

Universidad de las Ciencias Informáticas

Facultad 1



***Módulo estadístico para el Sistema de Gestión Documental de
Imágenes Digitales Ecumene Pyxel v1.0***

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

Autor:

Pavel Javier Suárez Abreu

Tutora:

Ing. Isandra Cruz Diaz.

Ciudad de La Habana, Junio 2013

“Año 55 de la Revolución”

DECLARACIÓN DE AUTORÍA

Por este medio, declaro que soy el autor oficial del presente Trabajo de Diploma, y otorgo a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales del mismo, en virtud de que pueda ser utilizado según los intereses y necesidades de la institución.

Para que así conste, firmo la presente a los _____ días del mes de _____ de 20_____

Pavel Javier Suárez Abreu

Ing. Isandra Cruz Diaz

Firma del autor

Firma del tutor

DATOS DE CONTACTO

Síntesis de la tutora: Ing. Isandra Cruz Diaz

Graduada en el 2010 de ingeniera en ciencias informáticas en la Universidad de las Ciencias Informáticas (UCI). Del 2010 al 2012 trabajó como Jefa de Proyecto en el Centro de Desarrollo del Facultad Regional de Artemisa, perteneciente a la UCI. Actualmente labora como especialista general en el Centro de Ideoinformática (CIDI) de dicha universidad. Ha obtenido experiencia en Gestión de Proyectos, Calidad de Software, y en el Diseño y Programación web.

Correo electrónico: icruz@uci.cu

AGRADECIMIENTOS

A mi abuelos Amparo, Cándida y Juan por estar siempre pendientes de mí, por ser mi faro guía.

A mi mamá por su cariño y apoyo incondicional.

A mi papá por demostrarme que siempre que se quiere se pueden lograr las grandes cosas que nos proponemos en la vida.

A mi familia en general por convertir el problema de uno en el de todos, por no parpadear a la hora de sacrificarnos por los que amamos, por sentir cada logro como un logro de todos. Gracias por estar siempre cuando los necesito, por hacerme sentir tan orgullosos de formar parte de ustedes

A mi pilla Yai por quererme tanto, por su dedicación y entrega, por hacer tuyas mis preocupaciones y desvelos.

A todas las personas lindas que han compartido conmigo durante estos 5 años de sacrificios, de tristezas y alegrías, a todos los que de una forma u otra han hecho posible que este sueño que parecía imposible se haga realidad.

DEDICATORIA

A mis abuelos Amparo, Candita y Juan.

A mis padres Olga y Pavel.

RESUMEN

Como parte de la rutina de producción mediática, los recursos fotográficos de los Medios de Prensa cubanos crecen considerablemente. A fin de garantizar su gestión, identificación y organización, se desarrolló en la Universidad de las Ciencias Informáticas el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel v1.0. Sin embargo, en su desarrollo no se tuvo en cuenta que brindara un manejo estadístico de dichos recursos, provocando un desconocimiento del patrimonio fotográfico, debido a la enorme cantidad de imágenes y la ausencia de procesos de control en este sentido.

El presente trabajo tiene como objetivo fundamental, desarrollar un módulo estadístico que permita la cuantificación y auditoría del uso de los recursos fotográficos en el sistema antes mencionado.

Su desarrollo se basó en la metodología OpenUp, haciendo uso del framework de desarrollo Symfony 2 y el Entorno de Desarrollo Integrado Netbeans 7.1. De igual modo, se utilizaron los lenguajes PHP 5.3, JavaScript, y UML para el modelado. Se emplearon el Visual Paradigm como herramienta CASE y el Sistema Gestor de Base de Datos PostgreSQL. Para los estilos se utilizaron CSS y el framework Bootstrap, y la librería HighChart para el graficado.

Como resultado, se obtuvo un módulo que permite al administrador del sistema filtrar y auditar los recursos fotográficos, así como visualizar los resultados del filtrado de forma gráfica. Posibilita además, establecer un control, tanto de los recursos como de los usuarios que interactúan con los mismos mediante la generación de diferentes reportes.

Palabras clave: recursos fotográficos, reportes estadísticos, sistema de gestión documental.

