

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 10



TÍTULO: Analizador de Registros de Servidores Proxy.

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN
CIENCIAS INFORMÁTICA**

AUTOR(ES): Carlos Javier Pérez Mutuberría

Jorge Luis León Vázquez

TUTOR: Ing. José Ramón Hermosilla Moreno

Ciudad de la Habana, Cuba

Junio 2009

“Año del 50 aniversario del triunfo de la Revolución”

DEDICATORIA

De Jorge Luis:

A mi papa:

Porque siempre creyó en mí y deseó siempre que fuera universitario.

A mi mama

Porque fue, ha sido y será todo en mi vida.

A mi hermano

Porque a pesar de todo los trabajos siempre tenemos el mismo camino y espero seguir sirviéndole de guía en la vida.

A mis abuelos

Por todo el cariño y amor que me brindaron.

De Carlos Javier:

A mi familia y en especial a mi abuelito Ñico.

AGRADECIMIENTOS

De Jorge Luis:

A mi papá, a mi mamá y a mi hermano:

Porque sin ellos no soy nada y no sería quien soy hoy.

A mi familia:

Por todas las preocupaciones y el apoyo brindado.

A Yuni:

Por aguantarme todos estos años.

A mis suegros:

Por acogerme como otro hijo.

A mi compañero de tesis Carlos Javier:

Por compartir juntos nuestra última experiencia como estudiantes.

A Lola:

Por todas las luchas que cogió con nosotros.

A mi grupo:

Por estos 5 años de buenos y malos momentos.

A todos mis amigos de la UCI y Santa Clara:

Por compartir un pedazo de nuestras vidas.

A la UCI:

Por permitirme encontrar nuevos amigos

De Carlos Javier:

A mi familia, mi mamá, mi papá y mis hermanos por la educación y el cariño que me han dado, sin ellos nada de esto hubiese sido posible.

A Lola por compartir todo este tiempo conmigo y apoyarme siempre.

A León por ser mi compañero de tesis y mi amigo.

A todos mis amigos, compañeros y los que de una forma u otra me han ayudado, que han estado ahí cuando los he necesitado, que han estado ahí siempre.

A todos muchas gracias.

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de junio del año 2009.

Autor(es): Carlos Javier Pérez Mutuberría

Jorge Luis León Vázquez

Tutor: Ing. José Ramón Hermosilla Moreno

RESUMEN

El control de las actividades de los usuarios que navegan en internet es una tarea importante en cualquier empresa. Conocer los sitios visitados, su clasificación en categorías de acuerdo al contenido, si están o no vinculados a la actividad laboral, con qué frecuencia y desde donde se consultaron y cuál fue la cantidad de megas transferidos, son algunas de las informaciones que brindan los registros (log) de servidores proxy. Dicho conocimiento es muy importante para realizar análisis estadísticos, utilizados fundamentalmente a la hora de tomar decisiones para mantener la seguridad y el buen funcionamiento de una organización.

En nuestro país, con el crecimiento del uso de la Internet en las instituciones laborales, se ha incrementado también el uso incorrecto de la navegación por la red y se hace cada vez más difícil llevar el control de la misma, por no contar con una herramienta lo suficientemente eficiente que permita llevar un análisis de los registros de los servidores proxy, provocando violaciones de seguridad en la organización, descargas no controladas de grandes cúmulos de datos, pérdida del tiempo laboral, dificultades a la hora de tomar decisiones por parte de los administrativos de la empresa al enfocarse en datos no reales u otros que no son imprescindibles, incidiendo directamente en los resultados de la institución.

El presente trabajo, está enmarcado en el desarrollo de una aplicación, que permita realizar el análisis de los registros de servidores proxy de una forma eficaz, con una interfaz sencilla para el intercambio con el usuario, en respuesta a la necesidad de mejorar y perfeccionar la manera en que actualmente se lleva a cabo esta tarea en el país. Se realizó un estudio de las diferentes herramientas existentes y sus particularidades, y se expone la fundamentación de las metodologías y las tecnologías utilizadas en la propuesta del sistema. Con la realización del mismo tendríamos una mejora considerable en cuanto a eficiencia, tiempo y organización, contribuyendo favorablemente al desenvolvimiento de la organización.

ABSTRACT

Controlling the activities of users who browse the internet is an important task in any enterprise. Some of the information that log proxy servers provide are the most visited sites, their classification into categories, how often from where they were consulted their access frequency and origin and the amount of megabytes transferred. This knowledge is very important for statistical analysis, used primarily for making decisions to maintain safety and smooth functioning of an organization.

In Cuba, the growth of internet use has brought about a parallel increase of navigation using networks, of which is highly difficult to keep track. The lack of efficient tools to continue an analysis of the logs of proxy servers causes security breaches in the organization, uncontrolled discharges of large clusters of data, loss of working time and difficulties in making decisions by the company managers and will have a direct impact on the results of the institution.

The objective of this work is to develop an application that allows the analysis of proxy log in an effective manner, with a simple interface for the user to exchange with the application. It was made a study of the available tools and their specifications and the methodologies and technologies used in the proposed system. This work contributes to the efficiency, time and organization and the positive development of the enterprise.

Índice

□	INTRODUCCIÓN	14
□	CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	19
	1.1 INTRODUCCIÓN	19
	1.2 ANALIZADORES DE REGISTRO DE SERVIDORES PROXY A NIVEL INTERNACIONAL.	19
	1.2.1 AWStats	19
	1.2.2 Webalizer	20
	1.2.3 Calamaris	21
	1.2.4 Sarg.....	21
	1.2.5 Squid-graph.....	21
	1.2.6 WebTrends Professional	22
	1.2.7 Sawmill.....	22
	1.3 ANALIZADORES DE REGISTRO DE SERVIDORES PROXY A NIVEL NACIONAL.	23
	1.3.1 ISASWeb	24
	1.3.2 AAInternet	24
	1.3.3 SICC.....	25
	1.3.4 SRNI.....	26
	1.4 LENGUAJES DE PROGRAMACIÓN.....	26
	1.4.1 Perl.....	26
	1.4.2 PHP	27
	1.4.2.1 Php 5.....	28
	1.4.3 JavaScript	29
	1.5 SERVIDOR DE BASE DE DATOS.....	30
	1.5.1 PostgreSQL.....	30
	1.6 SERVIDOR WEB	31
	1.6.1 Apache	31
	1.7 FRAMEWORK	32
	1.7.1 Symfony	32
	1.8 METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	33
	1.8.1 Rational Unified Process (RUP).....	33
	1.8.2 Lenguaje Unificado de Modelado (UML).....	34
	1.9 HERRAMIENTAS A UTILIZAR	36
	1.9.1 Eclipse.....	36
	1.9.2 Visual Paradigm	37
	1.10 CONCLUSIONES	38
□	CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA	39
	2.1 INTRODUCCIÓN	39
	2.2 DESCRIPCIÓN DEL PROBLEMA	39
	2.2.1 Servidor Proxy	39
	2.2.2 Log del servidor proxy Squid	41
	2.2.2.1 ¿Por qué Squid?.....	41
	2.2.2.2 ¿Qué información almacenan los log?	41
	2.3 MODELO DE DOMINIO.....	43
	2.4 SOLUCIÓN PROPUESTA	45
	2.5 ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA.....	45
	2.5.1 Requisitos funcionales.....	45

2.5.1.1	Interfaz web	46
2.5.1.2	Analizador.....	46
2.5.2	Requisitos no funcionales.....	47
2.5.2.1	Requerimientos de software	47
2.5.2.2	Requerimientos de hardware	47
2.5.2.3	Requerimientos de usabilidad.....	47
2.5.2.4	Requerimientos de portabilidad	47
2.5.2.5	Otros requerimientos	48
2.6	DEFINICIÓN DE LOS CASOS DE USO.....	48
2.6.1	Definición de los actores.....	48
2.6.2	Listado de casos de uso	48
2.6.2.1	Analizador.....	48
2.6.2.2	Interfaz Web	48
2.6.3	Diagrama de casos de uso	54
2.6.4	Casos de uso expandidos	54
2.6.4.1	Analizador.....	54
2.6.4.2	Interfaz web	56
2.7	CONCLUSIONES	59
□	CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA.....	60
3.1	INTRODUCCIÓN	60
3.2	MODELO DE ANÁLISIS	60
3.2.1	Diagrama de clases del análisis	60
3.2.1.1	Interfaz Web: Módulo Reporte general	60
3.2.1.1.1	CU Reporte por URL.....	60
3.2.1.1.2	CU Reporte por Categoría	61
3.2.1.1.3	CU Reporte por IP	61
3.2.1.1.4	CU Reporte por tiempo	61
3.2.1.1.5	CU Reporte por usuario	61
3.2.1.1.6	CU Reporte por directorio de URL	62
3.2.1.1.7	CU Reporte sobre la navegación general	62
3.2.1.2	Interfaz Web: Módulo Reporte por usuario.....	63
3.2.1.2.1	CU Reporte de URLs por usuario	63
3.2.1.2.2	CU Reporte de categorías por usuario	63
3.2.1.2.3	CU Reporte de IPs por usuario	63
3.2.1.2.4	CU Reporte de Tiempo por usuario.....	63
3.2.1.2.5	CU Reporte de directorios de URL por usuario	64
3.2.1.2.1	CU Reporte de navegación general por usuario.....	64
3.2.2	MODELO DE DISEÑO	65
3.3.1	Symfony	65
3.3.1.1	Implementación del MVC por Symfony	65
3.3.1.2	Organización de la aplicación	66
3.3.2	Diagrama de clases del diseño	66
3.3.2.1	Interfaz Web: Módulo Reporte General.....	66
3.3.2.1.1	CU Reporte por URL.....	66
3.3.2.1.2	CU Reporte por Categoría	67
3.3.2.1.3	CU Reporte por IP	68
3.3.2.1.4	CU Reporte por tiempo	68
3.3.2.1.5	CU Reporte por usuario	69
3.3.2.1.6	CU Reporte por directorio de URL	69
3.3.2.1.7	CU Reporte sobre navegación general	70

3.3.2.2 Interfaz Web: Módulo Reporte por usuario.....	70
3.3.2.2.1 CU Reporte de URLs por usuario	70
3.3.2.2.2 CU Reporte de categorías por usuario	71
3.3.2.2.3 CU Reporte de IPs por usuario	71
3.3.2.2.4 CU Reporte de tiempo por usuario.....	72
3.3.2.2.5 CU Reporte de directorio de URL por usuario.....	72
3.3.2.2.6 CU Reporte de navegación general por usuario.....	73
3.3.2.3 Analizador.....	73
3.3.2.3.1 CU Analizar Log.....	73
3.3.3 Diagramas de secuencia	74
3.3.3.1 Interfaz Web	74
3.3.3.2 Analizador.....	75
3.3.4 Diseño de la base de datos	76
3.3.4.1 Diagrama de clases persistentes	76
3.3.4.2 Diagrama de entidad Relación	77
3.4 CONCLUSIONES	78
<input type="checkbox"/> CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBA.....	79
4.1 INTRODUCCIÓN	79
4.2 DIAGRAMA DE DESPLIEGUE	79
4.3 DIAGRAMAS DE COMPONENTES.....	79
4.3.1 Interfaz web.....	80
4.3.2 Analizador	83
4.4 PRUEBAS.....	84
4.4.1 Pruebas de caja negra	84
4.4.1.1 Partición equivalente	85
4.4.2 Casos de Prueba.....	86
4.4.2.1 Interfaz web.....	86
4.4.2.1.1 CU Reporte de categorías por usuario.....	86
4.4.2.1.2 CU Reporte de navegación general por usuario.....	88
4.4.2.1.3 CU Reporte por categorías	91
4.5 CONCLUSIONES	93
<input type="checkbox"/> CONCLUSIONES	94
<input type="checkbox"/> RECOMENDACIONES	95
<input type="checkbox"/> REFERENCIAS BIBLIOGRÁFICAS	96
<input type="checkbox"/> BIBLIOGRAFÍA	102
<input type="checkbox"/> ANEXOS 1	106
<input type="checkbox"/> ANEXOS 2.....	121

Índice de Tablas

TABLA 1 FORMATO COMÚN DE LOG	42
TABLA 2 FORMATO NATIVO DE LOG	43
TABLA 3 DEFINICIÓN DE LOS ACTORES	48
TABLA 4 DEFINICIÓN DEL CU ANALIZAR LOG.....	48
TABLA 5 DEFINICIÓN DEL CU REPORTE POR URL.....	49
TABLA 6 DEFINICIÓN DEL CU REPORTE POR CATEGORÍAS.....	49
TABLA 7 DEFINICIÓN DEL CU REPORTE POR TIEMPO	49
TABLA 8 DEFINICIÓN DEL CU REPORTE POR IP	50
TABLA 9 DEFINICIÓN DEL CU REPORTE POR USUARIO.....	50
TABLA 10 DEFINICIÓN DEL CU REPORTE POR DIRECTORIO DE LA URL.....	50
TABLA 12 DEFINICIÓN DEL CU REPORTE DE URLS POR USUARIO.....	51
TABLA 13 DEFINICIÓN DEL CU REPORTE DE CATEGORÍAS POR USUARIO	52
TABLA 14 DEFINICIÓN DEL CU REPORTE DE TIEMPO POR USUARIO.....	52
TABLA 15 DEFINICIÓN DEL CU REPORTE DE IPS POR USUARIO.....	52
TABLA 16 DEFINICIÓN DEL CU REPORTE DE DIRECTORIOS DE URLS POR USUARIO	53
TABLA 17 DEFINICIÓN DEL CU REPORTE DE NAVEGACIÓN GENERAL POR USUARIO.....	53
TABLA 18 DESCRIPCIÓN DEL CU ANALIZAR LOG.....	55
TABLA 19 DESCRIPCIÓN DEL CU REPORTES POR URL.....	57
TABLA 20 DESCRIPCIÓN DEL CU REPORTES DE URLS POR USUARIO.....	58
TABLA 21 PRUEBA DEL CP REPORTE DE CATEGORÍAS POR USUARIO	88
TABLA 22 PRUEBA DEL CP REPORTE DE NAVEGACIÓN GENERAL POR USUARIO.....	91
TABLA 23 PRUEBA DEL CP REPORTE POR CATEGORÍAS.....	93

Índice de Figuras

FIG. 1 USO DE INTERNET DESDE SUS ORÍGENES.....	14
FIG. 2 SERVIDOR PROXY.....	40
FIG. 3 MODELO DE DOMINIO DEL PROBLEMA.....	44
FIG. 4 DIAGRAMA DE CASOS DE USOS.....	54
FIG. 5 DIAGRAMA DE CLASE DE ANÁLISIS: CU REPORTE GENERAL POR URL.....	60
FIG. 6 DIAGRAMA DE CLASE DE ANÁLISIS: CU REPORTE GENERAL POR CATEGORÍA....	61
FIG. 7 DIAGRAMA DE CLASE DE ANÁLISIS: CU REPORTE GENERAL POR IP.....	61
FIG. 8 DIAGRAMA DE CLASE DE ANÁLISIS: CU REPORTE GENERAL POR TIEMPO.....	61
FIG. 9 DIAGRAMA DE CLASE DE ANÁLISIS: CU REPORTE GENERAL POR USUARIO.....	62
FIG. 10 DIAGRAMA DE CLASE DE ANÁLISIS: CU REPORTE GENERAL POR DIRECTORIO URL.....	62
FIG. 11 DIAGRAMA DE CLASE DE ANÁLISIS: CU REPORTE DE LA NAVEGACIÓN GENERAL	62
FIG. 12 DIAGRAMA DE CLASE DE ANÁLISIS: CU REPORTE DE URLS POR USUARIO.....	63
FIG. 13 DIAGRAMA DE CLASE DE ANÁLISIS: CU REPORTE DE CATEGORÍAS POR USUARIO.....	63
FIG. 15 DIAGRAMA DE CLASE DE ANÁLISIS: CU REPORTE DE TIEMPO POR USUARIO....	64
FIG. 16 DIAGRAMA DE CLASE DE ANÁLISIS: CU REPORTE DE DIRECTORIO DE URL POR USUARIO.....	64
FIG. 17 DIAGRAMA DE CLASE DE ANÁLISIS: CU REPORTE DE NAVEGACIÓN GENERAL POR USUARIO.....	64
FIG. 19 ORGANIZACIÓN DE LA APLICACIÓN.....	66
FIG. 20 DIAGRAMA DE CLASE DEL DISEÑO: CU REPORTE GENERAL POR URL.....	67
FIG. 21 DIAGRAMA DE CLASE DEL DISEÑO: CU REPORTE GENERAL POR CATEGORÍA..	67
FIG. 22 DIAGRAMA DE CLASE DEL DISEÑO: CU REPORTE GENERAL POR IP.....	68
FIG. 23 DIAGRAMA DE CLASE DEL DISEÑO: CU REPORTE GENERAL POR TIEMPO.....	68
FIG. 24 DIAGRAMA DE CLASE DEL DISEÑO: CU REPORTE GENERAL POR USUARIO.....	69
FIG. 25 DIAGRAMA DE CLASE DEL DISEÑO: CU REPORTE GENERAL POR DIRECTORIO DE URL.....	69
FIG. 26 DIAGRAMA DE CLASE DEL DISEÑO: CU REPORTE DE NAVEGACIÓN GENERAL..	70
FIG. 27 DIAGRAMA DE CLASE DEL DISEÑO: CU REPORTE DE URLS POR USUARIO.....	70
FIG. 28 DIAGRAMA DE CLASE DEL DISEÑO: CU REPORTE DE CATEGORÍAS POR USUARIO.....	71
FIG. 29 DIAGRAMA DE CLASE DEL DISEÑO: CU REPORTE DE IPS POR USUARIO.....	71
FIG. 30 DIAGRAMA DE CLASE DEL DISEÑO: CU REPORTE DE TIEMPO POR USUARIO....	72
FIG. 31 DIAGRAMA DE CLASE DEL DISEÑO: CU REPORTE DE DIRECTORIO DE URL POR USUARIO.....	72

FIG. 32 DIAGRAMA DE CLASE DEL DISEÑO: CU REPORTE DE DIRECTORIO DE URL POR USUARIO.....	73
FIG. 33 DIAGRAMA DE CLASE DEL DISEÑO: CU ANALIZAR LOG	73
FIG. 34 DIAGRAMA DE SECUENCIA: CU REPORTE DE CATEGORÍAS POR USUARIO	74
FIG. 35 DIAGRAMA DE SECUENCIA: CU ANALIZAR LOG	75
FIG. 36 DIAGRAMA DE CLASES PERSISTENTES.....	76
FIG. 37 DIAGRAMA DE ENTIDAD RELACIÓN.....	77
FIG. 38 DIAGRAMA DE DESPLIEGUE.....	79
FIG. 39 DIAGRAMA DE COMPONENTES GENERAL DE LA INTERFAZ WEB	80
FIG. 40 DIAGRAMA DE COMPONENTES PAQUETE VIEW DE LA INTERFAZ WEB	81
FIG. 41 DIAGRAMA DE COMPONENTES PAQUETE MODEL DE LA INTERFAZ WEB.....	81
FIG. 42 DIAGRAMA DE COMPONENTES PAQUETE CONTROLER DE LA INTERFAZ WEB ..	82
FIG. 43 DIAGRAMA DE COMPONENTES DEL ANALIZADOR.....	83

INTRODUCCIÓN

La Internet se ha convertido en uno de los mayores exponentes de la comunicación mundial. La conexión a la misma es disponer de muchos de los servicios y productos existentes, que van desde realizar compras on-line hasta reservar en un hotel a miles de kilómetros de distancia.

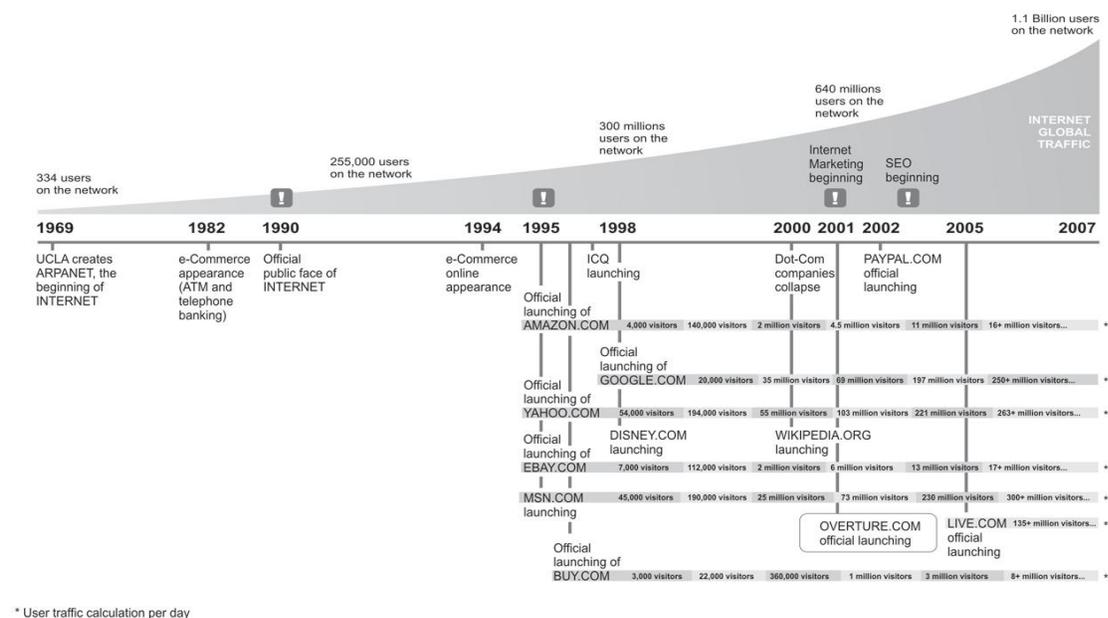


Fig. 1 Uso de Internet desde sus orígenes

La abundante información contenida en la misma ha puesto el conocimiento al alcance de un click. Internet ha propiciado además el surgimiento de nuevos estilos de trabajo basados en la colaboración al acortar las distancias entre los involucrados. Es por ello que casi la totalidad de los centros laborales cuentan con al menos una conexión a la red de redes.

Muchas son las ventajas que presenta la conexión a Internet pues fomenta la eficacia y la productividad en las empresas, pero su versatilidad puede en ocasiones volverse su principal desventaja cuando los trabajadores la utilizan en horario laboral para resolver asuntos personales tales como jugar, descargar música, chatear y concertar encuentros. Según

estudios de Optenet¹, empresa especializada en el filtraje de contenidos on-line, el 49% de las consultas realizadas a Internet en horario laboral no tienen que ver con su actividad profesional, sino con intereses personales [1]. El uso inadecuado de la red representa pérdidas exponenciales en cuanto a costo de conexión y pago de tiempo improductivo, sin analizar lo que representa el trabajo que debería realizar el obrero en el tiempo que emplea para navegar con fines ajenos a los objetivos de la institución. Las pérdidas anuales por este motivo en Estados Unidos ascienden los 178.000 millones de dólares según estudios realizados por el profesor Matthew C. Sonfield, del Departamento de Administración de la Universidad de Hofstra [2].

Cuba no se encuentra exenta de este problema, aunque no a tales niveles, debido a las peculiaridades con el acceso a Internet. En el país son escasas las conexiones a causa del recio bloqueo que mantiene Estados Unidos, trayendo como consecuencia que tengamos que utilizar un ancho de banda y una conexión de satélite de altos costos y muy lentos, 65 megabytes por segundo para la salida y 124 para la entrada [3]. A pesar de ello, se ha visto un desarrollo de la informática y las telecomunicaciones, pues ha crecido gradualmente el número de instituciones que cuentan con al menos una conexión a la red de redes.

Es en los centros educacionales, culturales y científicos donde se concentra en Cuba el uso de Internet y cualquier nueva tecnología para mayor desarrollo económico y beneficio social. Por consiguiente es aquí donde se encuentra la casi totalidad de los usuarios con acceso a la misma.

En el país no existen estudios profundos sobre el uso de la red por parte de los trabajadores en sus centros laborales, pero análisis realizados por el Instituto Central de Investigación Digital arrojaron que de 3379 sitios visitados en un trimestre solo el 48,57% estaban relacionados directamente con el trabajo [4]. Estos datos confirman las tendencias internacionales de emplear parte del horario laboral para navegar con fines personales. Por tal motivo se impone la necesidad de controlar dichas conexiones, uno de los métodos para lograrlo es a través de un servidor proxy.

¹ Optenet es una compañía global de seguridad informática con más de 10 años en el mercado. Reconocida internacionalmente por su tecnología de filtrado de contenidos por Internet, ha expandido su negocio hasta posicionarse como un proveedor líder de soluciones integradas de seguridad para proveedores de servicios gestionados de seguridad y grandes empresas.

Un servidor proxy “es un sistema intermediario entre ordenadores internos de una red y los servidores de Internet de forma tal que reciba las peticiones de unos y se las pase a los otros con previa verificación de accesos y privilegios [5].

Los servidores proxy guardan los registros de la navegación. Cada vez que un usuario realiza una petición toda la actividad queda registrada en archivos denominados log, que no son más que ficheros texto que almacenan líneas y líneas de información, que abarcan la dirección IP que se realizó la petición, el usuario, la página visitada y la cantidad de megas transferidos como datos más significativos [6]. Un análisis de estos ficheros permitiría conocer a ciencia cierta cuál fue el uso de la conexión, que tan costosa y restringida suele ser para el país.

Las instituciones muchas veces no pueden llevar un control de dichas conexiones, por no contar con una herramienta lo suficientemente eficaz que le permita realizar un análisis efectivo de los registros de navegación; además resulta prácticamente imposible realizar de forma manual el análisis de estos registros debido a la gran cantidad de información que pueden almacenar, llegando a alcanzar decenas de megabytes. Al país, se le dificulta la obtención de productos que ayuden a realizar este análisis, pues la gran mayoría son propietarios, por lo que resultan muy costosos y los que existen bajo licencias de software libre, no presentan todas las funcionalidades requeridas.

Después de observar la situación anterior, se determinó como **problema científico**: la inexistencia de un sistema para el análisis estadístico y por categorías de navegación de los registros de servidores proxy en el país.

El **objeto de estudio** de la investigación es el proceso de análisis de los registros generados por los servidores y el **campo de acción** en el cual se desarrolla la misma es el proceso de análisis de los registros generados por los servidores proxy.

El **objetivo general** de la investigación es el desarrollo de un sistema automatizado que permita realizar análisis estadísticos y por categorías de navegación de los registros de servidores proxy.

Para la realización de esta herramienta, se trazaron como **objetivos específicos**:

- Analizar las herramientas existentes de análisis de los registros de servidores proxy.
- Caracterizar el sistema.

- Analizar y diseñar el sistema.
- Implementar y probar el sistema.

El presente trabajo se basa en la siguiente **idea a defender**, el desarrollo de un sistema automatizado para el análisis estadístico y por categorías de navegación de los registros de servidores proxy mejorará el proceso de control sobre las conexiones a Internet en Cuba.

Para dar cumplimiento a los objetivos antes planteados, se proponen las siguientes **tareas de investigación**:

- Realizar un estudio de los diferentes sistemas de análisis de registros para servidores proxy existentes tanto propietarios como de software libre, dentro y fuera del país.
- Realizar un estudio de las herramientas de desarrollo y lenguaje de programación a utilizar en la solución.
- Realizar un estudio del funcionamiento de los servidores proxy e identificar que información almacenan los log generados.
- Definir las funcionalidades que debe cumplir el sistema.
- Definir la estructura de la base de datos.
- Investigar los tipos de pruebas existentes.
- Definir los casos de prueba.

Para darle cumplimiento a las tareas de investigación será necesario utilizar **métodos científicos de investigación**. Entre los métodos que guiarán el desarrollo de la investigación se encuentra el método **Histórico – Lógico**, pues es necesario realizar un estudio previo de las herramientas existentes y cómo han evolucionado, tomándolas como punto de referencia y comparación de posibles resultados. El **Analítico – Sintético** pues se hará énfasis en el análisis de la documentación existente acerca del tema, con el objetivo de extraer los elementos más importantes para procesar la información y elaborar conclusiones para una mayor utilidad en el desarrollo del trabajo y en el momento de proponer una solución acertada. La **Modelación** ya que mediante el lenguaje de modelado UML se reflejará la estructura, relaciones internas y características de la solución a través de diagramas.

El presente documento consta de cuatro capítulos:

Capítulo 1 “Fundamentación teórica” realiza un análisis de otras soluciones existentes nacional e internacionalmente, y describe las herramientas, metodologías y lenguajes de programación a utilizar.

Capitulo 2 “Características del sistema” realiza una descripción del problema, muestra el funcionamiento de un servidor proxy y la información que almacenan los log, se definen los requisitos del sistema y finalmente se propone una solución a la problemática planteada.

Capitulo 3 “Análisis y diseño del sistema” se realiza el análisis y diseño del sistema y la confección de la base de datos.

Capitulo 4 “Implementación y prueba del sistema” presenta la implementación de las clases del diseño y se realizan pruebas al sistema.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

La realización de este capítulo tiene como objetivo abordar la fundamentación teórica del tema propuesto a través del estudio y análisis de soluciones existentes así como las tendencias actuales. En primer lugar se ofrece una panorámica de las herramientas más utilizadas a nivel internacional viendo de cada una sus principales características y procesos además de algunas ventajas y desventajas que puedan presentar. Posteriormente se presentarán las soluciones que han adoptado las empresas en el país, a partir de la respuesta que le han dado algunos desarrolladores a esta situación. Finalmente se exponen aspectos sobre las tecnologías y la metodología seleccionada para el desarrollo de la aplicación.

1.2 Analizadores de registro de servidores proxy a nivel internacional.

En el mercado actual del software existen varios analizadores de registros de servidores proxy de gran prestigio y con distintas funcionalidades. Algunos de ellos realizan además análisis de los registros de servidores web. Dentro de las aplicaciones más usadas a nivel mundial se encuentran:

1.2.1 AWStats



AWStats es una poderosa herramienta que genera estadísticas avanzadas gráficamente de los servidores proxy, web, streaming, ftp o correo. Muestra toda la información posible contenida en los registros en forma de gráficas a través de páginas web. Puede analizar además los registros de Log extendido de SafeSquid² y generar reportes de estadísticas mensuales, semanales, diarios, listas de usuarios, tipos de archivos, URL³ de las páginas, sistemas

² Servidor proxy que provee mecanismos para llevar el control del contenido de Internet en una red

³ (Uniform Resource Locator) Utilizado para especificar un objeto en Internet. Puede ser un fichero, grupo de news, gopher, etc.

operativos y navegadores usados, búsqueda de palabras y frases claves y los códigos de estado del protocolo HTTP. [7]

También permite realizar personalizaciones para generar informes especiales, como las amenazas detectadas por el antivirus, las solicitudes bloqueadas por diversos filtros (Filtro de URL, la lista negra de URL, filtro de palabras clave, filtro de Mime (extensiones multipropósito de correo Internet), Filtro de Imagen) y el usuario. Esta herramienta está construida bajo la licencia GPL⁴. Su actualización de estadísticas se puede realizar desde un navegador web. [8]

1.2.2 Webalizer



Esta aplicación permite realizar análisis estadísticos del uso de Internet basándose en los log del servidor proxy Squid o servidor web Apache. Webalizer genera gráficos en páginas web de las estadísticas extraídas que se pueden utilizar para interpretar fácilmente el uso de estos servidores. Está escrito en el lenguaje de programación C y es extremadamente rápido, puede procesar un archivo con unas 70 000 entradas en un segundo. Una de sus características más interesantes es que soporta archivos de registros de tamaño ilimitado [9].

Este programa no sólo realiza reportes cuantitativos, sino que los muestra además en forma de gráficos, haciéndolos más elegantes y sencillos de observar. Esta herramienta está construida bajo licencia GPL. [10]

A pesar de tener todas estas facilidades, este software presenta algunos inconvenientes. Uno de ellos es que no está orientado a usuarios. Este programa es adecuado para llevar estadísticas generales para empresas e instituciones, para llevar el monitoreo de sitios en específicos, sin embargo no es funcional para mostrar informaciones de los usuarios durante su navegación. Sus reportes no son del todo eficientes para una persona que le pudiera interesar más los sitios específicos que se han visitado y qué usuarios lo han visitado, así como qué

⁴ General Public License: es una licencia creada por la Free Software Foundation orientada principalmente a proteger la libre distribución, modificación y uso del software. Tiene como propósito declarar que el software cubierto por esta licencia es libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

transferencias se han hecho y no estadísticas generales del comportamiento de la navegación en la Web.

