



# **UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

## **Facultad 1**

**Título:** Desarrollo del Módulo de Gestión de Reportes Estadísticos para el sistema AiresProxyAudit.

**Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.**

**Autores:** Evelyn Labrada Oduardo

Yaniel Lázaro Aragón Barreda

**Tutor:** Ing. Tamara Betancourt Santana

Ing. Yonny Mondelo Hernández

**La Habana, 20 de Junio de 2013.**

## Declaración de Autoría.

Declaramos que somos los únicos autores del trabajo “Desarrollo del Módulo de Gestión de Reportes Estadísticos para el sistema AiresProxyAudit” 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 \_\_\_\_\_ del año 2013.

Autores:

Tutores:

---

Evelyn Labrada Oduardo

---

Ing. Tamara Betancourt Santana

---

Yaniel Lázaro Aragón Barreda

---

Ing. Yonny Mondelo Hernández

## **Agradecimientos.**

Le agradezco a mi querida madre por haber forjado mi carácter y mi entrega al estudio, además de haberme guiado y apoyado durante todos estos años. A mi padre por ayudarme a no ser completamente igual a mi madre y enseñarme a tener un poco más de alegría y optimismo cómo él.

A todas aquellas personas que de alguna forma u otra me ayudaron y me enseñaron a ser cada día mejor.

Y por último y no menos importante agradezco al amor de mi vida M, por haberse mantenido a mi lado durante todos estos años, por haber creído en mí, por haberme acompañado a soñar y por todos los sacrificios que realizó por mi bienestar, Te Amo.

**Evelyn Labrada Oduardo.**

A mi mamá por todos los años de esfuerzo y sacrificio, por creer en mí, por el apoyo brindado y la constante preocupación por mi formación tanto académica como personal. Por el ejemplo siempre ha sido para mí, aunque de corazón espero mi carácter no se siga pareciendo cada día más al suyo.

A mi familia en general por el apoyo durante todos mis años de estudio.

A todos mis amigos que por ser pocos se convierten en los mejores, siendo una parte esencial en mi vida.  
Raylith, Dayami, Mary, Neyvis, Yosvany, Julio Cesar.

En especial agradezco a Arletys, a quien considero la hermana que no tengo, por aguantarme durante 5 años como compañero de aula, de mesa, de trabajo, de estudio y de fiesta.

A Vicente por soportar mis pesadeces y ayudarme a entender mejor el mundo que nos rodea.

A todas aquellas personas que me enseñaron con su ejemplo a ser mejor ser humano.

Agradezco a la FEU, organización que me enseñó a crecerme ante los problemas, me dio la posibilidad de hablar en nombre de los estudiantes, me sacó de la burbuja del estudio y me llevó a relacionarme más con las personas, me demostró que con perseverancia todo es posible y creó en mí una persona más organizada para con el trabajo.

Doy gracias por haber nacido en el seno de una Revolución, dirigida por quien para mí es el más grande visionario del siglo XX. Agradezco a Fidel por haber creado esta universidad que me forjó como ingeniero en Ciencias Informáticas.

**Yaniel Lázaro Aragón Barreda**

Gracias a nuestros tutores. En especial a Tamara por haber soportado nuestras malcriadeces y vagancia.

**Ambos**

## **Dedicatoria.**

A las personas que más quiero, mi madre y mi novio.

Evelyn Labrada Oduardo

A quien siempre será un ejemplo para mí y sé que desde donde quiera que esté se encuentra orgullosa de la persona que soy. A Moraima Barreda Collazo.

## Resumen.

Con el desarrollo de las Tecnologías de la Información y las Comunicaciones las empresas que cuentan con acceso a *Internet* presentan una mejora de sus condiciones de trabajo. Esto se debe a la gran cantidad de información existente en la misma y las nuevas formas de comunicación que ofrece. Sin embargo, el costo de utilizar este servicio es elevado, por lo que se hace necesario controlarlo.

La Universidad de las Ciencias Informáticas brinda este servicio a través de un sistema de cuotas y con el fin de supervisar el acceso al mismo se realizan auditorías a grupos de usuarios (GU) y por áreas de trabajo (AT) específicas. Para realizar dicho proceso se utiliza la herramienta Sawmill, la cual es privativa y ralentiza el análisis pues no permite el agrupamiento por GU o AT. En este trabajo se presenta un Módulo Generador de Reportes Estadísticos (MGRE) para el sistema AiresProxyAudit capaz de realizar un análisis de los registros de navegación del servidor *proxy* con agrupamiento. Dicho módulo basa su funcionamiento en el uso de un componente desarrollado con C++, el cual genera un grupo de reportes aplicando filtros que pueden ser seleccionados desde una aplicación web. Los casos de pruebas realizados al MGRE, rigieron el desarrollo del mismo permitiendo la entrega al cliente de una solución libre y confiable que pueda ser utilizada para la realización de estudios sociales que faciliten la toma de decisiones y la implantación de nuevas políticas de seguridad de forma ágil.

**Palabras Claves:** áreas de trabajo, auditorías, grupos de usuarios, *Internet*, registros, servidor *proxy*.

# Índice de Contenido.

<b>INTRODUCCIÓN.....</b>	<b>9</b>
<b>CAPÍTULO 1 “FUNDAMENTACIÓN TEÓRICA.” .....</b>	<b>13</b>
1.1. INTRODUCCIÓN.....	13
1.2. CONCEPTOS BÁSICOS.....	13
1.3. HERRAMIENTAS PARA EL ANÁLISIS Y CONTROL DE LOGS DE SERVIDORES PROXY.....	14
1.3.1. <i>Ámbito Internacional.</i> .....	14
1.3.2. <i>Ámbito Nacional.</i> .....	16
1.4. METODOLOGÍAS, TECNOLOGÍAS, LIBRERÍAS Y HERRAMIENTAS PROPUESTAS PARA EL DESARROLLO DEL SISTEMA.....	20
1.4.1. <i>Metodología de desarrollo.</i> .....	20
1.4.2. <i>Framework de desarrollo.</i> .....	22
1.4.3. <i>Lenguajes utilizados.</i> .....	23
1.4.4. <i>Gestor de Base de Datos.</i> .....	26
1.4.5. <i>Librerías utilizadas en el funcionamiento del MGRE.</i> .....	26
1.4.6. <i>Entornos Integrados de Desarrollo (IDEs)</i> .....	27
1.4.7. <i>Lenguaje y herramienta de Modelado.</i> .....	28
1.5. CONCLUSIONES.....	29
<b>CAPÍTULO 2 “CARACTERÍSTICAS DEL SISTEMA.” .....</b>	<b>30</b>
2.1. INTRODUCCIÓN.....	30
2.2. CARACTERÍSTICAS DEL SISTEMA.....	30
2.3. MODELO DEL DOMINIO.....	30
2.4. ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE.....	32
2.4.1. <i>Requisitos funcionales.</i> .....	32
2.4.2. <i>Requisitos no funcionales.</i> .....	34
2.5. MODELO DE CASOS DE USO DEL SISTEMA.....	35
2.6. PATRONES UTILIZADOS EN EL DESARROLLO DE LA APLICACIÓN.....	40
2.6.1. <i>Patrones de casos de uso.</i> .....	40
2.6.2. <i>Patrones Generales de Software para la Asignación de Responsabilidades (GRASP).</i> .....	41
2.6.3. <i>Patrones Gang of Four (GOF)</i> .....	42
2.6.4. <i>Patrón arquitectónico Modelo-Vista-Controlador.</i> .....	43
2.7. DIAGRAMA DE CLASES DEL DISEÑO.....	44
2.8. DIAGRAMAS DE SECUENCIA.....	46
2.9. MODELO DE DATOS.....	49

2.10.	DIAGRAMA DE DESPLIEGUE.....	51
2.11.	CONCLUSIONES.....	52
<b>CAPÍTULO 3 “IMPLEMENTACIÓN Y PRUEBAS DE VALIDACIÓN.” .....</b>		<b>53</b>
3.1.	INTRODUCCIÓN.....	53
3.2.	DIAGRAMA DE COMPONENTES.....	53
3.3.	ESTÁNDAR DE CODIFICACIÓN.....	58
3.4.	PANTALLAS PRINCIPALES DEL MGRE.....	61
3.5.	VALIDACIÓN DEL SISTEMA .....	63
3.5.1.	<i>Pruebas funcionales.....</i>	<i>63</i>
3.5.2.	<i>Pruebas de seguridad.....</i>	<i>66</i>
3.5.3.	<i>Prueba de carga y estrés .....</i>	<i>66</i>
3.7.	CONCLUSIONES.....	68
<b>CONCLUSIONES GENERALES.....</b>		<b>69</b>
<b>RECOMENDACIONES.....</b>		<b>70</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>71</b>
<b>BIBLIOGRAFÍA.....</b>		<b>76</b>
<b>GLOSARIO DE TÉRMINOS.....</b>		<b>81</b>
<b>ANEXOS.....</b>		<b>83</b>
ANEXO 1.	REPORTE GRÁFICO DE URLS MÁS VISITADAS.....	83
ANEXO 2.	REPORTE GRÁFICO DE VISITANTES Y CONSUMO POR MESES.....	84
ANEXO 3.	DIAGRAMA DE CLASES DEL DISEÑO DEL CASO DE USO EXPORTAR CONFIGURACIÓN.....	85
ANEXO 4.	DIAGRAMA DE CLASES DEL DISEÑO DEL CASO DE USO EXPORTAR REPORTE A PDF.....	86
ANEXO 5.	DIAGRAMA DE CLASES DEL DISEÑO DEL CASO DE USO GESTIONAR REPORTES GUARDADOS.....	87
ANEXO 6.	DIAGRAMA DE CLASES DEL DISEÑO DEL CASO DE USO GUARDAR CONFIGURACIÓN.....	88
ANEXO 7.	DIAGRAMA DE CLASES DEL DISEÑO DEL CASO DE USO GUARDAR REPORTE.....	89
ANEXO 8.	DIAGRAMA DE CLASES DEL DISEÑO DEL CASO DE USO IMPORTAR CONFIGURACIÓN.....	90
ANEXO 9.	DIAGRAMA DE CLASES DEL DISEÑO DEL CASO DE USO MOSTRAR ENTIDADES.....	91
ANEXO 10.	DIAGRAMA DE CLASES DEL DISEÑO DEL CASO DE USO ORDENAR RESULTADO.....	92
ANEXO 11.	DIAGRAMA DE SECUENCIA DEL CASO DE USO CARGAR REPORTE GUARDADO.....	93
ANEXO 12.	DIAGRAMA DE SECUENCIA DEL CASO DE USO ELIMINAR REPORTE GUARDADO.....	93
ANEXO 13.	DIAGRAMA DE SECUENCIA DEL CASO DE USO EXPORTAR CONFIGURACIÓN.....	94
ANEXO 14.	DIAGRAMA DE SECUENCIA DEL CASO DE USO GUARDAR CONFIGURACIÓN.....	94
ANEXO 15.	DIAGRAMA DE SECUENCIA DEL CASO DE USO GUARDAR REPORTE.....	95

ANEXO 16. DIAGRAMA DE SECUENCIA DEL CASO DE USO IMPORTAR CONFIGURACIÓN.....	95
ANEXO 17. DIAGRAMA DE SECUENCIA DEL CASO DE USO MOSTRAR ENTIDADES.....	96
ANEXO 18. PRUEBAS DE GESTIÓN DE SESIONES.....	96
ANEXO 19. COMPROBACIÓN DEL SISTEMA DE AUTENTICACIÓN.....	97
ANEXO 20. COMPROBACIÓN DEL SISTEMA DE AUTENTICACIÓN.....	99

## Índice de Tablas.

TABLA 1. COMPARACIÓN ENTRE LAS HERRAMIENTAS HOMÓLOGAS ANALIZADAS.....	19
TABLA 2. REQUISITOS FUNCIONALES DEL MGRE PARA EL SISTEMA AIRESPROXYAUDIT.....	33
TABLA 3. CASO DE USO AUTENTICAR USUARIO.....	37
TABLA 4. CASO DE USO SELECCIONAR FILTROS.....	38
TABLA 5. CASO DE USO GENERAR REPORTES.....	39
TABLA 6. CASO DE USO MOSTRAR GRÁFICOS.....	40
TABLA 7. CASO DE PRUEBA DEL CU AUTENTICAR USUARIO.....	64
TABLA 8. CASO DE PRUEBA DEL CU GENERAR REPORTES.....	65
TABLA 9. CASO DE PRUEBA DEL CU GESTIONAR REPORTES GUARDADOS.....	65
TABLA 10. RESULTADOS ARROJADOS POR LA PRUEBA DE CARGA Y ESTRÉS.....	67

## Índice de Imágenes.

IMAGEN 1. MODELO DE DOMINIO DEL MGRE DEL SISTEMA AIRESPROXYAUDIT.....	31
IMAGEN 2. MODELO DE CASOS DE USOS DEL SISTEMA DEL MGRE PARA EL SISTEMA AIRESPROXYAUDIT.....	35
IMAGEN 3. DIAGRAMA DE CLASES DEL DISEÑO DEL CASO DE USO AUTENTICAR USUARIO.....	44
IMAGEN 4. DIAGRAMA DE CLASES DEL DISEÑO DEL CASO DE USO SELECCIONAR FILTROS.....	45
IMAGEN 5. DIAGRAMA DE CLASES DEL DISEÑO DEL CASO DE USO GENERAR REPORTES.....	45
IMAGEN 6. DIAGRAMA DE CLASES DEL DISEÑO DEL CASO DE USO MOSTRAR GRÁFICOS.....	46
IMAGEN 7. DIAGRAMA DE SECUENCIA DEL CASO DE USO AUTENTICAR USUARIO.....	47
IMAGEN 8. DIAGRAMA DE SECUENCIA DEL CASO DE USO SELECCIONAR FILTROS.....	47
IMAGEN 9. DIAGRAMA DE SECUENCIA DEL CASO DE USO GENERAR REPORTES.....	48
IMAGEN 10. DIAGRAMA DE SECUENCIA DEL CASO DE USO MOSTRAR GRÁFICOS.....	48
IMAGEN 11. MODELO DE DATOS DEL MGRE.....	50
IMAGEN 12. DIAGRAMA DE DESPLIEGUE DEL MGRE PARA EL SISTEMA AIRESPROXYAUDIT.....	51
IMAGEN 13. DIAGRAMA COMPONENTES GENERAL AGRUPADOS POR PAQUETES.....	54

IMAGEN 14. PAQUETE CONTROLLER.....	55
IMAGEN 15. PAQUETE REPORTE. ....	55
IMAGEN 16. PAQUETE PDF. ....	56
IMAGEN 17. PAQUETE GRAFICOS. ....	56
IMAGEN 18. PAQUETE REPOSITORY.....	57
IMAGEN 19. PAQUETE DOCUMENT. ....	58
IMAGEN 20. PAQUETE AIRESPAUDIT.....	58
IMAGEN 21. FRAGMENTO DE CÓDIGO DEL MGRE CON INDENTACIÓN.....	60
IMAGEN 22. EJEMPLO DE COMENTARIO USADO EN EL MGRE.....	60
IMAGEN 23. INTERFAZ DE ACCESO AL MGRE. SOLUCIÓN AL CASO DE USO AUTENTICAR USUARIO.....	61
IMAGEN 24. INTERFAZ DE SELECCIÓN DE FILTROS PARA REPORTES. SOLUCIÓN AL CASO DE USO SELECCIONAR FILTROS.....	61
IMAGEN 25. INTERFAZ DE REPORTES GENERADOS. SOLUCIÓN AL CASO DE USO GENERAR REPORTES. ....	62
IMAGEN 26. INTERFAZ DE REPORTE GRÁFICO POR IP. SOLUCIÓN AL CASO DE USO MOSTRAR GRÁFICOS.....	62

## Introducción.

Con la evolución de las tecnologías, *Internet* se ha convertido en uno de los mayores exponentes de la comunicación mundial. La abundante información contenida en la misma ha puesto el conocimiento al alcance de un clic, propiciando el surgimiento de nuevos estilos de trabajo basados en la colaboración al acortar las distancias entre los involucrados. Por ello, las empresas que acceden a este servicio pueden contribuir a una mejora de sus condiciones de trabajo y su productividad. Sin embargo, debido al costo que representa para las instituciones el uso de este servicio es necesario controlarlo. Este proceso de control permite supervisar y comparar los resultados obtenidos contra los resultados esperados originalmente. Además, asegura que la acción dirigida se esté llevando a cabo de acuerdo con los planes de la organización y dentro de los límites de la estructura organizacional; posibilitando la implantación de mejoras continuamente.

Entre los métodos para un adecuado control empresarial se encuentran las auditorías. Estas pueden ser clasificadas en internas o externas en dependencia de la procedencia del ente que la realice. El objetivo de las mismas es el examen de todas las anotaciones contables a fin de comprobar su exactitud, así como la veracidad de los estados o situaciones que dichas anotaciones producen (Holmes, 1984). En términos informáticos se consideran un proceso formal ejecutado por los especialistas del área de auditoría y de informática. Esta se orienta a la verificación y aseguramiento de las políticas y procedimientos establecidos para el manejo y uso adecuado de las Tecnologías de la Información y las Comunicaciones (TIC) en la organización (SER, 2012). El auditor; autor principal de este proceso, tiene como objetivo observar la exactitud, integridad y autenticidad de registros, documentos, medios tangibles e intangibles.

Otra manera de controlar y regular el uso adecuado de *Internet* consiste en el mecanismo de regulación de acceso, el cual disminuye el riesgo de ataques de virus a la red, aumenta la velocidad de navegación y mejora el desempeño empresarial. Estas regulaciones pueden ser: por tiempo; acordada de manera “amigable” o regulada por programas, el filtrado de contenido; aplicado por niveles que van desde el URL<sup>1</sup> y palabras claves hasta el tipo de información a la cual accede el usuario o por cuotas de navegación.

Otro mecanismo de control lo constituye la monitorización de la información registrada en los servidores *proxy*<sup>2</sup> desde los que se tienen acceso a la red. Mediante el análisis correcto de esta información se puede adquirir una visión clara de las actividades realizadas por los usuarios. Destacando la importancia de la analítica web como factor a medir dentro del proceso de calidad de las empresas que hacen uso de *Internet* (Maldonado, 2010).

---

<sup>1</sup> Sigla en inglés de *Uniform Resource Locator* - Localizador Uniforme de Recursos.

<sup>2</sup> Ordenador que intercepta las conexiones de red que un cliente hace a un servidor de destino (Oracle, 2012).

En la Universidad de las Ciencias Informáticas (UCI) se establece un sistema de cuotas para acceder a redes informáticas de alcance global. El Código de Ética; documento oficial de la UCI que establece los compromisos para el uso de las tecnologías de la información, plasma que la cuota asignada no debe ser utilizada para transmitir, acceder o difundir información pornográfica, terrorista, contrarrevolucionaria o en general con fines lesivos a los intereses de la sociedad, la Institución, la Revolución o de terceros (UCI, 2012). A pesar de lo establecido en el documento antes mencionado, hay una variedad de sitios a los cuales el acceso no está prohibido, pero por sus características y los objetivos que persiguen se establecen reglas que regulan el horario para acceder a este tipo de información. Sin embargo, los usuarios incumplen dichas reglas y en ocasiones hacen uso de la cuenta en actividades de ocio tales como jugar, descargar música, chatear y concertar encuentros; ignorando las actividades de superación, de investigación científica y otras que elevan el prestigio de la institución.

La Resolución 27 del 2007 del Ministerio de la Informática y las Comunicaciones (MIC) de Cuba establece que, en todas las redes se deben habilitar las opciones de seguridad con que cuentan los sistemas operativos y los elementos que permitan el monitoreo y auditoría de los principales eventos por un tiempo no menor de un año (MIC, 2012). Estos monitoreos y auditorías permitirán tomar las decisiones correctas ante los incidentes y violaciones de seguridad informática, que afectan el crecimiento y prestigio de las instituciones. La UCI no está exenta de estas obligaciones y debido al uso inadecuado de la cuota de navegación se realizan auditorías a grupos de usuarios y áreas determinadas. Para ello se analiza la información almacenada en los archivos *logs*<sup>3</sup> generados por los servidores *proxy*.

En la universidad hay un sistema llamado AiresProxyAudit. El mismo permite representar a través de una estructura jerárquica los departamentos de una entidad y los trabajadores de las mismas; posibilitando el agrupamiento de usuarios por áreas. Sin embargo, no brinda la posibilidad de generar reportes de navegación sobre esta estructura, lo cual impide conocer de forma rápida el uso que le dan a la cuota de navegación los usuarios de la red. Debido ello se utiliza la herramienta privativa de análisis Sawmill. Esta aplicación analiza todos los registros almacenados en un directorio, imposibilitando el procesamiento de datos por grupos de usuarios específicos. Como consecuencia de ello se hace necesario el empleo de un script<sup>4</sup> que extrae los *logs* referentes a los usuarios especificados de forma manual y los almacena en el directorio de análisis de la herramienta. Estos procedimientos son lentos y reiterativos, pues una vez identificada la necesidad de analizar un nuevo grupo de usuarios hay que realizar todo el proceso.

Debido a la situación planteada se identifica el siguiente **problema de investigación**: ¿Cómo realizar un análisis más rápido de los registros de navegación referente a grupos de usuarios?

---

<sup>3</sup> Registro oficial de eventos durante un periodo de tiempo (Oxford, 2002).

<sup>4</sup> Guión o conjunto de instrucciones (SounDebian, 2009).

A partir del problema, se define como **objeto de estudio** el proceso de análisis de *logs* de los servidores *proxy*.

Enmarcando como **campo de acción** el proceso de análisis de *logs* de los servidores *proxy* referentes a la navegación de grupos de usuarios.

El **objetivo general** de esta investigación es, desarrollar un Módulo de Gestión de Reportes Estadísticos (MGRE) que permita analizar la información de los registros de navegación referentes a grupos de usuarios para el sistema AiresProxyAudit de la UCI.

Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**:

1. Construir el marco teórico conceptual y el estado del arte respecto a las tecnologías actuales relativas al análisis de *logs* de los servidores *proxy*.
2. Diseñar un módulo de reportes estadísticos sobre la navegación de grupos de usuarios.
3. Implementar una herramienta que permita la visualización, configuración y almacenamiento de diferentes reportes estadísticos referentes a la navegación de grupos de usuarios.
4. Validar el correcto funcionamiento del módulo de gestión de reportes estadísticos referentes a la navegación de grupos de usuarios.

Estos objetivos se fundamentan en la realización de las **tareas** que se reflejan a continuación:

1. Estudio de los conceptos asociados al marco teórico conceptual de la investigación.
2. Caracterización de los sistemas de reportes estadísticos sobre el uso de *Internet* utilizados en las empresas.
3. Identificación de las tecnologías idóneas para el diseño e implementación del MGRE.
4. Diseño de las interfaces del MGRE.
5. Implementación de las funcionalidades del MGRE.
6. Validación del MGRE a través de pruebas funcionales, de integración y de unidad.

Para dar cumplimiento a dichas tareas se emplean los siguientes métodos:

Métodos teóricos.

- Analítico-Sintético: Método utilizado durante el procesamiento del marco referencial de la tesis a partir de la sistematización del conocimiento científico relacionado con el estudio de los conceptos referentes al proceso de auditorías y el análisis de la bibliografía existente. Facilitando la integración de los conocimientos necesarios para el desarrollo del MGRE.
- Histórico Lógico: Empleado durante la realización del estado del arte, facilitando la implementación del MGRE al analizar con detalles los sistemas que en la actualidad se usan para el análisis de registros *proxy*, identificando las funcionalidades que brindan y las ventajas y desventajas que poseen.

Una vez cumplidas las tareas de investigación se presenta la siguiente **idea a defender**: Si se desarrolla un MGRE que utilice la información que almacenan los archivos *logs* se podrá realizar un análisis más rápido de los registros de navegación referentes a grupos de usuarios.

Para facilitar la lectura de esta investigación se propone la siguiente estructura del contenido:

- Capítulo 1 “Fundamentación Teórica”: Este capítulo se centra en el estado del arte del objeto de estudio. En él se analiza la existencia de trabajos similares en el ámbito nacional e internacional, se realiza un breve estudio de los lenguajes empleados, metodologías de desarrollo y se presenta además una descripción de las herramientas utilizadas para el desarrollo del MGRE.
- Capítulo 2 “Características del sistema”: En este capítulo se realiza una explicación de la teoría y las técnicas utilizadas para la realización del sistema. Se realiza una propuesta de solución y se identifican las funcionalidades que debe tener la misma. También se exponen los artefactos generados como parte de la metodología de desarrollo seleccionada.
- Capítulo 3 “Implementación y Pruebas de validación”: En este capítulo se describe todo el proceso de construcción del sistema y las pruebas realizadas a las funcionalidades del módulo en un entorno real, mostrando los resultados obtenidos para la valoración de la solución final.

## Capítulo 1 “Fundamentación Teórica.”

### 1.1. Introducción.

Los sistemas de análisis de registros de servidores *proxy* son herramientas útiles que permiten conocer el comportamiento de los usuarios en la red. En la actualidad se han desarrollado aplicaciones que tienen este fin, donde su principal objetivo es ayudar en la toma de decisiones en cuanto al uso de este servicio. Es por ello que se hace necesario el estudio de las mismas con el fin de identificar herramientas que puedan utilizarse para resolver el problema o posibles puntos comunes en el análisis de los registros de servidores *proxy*, además de las tecnologías a utilizar en el desarrollo del MGRE para el sistema AiresProxyAudit.

### 1.2. Conceptos Básicos.

Se utilizan en esta investigación continuamente varios términos importantes como el de **auditoría**, la cual constituye una herramienta de control y supervisión, que contribuye a la creación de una cultura de la disciplina de la organización y permite descubrir fallas en las estructuras o vulnerabilidades existentes en la organización (Villardefrancos, 2005). En el ámbito informático es el proceso de recoger, agrupar y evaluar evidencias para determinar si un sistema de información salvaguarda el activo empresarial, mantiene la integridad de los datos, lleva a cabo eficazmente los fines de la organización y utiliza eficientemente los recursos (SER, 2012).

Este proceso se puede realizar a través del análisis de los **logs**, los cuales constituyen un registro oficial de eventos durante un período de tiempo en particular (Oxford, 2002). Para los profesionales en seguridad informática un *log* es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué (*who, what, when, where y why, W5*) un evento ocurre para un dispositivo en particular o aplicación (Stewart, 2001). En principio, un archivo *log proxy* está diseñado para mostrar comportamientos de usuarios en la Web, a partir de ellos se puede obtener datos de calidad procesándolos previamente para construir modelos de usuarios (Brick Marketing, 2012). Entiéndase por **análisis de logs** la lectura de los registros de los servidores *proxy* a fin de extraer datos que pueden ser mostrados a través de **reportes**, informes que organizan y exhiben la información contenida en una base de datos. Donde su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios (Definición, 2013). Este análisis permite obtener los resultados de forma instantánea, pues no hay que esperar un análisis especial de expertos para entender lo que ha pasado y brinda un muestreo de grandes cantidades de usuarios (según la cantidad de usuarios con acceso a *Internet* dentro de la red) a lo largo de amplios plazos en el tiempo, detectando fácilmente el verdadero uso del servicio.

Con el fin de adaptar el MGRE a las necesidades de los usuarios, a estos reportes se le suman características de la **estadística**, ciencia formal que estudia la recolección, análisis e interpretación de datos de una muestra representativa, ya sea para ayudar en la toma de decisiones o para explicar condiciones regulares o irregulares de algún fenómeno o estudio aplicado. Dentro de este marco específicamente se aplica la **estadística descriptiva**, la cual se dedica a recolectar, ordenar, analizar y representar un conjunto de datos, con el fin de describir apropiadamente las características de ese conjunto. (Definición, 2013).

### 1.3. Herramientas para el análisis y control de *logs* de Servidores *Proxy*.

En el mercado actual del software existen varios analizadores de registros de servidores *proxy* que son usados en los diferentes niveles de acuerdo al objetivo que se desee alcanzar. Producto de la gran diversidad que existe, algunos brindan soluciones similares, mientras que otros difieren mucho en cuanto a sus funcionalidades, configuración, estándares de *logs* soportados, reportes y otros. Seguidamente se hace referencia a algunas de estas aplicaciones, exponiendo sus principales características, ventajas y desventajas.

#### 1.3.1. **Ámbito Internacional.**

##### **AWStats.**

Herramienta que genera estadísticas avanzadas sobre el uso de los servidores *proxy*, web, FTP<sup>5</sup> o correo en forma de gráficas a través de páginas web. Puede analizar además los registros de *logs* extendido de SafeSquid<sup>6</sup> y generar reportes de estadísticas mensuales, semanales, diarios, listas de usuarios, tipos de archivos, URL de las páginas, sistemas operativos, navegadores usados, búsqueda de palabras, frases claves y los códigos de estado del protocolo HTTP<sup>7</sup>. También permite realizar personalizaciones para generar informes especiales como son: las amenazas detectadas por el antivirus y las solicitudes bloqueadas por diversos filtros (URL, lista negra de URL, palabras clave, MIME<sup>8</sup>, Imagen). Esta herramienta está desarrollada haciendo uso del lenguaje PERL y se encuentra bajo licencia GPL<sup>9</sup> (AWStats, 2012). A pesar de generar un gran número de reportes y ser un software libre, la reutilización de los mecanismos de análisis o la integración con el sistema se dificulta por los términos de la licencia GPL. La inclusión de los reportes como posibles funcionalidades del módulo no se tuvo en cuenta, debido a que AWStats se caracteriza por brindar información estadística sobre la Web, no siendo este el objetivo perseguido por el módulo generador de reportes estadísticos, el cual pretende analizar los registros de un *proxy* para brindar información sobre la navegación de grupos de usuarios.

---

<sup>5</sup> Sigla en inglés de *File Transfer Protocol* - Protocolo de Transferencia de Archivos.

<sup>6</sup> Servidor *proxy* que provee mecanismos para llevar el control del contenido de *Internet* en una red (SafeSquid, 2012).

<sup>7</sup> Sigla en inglés de *Hypertext Transfer Protocol* - Protocolo de Transferencia de Hipertexto.

<sup>8</sup> Sigla en inglés de *Multipurpose Internet Mail Extensions* - Extensiones Multipropósito de Correo de *Internet*.

<sup>9</sup> Sigla en inglés de *General Public License*- Licencia Pública General creada por la *Free Software Foundation*.

### **Webalizer.**

Esta aplicación permite realizar análisis estadísticos del uso de *Internet* basándose en los *logs* del servidor *proxy* Squid o servidor web Apache. Las estadísticas extraídas facilitan la interpretación del uso de estos servidores al mostrarlas en páginas web acompañadas de reportes cuantitativos en forma de gráficos. Está escrito en el lenguaje de programación C, lo que permite rapidez en el procesamiento de datos. Puede analizar 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. Se encuentra bajo licencia GPL (Barrett, 2012).

A pesar de tener todas estas facilidades, este software presenta algunos inconvenientes. Uno de ellos es que no está orientado a usuarios, pues solo ofrece estadísticas generales del comportamiento de la navegación en la Web. Además, su uso es solo para reportes generales de empresas (no permite el análisis por área) o para el monitoreo de sitios en específico. Para su configuración inicial se requieren conocimientos técnicos sobre el sistema operativo GNU/Linux.

### **WebLog Expert.**

Analizador de registro de servidores que brinda información sobre los visitantes del sitio web monitorizado. Dentro de los reportes que genera se encuentran, estadísticas de actividad en el sitio, acceso a los archivos, rutas visitadas, páginas de referencia, motores de búsqueda, navegadores, sistemas operativos y otros.