ÍNDICE DE CONTENIDOS

Introducción	1
Capítulo I: Fundamentación Teórica	5
1.1 Introducción.....	5
1.2 Gestión documental.....	5
1.2.1 Sistemas de Gestión Documental	6
1.3 Generación de datos estadísticos.....	8
1.3.1 ¿Qué es la estadística?	8
1.3.2 Reportes estadísticos y su representación.....	9
1.3.3 Generación de datos estadísticos en SGD.	11
1.4 Herramientas y tecnologías asociadas al desarrollo del sistema.....	15
1.4.1 Metodología de desarrollo de software.	15
1.4.2 Herramienta CASE.....	17
1.4.3 Lenguaje de Modelado Unificado.	18
1.4.4 Entorno integrado de desarrollo.	19
1.4.5 Marcos de trabajo (<i>Framework</i>).....	20
1.4.6 Lenguajes a utilizar.	21
1.4.7 Sistema Gestor de Bases de Datos (SGBD).....	23
1.4.8 Librerías para el graficado	23
1.5 Conclusiones.....	25
Capítulo II: Concepción y características del sistema	26
2.1 Introducción.....	26
2.2 Modelo del dominio	26
2.2.1 Definición de las clases del Modelo de dominio.....	26
2.3 Requisitos funcionales	27
2.4 Requisitos no funcionales.....	28
2.5 Modelo de Casos de Uso del sistema.....	28
2.5.1 Actor del sistema.....	29
2.5.2 Casos de Uso del sistema.	29
2.5.3 Diagrama de Casos de Uso del sistema.	29
2.5.4 Descripción de Casos de Uso del sistema.....	30
2.6 Conclusiones del capítulo	34
Capítulo III: Análisis y diseño del sistema	35
3.1 Introducción.....	35
3.2 Descripción de estilos arquitectónicos	35
3.3 Patrones de diseño	36

ÍNDICE DE CONTENIDOS

3.3.1	Decorator.....	36
3.3.2	Experto en información	36
3.3.3	Controlador.....	37
3.4	Modelo del análisis	37
3.4.1	Diagramas de clases del análisis	37
3.4.2	Diagramas de clases del análisis por casos de uso	38
3.5	Modelo del diseño.....	38
3.5.1	Diagramas de clases del diseño.....	38
3.5.2	Diagramas de interacción	39
3.6	Modelo de despliegue.....	40
3.7	Conclusiones del capítulo.....	41
Capítulo IV: Implementación y pruebas.....		42
4.1	Introducción.....	42
4.2	Diagrama de componentes.....	42
4.3	Principales aspectos del código fuente del módulo estadístico.	44
4.3.1	Estándares de codificación.....	44
4.3.2	Clases más relevantes del módulo estadístico.	45
4.4	Principales pantallas del módulo estadístico.	45
4.5	Validación del módulo estadístico.	48
4.5.1	Pruebas funcionales. Método y Técnica utilizados.	49
4.5.2	Pruebas de unidad e integración.	52
4.5.3	Pruebas de seguridad.....	52
4.6	Conclusiones del capítulo.....	53
Conclusiones generales.....		54
Recomendaciones.....		55
Referencias Bibliográficas.....		56
Bibliografía consultada		58
Glosario de términos.....		61

ÍNDICE DE FIGURAS

Figura 1. Gráfico de barras.	10
Figura 2. Gráfico de líneas.....	10
Figura 3. Gráfico de pastel.....	11
Figura 4. Historial de documentos en Núxeo.	13
Figura 5. Modelo del dominio.....	26
Figura 6. Diagrama de CU del sistema.....	30
Figura 7. Diagrama de clases del análisis del caso de uso Auditar recursos fotográficos.....	38
Figura 8. Diagrama de clases del análisis del caso de uso Imprimir gráfica.....	38
Figura 9. Diagrama de clases del análisis del caso de uso Exportar gráfica.....	38
Figura 10. Diagrama de clases del diseño con estereotipos web.....	39
Figura 11. Diagrama de secuencia del CU: Auditar recursos fotográficos.....	40
Figura 12. Diagrama de secuencia del CU: Filtrar recursos fotográficos.....	40
Figura 13. Modelo de despliegue.....	41
Figura 14. Diagrama de Componentes.....	43
Figura 15. Interfaz principal del módulo estadístico.....	46
Figura 16. Resultados de una búsqueda de las fotografías.....	47
Figura 17. Auditoría a una fotografía.....	48
Figura 18. Resultados de las pruebas.....	51