1.2.3 Calamaris



Calamaris, está implementado para analizar los log generados por el servidor proxy Squid. Es capaz de generar tanto informes en texto plano como en páginas HTML⁵, con las estadísticas que nos interesen ordenadas por tablas. Existen versiones que generan gráficos de estas estadísticas. Aunque podemos usar Webalizer para este propósito, los informes de Squid son más sencillos de configurar y más detallados. Esta herramienta está construida bajo licencia GPL. [11]

Este analizador es muy útil para especialistas de red por la gran cantidad de información que puede mostrar, aun así sus reportes son muy generales, enfocados a la administración de la navegación sin realizar un análisis detallado del mismo.

1.2.4 Sarg



Generador de Reportes de Análisis de Squid, es una herramienta que le permite llevar un control de la navegación que realizan los usuarios por Internet. Este producto proporciona mucha información acerca de las actividades de los usuarios del proxy: tiempo, bytes, sitios, etc. [12]. Al contrario de otras herramientas como el Calamaris, este software, construye una estadística específica de usuario. Esta herramienta está construida bajo licencia GPL.

1.2.5 Squid-graph



El Squid-graph es una herramienta libre, sencilla pero potente, versión 2 del nativo Calamaris. Herramienta de análisis de log que genera informes con la representación gráfica del tráfico del

⁵ Lenguaje de Marcas de Hipertexto, utilizado para la creación de documentos de hipertexto e hipermedia

servidor proxy. Desarrollado en lenguaje Perl, es una herramienta multiplataforma que lee el archivo access.log de su servidor proxy Squid y genera una página Web que muestra las estadísticas acerca de su proxy, accesos y transferencias, incluyendo el número de visitas de caché y el porcentaje de solicitudes que fueron atendidas por la misma. La desventaja que presenta es que si se requiere alguna otra funcionalidad necesitará la línea de comandos para interactuar en lugar de utilizar la web. Para su uso es necesario tener instalado un intérprete del lenguaje Perl y la librería perl-GD. Además requiere el acceso al fichero de Squid access.log y al nombre del directorio donde una web obtenga y procese de forma gráfica el fichero log. [15]

1.2.6 WebTrends Professional



La versión 5.5 fue el primer producto del mercado que integró el análisis de ficheros "log" con el análisis de servidores Proxy, análisis de enlaces, control de calidad, monitorización y aviso de Webs, y otras herramientas de administración de servidores Web. La versión 3 de WebTrends Professional añade los módulos de alerta mejorados a los módulos ya existentes: el módulo de análisis de tráfico y ficheros "log", el módulo de análisis de servidores Proxy, y el módulo de control de calidad y enlaces.

Este módulo monitoriza múltiples dispositivos y avisa en el caso de problemas mediante un e-mail o mensaje acústico. La herramienta está construida bajo licencia propietaria [16].

Este producto está especializado en todo tipo de reportes referentes a la navegación de un sitio web, a diferencia de los anteriormente mencionados, a pesar de que en esta versión ya analiza algunos servidores proxy.

1.2.7 Sawmill



Sawmill es una potente herramienta de análisis jerárquico de reportes, optimizado para reportes Web pero en su proceso es capaz de soportar otros 818 formatos de log. Además de procesar log de formatos combinados y ampliados genera estadísticas de los mismos,

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 [13].

Permite la creación de diferentes perfiles en los que se especifica que información se presentara en el reporte. 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 log.

Presenta una serie de características que lo hacen uno de los sistemas más fuertes de su tipo, y son la facilidad de uso que tiene, además de ser altamente configurable y procesar casi cualquier tipo de reporte, presenta un avanzado rastreo de usuarios y muestra las estadísticas en formas de reportes y gráficos en línea. En versiones más avanzadas permite el filtrado de log además del control sobre la apariencia de las páginas de estadísticas. Esta herramienta está construida bajo licencia propietaria haciendo costoso su mantenimiento y actualización. [14]

Las herramientas internacionales analizadas poseen un alto grado de profesionalismo y fiabilidad en cuanto al análisis de registros pero no se adaptan a las necesidades del país, pues no presentan reportes relacionados con la clasificación de las URLs por categorías. Todas estas herramientas han aportado experiencias por la cantidad de reportes generados y la forma de mostrarlos.

1.3 Analizadores de registro de servidores proxy a nivel nacional.

En Cuba no existe un amplio desarrollo de herramientas dedicadas al análisis de registros de servidores proxy, las existentes son producto de la necesidad de algunas instituciones de controlar la navegación por Internet ya que los software existentes en el mercado mundial no se adaptaban a sus necesidades y vieron en sí la capacidad de elaborar sus propios analizadores de log.

1.3.1 ISAWeb



Producto realizado en el Instituto Central de Investigación Digital (ICID) que desarrolló su parte fundamental en ASP.NET. El software permite que un usuario autenticado pueda visualizar la información de su navegación por Internet. Este software analiza los ficheros log generados por servidores Proxy Isa Server. Solo la información relevante para los usuarios es extraída de estos ficheros y transformada en una base de datos desarrollada en SQL Server, para ello se crearon aplicaciones de consola que se ejecutan como tareas del sistema en los servidores de Internet y de datos. Este software, al ser una aplicación Web, puede ser consultada por cualquier usuario desde cualquier terminal de la red. Las bases de datos son centralizadas, y el proceso de inserción de datos es independiente a la aplicación principal. Lo más importante: se establece una jerarquía que permite que la labor de seguridad informática se descentralice entre todos los directivos, e incluso participe el trabajador. [17]

1.3.2 AAInternet



Software desarrollado por Segurmática⁶ capaz de extraer información de los ficheros log de los servidores Proxy y generar un reporte personalizado sobre la actividad del uso de Internet, donde puede observarse la fecha y hora de visita al sitio, la transferencia por usuarios y los dominios o sitios Web visitados, entre otros datos. Sus versiones están desarrolladas para funcionar sobre Windows pero difieren en dependencia de los log del servidor Proxy que está destinado a analizar. Tiene tres versiones, una para analizar la información de acceso generada por el servidor Proxy Squid, otra para Isa Server y la otra para Winroute. Utiliza una base de datos creada a partir del procesamiento de archivos log del servidor Proxy. De ahí que se hace necesario copiar manualmente hacia la estación donde está instalada esta herramienta

⁶ Empresa cubana dedicada a la consultoría y seguridad informática

los archivos log que se deseen procesar o a una estación dentro de la red de su empresa donde sean accesibles estos archivos desde la estación donde se encuentra instalado AAInternet. [18].

Cuando se analizan los log la base de datos se encuentra lista para ofrecer información, sobre el comportamiento de los usuarios en la Web. Algunas de estas consultas son:

- Sitios Visitados por Usuario: muestra los sitios en la WEB visitados por cada usuario autorizado dentro de su empresa.
- Sitios Visitados por Estación (Dirección IP): permite conocer los sitios visitados por cada estación de trabajo, apoyándose en la dirección IP.
- Páginas Visitadas por Usuario: informa las páginas visitadas en la WEB por cada usuario autorizado dentro de su empresa.
- Direcciones IP NO Autenticadas: descubre las estaciones de trabajo dentro de su empresa que están haciendo uso del WEB sin control de autenticación

Además puede construir reportes gráficos de análisis de la navegación de los usuarios. Ejemplo de ellos son:

- Transferencia por Usuario: reporta aquellos usuarios que más suben y descargan archivos desde la Web.
- Transferencia por Dirección IP: reporta los ordenadores que más información mueven desde y hacia la Web.
- Sitios más Visitados: como su nombre lo indica lista aquellos sitios más visitados de la empresa.

1.3.3 SICC

Software desarrollado por el Centro Universitario José Antonio Echeverría (CUJAE) que mezcla análisis de log y contabilidad con el fin de traducir al usuario su navegación en unidades monetarias. La conversión es realizada en dependencia de la clasificación del sitio al que se

accede y el horario de acceso al mismo. Los resúmenes de información contable son enviados a una base de datos centralizada en PostgreSQL ubicada en el servidor de contabilidad y los detalles de la navegación son conservados en ficheros log de la aplicación. El consumo de la navegación es almacenado en una estructura de ficheros serializados en lenguaje de programación Python. [19]

1.3.4 SRNI



El Sistema de Reportes de la Navegación por Internet (SRNI) es un software implementado en la Universidad de las Ciencias Informáticas. Es una aplicación web desarrollada bajo tecnología Java que utiliza como gestor de base de datos PostgreSQL. Al ser web puede consultarse desde cualquier lugar de la Universidad. El sistema proporciona a los usuarios reportes dinámicos de su navegación, que son creados a partir de las trazas del servidor proxy. Se puede consultar la información de accesos por URL, dirección IP, días y horas. El sistema sólo permite escoger un rango máximo de 30 días y la información que brinda puede llegar a tener hasta 24 horas de retraso.

Tras el estudio de las herramientas nacionales dedicadas al análisis de registros de servidores proxy es posible afirmar que ninguna cumple con el objetivo general de la investigación, pues no presentan reportes relacionados con la clasificación de las URLs por categorías, la mayoría de ellas son construidas de acuerdo con las necesidades específicas de un centro lo cual limita la cantidad reportes aunque los hace muy potentes como es el caso de SRNI y SICC.

1.4 Lenguajes de programación

1.4.1 Perl

Perl (*Practical Extraction and Report Language*) es un lenguaje que se ha usado para realizar script debido a las facilidades que brinda en la manipulación de cadenas de caracteres, archivos y procesos, que se ve simplificada por el importante número de operadores que pone a disposición del usuario. Es más rápido que la mayoría de los lenguajes interpretados ya que los programas se ejecutan como una sucesión de instrucciones. Además es un lenguaje que se distribuye de forma gratuita y se puede utilizar en cualquier plataforma con solo tener el intérprete instalado [20].

Sus características más importantes son: [21]

- Multiplataforma: está disponible de forma compilada para (Unix, Linux, MacOSX, Win32, OS2, etc).
- Potente en la manipulación de textos: sistema de reconocimiento de patrones y operadores de sustitución.
- Potente en la manipulación de ficheros: fácil manejo de funciones básicas de apertura, lectura y escritura, que soportan además ficheros binarios.
- Robusto: incorpora facilidades para inspeccionar y capturar errores, incluyendo señales UNIX (SIGINT, SIGTERM, SIGKILL).
- Depurador incorporado: puede trazar instrucción a instrucción los programas.
- Gestión de memoria: al igual que en Java, Perl gestiona por sí solo la memoria utilizada, y la libera cuando es necesario con la ayuda del Garbage collector (recolector de basura).
- Desarrollo rápido de aplicaciones: debido a la existencia de una amplia red de servidores conectados a través de CPAN⁷ con miles de módulos y ejemplos disponibles, todos clasificados en categorías con una instalación estándar y calidad probada.

1.4.2 PHP

PHP (acrónimo de: Hypertext Preprocessor") es un lenguaje de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Está orientado al desarrollo de aplicaciones web y permite insertar contenidos dinámicos en ellas. Hoy en día existe una gran comunidad de desarrolladores que continuamente implementan mejoras y nuevas funcionalidades para facilitar su uso [22].

Sus características más significativas son:

⁷ CPAN: Comprehensive Perl Archive Network según sus siglas, es el repositorio de módulos y documentación de Perl.

- Multiplataforma: Puede ejecutarse en cualquier plataforma (Linux, Unix, Solaris y OpenBSD, Mac OS, Windows, etc).
- Simplicidad: usuarios con experiencia en los lenguajes de programación Perl, C o C++ podrán utilizarlo rápidamente.
- Velocidad: alta velocidad de ejecución sin introducir demoras en la máquina, bajo consumo de recursos y muy buena integración con Apache.
- Estabilidad: ninguna aplicación está completamente libre de errores (bugs), pero con una amplia comunidad de programadores y usuarios, corregirlos es mucho más fácil. Posee un sofisticado manejo de variables que lo hacen muy robusto y estable.
- Seguridad: permite la protección contra diversos ataques a través de diferentes niveles de seguridad.
- Documentación: Tiene una de las comunidades más grandes en Internet, por lo que no es complicado encontrar ayuda, documentación, artículos, noticias, y más recursos.
- Open Source.⁸

1.4.2.1 Php 5

Con las primeras versiones de PHP, PHP 3 y PHP 4, se había conseguido una plataforma potente y estable para la programación de páginas del lado del servidor, sin embargo los objetos eran solamente colecciones de datos y funciones, agrupados en clases, las cuales soportaban herencia simple donde los métodos y atributos solo tenían visibilidad pública (no existía ni visibilidad privada, ni protegida) y los objetos se pasaban como parámetros por valor y no por referencia.

En PHP 5 el sistema para el tratamiento de objetos ha sido reescrito completamente, permitiendo un mejor rendimiento y nuevas características como. [23]

- Visibilidad pública, protegida y privada en métodos y atributos.

⁸ Término utilizado por algunos miembros de la comunidad del software libre, creando una nueva forma de distribución de software, permitiendo a los usuarios leer, modificar y redistribuir el código fuente de un programa.

- Clases y métodos abstractos.
- Uso de interfaces para soportar herencia múltiple.
- Permite restringir el tipo de los parámetros recibidos en los métodos (solo se puede restringir por clases, no por tipos de datos primitivos).
- Creación de métodos y variables estáticas.
- Los parámetros son pasados por referencia en los métodos.
- Soporte para el manejo de excepciones.

1.4.3 JavaScript

JavaScript fue diseñado como un lenguaje de elaboración de scripts que pudiera incrustarse en archivos HTML para brindar mayor dinamismo a las páginas web. No requiere compilación, porque es interpretado por el navegador, es fácil de utilizar y su sintaxis es semejante a la de los lenguajes Java y C.

Entre sus principales características encontramos: [24]

- Fácil de aprender: no es necesario tener amplios conocimientos de programación para poder hacer un programa.
- Usable: es el lenguaje más utilizado en la Web debido a que maneja objetos dentro de la página y sobre ellos puede definir eventos cuya respuesta se realiza en tiempo real.
- Rápido: los scripts pueden desarrollarse en un período de tiempo relativamente corto.
- Reduce la carga del servidor: se encarga de las validaciones del lado del cliente y otras funciones de las cuales se encargaba el servidor.
- Es independiente de la plataforma de hardware o sistema operativo, siempre y cuando exista un navegador con soporte JavaScript.

Limitaciones:

- El número de métodos integrados es insuficiente para gestionar documentos y ventanas.
- El código del script debe descargarse completamente antes de poderse ejecutar.
- No tiene protección del código, ya que se descarga en texto plano.

1.5 Servidor de Base de Datos

1.5.1 PostgreSQL

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (Object-Relational Database Management System (ORDBMS)), desarrollado en el Berkeley Computer Science Department Postgres y liberado bajo la licencia BSD⁹, permitiendo disponer de su código fuente, modificarlo a voluntad y redistribuirlo libremente. Presenta alta concurrencia, para esto utiliza la tecnología de Control de Concurrencia Multi-Versión (MVCC), con lo que logra que ningún lector sea bloqueado por un escritor. Soporta una gran parte del estándar SQL y ofrece muchas características modernas como:

- Consultas complejas
- Llaves foráneas
- Integridad transaccional
- Control de concurrencia multiversión.

Es altamente extensible, soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario [25].

Entre sus características más importantes encontramos [26]:

- Es multiplataforma, ha sido desarrollado para plataformas (Linux, Unix, BSDs, Mac OS, Beos, Windows, etc).

⁹ La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD(Berkeley Software Distribution)

- Presenta documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios.
- Presenta soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl, Python, etc.
- Presenta soporte de todas las características de una base de datos profesional (disparadores (triggers), procedimientos almacenados, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, vistas materializadas, etc.).
- Presenta soporte de protocolo de comunicación encriptado por SSL.
- Presenta una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones.

Además puede ser utilizado, modificado y distribuido por todo el mundo de forma gratuita para cualquier fin, ya sea privado, comercial o académico.

1.6 Servidor Web

1.6.1 Apache

Apache es un servidor Web gratuito desarrollado por el Apache Server Project¹⁰ cuyo objetivo es la creación de un servidor web fiable, eficiente y fácilmente extensible con código fuente abierto gratuito. Entre sus características se destacan [27]:

- Multiplataforma, ha sido desarrollado para plataformas (Unix, Linux, MacOSX, Vms, Win32, OS2, etc).
- Su desarrollo ha sido de acuerdo al protocolo HTTP/1.1 normalizado por el W3C (WWW Consortium).

¹⁰ Proyecto dedicado a mantener un servidor HTTP libre, multiplataforma y de código abierto.

- Modular, puede ser adaptado a diferentes entornos y necesidades, de acuerdo con los módulos de apoyo que proporciona.
- Facilita la integración como "plug-ins" de lenguajes de programación de páginas web dinámicas.
- Brinda soporte para bases de datos, el protocolo de comunicación segura SSL, lenguajes de programación Perl y PHP y demás.
- Open Source [28].

1.7 Framework

1.7.1 Symfony

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes, haciendo uso de buenas prácticas de programación. Proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener y facilita la implementación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas [29].

Symfony es un framework diseñado para optimizar el desarrollo de las aplicaciones web. Algunas de sus características son: [30]

- Está escrito en PHP5 basándose en el patrón Modelo Vista Controlador que separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web.
- Proporciona herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.
- Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft.
- Es multiplataforma, se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

- Cuenta con una excelente documentación en español que incluye la traducción del libro “The Definitive Guide to Symfony” que posee 19 capítulos en los que se explica todo lo relativo al trabajo con la versión 1.0.
- Numerosos plugins son creados y actualizados constantemente ante las necesidades de los desarrolladores.

1.8 Metodología de desarrollo de software.

Una metodología no es más que un proceso de software detallado y completo. Existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Todo desarrollo de software es riesgoso y difícil de controlar por lo que es necesario aplicar una metodología que permita obtener como resultado un producto de calidad. Las metodologías guían el proceso de desarrollo y la clave del éxito de un proyecto, además conducen al programador a desarrollar un buen sistema de software, logrando el crecimiento de la calidad y la productividad del producto. La elección de la metodología adecuada es más importante que utilizar las mejores y más potentes herramientas [31].

1.8.1 Rational Unified Process (RUP)

El Proceso Racional Unificado o RUP es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicaciones, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. Además junto con el Lenguaje Unificado de Modelado (UML) constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Este proceso se centra en su carácter dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. [32]

Esta metodología unifica todo el equipo de desarrollo de software y optimiza su comunicación proveyendo a cada miembro de una aproximación al desarrollo del producto con una base de conocimiento en línea adaptable de acuerdo a las necesidades específicas del proyecto y accesible desde cualquier estación de trabajo. La base de conocimiento unifica aún más al equipo identificando y asignando responsabilidades, artefactos y tareas de forma que cada miembro conozca cual es su contribución al proyecto. Unificando al grupo de desarrollo, se simplifica la comunicación, asegurando la asignación de recursos en forma eficiente, la entrega de los artefactos correctos, y el cumplimiento de los tiempos límite. [33].

RUP divide en 4 fases el desarrollo de software:

- Inicio: El objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.
- Construcción: En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- Transición: El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Define 3 elementos fundamentales:

- Actividades: Procesos que se determinan en cada iteración.
- Trabajadores: Personas o entes involucradas en cada proceso.
- Artefactos: Puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. [34]

1.8.2 Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML) prescribe un conjunto de notaciones y diagramas estándares para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real, ayudando al usuario a entender la realidad de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades en proyectos que no estén seguros en su desarrollo, reduciendo el coste y el tiempo empleado en la construcción de las piezas que constituirán el modelo [35].

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. [36]

Ofrece nueve diagramas en los cuales modelar sistemas: [37]

- Diagramas de Casos de Uso para modelar los procesos.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

Permite la modelación de sistemas con tecnología orientada a objetos y describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema

1.9 Herramientas a utilizar

1.9.1 Eclipse

Eclipse es un entorno de desarrollo integrado (IDE) de código abierto, que permite desarrollar software profesional sin presentar complicaciones. Se ejecuta sobre la máquina virtual de Java y está escrito en gran parte sobre este lenguaje, típicamente ha sido usado para desarrollar IDEs, como el de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ¹¹) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Una de las características principales que tiene es la expansibilidad, propiedad que le permite adaptarse a muchos lenguajes de programación a partir de la utilización de pequeños plugins o módulos que se incluyen a la herramienta para el desarrollo en otros lenguajes de programación o bien para agregar otros tipos de funcionalidades [38].

El empleo de módulos proporciona toda una funcionalidad al frente de la plataforma de cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de plugins es una plataforma ligera para componentes de software. Permite trabajar además con lenguajes para procesamiento de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos. La arquitectura plugins le facilita escribir cualquier extensión deseada en el ambiente, como sería la Gestión de la configuración. [39]

Desarrollado originalmente por IBM ¹² como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios [40].

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project¹³, cubriendo casi todas las áreas de Model Driven Engineering¹⁴.

¹¹ Versión autónoma del compilador Java Eclipse

¹² Empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática.

¹³ Proyecto de Modelado de Eclipse se centra en la evolución y promoción de modelos de desarrollo basados en las tecnologías dentro de la comunidad Eclipse, proporcionando un conjunto unificado de modelado de los marcos, herramientas, aplicaciones y normas.

Esta plataforma de desarrollo, apoyado por una extensa comunidad y con licencia que facilita el desarrollo, brinda una gran ayuda a los programadores. Su sistema basado en plugins es innovador y extensible, disponible para prácticamente todas las plataformas, es totalmente gratis por lo que lo convierte en una herramienta ideal para desarrollo [41].

1.9.2 Visual Paradigm

El Visual Paradigm para UML es una poderosa herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite crear tipos diferentes de diagramas de clases, código inverso, generar código desde diagramas y generar documentación ayudando a una más rápida construcción de aplicaciones de mayor calidad y a un menor costo. Es muy sencillo de usar, fácil de instalar y actualizar. Es nombrada por muchas bibliografías como la herramienta CASE por excelencia del software libre. Esta herramienta ofrece [42]:

- Un entorno de creación de diagramas para UML.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas.

Posibilita la representación gráfica de los diagramas permitiendo ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clase,

¹⁴ Es una metodología de desarrollo de software que se centra en la creación de modelos.

actividad, entre otros. Además, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, también permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico. Brinda la posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir del código [43]. Entre los lenguajes conocidos para la ingeniería inversa, se encuentran los siguientes:

- Java.
- C++.
- PHP 5.0.
- XML.

1.10 Conclusiones

En el presente capítulo se ha hecho referencia a las principales herramientas de análisis de servidores proxy tanto a nivel mundial como en Cuba, con el objetivo de tomar lo positivo de las herramientas que han implantado su primacía internacional así como aquellas locales que han surgido de la innovación y adaptadas a las características del país. Se ha hecho referencia además a los lenguajes de programación a utilizar en el desarrollo de la aplicación, las herramientas necesarias y la metodología a utilizar para la implementación de la solución propuesta.

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En el presente capítulo se realiza la descripción del problema que da lugar a la investigación, se explica el funcionamiento del servidor proxy así como la información que almacenan los log generados por el servidor proxy Squid. Además se muestra la propuesta de solución a la problemática antes planteada y detalles de la misma mediante los requisitos funcionales y la descripción de los casos de uso.

2.2 Descripción del problema

Cuba desde hace unos años ha mostrado un incremento de las tecnologías y con ello el número de usuarios conectados a la red. La gran mayoría de estos usuarios están concentrados en instituciones del estado. El acceso a internet en el país es sumamente costoso y restringido por lo que se hace necesario explotarlo al máximo en función de los objetivos de la organización.

La navegación en las instituciones del país se realiza a través de servidores Proxy ya que estos garantizan que todas las conexiones a internet se hagan de manera controlada, guardando registros de la navegación en ficheros textos denominados log. El análisis de estos ficheros es una tarea sumamente compleja ya que pueden almacenar decenas de megabytes de información y realizarlo manualmente es prácticamente imposible.

En el capítulo anterior se presentó la necesidad de desarrollar una herramienta para el análisis de estos log. Por lo que surgen las siguientes interrogantes: ¿Qué es un servidor Proxy? ¿Qué información se guarda en los ficheros log? Para dar respuesta a las mismas es necesario profundizar en el estudio de dicho servidor.

2.2.1 Servidor Proxy

Un servidor proxy es un intermediario para algún servicio de red, que consiste en permitir a los clientes realizar conexiones indirectas hacia otros servicios. Cuando un cliente navega a través de un proxy, cada petición que hace el navegador se delega al propio servidor proxy y este es el encargado de acceder al servicio solicitado.

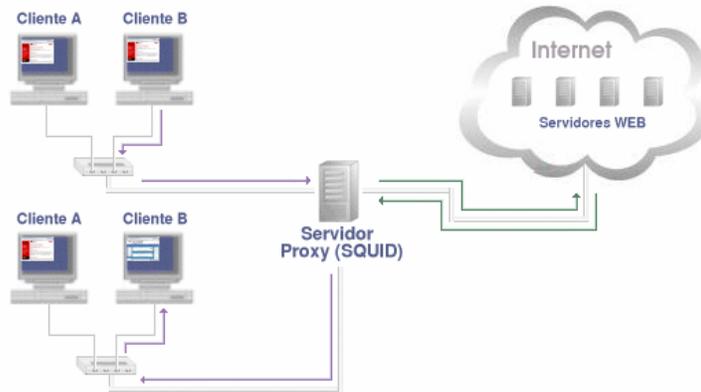


Fig. 2 Servidor proxy

Al estar en medio de este tráfico puede realizar dos funciones muy importantes: controlar los accesos (permite o deniega según se disponga en sus normas) y hacer caché de peticiones de elementos (páginas web, imágenes, etc.) en un espacio de disco de longitud variable [44].

Algunas de las ventajas más considerables son [45]:

- **Control:** Como actúa de intermediario es el que 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) es el equipado para hacer el trabajo real.
- **Velocidad:** Si varios usuarios realizan peticiones del mismo recurso, el servidor proxy puede hacer caché de los elementos, accediendo solo una vez al destino disminuyendo el tiempo de respuesta y ahorrando ancho de banda.
- **Filtrado:** El proxy puede negarse a responder algunas peticiones si detecta que están prohibidas.
- **Modificación:** Como intermediario que es, un proxy puede cambiar la información o modificarla siguiendo un algoritmo.

2.2.2 Log del servidor proxy Squid

2.2.2.1 ¿Por qué Squid?

Squid es uno de los proxy más usados en la actualidad principalmente en los sistemas operativos basados en UNIX. Es multiplataforma, muy confiable, robusto y versátil. Es de licencia GPL, por lo que además de brindar acceso al código fuente, está excepto del pago de costosas licencias por su uso o con restricción a determinado número de usuarios.

2.2.2.2 ¿Qué información almacenan los log?

Los registros generados por el servidor son una valiosa fuente de información sobre el volumen de trabajo y el rendimiento del mismo ya que guardan información de las peticiones de los usuarios, los errores de configuración del sistema y el consumo de recursos.

El documento se centrará en el acces.log ya que es el fichero que registra los datos de la navegación de los usuarios por el servidor. Este log se genera básicamente en dos formatos “nativo” y “común”.

El formato común es utilizado por numerosos servidores HTTP. Este formato se compone de los siguientes siete campos: [46]

Posición	Campo	Significado
1	Remotehost	Dirección IP o nombre del host del cliente.
2	rfc931	El nombre de usuario del cliente asociado a la conexión
3	Authuser	Siempre es “-” para los log de Squid
4	date	La fecha de conexión en formato yyyy/mm/dd :hh:mm:ss
5	method	El método de la petición HTTP (GET, POST, etc.)
5	URL	La dirección URL de la petición

6	status	Código de estado HTTP asociado a la petición
7	Bytes	Los bytes transferidos asociado a la petición

Tabla 1 Formato común de log

El formato nativo es:

Posición	Campo	Significado
1	time	El Timestamp de Unix (desde el 1 de enero de 1970) con resolución de milisegundos.
2	elapsed	El tiempo transcurrido en milisegundos desde que el cliente envía la petición hasta que recibe respuesta a la misma.
3	remotehost	Dirección IP o nombre del host del cliente.
4	code	El resultado de la caché para la petición.
4	status	Código de estado HTTP asociado a la petición.
5	bytes	Los bytes transferidos asociados a la petición.
6	method	El método de la petición HTTP (GET, POST, etc.).

7	URL	La dirección URL de la petición.
8	rfc931	El nombre de usuario del cliente asociado a la conexión.
9	peerstatus	Un código de estado que explica la forma en que la solicitud fue tramitada, ya sea de la caché, de la cache de un servidor vecino o del servidor origen.
9	peerhost	El host desde el cual fue tramitada la solicitud.

Tabla 2 Formato nativo de log

2.3 Modelo de dominio

Un modelo de dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema, teniendo como ventaja el permitir ayudar a los usuarios, clientes y desarrolladores a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema.

Las clases del dominio aparecen en tres formas típicas:

- Objetos del negocio que representan “cosas” que se manipulan en el negocio.
- Objetos del mundo real y conceptos de los que el sistema debe hacer un seguimiento.
- Sucesos que ocurrirán o han ocurrido.

El modelo de dominio se describe mediante diagramas UML. Estos muestran las clases del dominio y como se relacionan entre ellos mediante asociaciones.

Haciendo un análisis del problema se llega a la conclusión de que no es necesaria la realización de un Modelo de Negocio, siendo suficiente un Modelo de Dominio debido a la

simplicidad que presenta el sistema y el conocimiento presentado acerca del funcionamiento del mismo.

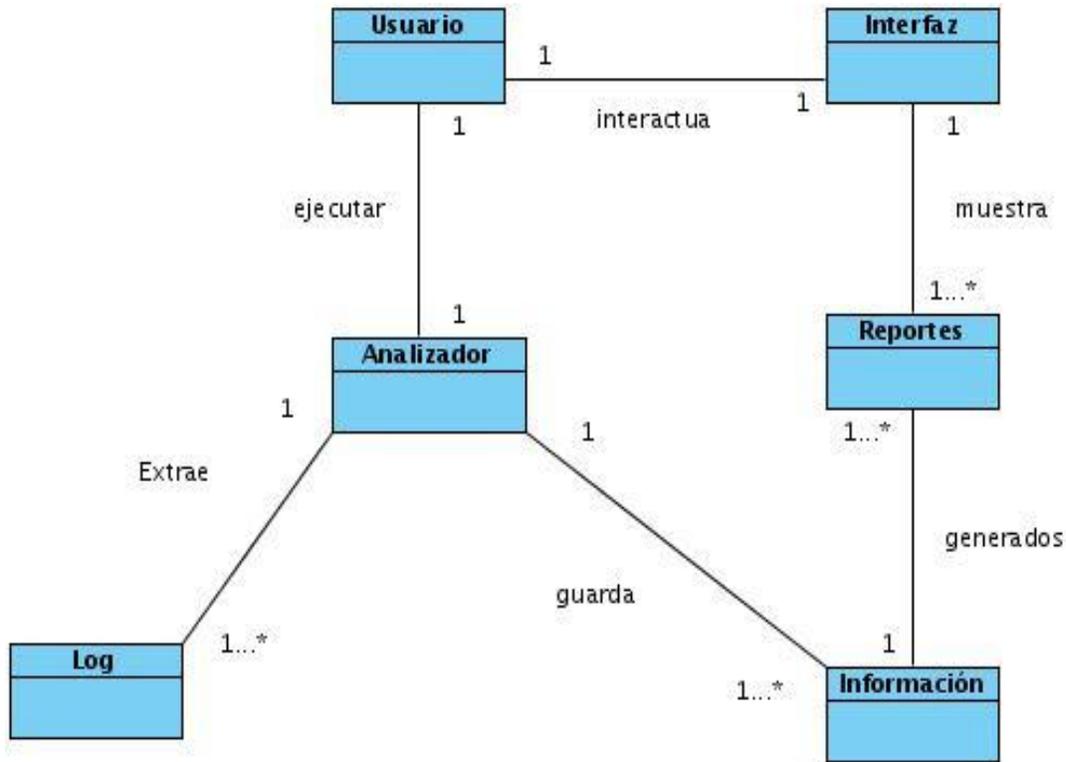


Fig. 3 Modelo de dominio del problema

Un usuario ejecuta el analizador que sería el encargado de extraer la información del log, transformarla y guardarla como información útil y organizada. El usuario además interactúa con una interfaz que muestra los reportes que son generados a partir de la información guardada previamente por el analizador del log.