Esta herramienta produce reportes fáciles de leer en formato HTML que incluyen información en texto y gráficos, mediante el uso de php<sup>10</sup>. Además, puede analizar los registros de Apache y de los servidores web bajo *Internet Information Server*<sup>11</sup> (IIS) en su versión libre. En otras versiones permite leer registros comprimidos a ZIP<sup>12</sup> y GZ<sup>13</sup>. Su interfaz es intuitiva y cuenta con un asistente automático que guía la creación de los perfiles de análisis (WebLog, 2012). Pese a que este analizador se encuentra bajo licencia pública y permite la obtención de reportes refinados por varios filtros, solo es aplicable para la obtención de estadísticas de un sitio por perfil, imposibilitando el análisis completo de los *logs* de servidores *proxy*.

### **Sawmill.**

Herramienta de análisis jerárquico de reportes, optimizado para reportes web pero en su proceso es capaz de soportar otros 915 formatos de *logs*. Además de procesar *logs* de formatos combinados y ampliados genera estadísticas de los mismos, presenta informes y análisis de acontecimientos. Tiene la capacidad de exportar

---

<sup>10</sup> Sigla en inglés de *Hypertext Preprocessor*-Pre procesador de Hipertexto. Lenguaje de programación interpretado del lado del servidor (PHP, 2013).

<sup>11</sup> Servidor *proxy* de *Internet*.

<sup>12</sup> Sigla en inglés de Zero Injection Pressure-Cero Inyección a Presión. Formato de compresión sin pérdida.

<sup>13</sup> Sigla en inglés de GNU ZIP.

## Capítulo 1 “Fundamentación Teórica.”

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, haciendo uso del lenguaje C (Sawmill, 2012).

Permite la creación de diferentes perfiles en los que se especifica que información se presentará 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 *logs*.

Presenta una serie de características como: facilidad de uso, de configuración y el procesamiento de casi cualquier tipo de reporte, convirtiéndolo en uno de los sistemas más fuertes de su tipo. 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 *logs* y el control sobre la apariencia de las páginas de estadísticas. Es una herramienta muy usada, no obstante se encuentra bajo licencia propietaria haciendo costoso su mantenimiento y actualización (Sawmill Analytics, 2012). Otra desventaja de esta aplicación es que no permite el agrupamiento de usuarios en varias áreas o departamentos.

### 1.3.2. **Ámbito Nacional.**

En Cuba existen un conjunto de herramientas para el análisis de registros de servidores *proxy*, las cuales son producto de la necesidad de controlar la navegación a través de *Internet* que presentan algunas empresas. Dentro de ellas se destacan:

#### **ISAWeb.**

Producto realizado en el Instituto Central de Investigación Digital (ICID) basado fundamentalmente en ASP.NET y se encuentra bajo licencia pública. El software permite que un usuario autenticado pueda visualizar la información de su navegación por *Internet*. Este software analiza los ficheros *logs* generados por servidores *proxy* Isa Server. Solo la información relevante (parámetros configurados) 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. 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. Es importante destacar que establece una jerarquía que permite descentralizar la labor de seguridad informática entre todos los directivos e incluye al trabajador.

## Capítulo 1 “Fundamentación Teórica.”

Sin embargo, este analizador tiene como desventaja su incapacidad para interpretar archivos *logs* de otros servidores distintos al IIS Server. Además, se resalta la utilización de software tales como IIS, *ASP.NET* y *Fox Pro* durante su creación, los cuales son gratis, pero su uso supone el pago de las herramientas que los soportan (Palenzuela, 2006).

### **SICC.**

El Sistema Integrado de Contabilidad y Configuración es un software desarrollado por el Centro Universitario José Antonio Echeverría (CUJAE), que mezcla análisis de *logs* 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 *logs* de la aplicación. El consumo de la navegación es almacenado en una estructura de ficheros serializados en lenguaje de programación Python y para el análisis hace uso de C++. Se caracteriza por estar bajo licencia pública (Garófalo, 2003).

Al ser una herramienta con fines contable no resuelve las necesidades de la investigación en curso, pues no permite el análisis de *logs* de los servidores *proxy* por grupos de usuarios.

### **SRNI.**

El Sistema de Reportes de la Navegación por *Internet* (SRNI) es un software implementado en la UCI que se encuentra bajo licencia GPL. 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. Toda esta información se enmarca en un rango máximo de 30 días y puede llegar a tener hasta 24 horas de retraso, lo que imposibilita la realización de reportes con un alto nivel de análisis y profundidad. Otras desventajas que esta aplicación posee son que no soporta la concurrencia de un alto número de usuarios y una vez detectado algún error en su funcionamiento no cuenta con un equipo de desarrollo que pueda resolver el problema. Por ende este sistema no permite el análisis de *logs* por áreas y al no contar con un grupo de soporte se dificulta su modificación.

### **AiresProxy.**

Esta herramienta fue desarrollada en la UCI con el fin de proveer a los administradores y usuarios de una aplicación capaz de generar reportes en forma de tablas, sobre la utilización del servidor *proxy*. Su

## Capítulo 1 “Fundamentación Teórica.”

funcionamiento se basa en la lectura y análisis de los *logs* mediante C++, posibilitando visualizar los resultados de manera rápida a través de una interfaz web. La información que muestra solo puede ser filtrada por fecha para los usuarios comunes y en el caso de los administradores pueden visualizar reportes de cualquier usuario bajo este mismo filtro. Permite una concurrencia de 400 usuarios por cada 1GB de procesamiento del servidor donde este alojado, se encuentra bajo licencia pública y por su facilidad de configuración permite la lectura de cualquier formato de archivos *logs*.

A pesar de su utilidad, esta aplicación no permite el procesamiento de *logs* agrupados por áreas o departamentos, la modificación de sus interfaces se hace compleja al no seguir un proceso de construcción estructurado y no genera información de forma gráfica que ayude en el análisis de la información, no siendo viable usarla como fin de esta investigación.

A continuación se propone de manera resumida una comparación de las herramientas abordadas (Ver tabla 1) teniendo en cuenta las siguientes métricas:

- **Licencia:** se refiere al estado jurídico de la aplicación en cuanto a su uso, modificación y distribución, puede tomar el valor público, para aquellas herramientas de acceso libre por los usuarios y privativo para las herramientas que necesitan de un pago para poder ser usadas durante un tiempo o modificadas.
- **Filtrado del reporte:** en dependencia de la posibilidad de filtrar los reportes que genera por más de un criterio (fecha, hora, usuarios, días de la semana, estación de trabajo, entre otros) su valor puede ser sí o no.
- **Lectura de varios formatos de logs:** la capacidad de la herramienta de leer más de un formato de *logs* de servidores *proxy*.
- **Lenguajes:** hace alusión a los lenguajes empleados en su implementación.
- **Visualización de reportes:** capacidad del sistema de mostrar los reportes en tablas y/o gráficos.
- **Análisis por grupos de usuarios:** posibilidad de análisis por áreas o departamentos dentro de la empresa o entidad que usen la aplicación.

Herramienta	Licencia	Filtrado del reporte	Lectura de varios formatos de logs	Lenguajes	Visualización de reportes	Análisis por grupo usuarios
AWStats	Público	Si	Si	Perl	Tablas y Gráficos	No
Webalizer	Público	No	Si	C	Tablas y Gráficos	No
WebLog Expert	Público	Si	Si	PHP-HTML	Tablas y Gráficos	No
Sawmill	Privativo	Si	Si	C	Tablas y Gráficos	No
ISAWeb	Público	Si	No	ASP.NET	Tablas y Gráficos	Si
SICC	Público	Si	No	C++-Python	Tablas y Gráficos	No
SRNI	Público	No	No	Java	Tablas	No
AiresProxy	Público	No	Si	PHP C++	Tablas	No

Tabla 1. Comparación entre las herramientas homólogas analizadas.

Tras el estudio de las herramientas dedicadas al análisis de registros de servidores *proxy* es posible arribar a las siguientes conclusiones:

- No analizan los registros por grupos de usuarios. A pesar de que la herramienta cubana ISAWeb realiza este tipo de análisis, solo es aplicable a empresas que usen Isa Server como *proxy*.
- Las herramientas cubanas basan su funcionamiento en la necesidad de las empresas que las desarrollaron, tal es el caso del SICC, donde sus reportes se llevan a unidades monetarias.
- En el ámbito internacional muchas de estas herramientas están orientadas a los administradores generales de red, imposibilitando la creación de jerarquías por áreas de trabajo.
- En el caso de la aplicación Sawmill a pesar de ser la más completa (aunque no permita el agrupamiento) se contempla dentro del grupo de herramientas privativas, lo cual atribuye un mayor

gasto para la empresa y dificulta la personalización de la misma al no tener acceso al código fuente.

Por todo lo antes expuesto se puede resumir que dichas herramientas no cumplen con el objetivo que se propone en esta investigación, pues no permiten el análisis de *logs* de los servidores *proxy* por grupos de usuarios. Sin embargo, el análisis realizado permitió detectar funcionalidades y tecnologías que permiten dotar al MGRE, propuesta de solución de la investigación, de una mayor rapidez (en el procesamiento de datos) y usabilidad.

### **1.4. Metodologías, tecnologías, librerías y herramientas propuestas para el desarrollo del sistema.**

Con el propósito de identificar las tendencias, tecnologías y buscar las posibles herramientas a utilizar durante el desarrollo del sistema se hace necesaria la realización de un estudio que permita la selección de las mismas. Teniendo en cuenta dicho estudio y las bases tecnológicas definidas en el centro de desarrollo que acoge la investigación se hacen las siguientes propuestas.

#### **1.4.1. Metodología de desarrollo.**

Para seleccionar que metodología de desarrollo es más adecuada utilizar, se debe tener en cuenta ciertos factores como: el dominio del producto a desarrollar, el contexto del proyecto, además del equipo de desarrollo. A continuación se expresa una serie de características que responden a dichos factores:

- La investigación en curso consta de la colaboración de pocas personas.
- Se necesita un ágil proceso de producción, asegurando la calidad del mismo.
- Se debe contemplar un mayor intercambio con el cliente, donde el ciclo de la investigación sea más completo al detectar posibles errores de comunicación con anterioridad.
- El sistema AiresProxyAudit emplea la metodología OpenUP para su desarrollo y documentación.

Debido a lo antes planteado la metodología que se decide emplear durante el desarrollo del MGRE es OpenUP. Esta es utilizada en los proyectos del departamento SINI<sup>14</sup>, permitiendo aprovechar la experiencia de los miembros del equipo de trabajo. Además, permite la adecuación a las características y necesidades propias del equipo de trabajo.

---

<sup>14</sup> Departamento de Soluciones Informáticas para *Internet* perteneciente al Centro de Ideoinformática – UCI.

## Capítulo 1 “Fundamentación Teórica.”

**OpenUP** adopta una filosofía pragmática y ágil que se centra en la naturaleza colaborativa de desarrollo de software. Es un proceso iterativo Mínimo, Completo y Extensible que puede utilizarse tal cual o ampliarse para tratar una amplia variedad de tipos de proyecto. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Está organizada dentro de cuatro áreas principales de contenido: Comunicación y Colaboración, Intención, Solución y Administración (Eclipse, 2012).

Algunas características de OpenUP son:

- Facilita la colaboración para alinear los intereses y un entendimiento compartido.
- Presenta un balance para confrontar las prioridades (necesidades y costos técnicos) y maximizar el valor para los *stakeholders*<sup>15</sup>.
- Enfoca la articulación de la arquitectura en facilitar la colaboración técnica, reducir los riesgos, minimizar excesos y trabajo extra.
- Continúa evolución para reducir riesgos, demostrar y obtener resultados (Eclipse, 2012).

Ventajas de utilizar OpenUP:

- Metodología de desarrollo de software de código abierto diseñada para pequeños equipos organizados que buscan tomar una aproximación ágil del desarrollo.
- Se valora la colaboración y el aporte de los *stakeholders* sobre los entregables y las formalidades innecesarias.
- No define un modelo de negocio ni de dominio necesario.
- Permite detectar errores tempranos a través de un ciclo iterativo.
- Evita la elaboración de documentación, iteraciones y diagramas innecesarios requeridos en la metodología RUP.
- Tiene un enfoque centrado al cliente y con iteraciones cortas (Eclipse, 2012).

---

<sup>15</sup> “Quienes pueden afectar o son afectados por las actividades de una empresa” (Freeman, 1984).

### 1.4.2. Framework de desarrollo.

Los framework de desarrollo podrían definirse como un conjunto de herramientas destinadas a la construcción de un determinado tipo de aplicación de manera generalizada, son una estructura conceptual que constituye la base para una aplicación. Su utilización simplifica el desarrollo de un sistema mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes; evita la re-implementación de funcionalidades que resultan ser frecuentes en este tipo de sistemas y que tradicionalmente incluyen administración de usuarios, persistencia de datos, motores de plantillas, entre otras; facilita la programación de aplicaciones, puesto que encapsula operaciones complejas en instrucciones sencillas y además, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener (De Los Angeles, 2012).

Durante la solución para el sistema AiresProxyAudit se pretende utilizar el marco de trabajo Symfony2, teniendo en cuenta una posible integración con el sistema AiresProxy, el cual está implementado sobre este framework.

**Symfony2** es la versión más reciente de Symfony, el popular framework para desarrollar aplicaciones PHP. Se anunció por primera vez a principios de 2009 y supone un cambio radical tanto en arquitectura interna como en filosofía de trabajo respecto a sus versiones anteriores. Ha sido ideado para aprovechar al máximo todas las nuevas características de PHP 5.3 y por eso es uno de los frameworks PHP de mejor rendimiento. Su arquitectura interna está completamente desacoplada, lo que permite adaptarlo completamente a las necesidades del equipo de trabajo.

Es el framework que más ideas incorpora del resto de los marcos de trabajo, incluso de aquellos que no están programados con PHP (Equiluz, 2011).

Una de las principales características que ofrece es el uso del patrón arquitectónico Modelo-Vista-Controlador (MVC). MVC es una aproximación al software que separa la lógica de la aplicación (modelo de datos y reglas del negocio) de la presentación (interfaz de usuario), lo cual es aceptado como una buena práctica dado que estructura el código modularmente, promueve la reutilización del código y disminuye la codificación de las vistas que pueden presentarse tanto como una página web de cara al usuario, como un servicio web que sirva de interfaz para aplicaciones remotas.

Otras características de este Framework son:

**Persistencia de Datos:** Capacidad de almacenar información y construir páginas dinámicas basadas en esta. Para ello soporta múltiples Sistemas Gestores de Base de Datos (SGBD), principalmente MySQL,

## Capítulo 1 “Fundamentación Teórica.”

PostgreSQL, Oracle, entre otros. Una condición que potencia el uso de este framework pues al soportar un mayor número de SGBDs permite al desarrollador abstraerse del gestor y sin modificar el código fuente pueda interactuar con los gestores que el framework soporta.

**Motor de plantilla:** En la versión 2 de este framework aparece el motor de plantillas Twig el cual permite separar la lógica de negocio de la de presentación. Entre sus principales características se encuentra el manejo de variable, la posibilidad de diferenciar código HTML y PHP y la utilización de la herencia entre plantillas, facilitando la reutilización de código.

**Trabajo con caché:** Mejora el rendimiento de los sistemas web, pues permite almacenar cierto contenido de uso muy frecuente, de modo que este no tenga que generarse ante cada petición de la página. El uso de la caché reduce considerablemente el consumo de ancho de banda y la recarga del servidor.

**Mapeo de URL:** Se traduce en simplificar la URL, es el mecanismo mediante el cual un framework interpreta una URL de manera que esta resulte amigable para el usuario a la vez que permite una mejor indización por parte de los motores de búsqueda.

**Seguridad:** Permite verificar y solicitar autenticación antes de conceder el acceso a un determinado recurso o contenido, característica que combina con la gestión de usuarios.

(De Los Angeles, 2012).

### 1.4.3. Lenguajes utilizados.

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente (Gutiérrez, 2007).

A continuación se realiza un estudio de los distintos lenguajes que se utilizan en el desarrollo del MGRE.

#### PHP 5.

*Hypertext Pre-processor* o Preprocesador de Hipertexto utilizado para la creación de sitios web. Es un lenguaje interpretado en el lado servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. No necesita ser compilado para ejecutarse.

Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Está diseñado

## Capítulo 1 “Fundamentación Teórica.”

específicamente para ser un lenguaje seguro. A continuación se relacionan algunas de sus características más relevantes:

- Es multiplataforma, puede ser utilizado sobre los sistemas operativos: GNU/Linux, Windows, etc.
- Funcionalidades de conexión con la mayoría de los sistemas de gestión de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Soporta la orientación a objeto y sus paradigmas.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.
- Fácil de aprender.
- Posee una extensa documentación con ejemplos de uso de sus funciones.
- Es distribuido bajo licencia libre.

(W3schools, 2012).

Este lenguaje es considerado la base del MGRE, pues forma parte de un sistema web dinámico que necesita la existencia de una comunicación entre un cliente y un servidor. Además, este sistema hace uso del framework Symfony2 basando en este lenguaje.

### **HTML 5.**

*Hyper Text Markup Language* o Lenguaje de Marcas de Hipertexto es el lenguaje de marcado predominante para construir páginas web, utilizado normalmente en la *World Wide Web*. Define la estructura y el contenido de las páginas permitiendo combinar textos, imágenes, sonidos, vídeos y enlaces a otras páginas. Su nivel de complejidad es bajo y además permite embeber dentro de su código otros lenguajes como PHP y JavaScript.

HTML5 es la última versión de HTML y XHTML. Define un lenguaje sencillo que se puede escribir en HTML y XML. Resuelve los problemas encontrados en versiones anteriores de HTML y atiende las necesidades de las aplicaciones web (Álvarez, 2009), siendo seleccionado por su facilidad de uso a la hora de trabajar con las interfaces del MGRE y teniendo en cuenta que es la base de las plantillas twig usadas para la generación de las vistas en Symfony2.

### **XML.**

El *Extensible Markup Language* o Lenguaje Extensible de Marcas (XML), es un metalenguaje extensible de etiquetas desarrollado por la *World Wide Web Consortium* (W3C). Permite definir la gramática de lenguajes específicos, por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

No ha sido creado sólo para su aplicación en *Internet*, se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas de manera segura, fiable y fácil (Mercer, 2002).

Teniendo en cuenta estas características este lenguaje será usado en la creación de configuraciones necesarias para el funcionamiento del MGRE, dotando al mismo de una mayor extensibilidad por la facilidad de configuración que posibilita.

### **C++.**

Lenguaje orientado a la implementación de sistemas operativos, aunque también se utiliza para crear aplicaciones. Débilmente tipificado de medio nivel pero con muchas características de bajo nivel. Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, dispone de construcciones que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria y a dispositivos periféricos. Entre sus características más importantes se encuentran:

- Un núcleo simple, con funcionalidades importantes añadidas, como funciones matemáticas y de manejo de archivos, proporcionado por bibliotecas.
- Flexible, permite programar con múltiples estilos. Uno de los más empleados es el estructurado.
- Acceso a memoria de bajo nivel mediante el uso de punteros.
- Punteros a funciones y variables estáticas, que permiten una forma rudimentaria de encapsulado y polimorfismo.
- Tipos de datos agregados (*struct*) que permiten que datos relacionados se combinen y se manipulen como un todo.
- Un conjunto reducido de palabras claves.

## Capítulo 1 “Fundamentación Teórica.”

Su éxito entre los programadores profesionales le ha llevado a ocupar el primer puesto como herramienta de desarrollo de aplicaciones. El C++ mantiene las ventajas del C en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original (Chávez, 2010).

Teniendo en cuenta los elementos que se enuncian el lenguaje C++ se empleará durante el proceso de análisis de *logs* y salva de los reportes estadísticos que se encuentran en la base de datos. Además, durante el estudio de los sistemas homólogos se detectó que de la muestra seleccionada el 50% de la misma debía su rapidez en el procesamiento de los datos al uso de dicho lenguaje. Esto posibilita al MGRE analizar la mayor cantidad de información, en el menor tiempo posible.

### 1.4.4. Gestor de Base de Datos.

Se seleccionó como gestor de base de datos MongoDB, el cual presenta interfaz para varios lenguajes programación como C++, Java, PHP, C# y otros. Utiliza un fichero asignado en memoria (ocupa una porción de la memoria virtual) como mecanismo de almacenamiento y consulta de la información. Lo anterior hace que delegue gran parte del manejo de la memoria al sistema operativo donde se ejecuta, por lo que su código para gestionar la memoria es pequeño y claro. Las operaciones de entrada y salida son mucho más rápidas con esta variante.

Cuenta con una comunidad estable y amplia en experiencia de desarrollo con la herramienta que brinda un espacio para consultar y solucionar los problemas que se presentan en el desarrollo con MongoDB.

Otra característica importante es que mantiene algunas características de SQL que permiten que los usuarios que desarrollan con MongoDB luego de haberlo hecho con SQL perciban de un modo mucho más sencillo el cambio de paradigma (MongoDB, 2012).

Tomando como referencia lo antes planteado y la necesidad de velocidad en el procesamiento de la información, todos los datos que maneja en MGRE se encontrarán bajo un servidor MongoDB. Además, se debe tener en cuenta que el sistema AiresProxyAudit para el cual se desarrolla este módulo hace uso de esta tecnología.

### 1.4.5. Librerías utilizadas en el funcionamiento del MGRE.

**Bootstrap v2.2:** librería que provee al desarrollador funcionalidades, estilos y formatos a través de solución limpia y uniforme, facilitando el maquetado y la implementación de aplicaciones web mediante el uso de CSS 3, HTML 5 y JavaScript. Permite controlar y formatear todos los elementos dentro de una vista (Twitter, 2013). En el caso de las hojas de estilo en cascada (CSS), constituyen el estándar para la inserción de estilo

(tamaños, colores, tipografías, espacios, bordes) a documentos estructurados, como por ejemplo, documentos HTML o XML. El objetivo de la definición de este estándar es permitir la separación entre las normas de presentación y el propio contenido a mostrar (W3schools, 2012).

**JQuery v1.9:** biblioteca JavaScript<sup>16</sup> que facilita el desarrollo de aplicaciones web interactivas. Distribuido bajo licencias MIT<sup>17</sup> y GPL. Ofrece una excelente API<sup>18</sup> que facilita el uso de las funciones del DOM<sup>19</sup> así como otras no soportadas por éste, manipulación de CSS, incluye funcionalidades para el trabajo con AJAX evitando la carga del servidor , manipulación de eventos, animaciones, diseño y efectos del contenido web de forma sencilla. Es de fácil aprendizaje y posee una amplia documentación en su sitio oficial (jQuery, 2013).

**HighChart v1.2:** biblioteca para la generación de gráficos, está implementada bajo lenguaje JavaScript, lo que posibilita la adición de estos componentes visuales de forma dinámica en las interfaces de las aplicaciones web. Puede ser usada para visualizar diferentes tipos de gráficos en diferentes navegadores (HighChart, 2013).

### 1.4.6. Entornos Integrados de Desarrollo (IDEs).

Los Entornos de Desarrollo Integrado (IDE) son herramientas que juegan un papel importante en el desarrollo de soluciones informáticas. Los mismos ofrecen facilidades al equipo de desarrollo cuando se implementan las aplicaciones debido a que permite corrección de errores comunes que se comenten a diario. Los entornos de desarrollo seleccionados para la realización de este trabajo fueron el NetBeans y el QtCreator.

#### **Netbeans 7.3.**

Herramienta gratuita y de código abierto para el desarrollo de aplicaciones web, de escritorio y móviles, disponible en los sistemas operativos Windows, GNU/Linux, Mac OS X y Solaris. La base en la que se sustenta su elección es que permite desarrollar aplicaciones utilizando el framework Symfony y ejecutar los comandos del mismo directamente desde el entorno de la aplicación. Además, este IDE es de fácil instalación, sus funcionalidades son extensibles mediante plugins y permite el auto completamiento de códigos en los lenguajes C++, HTML, PHP, JavaScript y XML, todos estos utilizados en el MGRE. Esta versión ofrece un mejor rendimiento y con nuevas capacidades de análisis de código estático (Oracle, 2012).

---

<sup>16</sup> Lenguaje de programación interpretado del lado del cliente (Álvarez, 2001).

<sup>17</sup> Licencia creada por *Massachusetts Institute of Technology*.

<sup>18</sup> Siglas en inglés de *Application Programming Interface* - Interfaz de Programación de Aplicaciones.

<sup>19</sup> Siglas en inglés de *Document Object Model* - Modelo de Objetos del Documento.

### **QTCreator 4.6.3.**

Editor de C++ creado por la compañía NOKIA para el desarrollo con las librerías Qt, lo cual implica una integración total con el diseñador de Qt y todas las funciones con las que cuenta. Es un entorno liviano rápido y puede instalarse tanto en Windows como en Linux. Este IDE permite trabajar directamente en los archivos .pro del proyecto, posee una búsqueda incremental, integra la ayuda de Qt, facilita el completamiento de código y resaltado de sintaxis, muestra los errores en línea mientras se escribe, cuenta una interfaz para depurar los proyectos y posibilita la integración para control de versiones (Digia, 2012).

Todos estos elementos los convierten en la herramienta ideal para el manejo del lenguaje C++. Además, el departamento SINI cuenta con especialistas capacitados en el uso de dicha herramienta, lo que constituye una fortaleza para el equipo de desarrollo.

### **1.4.7. Lenguaje y herramienta de Modelado.**

Las herramientas de modelado son las encargadas de ayudar en el ciclo de desarrollo del software, con el fin de aumentar la productividad y reducir el coste en términos de tiempo y dinero. Las mismas ayudan en el proceso de diseño del proyecto, en el cálculo de costes, implementan parte del código, compilación automática y documentación. Para ello hacen uso de lenguajes de modelado. En el presente trabajo se utilizó el Lenguaje Unificado de Modelado (UML) a través de Visual Paradigm 8 para UML.

**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 (Object Management Group, 2012). 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 (OO) (Jacobson, 2000) pues permite la modelación de sistemas con tecnología OO y describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema (Object Management Group, 2012).

**Visual Paradigm 8** es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Garantiza una rápida construcción de aplicaciones con una mayor calidad, y a un menor coste. Permite dibujar todos los tipos de diagramas de

## Capítulo 1 “Fundamentación Teórica.”

clases, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML.

Es una herramienta multiplataforma distribuida bajo licencia privativa en la Visual Paradigm Suite. Es orientada a objetos y se puede integrar con Eclipse, Netbeans, IntelliJ IDEA entre otras herramientas de desarrollo. Proporciona un entorno ágil y eficiente para el diseño (Visual Paradigm, 2012).

### **1.5. Conclusiones.**

Al término de este capítulo se pueden arribar a las siguientes conclusiones:

Identificar un grupo de conceptos asociados a la investigación permitió un mejor entendimiento del entorno que se desenvuelve esta investigación.

Estudiar las herramientas de análisis de *logs* ayudó a obtener una mayor claridad en el proceso de análisis y generación de reportes sobre el uso de servidores *proxy*.

Reconocer un grupo de funcionalidades y tecnologías e identificar una metodología de desarrollo posibilita llevar a cabo el desarrollo de la investigación, facilitando obtener una aplicación informática lo más robusta posible.

## Capítulo 2 “Características del Sistema.”

### 2.1. Introducción.

Establecer un entendimiento entre el cliente y el equipo de trabajo en relación con los objetivos a lograr, es una de las prioridades cuando se desea desarrollar un sistema. Las características del mismo, deben quedar claramente identificadas y redactadas para que el software quede con la calidad que requiere. Por esta razón en el presente capítulo se realiza una descripción de la solución propuesta para la implementación del módulo de gestión de reportes estadísticos para el sistema AiresProxyAudit. Para ello se utilizan los artefactos de información propuestos por la metodología OpenUP seleccionada en el capítulo anterior.

### 2.2. Características del sistema.

El MGRE para el sistema AiresProxyAudit se enmarca dentro de los sistemas analizadores del uso de servidores *proxy*. Una de las características fundamentales del módulo es que permitirá a los auditores (definidos por la UCI) generar diferentes reportes sobre las distintas áreas que estos atiendan, descentralizando el proceso de auditorías. Esto permite obtener datos relevantes sobre el uso de *Internet* por grupos de usuarios en un tiempo menor comparado con el flujo actual del proceso; el cual requiere la introducción manual de los usuarios, a fin de separar sus *logs* en el directorio de análisis de la herramienta privativa Sawmill.

Para lograr este resultado el MGRE basa su funcionamiento en el uso de una aplicación web que permite la selección de los filtros a aplicar en un reporte y la visualización de los mismos. Estos son creados y almacenados en una base de datos por un componente desarrollado en C++, lenguaje empleado para procesar los datos antes de que se hagan persistentes.

Durante los siguientes epígrafes se describen artefactos que permiten conceptualizar y entender con claridad el funcionamiento del MGRE.

### 2.3. Modelo del dominio.

Para lograr un mejor entendimiento de los procesos que requieren automatización se realizó un modelo del dominio (Ver Imagen 1), el cual recoge y describe los conceptos más importantes dentro del contexto del sistema así como las relaciones entre ellos.

Dentro de este modelo se aprecia un **auditor** como actor principal del proceso. Este es el encargado de **crear** las distintas **configuraciones** que pueden **filtrar** los **reportes** que se pueden **generar**. Dichos reportes son el

resultado de **procesar** los **logs** pertenecientes a los **usuarios** que **componen** las **entidades seleccionadas** por el auditor que **forman** la **estructura de la organización**.

Descripción de las entidades que se reflejan:

Auditor: Actor principal del sistema que puede accionar sobre algún área de la empresa.

Configuración: Conjunto de filtros que se le aplican a un reporte.

Reporte: Datos estadísticos que se obtiene del procesamiento de los *logs*.

Logs: Registro que almacena las peticiones de acceso de los usuarios al servidor *proxy*.

Estructura Organizacional: Jerarquía que agrupa todas las áreas en la que está estructurada la empresa.

Entidad: Área de la empresa que agrupa a las personas pertenecientes a ella.

Usuario: Cualquier persona que tenga acceso a *Internet* dentro de la empresa.

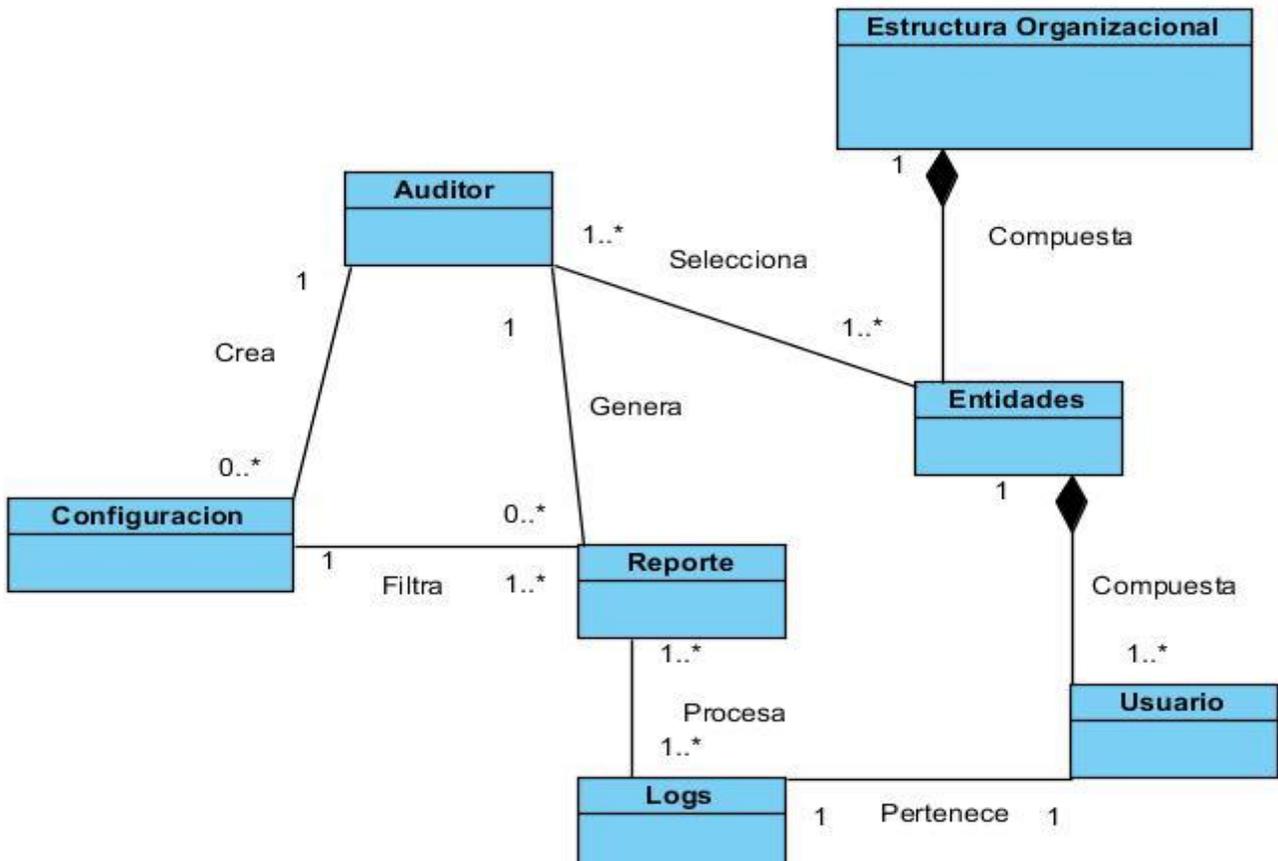


Imagen 1. Modelo de Dominio del MGRE del sistema AiresProxyAudit.

