de componentes representa, en términos de implementación, cómo un sistema de software es dividido en componentes, mostrando a su vez las relaciones entre ellos. Los diagramas de componentes son otra forma de mostrar una vista estática del sistema, donde se representan las dependencias entre los componentes físicos, sean estos, código fuente, archivos, bibliotecas o paquetes.

A continuación se muestra la organización y las relaciones de los componentes del sistema usando para ello el diagrama de componentes.

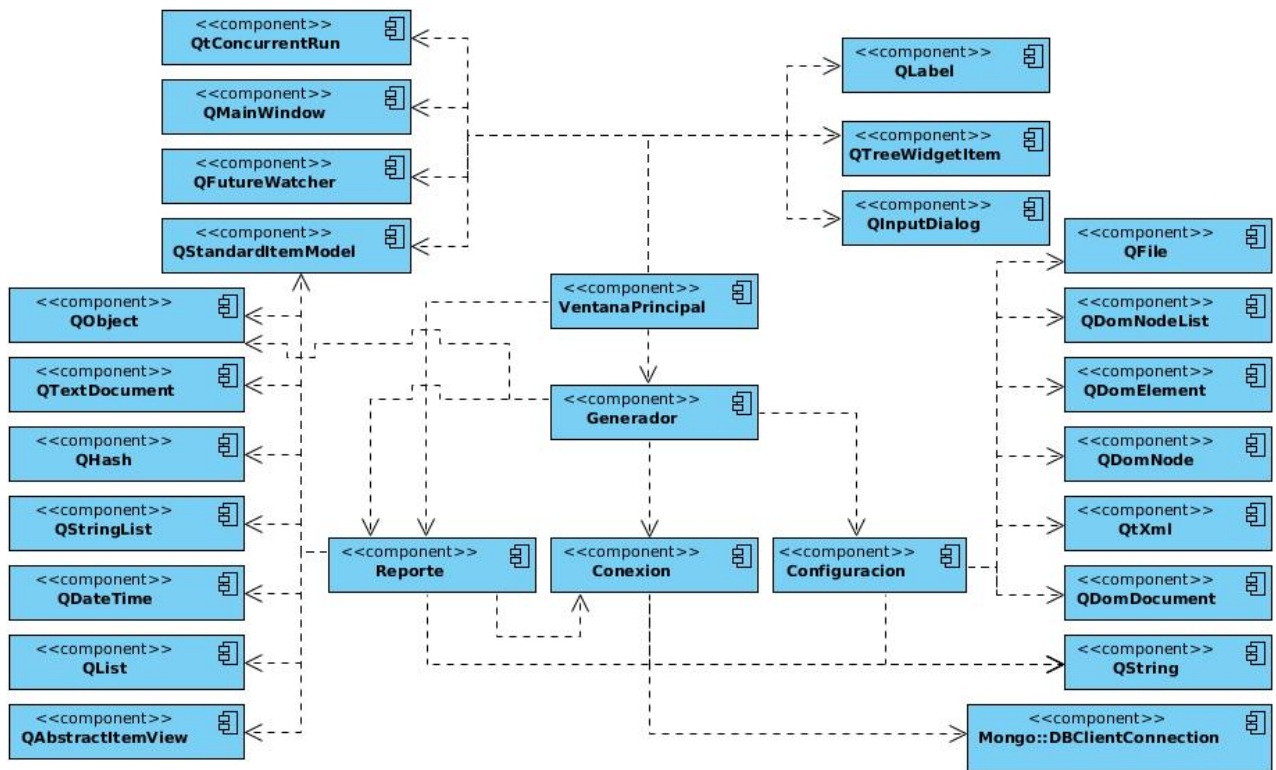


Figura 3.1: Diagrama de componentes

Componente VentanaPrincipal

El componente VentanaPrincipal es el encargado de la interfaz del sistema, mantiene dependencias de los componentes Generador y Reporte, además contiene una serie de componentes que son de vital importancia para su funcionamiento, entre ellos se encuentran:

Componente	Descripción
QTreeWidgetItem	La clase QTreeWidgetItem proporciona elementos para su uso con la clase QTreeWidgetItem, donde esta última ofrece un modelo de árbol predefinido. Estos elementos del árbol de widgets se utilizan para mantener filas de información, que por lo general contienen varias columnas de datos, donde cada una puede contener una etiqueta de texto y un icono.
QtConcurrentRun	El QtConcurrentRun permite la Ejecución Asíncrona, proporciona una manera de ejecutar una función en un hilo separado.

QStandardItemModel	La clase QStandardItemModel ofrece un modelo genérico para el almacenamiento de datos personalizados.
QMainWindow	La clase QMainWindow proporciona una ventana principal de la aplicación.
QLabel	QLabel ofrece una pantalla de texto o imagen.
QFutureWatcher	La clase QFutureWatcher permite la supervisión de un QFuture, usando señales y puertos que permiten monitorizar el comportamiento de QFuture, donde esta clase representa el resultado de un cálculo asíncronico.
QInputDialog	La clase QInputDialog proporciona un diálogo para obtener un único valor. El valor de entrada puede ser una cadena, un número o un elemento de una lista.

Tabla 3.1: Descripción de componentes incorporados en la VentanaPrincipal

Componente Generador

El componente Generador es el encargado de los eventos del sistema, recibe las notificaciones de la interfaz y según los eventos ocurridos le ordena una determinada tarea a los componentes asociados. Depende de los componentes Reporte, Conexión y Configuración. Para su correcto funcionamiento contiene además una agregación del componente QObject:

Componente	Descripción
QObject	La clase QObject es una de las clases base de todos los objetos de Qt. La característica central de este modelo es un mecanismo muy poderoso para la comunicación de objetos, denominado señales y puertos.

Tabla 3.2: Descripción de componentes incorporados en el Generador

Componente Reporte

El componente Reporte contiene todos los algoritmos del sistema que permiten la generación de los reportes a visualizar en la VentanaPrincipal. Mantiene una relación de composición con el componente

Conexión y de agregación con una serie de componentes que se pueden ver a continuación:

Componente	Descripción
QTextDocument	La clase QTextDocument es un contenedor de documentos estructurados de texto enriquecido, ofreciendo soporte para texto con estilo y diversos tipos de elementos de documentos, tales como listas, tablas, marcos e imágenes.
QString	La clase QString proporciona una cadena de caracteres Unicode ⁷ .
QHash	La clase QHash es una de las clases de contenedor genérico de Qt. Almacena el valor de la clave, en pares y ofrece una rápida búsqueda del valor asociado a una clave.
QDateTime	La clase QDateTime proporciona funciones de fecha y hora permitiendo compararlas y manipularlas.
QStringList	La clase QStringList ofrece una lista de cadenas.
QList	La clase QList es una clase de plantilla que proporciona una lista. Es una de las clases de contenedor genérico de Qt. Se almacena una lista de valores.
QAbstractItemView	La clase QAbstractItemView es la clase base para todas las vistas estándar que utiliza un QAbstractItemModel. QAbstractItemView es una clase abstracta y no puede ser instanciada. Se proporciona una interfaz estándar para la inter-operabilidad con los modelos a través de señales.

Tabla 3.3: Descripción de componentes incorporados en el Reporte

Componente Conexión

Es el encargado de conectarse a la base de datos, realizar las consultas y enviar las notificaciones de los datos al componente Reporte. Contiene una relación de agregación con el componente QString y con el componente Mongo::DBClientConnection, donde este último contiene todas las funcionalidades que

⁷Es un estándar de codificación de caracteres diseñado para facilitar el tratamiento informático, transmisión y visualización de textos.

hacen posible el trabajo con la base de datos MongoDB.

Componente Configuración

Es el componente responsable de todas las configuraciones, particularmente de la conexión con la base de datos. Presenta una relación de agregación con varios componentes entre ellos se encuentran:

Componente	Descripción
QFile	La clase QFile es un dispositivo para leer y escribir archivos de texto.
QtXml	El módulo de QtXml proporciona un lector y escritor de flujo de documentos XML y C++.
QDomDocument	La clase QDomDocument representa todo el documento XML. Conceptualmente, es la raíz del árbol del documento y proporciona el acceso primario a los datos del documento.
QDomNode	La clase QDomNode es la clase base para todos los nodos de un árbol DOM.
QDomElement	La clase QDomElement representa un elemento en el árbol DOM.
QDomNodeList	La clase QDomNodeList es una lista de objetos QDomNode.

Tabla 3.4: Descripción de componentes incorporados en Configuración

3.3. Diagrama de despliegue

El diagrama de despliegue permite describir la distribución física de un sistema, es utilizado para modelar los componentes de hardware sobre el que se ejecuta el software. En este tipo de diagrama intervienen una serie de nodos, componentes y asociaciones; donde cada nodo representa un recurso de cómputo, es decir, una máquina que se compone habitualmente de memoria, capacidad de procesamiento y de almacenamiento.

A continuación se muestra el diagrama de despliegue del sistema, el cual se encuentra dividido en tres escenarios diferentes:

Escenario 1:

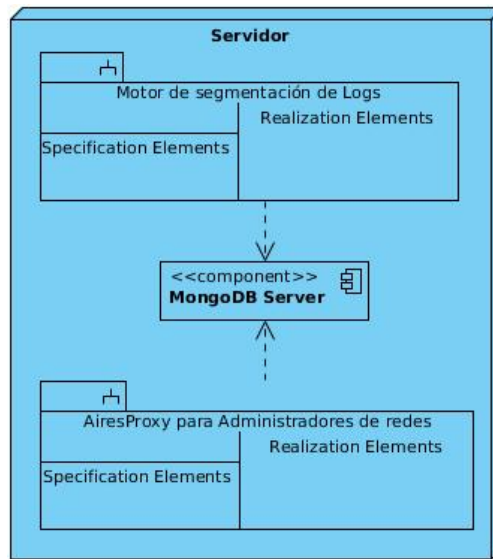


Figura 3.2: Escenario 1 del diagrama de despliegue

En el primer escenario se ve reflejado como en una misma computadora se pueden encontrar todos los paquetes y componentes necesarios para que funcione el sistema, incluyendo el propio sistema. El paquete Motor de segmentación de logs, es el subsistema encargado de procesar los archivos logs del servidor proxy y almacenarlo en la base de datos MongoDB, la cual será utilizada posteriormente por el subsistema AiresProxy para Administradores de redes.

Escenario 2:

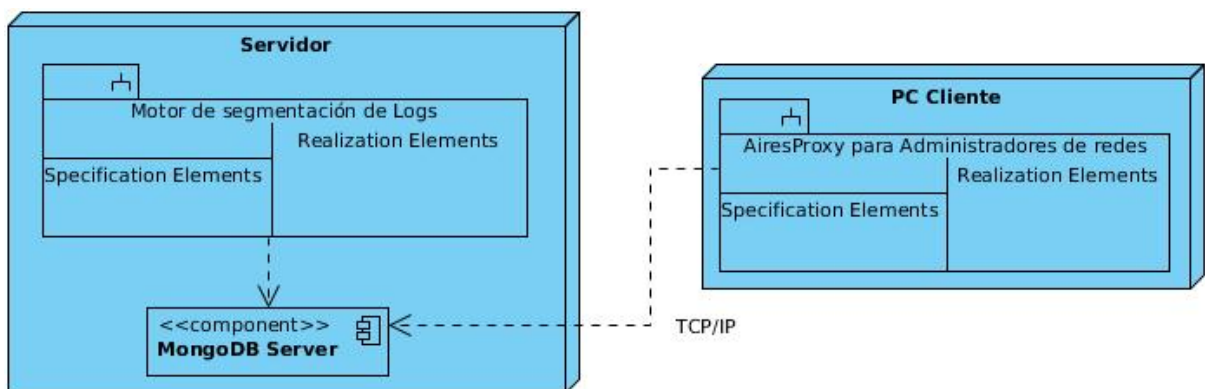


Figura 3.3: Escenario 2 del diagrama de despliegue

En este segundo escenario se ve reflejado en una PC Cliente el sistema AiresProxy para Administradores

de redes y en un Servidor, el Motor de segmentación de logs, con el gestor de base de datos MongoDB. La comunicación entre la PC Cliente y la base de datos se realiza mediante la familia de protocolos TCP/IP.