2.4 Solución Propuesta

La respuesta para dar solución a la problemática planteada consiste en elaborar una aplicación que conste de tres partes fundamentales:

- El analizador, un programa escrito en Perl cuya funcionalidad será cargar el log, interpretarlo según el formato especificado, extraer la información, transformarla y guardarla en la base de datos.
- La base de datos almacenará la información interpretada por el analizador.
- La interfaz web que será la encargada de elaborar los reportes a partir de la información contenida en la base de datos y mostrar la información obtenida en forma de gráficas y tablas.

Los requisitos funcionales y no funcionales que se presentan a continuación abordan con mayor detalle la propuesta de solución

2.5 Especificación de requisitos del sistema

En el proceso de producción de software son fundamentales tener bien claros los requerimientos del sistema, con el objetivo de tener definida una línea de lo que se desea desarrollar. La captura de estas especificaciones permite y asegura un sistema con eficiencia y calidad. Estas se pueden obtener utilizando la descripción de las condiciones y/o capacidades que el producto debe cumplir, debiendo ser lo suficientemente abarcadoras como para que se puedan llegar a acuerdos entre los clientes y los desarrolladores en cuanto a lo que debe o no hacer el software.

2.5.1 Requisitos funcionales

La solución propuesta tomó las URLs categorizadas de acuerdo a su contenido de la base de datos de Filpacon.

Filpacon es un software desarrollado en la universidad cuyo funcionamiento parte de una base de datos que posee gran cantidad de URLs clasificadas que ante la solicitud de un recurso de internet por parte de un usuario al servidor se analizan los parámetros (usuario, IP de origen y

URL) y según características de usuario y la categoría de la URL se toma la decisión de permitir o denegar la petición.

2.5.1.1 Interfaz web

R1. Mostrar reporte general por URL.

R2. Mostrar reporte general por categorías.

R3. Mostrar reporte general por tiempo (hora, día, día de semana, mes, año).

R4. Mostrar reporte general por IP.

R5. Mostrar reporte general por usuario.

R6. Mostrar reporte general por directorio de URL.

R7. Mostrar reporte de datos generales de la navegación.

R8. Mostrar reporte general de URL por usuario.

R9. Mostrar reporte general de categorías por usuario.

R10. Mostrar reporte general de tiempo por usuario.

R11. Mostrar reporte general de IP por usuario.

R12. Mostrar reporte general de directorio de URL por usuario.

R13. Mostrar reporte de datos generales de la navegación por usuario.

2.5.1.2 Analizador

R14. Extraer información del log.

R15. Interpretar información del log.

R16. Guardar información interpretada.

2.5.2 Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son los aspectos del sistema visibles para el usuario, que no están relacionados de forma directa con el comportamiento funcional, sin embargo imponen condiciones sobre los requerimientos funcionales como velocidad, tiempo de respuesta y uso de la memoria. Existen múltiples categorías para clasificar a los requerimientos no funcionales, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta para el sistema propuesto.

2.5.2.1 Requerimientos de software

Las computadoras de los usuarios necesitan tener un navegador web para poder acceder a la interfaz de la aplicación, ya sea desde sistema operativo Windows o Linux. Para poder realizar la compilación del analizador y analizar los ficheros log, debe tener instalado cualquier intérprete del lenguaje Perl, ya sea para uno o para otro sistema operativo.

2.5.2.2 Requerimientos de hardware

Para un óptimo funcionamiento del analizador del sistema debe contar con una computadora que contenga como mínimo 512 MB de RAM. Para la base de datos es necesario contar con un mínimo de 30 GB de capacidad de disco duro.

2.5.2.3 Requerimientos de usabilidad

La aplicación debe cumplir con los principales principios de usabilidad, debe permitir el trabajo a cualquier tipo de usuario, tenga o no conocimientos previos en informática, así como comodidad al acceder a las diferentes funcionalidades que brinda. Debe tener una interfaz amigable y fácil de entender y manejar.

2.5.2.4 Requerimientos de portabilidad

Al sistema se podrá acceder desde cualquier estación de trabajo de la red local y desde cualquier plataforma para ver los reportes.

2.5.2.5 Otros requerimientos

Uno de los requerimientos que debe cumplir es tener acceso a los log. Estos registros de acceso (access.log) deben ser copiados en la carpeta requerida por el sistema (log) con el formato requerido para su análisis.

2.6 Definición de los casos de uso

2.6.1 Definición de los actores

Actores	Descripción
Usuario	Persona que ejecutará el analizador de log y observará los reportes en la aplicación.

Tabla 3 Definición de los actores

2.6.2 Listado de casos de uso

2.6.2.1 Analizador

Caso de uso	Analizar Log
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea ejecutar el analizador de log. El analizador carga el log del directorio especificado, interpreta cada línea de acuerdo al formato y guarda la información.
Referencia	R14, R15, R16

Tabla 4 Definición del CU Analizar log

2.6.2.2 Interfaz Web

Caso de uso	Reporte por URL.
Actores	Usuario

Resumen	El caso de uso se inicia cuando el usuario desea obtener el reporte general por URL en un espacio de tiempo definido.
Referencia	R1

Tabla 5 Definición del CU Reporte por URL

Caso de uso	Reporte por categorías.
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea obtener el reporte general por categorías en un espacio de tiempo definido.
Referencia	R2

Tabla 6 Definición del CU Reporte por Categorías

Caso de uso	Reporte por tiempo
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea obtener el reporte general por tiempo (por: hora, día, día de semana, mes, año) en un espacio de tiempo definido.
Referencia	R3

Tabla 7 Definición del CU Reporte por Tiempo

Caso de uso	Reporte por IP.
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea obtener el reporte general por IP en un espacio de tiempo definido.
Referencia	R4

Tabla 8 Definición del CU Reporte por IP

Caso de uso	Reporte por usuario
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea obtener el reporte general por usuarios en un espacio de tiempo definido.
Referencia	R5

Tabla 9 Definición del CU Reporte por usuario

Caso de uso	Reporte por directorio de URL.
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea obtener el reporte general por directorio de URL en un espacio de tiempo definido.
Referencia	R6

Tabla 10 Definición del CU Reporte por directorio de la URL

Caso de uso	Reporte sobre navegación general
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea obtener el reporte de datos generales de la navegación en un espacio de tiempo definido.
Referencia	R7

Tabla 11 Definición del CU Reporte sobre navegación general

Caso de uso	Reporte de URLs por usuario.
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea obtener el reporte general por URLs en un espacio de tiempo definido y para un usuario seleccionado.
Referencia	R8

Tabla 12 Definición del CU Reporte de URLs por usuario

Caso de uso	Reporte de categorías por usuario
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea obtener el reporte general por categorías en un espacio de tiempo definido y para un usuario seleccionado.

Referencia	R9
------------	----

Tabla 13 Definición del CU Reporte de categorías por usuario

Caso de uso	Reporte de tiempo por usuario.
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea obtener el reporte general por tiempo (por: hora, día, día de semana, mes, año) en un espacio de tiempo definido y para un usuario seleccionado.
Referencia	R10

Tabla 14 Definición del CU Reporte de tiempo por usuario

Caso de uso	Reporte de IPs por usuario.
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea obtener el reporte general por IPs en un espacio de tiempo definido y para un usuario seleccionado.
Referencia	R11

Tabla 15 Definición del CU Reporte de IPs por usuario

Caso de uso	Reporte de directorios de URLs por usuario
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea obtener el reporte general por directorio de URL en un espacio de tiempo definido y para un usuario seleccionado.
Referencia	R12

Tabla 16 Definición del CU Reporte de directorios de URLs por usuario

Caso de uso	Reporte de navegación general por usuario
Actores	Usuario
Resumen	El caso de uso se inicia cuando el usuario desea obtener el reporte de datos generales de la navegación en un espacio de tiempo definido y para un usuario seleccionado.
Referencia	R13

Tabla 17 Definición del CU Reporte de navegación general por usuario

2.6.3 Diagrama de casos de uso

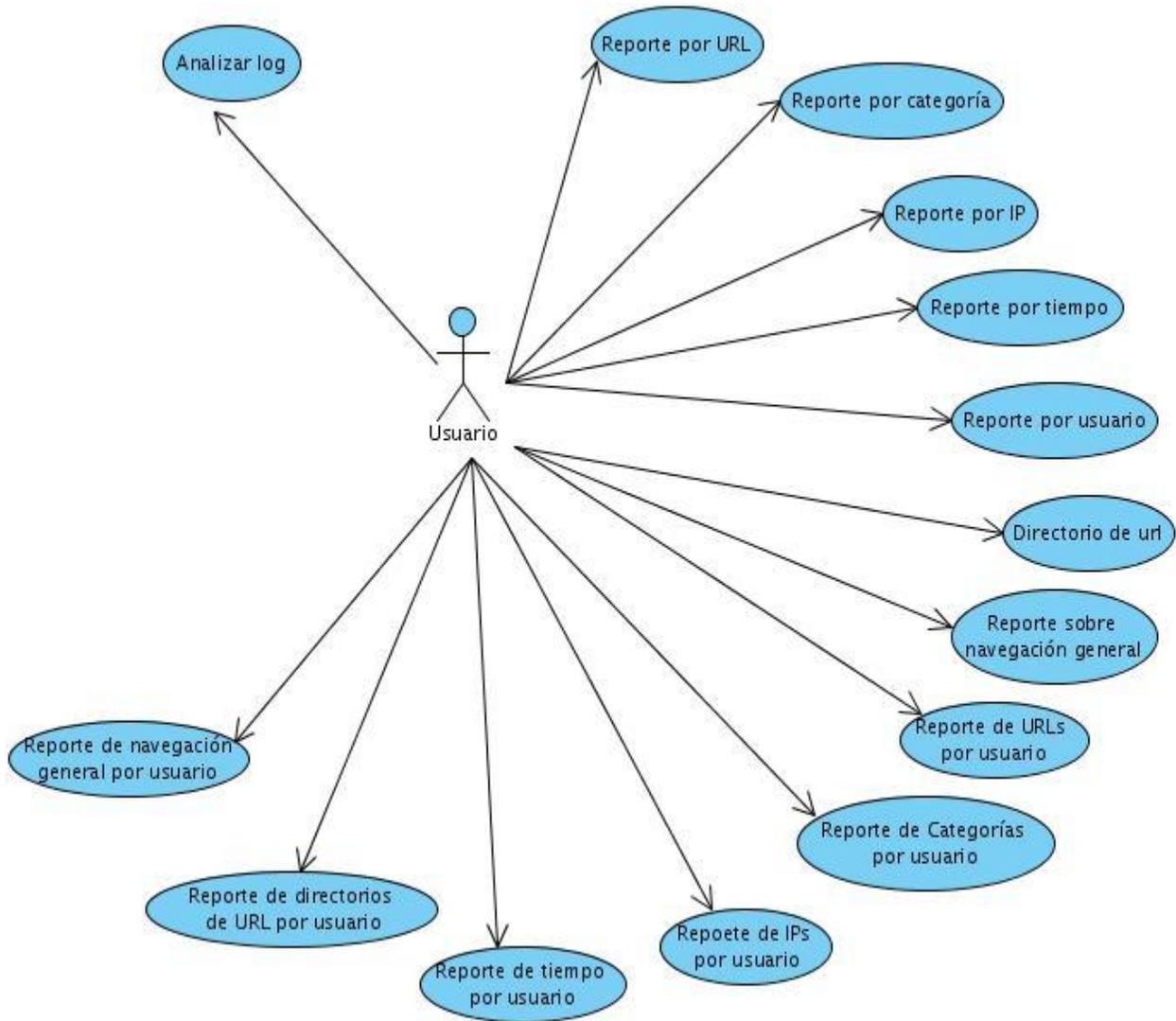


Fig. 4 Diagrama de casos de usos

2.6.4 Casos de uso expandidos

2.6.4.1 Analizador

Caso de uso:	Analizar log
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario desea ejecutar el

	analizador de log.	
Referencia:	R14, R15, R16	
CU asociados:		
Precondiciones		
Flujo normal de eventos		
Sección "Analizar Log"		
Acción del actor	Respuesta del sistema	
1. El usuario ejecuta el analizador	1.1 El analizador carga el log de la dirección especificada en los ficheros de configuración. 1.2 El analizador extrae la información del log. 1.3 El analizador interpreta la información obtenida. 1.4 El analizador almacena la información interpretada.	
Poscondiciones:	En la base de datos quedará la información extraída del log.	
Prioridad:	Crítico	
Especificaciones:		
Complementarias:		

Tabla 18 Descripción del CU Analizar log

2.6.4.2 Interfaz web

Caso de uso:	Reporte por URL	
Actores:	Usuario	
Resumen:	El caso de uso se inicia cuando el usuario desea observar el reporte por URL.	
Referencia:	R1	
CU asociados:	Analizar log	
Precondiciones	Que se halla ejecutado el caso de uso Analizar log.	
Flujo normal de eventos		
Sección "Reportes por URL"		
	Acción del actor	Respuesta del sistema
	<p>1. El usuario selecciona el reporte "Reporte por URL".</p> <p>2. El usuario selecciona los datos.</p>	<p>1.1 El sistema muestra un formulario en el que pide seleccionar los datos del reporte (fecha inicio, fecha fin y modo de ordenamiento).</p> <p>2.1 Si el reporte devuelve datos</p> <p>2.2 El sistema muestra reporte en forma de gráfica y más detallado en forma de tabla.</p>
Flujo Alternativo		
		2.1.a Se emite un mensaje informando que no hay reportes para la fecha seleccionada y

	regresa a la acción 2 de esta sección.
Poscondiciones:	El usuario observará el reporte en forma de gráfica y más detallado en forma de tabla
Prioridad:	Secundario
Especificaciones:	
Complementarias:	

Tabla 19 Descripción del CU Reportes por URL

Caso de uso:	Reporte de URLs por usuario
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario desea observar el reporte de URLs por usuario.
Referencia:	R8
CU asociados:	Analizar log
Precondiciones	Que se halla ejecutado el caso de uso Analizar log.
Flujo normal de eventos	
Sección "Reportes de URLs por usuario"	
Acción del actor	Respuesta del sistema
1. El usuario selecciona el reporte	1.1 El sistema muestra un formulario en el

<p>“Reporte de URLs por usuario”.</p> <p>2. El usuario llena los datos.</p>	<p>que pide llenar los datos del reporte (usuario, fecha inicio, fecha fin y modo de ordenamiento).</p> <p>2.1 Se verifica que los campos obligatorios estén llenos (usuario) y que exista en la base de.</p> <p>2.2 Si el reporte devuelve datos</p> <p>2.3 El sistema muestra reporte en forma de gráfica y más detallado en forma de tabla.</p>
<p>Flujo alternativo</p>	
	<p>2.1.a Se emite un mensaje para que llene los campos obligatorios y regresa a la acción 2 de esta sección.</p> <p>2.2.a Se emite un mensaje informando que no hay reportes para ese usuario en la fecha especificada y regresa a la acción 2 de esta sección.</p>
<p>Poscondiciones:</p>	<p>El usuario observará el reporte de URL por usuario en forma de gráfica y más detallado en forma de tabla</p>
<p>Prioridad:</p>	<p>Secundario</p>
<p>Especificaciones:</p>	
<p>Complementarias:</p>	

Tabla 20 Descripción del CU Reportes de URLs por usuario

La descripción de los restantes Casos de Uso se encuentra en el anexo 1.

2.7 Conclusiones

En el capítulo se mostró una visión general de la problemática que dio lugar a la investigación y se representó mediante un modelo de dominio ya que es el adecuado cuando existe una familiarización con los conceptos y procesos principales. Se elaboró además la propuesta de solución, se identificaron los requisitos funcionales y no funcionales que tiene que cumplir el sistema, así como una descripción de los casos de uso que servirá como base para fases posteriores.

CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

En el presente capítulo quedará reflejado el análisis y diseño de la aplicación mediante la metodología RUP, mostrando cómo se transforman los requisitos en el diseño del sistema ofreciendo una descripción detallada para la futura implementación. Además queda definido el modelo de datos que conforma la base de datos.

3.2 Modelo de análisis

El modelo de análisis persigue como objetivos fundamentales lograr una comprensión más exacta de los requisitos del sistema, describir lo que requiere el cliente y establecer una aproximación a lo que sería el modelo de diseño.

3.2.1 Diagrama de clases del análisis

3.2.1.1 Interfaz Web: Módulo Reporte general

3.2.1.1.1 CU Reporte por URL

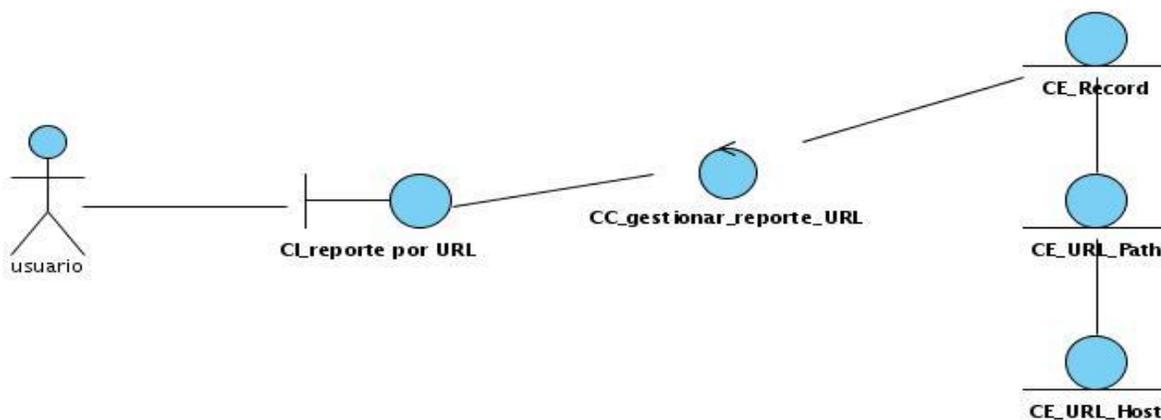


Fig. 5 Diagrama de clase de análisis: CU Reporte por URL

3.2.1.1.2 CU Reporte por Categoría

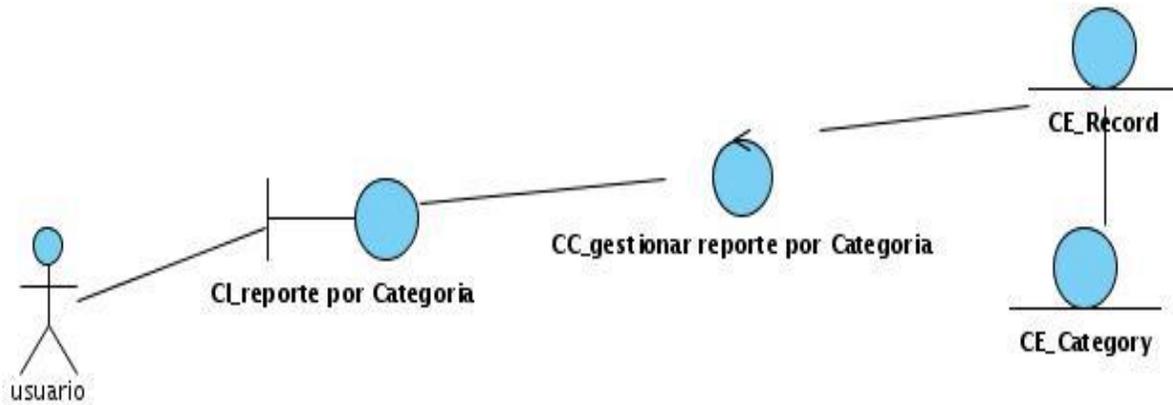


Fig. 6 Diagrama de clase de análisis: CU Reporte por Categoría

3.2.1.1.3 CU Reporte por IP

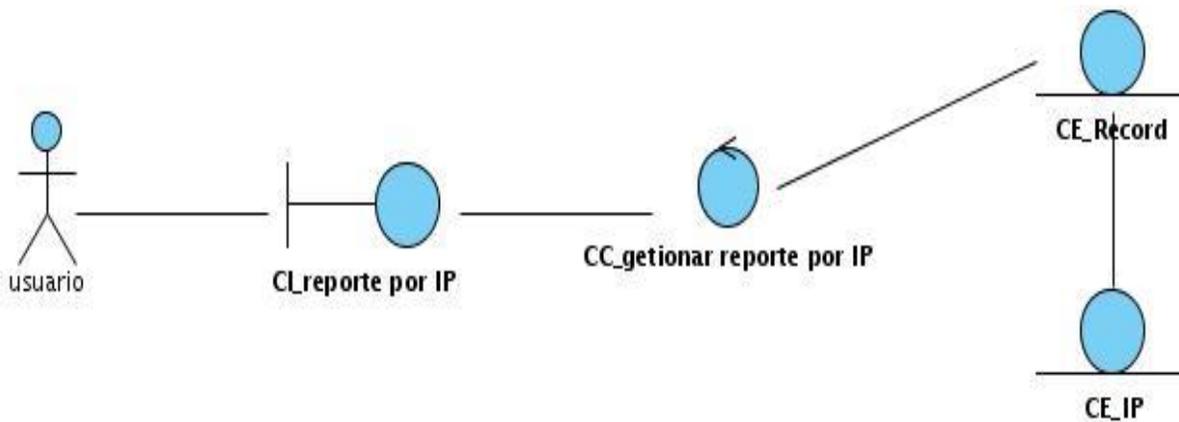


Fig. 7 Diagrama de clase de análisis: CU Reporte por IP

3.2.1.1.4 CU Reporte por tiempo

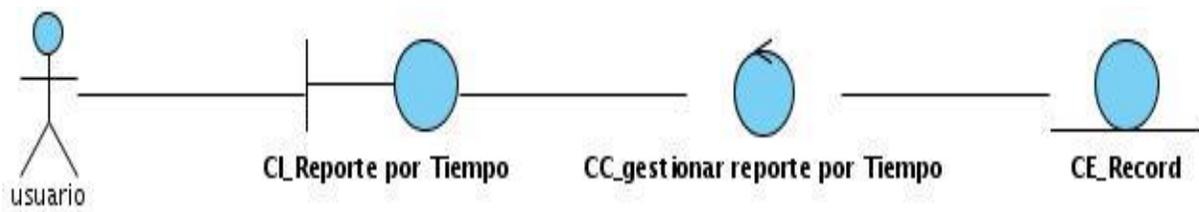


Fig. 8 Diagrama de clase de análisis: CU Reporte por Tiempo

3.2.1.1.5 CU Reporte por usuario

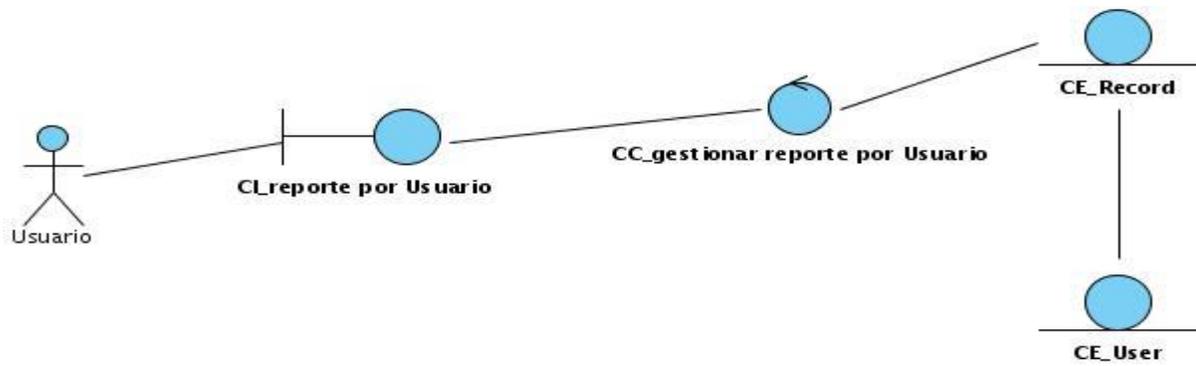


Fig. 9 Diagrama de clase de análisis: CU Reporte por Usuario

3.2.1.1.6 CU Reporte por directorio de URL



Fig. 10 Diagrama de clase de análisis: CU Reporte por directorio de URL

3.2.1.1.7 CU Reporte sobre la navegación general



Fig. 11 Diagrama de clase de análisis: CU Reporte sobre la navegación general

3.2.1.2 Interfaz Web: Módulo Reporte por usuario

3.2.1.2.1 CU Reporte de URLs por usuario

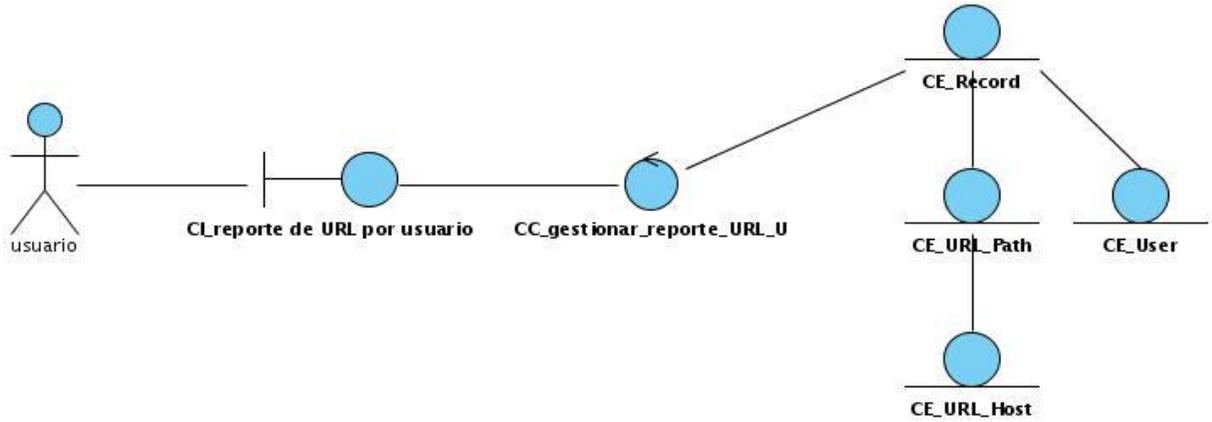


Fig. 12 Diagrama de clase de análisis: CU Reporte de URLs por usuario

3.2.1.2.2 CU Reporte de categorías por usuario

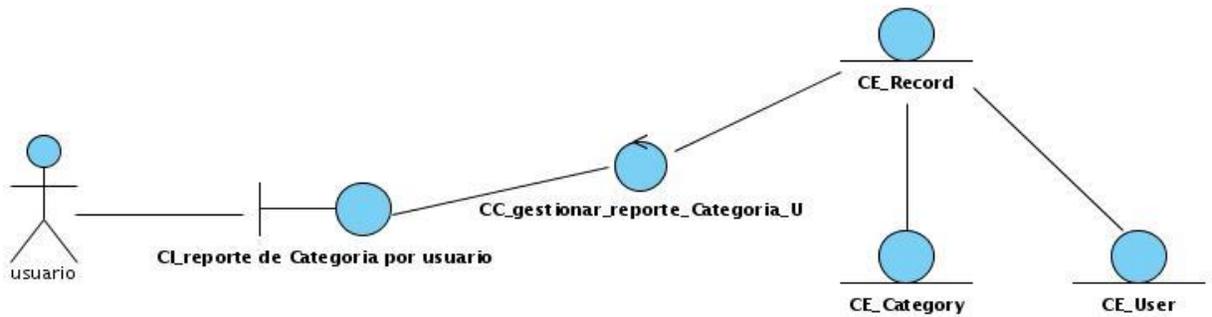


Fig. 13 Diagrama de clase de análisis: CU Reporte de categorías por usuario

3.2.1.2.3 CU Reporte de IPs por usuario

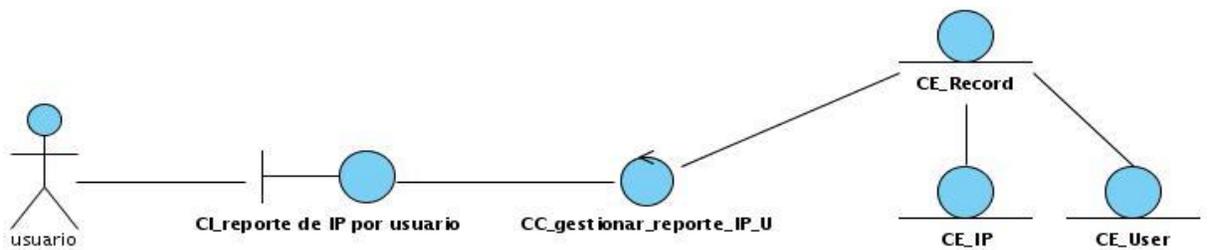


Fig. 14: Diagrama de clase del análisis: CU Reporte de IPs por usuario

3.2.1.2.4 CU Reporte de Tiempo por usuario

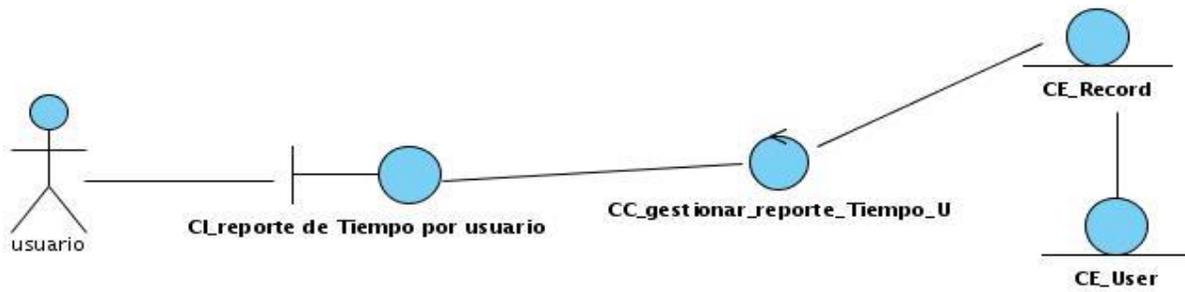


Fig. 15 Diagrama de clase de análisis: CU Reporte de Tiempo por usuario

3.2.1.2.5 CU Reporte de directorios de URL por usuario

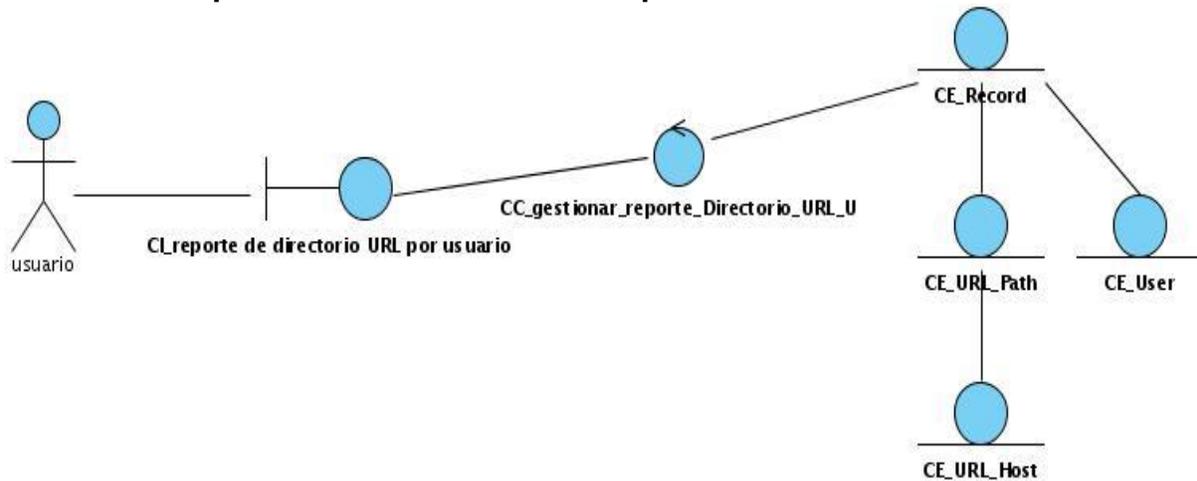


Fig. 16 Diagrama de clase de análisis: CU Reporte de directorios de URL por usuario

3.2.1.2.1 CU Reporte de navegación general por usuario

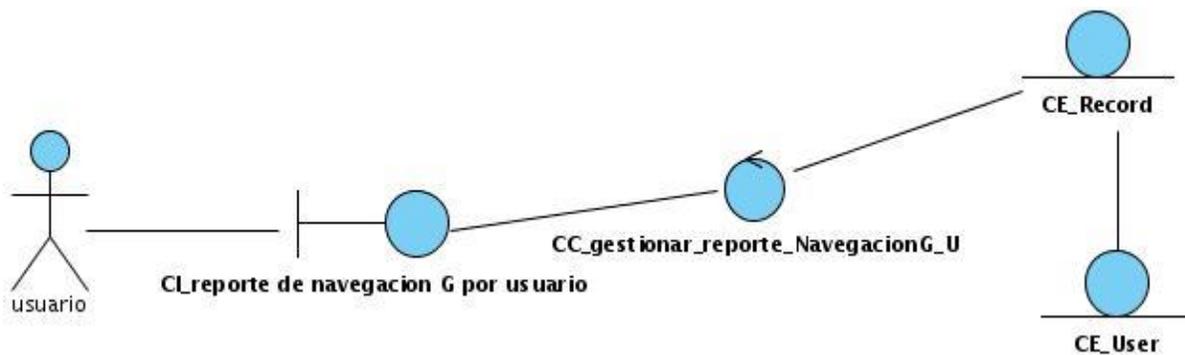


Fig. 17 Diagrama de clase de análisis: CU Reporte de navegación general por usuario

3.3 Modelo de diseño

El Modelo de Diseño, es un modelo de objetos que describe la realización física de los Casos de Uso, centrándose en como los requisitos funcionales y no funcionales tienen impacto en el sistema a desarrollar. El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción.