## 2.4. Especificación de los requisitos de Software.

Modelar el sistema posibilita reflejar las propiedades o restricciones que debe cumplir el MGRE, a fin de satisfacer las necesidades del cliente. Para ello se proponen los siguientes requisitos funcionales y no funcionales.

### 2.4.1. Requisitos funcionales.

Los requisitos funcionales especifican comportamientos particulares de un sistema, son características requeridas del sistema que expresan una capacidad de acción del mismo (Pressman, 2006). A continuación se listan los requerimientos funcionales que el MGRE debe cumplir, la prioridad y su asignación por caso de uso. Todos los requisitos son desarrollados por los programadores del módulo, entiéndase por programadores a los autores de este documento.

Requisito Funcional	Prioridad	Caso de Uso del sistema
Autenticar usuario.	Alta	Autenticar usuario
Seleccionar filtro de fecha para un reporte.	Alta	Seleccionar filtro
Seleccionar filtro de usuario para un reporte.	Alta	Seleccionar filtro
Seleccionar filtro de días para un reporte.	Alta	Seleccionar filtro
Seleccionar filtro de hora para un reporte.	Alta	Seleccionar filtro
Seleccionar filtro de ips para un reporte.	Alta	Seleccionar filtro
Seleccionar filtro de dominio para un reporte.	Alta	Seleccionar filtro
Mostrar entidades del auditor.	Alta	Mostrar entidades
Seleccionar entidades para un reporte.	Alta	Seleccionar filtro
Generar reporte estadístico general.	Alta	Generar reporte
Generar reporte estadístico de sitios visitados.	Alta	Generar reporte
Generar reporte estadístico por años.	Alta	Generar reporte
Generar reporte estadístico por meses.	Alta	Generar reporte

## Capítulo 2 “Características del Sistema.”

<b>Requisito Funcional</b>	<b>Prioridad</b>	<b>Caso de Uso del sistema</b>
Generar reporte estadístico por días del mes.	Alta	Generar reporte
Generar reporte estadístico por días de la semana.	Alta	Generar reporte
Generar reporte estadístico por horas.	Alta	Generar reporte
Generar reporte estadístico por ips.	Alta	Generar reporte
Generar reporte estadístico por usuarios.	Alta	Generar reporte
Generar reporte estadístico por categoría.	Alta	Generar reporte
Mostrar gráficos de reportes por fecha.	Alta	Mostrar gráficos
Mostrar gráficos de reportes por año.	Alta	Mostrar gráficos
Mostrar gráficos de reportes por mes.	Alta	Mostrar gráficos
Mostrar gráficos de reportes por día de la semana.	Alta	Mostrar gráficos
Mostrar gráficos de reportes por hora.	Alta	Mostrar gráficos
Ordenar resultados por diferentes campos.	Baja	Ordenar resultados
Exportar reporte a pdf.	Baja	Exportar reporte a pdf
Guardar configuración de reportes.	Media	Guardar configuración
Importar configuración de reportes.	Media	Importar configuración
Exportar configuración de reportes.	Media	Exportar configuración
Guardar reporte.	Media	Guardar reporte
Mostrar reporte guardado.	Media	Gestionar reporte guardado
Eliminar reporte guardado.	Media	Gestionar reporte guardado

Tabla 2. Requisitos Funcionales del MGRE para el sistema AiresProxyAudit.

### 2.4.2. Requisitos no funcionales.

Son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, de desarrollo y estándares. A menudo se aplican al sistema en su totalidad (Pressman, 2006). A continuación se presentan los del MGRE:

#### **Usabilidad:**

- El módulo debe mantener una adecuada distribución de los contenidos y colores de forma tal que resulte amigable al usuario.
- Debe ser intuitivo, agrupando las funcionalidades por responsabilidad, lo que permite a los auditores de poca experiencia hacer uso del mismo.

#### **Hardware:**

- El ordenador donde se ejecute el módulo debe tener una memoria RAM de 2 GB, además de poseer un procesador familia INTEL u otro con una velocidad mínima de 2.10 GHz y la capacidad del disco duro debe ser de 500Mb como mínimo.
- El disco duro del servidor de base de datos debe tener una capacidad de almacenamiento de 80 GB como mínimo.
- Las estaciones de trabajo (PC clientes y servidores) deben estar conectadas a una red y tener al menos 100 MB de velocidad.

#### **Software:**

- El procesador donde se ejecute el módulo debe tener instalado GNU/Linux como sistema operativo, se recomiendan las distribuciones Debian o Ubuntu.
- Se requiere la instalación de las librerías php5-mongo, qt4, gcc-c++ y libboost-dev en el servidor donde este alojado el sistema y mongodb-server en el servidor de base de datos.

#### **Seguridad:**

- La información manejada por el sistema estará protegida de accesos no autorizados.
- La información manejada por el sistema es objeto de cuidadosa protección contra la corrupción y

acciones inconsistentes.

- En todo momento la información estará disponible para aquellos usuarios que tienen acceso al sistema.

### 2.5. Modelo de casos de uso del sistema.

El Modelo de Casos de Uso del Sistema (MCUS) permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, es decir sobre las condiciones y posibilidades que debe cumplir el sistema. Además proporciona la entrada fundamental para el análisis, diseño y las pruebas (Jacobson, 2000).

Seguidamente se presenta el MCUS perteneciente al MGRE para el sistema AiresProxyAudit agrupando en cada caso de uso los requisitos funcionales según la Tabla 2.

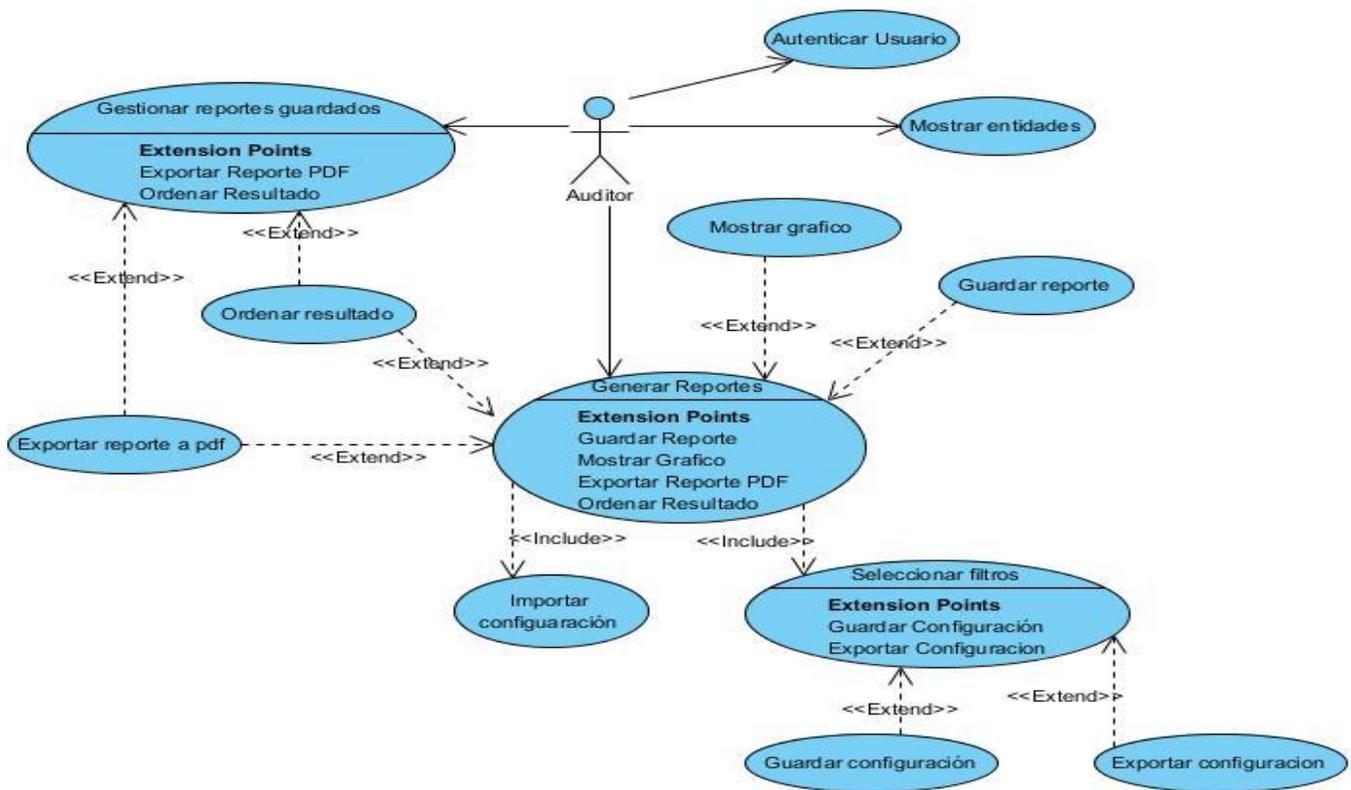


Imagen 2. Modelo de Casos de Usos del Sistema del MGRE para el sistema AiresProxyAudit.

A fin de lograr un mayor entendimiento por parte de los lectores se describen a continuación algunas especificaciones de casos de usos del sistema catalogados como críticos. El resto de las especificaciones se pueden encontrar en el artefacto “Especificación de Requisitos” que se encuentra archivado en el expediente de proyecto.

## Capítulo 2 “Características del Sistema.”

<b>Caso de Uso</b>	Autenticar usuario.	
<b>Objetivo</b>	Validar la presencia del usuario en el sistema.	
<b>Actores</b>	Auditor.	
<b>Resumen</b>	El caso de uso se inicia cuando el auditor desea autenticarse, el sistema muestra una interfaz para realizar esta acción, finaliza el CU cuando el usuario se ha autenticado en el sistema correctamente.	
<b>Complejidad</b>	Alta.	
<b>Prioridad</b>	Crítico.	
<b>Precondiciones</b>	El usuario ha sido creado en el sistema.	
<b>Postcondiciones</b>	El usuario ingresó al sistema.	
<b>Flujo de eventos</b>		
<b>Flujo básico: Flujo normal de eventos.</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Introduce en el navegador la dirección del sistema.	
2.		Muestra una interfaz que permite la autenticación a través de los campos <b>Usuario</b> y <b>Contraseña</b> .
3.	Introduce su usuario y contraseña y envía los datos dando clic en el botón <b>Iniciar Sesión</b> .	
4.		Valida los datos.
5.		Redirecciona al usuario hacia la interfaz principal.
6.		Termina el CU.
<b>Flujos alternos</b>		
4. Los datos introducidos por el usuario no tienen coincidencias en el sistema.		
	<b>Actor</b>	<b>Sistema</b>

## Capítulo 2 “Características del Sistema.”

5.		Muestra un mensaje de error donde se le comunica al usuario que no existen el par usuario y contraseña suministrados por él y regresa al flujo normal 3.
<b>Flujos alternos</b>		
4. <b>El usuario no tiene acceso al sistema</b>		
	<b>Actor</b>	<b>Sistema</b>
5.		Muestra un mensaje de error donde se le comunica al usuario que no tiene acceso al sistema y regresa al flujo normal 3.
<b>Flujos alternos</b>		
4. <b>Campo Usuario y/o Contraseña en blanco</b>		
	<b>Actor</b>	<b>Sistema</b>
5.		Muestra un mensaje de error donde se le comunica al usuario que no puede dejar el/los campo(s) usuario y/o contraseña en blanco y regresa al flujo normal 3.
<b>Requisitos no funcionales</b>		La información manejada por el sistema estará protegida de accesos no autorizados.

Tabla 3. Caso de Uso Autenticar usuario.

<b>Caso de Uso</b>	Seleccionar filtros.
<b>Objetivo</b>	Seleccionar uno o varios filtros para restringir los resultados del reporte.
<b>Actores</b>	Auditor.
<b>Resumen</b>	El caso de uso se inicia cuando el auditor desea seleccionar filtros para realizar reportes, el sistema muestra la interfaz correspondiente con la opción Configuración y el auditor selecciona uno o varios filtros, finaliza así el CU.
<b>Complejidad</b>	Alta.
<b>Prioridad</b>	Crítico.
<b>Precondiciones</b>	El auditor se ha autenticado en el sistema.
<b>Postcondiciones</b>	Se selecciona el filtro deseado.

Flujo de eventos		
Flujo básico: Flujo normal de eventos.		
	Actor	Sistema
1.	Selecciona la opción “Configuración”.	
2.		Muestra la interfaz correspondiente.
3.	Selecciona uno o varios filtros en la configuración inicial de los parámetros del reporte para restringir los resultados.	
4.		Termina el CU.
Relaciones	CU Incluidos	Mostrar entidades.
	CU Extendidos	Guardar configuración. Exportar configuración.

Tabla 4. Caso de Uso Seleccionar filtros.

<b>Caso de uso</b>	Generar Reportes.	
<b>Objetivo</b>	Generar todos los reportes que brinda el sistema.	
<b>Actores</b>	Auditor.	
<b>Resumen</b>	El caso de uso se inicia cuando el auditor desea generar reportes y selecciona la opción generar reportes, el sistema lo redirecciona a la página reporte general y finaliza así el CU.	
<b>Complejidad</b>	Alta.	
<b>Prioridad</b>	Crítico.	
<b>Precondiciones</b>	Se han seleccionado filtros o se ha importado una configuración.	
<b>Postcondiciones</b>	Visualización del reporte resumen general.	
Flujo de eventos		
Flujo básico: Flujo normal de eventos.		
	Actor	Sistema

## Capítulo 2 “Características del Sistema.”

1.	Selecciona la opción “Generar Reportes”.	
2.		Valida todos los filtros entrados.
3.		Muestra una interfaz con los reportes generados.
4.		Termina el CU.
<b>Flujos alternos</b>		
<b>2. Filtros entrados son incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>
3.		Devuelve un mensaje mostrando donde se encuentra el error y vuelve al flujo normal 1.
<b>Relaciones</b>	<b>CU Incluidos</b>	<p>Seleccionar filtros.</p> <p>Importar configuración.</p>
	<b>CU Extendidos</b>	<p>Guardar reporte.</p> <p>Mostrar gráfico.</p> <p>Exportar reporte a pdf.</p> <p>Ordenar resultado.</p>

Tabla 5. Caso de uso Generar Reportes.

<b>Caso de uso</b>	Mostrar Gráficos.
<b>Objetivo</b>	Mostrar gráficos de los reportes que sean posibles.
<b>Actores</b>	Auditor.
<b>Resumen</b>	El caso de uso se inicia cuando el auditor desea mostrar de forma gráfica algún reporte, el sistema muestra la interfaz correspondiente con el gráfico deseado, finaliza así el CU.
<b>Complejidad</b>	Alta.
<b>Prioridad</b>	Crítica.
<b>Precondiciones</b>	El auditor debe haber generado los reportes.
<b>Postcondiciones</b>	Visualización del gráfico deseado.
<b>Flujo de eventos</b>	

Flujo básico: Flujo normal de eventos.		
	Actor	Sistema
1.	Selecciona la opción “Mostrar Gráficos” desde la interfaz del reporte deseado.	
2.		Muestra el reporte de forma gráfica.
3.		Termina el CU.

Tabla 6. Caso de uso Mostrar Gráficos.

## 2.6. Patrones utilizados en el desarrollo de la aplicación.

Los patrones detallan una solución probada con éxito que aparece una y otra vez ante determinado tipo de problema en un contexto dado. Estos se definen por un nombre, un problema, una solución y las consecuencias de su aplicación. Son una posible solución correcta para un problema de diseño dentro de un contexto dado, describiendo las cualidades invariantes de todas las soluciones. Además, se categorizan según la escala o nivel de abstracción, sin embargo cada una de las categorías de patrones define un mismo nivel de abstracción o escala de aplicabilidad (Pressman, 2006). A continuación se presentan los patrones utilizados en el desarrollo de MGRE para el sistema AiresProxyAudit.

### 2.6.1. Patrones de casos de uso.

**Extensión Concreta:** Se utiliza en los casos de uso Mostrar gráfico, Guardar reporte, Ordenar resultado, Guardar configuración, Exportar configuración y Exportar reporte a pdf. Consiste en la existencia de una relación de extensión entre dos casos de uso. El caso de uso extendido puede ser o no instanciado por el caso de uso base. El caso de uso base puede ser concreto o abstracto. Este patrón se utiliza cuando un flujo puede extender del flujo de otro caso de uso o bien puede ejecutarse dentro de este.

**Inclusión Concreta:** Utilizado en los casos de uso Importar configuración y Seleccionar filtros. Consiste en la existencia de una relación de inclusión del caso de uso base con el caso de uso incluido. Este último puede ser instanciado como el mismo. El caso de uso base puede ser concreto o abstracto. Este patrón se utiliza cuando un flujo puede ser incluido en el flujo de un caso de uso y también puede ejecutarse dentro de este.

**CRUD Parcial:** Se evidencia su utilización en el caso de uso Gestionar reportes guardados. Se basa en la fusión de casos de uso simples para formar una unidad conceptual. Modela una de las vías de los casos de uso como un caso de uso separado. Es utilizado cuando una de las alternativas de los casos de uso es más significativa, larga o más compleja que las otras (Larman, 1999).

### 2.6.2. Patrones Generales de Software para la Asignación de Responsabilidades (GRASP).

Los GRASP son patrones generales de software para asignación de responsabilidades a objetos. Aunque se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y aplicable (Visconti, 2011).

**Experto:** La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados. Con el uso de este patrón se alientan las definiciones de clase “sencillas” y más cohesivas que son más fáciles de comprender y de mantener. Brinda soporte a una alta cohesión (Visconti, 2011). La aplicación de este patrón permite mover parte de la lógica del negocio hacia las clases modelo evitando así la sobrecarga del controlador.

En Symfony 2 este patrón es utilizado en la inclusión de Doctrine para mapear la Base de Datos. Se utiliza para crear una capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases en todas las funcionalidades comunes a las entidades; las clases de abstracción de datos poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla asociada.

**Creador:** Se encarga de darle a la clase A la responsabilidad de crear objetos de la clase B. En este caso A es creador de los objetos B (Visconti, 2011).

En Symfony 2 la capa del controlador contiene el código que une la lógica del negocio con la presentación. Este cuenta con varios componentes entre los que se identifican las acciones, que son los métodos que definen variables para la vista.

**Alta Cohesión:** La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Las mismas poseen un número relativamente pequeño de responsabilidades, definiendo así que cada clase realice solo las funcionalidades para las cuales fueron creadas, generando un bajo acoplamiento y fomentando la reutilización (Visconti, 2011).

Una de las características principales del Framework Symfony es la organización del trabajo en cuanto a la estructura del proyecto, lo cual permite crear y trabajar con clases con una alta cohesión. Por ejemplo, se puede observar en el sistema que cada clase controladora se ajusta a manejar solo las responsabilidades

correspondientes a las entidades con las que se relaciona. Esto hace posible que el sistema sea flexible a cambios sustanciales con efecto mínimo.

**Bajo Acoplamiento:** Es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. O sea, mide el grado de conexión de una clase en cuanto al conocimiento que tiene ésta de otras clases, la recurrencia y conexión a ellas. Acoplamiento bajo significa que una clase no depende de muchas clases.

Se evidencia en todo el sistema ya que las clases controladoras del sistema no se relacionan entre sí, lo que disminuye las dependencias entre las mismas.

**Controlador:** Un Controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. El mismo asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente el “sistema” global (Visconti, 2011).

Este patrón se pone de manifiesto en todo el sistema debido a que cada uno de los eventos generados por el usuario es redirigido a una clase controladora que realiza las operaciones solicitadas, manteniendo siempre la alta cohesión.

### 2.6.3. Patrones Gang of Four (GOF)

El catálogo de patrones más famoso es el contenido en el libro “Design Patterns: Elements of Reusable Object-Oriented Software”, también conocido como: El libro GOF (Gang-Of-Four Book). Según este documento, estos patrones se clasifican según su propósito en creacionales, estructurales y de composición, mientras que respecto a su ámbito se clasifican en clases y objetos:

#### **Estructural:**

**Decorator (Decorador):** Permite añadir responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades. En Symfony 2 la vista se separa en una plantilla base y varias plantillas que heredan de esta. Normalmente, la plantilla base es global en toda la aplicación y contiene el código HTML que es común a la mayoría de las páginas, lo cual es una implementación del patrón Decorador. Su uso aporta una mayor flexibilidad que la herencia estática, permitiendo, entre otras cosas, añadir una funcionalidad dos o más veces (AspAlliance, 2013).

#### **Comportamiento:**

**Command (Comando):** Es un patrón de diseño de comportamiento de objetos. Manipular y encapsular las peticiones de los usuarios y enviarlas a un objeto encargado de darle respuesta. Su uso es apropiado cuando lo fundamental en la relación entre una petición y la acción que la satisface es la flexibilidad (AspAlliance, 2013).

**Template Method (Método Plantilla):** Define una estructura algorítmica en la súper clase, delegando la implementación a las subclases. Es decir, define una serie de pasos, en donde los pasos serán redefinidos en las subclases (polimorfismo). Un ejemplo del uso de este patrón en Symfony2 se ve reflejado en las plantillas twig. Estas permiten la herencia entre clases, así como la redefinición de los métodos a fin de lograr el diseño deseado (AspAlliance, 2013).

**Otros:**

**Registry:** Este patrón es útil para los desarrolladores en la Programación Orientada a Objetos. Es un medio sencillo y eficiente de compartir datos y objetos en la aplicación sin la necesidad de preocuparse por conservar numerosos parámetros o hacer uso de variables globales. Se aplica en la clase de configuración que es la encargada de guardar todas las variables globales del sistema (AspAlliance, 2013).

### 2.6.4. Patrón arquitectónico Modelo-Vista-Controlador.

El patrón Modelo Vista Controlador separa los datos, la interfaz de usuario y la lógica de control de una aplicación en tres componentes distintos, principalmente separa la lógica del negocio de la lógica de presentación, ventaja esta que posibilita el mantenimiento de los sistemas, la seguridad y la simplificación en el desarrollo.

Los componentes de este patrón son los siguientes:

**Modelo:** Encapsula los datos y las funcionalidades. Es independiente de cualquier representación de salida y comportamiento de entrada. El modelo está representado por los archivos presentes en el directorio Repository.

**Vista:** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador. Las vistas del módulo están agrupadas en el directorio Resources/views.

**Controlador:** Reciben las entradas o eventos que codifican los movimientos o pulsación de botones del ratón o teclas. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista. Las clases controladoras están situadas en el directorio Controller.

En la capa Vista se van a encontrar las clases interfaces con sus respectivos formularios, en el Control estarán las páginas servidoras, y en el Modelo se encontrarán las clases que permiten el acceso y la manipulación de los datos. La vista realizaría peticiones al Control, éste es el encargado de enviarle las peticiones al modelo y el mismo responde a estas peticiones a través del Control hacia la vista, de esta forma el Control sirve de intermediario entre la Vista y el Modelo (Burbeck. 1992). Durante esta investigación se materializa el empleo de este patrón arquitectónico al usar el framework de desarrollo Symfony2, pues este marco de trabajo está diseñado sobre el mismo. A fin de tener una representación más clara de lo que aquí se plantea, se propone la visualización de los diagramas de clases del diseño en el siguiente epígrafe.

## 2.7. Diagrama de clases del diseño.

El diagrama de clases del diseño con estereotipos web presentado a continuación, describe gráficamente las especificaciones del modelo, la vista y el controlador de los casos de usos descritos en el epígrafe 2.5. Esta representación contiene información acerca de las clases, asociaciones, atributos, métodos y dependencias.

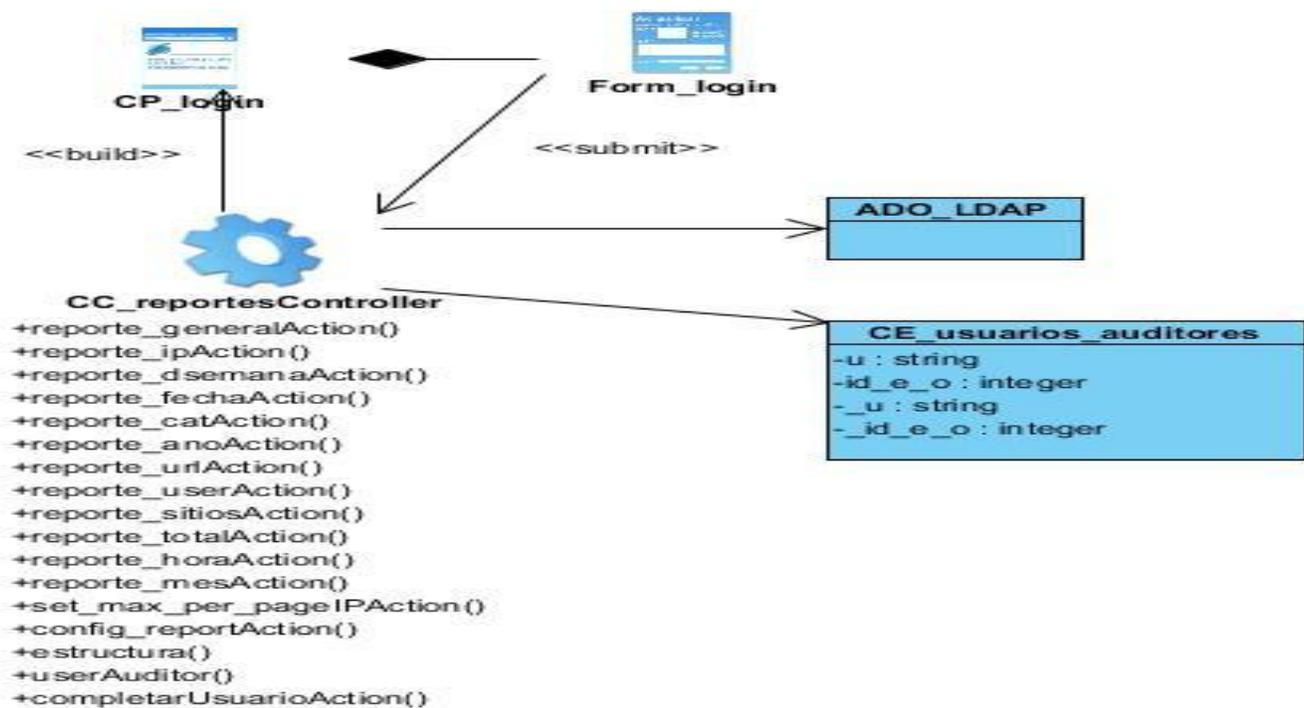


Imagen 3. Diagrama de clases del diseño del caso de uso Autenticar Usuario.

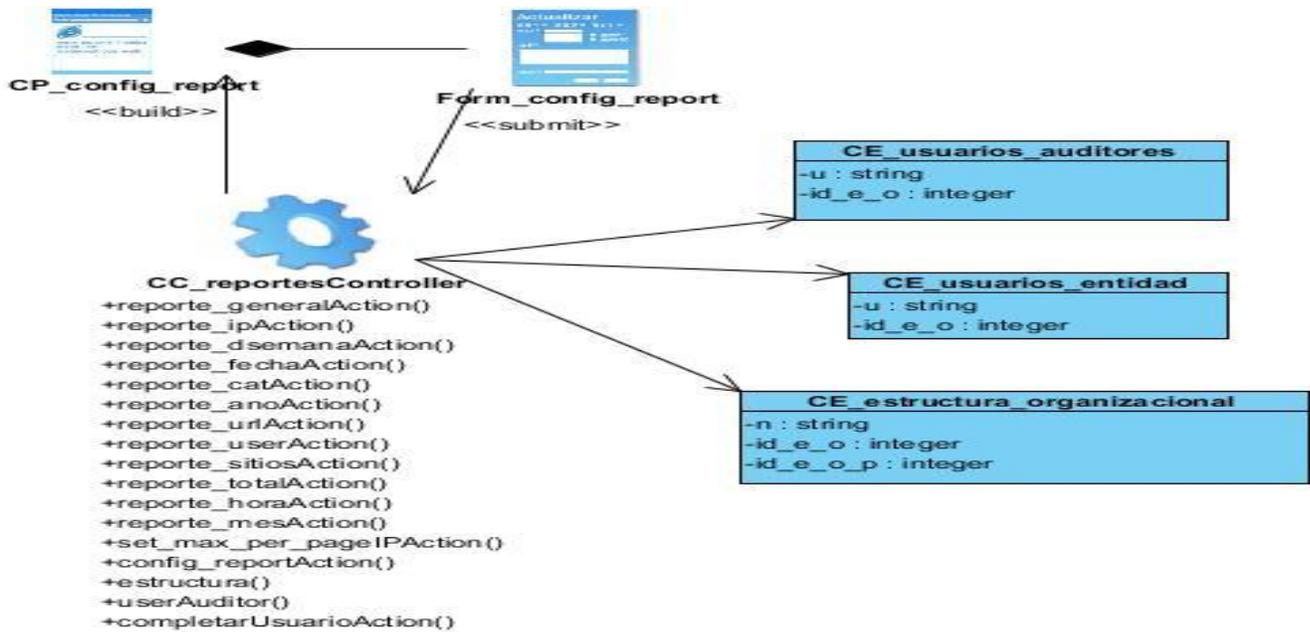


Imagen 4. Diagrama de clases del diseño del caso de uso Seleccionar FilTROS.