Escenario 3:

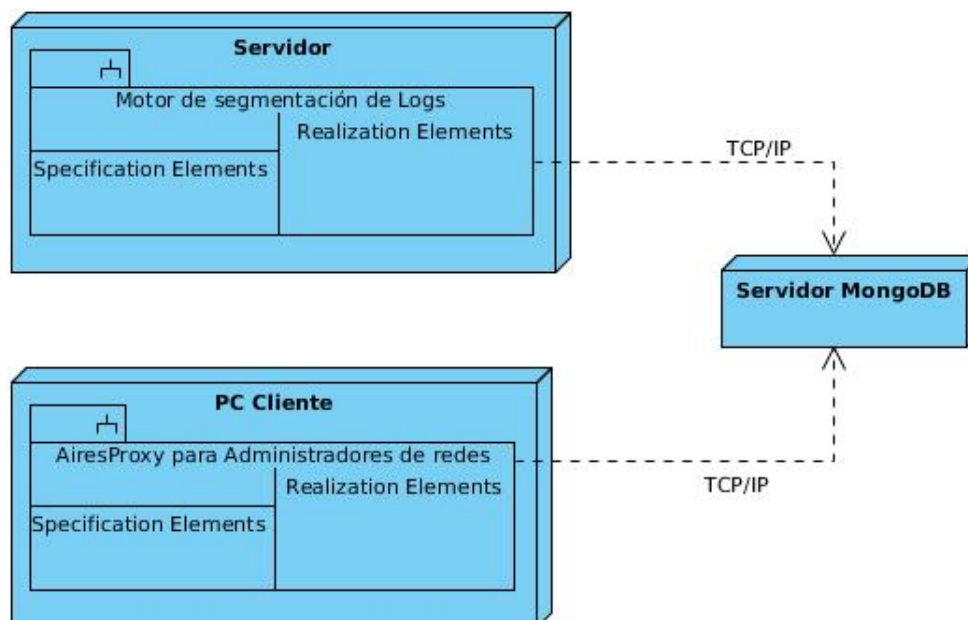


Figura 3.4: Escenario 3 del diagrama de despliegue

Como tercer escenario se muestran a los subsistemas Motor de segmentación de logs y AiresProxy para Administradores de redes en máquinas independientes, al igual que el servidor de base de datos MongoDB, donde la comunicación hacia el gestor de base de datos se realiza mediante la familia de protocolos TCP/IP.

3.4. Conclusiones

Luego de realizar la implementación del sistema, se ha obtenido un producto con una estructura basada en la generación de reportes estadísticos de la navegación por Internet, proporcionándole a los Administradores de redes elementos importantes de control. A partir del diagrama de componentes se identifica con claridad la estructura y las relaciones que existen entre los diferentes componentes empleados. Además queda reflejada la distribución física con que contará el sistema una vez que se despliegue el mismo.

Capítulo 4

Validación de la solución propuesta

4.1. Introducción

EN todo desarrollo de software las etapas de pruebas son de gran importancia, la metodología SXP requiere de una etapa de pruebas en cada iteración de desarrollo del producto. En este capítulo se representan los casos de pruebas a la que fue sometida la aplicación, con el objetivo de verificar el cumplimiento de las metas trazadas.

4.2. Pruebas de funcionalidad

Para determinar la calidad de un producto de software se realiza el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado de cumplimiento de los requerimientos del sistema.

Las pruebas funcionales son las pruebas de software que tienen como objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados. A este tipo de prueba también se les denomina pruebas de caja negra, ya que se enfocan en el análisis de los datos de entrada y en los de salida, esto se define en los casos de prueba que se elaboran antes de realizar las pruebas.

Casos de pruebas de funcionalidad

Los casos de prueba determinan si un requisito del sistema fue realizado satisfactoriamente. Con el objetivo de comprobar que todos los requisitos de una aplicación son revisados, debe haber al menos un caso de prueba por cada requisito funcional del sistema y de esta forma saber si está apto para ser liberado.

Durante el desarrollo del sistema se diseñó un conjunto de casos de prueba para comprobar su funcionamiento de acuerdo a las historias de usuario. A continuación se muestran, de las pruebas realizadas, las más significativas, en el Anexo 22 se describen el resto de las pruebas desarrolladas.

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-01	Nombre Historia de Usuario: Visualizar los dominios más visitados
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: El sistema visualiza los dominios más visitados dado un rango de fechas.	
Condiciones de Ejecución: <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Dominios más visitados”. 	
Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Navegación, posteriormente seleccionar Dominios más visitados. Introducir la fecha inicial y la fecha final, oprimir el botón Buscar.	
Resultado Esperado: El sistema muestra una tabla con los dominios que más han sido visitados, la cantidad de conexiones por cada dominio, el por ciento de conexiones, la cantidad de bytes consumidos y el por ciento de bytes. La tabla se encuentra ordenada descendentemente por la cantidad de conexiones.	
Evaluación de la Prueba: Satisfactorio.	

Tabla 4.1: Caso de prueba de aceptación SGREAR-01

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-02	Nombre Historia de Usuario: Visualizar las mayores descargas.
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: El sistema visualiza las URL donde las descargas han sido mayores.	

<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Mayores descargas”.
<p>Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Navegación, posteriormente seleccionar Mayores descargas. Introducir la fecha inicial y la fecha final, oprimir el botón Buscar, introducir la cantidad de elementos que se desean visualizar y pulsar el botón Aceptar.</p>
<p>Resultado Esperado: El sistema muestra una tabla con las URL, los IP, las fechas, los usuarios y la cantidad de bytes descargados. La tabla se encuentra ordenada descendientemente por la cantidad bytes.</p>
<p>Evaluación de la Prueba: Satisfactorio.</p>

Tabla 4.2: Caso de prueba de aceptación SGREAR-02

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-03	Nombre Historia de Usuario: Mostrar las diferentes respuestas de los servidores
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: Dado un rango de fechas el sistema visualiza las diferentes respuestas de los servidores (ejemplos: 200-ok, 304-Aceptado).	

<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Respuesta de servidores”.
<p>Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Servidor, posteriormente seleccionar Respuesta de servidores. Introducir la fecha inicial y la fecha final, oprimir el botón Buscar.</p>
<p>Resultado Esperado: El sistema muestra una tabla con las respuestas de los servidores, cantidad de conexiones, el por ciento de las mismas, los bytes consumidos por cada respuesta y el por ciento que representan. La tabla se encuentra ordenada descendentemente por la cantidad de conexiones.</p>
<p>Evaluación de la Prueba: Satisfactorio.</p>

Tabla 4.3: Caso de prueba de aceptación SGREAR-03

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-04	Nombre Historia de Usuario: Visualizar los dominios que han sido visitados por un usuario
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: El sistema visualiza los dominios visitados por un usuario determinado, según el rango de tiempo seleccionado.	

<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Dominios por usuario”.
<p>Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Navegación, introducir el rango de fechas, presionar el botón Buscar, insertar el usuario a verificar y presionar el botón Aceptar.</p>
<p>Resultado Esperado: El sistema muestra una tabla con los dominios visitados, la cantidad de conexiones por cada dominio, el por ciento de las conexiones, la cantidad de bytes consumidos y el por ciento que representan esos bytes. La tabla se encuentra ordenada descendientemente por la cantidad de conexiones.</p>
<p>Evaluación de la Prueba: Satisfactorio.</p>

Tabla 4.4: Caso de prueba de aceptación SGREAR-04

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-05	Nombre Historia de Usuario: Visualizar todos los usuarios que visitaron un dominio
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: Dado un rango de fechas el sistema visualiza los usuarios que han visitado un dominio dado.	

<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Usuarios por dominio”.
<p>Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Navegación, introducir el rango de fechas, presionar el botón Buscar, insertar el dominio a verificar y presionar el botón Aceptar.</p>
<p>Resultado Esperado: El sistema muestra una tabla con los usuarios que se han conectado al dominio dado, muestra además la cantidad de conexiones por cada usuario, el por ciento de esas conexiones, la cantidad de bytes consumidos y el por ciento que representan. La tabla se encuentra ordenada descendientemente por la cantidad de conexiones.</p>
<p>Evaluación de la Prueba: Satisfactorio.</p>

Tabla 4.5: Caso de prueba de aceptación SGREAR-05

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-06	<p>Nombre de Usuario: Permitir la configuración de la conexión a la base de datos mediante una ventana emergente</p> <p>Historia:</p>
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: Se realiza la correcta configuración de la conexión de la base de datos.	

<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema.
<p>Entrada / Pasos de ejecución: El usuario comprueba la conexión con la base de datos, se autentica, pulsar el menú Preferencias, Configuración, presionar la etiqueta Datos de Conexión, introducir los datos de la conexión, presionar el botón Guardar; o presionar directamente el icono Configuración, presionar la etiqueta Datos de Conexión, introducir los datos de conexión, presionar el botón Guardar.</p>
<p>Resultado Esperado: El sistema almacena los datos en el fichero de configuración XML.</p>
<p>Evaluación de la Prueba: Satisfactorio.</p>

Tabla 4.6: Caso de prueba de aceptación SGREAR-06

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-07	Nombre Historia de Usuario: Generar reportes en formato PDF
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: Se exportan los reportes en formato PDF.	
<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Ejecutar un reporte estadístico. 	

Entrada / Pasos de ejecución: El usuario comprueba la conexión con la base de datos, se autentica, ejecuta un reporte y pulsa el icono Exportar a PDF; o selecciona el menú Archivo, Exportar a PDF.
Resultado Esperado: El sistema genera el reporte en formato PDF.
Evaluación de la Prueba: Satisfactorio.

Tabla 4.7: Caso de prueba de aceptación SGREAR-07

4.2.1. Resultado de las pruebas

Las pruebas funcionales se realizaron en 3 iteraciones a medida que avanzaba el sistema, las cuales guiaron la calidad del sistema, y determinaron en cada momento si se estaba o no en condiciones de continuar avanzando con el ciclo de desarrollo. A continuación se muestran los resultados de las pruebas realizadas en cada iteración, las cuales arrojaron un total de 31 no conformidades que fueron corregidas satisfactoriamente.

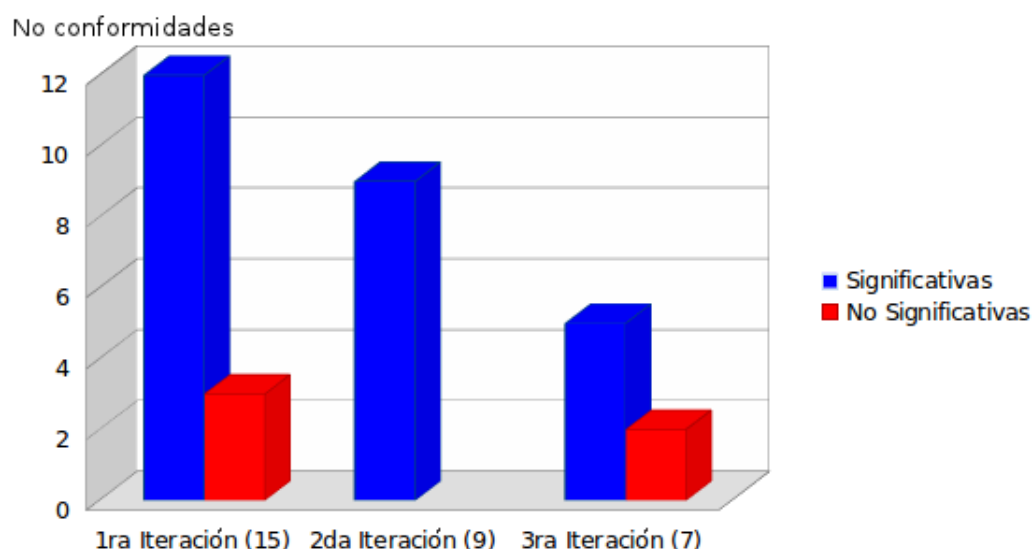


Figura 4.1: No conformidades detectadas en cada iteración

En la 1era Iteración se recogieron un total de 15 no conformidades, en la 2da Iteración 9 no conformidades y en la 3ra Iteración 7 no conformidades. Muchas de las no conformidades significativas se relacionaban con errores ortográficos, validaciones, excepciones, algunos reportes que no se encontraban en

correspondencia con el documento y de algunas funciones que no se desempeñaban correctamente; mientras que las no conformidades no significativas se vinculaban con el diseño o con características poco adecuadas en el sistema.