3.3.1. Symfony

Symfony es un framework diseñado para optimizar el desarrollo de las aplicaciones web, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación [29].

3.3.1.1 Implementación del MVC por Symfony

Symfony ha sido escrito completamente en PHP5 con el objetivo de aprovechar todas las ventajas de esta versión del lenguaje y se basa en el patrón arquitectónico Modelo-Vista-Controlador (MVC), implementándolo de modo que el desarrollo de aplicaciones sea rápido y sencillo [29].

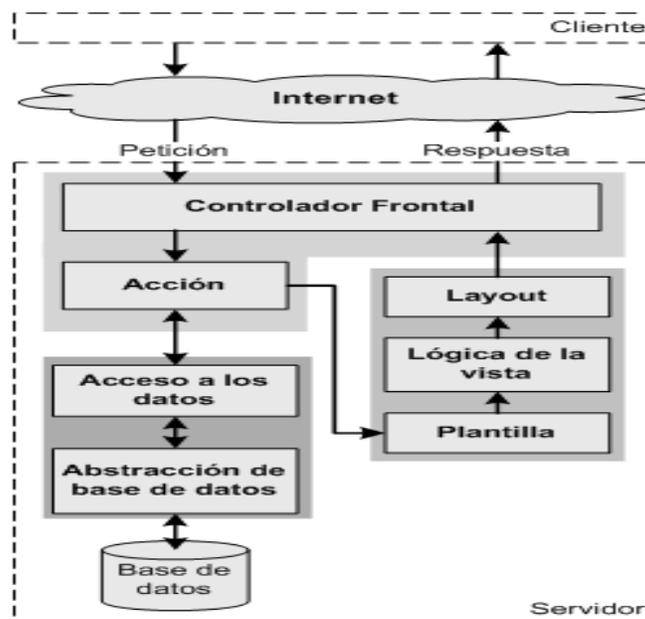


Fig. 18: Implementación del patrón MVC según Symfony

3.3.1.2 Organización de la aplicación

En Symfony un proyecto se considera como un conjunto de servicios y operaciones disponibles bajo un determinado nombre de dominio que comparten el mismo modelo de objetos. Dentro de un proyecto, las operaciones se agrupan de forma lógica en aplicaciones y cada aplicación está formada por uno o más módulos. Un módulo normalmente representa a una página Web o un grupo de páginas con un propósito relacionado y almacenan las acciones, que representan cada una de las operaciones que se pueden realizar [29]. La interfaz web seguiría una estructura aproximada a la que muestra la figura 19.