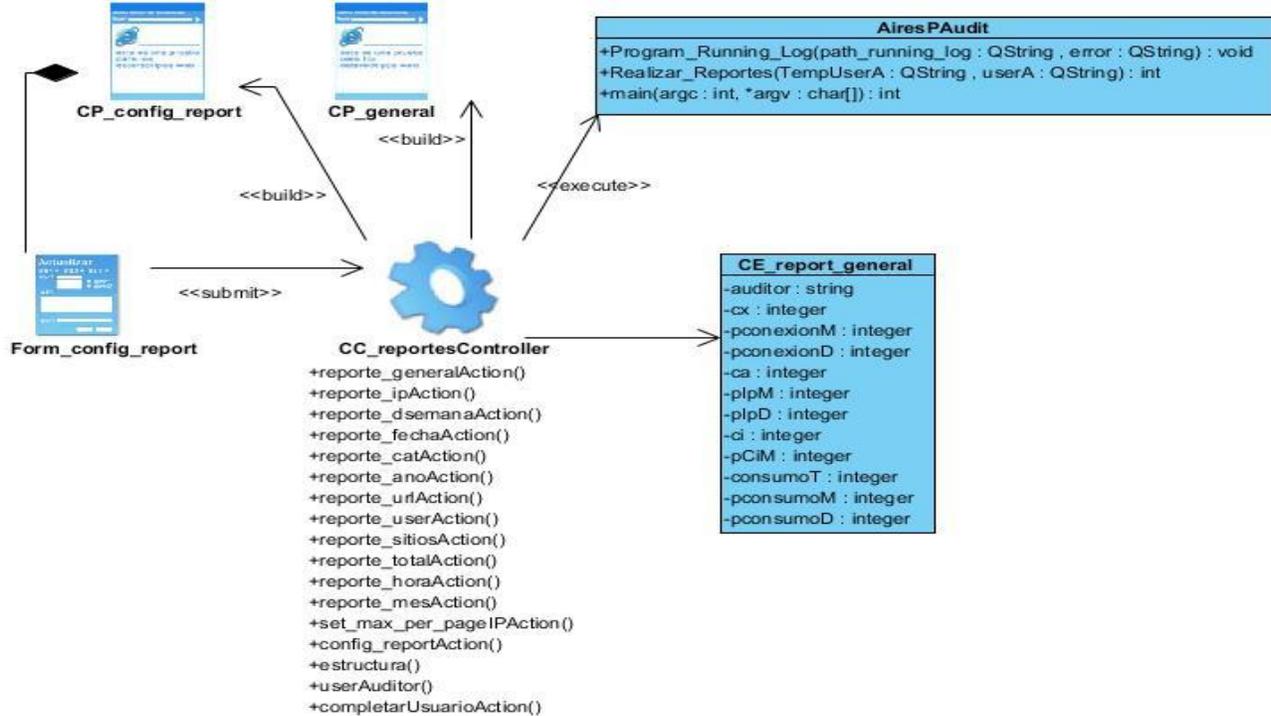


Imagen 5. Diagrama de clases del diseño del caso de uso Generar Reportes.

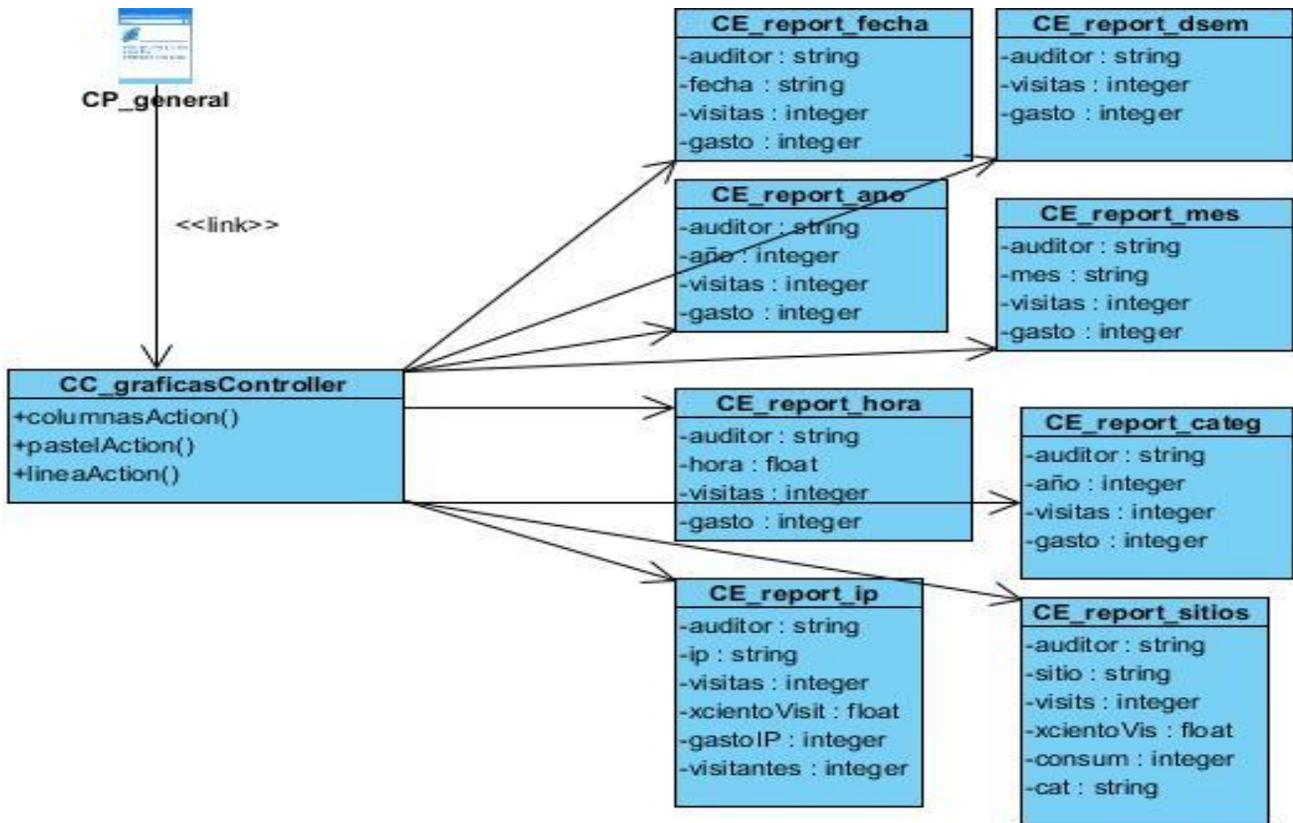


Imagen 6. Diagrama de clases del diseño del caso de uso Mostrar Gráficos.

## 2.8. Diagramas de secuencia.

Los diagramas de interacción son una descripción del modo en el que cada operación detectada lleva a cabo sus responsabilidades y modifica el estado del sistema. En UML los diagramas de interacción pueden representarse a través de los Diagramas de Colaboración y/o de los Diagramas de Secuencia.

El tipo de diagrama seleccionado para la construcción los diagramas de interacción del MGRE para el sistema AiresProxyAudit fue el de secuencia, debido a que muestra cómo los objetos se comunican entre sí en una secuencia de tiempo y para ello contienen los ciclos de vida y los mensajes que se envían entre ellos ordenados secuencialmente.

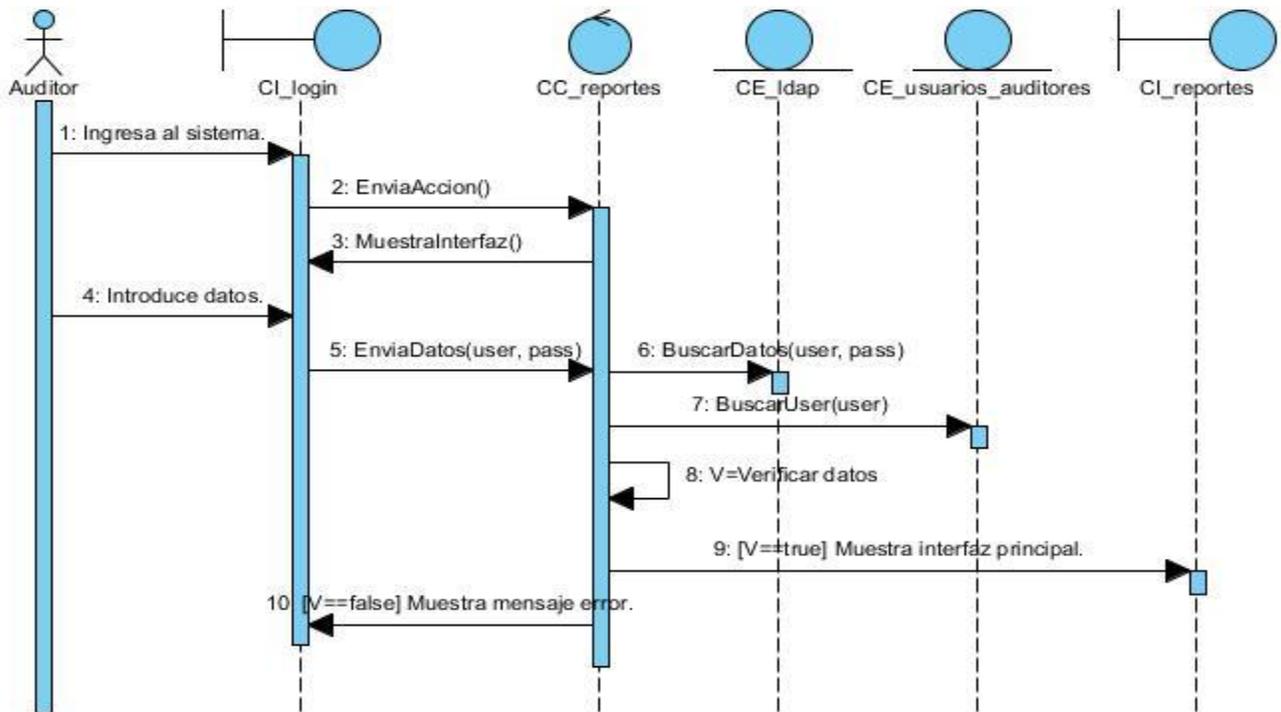


Imagen 7. Diagrama de secuencia del caso de uso Autenticar Usuario.

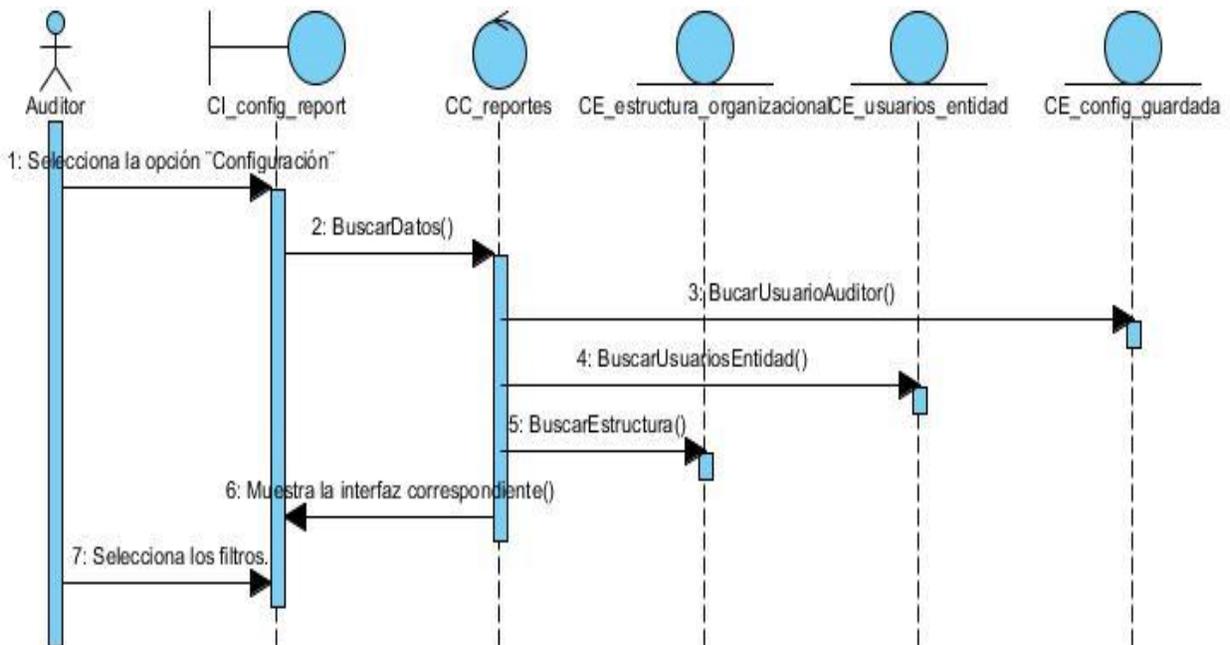


Imagen 8. Diagrama de secuencia del caso de uso Seleccionar Filtros.

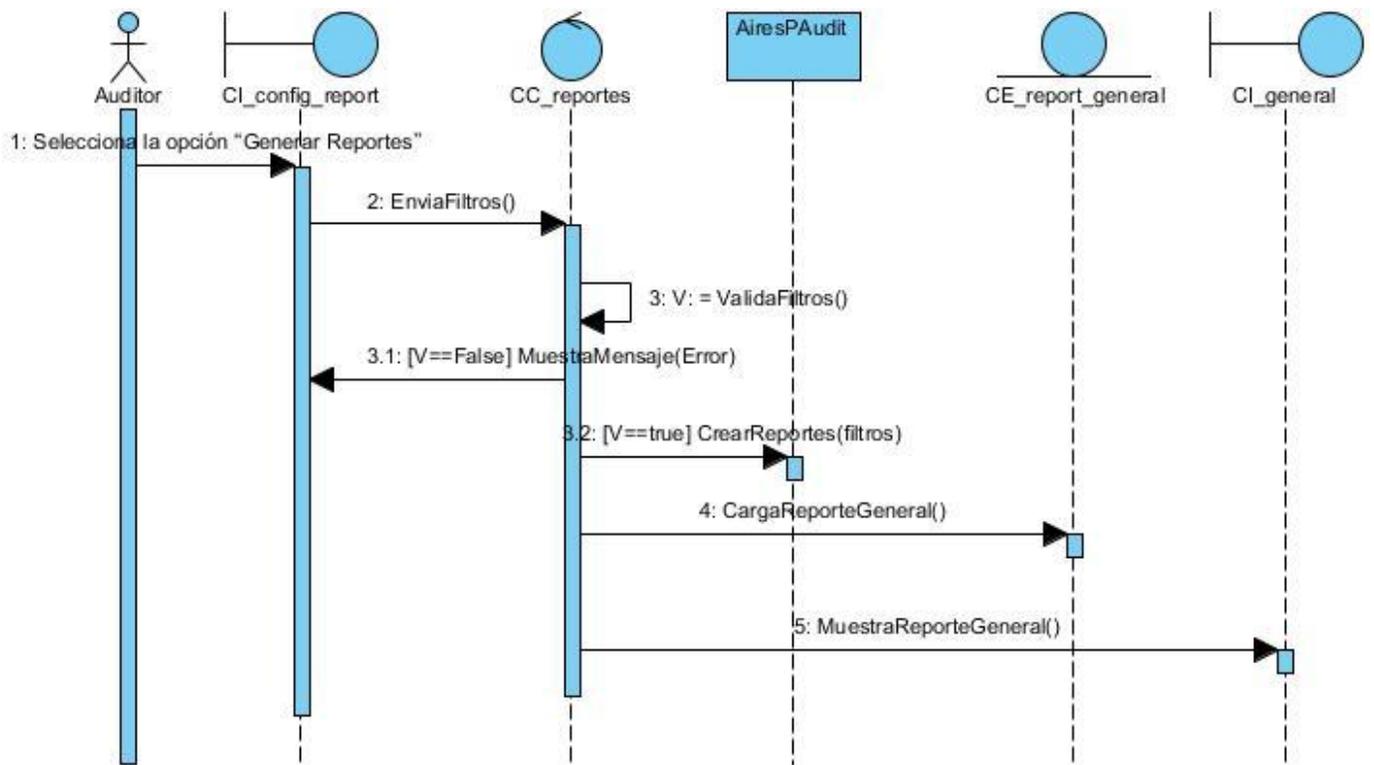


Imagen 9. Diagrama de secuencia del caso de uso Generar Reportes.

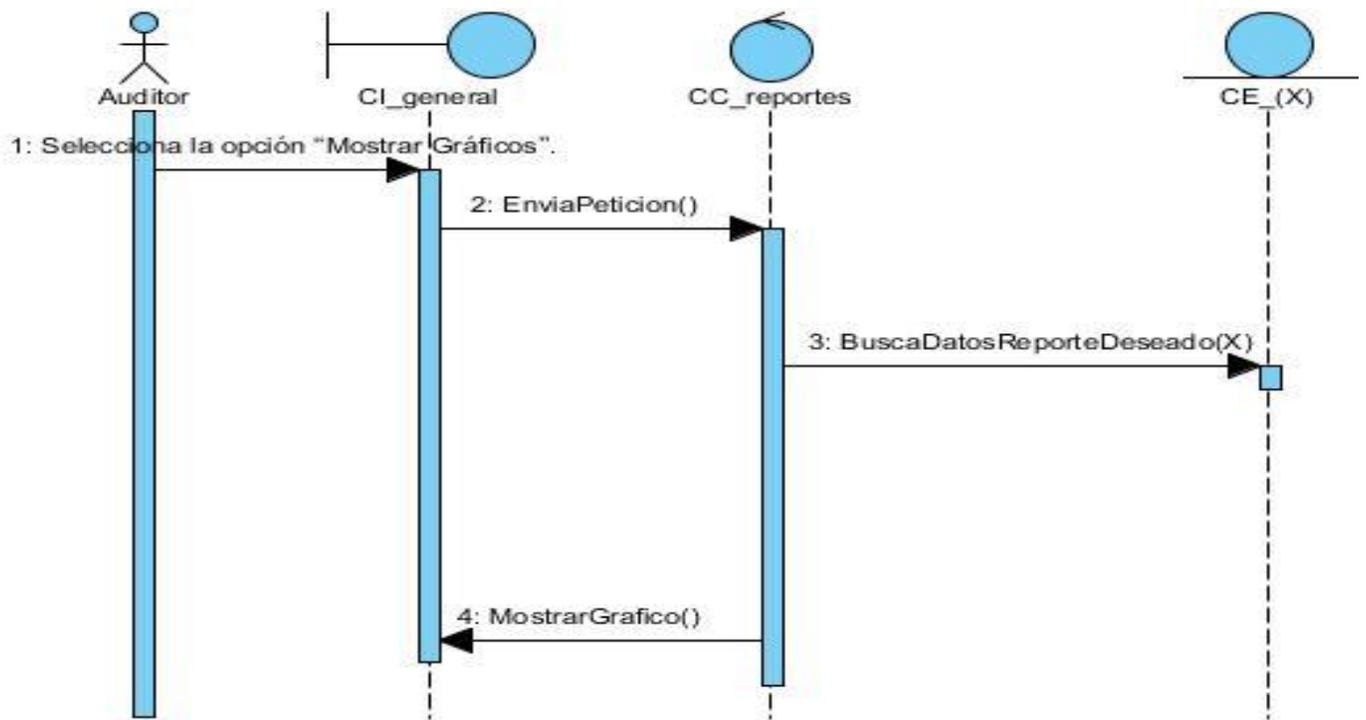


Imagen 10. Diagrama de secuencia del caso de uso Mostrar Gráficos.

### 2.9. Modelo de datos.

En los epígrafes anteriores se muestra la relación existente entre las clases controladoras y las clases entidades; estas últimas conforman los documentos mapeadores de objetos (ODM) pertenecientes al modelo de datos. Los cuales permiten la persistencia y manejo de los datos desde las clases controladoras, al hacer uso de una base de datos MongoDB.

Para un mejor entendimiento de dicho modelo se describe de manera general las colecciones<sup>20</sup> del MGRE reflejadas en la Imagen 11.

usuarios\_audidores: almacena el identificador del auditor (u) y el área a la que este tiene acceso (id\_e\_o).

usuarios\_entidad: almacena el identificador del usuario que puede ser auditado (u) y el identificador del área a la que pertenece (id\_e\_o).

estructura\_organizacional: contiene el nombre de todas las áreas y subáreas (n) y el identificador de cada una de ellas (id\_e\_o), en caso de que sea una subárea se almacena el identificador del área a la que pertenece (id\_e\_o\_p).

La colección cuyo nombre comienza por **report\_** almacena la información correspondiente al reporte al cual hace referencia la segunda parte de su nombre.

report: permite almacenar los reportes realizados, identificándolos por el auditor que lo realiza (auditor) y el nombre que este le asigna (save).

config\_save: se utilizan en la persistencia de las configuraciones realizadas, guardando el auditor (auditor) y los filtros seleccionados.

---

<sup>20</sup> Conjunto de documentos. Un documento es un registro único que tienes varios atributos.

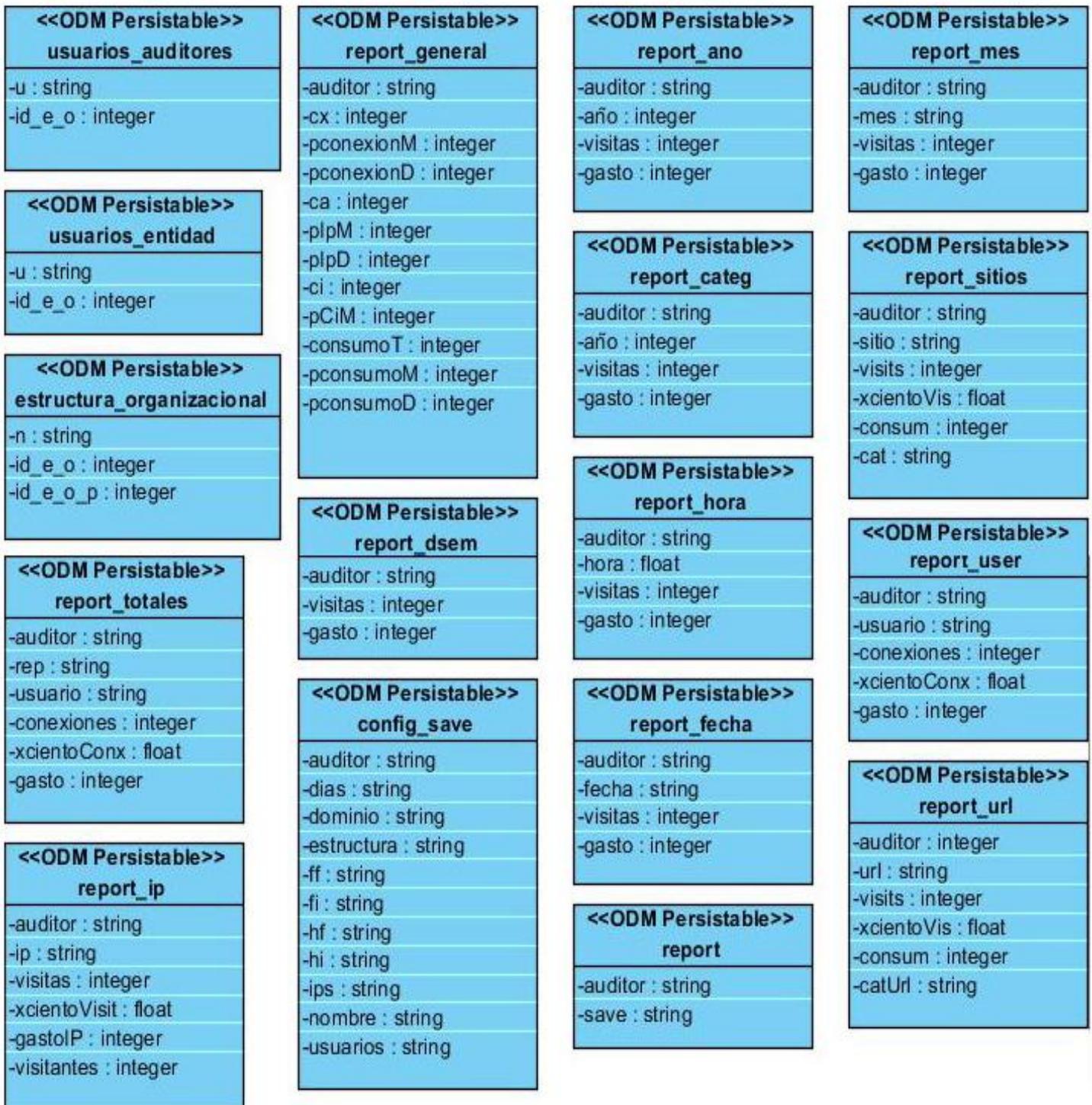


Imagen 11. Modelo de datos del MGRE.

### 2.10. Diagrama de despliegue.

Para lograr un despliegue exitoso de la aplicación se hace necesario mostrar la disposición física de los distintos nodos que componen el sistema y el reparto de los componentes en dichos nodos. El diagrama de despliegue que a continuación se presenta servirá para modelar la topología de hardware necesaria y la distribución del sistema en la UCI.

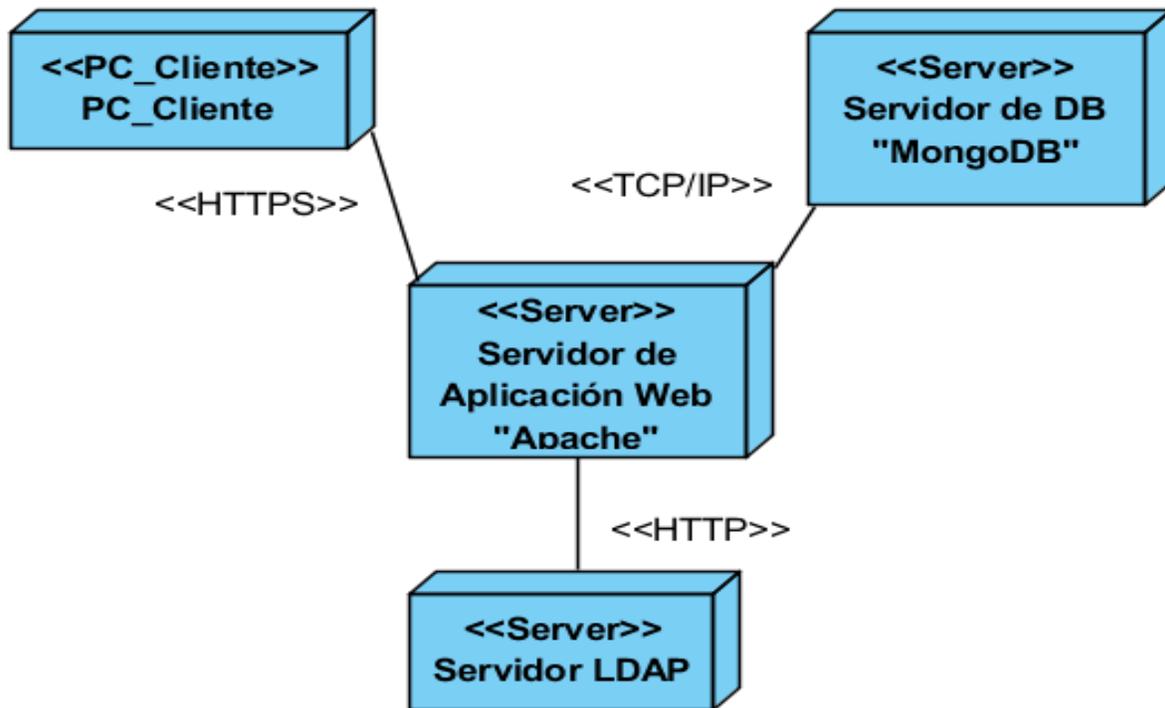


Imagen 12. Diagrama de Despliegue del MGRE para el sistema AiresProxyAudit.

En este diagrama se definen el uso de estaciones de trabajo (PC\_Cliente) que el usuario utilizará para conectarse vía HTTPS<sup>21</sup> con el servidor de aplicaciones web que hospeda al sistema AiresProxyAudit. También se puede apreciar la necesidad de conexión HTTP<sup>22</sup> con un servidor LDAP<sup>23</sup> que garantiza los datos necesarios para la autenticación en el sistema y el uso del protocolo TCP/IP<sup>24</sup> para conectarse con la base de datos del sistema AiresProxyAudit.

<sup>21</sup> Siglas en inglés de HyperText Transfer Protocol Secure- Protocolo Seguro de Transferencia de Hipertexto (Definición ABC, 2013).

<sup>22</sup> Siglas en inglés de HyperText Transfer Protocol - Protocolo de Transferencia de Hipertexto (Definición ABC, 2013).

<sup>23</sup> Protocolo Ligero de Acceso a Directorios (PHP, 2013).

<sup>24</sup> Protocolo de Control de Transmisión/Protocolo de *Internet*, conjunto o familia de protocolos de red sobre los cuales se basa *Internet* (Definición ABC, 2013).

### **2.11. Conclusiones.**

Generar los diferentes artefactos que dispone la metodología OpenUP permitió que:

Modelado el dominio, se identificaron los principales requisitos funcionales y no funcionales del MGRE para el sistema AiresProxyAudit, los cuales fueron agrupados y categorizados por casos de usos.

Identificados los patrones de diseño y arquitectura, la solución propuesta cuenta con un alto grado de resistencia ante posibles modificaciones.

Diseñado el conjunto de diagramas de clases y de componentes se facilitó la visión en cuanto a composición física y lógica del sistema.

Generados todos los artefactos requeridos por el modelo de desarrollo quedó documentada la solución, facilitando su mantenimiento posterior (actualización o adición de funcionalidades).

## Capítulo 3 “Implementación y Pruebas de validación.”

### 3.1. Introducción.

La fase de implementación es considerada el centro de las iteraciones durante la construcción del software, pues al implementar un sistema se materializan las ideas del diseño a través de componentes, es decir, código fuente, *scripts*, ejecutables, entre otros (Jacobson, 2000). Todo ello, asegura que el cliente cuente con una aplicación que funciona de acuerdo a los requerimientos definidos por él durante la fase de análisis.

A fin de garantizar el correcto funcionamiento de la solución se definen las pruebas del software, las cuales garantizan la calidad del software y la revisión final del cumplimiento de las especificaciones del diseño y de la codificación. En el presente capítulo se analiza la etapa de implementación, la cual denota el estado actual del sistema en términos de componentes. Se realiza además la validación de la solución propuesta en los capítulos anteriores.

### 3.2. Diagrama de Componentes.

El diagrama de componentes muestra las relaciones entre las partes físicas y reemplazables del sistema, por tanto, expresa las dependencias existentes entre estas estructuras denominadas componentes. Estos diagramas contienen relaciones de dependencia que se utilizan para indicar que un componente se refiere a los servicios ofrecidos por otro (Pressman, 2006).

El MGRE contiene 4 paquetes principales:

**Controller:** el cual contiene las clases controladoras del sistema, encargadas de la lógica del negocio.

**Views:** almacena las plantillas twig que facilitan el maquetado de las vistas del MGRE al permitir la herencia entre ellas.

**View:** compuesto por las vistas de la aplicación, agrupadas dentro de los paquetes Graficos, PDF y Reportes. Todas ellas intervienen directamente con el usuario.

**Modelo:** conformado por los paquetes Repository y Document los cuales agrupan a las clases cuya función es el acceder y depositar información en la base de datos.

**AiresPAudit:** agrupa los elementos que se encargan de procesar los registros almacenados, generar y almacenar los reportes en la base de datos.

A continuación se presenta el diagrama de componentes general del MGRE para el sistema AiresProxyAudit.