Algunas de las no conformidades detectadas en el sistema fueron:

No Conformidades	Iteración
El sistema no brinda la posibilidad de introducir la cantidad de elementos a mostrar por página.	1ra Iteración
No se realiza el paginado a los reportes.	1ra Iteración
No se mantiene el reporte en memoria una vez que se genera en el cambio de selección.	1ra Iteración
No se muestran los por cientos en el reporte “Dominios más visitados”.	1ra Iteración
No se ordenan los elementos de un reporte.	1ra Iteración
No se muestra un indicador que informe al usuario que se esta generando un reporte.	2da Iteración
La funcionalidad exportar PDF no verifica si existe un reporte seleccionado.	2da Iteración
La funcionalidad exportar PDF no muestra un indicador sobre la exportación del PDF.	2da Iteración
No se codifica la contraseña guardada de la autenticación con la base de datos en el fichero XML.	2da Iteración
Mejorar rendimiento del reporte “Conexiones por dominio”.	3ra Iteración
No se codifica la contraseña guardada de los usuarios del sistema.	3ra Iteración
No se muestran mensajes de error ni de notificación.	3ra Iteración
No se validan los campos de entrada de valores.	3ra Iteración

Tabla 4.8: No conformidades identificadas en el sistema

4.3. Conclusiones

Las pruebas funcionales efectuadas al producto permitieron identificar y corregir las no conformidades mejorando así la calidad del sistema. Se comprobó el correcto funcionamiento de la aplicación, demostrando que las funcionalidades no presentan irregularidades y que el desarrollo del software se ha realizado satisfactoriamente, cumpliendo así con los resultados esperados.

Conclusiones Generales

Con la culminación del presente trabajo se puede concluir que:

- ▶ Con el estudio realizado a las tecnologías de software y a las diferentes soluciones que generan reportes estadísticos de análisis de logs a nivel nacional e internacional, se facilitó el desarrollo del sistema informático, AiresProxy para Administradores de redes.
- ▶ El uso del lenguaje de programación C++ y MongoDB como tecnología de base de datos posibilitará un elevado poder de procesamiento de información, así como una mayor rapidez en el proceso de realización de reportes.
- ▶ El uso de la metodología SXP permitió guiar el desarrollo de forma acelerada y eficiente en correspondencia con las características de un equipo de pequeño formato.
- ▶ Con el desarrollo del sistema para la generación de reportes generales sobre el funcionamiento y el uso del servidor proxy, los Administradores de redes podrán contar con elementos de peso para la toma de decisiones administrativas y la aplicación de políticas de seguridad.
- ▶ Las pruebas realizadas con el jefe del proyecto AiresProxy arrojaron que el sistema se comporta de manera correcta. Por todo lo anteriormente planteado se concluye que fueron cumplidos todos los objetivos trazados.

Recomendaciones

Al concluir el presente trabajo se recomienda:

- ▶ Desarrollar una segunda versión del sistema, incorporando la generación de gráficas estadísticas, el trabajo con secciones de usuarios para el control de las trazas de usabilidad, y la asignación de roles y responsabilidades mediante la cual exista un administrador con permisos especiales de gestión.

Referencias

- [1] Centro Universitario de Ciencias Económico-Administrativas. Administración de redes. Universidad de Guadalajara: s.n., 2009.
- [2] Red Ethernet. [Disponible en: http://www.brivo.com/user_data/media/spanish_support/SP-ENG-DOC-020-EthernetAtAGlance.pdf], [En línea: 2003]. [Citado el: 9 de 10 de 2011].
- [3] Alzado. [Disponible en: http://www.alzado.org/articulo.php?id_art=83], Octubre [En línea: 03 de 9 de 2003]. [Citado el: 23 de 10 de 2011].
- [4] Windows. [Disponible en: <http://windows.microsoft.com/es-ES/windows-vista/What-is-a-proxy-server/>], [En línea: 2011]. [Citado el: 26 de 10 de 2011].
- [5] Ricardo Valenzuela. Universidad de Granada. [Disponible en: <http://hera.ugr.es/tesisugr/16474028.pdf>], [En línea: 2006]. [Citado el: 27 de 10 de 2011].
- [6] Mitchell, Bradley. [Disponible en: <http://compnetworking.about.com/cs/proxyservers/a/proxyservers.htm>], [En línea]. [Citado el: 28 de 10 de 2011].
- [7] Security Artwork Servidores Proxy caché. [Disponible en: <http://www.securityartwork.es>], [En línea: 3 de 09 de 2010]. [Citado el: 16 de 01 de 2012].
- [8] Squid Caché. [Disponible en: <http://www.squid-cache.org/>], [En línea: 10 de 01 de 2012]. [Citado el: 22 de 01 de 2012].
- [9] Microsoft. [Disponible en: <http://msdn.microsoft.com/en-us/library/ms826748.aspx>], [En línea: 2006]. [Citado el: 22 de 01 de 2012].
- [10] NetCenter. [Disponible en: <http://www.net-center.com.mx/html/bluecoat.html>], [En línea]. [Citado el: 22 de 01 de 2012].
- [11] Scribd. [Disponible en: <http://es.scribd.com/doc>], [En línea: 2011]. [Citado el: 12 de 11 de 2011].

- [12] Web Analytics Association. [Disponible en: <http://www.webanalyticsassociation.org/>], [En línea]. [Citado el: 12 de 11 de 2011].
- [13] Programming4. [Disponible en: <http://mscerts.programming4.us/es/310194.aspx>], [En línea]. [Citado el: 19 de 10 de 2011].
- [14] Sarg Squid Analysis Report Generator. [Disponible en: <http://sarg.sourceforge.net/>], [En línea]. [Citado el: 19 de 10 de 2011].
- [15] Itifix, Logrep. [Disponible en: <https://www.itefix.no/i2/logrepsrver>], [En línea: 2011]. [Citado el: 20 de 10 de 2011].
- [16] Sawmill. [Disponible en: <http://www.sawmill.net/>], [En línea]. [Citado el: 20 de 10 de 2011].
- [17] Otniel Barrera Palenzuela. ISAWEB Monitoreo de Tráfico en Internet. Ciudad de la Habana: s.n., 2006.
- [18] Xor Consultores. [Disponible en: <http://www.xorcon.com/servicios.php>], [En línea: 2007]. [Citado el: 05 de 11 de 2011].
- [19] GNU Consultores. [Disponible en: <http://www.gnuconsultores.com/es/ingenieria/desarrollo/escritorio>], [En línea]. [Citado el: 05 de 11 de 2011].
- [20] Peñalver Romero, Gladys Marsi. Metodología ágil para proyectos de software libre. Ciudad de La Habana: s.n., 2008.
- [21] R. M. Gregorio F. Z. Jorge. Programación extrema y Software Libre. [Disponible en: http://10.33.20.195/svn/Documentacion%20General/IGSW_Metodologias/libros/robles-ferrer-tod%20sobre%20XP.pdf], 2002.
- [22] Object Management Group, Business Process Model and Notation. [Disponible en: <http://www.bpmn.org/>], [En línea: 2012]. [Citado el: 22 de 01 de 2012].
- [23] Unified Modeling Language. [Disponible en: <http://www.uml.org/>], [En línea]. [Citado el: 29 de 10 de 2011].
- [24] C++ paso a paso Universidad de Alicante. [Disponible en: <http://gplsi.dlsi.ua.es/libros/cpp1/>], [En línea: 2006]. [Citado el: 05 de 11 de 2011].
- [25] Java. [Disponible en: <http://www.java.com/es/about/>], [En línea]. [Citado el: 22 de 01 de 2012].

- [26] Id Sistemas Informáticos. [Disponible en: <http://www.idsistemas.com/idsistemas.asp>], [En línea: 2006]. [Citado el: 29 de 10 de 2011].
- [27] Eclipse. [Disponible en: <http://www.eclipse.org/>], [En línea: noviembre del 2001]. [Citado el: 23 de 01 de 2012].
- [28] Geeks Linux Atelier. [Disponible en: <http://www.glatelier.org/2009/05/qt-creator-desarrollando-aplicaciones-rapidamente/>], [En línea: 15 de 05 de 2009]. [Citado el: 02 de 11 de 2011].
- [29] Netbeans. [Disponible en: http://netbeans.org/community/releases/61/index_es.html], [En línea]. [Citado el: 23 de 01 de 2012].
- [30] Página Oficial Visual Paradigm. [Disponible en: <http://www.visual-paradigm.com/>], [En línea]. [Citado el: 31 de 10 de 2011].
- [31] Página Oficial Rational. [Disponible en: <http://www.rational.com.ar/herramientas/roseenterprise.html>], [En línea: 2007]. [Citado el: 31 de 10 de 2011].
- [32] Página Oficial Oracle. [Disponible en: <http://www.oracle.com/es/index.html>], [En línea: 2010]. [Citado el: 23 de 01 de 2012].
- [33] Página Oficial MongoDB. [Disponible en: <http://www.mongodb.org/>], [En línea]. [Citado el: 01 de 11 de 2011].
- [34] Página Oficial MySQL. [Disponible en: <http://dev.mysql.com/doc/refman/5.0/es/what-is.html>], [En línea: 2011]. [Citado el: 23 de 01 de 2012].
- [35] Qt. [Disponible en: <http://qt.nokia.com/>], [En línea: 2008]. [Citado el: 02 de 11 de 2011].
- [36] O'Reilly XML.com. [Disponible en: <http://www.xml.com/>], [En línea: 2010]. [Citado el: 14 de 04 de 2012].
- [37] Universidad de Buenos Aires. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft Versión 1.0. [Disponible en: <http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:estiloypatron>], [En línea: Marzo de 2004]. [Citado el: 13 de 02 de 2012].
- [38] Proactiva Calidad. [Disponible en: <http://www.proactiva-calidad.com/java/patrones/index.html>], [En línea: Abril 2007]. [Citado el: 13 de 02 de 2012].

Bibliografía

- ✓ Carlos Billy Reynoso. Introducción a la Arquitectura de Software. Marzo de 2004.
- ✓ Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Design Patterns (“Patrones de Diseño”) (1995). 2005.
- ✓ Félix Prieto. Programación III.I.T.I. De Sistemas, Patrones de diseño. 2008/2009.
- ✓ Fernando Bellas Permuy. El lenguaje de programación C++.
- ✓ José Andrés García Romero de la Osa. Herramienta generadora de reportes de uso de una red local. Agosto de 2009.
- ✓ Programación Orientada a Objetos. Arquitecturas de software. Mayo de 2008.
- ✓ Ricardo Valenzuela Gaete. Aplicaciones del Soft Computing al análisis de ficheros log de sitios Web. 2006.

Glosario de Términos

AiresProxy: Analizador Inteligente de Registros Proxy.

Ejecución Asíncrona: Permite ejecutar hilos concurrentes dentro de una aplicación de software.

FTP: Protocolo de Transferencia de Archivos (siglas del inglés *File Transfer Protocol*).

GNU GPL: Licencia Pública General de GNU (siglas del inglés *GNU General Public License*).

GRASP: Patrones Generales de Software para Asignar Responsabilidades (siglas del inglés *General Responsibility Assignment Software Patterns*).

HTTP: Protocolo de Transferencia de Hipertexto (siglas del inglés *Hypertext Transfer Protocol*).

HTTPS: Protocolo Seguro de Transferencia de Hipertexto (siglas del inglés *Hypertext Transfer Protocol Secure*).

IDE: Entorno de Desarrollo Integrado (siglas del inglés *Integrated Development Environment*) es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios.

Internet: Gran red descentralizada de ordenadores, de ámbito global y públicamente accesible, que proporciona una ingente cantidad de servicios de comunicación de varios tipos, incluyendo la World Wide Web, el correo electrónico y muchos otros.

HIT: Peticiones aceptadas.

Host: Un host, literalmente anfitrión, es un ordenador directamente conectado a una red y que efectúa las funciones de un servidor, alberga servicios como correo electrónico y es accesible por otros ordenadores de la red.

KDE: Entorno de Escritorio K (siglas del inglés *K Desktop Environment*) es un proyecto de software libre para la creación de un entorno de escritorio e infraestructura de desarrollo para diversos sistemas operativos como GNU/Linux, Mac OS X y Windows. Se encuentra basado en las bibliotecas Qt, la "K", originariamente, representaba la palabra "Kool", pero su significado fue abandonado más tarde. Actualmente significa simplemente "K", la letra inmediatamente anterior a la "L" (inicial de Linux) en el

alfabeto.