ÍNDICE DE TABLAS

Tabla 1. Comparación entre algunos tipos de librerías para el graficado.	24
Tabla 2. CU Auditar recurso fotográfico.	30
Tabla 3. CU Filtrar recursos fotográficos.	31
Tabla 4. CU Graficar resultados.	33
Tabla 5. Exportar gráfica.	49
Tabla 6. DCP: Auditar recurso fotográfico.	50
Tabla 7. DCP: Graficar resultados.	50
Tabla 8. Número de NC detectadas por iteraciones.	51
Tabla 9. Resumen de las NC detectadas en las pruebas funcionales.	52
Tabla 10. CU Exportar gráfica.	62
Tabla 11. CU Eliminar filtro dinámico.	63
Tabla 12. CU Mostrar últimos usuarios creados.	64
Tabla 13. DCP: Seleccionar tipo de gráfica.	65
Tabla 14. DCP: Eliminar filtro dinámico.	65

Introducción

El hombre durante su historia, ha tenido la necesidad de plasmar todas sus actividades como expresión testimonial, sin importar el formato, lenguaje o soporte, haciendo uso para ello, de toda clase de materiales como la piedra, el papiro o el papel.

Con el transcurso de los años, estos materiales han evolucionado a fin de garantizar la preservación de la información, que en correspondencia al nivel de desarrollo alcanzado, se ha convertido en lo que se conoce como gestión documental.

En sus inicios, *“... el dominio de la gestión documental correspondía exclusivamente a administradores y bibliotecarios, cuyas herramientas manuales básicas eran los libros de registro, las carpetas, archivadores, cajas y estanterías. En dichas herramientas se guardaban los documentos de papel y ficheros que permitían hacer referencias cruzadas, así como una larga lista de técnicas de recuperación de información mediante sistemas de codificación y clasificación”* (Ministerio de Fomento, 2009).

Como resultado del desarrollo actual de las Tecnologías de la Información y las Comunicaciones (TIC), la información crece de un modo vertiginoso, manifestándose en todas sus modalidades, y con ella la necesidad de procesarla y controlarla. Como respuesta, *“...surgen soluciones y servicios asociados a la gestión documental, permitiendo que la información pueda ser manipulada en formato electrónico; desempeñando un papel muy significativo para empresas que requieran digitalizar toda la información, y que presenten problemas en controlar las actividades y funciones que afectan el procesamiento, clasificación, almacenamiento y acceso de los archivos”*. (Mosquera, 2010)

En la actualidad, *“...los medios de comunicación masiva (MCM), especialmente los medios de prensa, se han visto inmersos en constantes cambios, destinados al mejoramiento y optimización de los procesos de producción y organización de la información, a fin de proporcionar a sus usuarios un mejor servicio”* (Medina, 2005) Uno de estos cambios encaminados a garantizar la inmediatez de la publicación lo constituyó la digitalización de la información, dígase textos, vídeos e imágenes.

Debido a esto, los fondos fotográficos de los MCM poseen en sus colecciones, millares de ejemplares, únicos e irrepetibles que crecen diariamente, como parte de la rutina de producción mediática; y son, además, herederos de la fotografía de importantes medios de prensa que existieron en diferentes épocas en Cuba. Por otro lado, constituyen un material permanentemente utilizado por los profesionales del sector periodístico, además de investigadores y otras instituciones.

La Universidad de las Ciencias Informáticas (UCI), es una de las instituciones que acomete acciones para informatizar la sociedad, incluyendo los MCM; desarrollando aplicaciones configurables de acuerdo a las

características de las redacciones correspondientes a instituciones periodísticas cubanas.