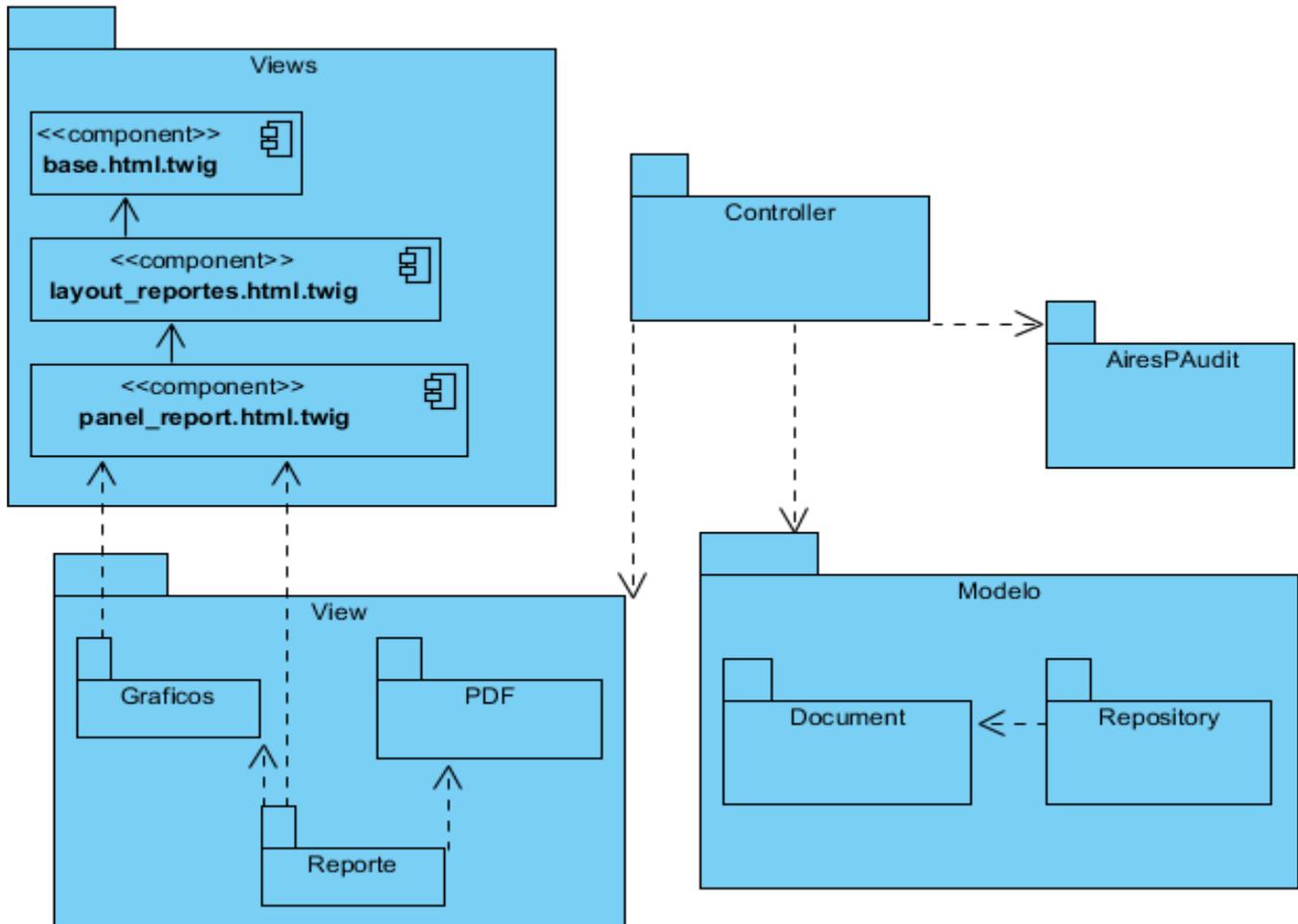


Imagen 13. Diagrama Componentes general agrupados por paquetes.

El paquete Controller, como podemos observar en la Imagen 14 contiene 3 componentes:

**ReportesController.php:** encargado de manejar todos los datos correspondientes con los reportes estadísticos, dígame configuración, visualización y la responsabilidad de enviar la orden de generación de los reportes al paquete AiresPAudit.

**GraficasController.php:** controla todos los eventos referentes a la representación gráfica de los datos que se muestran en los reportes. Estos son obtenidos desde la base de datos, haciendo uso de los componentes agrupados en el paquete Modelo.

**PdfController.php:** este componente se encarga de exportar en formato *pdf* cada reporte, según la necesidad del usuario.

## Capítulo 3 “Implementación y Pruebas de validación.”

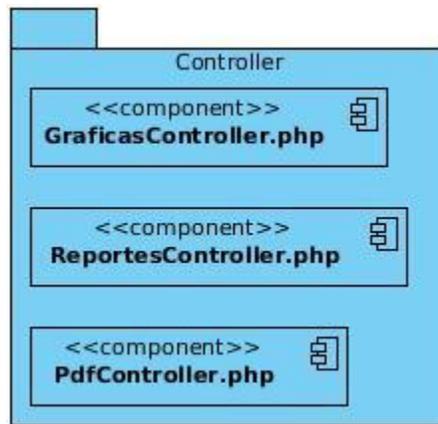


Imagen 14. Paquete Controller.

Dentro del paquete Reporte se agrupan las vistas de los reportes estadísticos que un auditor puede consultar. Además, la interfaz de bienvenida del módulo (index.html.twig) y la página para la selección de los filtros que desean aplicar a los reportes (config\_report.html.twig). Aquí también se encuentran agrupadas todas las interfaces de consulta de los reportes generados, las cuales se pueden observar en la Imagen 15. El nombre de las mismas se relaciona con la información que proporciona.

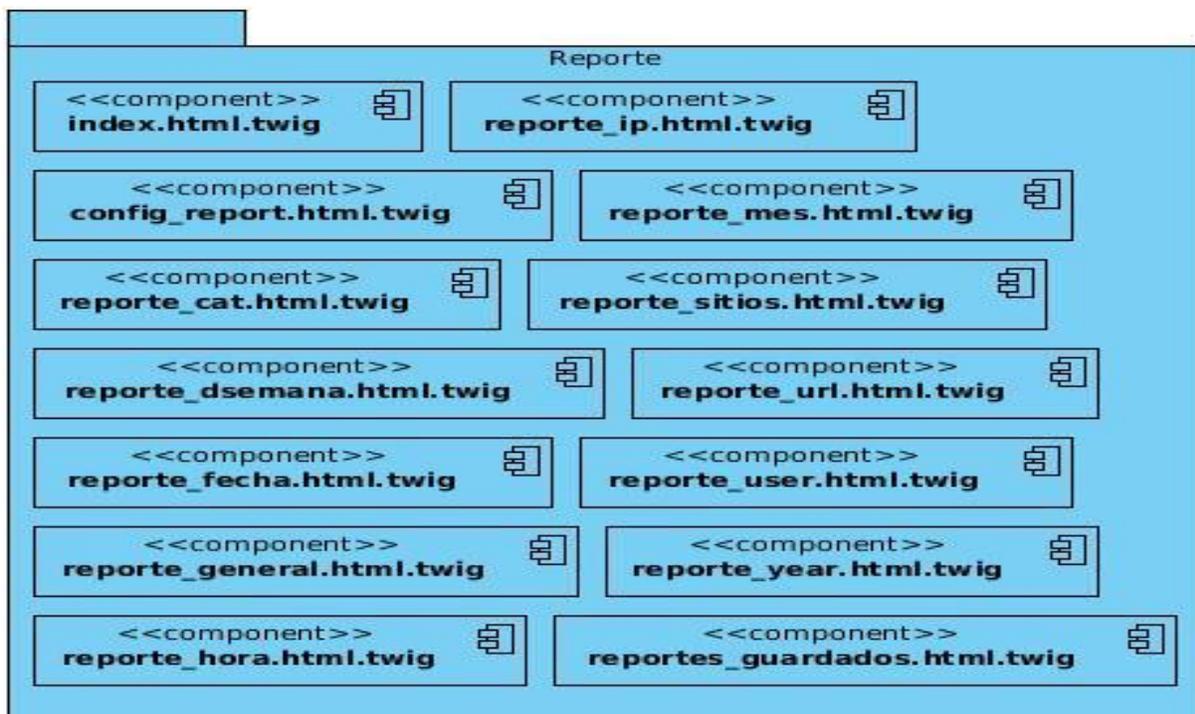


Imagen 15. Paquete Reporte.

## Capítulo 3 “Implementación y Pruebas de validación.”

En el caso de PDF, se compone por las clases que permiten exportar cada reporte estadístico a un documento en formato *pdf*. Una muestra de ellas son: *pdfuser.pdf.twig* y *pdfhora.pdf.twig* las cuales se encargan de exportar a *pdf* los reportes por usuario y por hora respectivamente. En la Imagen 16 se muestra el resto de componentes que pertenecen a este grupo. Los auditores pueden acceder a estas vistas desde la interfaz del reporte al cual hace referencia el nombre de la misma.

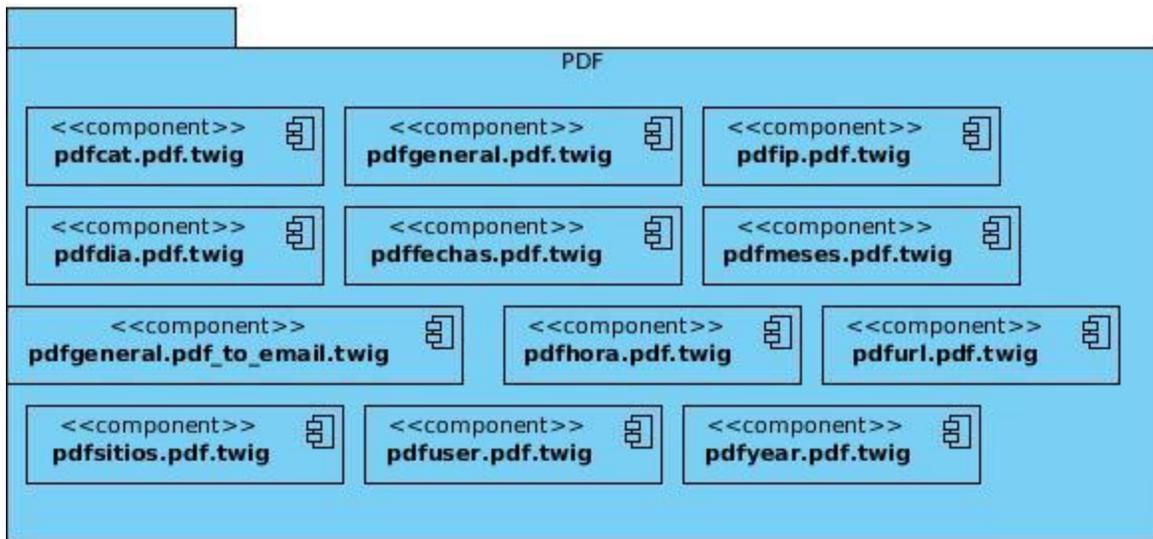


Imagen 16. Paquete PDF.

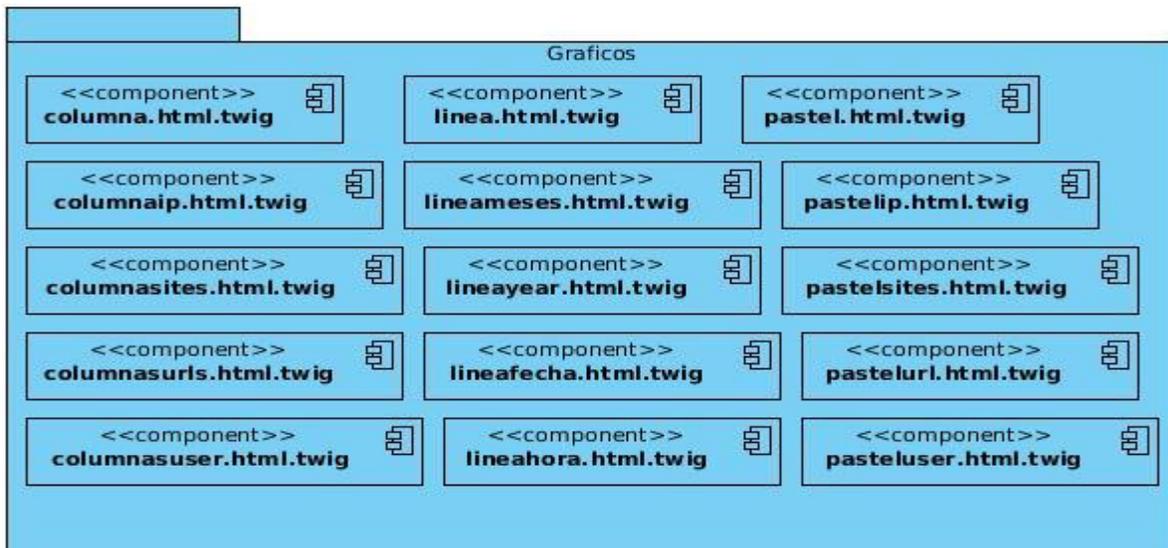


Imagen 17. Paquete Graficos.

Otro de los paquetes del MGRE es Graficos. En él se encuentran las clases que permiten visualizar un reporte estadístico en forma de gráfico. Por ejemplo, las clases *columnasites.html.twig* y *pastelsites.html.twig* permiten

## Capítulo 3 “Implementación y Pruebas de validación.”

la visualización del reporte por sitios en una gráfica de barra y pastel respectivamente. A fin de conocer mejor la estructura de este paquete puede observar la Imagen 17. Estas vistas son accesibles desde la interfaz del reporte al cual hace referencia el nombre de la misma.

Dentro de Repository se agrupan las clases que interactúan directamente con los reportes almacenados en la base de datos. En las clases RporSitiosRepository.php, RporCatRepository.php y RporUserRepository.php se realizan las consultas relacionadas con los reportes por sitios, categoría y usuario respectivamente. El paquete completo puede ser observado en la Imagen 18.

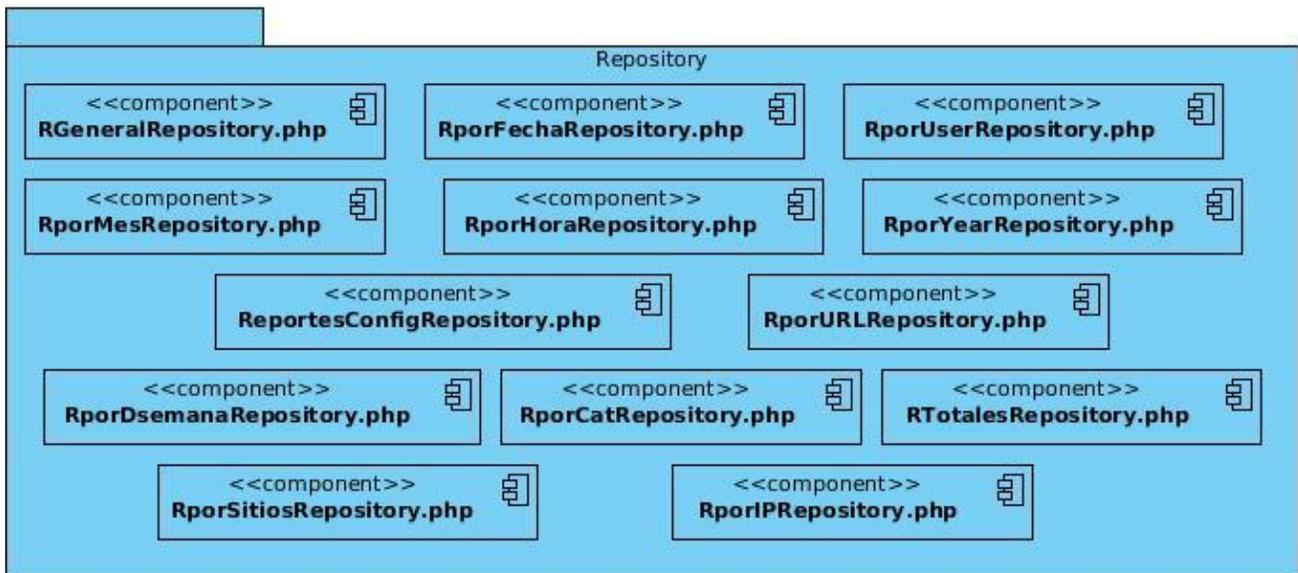


Imagen 18. Paquete Repository.

Todas las clases que se encargan de mapear la información que el usuario inserta o genera desde la aplicación con el fin de guardarla o extraerla de la base de datos componen el paquete Document (Ver Imagen 19). Estos documentos son una representación de las colecciones de datos existente, cuyo fin es facilitar el entendimiento entre el MGRE y la información persistente. La relación existente entre los componentes del paquete Repository y Document se puede observar en el nombre de los mismos. Por ejemplo, **RporCatRepository.php** y **RporCat.php** se encargan de manejar los reportes por categoría almacenados.

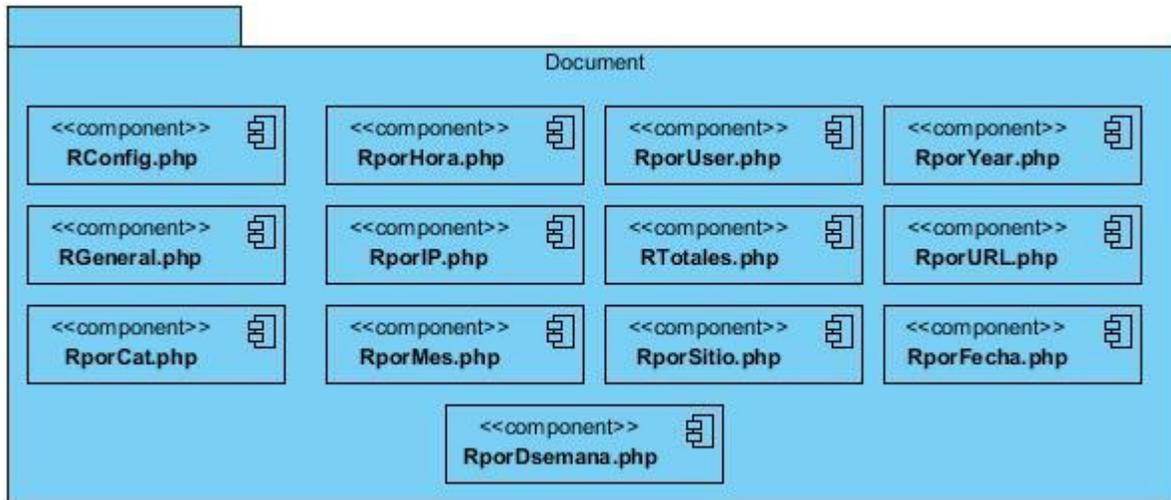


Imagen 19. Paquete Document.

Por último se hace referencia al paquete AiresPAudit, el cual recoge todos los archivos y clases que hacen uso del lenguaje C++ durante el procesamiento y salva de la información en la base de datos. Además, como se puede observar en la Imagen 20, en él se encuentra el archivo AiresPAudit\_DB\_Config.xml que contempla la configuración necesaria para el acceso a los datos.

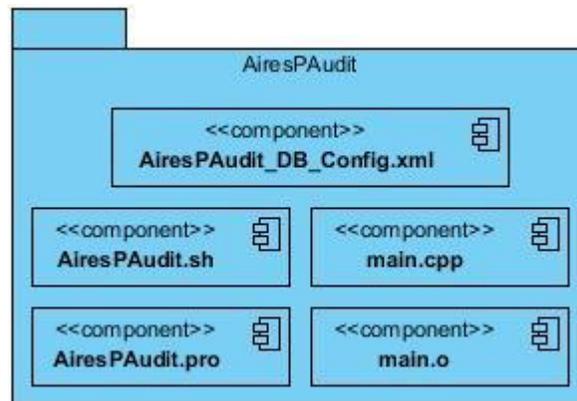


Imagen 20. Paquete AiresPAudit.

Como es posible observar durante este epígrafe todos estos paquetes, excepto AiresPAudit, presentan una similitud en su nomenclatura, ello se debe a la puesta en práctica de algunos estándares de codificación, los cuales se abordan a continuación.

### 3.3. Estándar de codificación.

Un estándar de codificación completo comprende todos los aspectos de la generación de código, asegurando que todos los programadores del proyecto trabajen de forma coordinada. Usar técnicas de codificación sólidas

## Capítulo 3 “Implementación y Pruebas de validación.”

y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es sinónimo de un software de buen rendimiento y fácil mantenimiento (Microsoft, 2013).

En el MGRE la aplicación de este enfoque es visible desde la nomenclatura de los archivos como se mencionaba anteriormente, hasta la composición de los mismos. A continuación se detallan aspectos referentes a los estándares utilizados.

### **Identificadores**

Todo el código del MGRE sigue un estándar para nombrar los archivos, las clases, funcionalidades y variables, pues se establece que el fin de estos identificadores sea entendible solo con ser leídos por cualquier usuario ya sea miembro de equipo de desarrollo o no. Se garantiza así la facilidad en la realización de posibles mejoras y mantenimientos que puedan ser llevados a cabo por otras personas ajenas al proceso de implementación inicial.

Al observar con detenimiento las clases controladoras de la solución se observan funcionalidades que se rigen por el estilo UpperCamelCase, donde todas las palabras que conforman un identificador comienzan con mayúsculas y de forma continua (ReportesController y RporFecha.php). También podemos observar el lowerCamelCase, el cual establece el inicio con minúscula de un identificador y en caso de estar conformado por más palabras estas comienzan con mayúscula inmediatamente después de la anterior. Por ejemplo: `indexAction()`, `getRepository()`, entre otros.

Además, también se encuentran funcionalidades que sus nombres están escritos siguiendo el estándar lowerCamelCase pero que las palabras que conforman el mismo están separadas por un guión (`set_max_per_pageIPAction()`). Esto se debe a que son métodos utilizados a través del lenguaje JavaScript, permitiendo la diferenciación en cuanto a la finalidad de los mismos.

### **Indentación**

La Indentación es una práctica de programación que consiste en comenzar a escribir cada línea de código a diferentes distancias desde el borde izquierdo del área de texto del editor. Esta separación está determinada por la jerarquía que se forma al introducir sentencias dentro de bloques de estructuras, permitiendo una mayor legibilidad. En el fragmento de código que a continuación se muestra, es posible observar con se aplica este procedimiento en el MGRE.

## Capítulo 3 “Implementación y Pruebas de validación.”

```
if (count($hijos) != 0) {
    $sus [] = "hijos";
    foreach ($hijos as $v) {
        $res = $this->arbol($v->getIdEO());
        foreach ($res as $key => $value) {
            $sus[] = $value;
        }
    }
    $sus [] = "endhijos";
}
```

Imagen 21. Fragmento de código del MGRE con indentación.

### Comentarios

La función de los comentarios es ayudar a entender lo que escriben los programadores y facilitar el trabajo durante el proceso de desarrollo, pues de ser necesario reutilizar un código, resulta más factible si este presenta notas que describan su funcionamiento evitando el estudio completo del mismo.

El MGRE utiliza el estilo javadoc, que como se muestra en la Imagen 22, establece comentarios en bloques, especificando detalladamente el funcionamiento de las clases y los métodos implementados dentro del módulo al inicio de los mismos.

```
/*
 * Método encargado de renderizar la plantilla de bienvenida del módulo.
 * Además, crea y obtiene las variables de sesión que son utilizadas por la interfaz.
 * @param null
 * @return plantilla index.html.twig renderizada
 * @throws null
 */
public function indexAction() {
    $session = $this->getRequest()->getSession();
    $session->set('menu', 'homepage');
    if ($this->get('security.context')->isGranted('ROLE_administrador')) {
        $this->get('request')->getSession()->set('admin', "true");
        $this->get('request')->getSession()->save();
        session_write_close();
    } else {
        $this->get('request')->getSession()->set('admin', "false");
        session_write_close();
    }
    return $this->render('ReportesBundle:Reportes:index.html.twig', array());
}
```

Imagen 22. Ejemplo de comentario usado en el MGRE.

## 3.4. Pantallas principales del MGRE

Una vez desarrollado el software es posible visualizar las pantallas principales del mismo donde se observa el resultado obtenido durante la implementación de los casos de usos críticos descritos en el capítulo anterior. Como anexos a esta investigación se presentan otras interfaces del mismo.



Imagen 23. Interfaz de acceso al MGRE. Solución al caso de uso Autenticar Usuario.

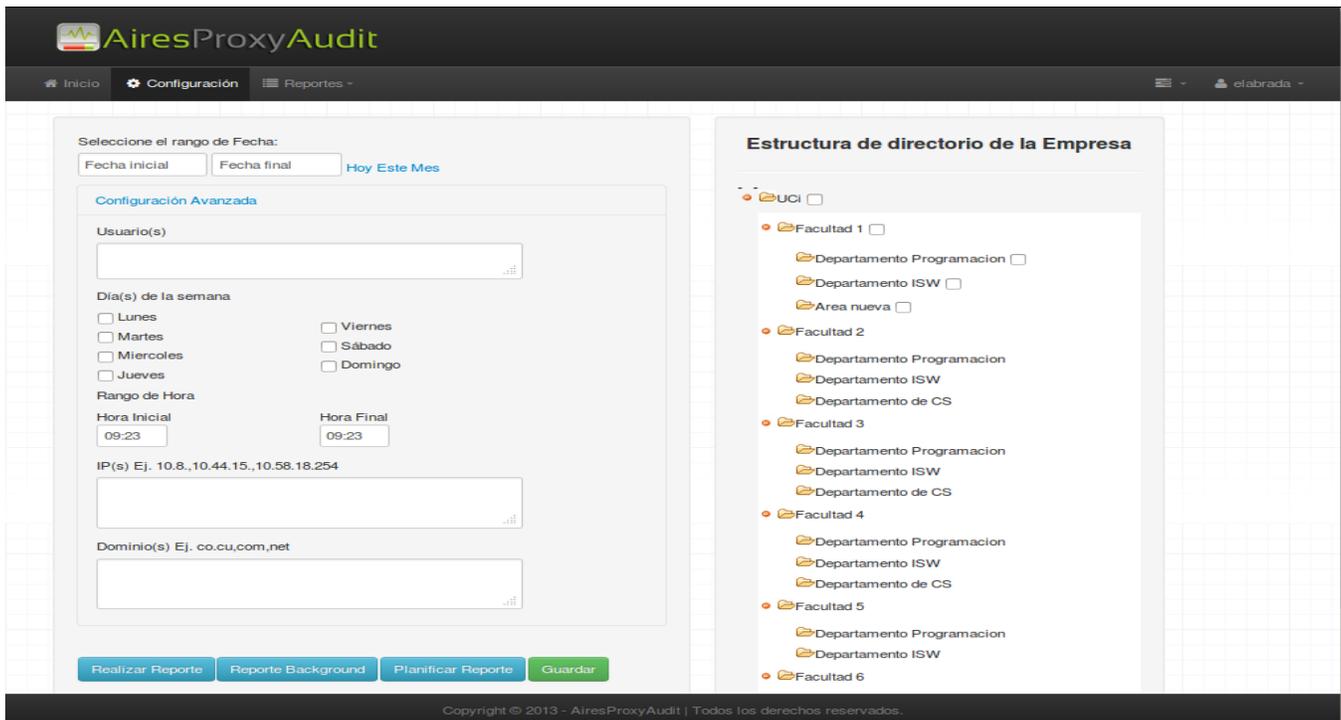


Imagen 24. Interfaz de selección de filtros para reportes. Solución al caso de uso Seleccionar filtros.

## Capítulo 3 “Implementación y Pruebas de validación.”

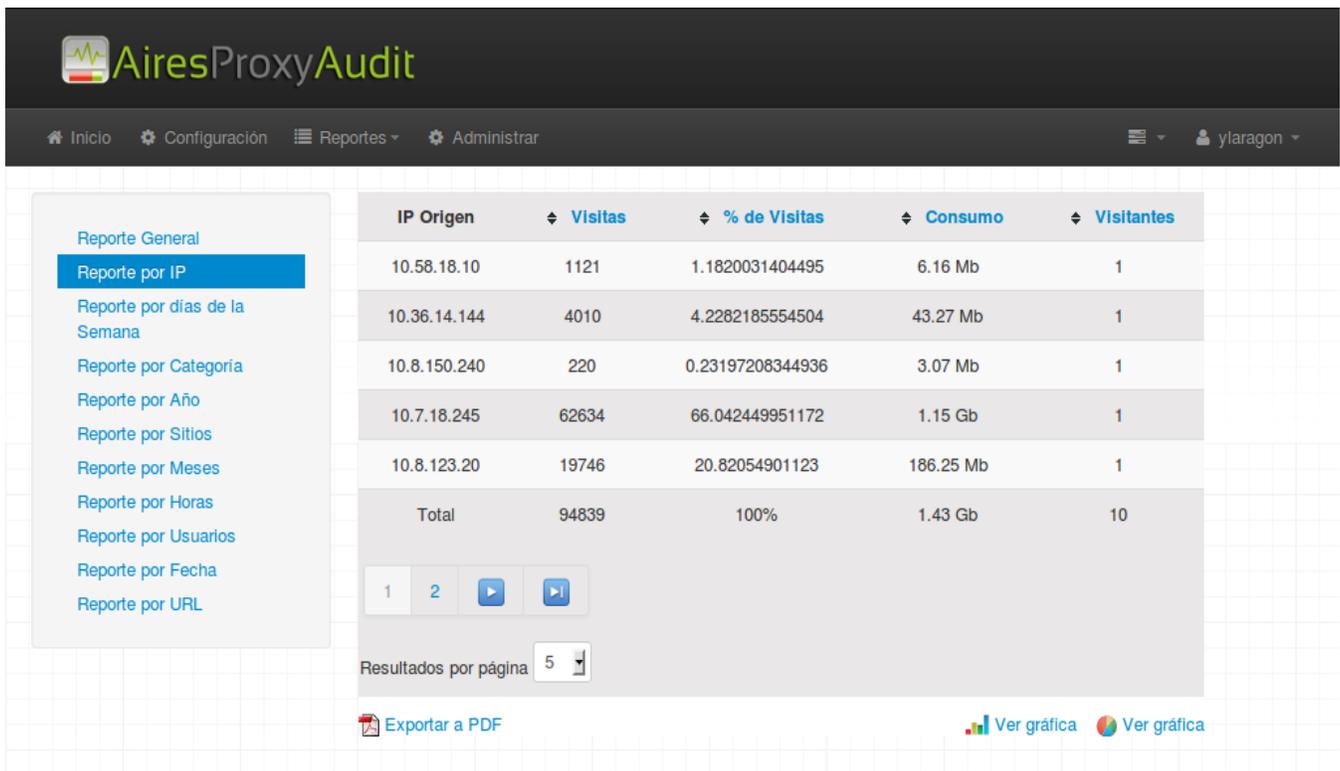


Imagen 25. Interfaz de reportes generados. Solución al caso de uso Generar Reportes.

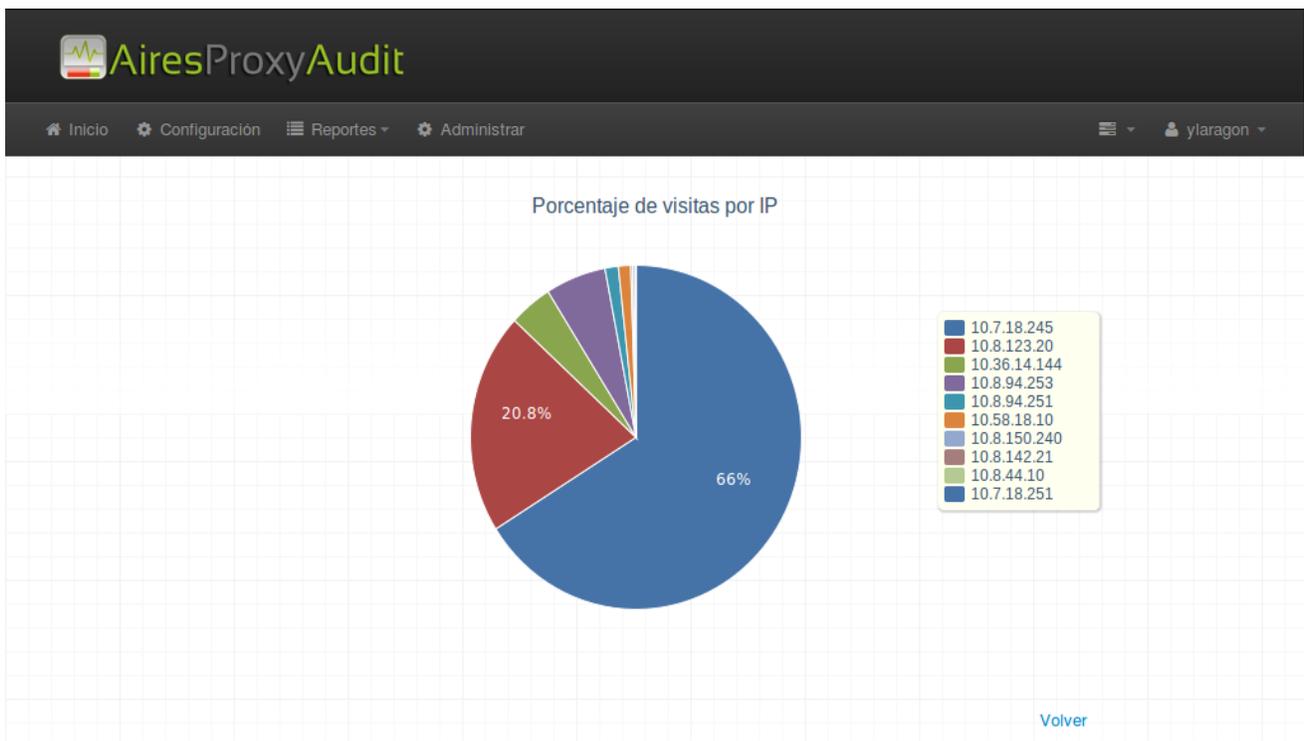


Imagen 26. Interfaz de reporte gráfico por IP. Solución al caso de uso Mostrar Gráficos.