MISS: Peticiones perdidas.

Plugin: Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

Software: Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

Sprint: Un Sprint es el procedimiento de adaptación de las cambiantes variables del entorno (requerimientos, tiempo, recursos, conocimiento, tecnología). Son ciclos iterativos en los cuales se desarrolla o mejora una funcionalidad para producir nuevos incrementos. Durante un Sprint el producto es diseñado, codificado y probado. Y su arquitectura y diseño evolucionan durante el desarrollo.

URI: (Siglas del inglés *Uniform Resource Identifier*) es un identificador uniforme de recursos. Es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, vídeos, presentaciones y presentaciones digitales. Se diferencia de la URL en que permite incluir en la dirección una subdirección, determinada por un “fragmento”.

URL: (Siglas del inglés *Uniform Resource Locator*) es un 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.

Anexos

Anexo # 1. Jerarquía de caché.

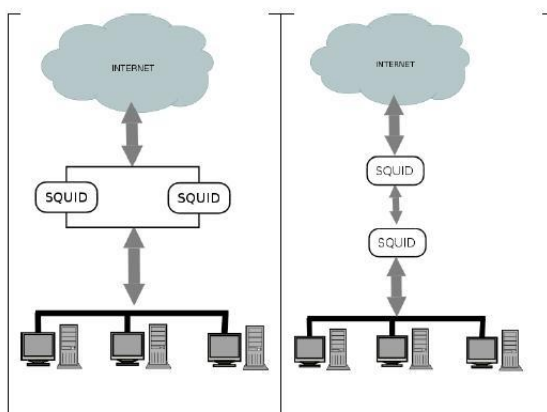


Figura 4.2: Jerarquía de caché

Anexo # 2. Interfaz de AiresProxy para Usuarios.



Figura 4.3: AiresProxy para Usuarios

Anexo # 3. Interfaz de AiresProxy para Administradores de redes.

The screenshot shows the AiresProxy web interface. The browser window title is "AiresProxy". The menu bar includes "Archivo", "Preferencias", and "Ayuda". The user is logged in as "Admin".

Reportes

- ▼ **Navegación**
 - Dominios más visitados
 - Mayores descargas
 - Dominios por usuario
 - Usuarios por dominio
 - Conexiones por dominio
 - Tipos de contenido
 - Mayores consumos por usua...
 - Direcciones IP de peticiones
- ▼ **Servidor**
 - Resumen general
 - Métodos de conexiones
 - Clasificación de protocolos
 - Respuesta de servidores
 - Direcciones IP de servidores
 - Jerarquía

Fecha Inicial
08-06-2012

Fecha Final
08-06-2012

Buscar

¡Bienvenido a AIRESPROXY!

ANALIZADOR INTELIGENTE DE REGISTROS PROXY

AiresProxy para Administradores de Redes visualiza diferentes reportes agrupados en dos categorías, los reportes asociados a las visitas y conexiones a Internet (Navegación) y los asociados al funcionamiento y rendimiento del servidor proxy (Servidor). Cada vista de los reportes se encuentra ordenada por la cantidad de conexiones realizadas.

AiresProxy

Departmento SINI | Centro: CIDI
Dirección de Redes y Seguridad Informática | Universidad de las Ciencias Informáticas

Figura 4.4: AiresProxy para Administradores de redes

Anexo # 4. Interfaz de la funcionalidad “Dominios más visitados”.

Monitorizando del 03-06-2008 al 08-06-2012

	Rank	Dominio	Conexiones	% Conexiones	Bytes	% Bytes
	1	foro.jovenclub.cu	21478	23.23%	48.93M	9.536%
	2	ak.fbcdn.net	13899	15.03%	83.38M	16.25%
	3	www.facebook.com	5462	5.907%	21.25M	4.141%
	4	google.com.cu	3578	3.869%	14.37M	2.8%
	5	channel.facebook.com	3448	3.729%	1.584M	0.3086%
	6	vanguardia.co.cu	2877	3.111%	7.063M	1.376%
	7	oscars.hola.com	1938	2.096%	5.71M	1.113%
	8	1hi.info	1536	1.661%	25.53M	4.975%
	9	www.talksms.com	1508	1.631%	10.55M	2.056%
	10	www.google.com	1139	1.232%	4.878M	0.9506%
	11	cfg.jovenclub.cu	1086	1.174%	5.758M	1.122%
	12	multimedia.hola.com	1012	1.094%	749K	0.1425%
	13	clients1.google.com	945	1.022%	1.396M	0.272%
	14	talkgadget.google.com	864	0.9344%	1.731M	0.3373%
	15	www.hola.com	708	0.7657%	2.971M	0.5789%
	16	connect.facebook.com	621	0.6716%	2.484M	0.484%
	17	s0.uvnimg.com	616	0.6662%	5.803M	1.131%

Fecha Inicial: 03-06-2008
Fecha Final: 08-06-2012
Buscar

1 de 16

Departamento SINI | Centro: CIDI
Dirección de Redes y Seguridad Informática | Universidad de las Ciencias Informáticas

Figura 4.5: Interfaz de la funcionalidad “Dominios más visitados”

Anexo # 5. Interfaz de la funcionalidad “Dominios por usuario”.

Monitorizando del 03-06-2008 al 08-06-2012

	Dominio	Conexiones	% Conexiones	Bytes	% Bytes	Usuario
1	ak.fbcdn.net	796	12.48%	3.668M	8.457%	anonymous_user232236
2	sony.com.mx	280	4.391%	1.961M	4.523%	anonymous_user232236
3	masoportunidades.com.ar	266	4.172%	1.571M	3.623%	anonymous_user232236
4	google.com.cu	266	4.172%	1.236M	2.849%	anonymous_user232236
5	www.facebook.com	262	4.109%	978.6K	2.203%	anonymous_user232236
6	urs.microsoft.com	260	4.078%	1024K	2.305%	anonymous_user232236
7	webmail.sld.cu	244	3.827%	852.3K	1.919%	anonymous_user232236
8	s5.scribdassets.com	214	3.356%	381.4K	0.8589%	anonymous_user232236
9	www.ojolink.com	210	3.294%	624.8K	1.407%	anonymous_user232236
10	www.google.com	202	3.168%	1.308M	3.015%	anonymous_user232236
11	channel.facebook.com	198	3.105%	70.33K	0.1584%	anonymous_user232236
12	pgnet.imagen-programa.com	182	2.854%	1.146M	2.641%	anonymous_user232236
13	img2.mlstatic.com	160	2.509%	451.2K	1.016%	anonymous_user232236
14	upload.wikimedia.org	146	2.29%	720.4K	1.622%	anonymous_user232236
15	img1.mlstatic.com	140	2.196%	394.2K	0.8877%	anonymous_user232236
16	o1.t26.net	118	1.851%	624.9K	1.407%	anonymous_user232236
17	s7.scribdassets.com	112	1.757%	798.4K	1.798%	anonymous_user232236

Fecha Inicial: 03-06-2008
Fecha Final: 08-06-2012
Buscar

Departamento SINI | Centro: CIDI
Dirección de Redes y Seguridad Informática | Universidad de las Ciencias Informáticas

Figura 4.6: Interfaz de la funcionalidad “Dominios por usuario”

Anexo # 6. Interfaz de la funcionalidad “Usuarios por dominio”.

Monitorizando del 03-06-2008 al 08-06-2012

	Usuario	Conexiones	% Conexiones	Bytes	% Bytes	Dominio
1	anonymous_user232236	202	17.73%	1.308M	26.8%	www.google.com
2	anonymous_user1565745	152	13.35%	545.6K	10.92%	www.google.com
3	anonymous_user1574812	130	11.41%	478.7K	9.583%	www.google.com
4	anonymous_user1383481	78	6.848%	20.64K	0.4132%	www.google.com
5	anonymous_user2395115	74	6.497%	158.1K	3.164%	www.google.com
6	anonymous_user185248	68	5.97%	115.2K	2.307%	www.google.com
7	anonymous_user691791	66	5.795%	449.3K	8.994%	www.google.com
8	anonymous_user1131717	42	3.687%	156.7K	3.137%	www.google.com
9	anonymous_user29973	36	3.161%	287.9K	5.763%	www.google.com
10	anonymous_user2019658	35	3.073%	297.2K	5.949%	www.google.com
11	anonymous_user138904	30	2.634%	284.6K	5.696%	www.google.com
12	anonymous_user2177201	28	2.458%	57.99K	1.161%	www.google.com
13	anonymous_user365550	20	1.756%	259.3K	5.192%	www.google.com
14	anonymous_user1965399	19	1.668%	81.15K	1.625%	www.google.com
15	anonymous_user187754	18	1.58%	4.742K	0.09493%	www.google.com
16	anonymous_user1310429	18	1.58%	122.9K	2.459%	www.google.com
17	anonymous_user1533776	16	1.405%	82.73K	1.656%	www.google.com

Departamento SINI | Centro: CIDI
Dirección de Redes y Seguridad Informática | Universidad de las Ciencias Informáticas

Figura 4.7: Interfaz de la funcionalidad “Usuarios por dominio”

Anexo # 7. Interfaz de la funcionalidad “Direcciones IP de peticiones”.

Monitorizando del 03-06-2008 al 08-06-2012

	Dirección	Conexiones	% Conexiones	Bytes	% Bytes
1	10.35.18.123	11192	12.1%	11.25M	2.193%
2	10.31.18.35	6376	6.895%	43.37M	8.452%
3	10.34.12.150	6280	6.792%	24.1M	4.696%
4	10.8.79.202	5744	6.212%	17.62M	3.434%
5	10.8.83.204	5568	6.022%	23.19M	4.519%
6	10.35.30.200	4620	4.996%	19.41M	3.782%
7	10.8.8.113	4048	4.378%	45.35M	8.838%
8	10.8.128.110	3394	3.67%	8.423M	1.641%
9	10.8.1.4	3041	3.289%	7.407M	1.443%
10	10.8.3.89	2230	2.412%	23.11M	4.503%
11	10.8.88.204	2032	2.198%	11.14M	2.17%
12	10.34.17.36	1866	2.018%	11.43M	2.227%
13	10.12.68.199	1806	1.953%	14.63M	2.852%
14	10.33.13.44	1701	1.84%	19.93M	3.884%
15	10.36.30.77	1674	1.81%	8.17M	1.592%
16	10.8.71.101	1648	1.782%	20.45M	3.986%
17	10.31.5.21	1632	1.765%	7.504M	1.462%

Departamento SINI | Centro: CIDI
Dirección de Redes y Seguridad Informática | Universidad de las Ciencias Informáticas

Figura 4.8: Interfaz de la funcionalidad “Direcciones IP de peticiones”

Anexo # 8. Interfaz de la funcionalidad “Resumen general”.

The screenshot shows the AiresProxy web interface. The top navigation bar includes 'Archivo', 'Preferencias', and 'Ayuda'. The user is logged in as 'Admin'. The main content area displays a report titled 'Monitorizando del 11-06-2008 al 08-06-2012'.

The report is organized into two main sections: 'Reportes' and 'Resumen'. The 'Reportes' section is expanded to show 'Servidor', which includes the 'Resumen general' report. The 'Resumen' section contains a table with the following data:

	Resumen	Total
1	Importe de peticiones	92468
2	Tráfico en bytes	513.1M
3	Cantidad de usuarios	61
4	Velocidad media de respuesta	1.88185s
5	Dominios irrelevantes	1
6	Dominios de ocio	153
7	Dominios bibliotecas	0
8	Dominios nacionales	3578
9	Dominios de interés	2

At the bottom of the interface, there is a footer with the text: 'Departamento SINI | Centro: CIDI' and 'Dirección de Redes y Seguridad Informática | Universidad de las Ciencias Informáticas'.

Figura 4.9: Interfaz de la funcionalidad “Resumen general”

Anexo # 9. Interfaz de la funcionalidad “Clasificación de protocolos”.