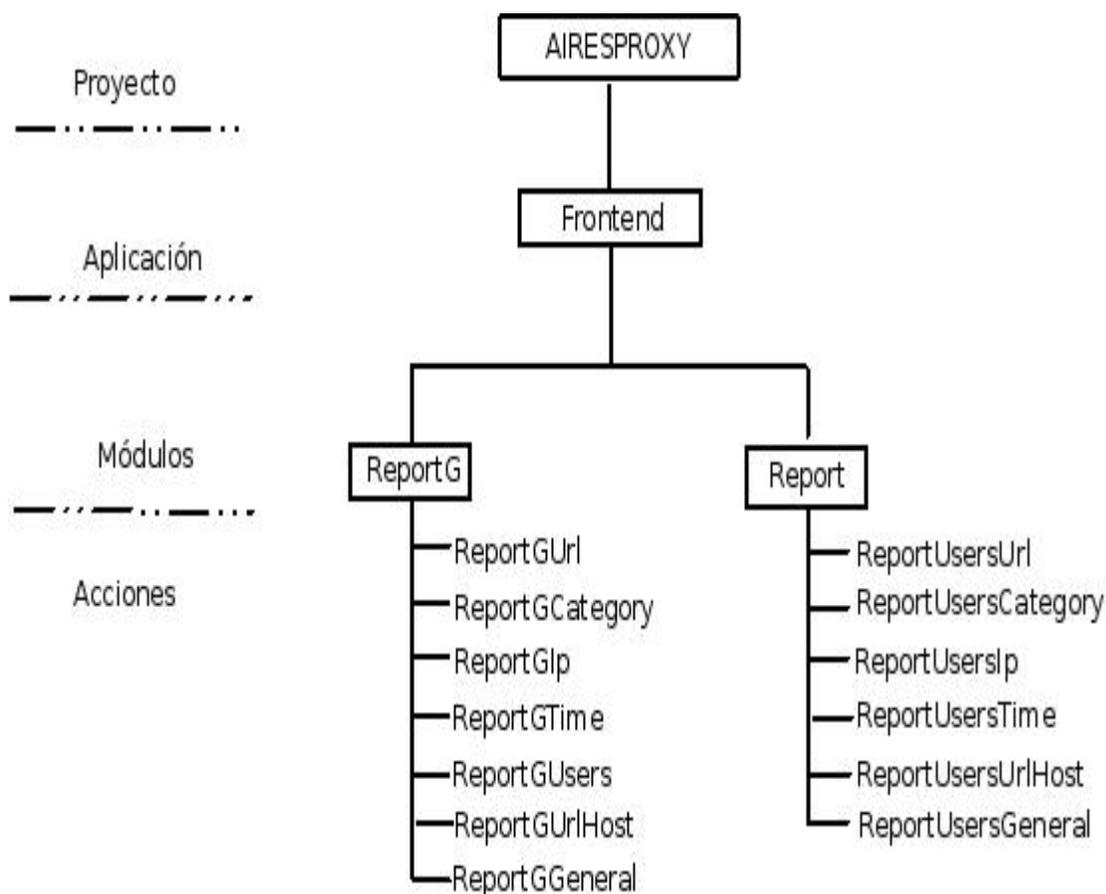


Fig. 19 Organización de la aplicación

3.3.2 Diagrama de clases del diseño

3.3.2.1 Interfaz Web: Módulo Reporte General

3.3.2.1.1 CU Reporte por URL

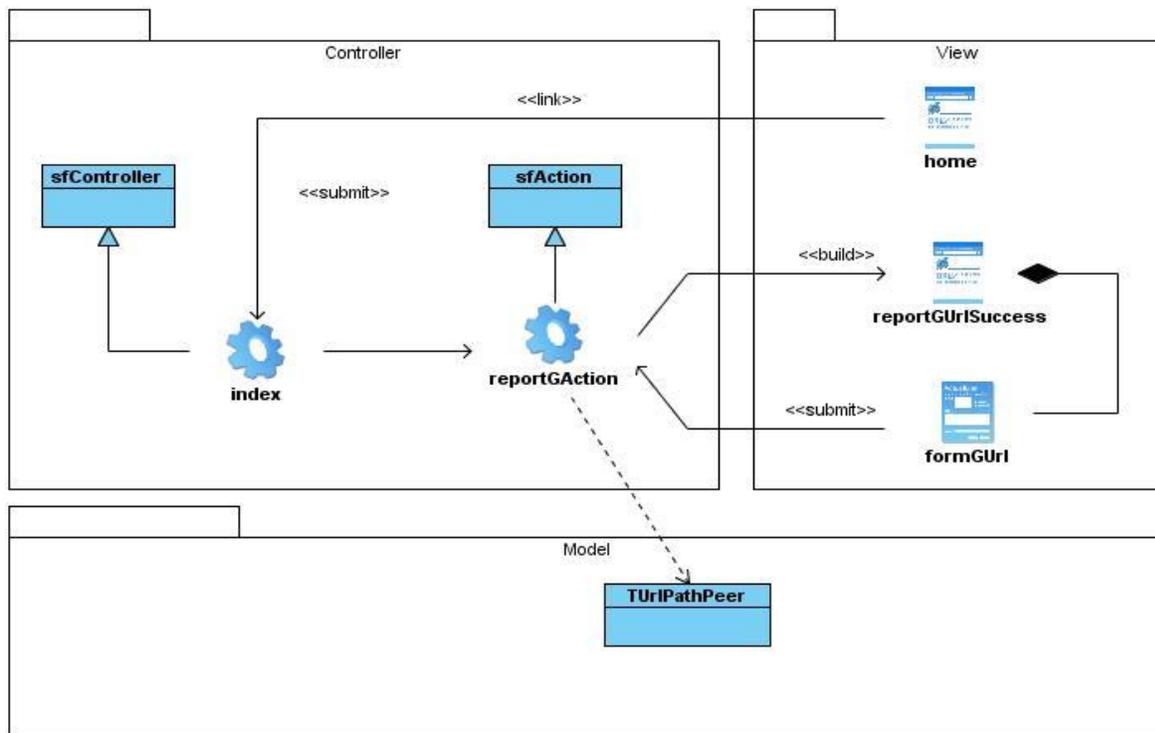


Fig. 20 Diagrama de clase del diseño: CU Reporte por URL

3.3.2.1.2 CU Reporte por Categoría

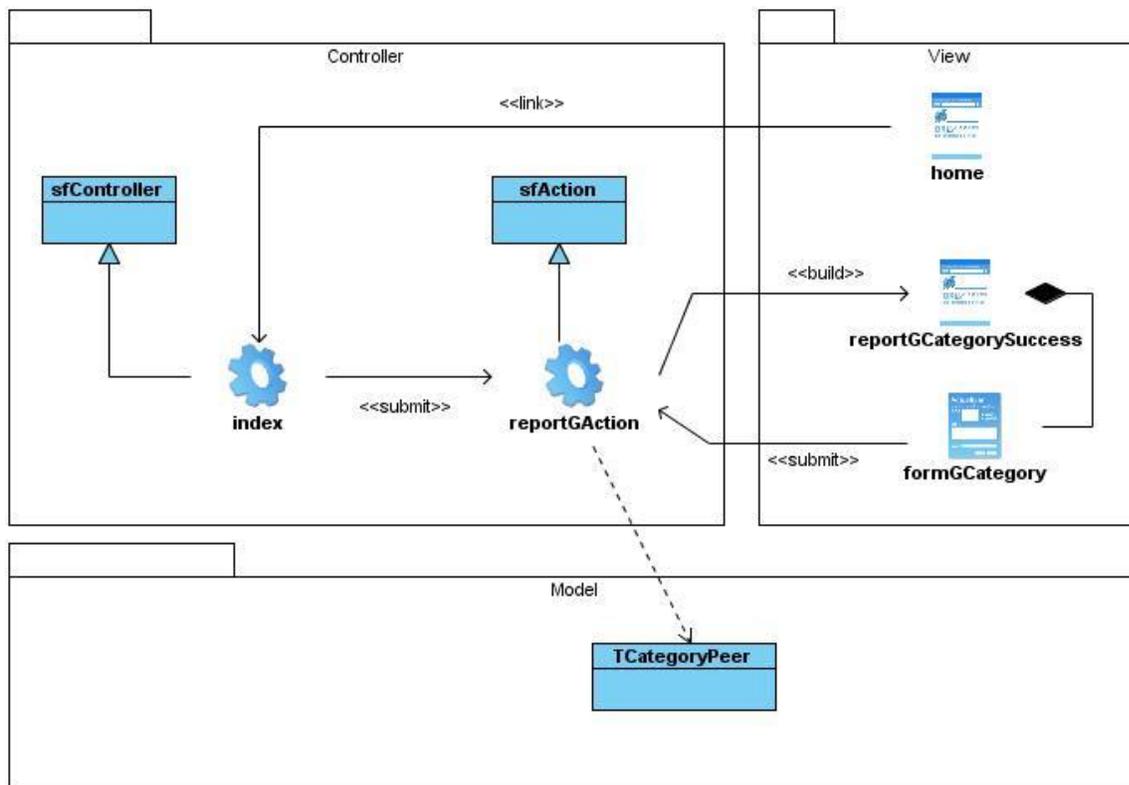


Fig. 21 Diagrama de clase del diseño: CU Reporte por Categoría

3.3.2.1.3 CU Reporte por IP

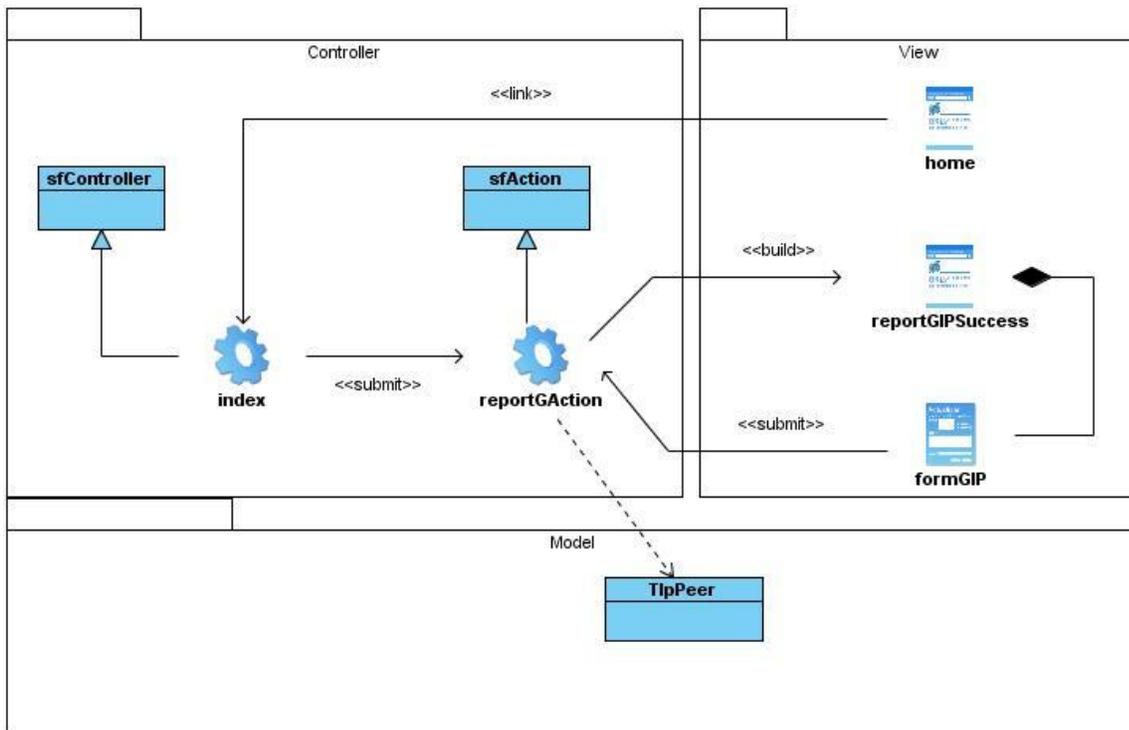


Fig. 22 Diagrama de clase del diseño: CU Reporte por IP

3.3.2.1.4 CU Reporte por tiempo

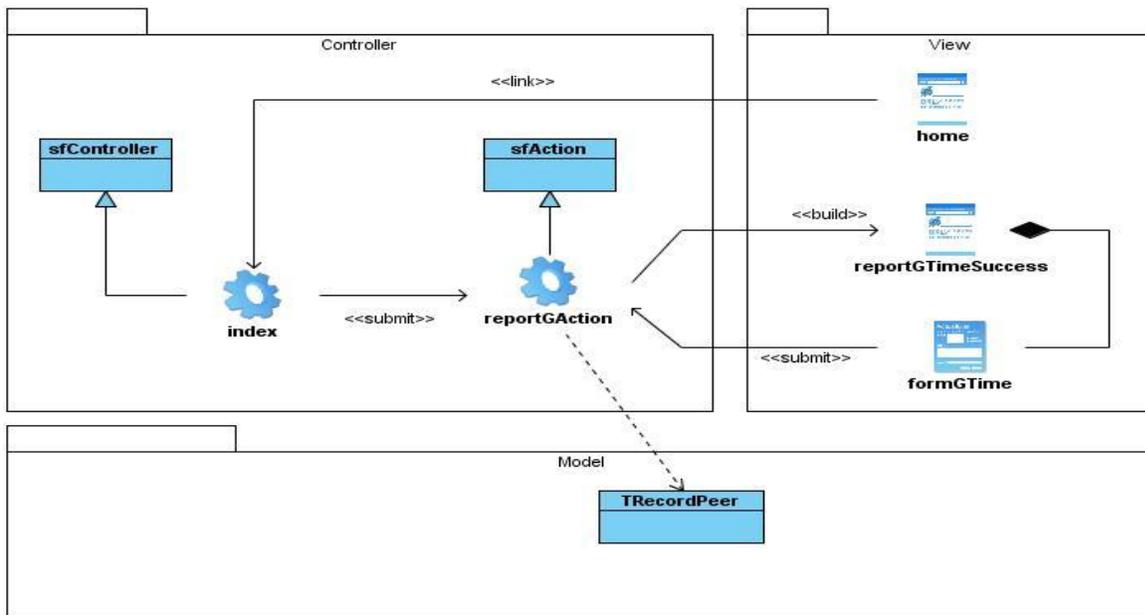


Fig. 23 Diagrama de clase del diseño: CU Reporte por tiempo

3.3.2.1.5 CU Reporte por usuario

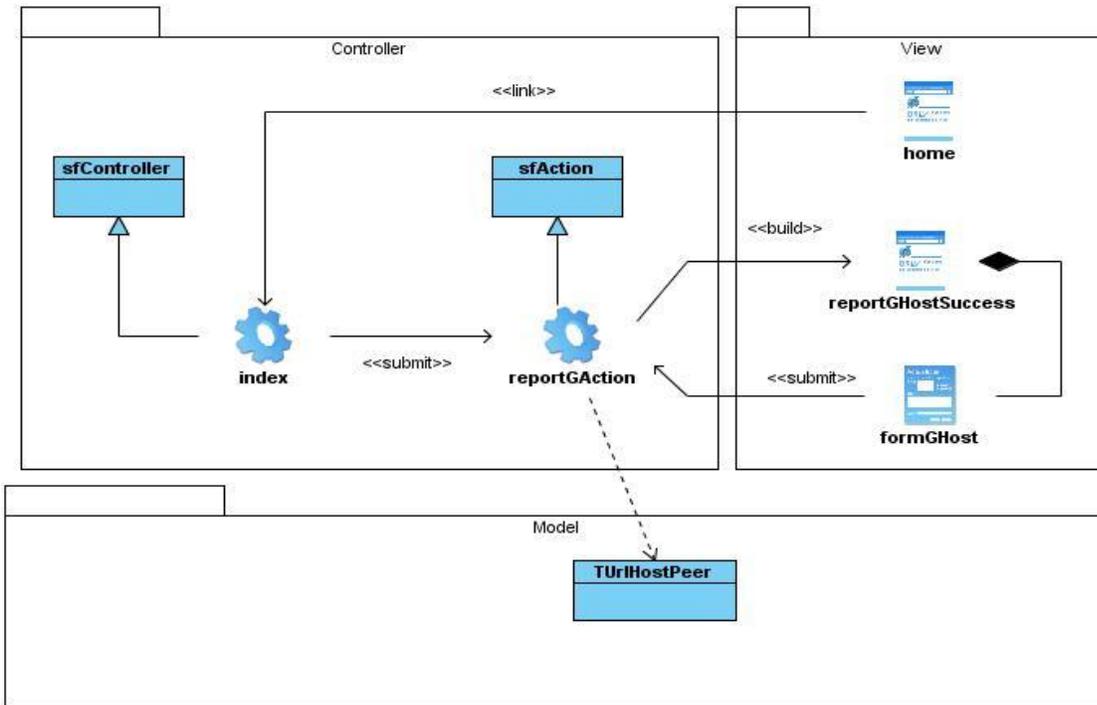


Fig. 24 Diagrama de clase del diseño: CU Reporte por usuario

3.3.2.1.6 CU Reporte por directorio de URL

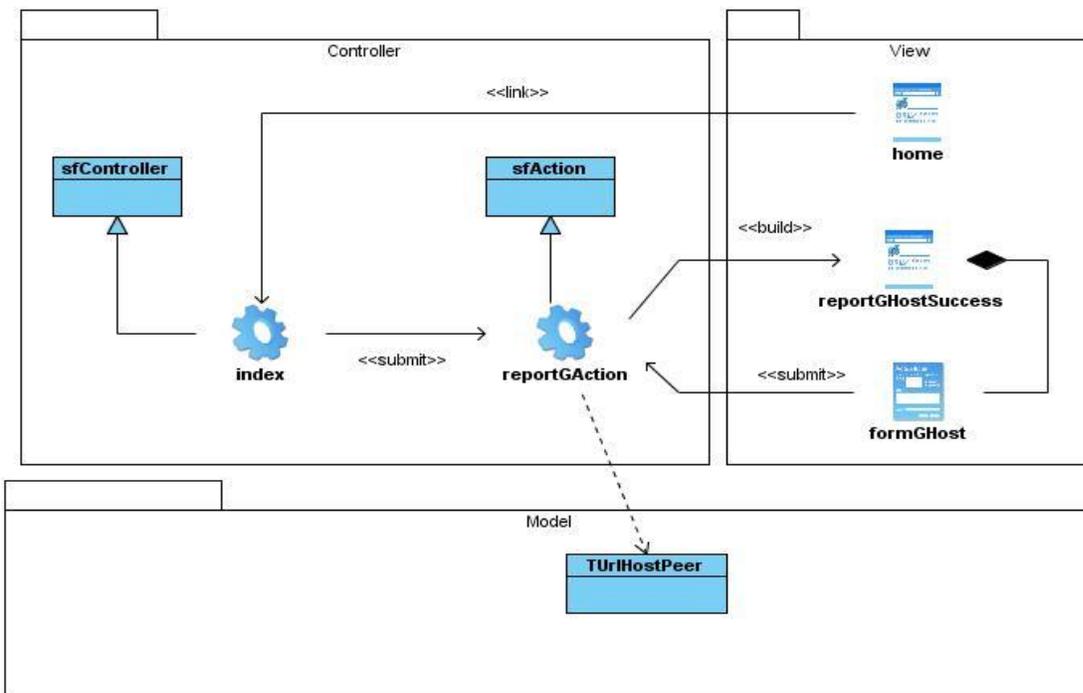


Fig. 25 Diagrama de clase del diseño: CU Reporte por directorio de URL

3.3.2.1.7 CU Reporte sobre navegación general

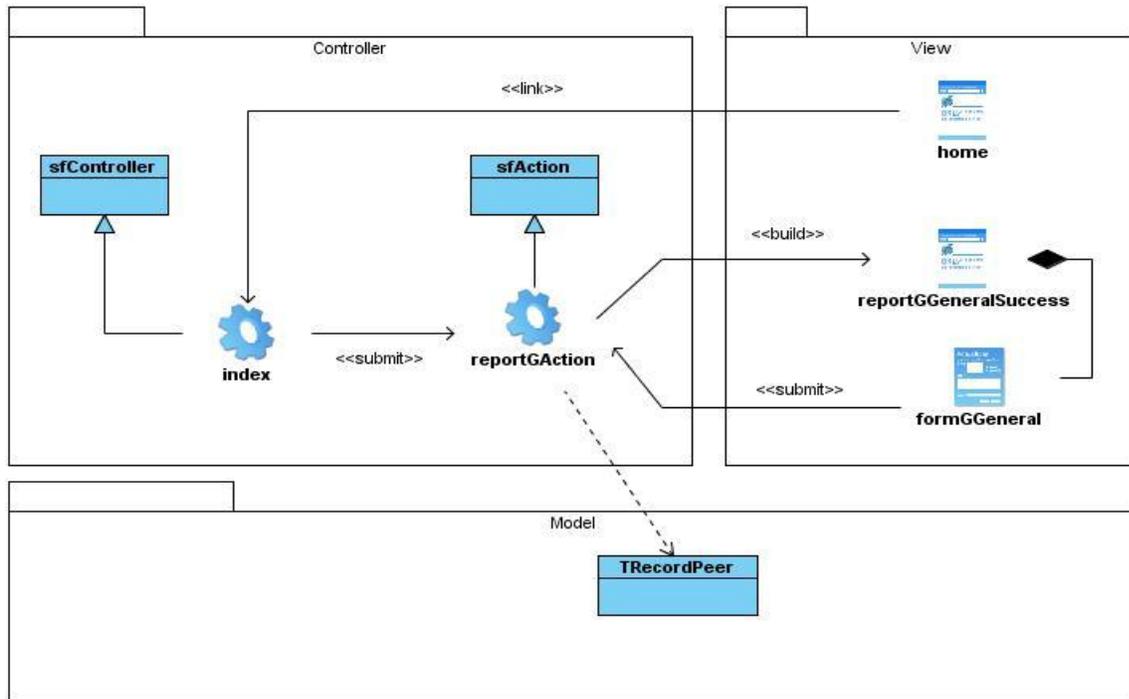


Fig. 26 Diagrama de clase del diseño: CU Reporte sobre navegación general

3.3.2.2 Interfaz Web: Módulo Reporte por usuario

3.3.2.2.1 CU Reporte de URLs por usuario

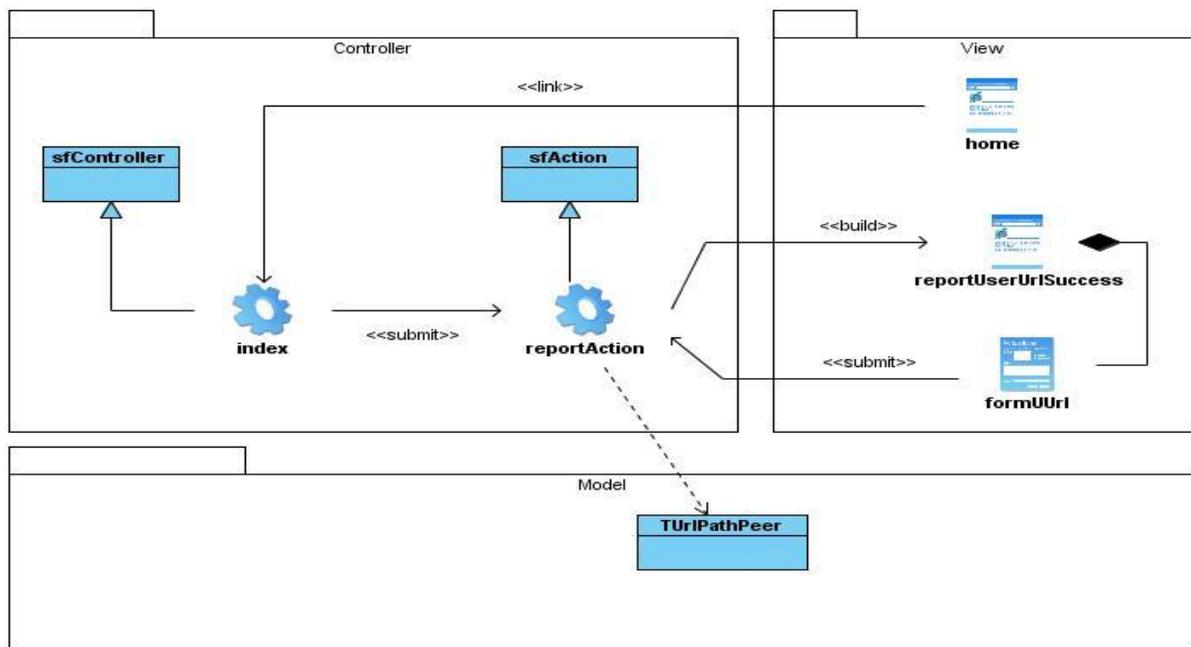


Fig. 27 Diagrama de clase del diseño: CU Reporte de URLs por usuario

3.3.2.2 CU Reporte de categorías por usuario

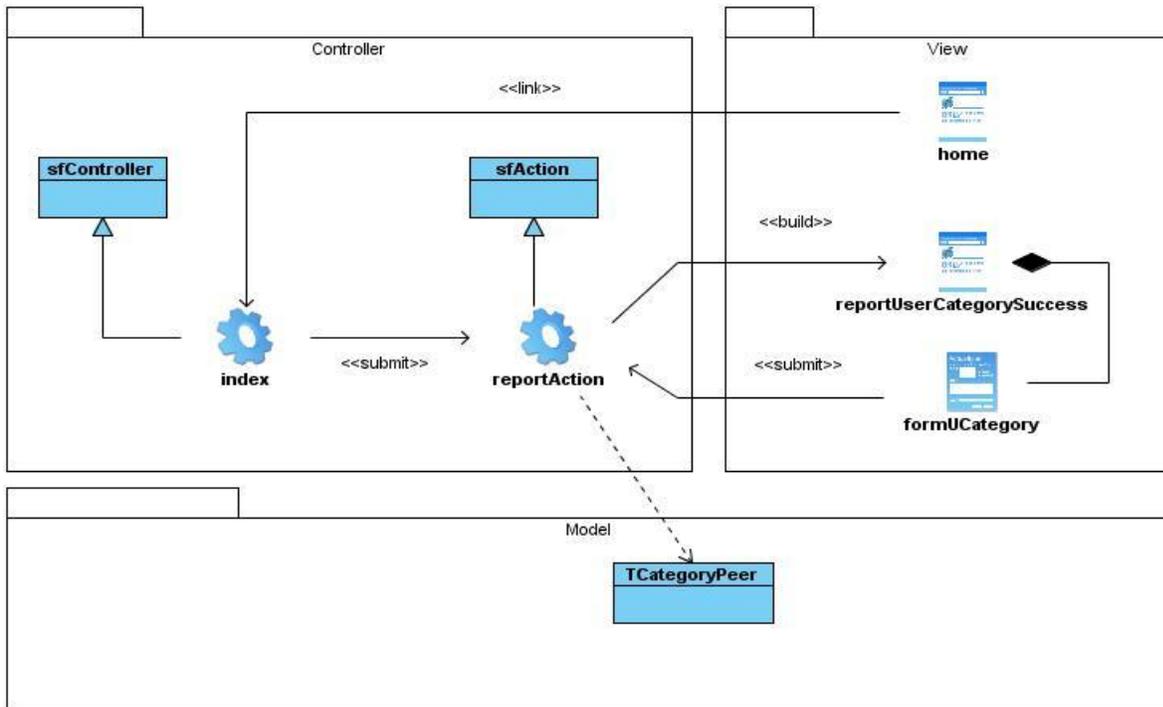


Fig. 28 Diagrama de clase del diseño: CU Reporte de categorías por usuario

3.3.2.2.3 CU Reporte de IPs por usuario

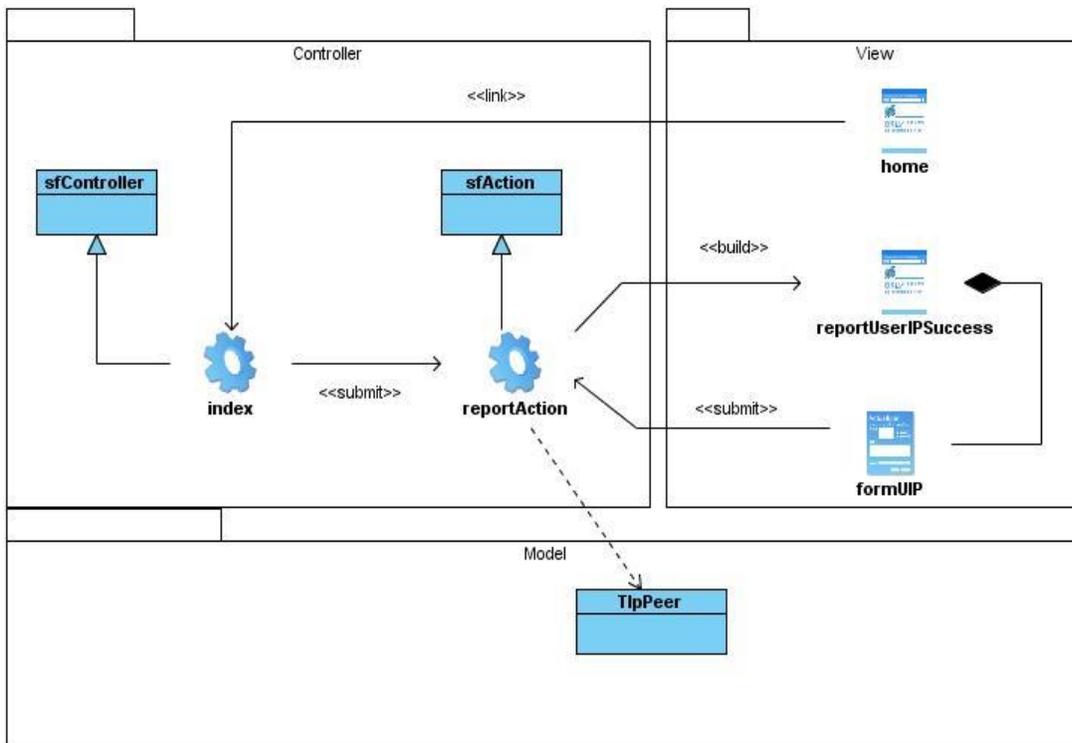


Fig. 29 Diagrama de clase del diseño: CU Reporte de IPs por usuario

3.3.2.2.4 CU Reporte de tiempo por usuario

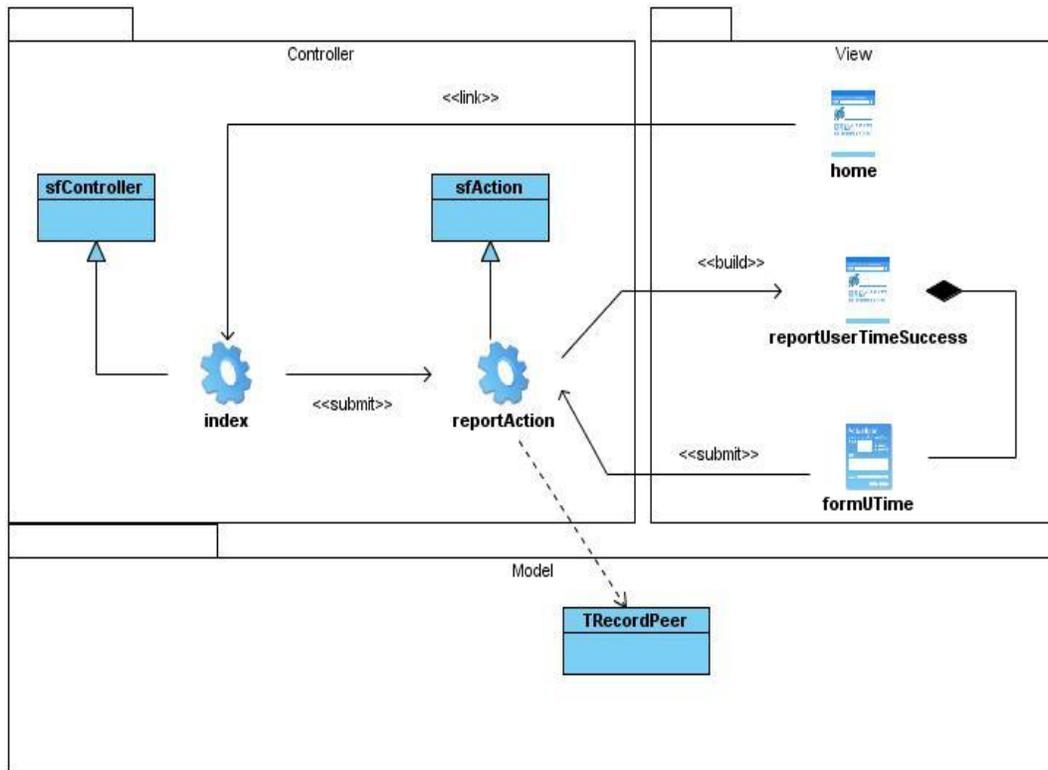


Fig. 30 Diagrama de clase del diseño: CU Reporte de tiempo por usuario

3.3.2.2.5 CU Reporte de directorio de URL por usuario

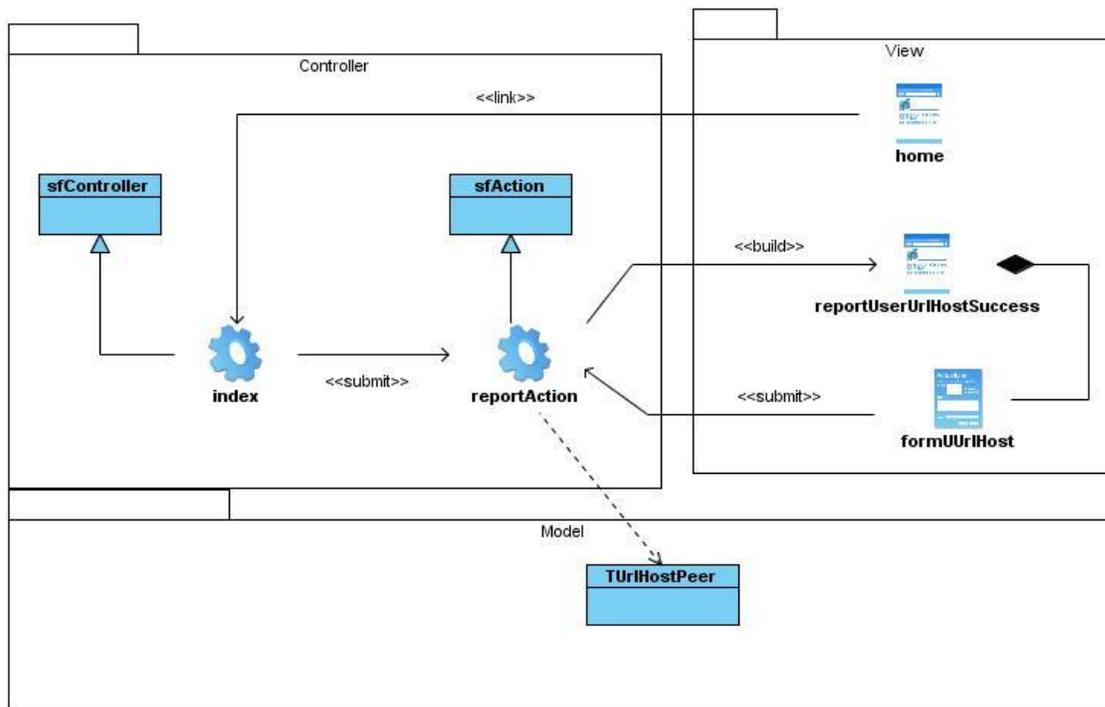


Fig. 31 Diagrama de clase del diseño: CU Reporte de directorio de URL por usuario

3.3.2.2.6 CU Reporte de navegación general por usuario

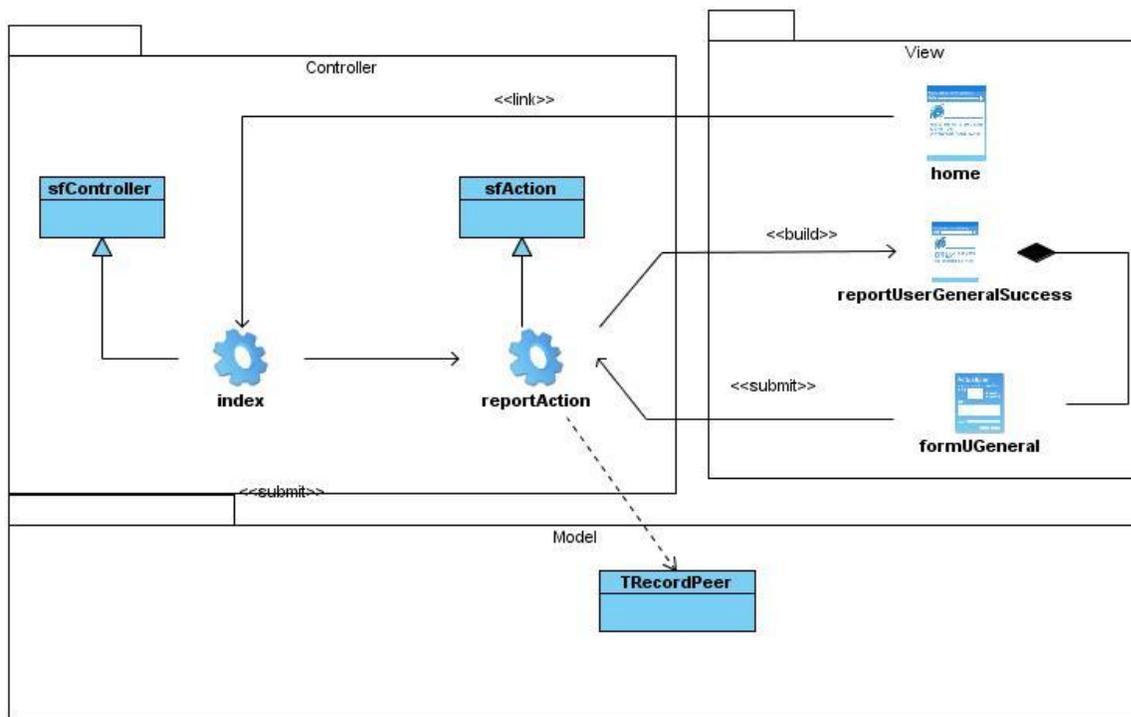


Fig. 32 Diagrama de clase del diseño: CU Reporte de directorio de URL por usuario

3.3.2.3 Analizador

3.3.2.3.1 CU Analizar Log

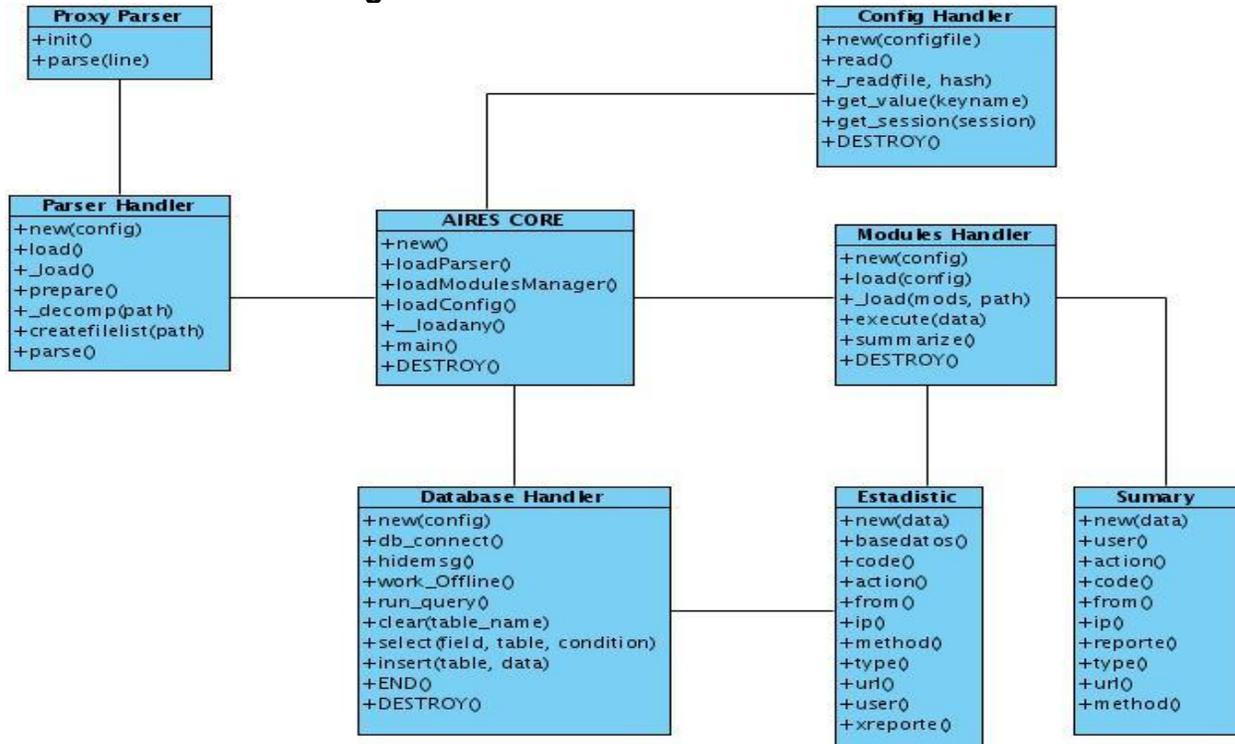


Fig. 33 Diagrama de clase del diseño: CU Analizar Log

3.3.3 Diagramas de secuencia

3.3.3.1 Interfaz Web

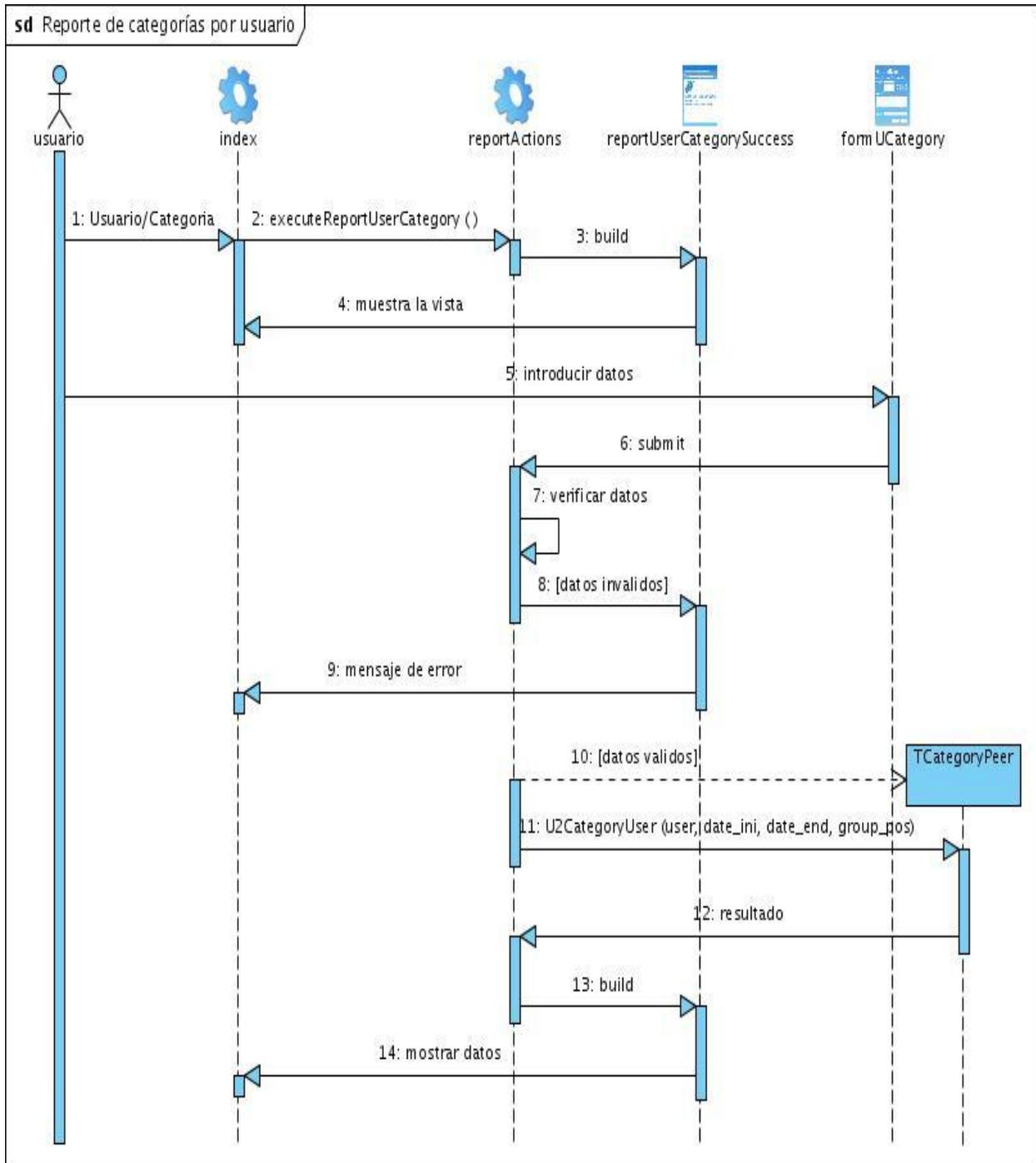


Fig. 34 Diagrama de secuencia: CU Reporte de categorías por usuario

3.3.3.2 Analizador

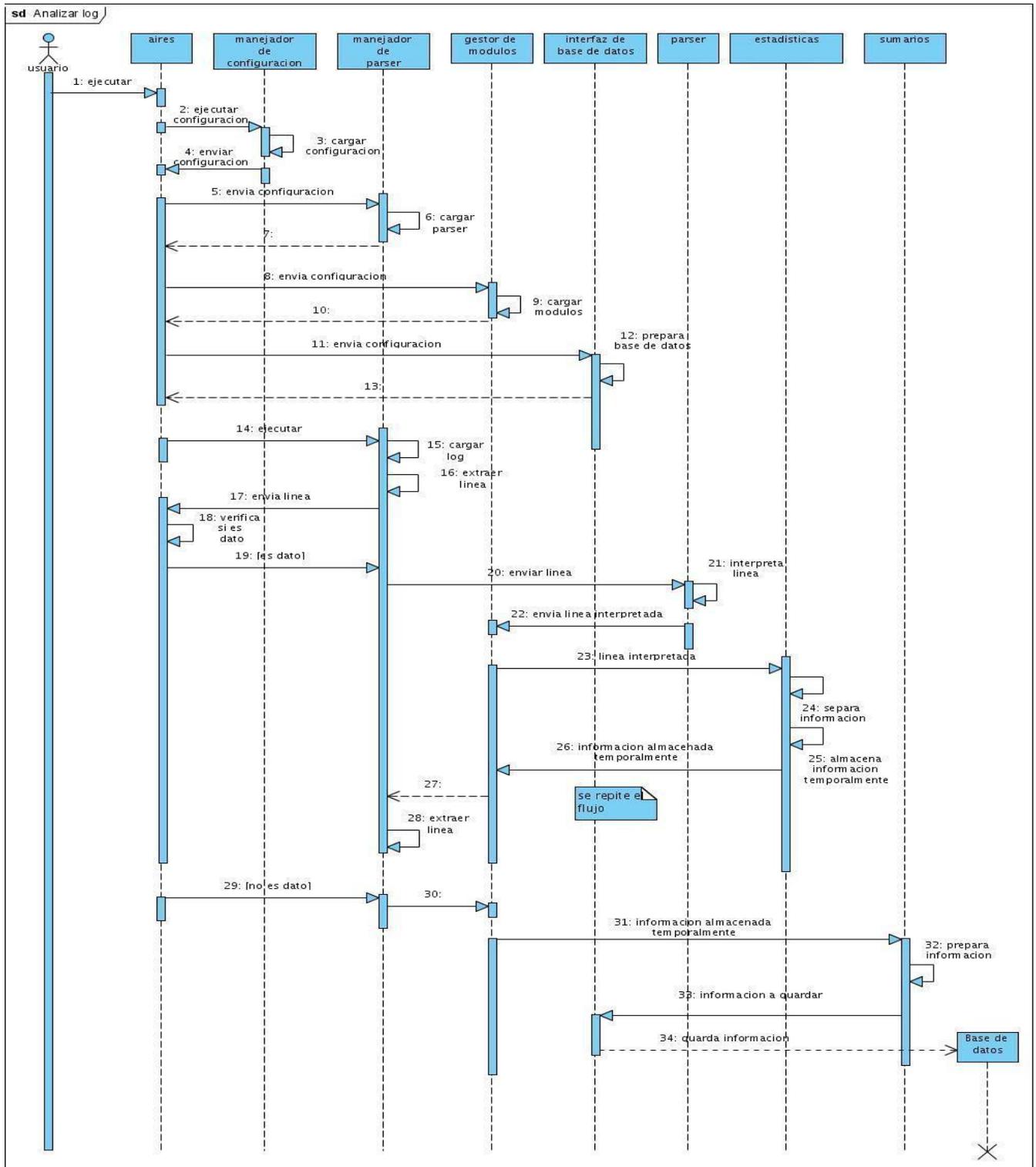


Fig. 35 Diagrama de secuencia: CU Analizar log

Los restantes diagramas de secuencia se encuentran en el anexo 2.

3.3.4 Diseño de la base de datos

3.3.4.1 Diagrama de clases persistentes

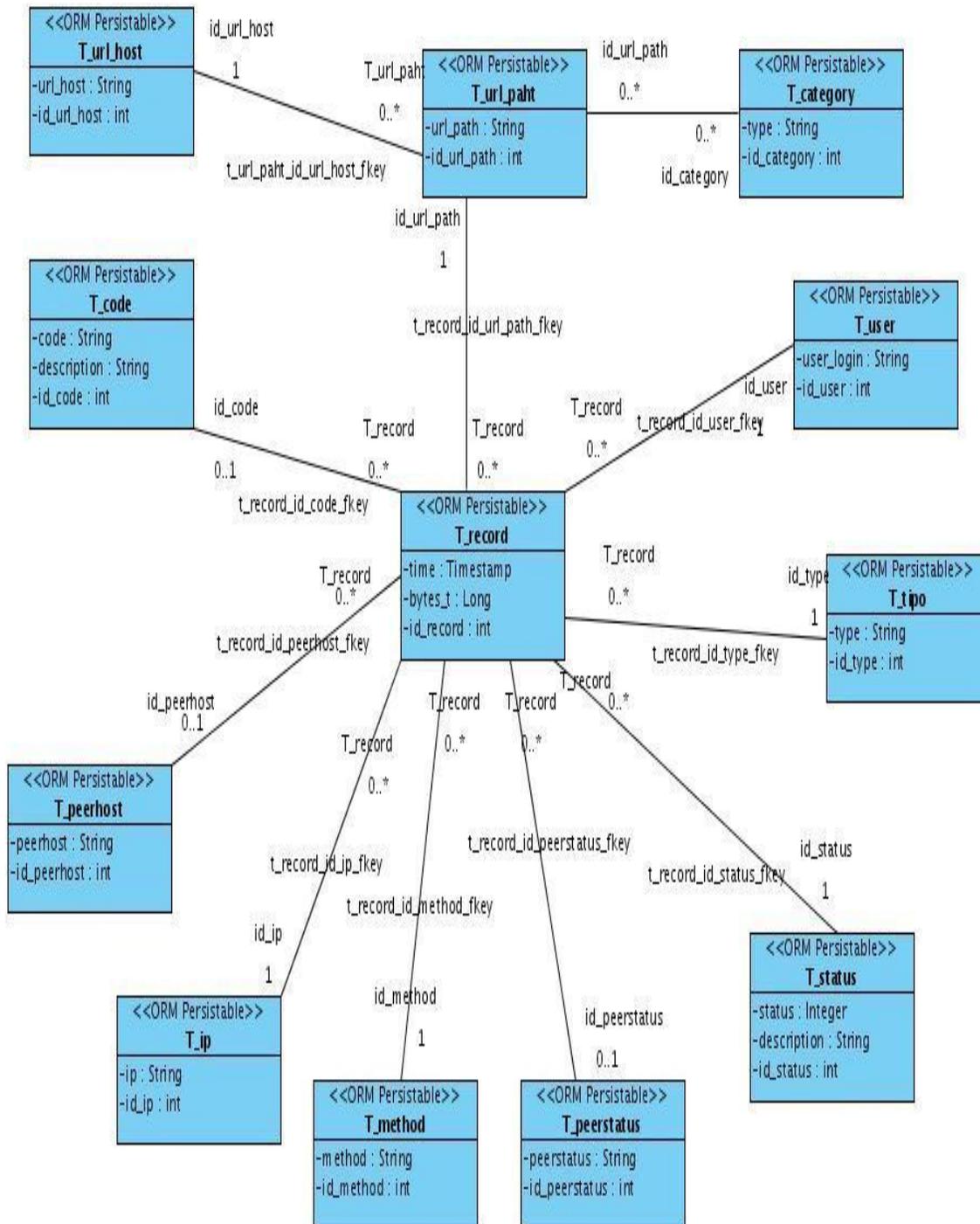


Fig. 36 Diagrama de clases persistentes

3.3.4.2 Diagrama de entidad Relación

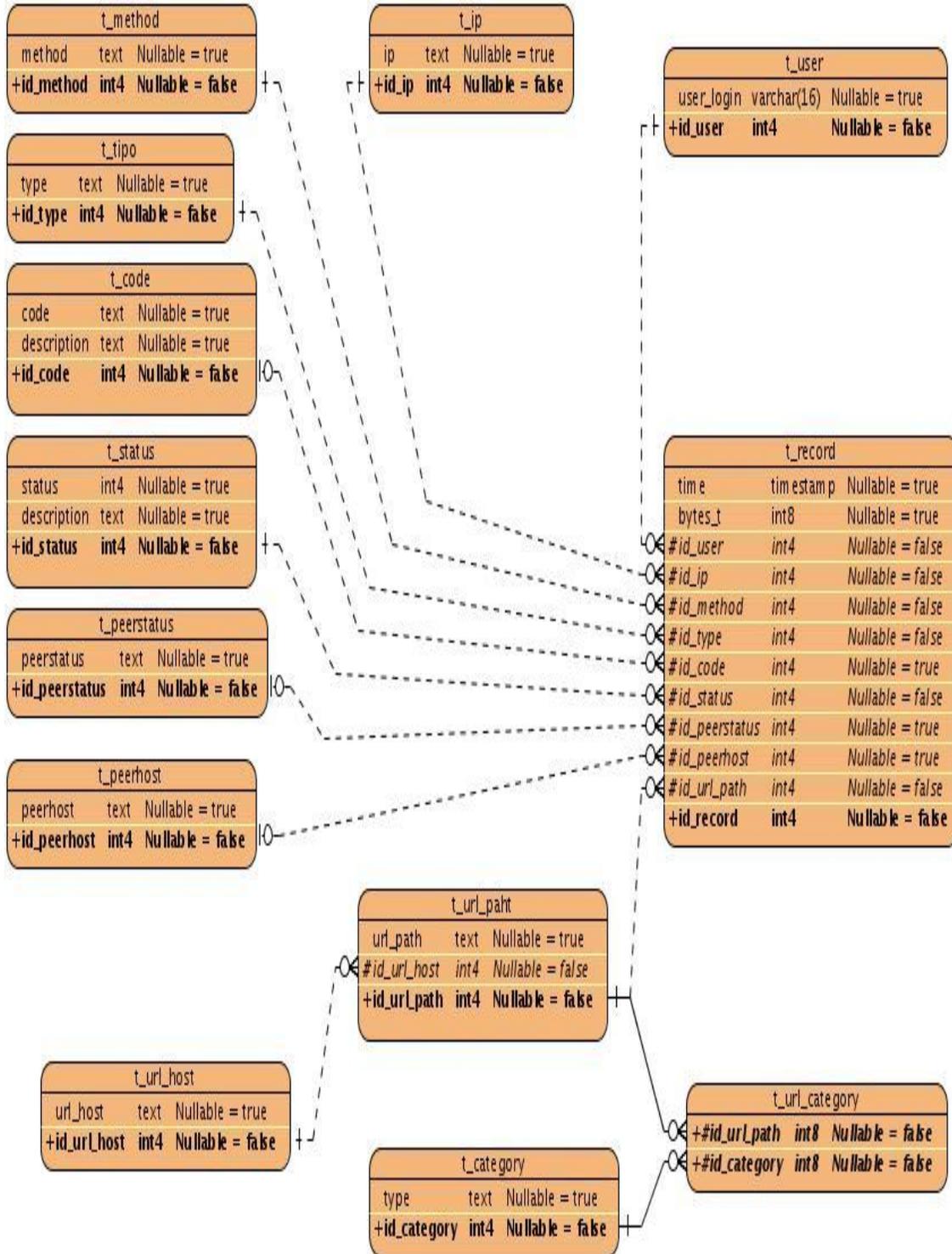


Fig. 37 Diagrama de entidad relación

3.4 Conclusiones

En el capítulo se modeló el flujo de trabajo de análisis y diseño donde se logró un acercamiento a la solución de los requisitos funcionales, se explicó la arquitectura MVC que establece symfony y se describen en detalle los elementos que forman parte del sistema y las relaciones entre ellos sirviendo como base para su posterior implementación.

CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción

La implementación comienza con el resultado del diseño. El presente capítulo trata sobre los flujos de trabajo Implementación y Prueba. Se representan los diagramas de despliegue y de componentes de los cuales se brinda un sencillo concepto y la representación gráfica para un mejor entendimiento. Además se realizan pruebas de caja negra las cuales permiten obtener un conjunto de entradas con el objetivo de ejercitar completamente todos los requisitos funcionales de la aplicación.

4.2 Diagrama de despliegue

Un diagrama de despliegue muestra la disposición física de los recursos de ejecución computacional, tales como computadores y sus interconexiones los cuales se denominan nodos. Durante la ejecución, los nodos pueden contener componentes y objetos. La asignación de componentes y de objetos a los nodos puede ser estática, o pueden migrar entre ellos [41]. El diagrama de despliegue representa en sí mismo la correspondencia entre la arquitectura del software y el hardware del sistema.

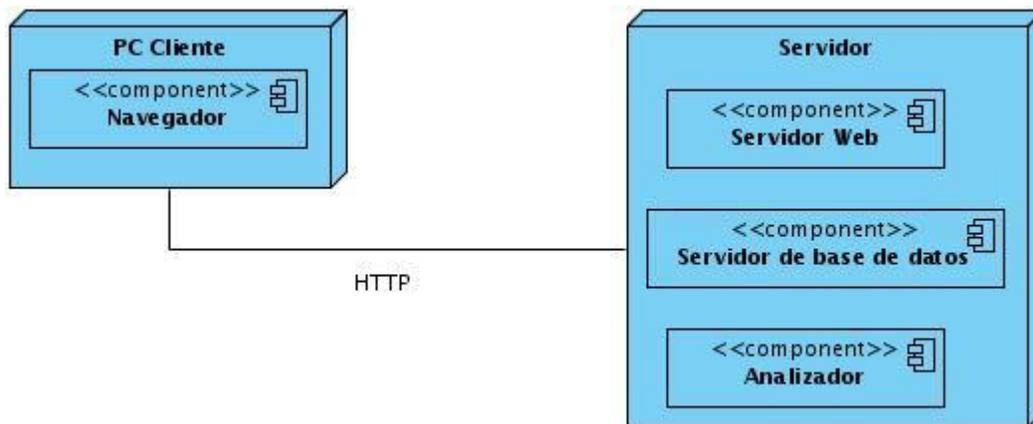


Fig. 38 Diagrama de despliegue

4.3 Diagramas de componentes

El diagrama de componentes describe como los elementos del modelo de diseño se implementan en términos de componentes. Además como se organizan de acuerdo con los

mecanismos de estructuración y modulación disponibles en el entorno de implementación y lenguaje de programación utilizado [42].