### 3.5. Validación del sistema

El desarrollo de un software es algo complejo y son innumerables las posibilidades de cometer errores. Por esta razón todo proceso de implementación debe ir acompañado de alguna actividad que garantice la calidad. Las pruebas constituyen una base para garantizar la aceptación favorable de una aplicación informática por parte del usuario. Con la realización de las mismas se pretende encontrar y documentar los errores que tiene un sistema, validar los requisitos y comprobar que estos fueron implementados correctamente. Con el objetivo de garantizar la calidad del MGRE se documenta en el presente epígrafe las pruebas realizadas al mismo.

#### 3.5.1. Pruebas funcionales

Las pruebas funcionales tienen como objetivo verificar el funcionamiento del sistema al fijar la tensión en la validación de las funciones, métodos, servicios y casos de usos (Toll, 2007). Con el propósito de desarrollar las mismas sobre el MGRE, se diseñaron un conjunto de casos de pruebas referentes a los casos de usos críticos especificados en el capítulo 2. Dichas pruebas se reflejan a continuación.

#### Caso de prueba del CU Autenticar usuario.

El objetivo de este caso es comprobar el correcto funcionamiento en el acceso al sistema. Se comprueban los 3 escenarios posibles según la descripción del CU Autenticar Usuario.

**Condiciones de ejecución:** El usuario ha sido creado en el sistema.

Escenario	Descripción	Usuario	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 Loguearse en el sistema correctamente	El usuario introduce los datos al sistema y este comprueba que los mismos sean correctos y que el usuario exista en el sistema, entonces se redirecciona al usuario a la interfaz	V	V	Comprueba que los datos introducidos sean correctos y que el usuario exista en el sistema. Redirecciona al usuario a la interfaz principal.	1. Se llenan los datos de entrada (Usuario, Contraseña) 2. Selecciona la opción “Iniciar Sesión”

### Capítulo 3 “Implementación y Pruebas de validación.”

	principal.				
		elabrada	ElabradaO*-		
EC 1.2 Loguearse en el sistema con campos vacíos	El usuario introduce los datos al sistema y este comprueba que los mismos sean correctos. En caso de no serlo se muestra un mensaje de error.	V	I	Comprueba que los datos introducidos sean correctos, en caso contrario muestra el mensaje "Introduzca sus datos"	1. Se llenan los datos de entrada (Usuario, Contraseña) 2. Selecciona la opción "Iniciar Sesión"
		elabrada			
EC 1.3 Loguearse en el sistema con datos incorrectos.	El usuario introduce los datos al sistema y este comprueba que el mismo exista en la base de datos. En caso contrario se muestra un mensaje de error.	V	V	Comprueba que el usuario exista en el sistema, de no existir muestra el mensaje "Verifique sus datos".	1. Se llenan los datos de entrada (Usuario, Contraseña) 2. Selecciona la opción "Iniciar Sesión"
		machavez	Contraseña		

Tabla 7. Caso de prueba del CU Autenticar usuario.

#### Caso de prueba del CU Generar reportes.

Esta prueba comprueba el correcto funcionamiento de módulo en cuanto a la generación de reportes estadísticos.

## Capítulo 3 “Implementación y Pruebas de validación.”

**Condiciones de ejecución:** Se han seleccionado filtros o se ha importado una configuración.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Mostrar reportes.	Se muestra una interfaz con los reportes generados.	Muestra una interfaz con los reportes generados.	1, Selecciona la opción “Generar Reportes”

Tabla 8. Caso de prueba del CU Generar reportes.

### Caso de prueba del CU Gestionar reportes guardados.

Este caso persigue comprobar que una vez autenticado el auditor este tenga acceso a visualizar sus reportes almacenados.

**Condiciones de ejecución:** El auditor se ha autenticado en el sistema.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Cargar reporte guardado	Se carga un reporte previamente guardado.	Muestra la interfaz que permitirá cargar un reporte.	1. Se selecciona. La opción “Cargar reporte guardado”. 2. Selecciona el reporte a cargar. 3. Selecciona la opción “Cargar”.

Tabla 9. Caso de prueba del CU Gestionar reportes guardados.

Como resultado final de este tipo de pruebas se obtuvo que:

Fueron diseñados 12 casos de pruebas donde 10 arrojaron los resultados esperados y 2 insatisfactorios en la primera iteración. Se detectaron 2 no conformidades, las mismas referentes a la validación del campo Ip(s) y Dominio(s) en la vista Configuración.

Con el objetivo de obtener un producto de calidad y con la menor cantidad posible de errores se ejecutó una segunda iteración de prueba, donde fueron corregidas las no conformidades detectadas en un tiempo de 72 horas obteniendo así resultados satisfactorios.

## Capítulo 3 “Implementación y Pruebas de validación.”

### 3.5.2. Pruebas de seguridad.

Las pruebas de seguridad se realizan con el objetivo de verificar que los mecanismos de protección incorporados en el sistema lo protegen de datos y accesos impropios. Además, se encargan de certificar que los datos y las funciones del sistema solo son accesibles por los actores debidamente autorizados (Toll, 2007).

Para la realización de estas pruebas se utilizó la herramienta Acunetix, la cual realiza automáticamente auditorías a aplicaciones web comprobando vulnerabilidades de Inyección SQL, Cross site scripting y otras vulnerabilidades que puedan ser explotadas por hackers. Al realizar las pruebas a la aplicación solo se detectaron dos enlaces rotos. Estos fueron eliminados pues formaban parte de la estructura original del framework, no siendo utilizados por el MGRE.

Otro de los mecanismos utilizados para la realización de estas pruebas fue el empleo de la lista de chequeo definida por Calisoft<sup>25</sup> en el primer nivel, a fin de detectar vulnerabilidades de forma manual. De acuerdo a las especificaciones descritas en la misma se realizaron pruebas a nivel de usuario para comprobar la efectividad del sistema a la hora de autenticarse, se observó si se mostraban mensajes inadecuados que pudieran revelar las cuentas válidas existentes en el mismo, se verificó que solo los usuarios habilitados tuviesen acceso al sistema, entre otras. Como resultado de este proceso se obtuvo que para los elementos definidos en la lista de chequeo el sistema no arrojó vulnerabilidades, respondiendo de forma correcta ante acciones inapropiadas.

### 3.5.3. Prueba de carga y estrés

Esta prueba generalmente se refiere a la práctica de comprobar el comportamiento de una aplicación mediante cargas o entradas pesadas. Las mismas se realizan con el fin de verificar si el sistema satisface los requisitos de rendimiento para situaciones críticas como pueden ser: la cantidad límite de usuarios accediendo de forma concurrente a los servicios brindados, documentos extremadamente grandes, cantidad de transacciones que se pueden procesar de forma concurrente cada minuto, tiempo de respuesta, entre otros (ITI, 2010).

Para la realización de estas pruebas se utiliza Apache Jmeter. Esta herramienta se caracteriza por ser libre de uso y mostrar los resultados de las pruebas en una amplia variedad de informes y gráficas, con grandes cantidad de variables que permiten interpretar los resultados desde diferentes puntos de vista (Apache, 2013). A continuación se describen las variables que exponen el estado en el que se encuentra un software:

**Muestras:** Cantidad de páginas (Hilos) que simulan la cantidad de usuarios que están interactuando con el sistema desde la misma URL.

---

<sup>25</sup> Centro que regula y comprueba la calidad del software en la UCI.

### Capítulo 3 “Implementación y Pruebas de validación.”

**Media:** Media de páginas que se cargaron de manera satisfactoria.

**Mediana:** Tiempo promedio que han tardado en cargarse las páginas.

**Min:** Tiempo mínimo que ha demorado en cargarse una página.

**Max:** Tiempo máximo que ha tardado en cargarse una página.

**Línea 90 %:** 90 por ciento de las páginas que se cargaron de manera satisfactoria.

**%Error:** Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

**Kb/Seg:** Velocidad de carga de las páginas.

**Tiempos de Respuestas:** Total del tiempo que demoró en cargarse la cantidad de hilos de esa prueba.

Se realizó una simulación de carga al sistema con un total de 55 y 125 peticiones enviadas respectivamente mostrando los siguientes resultados:

Usuarios	Muestras	Media	Mediana	Min	Max	Línea 90 %	%Error	Kb/Seg	Tiempos de Respuestas
55	14890	15137	53	11935	28402	15.18	0.00%	38.84	1.4/seg
125	5335	7010	110	28332	415516	18.985	0.009%	80022.7	4.090/seg

Tabla 10. Resultados arrojados por la prueba de carga y estrés.

De esta prueba se destacan las siguientes observaciones a tener en cuenta por el equipo de desarrollo de la aplicación a fin de lograr una mayor eficiencia:

1. Para un total de 125 usuarios el sistema muestra un soporte de concurrencia de aproximadamente 5335 peticiones simultáneas. Significando esto que con las prestaciones del servidor en el que se encuentra instalado puede soportar un máximo de 125 usuarios concurrentes ya que con una cantidad superior a esta cifra no respondería las peticiones.
2. El sistema ha sido probado en condiciones inferiores a las reales en las que se debe desplegar, siendo esta la causa principal del 0.009% de error. Con las prestaciones anteriormente mencionadas y con la

## Capítulo 3 “Implementación y Pruebas de validación.”

cantidad de usuarios que soporta esta aplicación se considera satisfactorio su rendimiento. En general las peticiones tienen un tiempo de respuesta de 18.985 milisegundos lo cual demuestra que ha sido implementado haciendo uso de buenas prácticas de programación.

### 3.7. Conclusiones

Modelar el diagrama de componentes permitió obtener una visión más clara sobre la estructura del sistema desarrollado.

Establecer estándares de codificación posibilitó que el código generado contara con la legibilidad necesaria para facilitar la lectura, comprensión y mantenimiento del mismo.

Realizar la implementación del sistema viabilizó la obtención de una aplicación funcional y completamente operativa.

Con la realización de pruebas al *software* se validó la solución desarrollada y se comprobó que la misma respondiera a los requerimientos definidos.

## Conclusiones Generales.

De manera general la presente investigación concluyó con el desarrollo del Módulo de Gestión de Reportes Estadísticos para el sistema AiresProxyAudit de la UCI, el cual agiliza el proceso de auditorías de áreas por parte de los encargados de las mismas. Esta rapidez se debe a que esta herramienta es capaz de extraer y analizar automáticamente los *logs* provenientes del servidor *proxy* y facilita el agrupamiento de los mismos por grupos de usuarios o áreas de trabajo dentro de la entidad en cuestión.

### Otros aspectos significativos que se pueden destacar son:

- El desarrollo de una interfaz web para la visualización de reportes y la selección de los filtros aplicar sobre este, dota a los auditores de una herramienta accesible desde cualquier estación de trabajo dentro de la UCI.
- Al procesar y almacenar la información haciendo uso del lenguaje C++ y de una base de datos no relacional, se redujo considerablemente el tiempo de procesamiento de la información. Permitiendo al auditor contar con 11 reportes en el menor tiempo posible.
- La realización de pruebas permitió validar el MGRE, a fin de entregar al cliente una solución confiable que puede ser usada para la realización de estudios sociales que faciliten la toma de decisiones y la implantación de nuevas políticas de seguridad.
- El MGRE es una herramienta libre que puede ser adaptada a las condiciones de otras empresas que lo requieran, por lo que puede ser considerado una fuente de ingreso a la UCI y/o a Cuba.

## **Recomendaciones.**

Para el desarrollo de futuras investigaciones que guarden relación con la presente, se propone:

- Adicionar técnicas de inteligencia artificial, que permitan la detección de anomalías de forma automática.
- Integrar los sistemas AiresProxy y AiresProxyAudit.

## Referencias Bibliográficas.

**Álvarez, Miguel Ángel. 2009.** *Qué es HTML 5.* [En línea] 2012. [Citado el: 22 de Octubre de 2012.] <http://www.desarrollo.com/articulos/que-es-html5.html>.

—. **2001.** *Veamos qué es JavaScript y las posibilidades que nos ofrece utilizar este lenguaje a la hora de desarrollar páginas web.* [En línea]. 2013 [Citado el: 7 de abril de 2013.] <http://www.desarrolloweb.com/articulos/490.php>

**Apache Jmeter, 2013.** *Apache JMeter™.* [En línea]. 2013 [Citado el: 7 de abril de 2013.] <http://jmeter.apache.org/>.

**AWStats. 2012.** *Free log file analyzer for advanced statistics.* [En línea] 2012. [Citado el: 23 de noviembre de 2012.] <http://awstats.sourceforge.net/>.

**Barrera, Otniel. 2006.** *ISAWEB. Monitoreo de Tráfico en Internet.* La Habana: MIC, 2006.

**Barrett, B, 2012.** *Home of the Webalizer.* [En línea] Enero de 2012. [Citado el: 23 de noviembre de 2012.] <http://www.webalizer.org/>.

**Belgrano, Universidad de Argentina.** *Metodología de Auditoría Informática.* [En línea] [Citado el: 18 de noviembre de 2012.]

<http://www.ub.edu.ar/catedras/ingenieria/auditoria/metodologia/metodologia.htm>.

**Brick, Marketing. 2012.** *What is a Log File? - Define a Log File.* [En línea] 2012. [Citado el: 10 de noviembre de 2012.] <http://www.brickmarketing.com/define-log-file.htm>.

**Burbeck, Steve. 1992.** *Application programming in Smalltalk-80: How to use Model-View-Controller (MVC).*

**Chávez, Miguel Ángel; Hernández, Jorge Luis, 2010.** *QTOCTAVIZ: Módulo para la visualización científica integrado al sistema matemático Octaviz.* La Habana, Cuba: Universidad de las Ciencias Informáticas.

**De Los Angeles, Leodán. 2012.** *MARCOS DE TRABAJO PARA EL DESARROLLO DE APLICACIONES WEB.* [En línea] 2012. [Citado el: 7 de Abril de 2013.] [http://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/4139/1/UCIENCIA-2012-T47-P512-Ponencia-1772.pdf](http://repositorio_institucional.uci.cu/jspui/bitstream/ident/4139/1/UCIENCIA-2012-T47-P512-Ponencia-1772.pdf).

**Definicion.de, 2013** [En línea] 2013. [Citado el: 15 de febrero de 2013] <http://definicion.de/reporte/>

## Referencias Bibliográficas.

- . 2013 [En línea] 2013. [Citado el: 15 de febrero de 2013.] <http://definicion.de/estadistica/>
- . 2013 [En línea] 2013. [Citado el: 25 de mayo de 2013.] <http://definicion.de/modelo-de-datos/>
- Definicion ABC, 2013** [En línea] 2013. [Citado el: 20 de febrero de 2013.] <http://www.definicionabc.com/tecnologia/http.php>
- . 2013 [En línea] 2013. [Citado el: 20 de febrero de 2013.] <http://www.definicionabc.com/tecnologia/https.php>
- . 2013 [En línea] 2013. [Citado el: 20 de febrero de 2013.] <http://www.definicionabc.com/tecnologia/tcpip.php>
- DIGIA. 2012.** *Developer Tools - Digia Plc.* [En línea] 2012. [Citado el: 26 de noviembre de 2012.] <http://qt.digia.com/Product/Developer-Tools/>.
- Eclipse. 2012.** *OpenUP.* [En línea] 2012. [Citado el: 14 de noviembre de 2012.] <http://epf.eclipse.org/wikis/openup/>.
- Equiluz, Javier. 2011.** *Desarrollo Web Ágil con Symfony2.*
- Freeman, R. Edward. 1984.** *Strategic Management: A Stakeholder Approach.* s.l: Pitman, 1984. ISBN.
- Garófalo, Alain Abel. 2003.** *SICC: Sistema Integrado de Contabilidad y Configuración.* [ed.] CUJAE. 9, Cuba: s.n., 11 de julio de 2003, TELEM@TICA, Vol. II.
- Gutiérrez, Jorge. 2007.** *El Mundo Informático. Software Libre.* [En línea]. 2012 [Citado el: 11 de diciembre de 2012.] <http://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/>
- HighCharts, 2013.** *Highcharts JS.* [En línea]. 2013 [Citado el: 7 de abril de 2013.] <http://www.highcharts.com/products/highcharts/>
- Holmes, Arthur W. 1984.** *Principios Básicos de Auditoría.* México: C.E.C.S.A, 1984.
- ITI (Instituto Tecnológico de Informática), 2010.** *Testeo de estrés y carga.* [En línea]. 2013 [Citado el: 7 de abril de 2013.] <http://www.iti.es/servicios/servicio/resource/7240/index.html>.
- Jacobson, Ivar Booch; Rumbaugh, James. 2000.** *El proceso unificado de desarrollo de software.* Madrid, España: Pearson Educación S.A., 2000.
- JQuery, 2013.** *Página Oficial de JQuery.* [En línea]. 2013 [Citado el: 7 de abril de 2013.] <http://www.jquery.com>

## Referencias Bibliográficas.

- Larman, Craig. 1999.** *UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos.* ISBN 970-17-0261-1. México: Addison Wesley.
- Maldonado, Sergio. 2010.** *Analítica Web: medir para triunfar.* Madrid: ESIC, 2010. pág. 220.
- Mercer, Dave. 2002.** *Fundamentos de programación en XML.* Colombia: Quebecor World Bogota S.A., 2002. págs. 46-47.
- Microsoft, 2013.** *Revisiones de código y estándares de codificación.* [En línea] 2013. [Citado el: 1 de abril de 2013.] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>
- Ministerio de la Informática y las Comunicaciones (MIC). 2012.** *RESOLUCIÓN NO. 127/2007 ARTÍCULO 58: Reglamento de Seguridad Informática,* La Habana, Cuba, 2012
- MongoDB. 2012.** *The MongoDB Manual — MongoDB Manual.* [En línea] 2012. [Citado el: 24 de noviembre de 2012.] <http://docs.mongodb.org/manual/>.
- Object Management Group. 2012.** *Object Management Group - UML.* [En línea] 2012. [Citado el: 23 de noviembre de 2012.] <http://www.uml.org/>.
- Oracle. 2012.** *¿Qué es un servidor proxy? ¿Cómo puedo conseguir información sobre el servidor proxy?* [En línea] 2012. [Citado el: 24 de noviembre de 2012.] [http://www.java.com/es/download/help/proxy\\_server.xml](http://www.java.com/es/download/help/proxy_server.xml).
- . 2012.** *NetBeans IDE 7.2.1 Release Information.* [En línea] 2012. [Citado el: 25 de noviembre de 2012.] <http://netbeans.org/community/releases/72/>.
- Oxford University Press. 2002.** *Oxford Advanced Learner's Dictionary of current English.*
- Pérez, Yenly. 2007.** *Análisis y diseño de un editor de sonido.* 102p.
- PHP Group. 2013.** *PHP.* [En línea] 2013. [Citado el: 1 de abril de 2013.] <http://php.net>
- . 2013.** *PHP.* [En línea] 2013. [Citado el: 1 de abril de 2013.] <http://php.net/manual/es/book ldap.php>
- Pressman, Roger. 2006.** *Ingeniería de Software: Un enfoque práctico.*
- SafeSquid. 2012.** *Content Filtering Internet Proxy - SafeSquid.* [En línea] 2012. [Citado el: 14 de noviembre de 2012.] <http://www.safesquid.com/>.
- Sawmill Analytics. 2012.** *Sawmill Professional.* [En línea] 2012. [Citado el: 25 de noviembre de 2012.]

## Referencias Bibliográficas.

<http://www.sawmill.co.uk/professional.php>.

**Sawmill. 2012.** *Office Efficiencies SafeSquid (Extended Logging) Log Analyzer*. [En línea] 2012. [Citado el: 21 de noviembre de 2012.] [http://www.sawmill.net/formats/safesquid\\_combined.html](http://www.sawmill.net/formats/safesquid_combined.html).

**Segurmática. 2012.** *Analizador De Acceso a Internet*. [En línea] 2012. [Citado el: 25 de noviembre de 2012.] <http://www.segurmatica.cu/laboratorio/lab4.jsp>.

**SER. 2012.** *Glosario | SER - SERVICIO, ETICA Y RESPONSABILIDAD*. [En línea] 2012. [Citado el: 14 de noviembre de 2012.] <http://www.serconsultorias.com/2010/12/glosario/>.

**SOUNDEBIAN. 2009.** *Glosario @ SounDebian*. [En línea] 2009. [Citado el: 20 de octubre de 2012.] <http://www.soundebian.com.ar/glosario/#script>.

**Stewart, Allen. 2001.** *Importance of Understanding Logs from an Information Security Standpoint*. Estados Unidos: SANS Institute, 2001.

**Terry, George Robert; Franklin, Stephen G. 1988.** *Principios de administración*. México: C.E.C.S.A., 1988.

**Toll, Yuniet; Mendoza, Yilennis. 2007.** *Propuesta de manual de procedimiento de Pruebas de Sistema y su aplicación en el Proyecto CICPC*. [En línea]. 2013 [Citado el: 7 de abril de 2013.] [http://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/TD\\_0717\\_07/1/TD\\_0717\\_07.pdf](http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_0717_07/1/TD_0717_07.pdf).

**Twitter, Bootstrap. 2013.** *Bootstrap Twitter*. [En línea] 2013. [Citado el: 7 de abril de 2013.] <https://dev.twitter.com/blog/bootstrap-twitter>

**Universidad de las Ciencias Informáticas (UCI). 2012.** *Código de Ética: Compromisos para el uso de las Tecnologías de la Información en la Universidad de las Ciencias Informáticas*, La Habana, Cuba, 2012.

**Villardefrancos, María del Carmen. 2005.** *Propuesta de un modelo integral para auditar organizaciones de información en Cuba*. La Habana, Cuba: Universidad de La Habana.

**Visconti, Marcello; Astudillo, Hernán. 2011.** *Fundamentos de Ingeniería de Software*. Universidad Técnica Federico Santa María

**Visual Paradigm. 2012.** *UML CASE tool for software development*. [En línea] 2012. [Citado el: 23 de noviembre de 2012.] <http://www.visual-paradigm.com/product/vpuml/>.

**W3School. 2012.** *CSS Introduction*. [En línea] 2012. [Citado el: 24 de octubre de 2012.]

## Referencias Bibliográficas.

[http://www.w3schools.com/css/css\\_intro.asp](http://www.w3schools.com/css/css_intro.asp).

—. 2012. *JavaScript Introduction*. [En línea] 2012. [Citado el: 15 de noviembre de 2012.]

[http://www.w3schools.com/js/js\\_intro.asp](http://www.w3schools.com/js/js_intro.asp).

—. 2012. *PHP Introduction*. [En línea] 2012. [Citado el: 12 de noviembre de 2012.]

[http://www.w3schools.com/php/php\\_intro.asp](http://www.w3schools.com/php/php_intro.asp).

**WebLog Expert. 2012** [En línea] 2012. [Citado el: 15 de Diciembre de 2012.] <http://www.weblogexpert.com/>

**Webstudio. 2012.** *Kerio Winroute Firewall*. [En línea] Programática de Costa Rica S.A., 2012. [Citado el: 12 de noviembre de 2012.] <http://www.programatica.net/es-es/productos/keriowinroutefirewall.aspx>

## Bibliografía.

**Acevedo, Sergio.** *Desarrollo de capas de abstracción para mejorar la eficiencia en la construcción de aplicaciones web.* Enero 2008.

**Álvarez, Miguel Ángel. 2009.** *Qué es HTML 5.* [En línea] 2012. [Citado el: 22 de Octubre de 2012.] <http://www.desarrollo.com/articulos/que-es-html5.html>.

—. **2001.** *Veamos qué es JavaScript y las posibilidades que nos ofrece utilizar este lenguaje a la hora de desarrollar páginas web.* [En línea]. 2013 [Citado el: 7 de abril de 2013.] <http://www.desarrolloweb.com/articulos/490.php>

—. **2004** *Introducción al manual del lenguaje PHP en su versión 5.* [En línea]. 2013 [Citado el: 7 de abril de 2013.] <http://www.desarrolloweb.com/articulos/1696.php>

**AWStats. 2012.** *Free log file analyzer for advanced statistics.* [En línea] 2012. [Citado el: 23 de noviembre de 2012.] <http://awstats.sourceforge.net/>.

**Barrera, Otniel. 2006.** *ISAWEB. Monitoreo de Tráfico en Internet.* La Habana: MIC, 2006.

—; **García Pérez, Deiny, 2007.** *Monitoreando el tráfico de Internet.* Una solución al alcance de la mano. [En línea] 2012. [Citado el: 23 de noviembre de 2012.] [http://www.informaticahabana.com/evento\\_virtual/files/seg034.doc](http://www.informaticahabana.com/evento_virtual/files/seg034.doc)

**Barrett, B, 2012.** *Home of the Webalizer.* [En línea] 2012. [Citado el: 23 de noviembre de 2012.] <http://www.webalizer.org/>.

**Barrio, Francisco José Sáez, 2010.** *La calidad de las aplicaciones.* 2010.

**Belgrano, Universidad de Argentina.** *Metodología de Auditoría Informática.* [En línea] [Citado el: 18 de noviembre de 2012.] <http://www.ub.edu.ar/catedras/ingenieria/auditoria/metodologia/metodologia.htm>.

**Bradenbaugh, Jerry, 2000.** *Aplicaciones JavaScript.* [En línea] 2012. [Citado el: 23 de noviembre de 2012.] <http://biblioteca.uci.cu/titdigitales.htm#pro>

**Brick, Marketing. 2012.** *What is a Log File? - Define a Log File.* [En línea] 2012. [Citado el: 10 de noviembre de 2012.] <http://www.brickmarketing.com/define-log-file.htm>.

**Burbeck, Steve. 1992.** *Application programming in Smalltalk-80: How to use Model-View-Controller (MVC).*

**Carbonell, Alberto Carlos; Morciego, Carlos Manuel. 2011.** *Sistema Distribuido Cliente-Servidor.* Universidad de las Ciencias Informáticas. Ciudad de La Habana.

**Carrillo, Luis Vinicio León, 2005.** *El Contexto de la Prueba de Software. Probar para incrementar la calidad.*

**Chávez, Miguel Ángel; Hernández, Jorge Luis, 2010.** *QTOCTAVIZ: Módulo para la visualización científica integrado al sistema matemático Octaviz.* La Habana, Cuba: Universidad de las Ciencias Informáticas.

**De Los Angeles, Leodán. 2012.** *MARCOS DE TRABAJO PARA EL DESARROLLO DE APLICACIONES WEB.* [En línea] 2012. [Citado el: 7 de Abril de 2013.] [http://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/4139/1/UCIENCIA-2012-T47-P512-Ponencia-1772.pdf](http://repositorio_institucional.uci.cu/jspui/bitstream/ident/4139/1/UCIENCIA-2012-T47-P512-Ponencia-1772.pdf).

**Díaz, Moisés Daniel, 2003.** *Diseño de aplicaciones Internet usando los Patrones de diseño.*

**DIGIA. 2012.** *Developer Tools - Digia Plc.* [En línea] 2012. [Citado el: 26 de noviembre de 2012.] <http://qt.digia.com/Product/Developer-Tools/>.

**Eclipse. 2012.** *OpenUP.* [En línea] 2012. [Citado el: 14 de noviembre de 2012.] <http://epf.eclipse.org/wikis/openup/>.

**Equiluz, Javier. 2011.** *Desarrollo Web Ágil con Symfony2.*

**Freeman, R. Edward. 1984.** *Strategic Management: A Stakeholder Approach.* s.l: Pitman, 1984. ISBN.

**HighCharts. 2013.** *Highcharts JS.* [En línea]. 2013 [Citado el: 7 de abril de 2013.] <http://www.highcharts.com/products/highcharts>

**Garófalo, Alain Abel. 2003.** *SICC: Sistema Integrado de Contabilidad y Configuración.* [ed.] CUJAE. 9, Cuba: s.n., 11 de julio de 2003, TELEM@TICA, Vol. II.

**Gutiérrez, Jorge. 2007.** *El Mundo Informático. Software Libre.* [En línea]. 2012 [Citado el: 11 de diciembre de 2012.] <http://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/>

**Herrero Núñez, Julio Alberto.** *Minería de Textos Web. Recuperación y organización de la información.* [En línea] 2012. [Citado el: 8 de diciembre de 2012.] <http://mineria-textos-web.awardspace.com>

**Holmes, Arthur W. 1984.** *Principios Básicos de Auditoría.* México: C.E.C.S.A, 1984.

**IEEE, 1998.** *Guía para el desarrollo de Especificaciones de Requerimientos de Sistemas.*

**Jacobson, Ivar Booch; Rumbaugh, James. 2000.** *El proceso unificado de desarrollo de software.* Madrid, España: Pearson Educación S.A., 2000.

**JQuery, 2013.** *Página Oficial de JQuery.* [En línea]. 2013 [Citado el: 7 de abril de 2013.] <http://www.jquery.com>

**Lamancha, Beatriz Pérez. 2007.** *Gestión de las Pruebas Funcionales.* Montevideo: s.n., 2007.

**Larman, Craig. 1999.** *UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos.* ISBN 970-17-0261-1. México: Addison Wesley.

**Maldonado, Sergio. 2010.** *Analítica Web: medir para triunfar.* Madrid: ESIC, 2010. pág. 220.

**Mercer, Dave. 2002.** *Fundamentos de programación en XML.* Colombia: Quebecor World Bogota S.A., 2002. págs. 46-47.

**Ministerio de la Informática y las Comunicaciones (MIC). 2012.** *RESOLUCIÓN NO. 127/2007 ARTÍCULO 58: Reglamento de Seguridad Informática,* La Habana, Cuba, 2012

**MongoDB. 2012.** *The MongoDB Manual — MongoDB Manual.* [En línea] 2012. [Citado el: 24 de noviembre de 2012.] <http://docs.mongodb.org/manual/>.

—. **2012.** *Sitio Oficial de MongoDB.* [En línea] 2012. [Citado el: 24 de noviembre de 2012.] <http://www.mongodb.org>.

**Object Management Group. 2012.** *Object Management Group - UML.* [En línea] 2012. [Citado el: 23 de noviembre de 2012.] <http://www.uml.org/>.

**Oracle. 2012.** *¿Qué es un servidor proxy? ¿Cómo puedo conseguir información sobre el servidor proxy?* [En línea] 2012. [Citado el: 24 de noviembre de 2012.] [http://www.java.com/es/download/help/proxy\\_server.xml](http://www.java.com/es/download/help/proxy_server.xml).

—. **2012.** *NetBeans IDE 7.2.1 Release Information.* [En línea] 2012. [Citado el: 25 de noviembre de 2012.] <http://netbeans.org/community/releases/72/>.

**Ordoñez, Yoanni; Avilés, Ernesto, 2010.** *Herramienta informática de Minería de Uso de la Web sobre los registros de navegación por Internet.* Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2010.

**Oxford University Press. 2002.** *Oxford Advanced Learner's Dictionary of current English.*

**Pérez, Yenly. 2007.** *Análisis y diseño de un editor de sonido.* 102p.

**Pérez, Patricia.** *El mal uso de Internet preocupa a las empresas. Las provincias.* [En línea] 2012. [Citado el: 25 de noviembre de 2012.] <http://www.lasprovincias.es/valencia/pg060313/economia/200603/13/VAL-ECO-281.html>

**PHP Group. 2013.** *PHP.* [En línea] 2013. [Citado el: 1 de abril de 2013.] <http://php.net>

**Pressman, Roger. 2006.** *Ingeniería de Software: Un enfoque práctico.*

**SafeSquid. 2012.** *Content Filtering Internet Proxy - SafeSquid.* [En línea] 2012. [Citado el: 14 de noviembre de 2012.] <http://www.safesquid.com/>.