Monitorizando del 11-06-2008 al 08-06-2012

	Protocolos	Conexiones	% Conexiones	Bytes	% Bytes
1	TCP_MISS	24798	46.5%	191.4M	63.36%
2	TCP_IMS_HIT	13968	26.19%	4.305M	1.425%
3	TCP_HIT	8266	15.5%	80.02M	26.49%
4	TCP_REFRESH_HIT	3244	6.083%	16.73M	5.54%
5	TCP_MEM_HIT	2012	3.773%	4.931M	1.632%
6	TCP_NEGATIVE_HIT	499	0.9358%	315.7K	0.1021%
7	TCP_REFRESH_MISS	405	0.7595%	4.102M	1.358%
8	TCP_CLIENT_REFRESH_MISS	134	0.2513%	273.5K	0.08842%

Departamento SINI | Centro: CIDI
Dirección de Redes y Seguridad Informática | Universidad de las Ciencias Informáticas

Figura 4.10: Interfaz de la funcionalidad “Clasificación de protocolos”

Anexo # 10. Interfaz de la funcionalidad “Respuesta de servidores”.

Monitorizando del **11-06-2008** al **08-06-2012**

		Conexiones	% Conexiones	Bytes	% Bytes
1	200	33010	61.9%	290.9M	96.31%
2	304	17650	33.1%	5.469M	1.811%
3	302	1113	2.087%	635.9K	0.2056%
4	204	706	1.324%	249K	0.0805%
5	404	424	0.7951%	1.166M	0.3861%
6	400	119	0.2232%	127.6K	0.04124%
7	000	115	0.2157%	0	0%
8	301	90	0.1688%	53.17K	0.01719%
9	503	45	0.08439%	76.53K	0.02474%
10	206	31	0.05813%	3.351M	1.109%
11	403	7	0.01313%	4.944K	0.001599%
12	500	5	0.009376%	10.38K	0.003356%
13	504	4	0.007501%	1.972K	0.0006375%
14	401	3	0.005626%	2.502K	0.0008089%
15	502	2	0.003751%	1.896K	0.0006132%
16	410	2	0.003751%	910	0.0002873%

Departamento SINI | Centro: CIDI
Dirección de Redes y Seguridad Informática | Universidad de las Ciencias Informáticas

Figura 4.11: Interfaz de la funcionalidad “Respuesta de servidores”

Anexo # 11. Interfaz de la funcionalidad “Gestionar usuario”.



Figura 4.12: Interfaz de la funcionalidad “Gestionar usuario”

Anexo # 12. Documento en formato PDF generado por el sistema.



Figura 4.13: Documento en formato PDF generado por el sistema

Anexo # 13. Historias de Usuario.

Historia de Usuario	
Número: SGREAR-08	Nombre historia: Mostrar un resumen general sobre el uso del servidor proxy
Modificación de Historia de Usuario Número: 1	
Usuario: Alibech Nuñez Padilla	Iteración asignada: Segunda Iteración
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en desarrollo: Bajo	Puntos reales: 1

Descripción:	
El sistema mostrará, dado un rango de fechas, un resumen con estadísticas tales como, el importe total de peticiones, la cantidad de usuarios que han realizado peticiones, total de tráfico en bytes, velocidad media de respuesta, cantidad de sitios de ocio, de interés, nacionales, bibliotecas e irrelevantes.	
Observaciones:	
Prototipo de interfaces:	
<i>Monitorizando del 18-06-2009 al 08-06-2012</i>	
	<i>Resumen</i>
	<i>Total</i>
1	Importe de peticiones 92468
2	Tráfico en bytes 513.1M
3	Cantidad de usuarios 61
4	Velocidad media de respuesta 1.88185s
5	Dominios irrelevantes 1
6	Dominios de ocio 153
7	Dominios bibliotecas 0
8	Dominios nacionales 3578
9	Dominios de interés 2

Tabla 4.9: Historia de Usuario SGREAR-08. Mostrar resumen general del uso servidor proxy

Historia de Usuario	
Número: SGREAR-09	Nombre historia: Visualizar las direcciones IP por peticiones
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alibech Nuñez Padilla	Iteración asignada: Segunda Iteración
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en desarrollo: Bajo	Puntos reales: 1
Descripción:	
La aplicación mostrará, dado un rango de fechas, las direcciones IP donde se realizaron las conexiones a través de una tabla con las columnas: dirección (IP), la cantidad de peticiones, el por ciento que representa esas peticiones, los bytes y el por ciento que representan esos bytes.	
Observaciones:	

Prototipo de interfaces:

Monitorizando del 18-06-2009 al 08-06-2012					
	Dirección	Conexiones	% Conexiones	Bytes	% Bytes
1	10.35.18.123	11192	12.1%	11.25M	2.193%
2	10.31.18.35	6376	6.895%	43.37M	8.452%
3	10.34.12.150	6280	6.792%	24.1M	4.696%
4	10.8.79.202	5744	6.212%	17.62M	3.434%
5	10.8.83.204	5568	6.022%	23.19M	4.519%
6	10.35.30.200	4620	4.996%	19.41M	3.782%
7	10.8.8.113	4048	4.378%	45.35M	8.838%
8	10.8.128.110	3394	3.67%	8.423M	1.641%
9	10.8.1.4	3041	3.289%	7.407M	1.443%

Tabla 4.10: Historia de Usuario SGREAR-09. Visualizar las direcciones IP por peticiones

Historia de Usuario					
Número: SGREAR-10	Nombre historia: Mostrar un resumen estadístico sobre los diferentes métodos de conexiones				
Modificación de Historia de Usuario Número: Ninguna					
Usuario: Isabelys Luisa Suarez Sevilla	Iteración asignada: Segunda Iteración				
Prioridad en negocio: Alta	Puntos estimados: 1				
Riesgo en desarrollo: Bajo	Puntos reales: 1				
Descripción:					
Dado un rango de fechas, la aplicación mostrará un resumen estadístico sobre los diferentes métodos de conexiones (POST, GET, CONNECT, etc) en una tabla con las siguientes columnas: métodos, la cantidad de peticiones por cada método, el por ciento que representa estas peticiones, los bytes y el por ciento que representan esos bytes.					
Observaciones:					
Prototipo de interfaces:					
Monitorizando del 18-06-2009 al 08-06-2012					
	Métodos	Conexiones	% Conexiones	Bytes	% Bytes
1	GET	50706	95.09%	293.8M	97.27%
2	POST	2283	4.281%	2.946M	0.9753%
3	CONNECT	336	0.6301%	5.304M	1.756%
4	OPTIONS	1	0.001875%	1.534K	0.000496%

Tabla 4.11: Historia de Usuario SGREAR-10. Mostrar los diferentes métodos de conexiones

Historia de Usuario					
Número: SGREAR-11		Nombre historia: Visualizar la clasificación de los protocolos del resultado de las transacciones			
Modificación de Historia de Usuario Número: Ninguna					
Usuario: Isabelys Luisa Suarez Sevilla		Iteración asignada: Segunda Iteración			
Prioridad en negocio: Alta		Puntos estimados: 1			
Riesgo en desarrollo: Bajo		Puntos reales: 1			
Descripción: Dado un rango de fechas, el sistema mostrará la clasificación que tienen los protocolos cuando se realiza una transacción a través de una tabla con los estados, la cantidad de peticiones por cada estado, el por ciento que representa estas peticiones, los bytes, el por ciento que representan esos bytes.					
Observaciones:					
Prototipo de interfaces:					
<i>Monitorizando del 18-06-2009 al 08-06-2012</i>					
	<i>Protocolos</i>	<i>Conexiones</i>	<i>% Conexiones</i>	<i>Bytes</i>	<i>% Bytes</i>
1	TCP_MISS	24798	46.5%	191.4M	63.36%
2	TCP_IMS_HIT	13968	26.19%	4.305M	1.425%
3	TCP_HIT	8266	15.5%	80.02M	26.49%
4	TCP_REFRESH_HIT	3244	6.083%	16.73M	5.54%
5	TCP_MEM_HIT	2012	3.773%	4.931M	1.632%
6	TCP_NEGATIVE_HIT	499	0.9358%	315.7K	0.1021%
7	TCP_REFRESH_MISS	405	0.7595%	4.102M	1.358%
8	TCP_CLIENT_REFRESH_MISS	134	0.2513%	273.5K	0.08842%

Tabla 4.12: Historia de Usuario SGREAR-11. Visualizar la clasificación de los protocolos

Historia de Usuario

Número: SGREAR-12	Nombre historia: Mostrar la información de los códigos de jerarquía																													
Modificación de Historia de Usuario Número: Ninguna																														
Usuario: Alibech Nuñez Padilla	Iteración asignada: Tercera Iteración																													
Prioridad en negocio: Media	Puntos estimados: 0.5																													
Riesgo en desarrollo: Bajo	Puntos reales: 0.5																													
Descripción: El sistema permitirá visualizar, dado un rango de fechas, una tabla con el código de jerarquía, la cantidad de peticiones, el por ciento que representa estas peticiones, los bytes consumidos y el por ciento que representan esos bytes.																														
Observaciones:																														
Prototipo de interfaces:																														
<table border="1"> <thead> <tr> <th colspan="5">Monitorizando del 18-06-2009 al 08-06-2012</th> </tr> <tr> <th></th> <th>Jerarquía</th> <th>Conexiones</th> <th>% Conexiones</th> <th>Bytes</th> <th>% Bytes</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>DIRECT</td> <td>27989</td> <td>52.49%</td> <td>212.1M</td> <td>70.22%</td> </tr> <tr> <td>2</td> <td>NONE</td> <td>25139</td> <td>47.14%</td> <td>89.5M</td> <td>29.63%</td> </tr> <tr> <td>3</td> <td>FIRST_UP_PARENT</td> <td>198</td> <td>0.3713%</td> <td>433K</td> <td>0.14%</td> </tr> </tbody> </table>		Monitorizando del 18-06-2009 al 08-06-2012						Jerarquía	Conexiones	% Conexiones	Bytes	% Bytes	1	DIRECT	27989	52.49%	212.1M	70.22%	2	NONE	25139	47.14%	89.5M	29.63%	3	FIRST_UP_PARENT	198	0.3713%	433K	0.14%
Monitorizando del 18-06-2009 al 08-06-2012																														
	Jerarquía	Conexiones	% Conexiones	Bytes	% Bytes																									
1	DIRECT	27989	52.49%	212.1M	70.22%																									
2	NONE	25139	47.14%	89.5M	29.63%																									
3	FIRST_UP_PARENT	198	0.3713%	433K	0.14%																									

Tabla 4.13: Historia de Usuario SGREAR-12. Mostrar los códigos de jerarquía

Historia de Usuario	
Número: SGREAR-13	Nombre historia: Visualizar los mayores consumos por usuarios
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alibech Nuñez Padilla	Iteración asignada: Tercera Iteración
Prioridad en negocio: Media	Puntos estimados: 0.5
Riesgo en desarrollo: Bajo	Puntos reales: 0.5
Descripción: El sistema permitirá visualizar los usuarios que más consumen en la red según la cantidad de conexiones realizadas. Dado un rango de fechas se mostrará una tabla con: el usuario, la cantidad de peticiones, el por ciento que representa estas peticiones, los bytes consumidos y el por ciento que representa esos bytes.	