4.3.1 Interfaz web

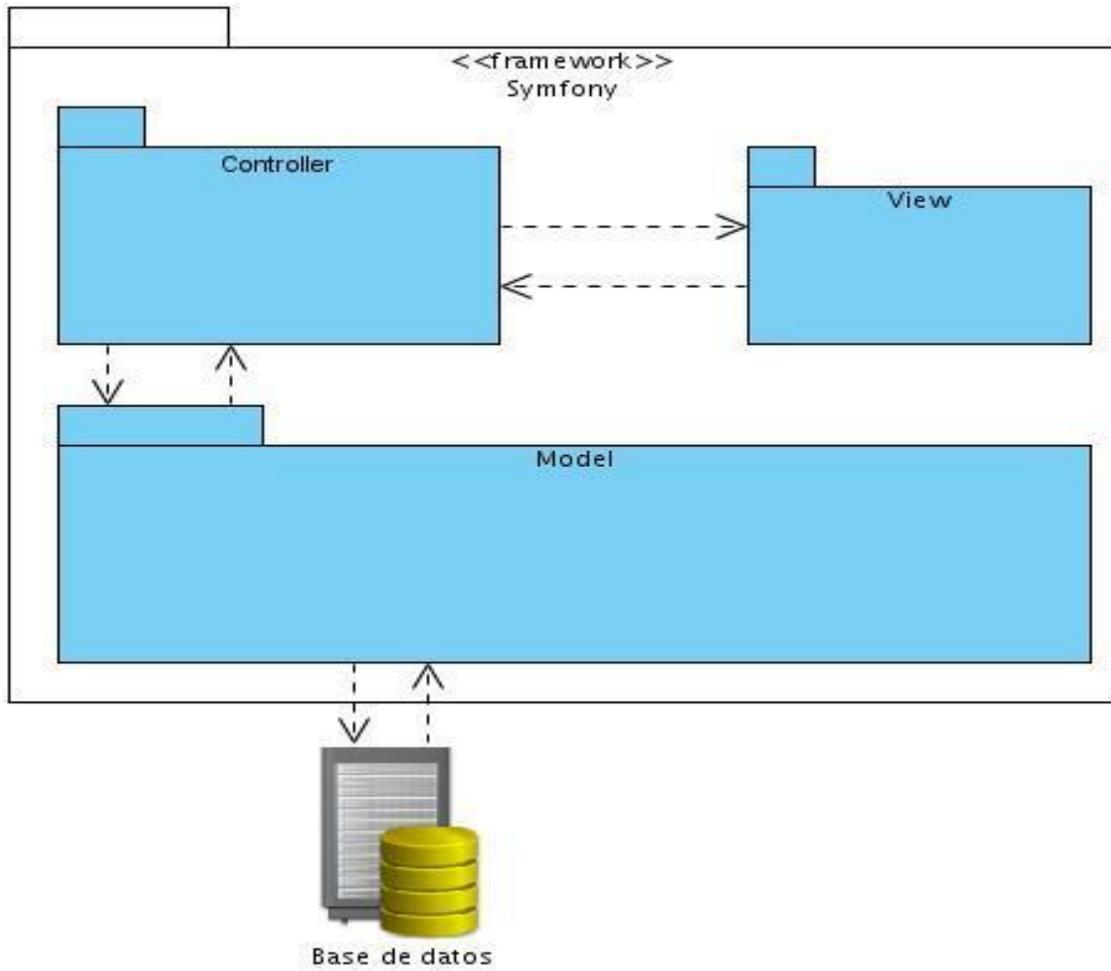


Fig. 39 Diagrama de componentes general de la interfaz web

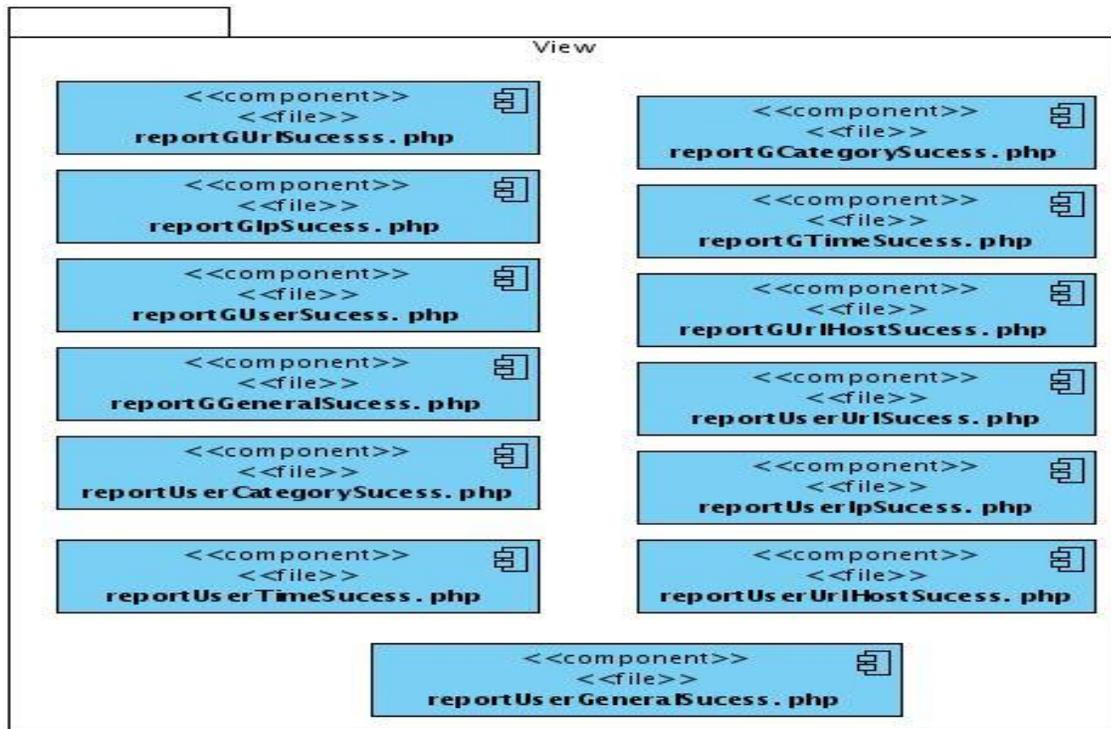


Fig. 40 Diagrama de componentes paquete View de la interfaz web

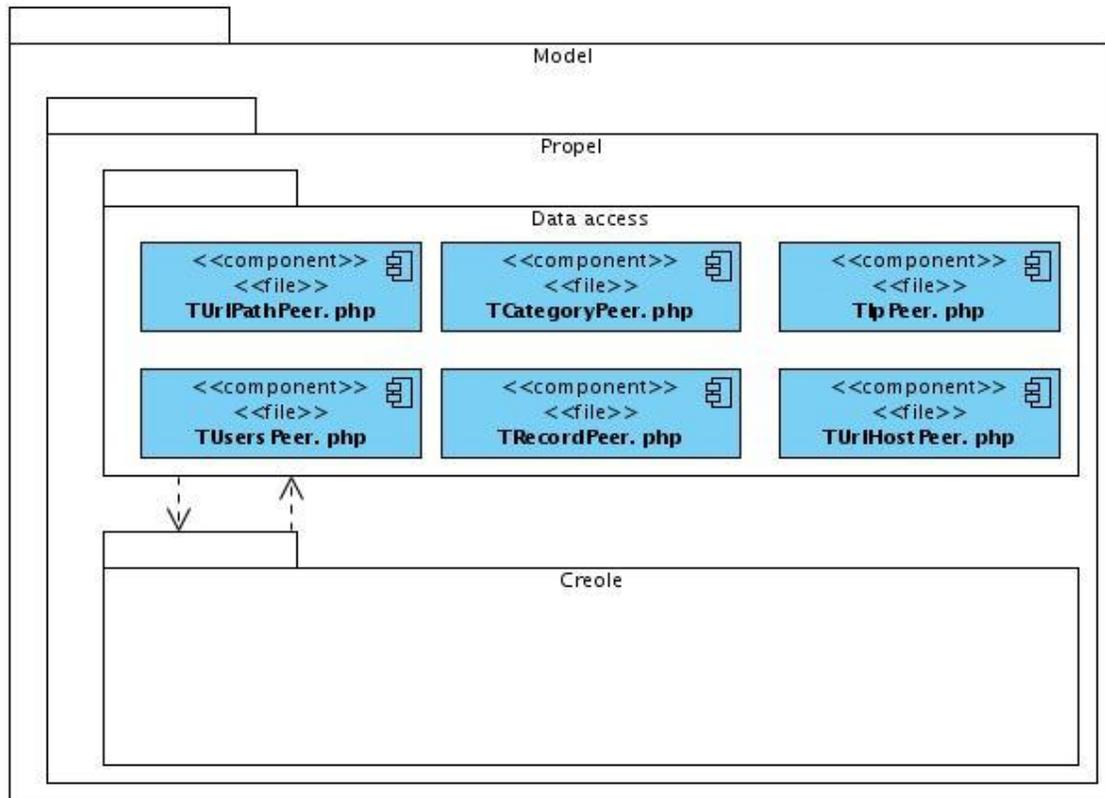


Fig. 41 Diagrama de componentes paquete Model de la interfaz web

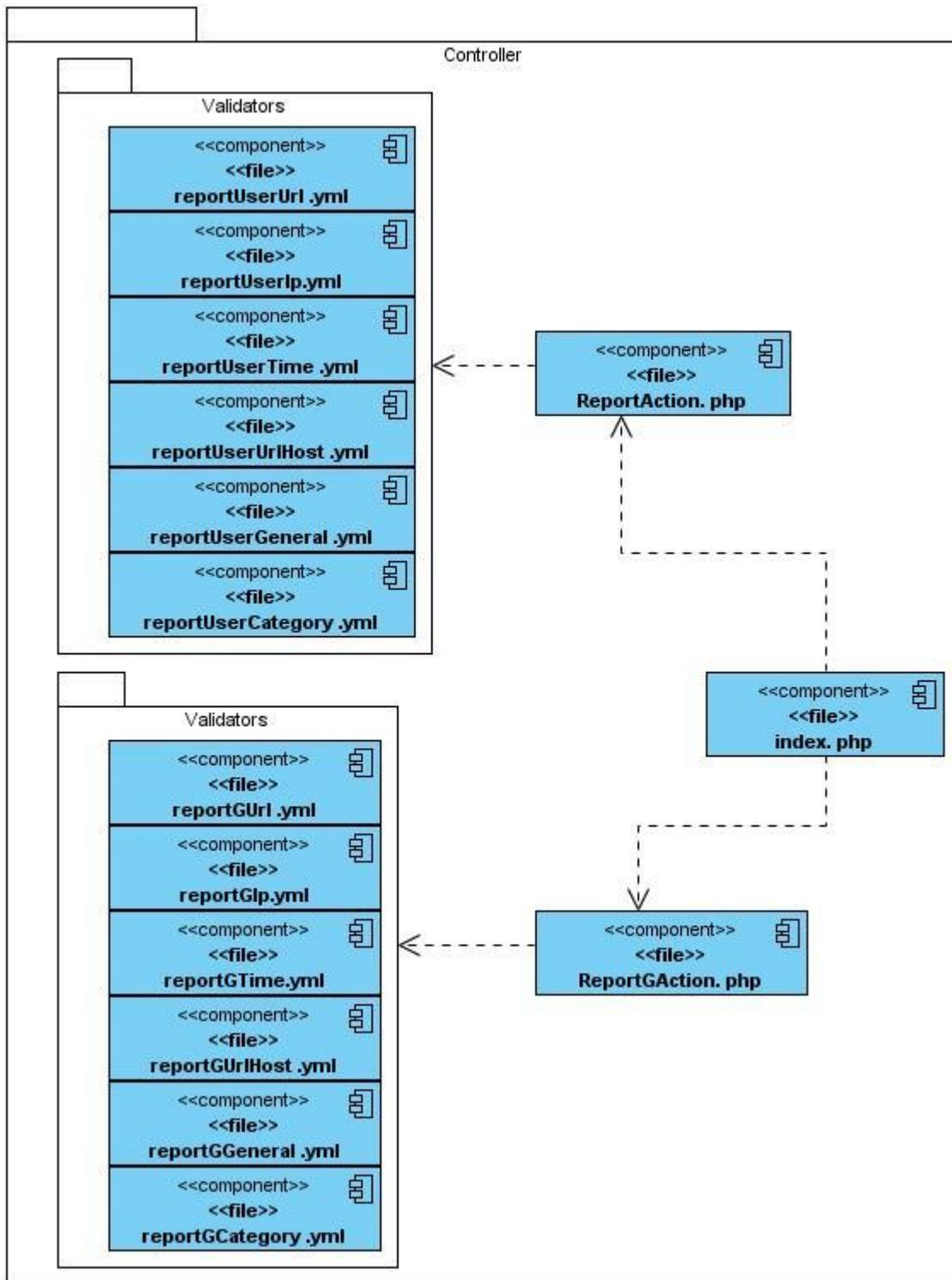


Fig. 42 Diagrama de componentes paquete Controller de la interfaz web

4.3.2 Analizador

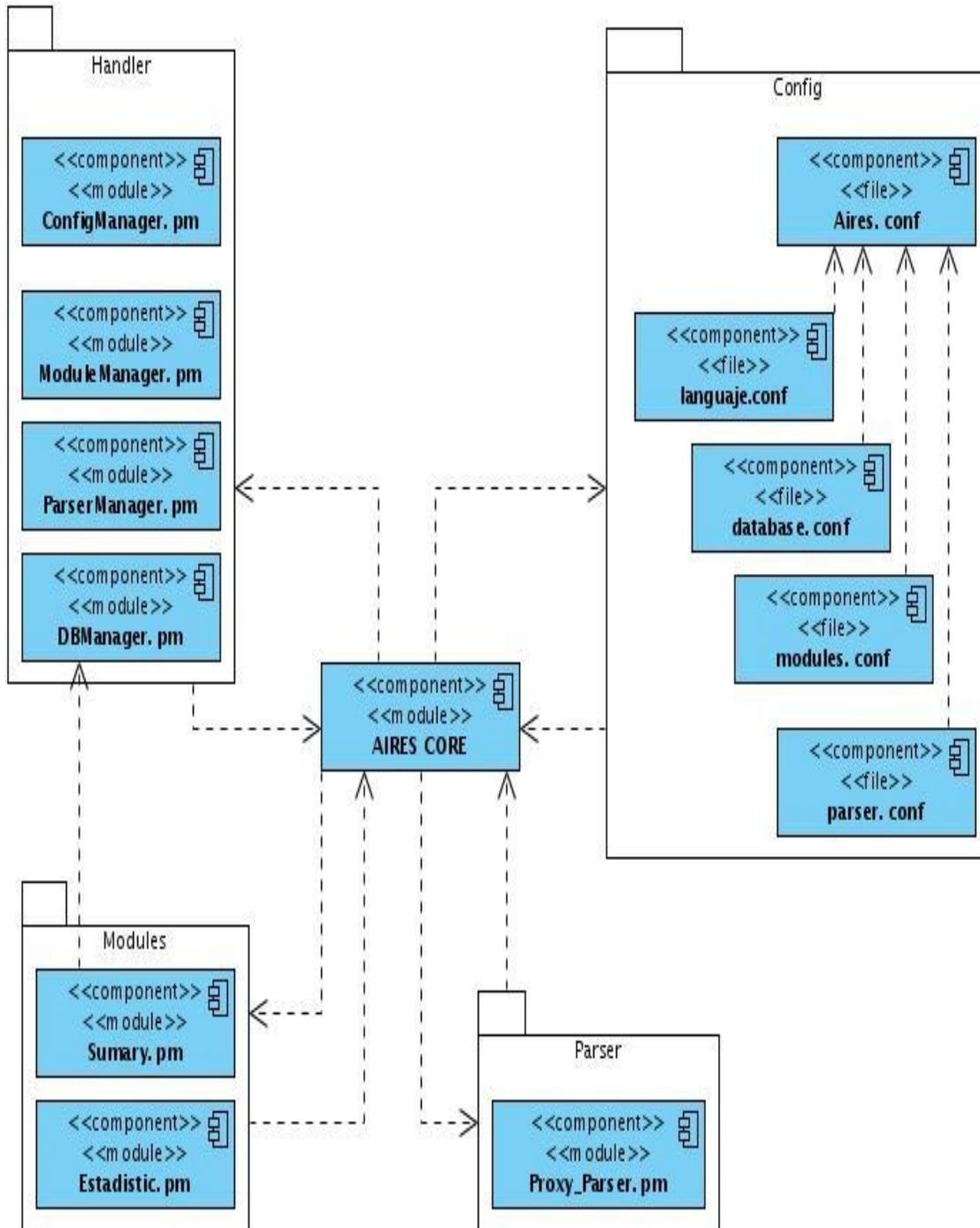


Fig. 43 Diagrama de componentes del analizador

4.4 Pruebas

Las pruebas del software son un elemento crítico para la garantía de calidad del producto y representan una revisión final de las especificaciones, del diseño y de la codificación. Demuestran hasta qué punto las funciones del software cumplen con los requisitos funcionales. Además los datos recogidos en el desarrollo de la prueba proporcionan un buen indicador de la fiabilidad del software y, de alguna manera, la calidad del mismo, aunque no asegura la ausencia de defectos si puede demostrar la existencia de errores [47].

Cualquier producto de ingeniería puede probarse de dos formas distintas a través de prueba de caja negra y prueba de caja blanca.

Las pruebas de caja negra se refieren a las pruebas que se llevan a cabo sobre la interfaz del software. Los casos de pruebas pretenden demostrar que las funciones del producto son operativas, que la entrada se realiza de forma adecuada, que se produce un resultado correcto y que la integridad de la información externa se mantiene. Las pruebas de caja negra examinan algunos aspectos del modelo fundamental del sistema sin valorar demasiado la estructura lógica interna del software [47].

La prueba de caja blanca del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado [47].

4.4.1 Pruebas de caja negra

Las pruebas de caja negra, también denominada pruebas de comportamiento se centran en los requisitos funcionales del software. Permiten obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Este tipo de prueba no es una alternativa a las técnicas de prueba de caja blanca, más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de caja blanca no pueden detectar. Algunos de estos errores pueden ser funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a base de datos externas, errores de rendimiento y errores de inicialización y determinación. Este tipo de prueba, tiende a aplicarse durante fases posteriores de la prueba ya que ignora

intencionadamente la estructura de control y centra su atención en el campo de información [47].

Cada tipo de prueba tiene sus propias técnicas para realizar los casos de pruebas. Dentro de las técnicas de caja negra existentes se utilizó la de partición equivalente.

4.4.1.1 Partición equivalente

Es una técnica de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores, que de otro modo, requerirá la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar [47].

El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Típicamente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. Las clases de equivalencia se pueden definir de acuerdo con las siguientes directrices: [47]

- Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos no válidas.
- Si una condición de entrada requiere un valor específico, se define una clase de equivalencia válida y dos no válidas.
- Si una condición de entrada especifica un miembro de un conjunto, se define una clase de equivalencia válida y una no válida.
- Si una condición de entrada es lógica, se define una clase de equivalencia válida y una no válida.

Aplicando las directrices para la obtención de clases de equivalencia, se pueden desarrollar y ejecutar casos de prueba para cada elemento de datos del campo de entrada. Los casos de

prueba se seleccionan de forma que se ejercite el mayor número de atributos de cada clase de equivalencia a la vez.

4.4.2 Casos de Prueba

4.4.2.1 Interfaz web

4.4.2.1.1 CU Reporte de categorías por usuario

Descripción

El caso de uso se inicia cuando el usuario desea obtener el reporte general por categorías en un espacio de tiempo definido y para un usuario seleccionado.

Flujo Central:

Se especifica el usuario, fecha inicio, fecha fin y criterio de ordenamiento para realizar la búsqueda.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El campo usuario solo permite usuarios que existan en la base de datos.		El sistema solo permite usuarios que existan en la base de datos.	satisfactorio	
	El campo usuario permite usuarios que no existan en la base de datos, permite además caracteres especiales y la	El sistema no permite usuarios que no estén en la base de datos, ni caracteres especiales, ni que la longitud del campo sea mayor de 128 caracteres.	satisfactorio	

	longitud del campo es mayor de 128 caracteres.			
Los campos desde y hasta solo permiten fechas en formato aaaa/mm/dd y el campo hasta debe ser mayor que el campo desde .		El sistema permite el formato de fecha especificado (aaaa/mm/dd).	Satisfactorio	
	Los campos desde y hasta permiten cualquier cadena de caracteres, y el campo hasta no es mayor que el campo desde .	El sistema no permite la entrada de cualquier cadena, ni acepta que el campo hasta no sea mayor que el campo desde .	Satisfactorio	
Existen reportes para el usuario y el rango de fecha especificada.		El sistema muestra reporte de categorías por el usuario en una tabla con las 10 primeras categorías y la cantidad de Bytes transferidos, de IPs, de	Satisfactorio	

		URLs y de visitas por cada una. Además una gráfica con las categorías y la cantidad del elemento especificado en el campo ordenamiento del formulario. Además muestra un paginado para acceder a las soluciones restantes		
	No existen reportes para el usuario y el rango de fecha especificada.	El sistema muestra mensaje al usuario informándole que no existen datos para el usuario en la fecha especificada.	Satisfactorio	

Tabla 21 Prueba del CP Reporte de categorías por usuario

4.4.2.1.2 CU Reporte de navegación general por usuario

Descripción

El caso de uso se inicia cuando el usuario desea obtener el reporte de navegación general en un espacio de tiempo definido y para un usuario seleccionado.

Flujo Central:

Se especifica el usuario, fecha inicio, fecha fin y criterio de ordenamiento para realizar la búsqueda.

Clases Válidas	Clases Inválidas	Resultado	Resultado	Observaciones
----------------	------------------	-----------	-----------	---------------

		Esperado	de la Prueba	
El campo usuario solo permite usuarios que existan en la base de datos.		El sistema solo permite usuarios que existan en la base de datos.	satisfactorio	
	El campo usuario no permite usuarios que no existan en la base de datos, permite además caracteres especiales y la longitud del campo es mayor de 128 caracteres.	El sistema no permite usuarios que no estén en la base de datos, ni caracteres especiales, ni que la longitud del campo sea mayor de 128 caracteres.	satisfactorio	
Los campos desde y hasta solo permiten fechas en formato aaaa/mm/dd y el campo hasta debe ser mayor que el campo desde .		El sistema permite el formato de fecha especificado (aaaa/mm/dd).	Satisfactorio	
	Los campos desde y hasta permiten	El sistema no permite la entrada	Satisfactorio	

	cualquier cadena de caracteres, y el campo hasta no es mayor que el campo desde .	de cualquier cadena, ni acepta que el campo hasta no sea mayor que el campo desde .		
Existen reportes para el usuario y el rango de fecha especificada.		El sistema muestra reporte de navegación general del usuario en una tabla con el usuario, la cantidad de Bytes transferidos, de IPs, de URLs, de visitas y de categorías de la navegación en el rango de fechas especificado. Además una gráfica con todas las cantidades de los elementos antes mencionados.	Satisfactorio	
	No existen reportes para el usuario y el rango de fecha especificada.	El sistema muestra mensaje al usuario informándole que no existen datos para el usuario en	Satisfactorio	

		la fecha especificada.		
--	--	------------------------	--	--

Tabla 22 Prueba del CP Reporte de navegación general por usuario

4.4.2.1.3 CU Reporte por categorías

Descripción

El caso de uso se inicia cuando el usuario desea obtener el reporte general por categorías en un espacio de tiempo definido.

Flujo Central:

Se especifica la fecha inicio, fecha fin y criterio de ordenamiento para realizar la búsqueda.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Los campos desde y hasta solo permiten fechas en formato aaaa/mm/dd y el campo hasta debe ser mayor que el campo desde .		El sistema permite el formato de fecha especificado (aaaa/mm/dd).	Satisfactorio	
	Los campos desde y hasta permiten cualquier cadena de caracteres, y el campo hasta no es mayor que el campo desde .	El sistema no permite la entrada de cualquier cadena, ni acepta que el campo hasta no sea mayor que el campo desde .	Satisfactorio	

<p>Existen reportes para el usuario y el rango de fecha especificada.</p>		<p>El sistema muestra reporte por categorías en una tabla con las 10 primeras categorías y la cantidad de Bytes transferidos, de IPs, de URLs y de visitas por cada una. Además una gráfica con las categorías y la cantidad del elemento especificado en el campo ordenamiento del formulario. Además muestra un paginado para acceder a las soluciones restantes</p>	<p>Satisfactorio</p>	
	<p>No existen reportes para el usuario y el rango de fecha especificada.</p>	<p>El sistema muestra mensaje al usuario informándole que no existen datos en la fecha especificada.</p>	<p>Satisfactorio</p>	

4.5 Conclusiones

En el capítulo se desarrollaron los flujos de trabajo de implementación y prueba mediante la utilización de los casos de prueba, diagrama de despliegue y de componentes descritos por RUP. Las pruebas son una herramienta para asegurar la calidad del producto detectando los errores, por lo que tras la realización de las mismas quedó la aplicación lista para su puesta en marcha y con mayor calidad.

CONCLUSIONES

La realización de la investigación arrojó los siguientes resultados que dan cumplimiento al objetivo propuesto:

Se realizó un estudio de los principales analizadores de registros de servidores proxy existentes demostrando la necesidad de implementar un sistema que permita realizar un análisis estadístico y por categorías de navegación de estos ficheros.

Se definieron los requisitos funcionales y no funcionales, y los casos de uso con sus descripciones permitiendo definir las características del sistema y sirviendo de guía para los flujos y fases posteriores.

Se analizó, diseñó e implementó la solución teniendo en cuenta las características definidas para el sistema, y la arquitectura del framework symfony en conjunto con la del analizador.

Se realizaron pruebas al sistema que permitieron asegurar la calidad del producto.

El sistema implementado como resultado de la investigación realizada será de gran utilidad para mejorar el control de la navegación de los usuarios por internet en el país, incidiendo favorablemente en su eficiencia y aprovechamiento.

RECOMENDACIONES

Al concluir el presente trabajo se recomienda:

- Agregar un módulo de administración de usuario que gestione el acceso a la aplicación con previa verificación de acceso y privilegios.
- Incrementar el número de reportes que brinda la aplicación.
- Estudiar y agregar un componente inteligente a la aplicación mediante la minería de datos con el objetivo de predecir el comportamiento de la navegación de los usuarios.

REFERENCIAS BIBLIOGRÁFICAS

- [1] PÉREZ Zaragoza, Patricia. *El mal uso de Internet preocupa a las empresas*. Las provincias [Internet]. Marzo 2006. Disponible en: <<http://www.lasprovincias.es/valencia/pg060313/economia/200603/13/VAL-ECO-281.html>> [Citado 10 Diciembre 2008].
- [2] SONFIELD, Matthew C. *Employee personal internet use: Costs and remedies* [Internet]. 2007. Disponible en: <<http://www.sbaer.uca.edu/research/usasbe/2007/data/papers/cases/101.pdf>> [Citado 10 Diciembre 2008].
- [3] DEL VALLE, Amaury E. *Estados Unidos bloquea Internet en Cuba I* [Internet]. Noviembre 2006. Disponible en: <<http://www.cubavsbloqueo.cu/Default.aspx?tabid=1497>> [Citado 10 Diciembre 2008].
- [4] BARRERA Palenzuela, Otniel; GARCÍA Pérez, Deiny. *Monitoreando el tráfico de Internet. Una solución al alcance de la mano*. 2007. Disponible en: <http://www.informaticahabana.com/evento_virtual/files/seg034.doc> [Citado 10 Diciembre 2008].
- [5] MIFSUD, Elvira. *Squid: servidor proxy-cache* [Internet]. 2008. Disponible en: <<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=589&mode=thread&order=0&thold=0&POSTNUKESID=f339a10ca45c0a045f79b297c43da954>> [Citado 10 Enero 2009].
- [6] Content Filtering Internet Proxy – SafeSquid. Disponible en: <<http://safesquid.com/html/portal.php?page=135>> [Citado 20 Enero 2009].
- [7] Content Filtering Internet Proxy – SafeSquid. Disponible en: <<http://www.safesquid.com/html/portal.php?page=93>> [Citado 20 Enero 2009].
- [8] AWStats - Free log file analyzer for advanced statistics. NLTechno. Disponible en: <<http://awstats.sourceforge.net/>> [Citado 20 Enero 2009].

- [9] Borg: The Collective. Disponible en: <<http://www.mrunix.net/webalizer/>> [Citado 20 Enero 2009]
- [10] SUÁREZ, José Alberto. *Análisis de ficheros log en GNU/Linux* [Internet]. 2007. Disponible en: < www.iberprensa.com/todolinux/articulos/TL65_42-46%20Taller_Log_.pdf > [Citado 22 Enero 2009].
- [11] SUÁREZ, José Alberto. *Análisis de ficheros log en GNU/Linux* [Internet]. 2007 Disponible en: < www.iberprensa.com/todolinux/articulos/TL65_42-46%20Taller_Log_.pdf > [Citado 22 Enero 2009].
- [12] Content Filtering Internet Proxy – SafeSquid. Disponible en: <<http://www.safesquid.com/html/portal.php?page=135>> [Citado 20 Enero 2009].
- [13] Sawmill - Universal log file analysis and reporting. Disponible en: < http://www.sawmill.net/formats/safesquid_combined.html > [Citado 22 Enero 2009].
- [14] Sawmill professional | log file analyser | streaming analysis | sendmail analysis. EMEAA Sales. Disponible en: <<http://www.sawmill.co.uk/pro.html>> [Citado 24 Enero 2009].
- [15] MARTIN, Ben. *Keeping an eye on your web proxy usage whit Squid-Graph* [Internet]. 2007. Disponible en: < <http://www.linux.com/feature/154700> > [Citado 24 Enero 2009].
- [16] WebTrends Log Analyzer. SOFTONIC. 2001. Disponible en: <<http://webtrends-log-analyzer.softonic.com/> > [Citado 24 Enero 2009].
- [17] MARTÍN Álvarez, Luis Orlando; GARCÍA Martínez, Yassier. *Sistema de reportes de la navegación por Internet*. Tesis (Pregrado en Ingeniería en Ciencias Informáticas). Habana, Cuba. Universidad de las Ciencias Informáticas (UCI). Junio 2007. 16 p.
- [18] MARTÍN Álvarez, Luis Orlando; GARCÍA Martínez, Yassier. *Sistema de reportes de la navegación por Internet*. Tesis (Pregrado en Ingeniería en Ciencias Informáticas). Habana, Cuba. Universidad de las Ciencias Informáticas (UCI). Junio 2007. 17 p.
- [19] MARTÍN Álvarez, Luis Orlando; GARCÍA Martínez, Yassier. *Sistema de reportes de la navegación por Internet*. Tesis (Pregrado en Ingeniería en Ciencias Informáticas). Habana, Cuba. Universidad de las Ciencias Informáticas (UCI). Junio 2007. 18 p.