**Sawmill Analytics. 2012.** *Sawmill Professional.* [En línea] 2012. [Citado el: 25 de noviembre de 2012.] <http://www.sawmill.co.uk/professional.php>.

**Sawmill. 2012.** *Office Efficiencies SafeSquid (Extended Logging) Log Analyzer.* [En línea] 2012. [Citado el: 21 de noviembre de 2012.] [http://www.sawmill.net/formats/safesquid\\_combined.html](http://www.sawmill.net/formats/safesquid_combined.html).

**Segurmática. 2012.** *Analizador De Acceso a Internet.* [En línea] 2012. [Citado el: 25 de noviembre de 2012.] <http://www.segurmatica.cu/laboratorio/lab4.jsp>.

**SER. 2012.** *Glosario | SER - SERVICIO, ETICA Y RESPONSABILIDAD.* [En línea] 2012. [Citado el: 14 de noviembre de 2012.] <http://www.serconsultorias.com/2010/12/glosario/>.

**SOUNDEBIAN. 2009.** *Glosario @ SounDebian.* [En línea] 2012. [Citado el: 20 de octubre de 2012.] <http://www.soundebian.com.ar/glosario/#script>.

**Stewart, Allen. 2001.** *Importance of Understanding Logs from an Information Security Standpoint.* Estados Unidos: SANS Institute, 2001.

**Suárez, José Alberto.** *Análisis de ficheros log en GNU/Linux.* [En línea] 2012. [Citado el: 20 de octubre de 2012.] [http://www.iberprensa.com/todolinux/articulos/TL65\\_42-46%20Taller\\_Log.pdf](http://www.iberprensa.com/todolinux/articulos/TL65_42-46%20Taller_Log.pdf)

**Terry, George Robert; Franklin, Stephen G. 1988.** *Principios de administración.* México: C.E.C.S.A., 1988.

**Twitter, Bootstrap. 2013.** *Bootstrap Twitter.* [En línea] 2013. [Citado el: 7 de abril de 2013.] <https://dev.twitter.com/blog/bootstrap-twitter>

**Universidad de las Ciencias Informáticas (UCI). 2012.** *Código de Ética: Compromisos para el uso de las Tecnologías de la Información en la Universidad de las Ciencias Informáticas*, La Habana, Cuba, 2012.

**Villardefrancos, María del Carmen. 2005.** *Propuesta de un modelo integral para auditar organizaciones de información en Cuba*. La Habana, Cuba: Universidad de La Habana.

**Visconti, Marcello; Astudillo, Hernán. 2011.** *Fundamentos de Ingeniería de Software*. Universidad Técnica Federico Santa María

**Visual Paradigm. 2012.** *UML CASE tool for software development*. [En línea] 2012. [Citado el: 23 de noviembre de 2012.] <http://www.visual-paradigm.com/product/vpuml/>.

**W3School. 2012.** *CSS Introduction*. [En línea] 2012. [Citado el: 24 de octubre de 2012.] [http://www.w3schools.com/css/css\\_intro.asp](http://www.w3schools.com/css/css_intro.asp).

—. **2012.** *JavaScript Introduction*. [En línea] 2012. [Citado el: 15 de noviembre de 2012.] [http://www.w3schools.com/js/js\\_intro.asp](http://www.w3schools.com/js/js_intro.asp).

—. **2012.** *PHP Introduction*. [En línea] 2012. [Citado el: 12 de noviembre de 2012.] [http://www.w3schools.com/php/php\\_intro.asp](http://www.w3schools.com/php/php_intro.asp).

**WebLog Expert. 2012** [En línea] 2012. [Citado el: 15 de Diciembre de 2012.] <http://www.weblogexpert.com/>

**Webstudio. 2012.** *Kerio Winroute Firewall*. [En línea] Programática de Costa Rica S.A., 2012. [Citado el: 12 de noviembre de 2012.] <http://www.programatica.net/es-es/productos/keriowinroutefirewall.aspx>

**Zaninotto, Francois; Potencier, Fabien, 2007.** *Symfony la guía definitiva* [Citado el: 12 de noviembre de 2012.] <http://www.librosweb.es/>

## Glosario de Términos.

### A

API Application Programming Interface - Interfaz de Programación de Aplicaciones.

### C

CSS Hojas de estilo en cascada.

CUJAE Ciudad Universitaria José Antonio Echeverría.

### D

DOM Document Object Model - Modelo de Objetos del Documento.

### F

FTP File Transfer Protocol - Protocolo de Transferencia de Archivos.

### G

GPL General Public License- Licencia Pública General creada por la Free Software Foundation.

GRASP Patrones Generales de Software para la Asignación de Responsabilidades.

GZ Zero Injection Pressure-Cero Inyección a Presión. Formato de compresión sin pérdida para sistemas operativos basados en GNU/Linux.

### H

HTTP Hypertext Transfer Protocol - Protocolo de Transferencia de Hipertexto.

### I

ICID Instituto Central de Investigación Digital.

IDE Entornos de Desarrollo Integrado.

### L

Logs Registro oficial de eventos durante un periodo de tiempo.

### M

MCUS Modelo de Casos de Uso del Sistema.

MGRE Módulo de Gestión de Reportes Estadísticos.

MIC Ministerio de la Informática y las Comunicaciones.

MIME Multipurpose *Internet* Mail Extensions - Extensiones Multipropósito de Correo de *Internet*.

MIT Licencia creada por Massachusetts Institute of Technology.

MVC Modelo-Vista-Controlador.

### P

PHP	Hypertext Preprocessor-Pre procesador de Hipertexto. Lenguaje de programación interpretado del lado del servidor.
Proxy	Ordenador que intercepta las conexiones de red que un cliente hace a un servidor de destino.

### R

RUP	Rational Unified Process- Proceso Unificado Racional.
-----	---

### S

Script	Guión o conjunto de instrucciones.
SGBD	Sistemas Gestores de Base de Datos.
SICC	Sistema Integrado de Contabilidad y Configuración.
SINI	Departamento de Soluciones Informáticas para <i>Internet</i> perteneciente al Centro de Ideoinformática – UCI.
SRNI	Sistema de Reportes de la Navegación por <i>Internet</i> .

### T

TCP/IP	Transmisión/Protocolo de <i>Internet</i> , conjunto o familia de protocolos de red sobre los cuales se basa <i>Internet</i> .
TIC	Tecnologías de la Información y las Comunicaciones.

### U

UCI	Universidad de las Ciencias Informáticas.
UML	Lenguaje Unificado de Modelado.
URL	Uniform Resource Locator - Localizador Uniforme de Recursos.

### W

W3C	World Wide Web Consortium.
-----	----------------------------

### X

XML	Extensible Markup Language o Lenguaje Extensible de Marcas.
-----	---

### Z

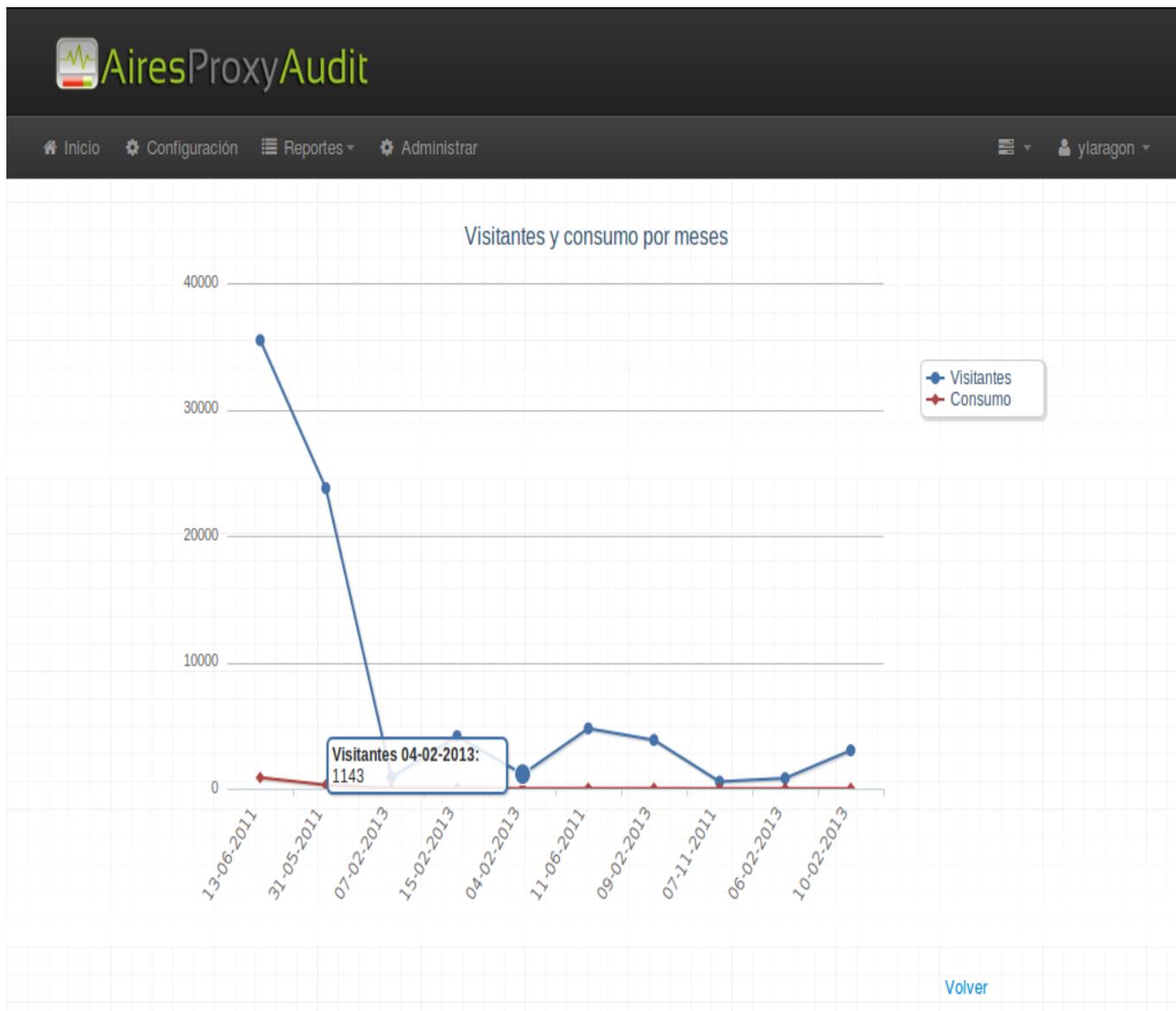
ZIP	Zero Injection Pressure-Cero Inyección a Presión. Formato de compresión sin pérdida.
-----	--

## Anexos.

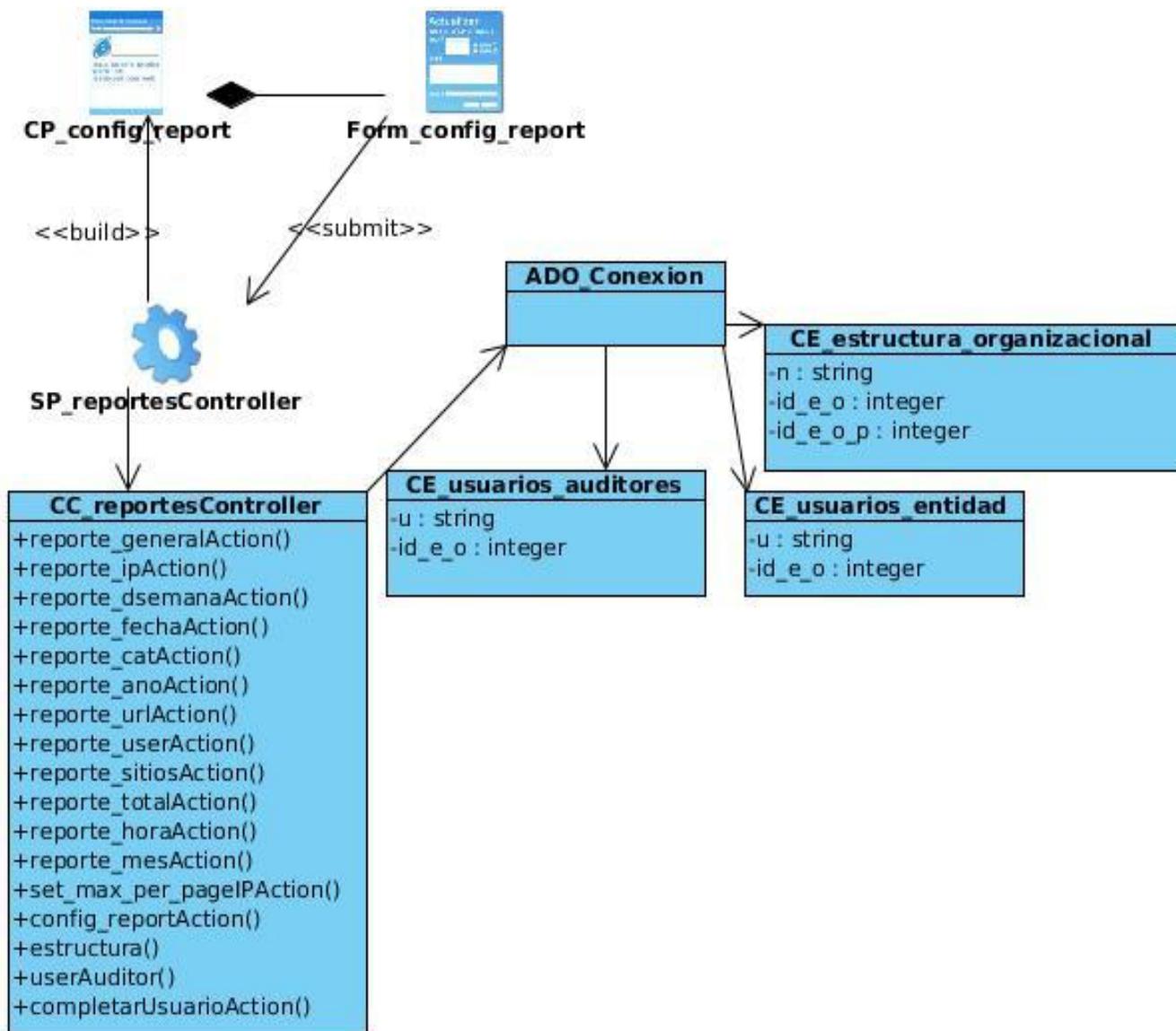
### Anexo 1. Reporte gráfico de URLs más visitadas.



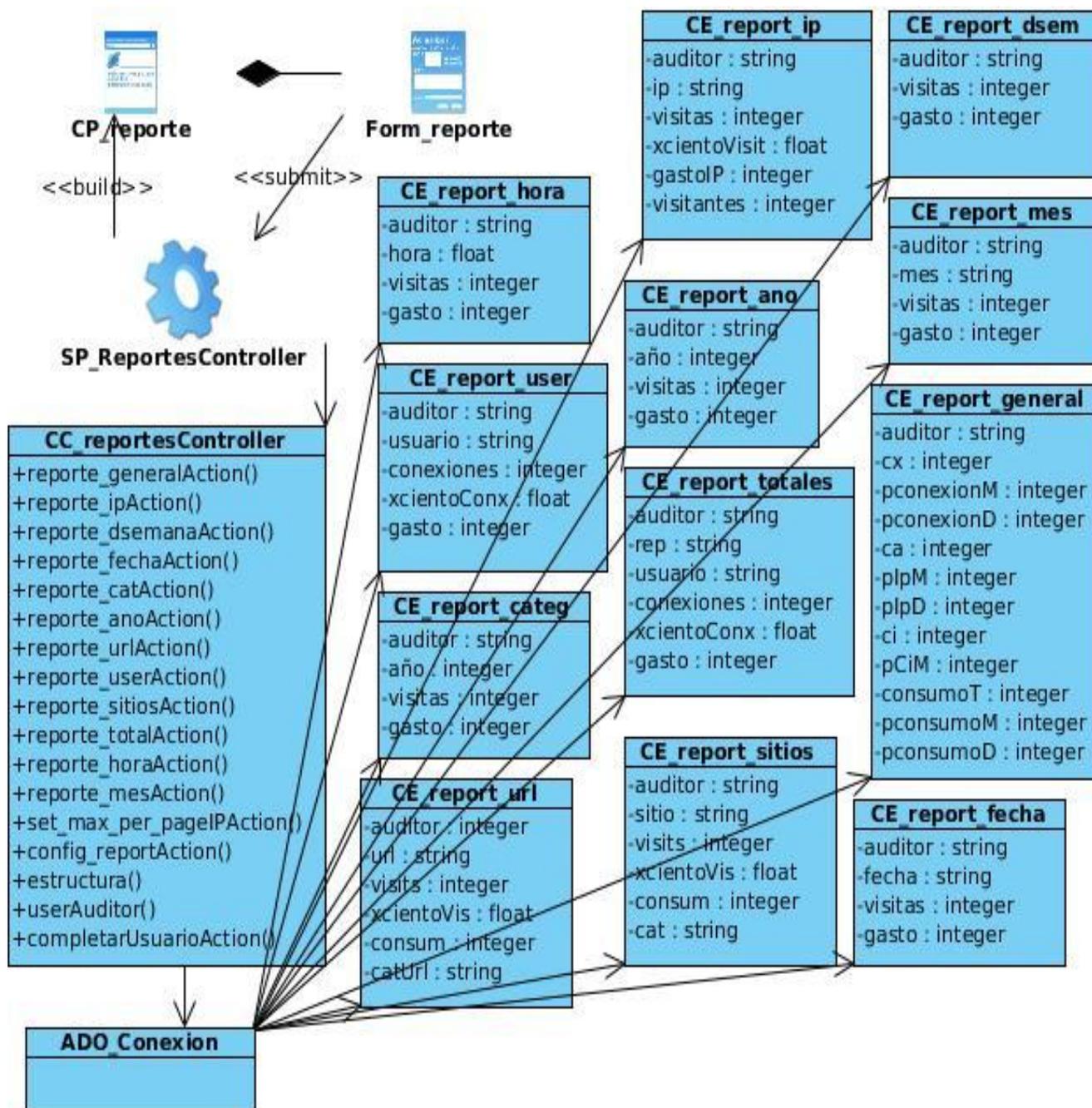
### Anexo 2. Reporte gráfico de visitantes y consumo por meses.



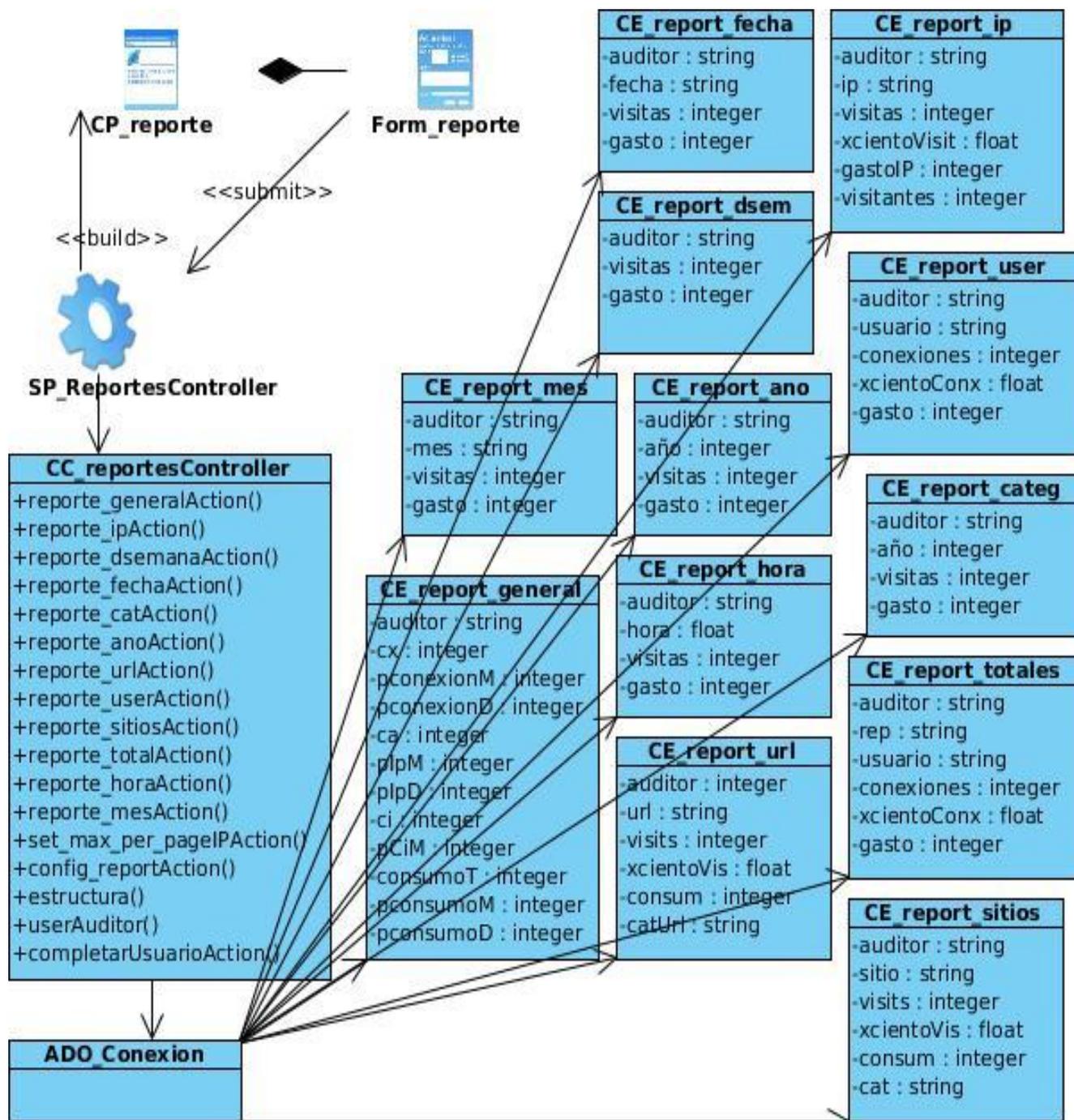
Anexo 3. Diagrama de clases del diseño del caso de uso Exportar configuración.



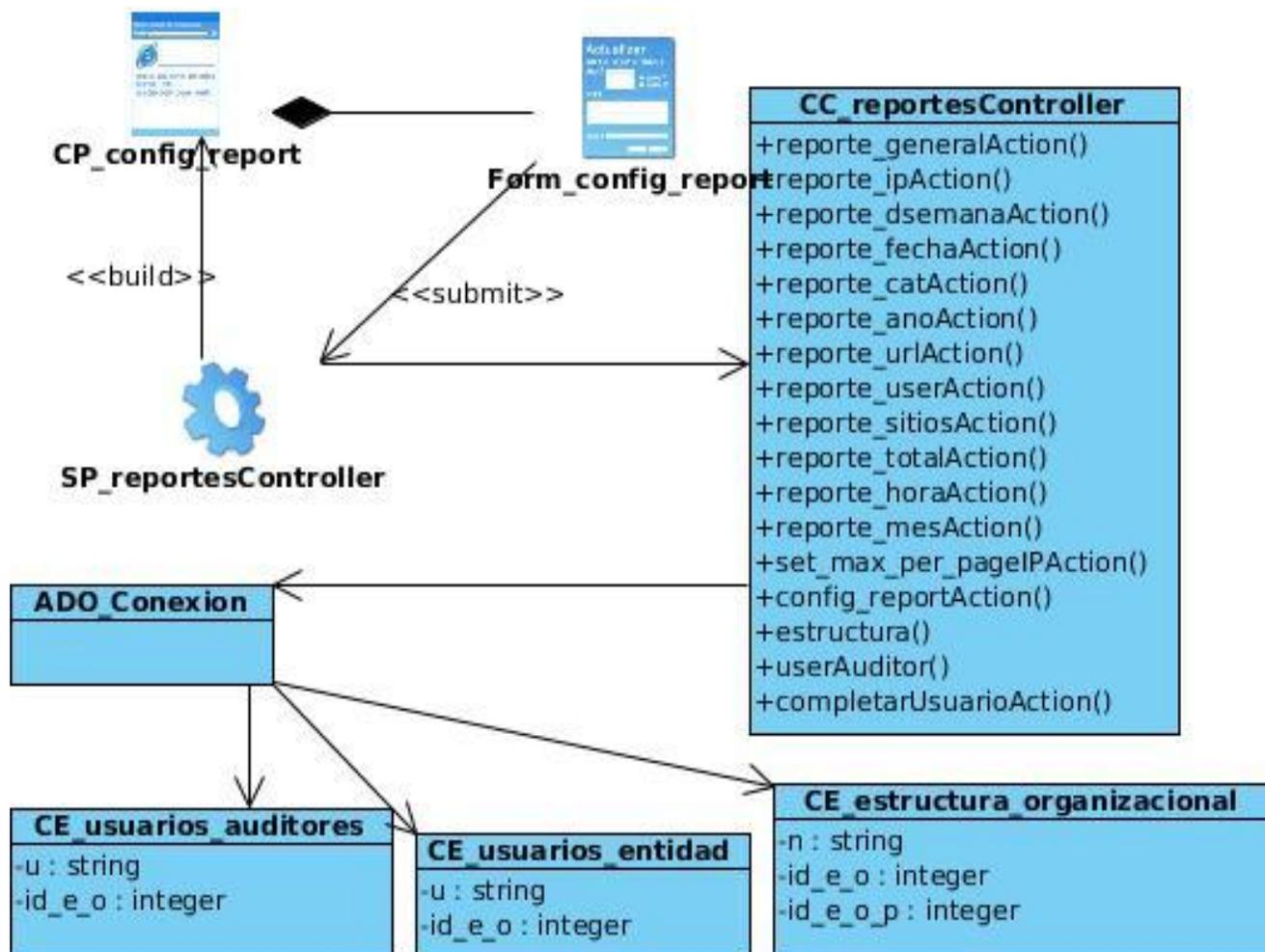
Anexo 4. Diagrama de clases del diseño del caso de uso Exportar reporte a pdf.



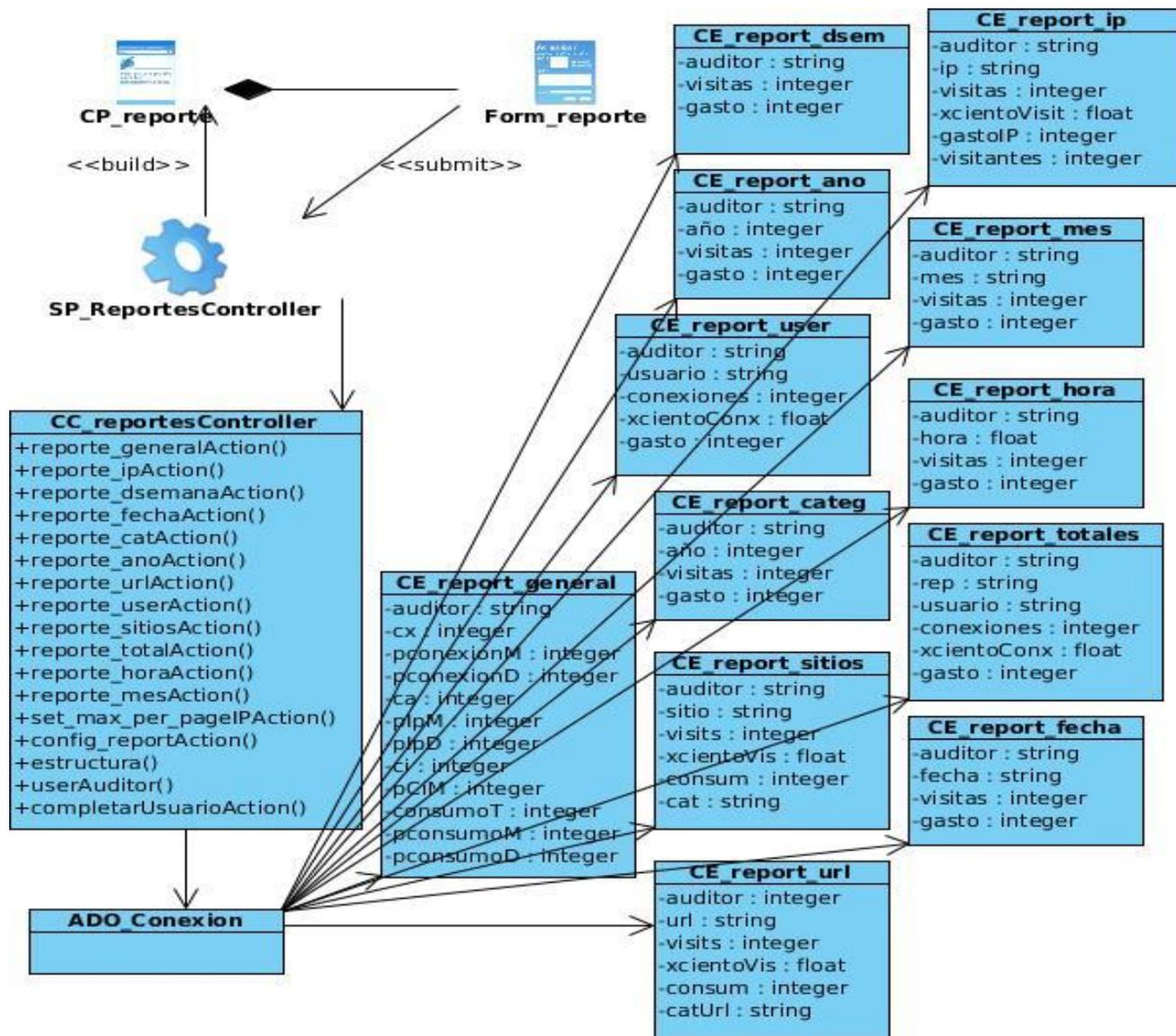
Anexo 5. Diagrama de clases del diseño del caso de uso Gestionar reportes guardados.