Observaciones:					
Prototipo de interfaces:					
<i>Monitorizando del 18-06-2009 al 08-06-2012</i>					
	<i>Usuario</i>	<i>Conexiones</i>	<i>% Conexiones</i>	<i>Bytes</i>	<i>% Bytes</i>
1	anonymous_user115800	11874	12.84%	13.35M	2.601%
2	anonymous_user232236	6376	6.895%	43.37M	8.452%
3	anonymous_user1280150	6280	6.792%	24.1M	4.696%
4	anonymous_user1176682	5904	6.385%	18.9M	3.682%
5	anonymous_user1533776	5618	6.076%	26.63M	5.19%
6	anonymous_user1301985	5568	6.022%	23.19M	4.519%
7	anonymous_user29973	4620	4.996%	19.41M	3.782%
8	anonymous_user1343432	4048	4.378%	45.35M	8.838%
9	anonymous_user1985368	3209	3.47%	7.712M	1.503%

Tabla 4.14: Historia de Usuario SGREAR-13. Mayores consumos por usuarios

Historia de Usuario	
Número: SGREAR-14	Nombre historia: Mostrar un resumen de las conexiones realizadas a los diferentes dominios
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alibech Nuñez Padilla	Iteración asignada: Tercera Iteración
Prioridad en negocio: Media	Puntos estimados: 0.5
Riesgo en desarrollo: Bajo	Puntos reales: 0.5
Descripción: Dado un rango de fechas, la aplicación permitirá mostrar un resumen sobre las conexiones a los diferentes tipos de dominios en una tabla con los siguientes datos: destino, la cantidad de peticiones por destino, el por ciento de las peticiones, los bytes y el por ciento de los bytes.	
Observaciones:	
Prototipo de interfaces:	

Monitorizando del 18-06-2009 al 08-06-2012					
	Destino	Conexiones	% Conexiones	Bytes	% Bytes
1	*.com	37048	40.07%	194.9M	37.99%
2	*.cu	32944	35.63%	130.2M	25.37%
3	*.net	14944	16.16%	90.9M	17.71%
4	*.org	3155	3.412%	39.19M	7.637%
5	*.info	1690	1.828%	27.4M	5.34%
6	*.es	760	0.8219%	4.034M	0.7862%
7	*.ar	399	0.4315%	3.131M	0.6101%
8	*.pe	366	0.3958%	7.802M	1.52%
9	*.mx	283	0.3061%	1.966M	0.3831%

Tabla 4.15: Historia de Usuario SGREAR-14. Mostrar las conexiones realizadas a los diferentes dominios

Historia de Usuario	
Número: SGREAR-15	Nombre historia: Visualizar las conexiones realizadas por cada tipo de contenido
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alibech Nuñez Padilla	Iteración asignada: Tercera Iteración
Prioridad en negocio: Media	Puntos estimados: 0.5
Riesgo en desarrollo: Bajo	Puntos reales: 0.5
Descripción: Dado un rango de fechas, el sistema permitirá visualizar las conexiones realizadas por cada tipo de contenido. Mostrará una tabla con los siguientes datos: tipo de contenido, la cantidad de peticiones por tipo de contenido, el por ciento de las peticiones, los bytes y el por ciento de los bytes.	
Observaciones:	
Prototipo de interfaces:	

Monitorizando del 18-06-2009 al 08-06-2012					
	Contenido	Conexiones	% Conexiones	Bytes	% Bytes
1	image/jpeg	11475	21.52%	100.2M	33.18%
2	image/gif	10343	19.4%	14.89M	4.93%
3	image/png	8750	16.41%	36.37M	12.04%
4	text/html	5694	10.68%	47.76M	15.81%
5	application/x-javascript	5177	9.708%	27.44M	9.085%
6	text/plain	2520	4.726%	2.835M	0.9387%
7	text/css	2404	4.508%	10.13M	3.355%
8	text/javascript	2107	3.951%	8.553M	2.832%
9	application/javascript	497	0.932%	5.132M	1.699%

Tabla 4.16: Historia de Usuario SGREAR-15. Visualizar las conexiones por tipo de contenido

Historia de Usuario	
Número: SGREAR-16	Nombre historia: Visualizar las direcciones IP de los servidores de origen
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Isabelys Luisa Suarez Sevilla	Iteración asignada: Tercera Iteración
Prioridad en negocio: Media	Puntos estimados: 0.5
Riesgo en desarrollo: Bajo	Puntos reales: 0.5
Descripción: El sistema mostrará, dado un rango de fechas, una tabla con el IP de los servidores de origen, la cantidad de peticiones, el por ciento de las peticiones, los bytes consumidos y el por ciento de los bytes.	
Observaciones:	
Prototipo de interfaces:	

Monitorizando del 18-06-2009 al 08-06-2012					
	Dirección	Conexiones	% Conexiones	Bytes	% Bytes
1	200.55.154.11	1363	4.87%	4.4M	2.075%
2	193.142.108.196	811	2.898%	4.985M	2.351%
3	204.12.225.170	766	2.737%	12.76M	6.017%
4	200.55.184.225	699	2.497%	6.348M	2.993%
5	209.85.157.101	676	2.415%	2.175M	1.026%
6	66.220.145.39	573	2.047%	285.8K	0.1316%
7	209.85.157.100	567	2.026%	1.754M	0.8272%
8	200.55.152.27	507	1.811%	3.649M	1.721%
9	69.63.180.43	396	1.415%	196.2K	0.09032%

Tabla 4.17: Historia de Usuario SGREAR-16. Visualizar las direcciones IP de los servidores

Historia de Usuario	
Número: SGREAR-17	Nombre historia: Permitir la autenticación con el sistema
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alibech Nuñez Padilla	Iteración asignada: Tercera Iteración
Prioridad en negocio: Media	Puntos estimados: 0.5
Riesgo en desarrollo: Bajo	Puntos reales: 0.5
Descripción: Para la autenticación con el sistema se deben introducir los datos relacionados con la conexión al servidor de base de datos MongoDB, ya sea la dirección del servidor, usuario y contraseña, además de introducir el usuario y la contraseña con que se iniciará la aplicación perteneciente a cada administrador de red.	
Observaciones:	
Prototipo de interfaces:	



Tabla 4.18: Historia de Usuario SGREAR-17. Permitir la autenticación con el sistema

Historia de Usuario	
Número: SGREAR-18.1	Nombre historia: Adicionar usuario
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alibech Nuñez Padilla	Iteración asignada: Tercera Iteración
Prioridad en negocio: Media	Puntos estimados: 0.5
Riesgo en desarrollo: Bajo	Puntos reales: 0.5
Descripción: El sistema permitirá la creación de nuevas cuentas de Administradores de redes, de manera que el usuario autenticado pueda adicionar nuevos usuarios.	
Observaciones:	
Prototipo de interfaces:	

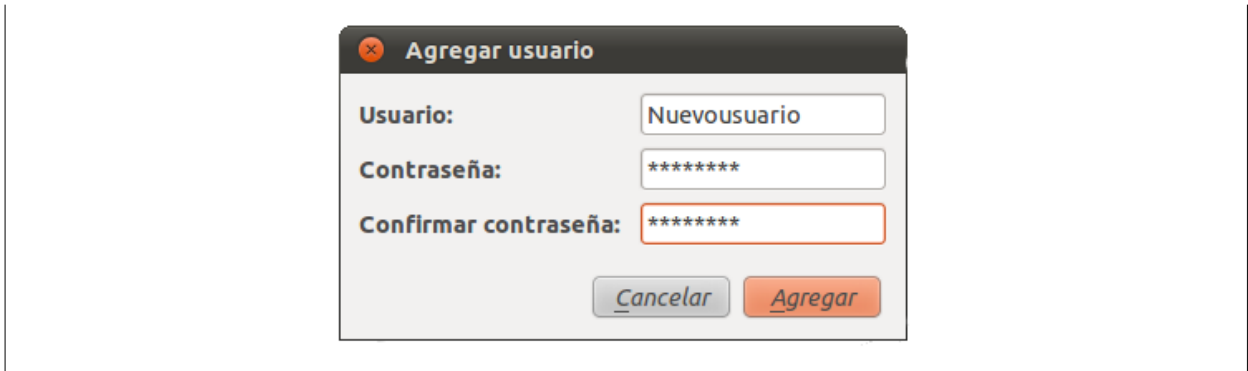


Tabla 4.19: Historia de Usuario SGREAR-18.1. Adicionar usuario

Historia de Usuario	
Número: SGREAR-18.2	Nombre historia: Eliminar usuario
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alibech Nuñez Padilla	Iteración asignada: Tercera Iteración
Prioridad en negocio: Media	Puntos estimados: 0.5
Riesgo en desarrollo: Bajo	Puntos reales: 0.5
Descripción: El sistema permitirá la eliminación de cuentas de Administradores de redes, de manera que el usuario autenticado pueda eliminar usuarios.	
Observaciones:	
Prototipo de interfaces:	

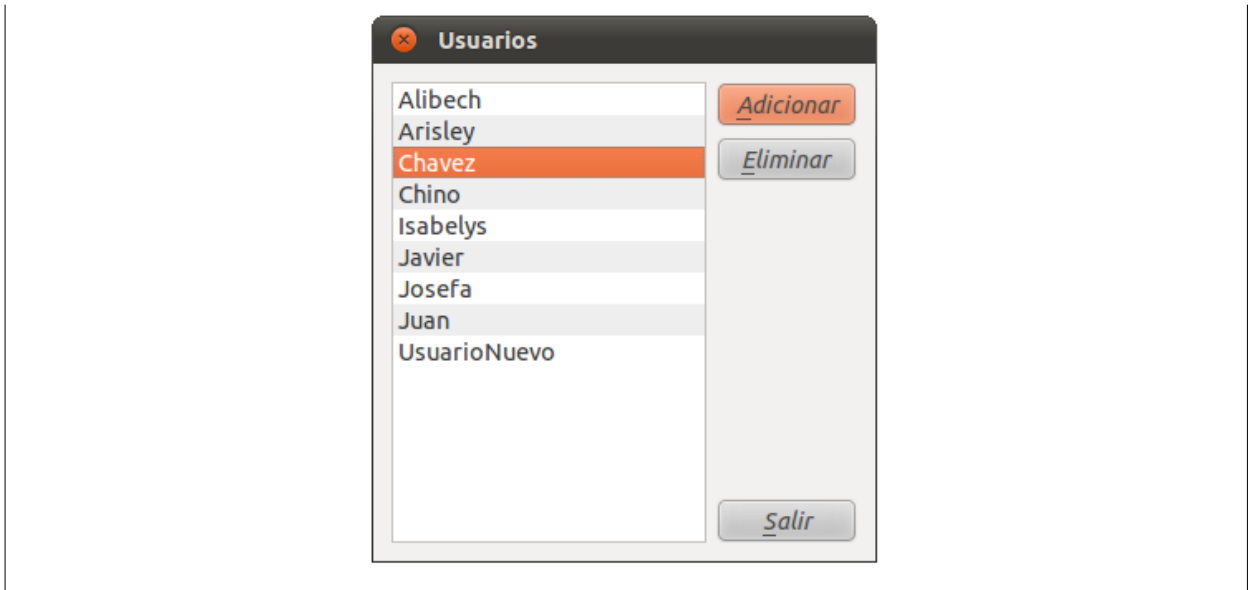


Tabla 4.20: Historia de Usuario SGREAR-18.2. Eliminar usuario

Historia de Usuario	
Número: SGREAR-18.3	Nombre historia: Cambiar contraseña de usuario
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Alibech Nuñez Padilla	Iteración asignada: Tercera Iteración
Prioridad en negocio: Media	Puntos estimados: 0.5
Riesgo en desarrollo: Bajo	Puntos reales: 0.5
Descripción: El sistema brindará la posibilidad de cambiar la contraseña del usuario autenticado.	
Observaciones:	
Prototipo de interfaces:	
<p>The image shows a window titled 'AiresProxy' with three input fields: 'Usuario:' containing 'Admin', 'Nueva contraseña:' containing '*****', and 'Confirmar contraseña:' containing '*****'. Below the fields are two buttons: 'Cancelar' (orange) and 'Aceptar' (grey).</p>	

Tabla 4.21: Historia de Usuario SGREAR-18.3. Cambiar contraseña de usuario

Anexo # 14. Descripción de la clase VentanaPrincipal.

Nombre:	VentanaPrincipal
Tipo de clase:	Interfaz
Atributo	Tipo
generador	Generador
observador	QFutureWatcher
arbolModelo	QStandardItemModel
Para cada responsabilidad	
Nombre:	VentanaPrincipal()
Descripción:	Constructor de la clase VentanaPrincipal, es donde se inicializan todos los atributos de la clase.
Nombre:	obtenerParametroTipoTexto(QString, QString) : QString
Descripción:	Permite levantar una ventana de diálogo, para introducir datos.
Nombre:	tituloRangoInicial() : void
Descripción:	Permite establecer el texto inicial de selección de rango.
Nombre:	cambiarTituloRango() : void
Descripción:	Permite modificar el mensaje de rango monitorizado según las fechas seleccionadas.
Nombre:	botonBuscarPresionado() : void
Descripción:	Función que se ejecuta al recibir el evento clicked del botón Buscar. Envía la señal al Generador para que la procese.
Nombre:	reportePresionado(QTreeWidgetItem *) : void
Descripción:	Permite mostrar las cabeceras de las columnas de las tablas una vez que se selecciona el reporte.
Nombre:	nuevaSeleccion(QTreeWidgetItem *) : void
Descripción:	Se ejecuta cuando hay un cambio de selección en los reportes.
Nombre:	autenticar(): void

Descripción:	Permite comprobar los datos de la autenticación con el sistema, enviando notificación al Generador.
Nombre:	comprobarConexion(): void
Descripción:	Permite comprobar la conexión con la base de datos del sistema, enviando notificación al Generador.
Nombre:	cambioContrasenna(): void
Descripción:	Le envía la notificación con los datos al Generador para el cambio de contraseña.
Nombre:	eliminarUsuario(): void
Descripción:	Le envía la notificación al Generador del usuario a eliminar.
Nombre:	crearUsuario(): void
Descripción:	Le envía la notificación al Generador del usuario a crear.
Nombre:	actualizarFicheroConfiguracion(): void
Descripción:	Actualiza el fichero XML con los nuevos datos de la conexión.