En este sentido, se desarrolló el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0, con el Administrador de Contenidos Empresariales (ECM por sus siglas en inglés) Alfresco. Este sistema permite la gestión, identificación y organización de los recursos fotográficos. Sin embargo, no se tuvo en cuenta que brindara un manejo estadístico de sus colecciones. Como consecuencia, actualmente, dicho sistema no permite conocer cuáles recursos han sido visualizados o modificados con mayor frecuencia, ni los archivados o modificados recientemente. De igual modo, no posibilita saber qué usuarios se encuentran conectados al sistema, así como aquellos que no han realizado visitas al mismo. Tampoco facilita la realización de búsquedas teniendo en cuenta criterios como: el autor, fecha de modificación o creación, nombre y descripción, lo que trae consigo un desconocimiento del patrimonio fotográfico, debido a la enorme cantidad de imágenes. Hasta el momento, el administrador se ve imposibilitado de establecer un control que le permita conocer la cantidad de recursos que han sido publicados, procesados o archivados y quiénes han sido los autores de dichos cambios, además de las fechas en las que se han efectuado.

De la situación antes planteada se identifica como **problema de investigación**: ¿Cómo contribuir al control de los recursos fotográficos en el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0, para la cuantificación y auditoría de su uso?

El **objeto de estudio** se centra en los procesos de Gestión Documental y el **campo de acción** está enmarcado en la generación de datos estadísticos en el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel v1.0.

Se propone como **objetivo general** desarrollar un módulo estadístico que permita la cuantificación y auditoría del uso de los recursos fotográficos en el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0.

Para complementar el objetivo de la investigación se formulan los siguientes **objetivos específicos** que posibilitan el desarrollo del mismo:

- Caracterizar el tratamiento estadístico en los Sistemas de Gestión Documental.
- Diseñar el Módulo estadístico para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel v1.0.
- Implementar las funcionalidades del Módulo estadístico para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0.
- Probar las funcionalidades del Módulo estadístico para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0.

Para dar cumplimiento al objetivo planteado se aplicaron los métodos científicos siguientes:

Métodos teóricos:

- **Histórico–lógico:** Mediante esta práctica se analizaron las tendencias de los Sistemas de Gestión Documental existentes, permitiendo documentar los elementos necesarios para garantizar la comprensión y un mejor funcionamiento de estos sistemas.
- **Analítico-Sintético:** Se utilizó en la revisión de documentos y artículos, donde se identificaron las ideas centrales relacionadas con el funcionamiento de los Sistemas de Gestión Documental; permitiendo obtener el conocimiento esencial para generar una propuesta de solución adecuada a las exigencias de los medios de prensa.

Métodos Empíricos:

- **Entrevista:** Se utilizó para obtener a partir de los criterios emitidos por miembros del equipo de desarrollo del proyecto, una parte de los requisitos funcionales y no funcionales de módulo estadístico desarrollado.

La presente investigación está estructurada de la siguiente manera: introducción, cuatro capítulos que se describen a continuación, conclusiones, recomendaciones, referencias bibliográficas y bibliografía.

Capítulo 1: “Fundamentación teórica”.

Se precisan elementos teóricos que sustentan la investigación y el desarrollo del tema propuesto, a través del estudio y análisis de soluciones existentes. Se hace referencia a los principales aspectos relacionados con los Sistemas de Gestión Documental, así como el análisis estadístico en los mismos. Se describen cada una de las diferentes tecnologías, herramientas y técnicas que se utilizarán para el desarrollo del módulo.

Capítulo 2: “Concepción y características del sistema”.

Se describe la propuesta del módulo estadístico, representando el Modelo de dominio donde se capturan los objetos importantes del entorno donde será empleado el módulo estadístico. Además se identifican los requerimientos funcionales y no funcionales, se elabora el diagrama de Casos de Uso y las descripciones de cada uno de ellos.

Capítulo 3: “Análisis y diseño del sistema”.

Se describe la solución que se propone a partir de los diagramas de clases del análisis, asociados a las funcionalidades del sistema y se representan los diagramas de clases del diseño, que reflejan una vista interna del sistema, así como el diagrama de despliegue.

Capítulo 4: “Implementación y pruebas al sistema”.

Se presentan los resultados arrojados por las pruebas al sistema en las diferentes iteraciones. Se muestra el diagrama de componentes, además de los diseños de casos de prueba basados en Casos de Uso.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Capítulo I: Fundamentación Teórica.