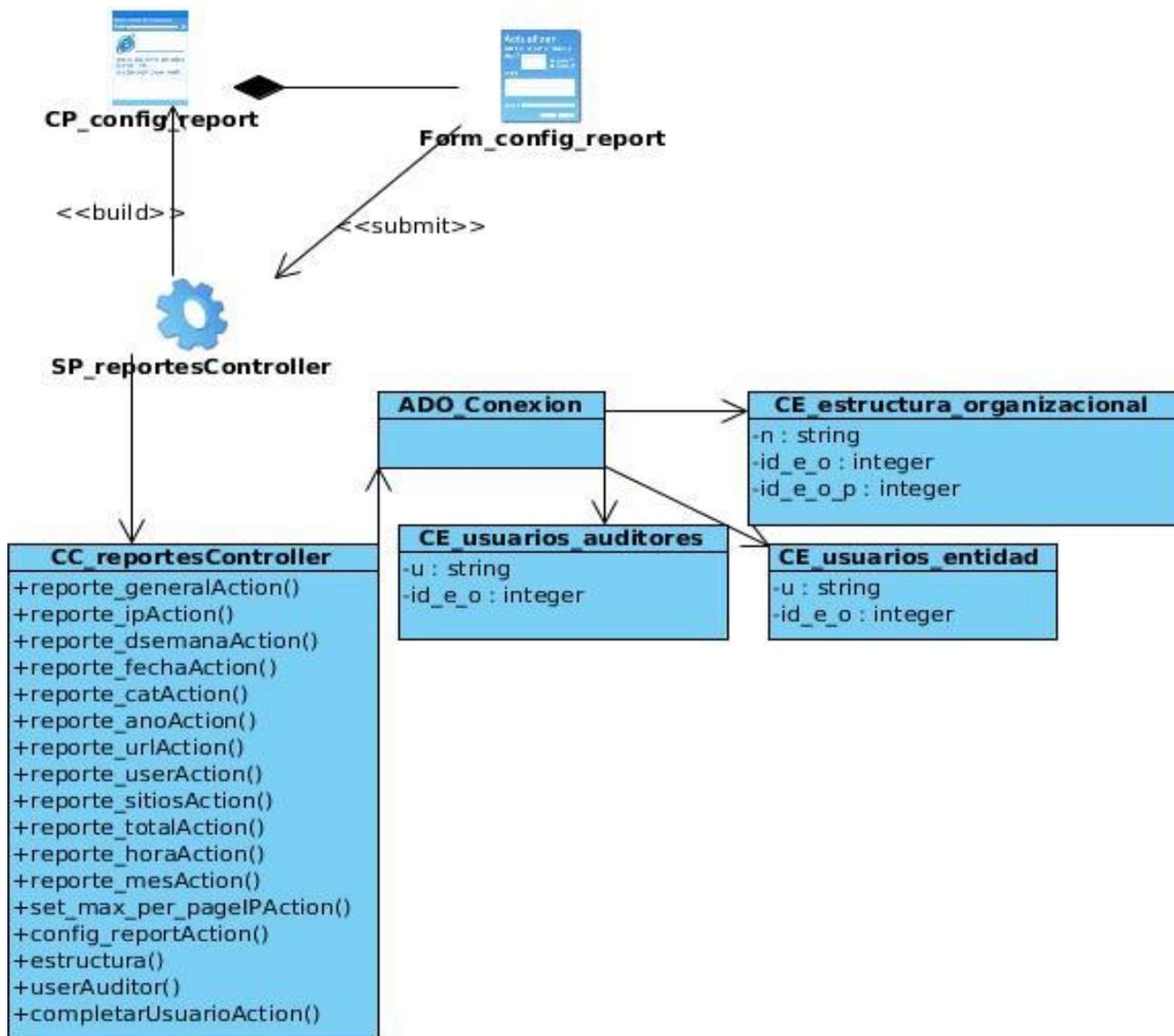
Anexo 6. Diagrama de clases del diseño del caso de uso Guardar configuración.



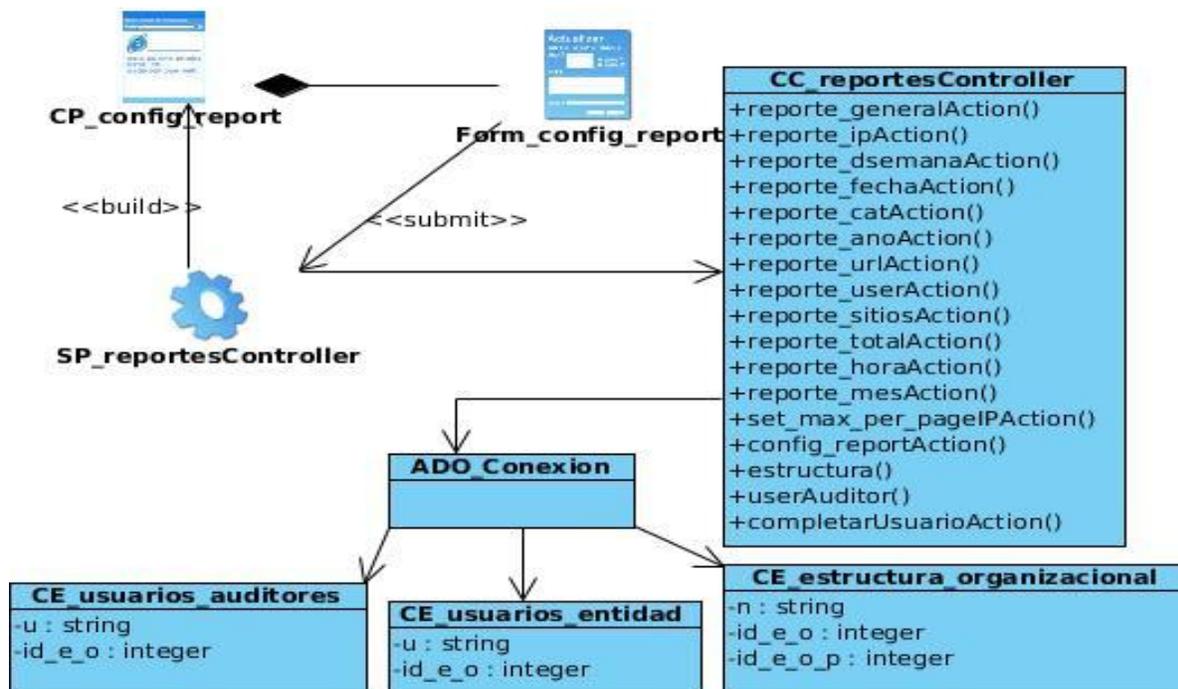
Anexo 7. Diagrama de clases del diseño del caso de uso Guardar reporte.



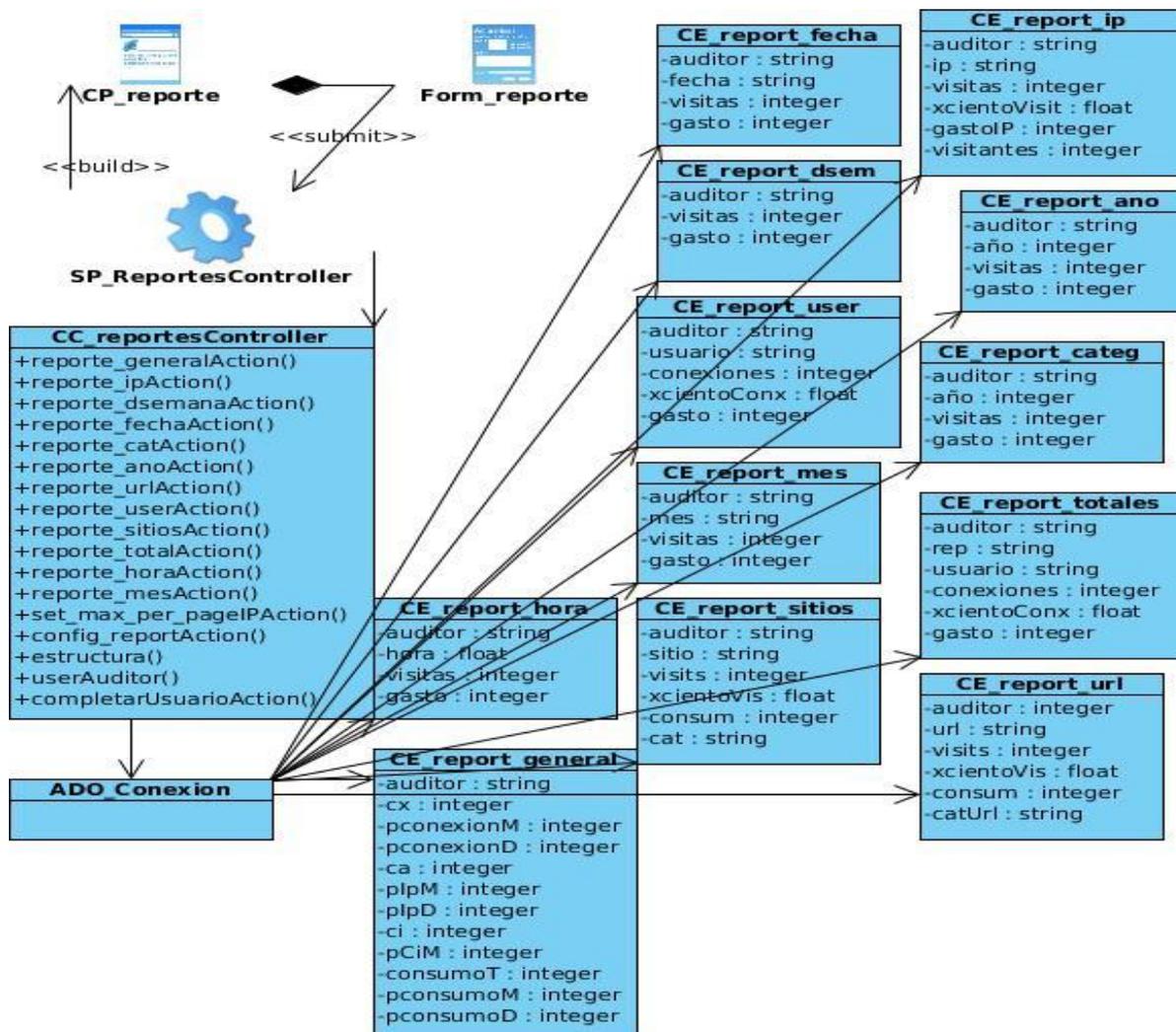
Anexo 8. Diagrama de clases del diseño del caso de uso Importar configuración.



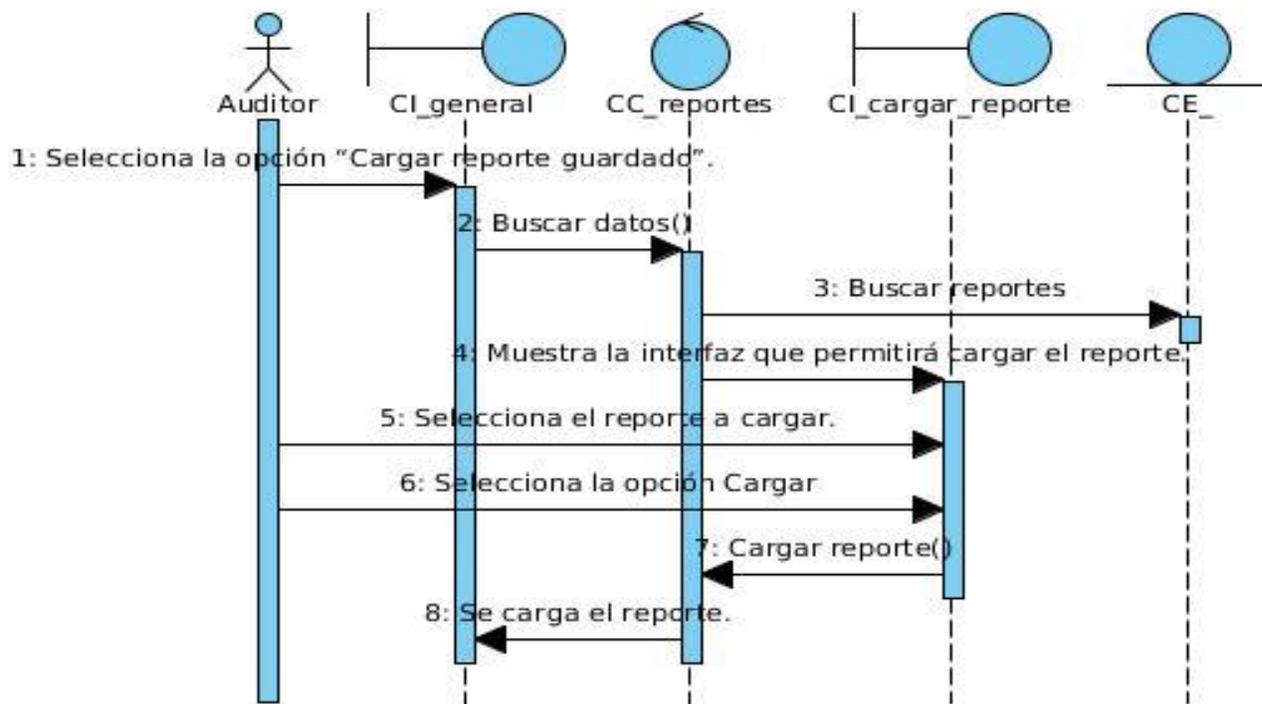
Anexo 9. Diagrama de clases del diseño del caso de uso Mostrar entidades.



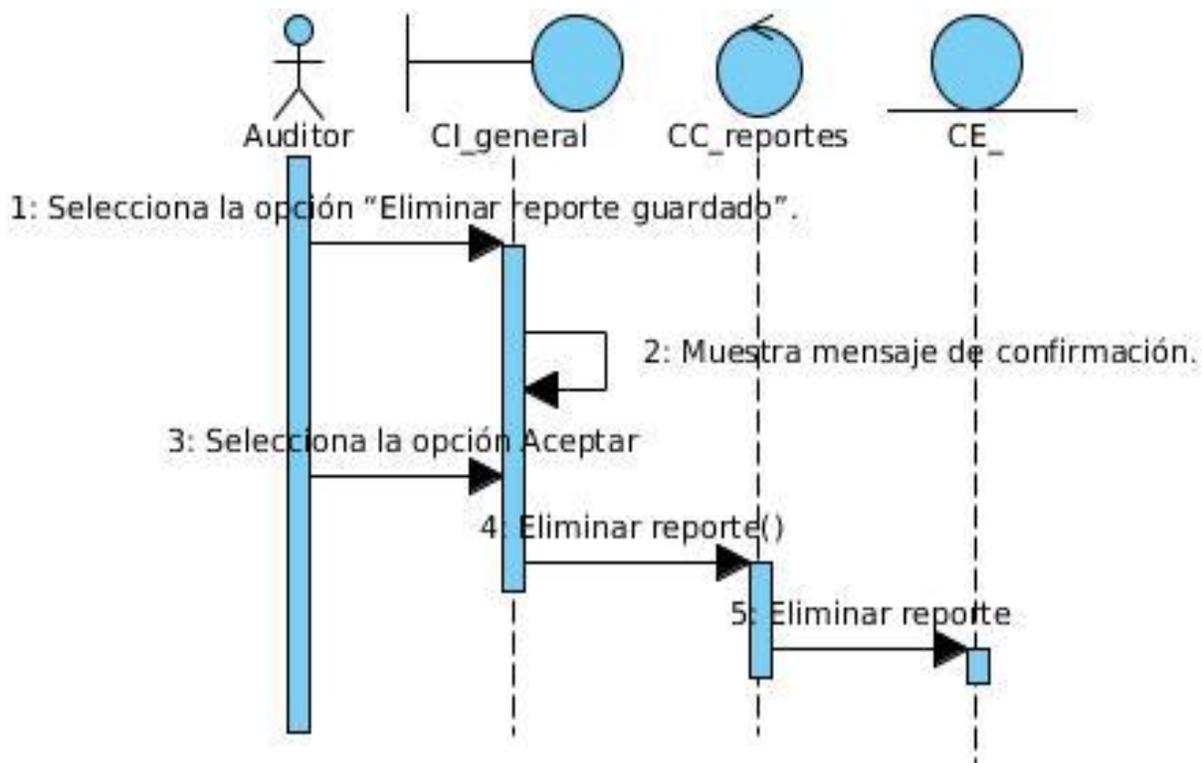
Anexo 10. Diagrama de clases del diseño del caso de uso Ordenar resultado.



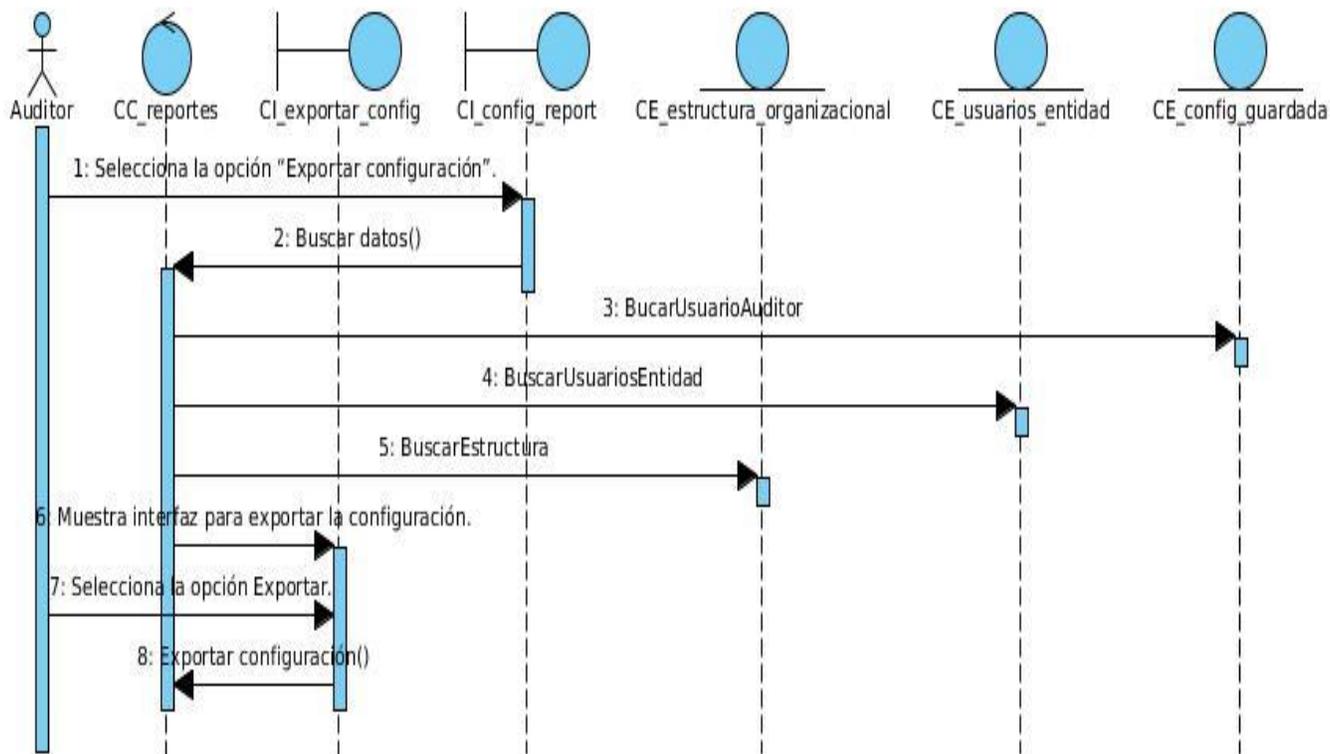
**Anexo 11. Diagrama de secuencia del caso de uso Cargar reporte guardado.**



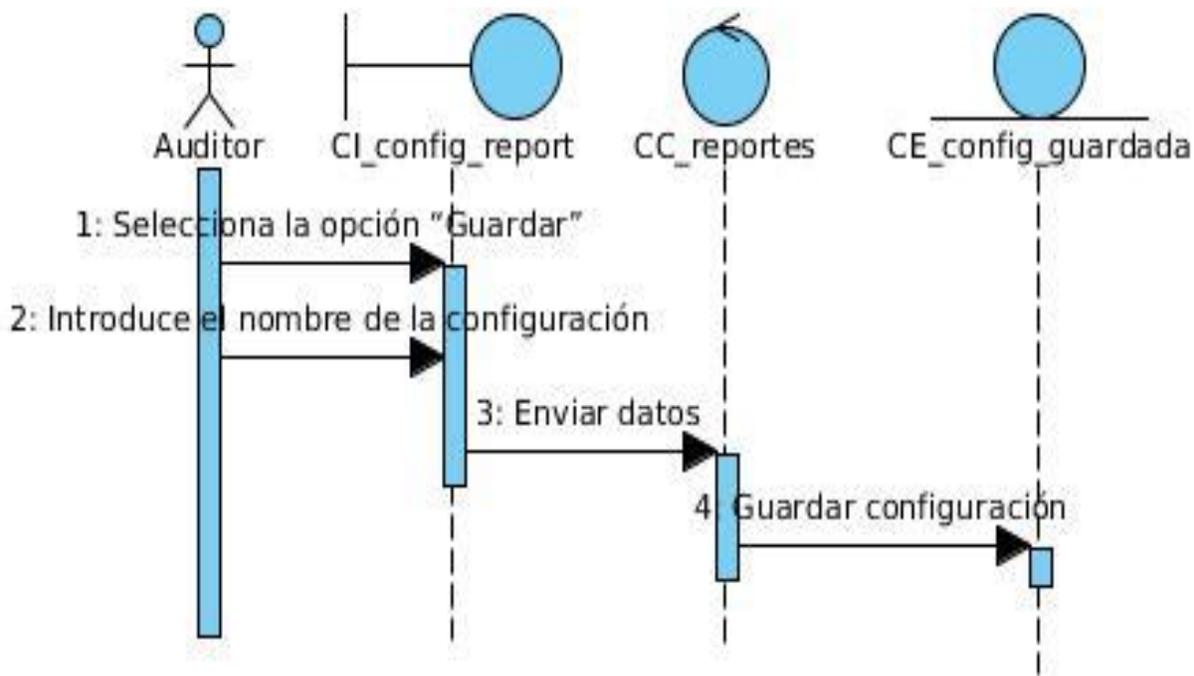
**Anexo 12. Diagrama de secuencia del caso de uso Eliminar reporte guardado.**



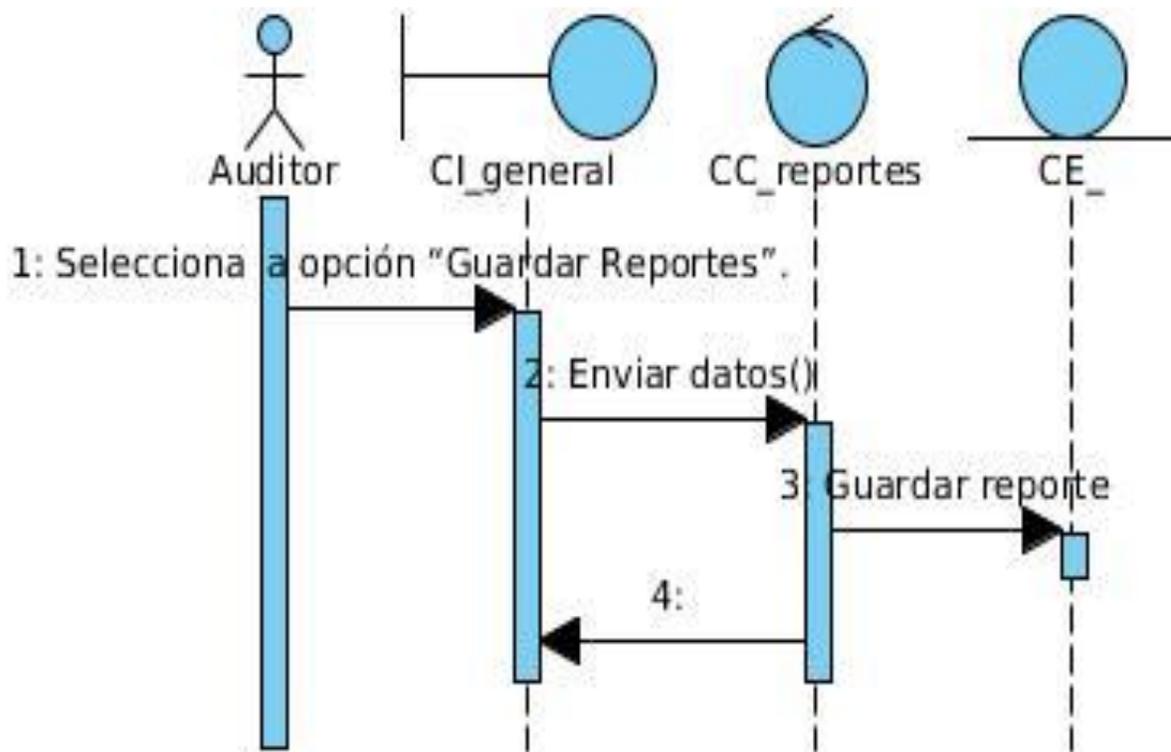
**Anexo 13. Diagrama de secuencia del caso de uso Exportar configuración.**



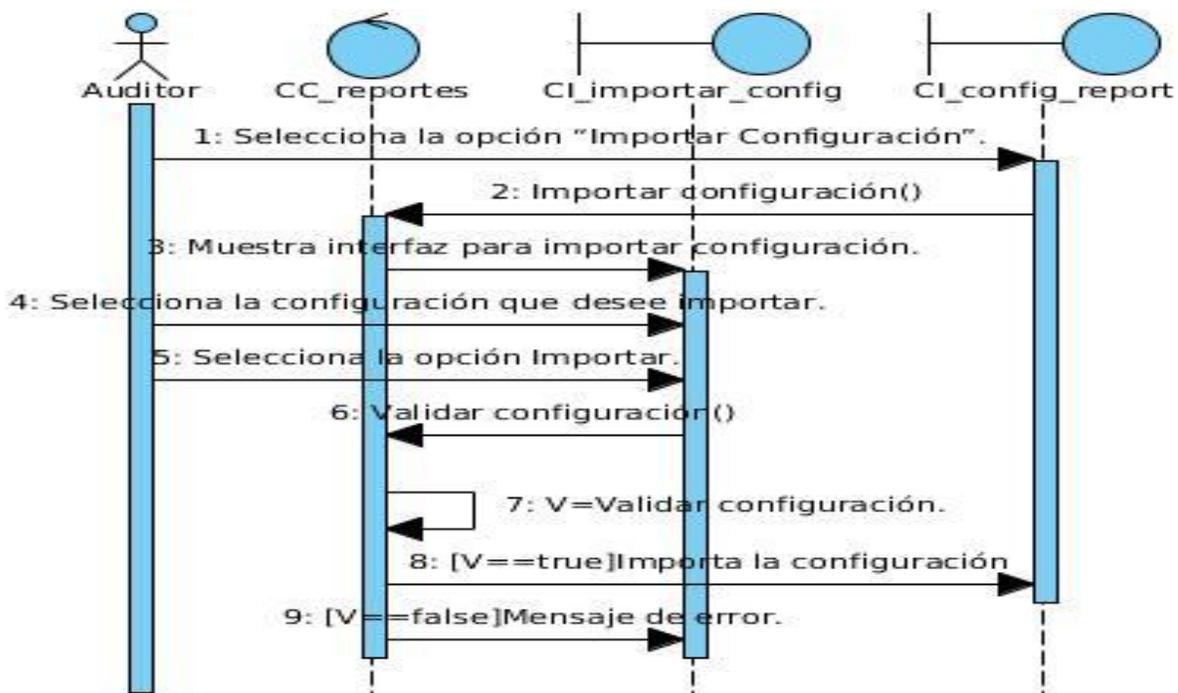
**Anexo 14. Diagrama de secuencia del caso de uso Guardar configuración.**



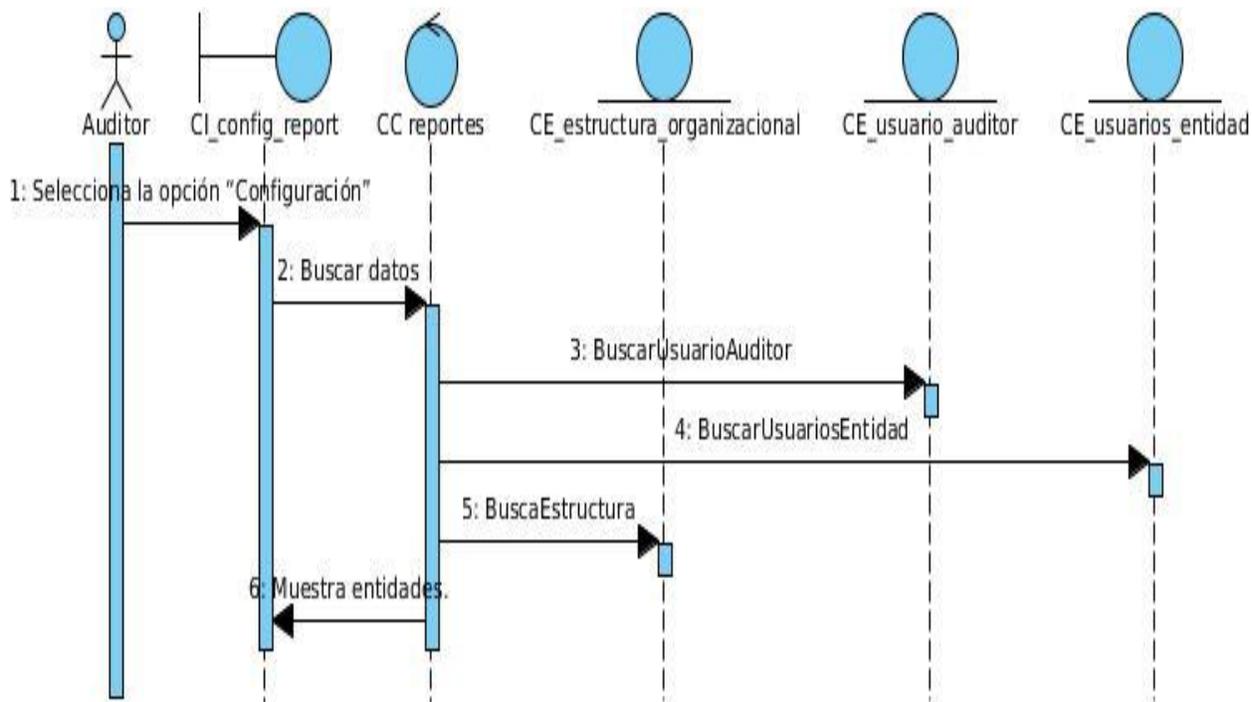
Anexo 15. Diagrama de secuencia del caso de uso Guardar reporte.



Anexo 16. Diagrama de secuencia del caso de uso Importar configuración.



**Anexo 17. Diagrama de secuencia del caso de uso Mostrar entidades.**



**Anexo 18. Pruebas de Gestión de Sesiones.**

Pruebas de Gestión de Sesiones					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Al copiar la URL de la aplicación después de estar autenticado, cerrar el navegador y volver a abrirlo para pegar la dirección copiada anteriormente, la aplicación permite que el usuario entre a la aplicación.	0			
Crítico	Al cerrar la sesión de un usuario y dar clic en el botón del navegador "Atrás" la aplicación vuelve entrar a la sesión		NP		

autenticada.				
--------------	--	--	--	--

### Anexo 19. Comprobación del Sistema de Autenticación.

Comprobación del Sistema de Autenticación					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Se bloquea la sesión del usuario después de un tiempo X (establecido por la aplicación) estando sin usar la misma.	0			
Crítico	Se bloquea la cuenta del usuario después de un número X (establecidos por la aplicación) de intento de login fallidos por el usuario. De ser así definir la cantidad de intentos en la columna Comentarios.	0			
Crítico	Se muestran diferentes mensajes de error al usuario "intentar autenticarse con un usuario correcto y una contraseña incorrecta" y "al autenticarse con un usuario incorrecto y la misma contraseña del intento anterior".	0			
Crítico	El mensaje devuelto al intentar autenticarse con un usuario correcto y contraseña incorrecta, es el mismo al intento con usuario y contraseña incorrecta, si es el mismo intento verificar el tiempo de respuesta,	0			

	pues podría apreciarse una media ligeramente diferente en el tiempo de respuesta, que igualmente puede usarse esta información para descubrir cuentas válidas.				
Crítico	El campo usuario de la autenticación al sistema tiene el auto completamiento activado (guarda los usuarios que se autentican). Para ver esto se debe autenticar al sistema, luego salir y poner la primera letra del usuario autenticado para ver si muestra el nombre de usuario completo.	0			
Crítico	El sistema protege el envío de los datos mediante protocolo seguro (https).	0			
Crítico	El sistema usa algún certificado.	0			

### Anexo 20. Comprobación del Sistema de Autenticación.

Validación de Datos					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Se enmascaran datos confiables cuando se visualicen en la aplicación (Por ejemplo: Contraseñas)	0			
Crítico	La aplicación solamente permite contraseñas alfanuméricas, que incluyan caracteres especiales y que tengan seis caracteres mínimos de longitud.	0			
Crítico	La aplicación permitir la funcionalidad de cambio de contraseña únicamente a usuarios autenticados validando la antigua contraseña, la nueva contraseña y la respuesta a la pregunta de seguridad (opcional en dependencia del proyecto).		NP		
Crítico	El sistema muestra algún mensaje indebido, al colocar en la barra de dirección o en campos de entrada los caracteres: comillas simples ('), signos de ampersand (&), signos: + - /.	0			