Tabla 4.22: Descripción de la clase VentanaPrincipal

Anexo # 15. Descripción de la clase Generador.

Nombre:	Generador
Tipo de clase:	Controladora
Atributo	Tipo
reporte	Reporte
conexión	Conexión
configuración	Configuración
usuarioApp	QString
usuariosApp	QStandardItemModel
Para cada responsabilidad	
Nombre:	Generador()
Descripción:	Constructor de la clase Generador, es donde se inicializan todos los atributos de la clase.

Nombre:	reporteSeleccionado(): void
Descripción:	Permite mostrar las cabeceras de las columnas de las tablas una vez que se selecciona el reporte.
Nombre:	obtenerReporte(): Reporte
Descripción:	Permite devolver el objeto reporte.
Nombre:	obtenerConexion(): Conexión
Descripción:	Permite devolver el objeto conexión.
Nombre:	obtenerConfiguracion(): Configuración
Descripción:	Permite devolver el objeto configuración.
Nombre:	cambiarUsuarioApp(QString): void
Descripción:	Permite cambiar al usuario de la aplicación.
Nombre:	obtenerUsuarioApp(): QString
Descripción:	Permite devolver al usuario de la aplicación.
Nombre:	comprobarUsuarioApp(QString, QString): bool
Descripción:	Permite comprobar si el usuario de la aplicación entrado es válido en el sistema.
Nombre:	cambiarContrasenna(QString): void
Descripción:	Permite el cambio de contraseña personal de cada usuario.
Nombre:	llenarListaUsuariosApp(): void
Descripción:	Se obtiene una lista de usuarios del sistema ordenados.
Nombre:	limpiarListaUsuarioApp(): void
Descripción:	Limpia la lista de usuarios de la aplicación.
Nombre:	eliminarUsuarioApp(QString): void
Descripción:	Permite eliminar usuarios con acceso al sistema.
Nombre:	crearUsuarioApp(QString, QString): void
Descripción:	Actualiza el fichero XML con los nuevos datos de la conexión.
Nombre:	existeUsuarioApp(QString): bool
Descripción:	Comprueba si existe o no un usuario con permisos en la aplicación.
Nombre:	actualizarDatos(QString): void
Descripción:	Actualiza los datos en el fichero de configuración.

Tabla 4.23: Descripción de la clase Generador

Anexo # 16. Descripción de la clase Reporte.

Nombre:	Reporte
Tipo de clase:	Entidad
Atributo	Tipo
fechaInicial	QDateTime
fechaFinal	QDateTime
listaReporte	QList
Para cada responsabilidad	
Nombre:	Reporte()
Descripción:	Constructor de la clase Reporte, es donde se inicializan todos los atributos de la clase.
Nombre:	convertir(float): QString
Descripción:	Permite convertir las cantidades, una vez que sobrepasen el límite, en Kilo-bytes, Mega-bytes, Giga-bytes
Nombre:	obtenerFechaDt(cadena : QString) : QDateTime
Descripción:	Permite obtener la fecha y devolverla en un formato legible.
Nombre:	obtenerFechaTs(QDateTime*) : long
Descripción:	Obtiene la fecha en el formato QDateTime y lo devuelve en el formato Timestamp para almacenarlo en la base de datos.
Nombre:	cambiarFechaInicial(QDateTime*) : void
Descripción:	Permite cambiar el valor de la fecha inicial.
Nombre:	cambiarFechaFinal(QDateTime*) : void
Descripción:	Permite cambiar el valor de la fecha final.
Nombre:	dominiosMasVisitados() : void
Descripción:	Realiza el algoritmo para obtener los dominios más visitados.
Nombre:	dominiosDescargas(): void
Descripción:	Realiza el algoritmo para obtener las mayores descargas.

Nombre:	dominiosPorUsuario(QString) : void
Descripción:	Realiza el algoritmo para mostrar los dominios accedidos por ese usuario.
Nombre:	usuariosPorDominio(QString) : void
Descripción:	Realiza el algoritmo para mostrar los usuarios que accedieron al dominio.
Nombre:	llenarDatosPaginado(): void
Descripción:	Calcula la cantidad de páginas a mostrar por cada reporte.
Nombre:	conexionesPorDominio(): void
Descripción:	Realiza el algoritmo para mostrar la cantidad de conexiones por cada dominio.
Nombre:	tipoContenido(): void
Descripción:	Realiza el algoritmo para mostrar la cantidad de conexiones por cada tipo de contenido.
Nombre:	direccionesIP(): void
Descripción:	Realiza el algoritmo para mostrar la cantidad de conexiones por cada dirección IP desde la cual se ha realizado una petición al servidor.
Nombre:	direccionesIpServidor(): void
Descripción:	Realiza el algoritmo para mostrar la cantidad de conexiones por cada dirección IP a la que se le ha realizado la petición.
Nombre:	jerarquia(): void
Descripción:	Realiza el algoritmo para mostrar la cantidad de conexiones por cada jerarquía.
Nombre:	metodosConexion(): void
Descripción:	Realiza el algoritmo para mostrar la cantidad de conexiones por cada método de conexión.
Nombre:	clasificacionProtocolos(): void
Descripción:	Realiza el algoritmo para mostrar la cantidad de conexiones por protocolo.
Nombre:	respuestaServidores(): void
Descripción:	Realiza el algoritmo para mostrar la cantidad de conexiones por respuesta del servidor.
Nombre:	llenarPaginaActual(int): void
Descripción:	Permite llenar la página actual con la lista de datos del reporte actual.

Nombre:	constructorReportes(): void
Descripción:	Permite construir los reportes con los datos de las páginas y los encabezados de las columnas.
Nombre:	adicionarObservador(QAbstractItemView *vista): void
Descripción:	Permite a las vistas actuar como observadoras esperando las señales del Reporte.

Tabla 4.24: Descripción de la clase Reporte

Anexo # 17. Descripción de la clase Conexión.

Nombre:	Conexion
Tipo de clase:	Entidad
Atributo	Tipo
usuario	QString
contraseña	QString
servidor	QString
conexión	DBClientConnection
cursor	auto_ptr<DBClientCursor>
Para cada responsabilidad	
Nombre:	Conexion()
Descripción:	Constructor de la clase Conexión, es donde se inicializan todos los atributos de la clase.
Nombre:	conectar(): void
Descripción:	Permite la conexión a la base de datos.
Nombre:	buscarRango(QString, float, float) : void
Descripción:	Busca el rango de fechas en la dirección de la colección.
Nombre:	buscarRangoUsuario(QString, float, float, QString): void
Descripción:	Permite buscar un usuario dentro de un rango determinado en la base de datos.
Nombre:	buscarRangoDominio(QString, float, float, int): void

Descripción:	Permite buscar un dominio dentro de un rango determinado en la base de datos.
Nombre:	continuar() : bool
Descripción:	Verifica si se puede seguir iterando por los resultados de una búsqueda.
Nombre:	buscar(QString, BSONObj): BSONObj
Descripción:	Permite la búsqueda de un documento en la base de datos.
Nombre:	obtenerCursor() : auto_ptr<DBClientCursor>
Descripción:	Permite obtener el puntero de la búsqueda actual.
Nombre:	proximo(): BSONObj
Descripción:	Representa el documento que se obtuvo, que se está analizando.
Nombre:	buscarElementos(QString, BSONObj): void
Descripción:	Busca un elemento dentro de una colección.
Nombre:	comprobarConexion(QString, QString, QString): bool
Descripción:	Comprueba la conexión con la base de datos.
Nombre:	eliminarDocumento(QString, BSONObj): void
Descripción:	Permite eliminar un elemento de una colección.
Nombre:	crearDocumento(QString, BSONObj): void
Descripción:	Permite crear un elemento de una colección.
Nombre:	obtenerServidor(): QString
Descripción:	Permite devolver el servidor.
Nombre:	obtenerUsuario(): QString
Descripción:	Permite devolver el usuario.
Nombre:	obtenerContrasenna(): QString
Descripción:	Permite devolver la contraseña.
Nombre:	cambiarServidor(QString): void
Descripción:	Permite cambiar el servidor.
Nombre:	cambiarUsuario(QString): void
Descripción:	Permite cambiar el usuario.
Nombre:	cambiarContrasenna(QString): void
Descripción:	Permite cambiar la contraseña.

Tabla 4.25: Descripción de la clase Conexión

Anexo # 18. Descripción de la clase Configuración.

Nombre:	Configuracion
Tipo de clase:	Entidad
Atributo	Tipo
dirConfiguracion	QString
mservidor	QString
musuario	QString
mcontrasenna	QString
Para cada responsabilidad	
Nombre:	Configuracion()
Descripción:	Constructor de la clase Configuración, es donde se inicializan todos los atributos de la clase.
Nombre:	obtenerMServidor(): QString
Descripción:	Permite obtener el servidor del fichero de la configuración.
Nombre:	obtenerMUsuario(): QString
Descripción:	Permite obtener el usuario del fichero de la configuración.
Nombre:	obtenerMContrasenna(): QString
Descripción:	Permite obtener la contraseña del fichero de la configuración.
Nombre:	cambiarMServidor(QString): void
Descripción:	Permite cambiar el servidor del fichero de la configuración.
Nombre:	cambiarMUsuario(QString): void
Descripción:	Permite cambiar el usuario del fichero de la configuración.
Nombre:	cambiarMContrasenna(QString): void
Descripción:	Permite cambiar la contraseña del fichero de la configuración.
Nombre:	obtenerMServidorInterno(): QString
Descripción:	Permite realizar llamadas internas para devolver el servidor del fichero de la configuración.

Nombre:	obtenerMUsuarioInterno(): QString
Descripción:	Permite realizar llamadas internas para devolver el usuario del fichero de la configuración.
Nombre:	obtenerMContrasenaInterna(): QString
Descripción:	Permite realizar llamadas internas para devolver la contraseña del fichero de la configuración.
Nombre:	actualizarDatos(QString): void
Descripción:	Actualiza los datos de la dirección y de los valores del fichero de configuración.
Nombre:	actualizarDatosInternos(): void
Descripción:	Actualiza los datos internamente de los valores del fichero de configuración.
Nombre:	actualizarFichero(QString, QString, QString): void
Descripción:	Actualiza el fichero de configuración.

Tabla 4.26: Descripción de la clase Configuración

Anexo # 19. Descripción de la colección Domains.

Nombre de la colección:	Domains	
Descripción:	En esta colección es donde se guardan los dominios y los identificadores relacionados con los logs.	
Nombre	Tipo	Descripción
_id	ObjectId	Contiene un valor especial de tipo BSON utilizado como llave primaria para identificar de forma única a cada documento.
id1	Float	Almacena el identificador que relaciona el dominio con la información de la url.
d	String	Almacena el nombre del dominio.
cl	String	Contiene la categoría que representa cada dominio, si es de interés, de ocio, irrelevante, nacional o biblioteca.

Tabla 4.27: Descripción de la colección Domains

Anexo # 20. Descripción de la colección Logs.

Nombre de la colección:	Logs	
Descripción:	En esta colección es donde se guardan todos los datos de los logs generados.	
Nombre	Tipo	Descripción
_id	ObjectId	Contiene un valor especial de tipo BSON utilizado como llave primaria para identificar de forma única a cada documento.
ts	Float	Guarda la información relacionada con la fecha de consulta en formato <i>timestamp</i> .
tr	String	Representa el tamaño de la URL en bytes.
ca	String	Contiene la dirección IP desde donde se realizó la conexión.
u	String	Guarda la URL de la solicitud del cliente.
ci	String	Almacena el nombre del usuario que realizó la conexión.
d	Float	Almacena un número que representa un identificador de la colección Domains.
rm	String	Contiene el método de solicitud. El método más común de solicitud HTTP es GET.
rt	Float	Este campo indica la cantidad de tiempo que se tardó en procesar la solicitud.
p	String	Almacena la información de interconexión que consta de dos fichas, separadas por una barra. El primer elemento indica el siguiente salto que fue elegido. El segundo elemento es la dirección de ese salto.
ct	String	Almacena el tipo de contenido de la respuesta HTTP de la solicitud.
hrq	String	Almacena la cabecera de petición HTTP.
hrp	Float	Almacena el encabezado de respuesta HTTP.

rs	String	Almacena dos fichas separadas por una barra. El primer código de identificador, el resultado, clasifica el protocolo y el resultado de una transacción. La segunda razón es la respuesta HTTP de código de estado.
----	--------	--

Tabla 4.28: Descripción de la colección Logs

Anexo # 21. Descripción de la colección Usuarios.