1.1 Introducción.

La gestión documental constituye *“un área de la administración general que se encarga de garantizar la economía y la eficiencia en la creación, mantenimiento, uso y disposición de los documentos administrativos durante todo su ciclo de vida”* (Walne, 1988). A fin de garantizar estos dos aspectos, surgen los Sistemas de Gestión Documental, permitiendo obtener una mejor organización y utilización del fondo documental, garantizando a su vez, la autenticidad y veracidad del mismo.

Debido al gran número de acciones que pueden realizar los usuarios en estos sistemas, durante el ciclo de vida de los documentos, se hace necesario establecer un control de dichas acciones.

En el presente capítulo, se estudiarán los principales conceptos relacionados a la gestión documental, así como algunos sistemas existentes en Cuba y el mundo. Además se abordarán los aspectos relacionados a la obtención de datos estadísticos en dichos sistemas. Por último, se analizarán y describirán un conjunto de herramientas y tecnologías que servirán de apoyo para el desarrollo del módulo estadístico.

1.2 Gestión documental.

“Desde la antigüedad, los documentos se han considerado una especie de objeto portador de información que tiene una base textual, registrados en libros, manuscritos, diarios y revistas.” (Campillo, 2010).

Con el objetivo de proporcionar una correcta organización a los mismos, surge la gestión documental *“como área de gestión responsable de un control eficaz y sistemático de la creación, la recepción, el mantenimiento, el uso y la disposición de documentos de archivo, incluidos los procesos para incorporar y mantener en forma de documentos la información y prueba de las actividades y operaciones de la organización”*. (ISO 15489-1:2006).

Según Alberch Fugueras, la gestión documental engloba un conjunto de operaciones comprometidas con la búsqueda de la economía y la eficacia en la producción, el mantenimiento, uso y destino final de los documentos a lo largo de su ciclo de vida; es decir, desde el momento de su concepción en las oficinas administrativas hasta su ingreso en las instituciones de archivo (Fugueras, 2003).

Por otra parte, Ponjuan Dante, define la gestión documental como un proceso administrativo que permite analizar y controlar sistemáticamente, a lo largo de su ciclo de vida, la información registrada que crea, recibe, mantiene o utiliza la organización en correspondencia con su misión, objetivos y operaciones. También la considera como un proceso para mantener la información en un formato que permita su acceso oportuno, y por ello se requiere de la realización de tareas y procedimientos particulares para cada fase de

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

su ciclo de vida y su explotación. La información registrada es evidencia de las actividades y transacciones de las organizaciones, y su uso oportuno permite a la organización una mayor eficacia en su labor (Gestión documental, gestión de información y gestión del conocimiento: evolución y sinergias, 2005).

Para Mena Mujica, uno de los aspectos más significativos de la gestión documental desde el enfoque de los archivos totales, es la implantación de sistemas integrales de tratamiento de la documentación que se ocupen de su producción, conservación, uso, explotación, y eliminación, según las necesidades de cada tipo de usuario, sean los propios productores o investigadores (Mujica, 2005).

Partiendo de los conceptos anteriores, puede afirmarse que la gestión documental es el proceso por el que transitan los documentos desde su creación en las oficinas hasta su almacenamiento final, buscando una mayor eficacia en la producción, el mantenimiento, uso y destino de estos a lo largo de su ciclo de vida.

Lograr esta eficacia en lugares donde el cúmulo de documentos es considerable, resulta algo engorroso, sobre todo cuando se pretende consultar grandes volúmenes de información física y electrónica. A fin de resolver los problemas de pérdida de tiempo tanto en la consulta como almacenamiento de la misma, surgen los Sistemas de Gestión Documental, constituyendo poderosas herramientas para el manejo y organización de la información.

1.2.1 Sistemas de Gestión Documental

Un Sistema de Gestión Documental (SGD), *“es una parte del sistema de información de la empresa desarrollado con el propósito de almacenar y recuperar documentos, que debe estar diseñado para coordinar y controlar todas aquellas funciones y actividades específicas que afectan a la creación, recepción, almacenamiento, acceso y preservación de los documentos, salvaguardando sus características estructurales, y contextuales, y garantizando su autenticidad y veracidad”* (Bustelo, 2000).