- [20] BARQUERO Chaves, Byron I; MÉNDEZ Rodríguez, William. *Características del lenguaje Perl 5.0 y su aplicación como herramienta de desarrollo en la elaboración de un Servidor Web* [Internet]. 2007. Disponible en: < www.di-mare.com/adolfo/cursos/2007-2/pp-Perl.pdf> [Citado 30 Enero 2009]
- [21] URÍA Avellanal, Álvaro. *Comparativa entre Perl y Monad* [Internet]. 2005 Disponible en: < <http://fermat.movimage.com/docs/comparativaPerlMonad.pdf> > [Citado 30 Enero 2009].
- [22] ADR Formación – Cursos online y soluciones e-learning. ADRInfor. Disponible en: <http://www.adrformacion.com/curso/php/leccion1/Introduccion_php.htm> [Citado 1 Febrero 2009].
- [23] ÁLVAREZ, Miguel Ángel. *Introducción al manual del lenguaje PHP en su versión 5* [Internet]. 2004. Disponible en: <<http://www.desarrolloweb.com/articulos/1696.php>> [Citado 1 Febrero 2009].
- [24] BRADENBAUGH, Jerry. *Aplicaciones JavaScript*. 2000. Disponible en: < <http://biblioteca.uci.cu/titdigitales.htm#pro>> [Citado 1 Febrero 2009].
- [25] PostgreSQL: The world's most advanced open source database. Tinysofa. Disponible en: <<http://www.postgresql.org/docs/current/static/intro-what-is.html>> [Citado 1 Febrero 2009].
- [26] PostgreSQL PE | La base de datos libre más avanzada del mundo. Dravio. Disponible en: <http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf> [Citado 1 Febrero 2009].
- [27] The Apache HTTP Server Project. The Apache Software Foundation. Disponible en: <<http://httpd.apache.org/>> [Citado 2 Febrero 2009].
- [28] MORRISON, Aileen. *Servidor Web Apache* [Internet]. 2003. Disponible en: <<http://2001.encuentrolinux.cl/documentacion/ServidorWebApache.pdf.gz>> [Citado 2 Febrero 2009].
- [29] FABIEN Potencier, Francois Zaninotto. *Symfony, la guía definitiva* [Internet]. 2007. Disponible en: <www.librosweb.es/symfony/> [Citado 3 Febrero 2009].
- [30] HERMOSILLA Moreno, José Ramón; SÁNCHEZ Arce, Luis Enrique. *Interfaz de Administración Web para el Sistema de Filtrado Filpacon*. Tesis (Pregrado en Ingeniería en

Ciencias Informáticas). Habana, Cuba. Universidad de las Ciencias Informáticas (UCI).
Julio 2008. 24 p

- [31] MENDOZA Sánchez, María A. *Metodologías de desarrollo de software* [Internet]. 2004. Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html [Citado 2 Febrero 2009]
- [32] BOOCH, Grady; JACOBSON, Ivar; RUMBAUGH, James. *Proceso unificado de desarrollo de software*. 2000. Disponible en: <http://biblioteca.uci.cu/titdigitales.htm#igs> [Citado 2 Febrero 2009]
- [33] Soluciones y Propuestas RATIONAL, Servicios, Capacitación, Consultoría; Outsourcing de Testing - Software y Hardware IBM. 2007. Disponible en: <http://www.rational.com.ar/herramientas/rup.html> [Citado 2 Febrero 2009]
- [34] MENDOZA Sánchez, María A. *Metodologías de desarrollo de software* [Internet]. 2004. Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html [Citado 2 Febrero 2009]
- [35] POPKIN Software and Systems. *Modelado de sistemas con UML*. 2002. Disponible en: <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf> [Citado 3 Febrero 2009].
- [36] BOOCH, Grady; JACOBSON, Ivar; RUMBAUGH, James. *El lenguaje unificado de modelado. Manual de referencia*. 2002. Disponible en: <http://biblioteca.uci.cu/titdigitales.htm#igs> [Citado 3 Febrero 2009].
- [37] POPKIN Software and Systems. *Modelado de sistemas con UML*. 2002. Disponible en: <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf> [Citado 3 Febrero 2009].
- [38] Kubuntu-es | Portal hispano de Kubuntu. Disponible en: <http://www.kubuntu-es.org/wiki/desarrollo-programacion/programas-desarrollo-libres> [Citado 4 Febrero 2009]

- [39] Plataforma Eclipse: Comunidad en español de Eclipse IDE. Disponible en: <<http://plataformaeclipse.com/manuales/>> [Citado 4 Febrero 2009]
- [40] Kubuntu-es | Portal hispano de Kubuntu. Disponible en: <<http://www.kubuntu-es.org/wiki/desarrollo-programacion/programas-desarrollo-libres>> [Citado 4 Febrero 2009]
- [41] About the Eclipse Foundation. Disponible en: <<http://www.eclipse.org/org/>> [Citado 4 Febrero 2009]
- [42] VIZCAÍNO, Aurora; GARCÍA, Félix Oscar; CABALLERO, Ismael. *Trabajando con Visual Paradigm for UML* [Internet]. Disponible en: <<http://personales.unican.es/ruizfr/is1/doc/lab/01/is1-p01-trans.pdf>> [Citado 5 Febrero 2009]
- [43] Increase productivity, communication, and collaboration using UML visual modeling platform. Disponible en: <<http://www.visual-paradigm.com/product/vpuml/>> [Citado 5 Febrero 2009].
- [44] LÓPEZ, Emiliano. *Servidor Proxy Squid* [Internet]. 2006. Disponible en: <<http://linuxemb.wdfiles.com/local--files/tutoriales/squid.pdf>> [Citado 6 Febrero 2009].
- [45] ALLENDE, Claudio; TITO, Vilma. *Proxy. Patrón estructural* [Internet]. Disponible en: <<http://dc.exa.unrc.edu.ar/nuevodc/materias/sistemas/2007/Patrones/1181918751/ProxyRes2.doc>> [Citado 6 Febrero 2009].
- [46] Tenon Intersystems. Disponible en: <<http://www.tenon.com/support/webten/papers/squidlog.shtml>> [Citado 15 Febrero 2009].
- [47] PRESSMAN, Roger S. *Ingeniería de software. Un enfoque práctico*. 2002. Disponible en: <<http://biblioteca.uci.cu/titdigitales.htm#igs>> [Citado 30 Abril 2009].

BIBLIOGRAFÍA

PÉREZ Zaragoza, Patricia. *El mal uso de Internet preocupa a las empresas*. Las provincias [Internet]. Marzo 2006. Disponible en: <<http://www.lasprovincias.es/valencia/pg060313/economia/200603/13/VAL-ECO-281.html>>

SONFIELD, Matthew C. *Employee personal internet use: Costs and remedies* [Internet]. 2007 Disponible en: <<http://www.sbaer.uca.edu/research/usasbe/2007/data/papers/cases/101.pdf>>

DEL VALLE, Amaury E. *Estados Unidos bloquea Internet en Cuba I* [Internet]. Noviembre 2006. Disponible en: <<http://www.cubavsbloqueo.cu/Default.aspx?tabid=1497>>

BARRERA Palenzuela, Otniel; GARCÍA Pérez, Deiny. *Monitoreando el tráfico de Internet. Una solución al alcance de la mano*. 2007. Disponible en: <http://www.informicahabana.com/evento_virtual/files/seg034.doc>

MIFSUD, Elvira. *Squid: servidor proxy-cache* [Internet]. 2008. Disponible en: <<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=589&mode=thread&order=0&thold=0&POSTNUKESID=f339a10ca45c0a045f79b297c43da954>>

Content Filtering Internet Proxy – SafeSquid. Disponible en: <http://safesquid.com/>

AWStats - Free log file analyzer for advanced statistics. NLTechno. Disponible en: <<http://awstats.sourceforge.net/>>

Borg: The Collective. Disponible en: <http://www.mrunix.net/>

SUÁREZ, José Alberto. *Análisis de ficheros log en GNU/Linux* [Internet]. 2007. Disponible en: <www.iberprensa.com/todolinux/articulos/TL65_42-46%20Taller_Log.pdf>

Sawmill - Universal log file analysis and reporting. Disponible en: <<http://www.sawmill.net/>>

Sawmill professional | log file analyser | streaming analysis | sendmail analysis. EMEA Sales. Disponible en: <<http://www.sawmill.co.uk/>>.

MARTIN, Ben. *Keeping an eye on your web proxy usage whit Squid-Graph* [Internet]. 2007. Disponible en: <<http://www.linux.com/feature/154700>>

WebTrends Log Analyzer. SOFTONIC. 2001. Disponible en: <<http://webtrends-log-analyzer.softonic.com/>>

MARTÍN Álvarez, Luis Orlando; GARCÍA Martínez, Yassier. *Sistema de reportes de la navegación por Internet*. Tesis (Pregrado en Ingeniería en Ciencias Informáticas). Habana, Cuba. Universidad de las Ciencias Informáticas (UCI). Junio 2007. 16 p.

BARQUERO Chaves, Byron I; MÉNDEZ Rodríguez, William. *Características del lenguaje Perl 5.0 y su aplicación como herramienta de desarrollo en la elaboración de un Servidor Web* [Internet]. 2007. Disponible en: <www.di-mare.com/adolfo/cursos/2007-2/pp-Perl.pdf>

URÍA Avellanal, Álvaro. *Comparativa entre Perl y Monad* [Internet]. 2005 Disponible en: <<http://fermat.movimage.com/docs/comparativaPerlMonad.pdf>>

ADR Formación – Cursos online y soluciones e-learning. ADRInfor. Disponible en: <<http://www.adrformacion.com/>>.

ÁLVAREZ, Miguel Ángel. *Introducción al manual del lenguaje PHP en su versión 5* [Internet]. 2004. Disponible en: <<http://www.desarrolloweb.com/articulos/1696.php>>

BRADENBAUGH, Jerry. *Aplicaciones JavaScript*. 2000. Disponible en: <<http://biblioteca.uci.cu/titdigitales.htm#pro>>

PostgreSQL: The world's most advanced open source database. Tinysofa. Disponible en: <<http://www.postgresql.org/>>

[27] The Apache HTTP Server Project. The Apache Software Foundation. Disponible en: <<http://httpd.apache.org/>> [Citado 2 Febrero 2009].

MORRISON, Aileen. *Servidor Web Apache* [Internet]. 2003. Disponible en: <<http://2001.encuentrolinux.cl/documentacion/ServidorWebApache.pdf.gz>>

FABIEN Potencier, Francois Zaninotto. *Symfony, la guía definitiva* [Internet]. 2007. Disponible en: <www.librosweb.es/symfony/>

HERMOSILLA Moreno, José Ramón; SÁNCHEZ Arce, Luis Enrique. *Interfaz de Administración Web para el Sistema de Filtrado Filpacon*. Tesis (Pregrado en Ingeniería en Ciencias Informáticas). Habana, Cuba. Universidad de las Ciencias Informáticas (UCI). Julio 2008. 24 p

MENDOZA Sánchez, María A. *Metodologías de desarrollo de software* [Internet]. 2004. Disponible en: <http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html>

BOOCH, Grady; JACOBSON, Ivar; RUMBAUGH, James. *Proceso unificado de desarrollo de software*. 2000. Disponible en: <<http://biblioteca.uci.cu/titdigitales.htm#igs>>

Soluciones y Propuestas RATIONAL, Servicios, Capacitación, Consultoría; Outsourcing de Testing - Software y Hardware IBM. 2007. Disponible en: <<http://www.rational.com.ar/herramientas/rup.html>>

MENDOZA Sánchez, María A. *Metodologías de desarrollo de software* [Internet]. 2004. Disponible en: <http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html>

POPKIN Software and Systems. *Modelado de sistemas con UML*. 2002. Disponible en: <<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>>

Kubuntu-es | Portal hispano de Kubuntu. Disponible en: <<http://www.kubuntu-es.org/wiki/desarrollo-programacion/programas-desarrollo-libres>>

Plataforma Eclipse: Comunidad en español de Eclipse IDE. Disponible en: <<http://plataformaeclipse.com/manuales/>>

About the Eclipse Foundation. Disponible en: <<http://www.eclipse.org/org/>>

VIZCAÍNO, Aurora; GARCÍA, Félix Oscar; CABALLERO, Ismael. *Trabajando con Visual Paradigm for UML* [Internet]. Disponible en: <<http://personales.unican.es/ruizfr/is1/doc/lab/01/is1-p01-trans.pdf>>

Increase productivity, communication, and collaboration using UML visual modeling platform. Disponible en: <<http://www.visual-paradigm.com/product/vpuml/>>

LÓPEZ, Emiliano. *Servidor Proxy Squid* [Internet]. 2006. Disponible en: <<http://linuxemb.wdfiles.com/local--files/tutoriales/squid.pdf>>.

ALLENDE, Claudio; TITO, Vilma. *Proxy. Patrón estructural* [Internet]. Disponible en: <<http://dc.exa.unrc.edu.ar/nuevodic/materias/sistemas/2007/Patrones/1181918751/ProxyRes2.doc>> .

Tenon Intersystems. Disponible en: <<http://www.tenon.com/support/webten/papers/squidlog.shtml>>

PRESSMAN, Roger S. *Ingeniería de software. Un enfoque práctico*. 2002. Disponible en: <<http://biblioteca.uci.cu/titdigitales.htm#igs>>.

ANEXOS 1

Casos de uso expandidos

Caso de uso:	Reporte por categoría
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario desea observar el reporte por categoría.
Referencia:	R2
CU asociados:	Analizar log
Precondiciones	Que se halla ejecutado el caso de uso Analizar log.
Flujo normal de eventos	
Sección “Reportes por categoría”	
Acción del actor	Respuesta del sistema
1. El usuario selecciona el reporte “Reporte por categoría”. 2. El usuario selecciona los datos.	1.1 El sistema muestra un formulario en el que pide seleccionar los datos del reporte (fecha inicio, fecha fin y modo de ordenamiento). 2.1 Si el reporte devuelve datos 2.2 El sistema muestra reporte en forma de gráfica y más detallado en forma de tabla.
Flujo Alterno	

	2.1. a Se emite un mensaje informando que no hay reportes para la fecha seleccionada y regresa a la acción 2 de esta sección.
Poscondiciones:	El usuario observará el reporte en forma de gráfica y más detallado en forma de tabla
Prioridad:	Secundario
Especificaciones:	
Complementarias:	

Caso de uso:	Reporte por tiempo
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario desea observar el reporte por tiempo.
Referencia:	R3
CU asociados:	Analizar log
Precondiciones	Que se halla ejecutado el caso de uso Analizar log.
Flujo normal de eventos	
Sección “Reportes por tiempo”	
Acción del actor	Respuesta del sistema
1. El usuario selecciona el reporte “Reporte por	1.1 El sistema muestra un formulario en el que

tiempo".	pide seleccionar los datos del reporte (fecha inicio, fecha fin y modo de ordenamiento).
2. El usuario selecciona los datos.	<p>2.1 Si el reporte devuelve datos</p> <p>2.2 El sistema muestra reporte en forma de gráfica y más detallado en forma de tabla.</p>
Flujo Alterno	
	2.1.a Se emite un mensaje informando que no hay reportes para la fecha seleccionada y regresa a la acción 2 de esta sección.
Poscondiciones:	El usuario observará el reporte en forma de gráfica y más detallado en forma de tabla
Prioridad:	Secundario
Especificaciones:	
Complementarias:	

Caso de uso:	Reporte por IP
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario desea observar el reporte por IP.
Referencia:	R4
CU asociados:	Analizar log

Precondiciones	Que se halla ejecutado el caso de uso Analizar log.
----------------	---

Flujo normal de eventos

Sección “Reportes por IP”

Acción del actor	Respuesta del sistema
-------------------------	------------------------------

<p>1. El usuario selecciona el reporte “Reporte por IP”.</p> <p>2. El usuario selecciona los datos.</p>	<p>1.1 El sistema muestra un formulario en el que pide seleccionar los datos del reporte (fecha inicio, fecha fin y modo de ordenamiento).</p> <p>2.1 Si el reporte devuelve datos</p> <p>2.2 El sistema muestra reporte en forma de gráfica y más detallado en forma de tabla.</p>
---	---

Flujo Alterno

	<p>2.1.a Se emite un mensaje informando que no hay reportes para la fecha seleccionada y regresa a la acción 2 de esta sección.</p>
--	---

<p>Poscondiciones:</p>	<p>El usuario observará el reporte en forma de gráfica y más detallado en forma de tabla</p>
------------------------	--

<p>Prioridad:</p>	<p>Secundario</p>
-------------------	-------------------

<p>Especificaciones:</p>	
--------------------------	--

<p>Complementarias:</p>	
-------------------------	--

Caso de uso:	Reporte por usuarios
--------------	----------------------

Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario desea observar el reporte por usuarios.
Referencia:	R5
CU asociados:	Analizar log
Precondiciones	Que se halla ejecutado el caso de uso Analizar log.
Flujo normal de eventos	
Sección “Reportes por usuarios”	
Acción del actor	Respuesta del sistema
<p>1. El usuario selecciona el reporte “Reporte por usuarios”.</p> <p>2. El usuario selecciona los datos.</p>	<p>1.1 El sistema muestra un formulario en el que pide seleccionar los datos del reporte (fecha inicio, fecha fin y modo de ordenamiento).</p> <p>2.1 Si el reporte devuelve datos</p> <p>2.2 El sistema muestra reporte en forma de gráfica y más detallado en forma de tabla.</p>
Flujo Alternativo	
	<p>2.1.a Se emite un mensaje informando que no hay reportes para la fecha seleccionada y regresa a la acción 2 de esta sección.</p>
Poscondiciones:	El usuario observará el reporte en forma de gráfica y más detallado en forma de tabla

Prioridad:	Secundario
Especificaciones:	
Complementarias:	

Caso de uso:	Reporte por directorio de la URL
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario desea observar el reporte por directorio de la URL.
Referencia:	R6
CU asociados:	Analizar log
Precondiciones	Que se halla ejecutado el caso de uso Analizar log.

Flujo normal de eventos

Sección “Reportes por directorio de la URL”

Acción del actor	Respuesta del sistema
1. El usuario selecciona el reporte “Reporte por directorio de la URL”.	1.1 El sistema muestra un formulario en el que pide seleccionar los datos del reporte (fecha inicio, fecha fin y modo de ordenamiento).
2. El usuario selecciona los datos.	2.1 Si el reporte devuelve datos 2.2 El sistema muestra reporte en forma de gráfica y más detallado en forma de tabla.

Flujo Alterno	
	2.1.a Se emite un mensaje informando que no hay reportes para la fecha seleccionada y regresa a la acción 2 de esta sección.
Poscondiciones:	El usuario observará el reporte en forma de gráfica y más detallado en forma de tabla
Prioridad:	Secundario
Especificaciones:	
Complementarias:	

Caso de uso:	Reporte por navegación general
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario desea observar el reporte por navegación general.
Referencia:	R7
CU asociados:	Analizar log
Precondiciones	Que se halla ejecutado el caso de uso Analizar log.
Flujo normal de eventos	
Sección “Reportes por navegación general”	

Acción del actor	Respuesta del sistema
<p>1. El usuario selecciona el reporte “Reporte por navegación general”.</p> <p>2. El usuario selecciona los datos.</p>	<p>1.1 El sistema muestra un formulario en el que pide seleccionar los datos del reporte (fecha inicio, fecha fin y modo de ordenamiento).</p> <p>2.1 Si el reporte devuelve datos</p> <p>2.2 El sistema muestra reporte en forma de gráfica y más detallado en forma de tabla.</p>
Flujo Alterno	
	<p>2.1.a Se emite un mensaje informando que no hay reportes para la fecha seleccionada y regresa a la acción 2 de esta sección.</p>
Poscondiciones:	El usuario observará el reporte en forma de gráfica y más detallado en forma de tabla
Prioridad:	Secundario
Especificaciones:	
Complementarias:	

Caso de uso:	Reporte de categorías por usuario
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario desea observar el reporte por categorías por usuario.

Referencia:	R9
CU asociados:	Analizar log
Precondiciones	Que se halla ejecutado el caso de uso Analizar log.
Flujo normal de eventos	
Sección “Reportes de categorías por usuario”	
Acción del actor	Respuesta del sistema
<p>1. El usuario selecciona el reporte “Reporte de categorías por usuario”.</p> <p>2. El usuario selecciona los datos.</p>	<p>1.1 El sistema muestra un formulario en el que pide seleccionar los datos del reporte (fecha inicio, fecha fin y modo de ordenamiento).</p> <p>2.1 Si el reporte devuelve datos</p> <p>2.2 El sistema muestra reporte en forma de gráfica y más detallado en forma de tabla.</p>
Flujo Alterno	
	2.1.a Se emite un mensaje informando que no hay reportes para la fecha seleccionada y regresa a la acción 2 de esta sección.
Poscondiciones:	El usuario observará el reporte en forma de gráfica y más detallado en forma de tabla
Prioridad:	Secundario
Especificaciones:	

Complementarias:	
------------------	--

Caso de uso:	Reporte de tiempo por usuario
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario desea observar el reporte de tiempo por usuario.
Referencia:	R10
CU asociados:	Analizar log
Precondiciones	Que se halla ejecutado el caso de uso Analizar log.

Flujo normal de eventos

Sección "Reporte de tiempo por usuario"

Acción del actor	Respuesta del sistema
1. El usuario selecciona el reporte "Reporte de tiempo por usuario". 2. El usuario selecciona los datos.	1.1 El sistema muestra un formulario en el que pide seleccionar los datos del reporte (fecha inicio, fecha fin y modo de ordenamiento). 2.1 Si el reporte devuelve datos 2.2 El sistema muestra reporte en forma de gráfica y más detallado en forma de tabla.

Flujo Alterno

	2.1.a Se emite un mensaje informando que no
--	---

	hay reportes para la fecha seleccionada y regresa a la acción 2 de esta sección.
Poscondiciones:	El usuario observará el reporte en forma de gráfica y más detallado en forma de tabla
Prioridad:	Secundario
Especificaciones:	
Complementarias:	

Caso de uso:	Reporte de Ips por usuarios
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario desea observar el reporte de Ips por usuarios
Referencia:	R11
CU asociados:	Analizar log
Precondiciones	Que se halla ejecutado el caso de uso Analizar log.
Flujo normal de eventos	
Sección “Reporte de Ips por usuarios”	
Acción del actor	Respuesta del sistema
1. El usuario selecciona el reporte “Reporte de Ips por usuarios”.	1.1 El sistema muestra un formulario en el que pide seleccionar los datos del reporte (fecha

2. El usuario selecciona los datos.	<p>inicio, fecha fin y modo de ordenamiento).</p> <p>2.1 Si el reporte devuelve datos</p> <p>2.2 El sistema muestra reporte en forma de gráfica y más detallado en forma de tabla.</p>
Flujo Alterno	
	2.1.a Se emite un mensaje informando que no hay reportes para la fecha seleccionada y regresa a la acción 2 de esta sección.
Poscondiciones:	El usuario observará el reporte en forma de gráfica y más detallado en forma de tabla
Prioridad:	Secundario
Especificaciones:	
Complementarias:	

Caso de uso:	Reporte de directorios de URL por usuario
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario desea observar el reporte de directorios de URL por usuario
Referencia:	R12
CU asociados:	Analizar log

Precondiciones	Que se halla ejecutado el caso de uso Analizar log.
Flujo normal de eventos	
Sección “Reporte de directorios de URL por usuario”	
Acción del actor	Respuesta del sistema
<p>1. El usuario selecciona el reporte “Reporte de directorios de URL por usuario”.</p> <p>2. El usuario selecciona los datos.</p>	<p>1.1 El sistema muestra un formulario en el que pide seleccionar los datos del reporte (fecha inicio, fecha fin y modo de ordenamiento).</p> <p>2.1 Si el reporte devuelve datos</p> <p>2.2 El sistema muestra reporte en forma de gráfica y más detallado en forma de tabla.</p>
Flujo Alterno	
	2.1.a Se emite un mensaje informando que no hay reportes para la fecha seleccionada y regresa a la acción 2 de esta sección.
Poscondiciones:	El usuario observará el reporte en forma de gráfica y más detallado en forma de tabla
Prioridad:	Secundario
Especificaciones:	
Complementarias:	

Caso de uso:	Reporte de navegación general por usuario
--------------	--

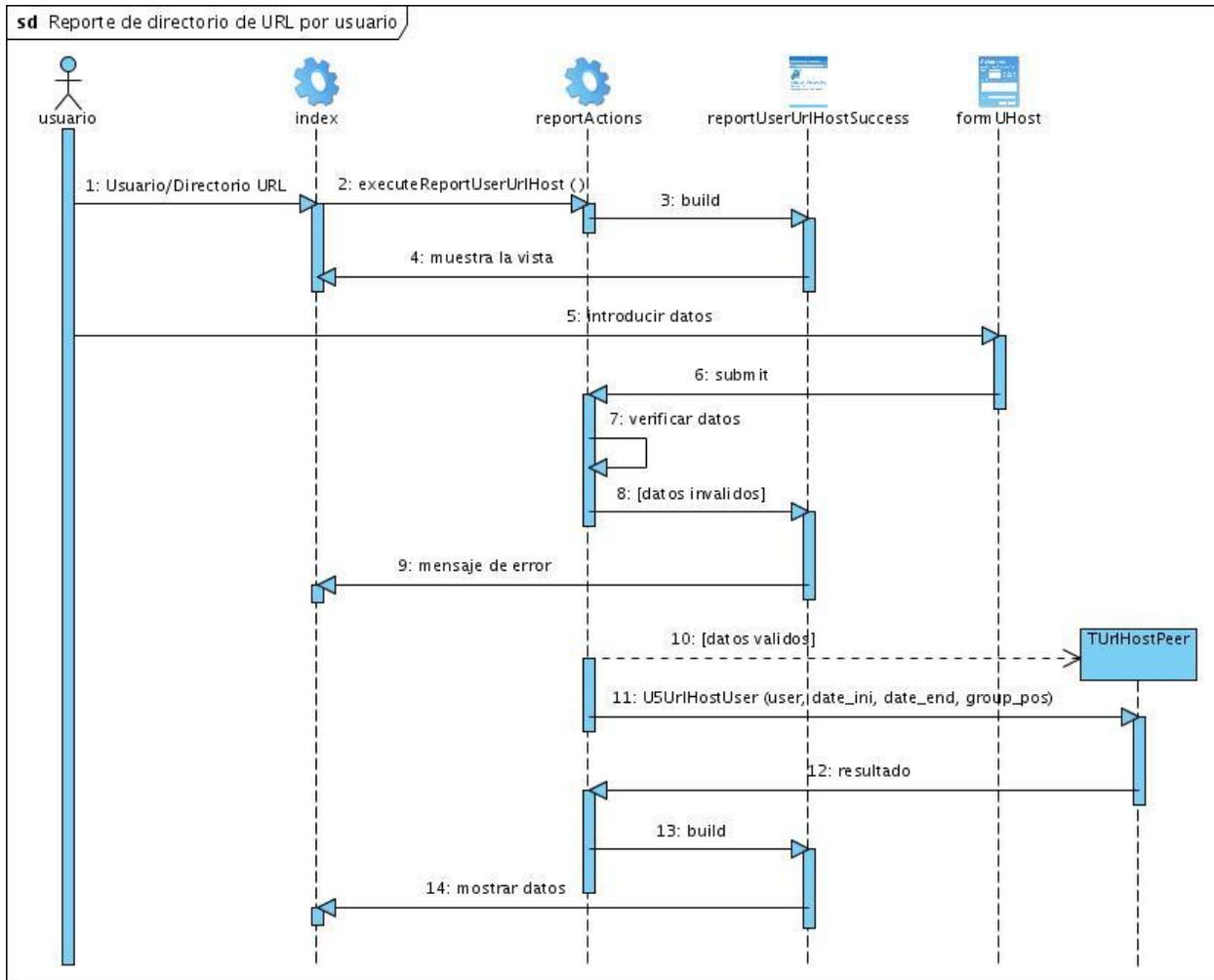
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario desea observar el reporte de navegación general por usuario
Referencia:	R13
CU asociados:	Analizar log
Precondiciones	Que se halla ejecutado el caso de uso Analizar log.
Flujo normal de eventos	
Sección “Reporte de navegación general por usuario”	
Acción del actor	Respuesta del sistema
<p>1. El usuario selecciona el reporte “Reporte de navegación general por usuario”.</p> <p>2. El usuario selecciona los datos.</p>	<p>1.1 El sistema muestra un formulario en el que pide seleccionar los datos del reporte (fecha inicio, fecha fin y modo de ordenamiento).</p> <p>2.1 Si el reporte devuelve datos</p> <p>2.2 El sistema muestra reporte en forma de gráfica y más detallado en forma de tabla.</p>
Flujo Alterno	
	2.1.a Se emite un mensaje informando que no hay reportes para la fecha seleccionada y regresa a la acción 2 de esta sección.
Poscondiciones:	El usuario observará el reporte en forma de gráfica y más detallado en forma de tabla

Prioridad:	Secundario
Especificaciones:	
Complementarias:	

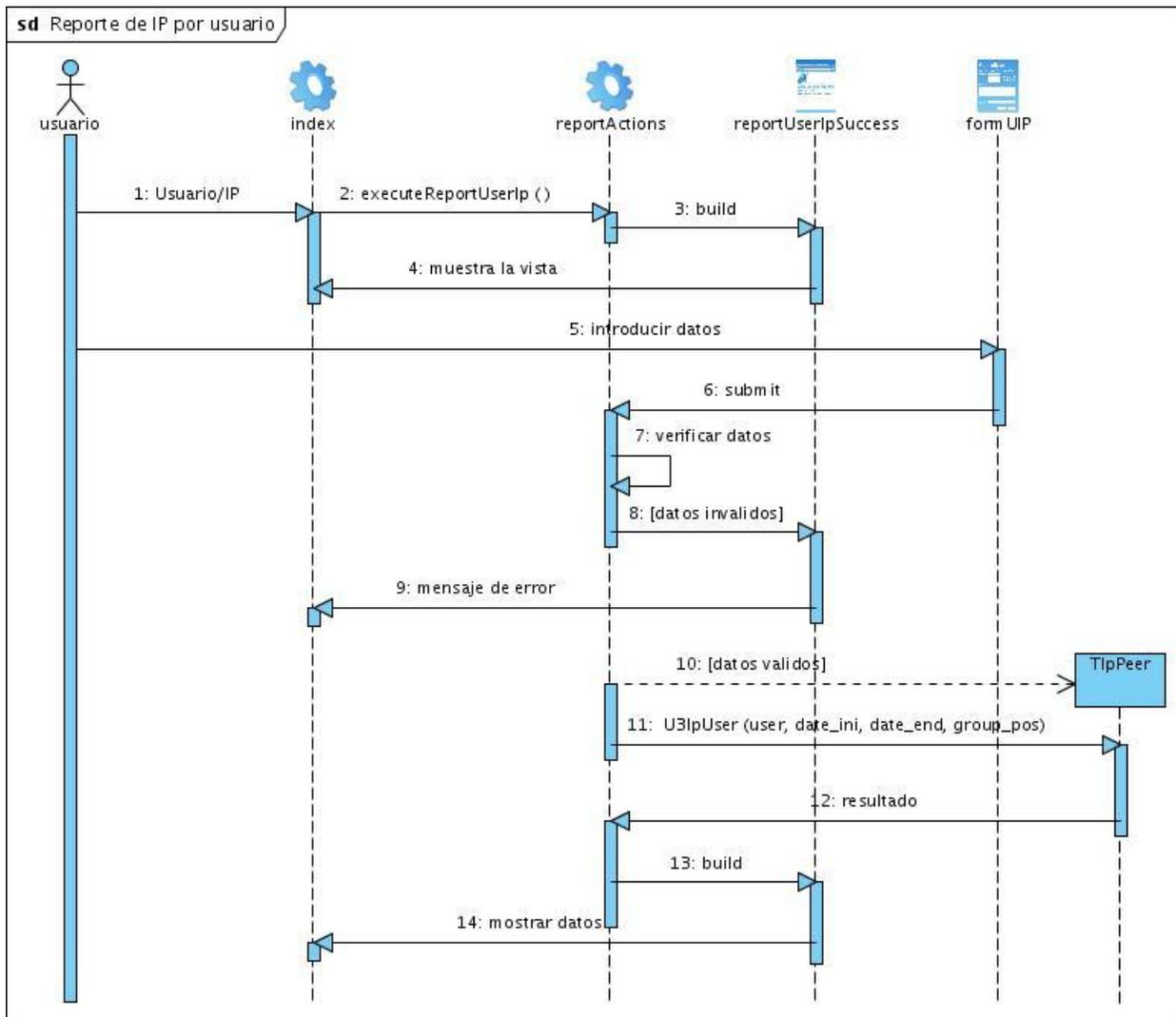
ANEXOS 2

Diagramas de secuencia

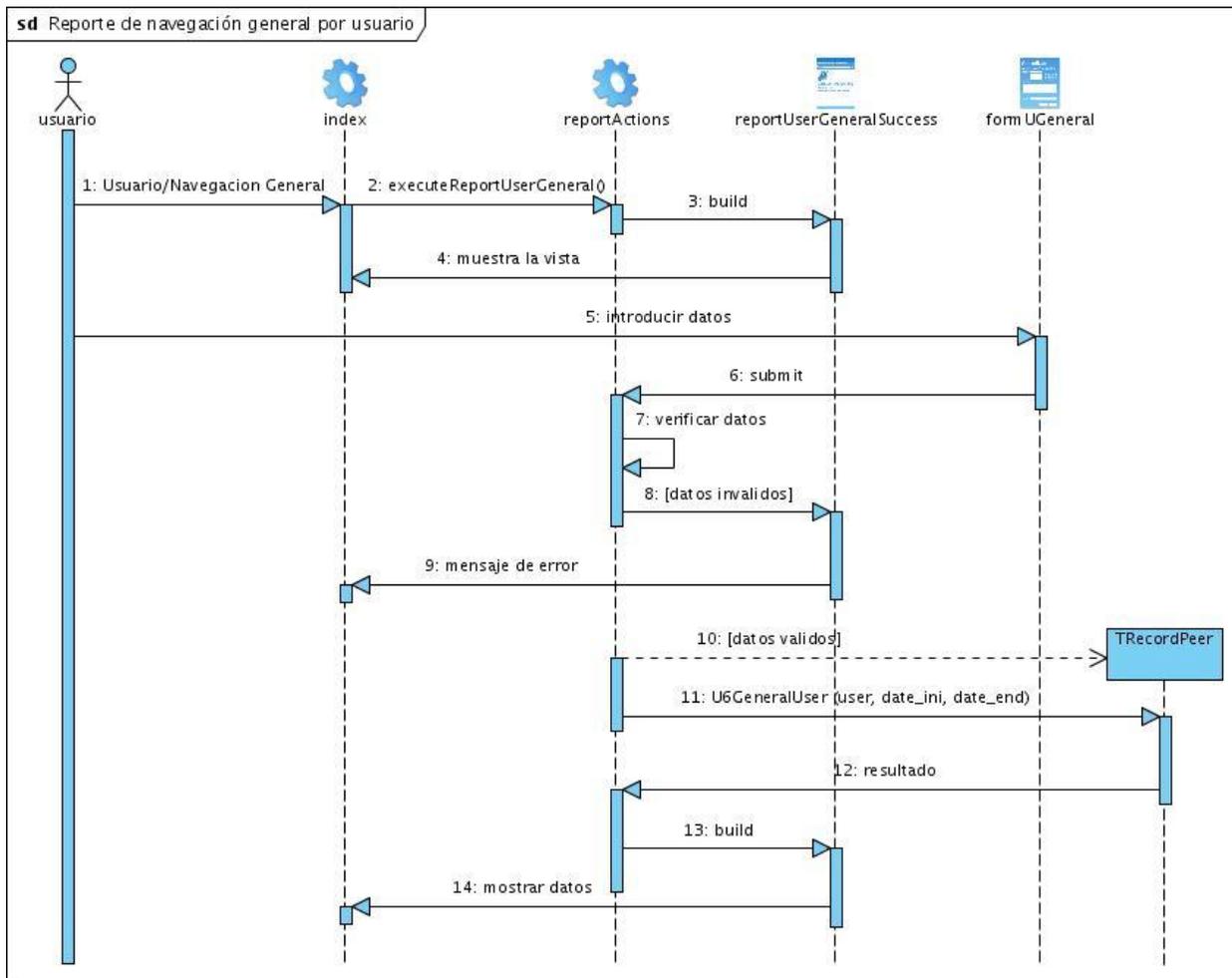
CU Reporte de directorio de URL por usuario