Nombre de la colección:	Usuarios	
Descripción:	En esta colección se registran los datos con los que se iniciará el sistema, almacenando el usuario y contraseña de cada administrador de red.	
Nombre	Tipo	Descripción
_id	ObjectId	Contiene un valor especial de tipo BSON utilizado como llave primaria para identificar de forma única a cada documento.
user	String	Almacena los usuarios de los Administradores de redes encargados de iniciar el sistema.
pass	String	Almacena la contraseña codificada por el método SHA-1 ⁸

Tabla 4.29: Descripción de la colección Usuarios

Anexo # 22. Casos de pruebas de aceptación.

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-08	Nombre Historia de Usuario: Mostrar un resumen general sobre el uso del servidor proxy
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: El sistema muestra un resumen estadístico del uso del servidor proxy a partir de un rango de fechas determinado.	

⁸Algoritmo de Hash Seguro (siglas del inglés, *Secure Hash Algorithm*), es un sistema de funciones hash criptográficas.

<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Resumen general”.
<p>Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Servidor, posteriormente seleccionar Resumen general. Introducir la fecha inicial y la fecha final, oprimir el botón Buscar.</p>
<p>Resultado Esperado: El sistema muestra una tabla con el importe total de peticiones, la cantidad de usuarios que han realizado peticiones, total de tráfico en bytes, la velocidad media de respuesta y la cantidad de sitios de ocio, de interés, nacionales, bibliotecas e irrelevantes.</p>
<p>Evaluación de la Prueba: Satisfactorio.</p>

Tabla 4.30: Caso de prueba de aceptación SGREAR-08

Casos de Pruebas de Funcionalidad	
Código Caso de Prueba: SGREAR-09	Nombre Historia de Usuario: Visualizar las direcciones IP por peticiones
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: El sistema visualiza las direcciones IP donde se realizaron las conexiones a partir de un rango de fechas determinado.	
<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Direcciones IP de peticiones”. 	

<p>Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Navegación, posteriormente seleccionar Direcciones IP de peticiones. Introducir la fecha inicial y la fecha final, oprimir el botón Buscar.</p>
<p>Resultado Esperado: El sistema muestra una tabla con las direcciones, la cantidad de conexiones por cada dirección, el por ciento que representan estas peticiones, la cantidad de bytes consumidos y el por ciento que representan esos bytes. La tabla se encuentra ordenada de forma descendente por la cantidad de conexiones.</p>
<p>Evaluación de la Prueba: Satisfactorio.</p>

Tabla 4.31: Caso de prueba de aceptación SGREAR-09

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-10	Nombre Historia de Usuario: Mostrar los diferentes métodos de conexiones
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: El sistema visualiza los diferentes métodos de conexión (POST, GET, CONNECT) a partir de un rango de fechas determinado.	
<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Métodos de conexiones”. 	
<p>Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Servidor, posteriormente seleccionar Métodos de conexiones. Introducir la fecha inicial y la fecha final, oprimir el botón Buscar.</p>	
<p>Resultado Esperado: El sistema muestra una tabla con los métodos, la cantidad de conexiones por cada método, el por ciento que representan esos métodos, la cantidad de bytes consumidos y el por ciento que representan esos bytes. La tabla se encuentra ordenada de forma descendente por la cantidad de conexiones.</p>	

Evaluación de la Prueba: Satisfactorio.

Tabla 4.32: Caso de prueba de aceptación SGREAR-10

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-11	Nombre Historia de Usuario: Visualizar la clasificación de los protocolos
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: El sistema visualiza las diferentes clasificaciones de los protocolos (TCP_MISS, TCP_HIT) a partir de un rango de fechas determinado.	
Condiciones de Ejecución: <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Clasificación de protocolos”. 	
Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Servidor, posteriormente seleccionar Clasificación de protocolos. Introducir la fecha inicial y la fecha final, oprimir el botón Buscar.	
Resultado Esperado: El sistema muestra una tabla con los protocolos, la cantidad de conexiones por cada protocolo, el por ciento que representan los protocolos, la cantidad de bytes consumidos y el por ciento que representan. La tabla se encuentra ordenada de forma descendente por la cantidad de conexiones.	
Evaluación de la Prueba: Satisfactorio.	

Tabla 4.33: Caso de prueba de aceptación SGREAR-11

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-12	Nombre Historia de Usuario: Mostrar los códigos de jerarquía

Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso
Descripción de la Prueba: El sistema muestra los diferentes códigos de jerarquía (DIRECT, NONE) a partir de un rango de fechas determinado.
Condiciones de Ejecución: <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Jerarquía”.
Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Servidor, posteriormente seleccionar Jerarquía. Introducir la fecha inicial y la fecha final, oprimir el botón Buscar.
Resultado Esperado: El sistema muestra una tabla con los códigos de jerarquía, la cantidad de conexiones por cada código, el por ciento que representa cada código, la cantidad de bytes consumidos y el por ciento que representan los bytes. La tabla se encuentra ordenada de forma descendente por la cantidad de conexiones.
Evaluación de la Prueba: Satisfactorio.

Tabla 4.34: Caso de prueba de aceptación SGREAR-12

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-13	Nombre Historia de Usuario: Visualizar los usuarios más consumidores
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: El sistema muestra los usuarios que más consumen en la red ordenados de forma descendente a partir de un rango de fechas determinado.	

<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Mayores consumos por usuarios”.
<p>Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Navegación, posteriormente seleccionar Mayores consumos por usuarios. Seleccionar la fecha inicial y la fecha final, oprimir el botón Buscar. Introducir la cantidad de usuarios que desea visualizar, pulsar el botón Aceptar.</p>
<p>Resultado Esperado: El sistema muestra una tabla con los usuarios, la cantidad de conexiones por cada uno de ellos, el por ciento que representan esas conexiones, la cantidad de bytes consumidos y el por ciento que representan los bytes. La tabla se encuentra ordenada de forma descendente por la cantidad de conexiones.</p>
<p>Evaluación de la Prueba: Satisfactorio.</p>

Tabla 4.35: Caso de prueba de aceptación SGREAR-13

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-14	Nombre Historia de Usuario: Mostrar las conexiones realizadas a los diferentes dominios
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: El sistema muestra las diferentes conexiones por tipos de dominios (*.com, *.cu) a partir de un rango de fechas determinado.	

<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Conexiones por Dominio”.
<p>Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Navegación, posteriormente seleccionar Conexiones por dominio. Introducir la fecha inicial y la fecha final, oprimir el botón Buscar.</p>
<p>Resultado Esperado: El sistema muestra una tabla con los destinos, la cantidad de conexiones, el por ciento que representan, la cantidad de bytes consumidos y el por ciento que representan esos bytes. La tabla se encuentra ordenada de forma descendente por la cantidad de conexiones.</p>
<p>Evaluación de la Prueba: Satisfactorio.</p>

Tabla 4.36: Caso de prueba de aceptación SGREAR-14

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-15	Nombre Historia de Usuario: Visualizar las conexiones por tipo de contenido
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: El sistema muestra las diferentes conexiones por tipos de contenidos (imagen/jpeg, text/html) a partir de un rango de fechas determinado.	

<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Tipos de contenido”.
<p>Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Navegación, posteriormente seleccionar Tipos de contenido. Introducir la fecha inicial y la fecha final, oprimir el botón Buscar.</p>
<p>Resultado Esperado: El sistema muestra una tabla con los contenidos, la cantidad de conexiones por cada tipo de contenido, el por ciento que representan los contenidos, la cantidad de bytes consumidos y el por ciento que representan los bytes. La tabla se encuentra ordenada de forma descendente por la cantidad de conexiones.</p>
<p>Evaluación de la Prueba: Satisfactorio.</p>

Tabla 4.37: Caso de prueba de aceptación SGREAR-15

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-16	Nombre Historia de Usuario: Visualizar las direcciones IP de los servidores
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: A partir de un rango de fechas determinado, el sistema visualiza las diferentes direcciones IP de los servidores de origen.	

<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. ▶ Seleccionar el reporte “Direcciones IP de servidores”.
<p>Entrada / Pasos de ejecución: El usuario debe comprobar la conexión y autenticarse en el sistema. Pulsar sobre el vínculo Servidor, posteriormente seleccionar Direcciones IP de servidores. Introducir la fecha inicial y la fecha final, oprimir el botón Buscar.</p>
<p>Resultado Esperado: El sistema muestra una tabla con las direcciones, la cantidad de conexiones por cada dirección, el por ciento que representan las direcciones, la cantidad de bytes consumidos y el por ciento que representan los bytes. La tabla se encuentra ordenada de forma descendente por la cantidad de conexiones.</p>
<p>Evaluación de la Prueba: Satisfactorio.</p>

Tabla 4.38: Caso de prueba de aceptación SGREAR-16

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-17	Nombre Historia de Usuario: Permitir la autenticación con el sistema
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: La aplicación permite la correcta autenticación.	
<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. 	

Entrada / Pasos de ejecución: El usuario debe introducir los datos de la conexión, servidor, usuario y contraseña; además de introducir los datos (usuario y contraseña) del usuario con que se inicializará la aplicación.
Resultado Esperado: Se inicia el sistema.
Evaluación de la Prueba: Satisfactorio.

Tabla 4.39: Caso de prueba de aceptación SGREAR-17

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-18.1	Nombre Historia de Usuario: Adicionar usuario
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: La aplicación permite la creación de nuevas cuentas de usuarios.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. 	
Entrada / Pasos de ejecución: Ir a la barra de herramientas, pulsar el menú Preferencias, Gestionar usuarios, pulsar el botón Adicionar, introducir los datos de la nueva cuenta, presionar el botón Agregar; o presionar directamente el icono Gestionar usuarios, pulsar el botón Adicionar, introducir los datos de la nueva cuenta, presionar el botón Agregar.	
Resultado Esperado: Se adiciona una nueva cuenta de usuario en la aplicación.	
Evaluación de la Prueba: Satisfactorio.	

Tabla 4.40: Caso de prueba de aceptación SGREAR-18.1

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-18.2	Nombre Historia de Usuario: Eliminar usuario
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: La aplicación permite la eliminación de cuentas de usuarios.	

<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema.
<p>Entrada / Pasos de ejecución: Ir a la barra de herramientas, pulsar el menú Preferencias, Gestionar usuarios, seleccionar el usuario que se desea eliminar, pulsar el botón Eliminar; o presionar directamente el icono Gestionar usuarios, seleccionar el usuario que se desea eliminar, pulsar el botón Eliminar.</p>
<p>Resultado Esperado: Se elimina una cuenta de usuario de la aplicación.</p>
<p>Evaluación de la Prueba: Satisfactorio.</p>

Tabla 4.41: Caso de prueba de aceptación SGREAR-18.2

Caso de Prueba de Funcionalidad	
Código Caso de Prueba: SGREAR-18.3	Nombre Historia de Usuario: Cambiar contraseña de usuario
Nombre de la persona que realiza la prueba: Miguel Ángel Chávez Alfonso	
Descripción de la Prueba: El sistema permite cambiar la contraseña del usuario que inicializa la aplicación.	
<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> ▶ Servidor de MongoDB en ejecución. ▶ Conexión entre la aplicación y la base de datos. ▶ Autenticarse en el sistema. 	
<p>Entrada / Pasos de ejecución: Ir a la barra de herramientas, pulsar el menú Preferencias, Cambiar contraseña, introducir la nueva contraseña, pulsar el botón Aceptar; o presionar directamente el icono Cambiar contraseña, introducir la nueva contraseña, pulsar el botón Aceptar.</p>	

Resultado Esperado: Se cambia la contraseña del usuario.
Evaluación de la Prueba: Satisfactorio.

Tabla 4.42: Caso de prueba de aceptación SGREAR-18.3