El uso de un SGD por una determinada empresa, arroja una serie de ventajas a la hora de almacenar, gestionar y hacer un buen seguimiento de los documentos pertinentes, como por ejemplo (IBAI Intranets, 2005):

- Rapidez y acceso inmediato a la hora del acceso instantáneo, centralizado y distribuido a toda la documentación, reducción de tiempo de consultas y tareas de archivo. Por otra parte, se soluciona el problema de localización, controlándose de forma total la documentación e información. Además, que la información puede ser compartida entre diferentes usuarios existiendo una fácil y rápida distribución o envío de documentos.
- Ahorro de material en la impresión de documentos, dígame fotocopias, copias impresas, entre otras; eliminación de desplazamientos y ahorro de espacio físico.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Seguridad y fiabilidad en la custodia de documentos de gran valor o confidencialidad, en la sustitución de documentos impresos por réplicas electrónicas, además que evita la duplicidad de documentos.

A pesar de contener numerosos beneficios, estos sistemas suelen proporcionar ciertas desventajas principalmente relacionadas al factor económico, debido a que su implantación puede suponer un alto costo, de igual modo, puede conllevar a emplear mucho tiempo y recursos si el volumen de datos a introducir es muy elevado. Sin embargo, puede afirmarse, que las ventajas se imponen sobre las desventajas, por tal razón, son muchas las organizaciones emprendedoras que optan por hacer uso de SGD para la preservación de su patrimonio documental.

De manera general, un SGD provee facilidad de almacenamiento y rapidez de acceso a la gran cantidad de información y documentos que deben ser visitados, por lo cual deben poseer una serie de características, como son: una alta escalabilidad y rendimiento, integración con otro software empresarial, herramientas ofimáticas y de escritorio, además de poder hacer empleo de tecnologías de la web.

Independientemente de los beneficios o inconvenientes que presenten los SGD, en los mismos suelen desarrollarse diferentes procesos, los cuales a consideración de varios autores consultados, pueden variar de acuerdo a las características del sistema y de la institución que lo implementa. Dichos procesos “... *son el conjunto de actos concatenados, mediante los cuales se le da seguimiento al ciclo de vida de los documentos, desde su producción o incorporación, hasta su disposición final*” (Martinic, 2011). A continuación, se describen los procesos mínimos que según Beatriz Mercado, deben estar presentes en un SGD:

- **Proceso de incorporación:** Conjunto de operaciones de verificación y control que se deben realizar para la admisión de los documentos que son remitidos por las unidades productoras de documentos hacia el SGD. Deberá incluir además, su digitalización, pues no necesariamente pueden estar todos inicialmente en este formato, de esta manera se obtendrá una primera homogenización de los mismos. De igual modo, se hace necesario el registro de los datos más importantes de los documentos a incorporar como pueden ser: autor, identificador, una breve descripción, tipo de documento y la fecha en la que es recepcionado.
- **Proceso de organización:** una vez que han sido recepcionados y digitalizados los documentos, es necesario llevar a cabo un conjunto de acciones orientadas a su clasificación, ordenación, indexación y descripción. La Indexación se realiza a nivel de metadatos o, si es accesible, a través de los contenidos del propio documento, permitiendo elaborar un índice sobre el cual hacer

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

búsquedas más eficientes. Este proceso adquiere suma importancia debido a que de lograr una correcta organización, depende el éxito y rapidez de las consultas que se deseen realizar posteriormente.

- **Proceso de almacenamiento:** posteriormente, debe procederse a introducir en la base de datos del SGD los diferentes documentos, garantizando su integridad.
- **Proceso de recuperación o consulta de documentos:** conjunto de acciones encaminadas a brindar acceso a los mismos a fin de visualizarlos o gestionarlos. De este modo, los administradores del SGD, pueden realizarles auditorías, permitiendo establecer un control sistemático de las acciones que realizan los usuarios sobre estos, así como generar un conjunto de reportes estadísticos que pueden incluir la visualización del estado de los documentos, cantidad de usuarios que han creado o modificado dichos documentos y las fechas en que se han producido tales cambios.
- **Proceso de transferencia de documentos:** traspaso periódico de la documentación de un archivo a otro del sistema o fuera de éste, de acuerdo a plazos de permanencia establecidos para la documentación en cada uno de ellos.