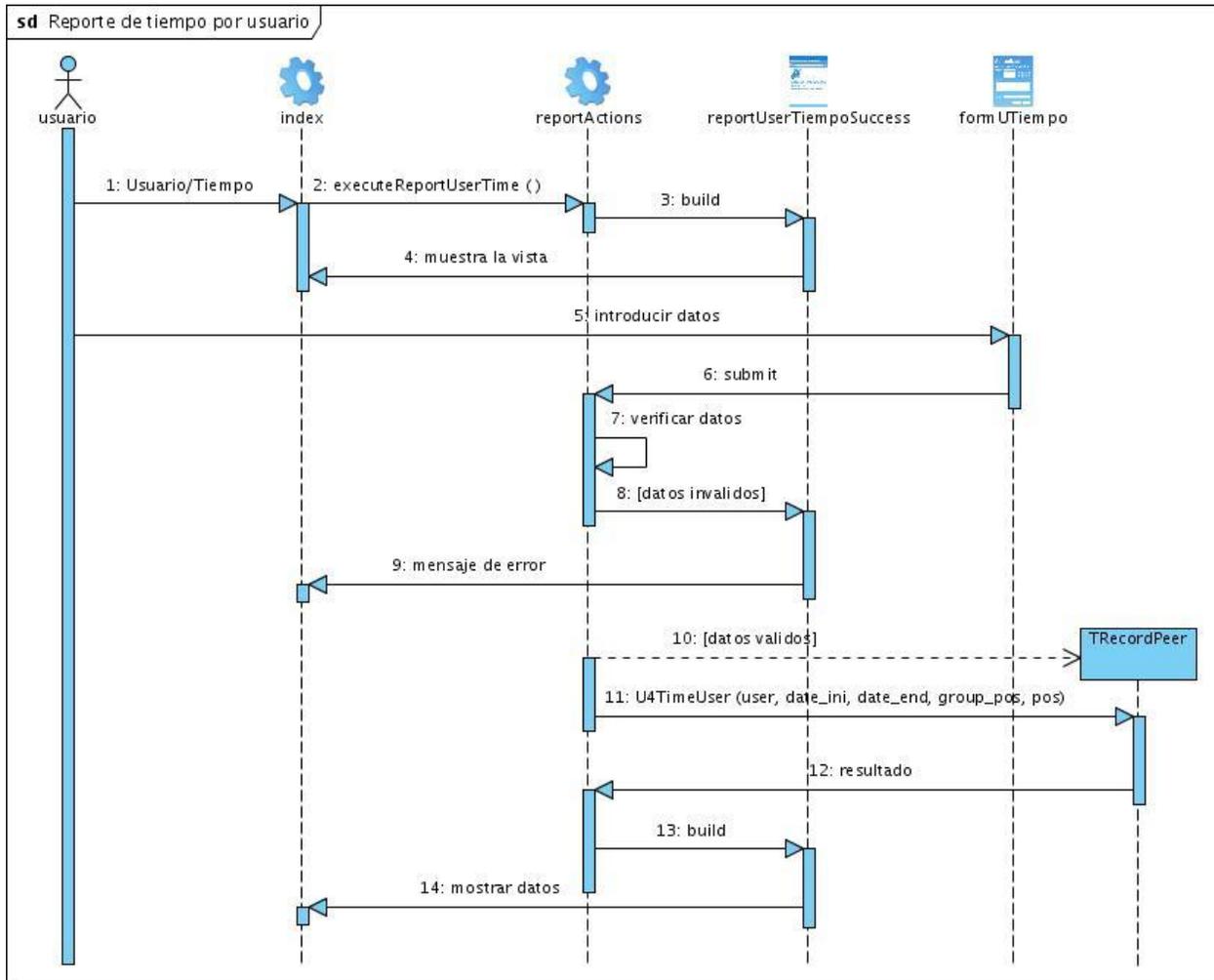
CU Reporte de IP por usuario



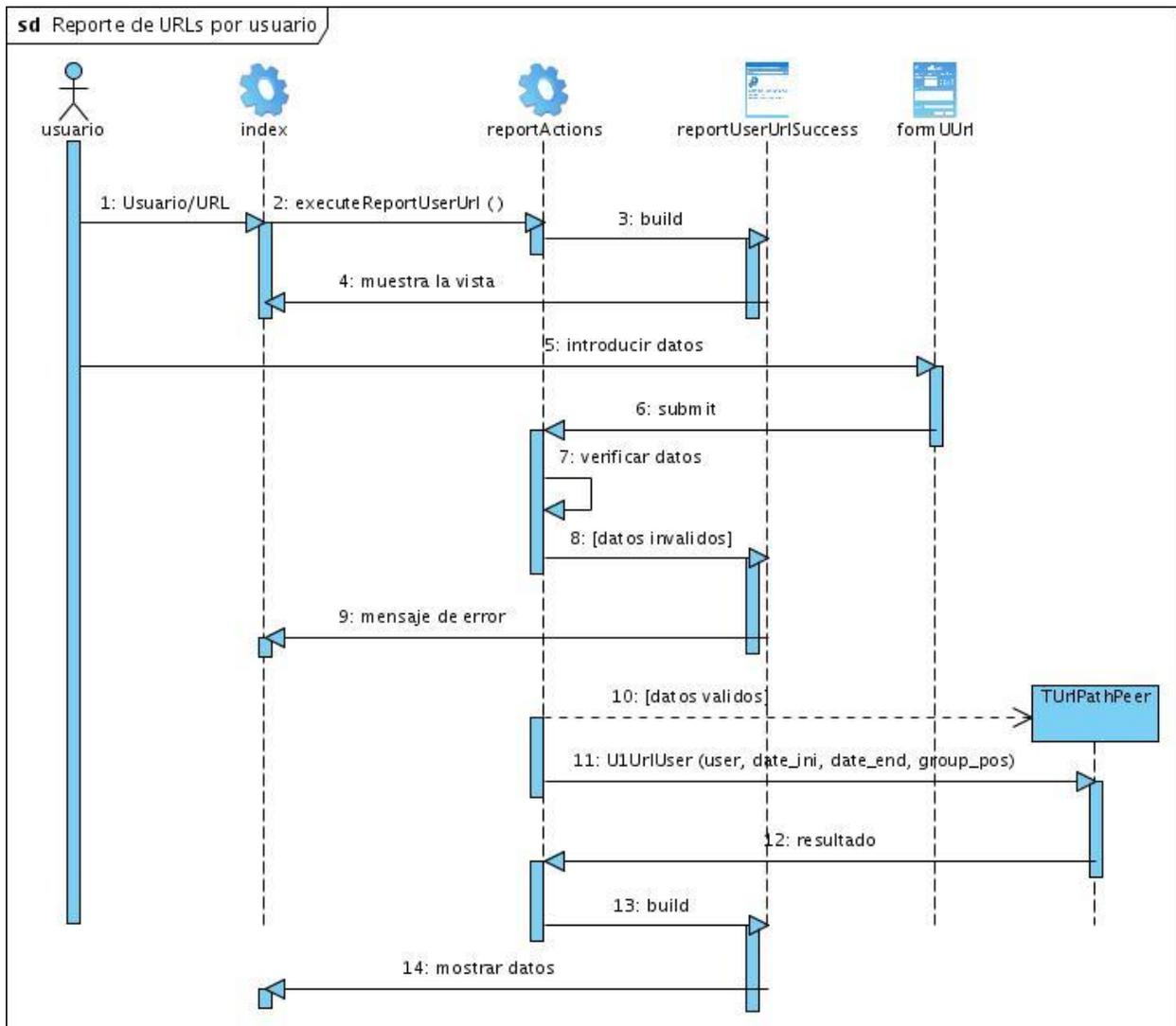
CU Reporte de navegación general por usuario



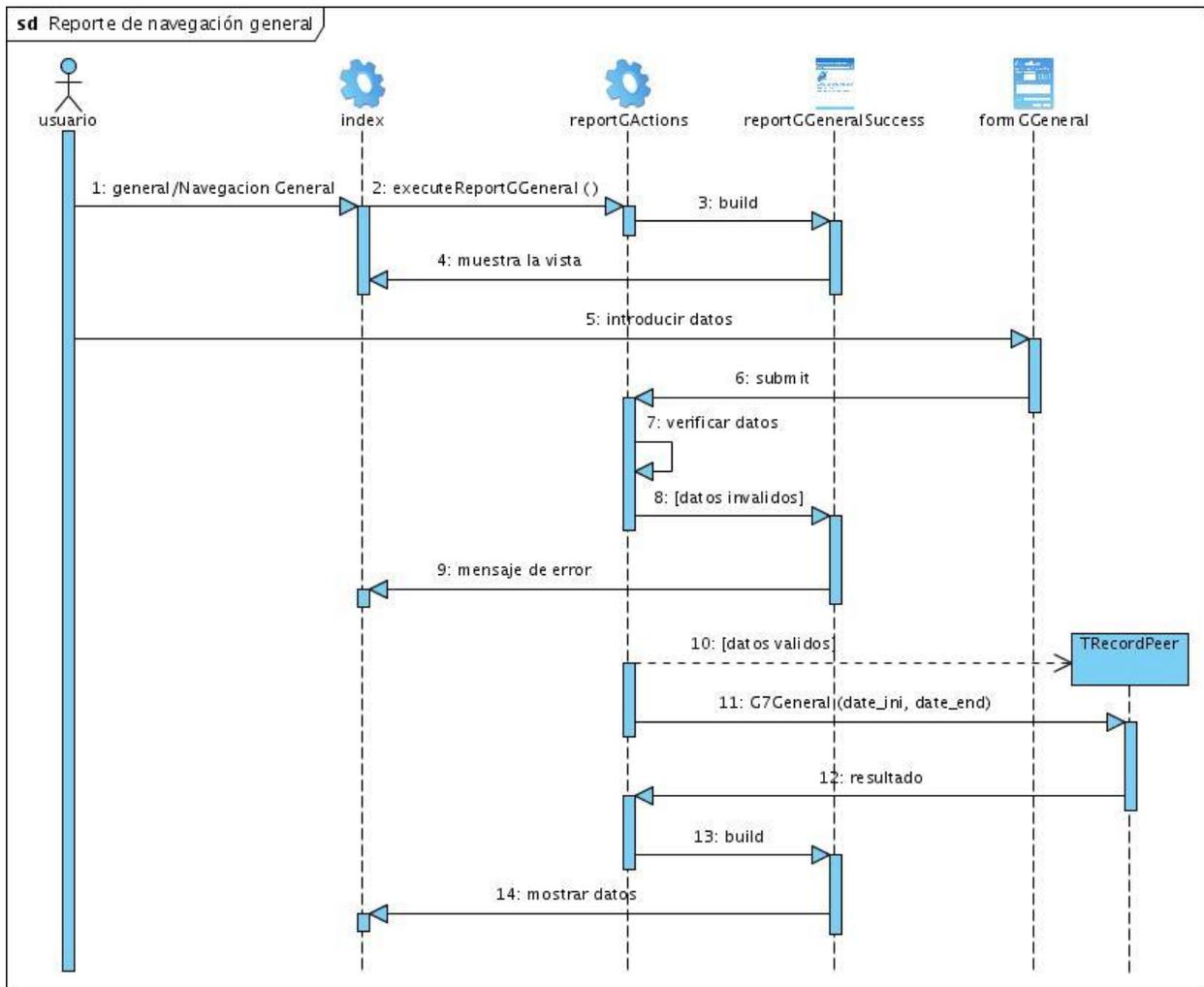
CU Reporte de tiempo por usuario



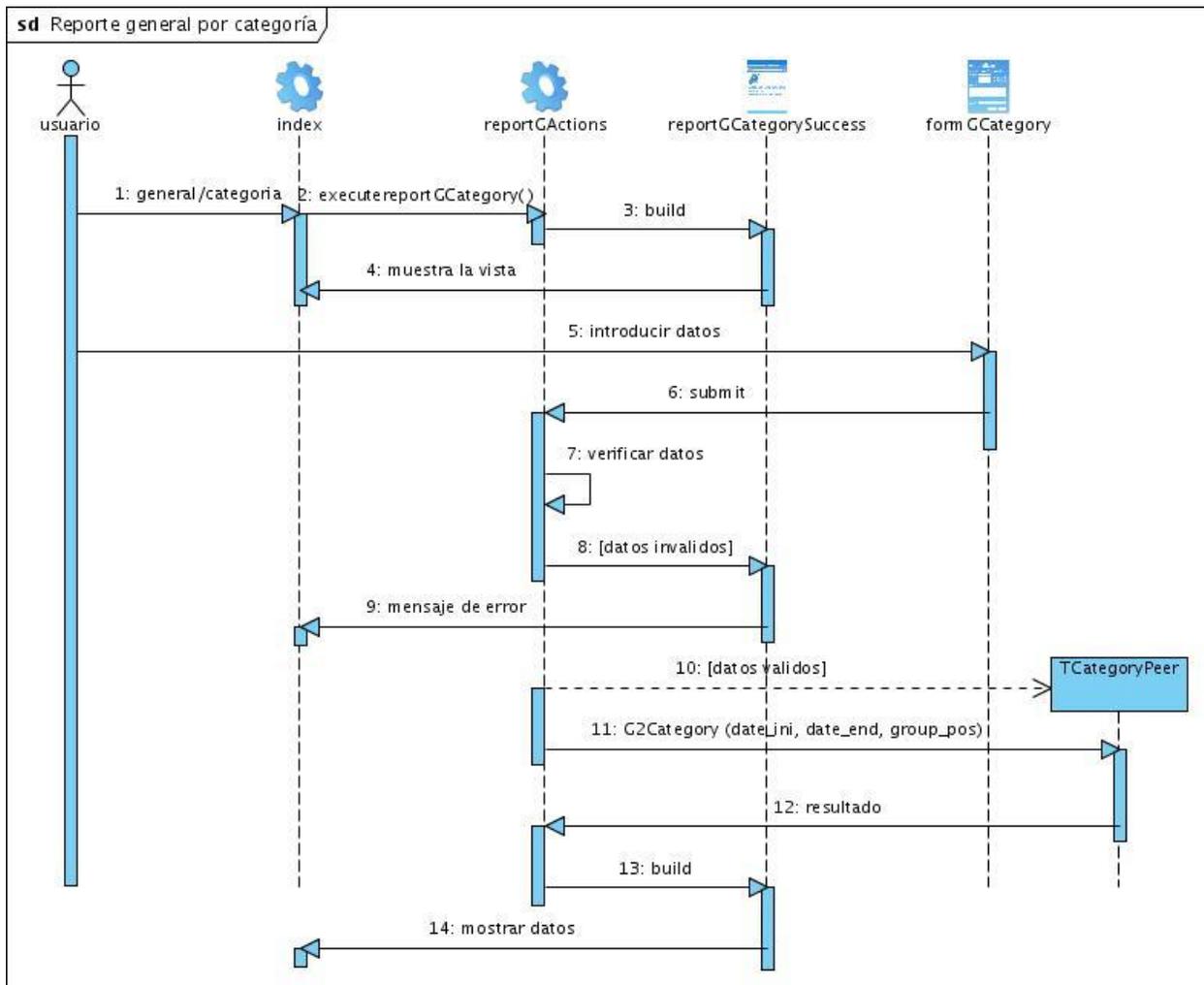
CU Reporte de URLs por usuario



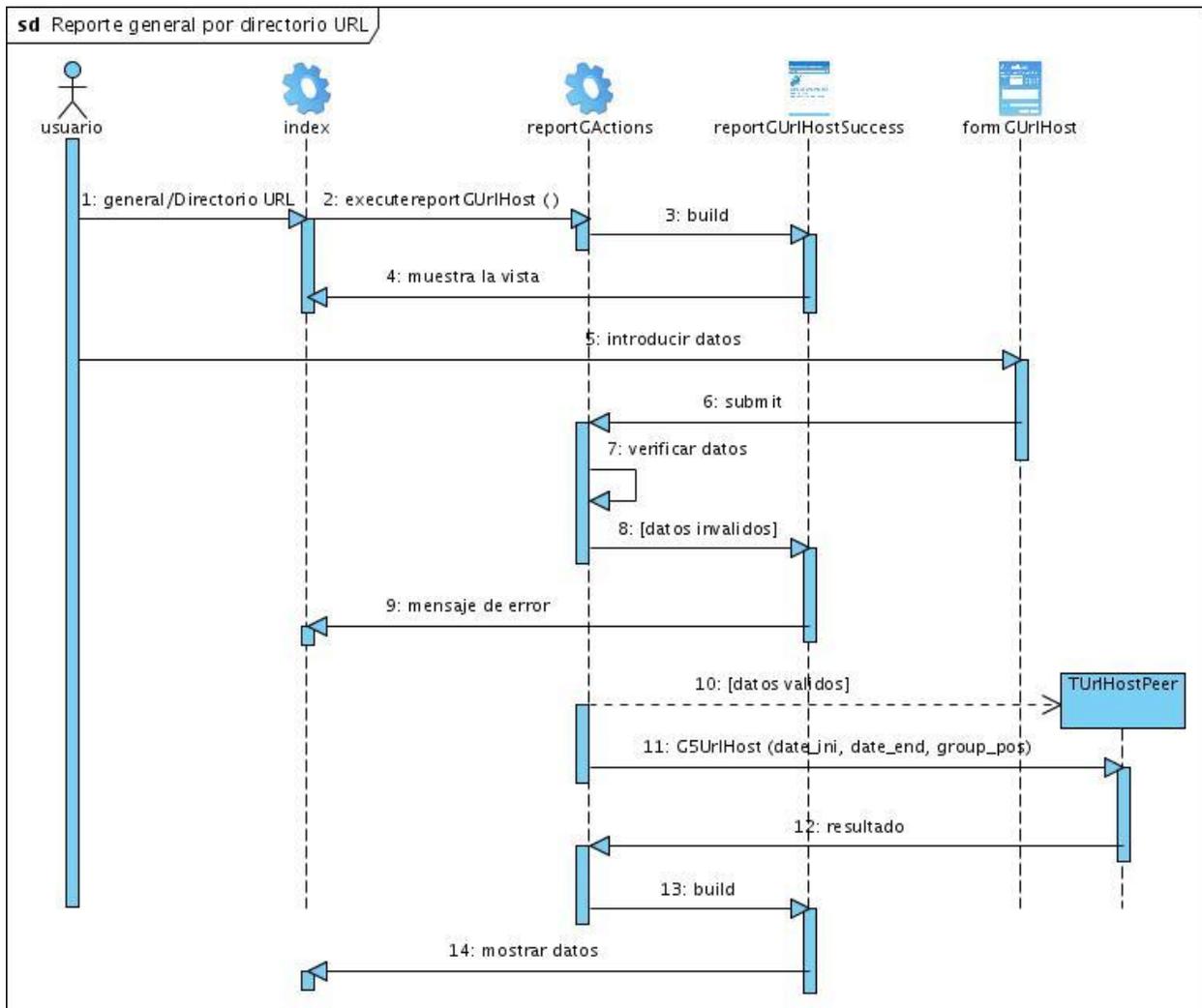
CU Reporte de navegación general



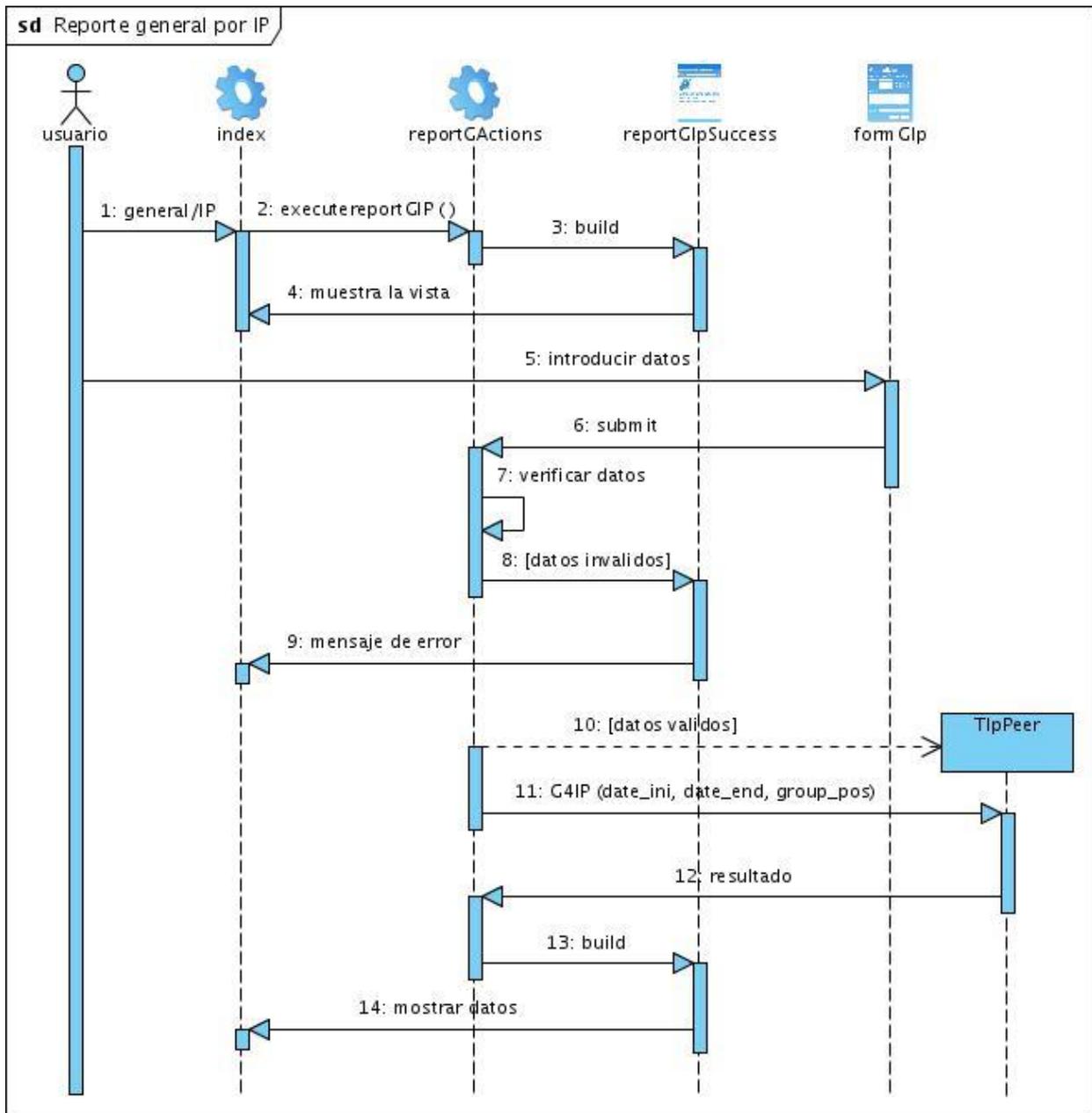
CU Reporte general por categoría



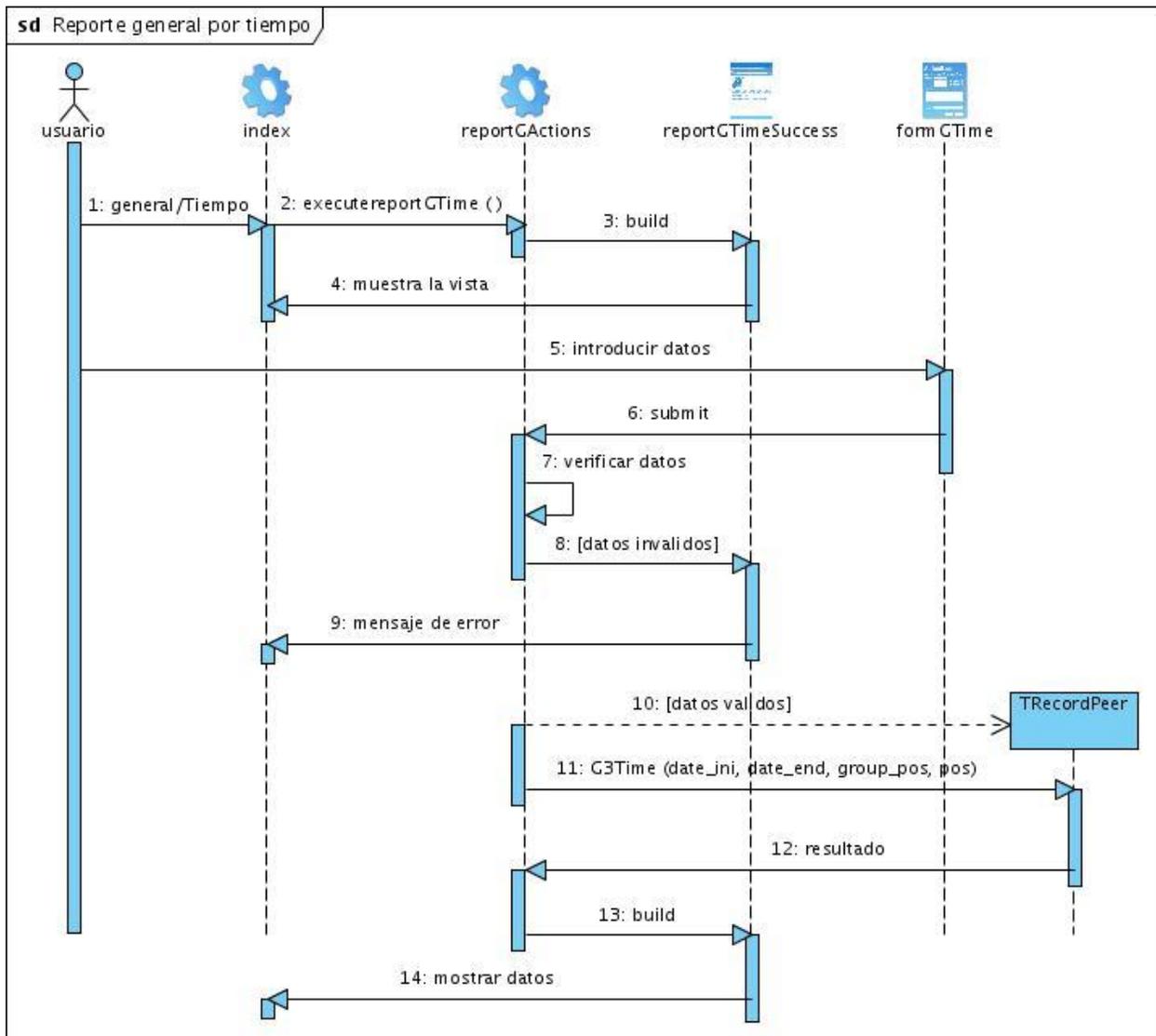
CU Reporte general por directorio URL



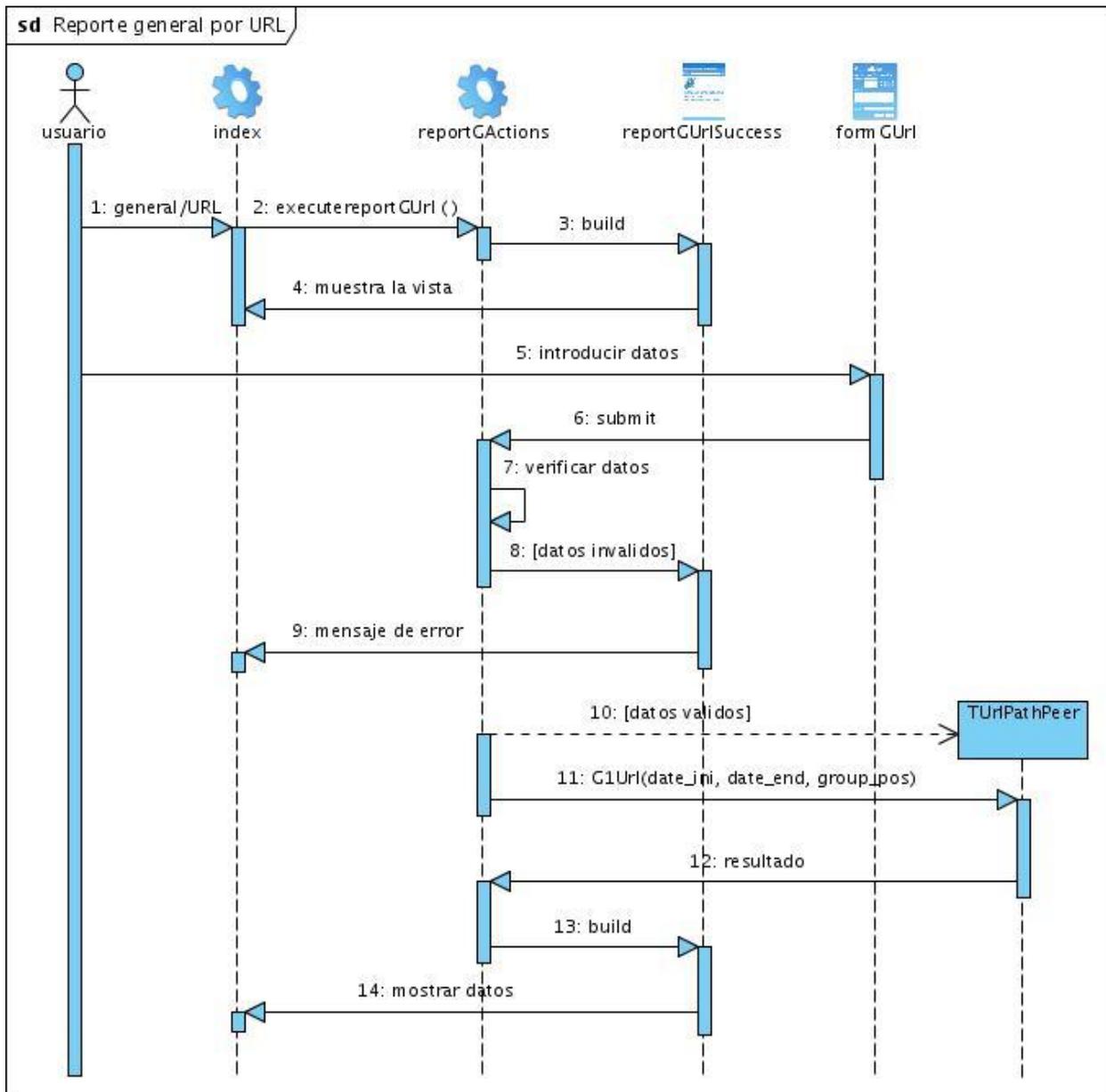
CU Reporte general por IP



CU Reporte general por tiempo



CU Reporte general por URL



CU Reporte general por usuario