1.3 Generación de datos estadísticos.

1.3.1 ¿Qué es la estadística?

“La Estadística es la disciplina que le facilita al hombre el estudio de datos masivos, para obtener conclusiones válidas y efectuar predicciones razonables de ellos; y así mostrar una visión de conjunto clara y de fácil apreciación. De forma práctica, la Estadística proporciona los métodos científicos para la recopilación, organización, resumen, representación y análisis de datos, o de hechos, que se presenten a una evaluación numérica; tales como: características biológicas o sociológicas, fenómenos físicos, producción, calidad, población, riqueza, y otros” (Unidad Educativa Colegio Loyola–Gumilla, 2001).

Es considerada además como la *“ciencia que se ocupa del estudio de fenómenos de tipo genérico, normalmente complejos y enmarcados en un universo variable, mediante el empleo de modelos de reducción de la información y de análisis de validación de los resultados en términos de representatividad”* (Sánchez, 2008).

La estadística en función de su metodología, por sus procedimientos y alcances bien definidos, se clasifica en dos grandes ramas:

La **Estadística Inductiva o Inferencial:** provee conclusiones o inferencias, basándose en los datos simplificados y analizados; detectando las interrelaciones que pueden unirlos, las leyes que los rigen y eliminando las influencias del azar; llegando más allá de las verificaciones físicas posibles. Sobre la base

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

de la muestra estudiada saca conclusiones, o sea, hace inferencia o inducción, en cuanto al universo o población, de donde se obtuvo dicha muestra.

La **Estadística Descriptiva o Deductiva** que analiza metódicamente los datos, simplificándolos y presentándolos en forma clara; eliminando la confusión característica de los datos preliminares. Permite la elaboración de cuadros, gráficos e índices bien calculados; suficientemente claros, como para disipar las dudas y la obscuridad de los datos masivos. Se limita a describir los datos que se analizan, sin hacer inferencias en cuanto a los no incluidos en la muestra.

La presentación de datos estadísticos, constituye en sus diferentes modalidades, uno de los aspectos de más uso en la estadística descriptiva, permitiendo visualizar la presentación de los mismos sobre el comportamiento de las diferentes variables. Una forma de mostrar estos datos, es a través de los reportes estadísticos.

1.3.2 Reportes estadísticos y su representación.

Un reporte es un informe o noticia que puede ser impreso, digital u audiovisual, con la finalidad de transmitir una información, aunque puede tener diversos objetivos.

Los reportes estadísticos por su parte, son informes que organizan y exhiben la información contenida en una base de datos. Su función es aplicar un formato determinado a los datos, para mostrarlos por medio de un diseño atractivo y de fácil interpretación para los usuarios. Los reportes presentan diversos niveles de complejidad, desde una lista o enumeración, hasta gráficos mucho más desarrollados.

Existen tres tipos de representaciones de los datos, la representación escrita, que es usada cuando la información a representar contiene pocos valores; la representación tabular, cuando los datos se presentan a través de un conjunto de filas y de columnas que responden a un ordenamiento lógico; y por último la presentación gráfica que proporciona mayor rapidez en la comprensión de los datos.

En este último tipo, *“los gráficos se pueden utilizar para ilustrar patrones de grandes cantidades de datos o para comunicar un hallazgo clave o un mensaje. Se debe considerar el uso de gráficos si se desea mostrar”* (Ginebra, 2009):

- Comparación: ¿Cuánto? ¿Qué elemento es más grande o más pequeño?
- Cambios a lo largo del tiempo: ¿Cómo evoluciona una variable?
- Distribución de frecuencia: ¿Cómo se distribuyen los elementos? ¿Cuáles son las diferencias?
- Correlación: ¿Están vinculadas dos variables?
- Parte relativa de un todo: ¿Cómo se encuentra un elemento en comparación con el total?

Un aspecto crucial cuando se decide representar los datos a través de gráficos, es la correcta elección del tipo a utilizar, debido a que no todos se ajustan a la naturaleza de lo que se desea representar. A

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

continuación se hace referencias a algunos tipos de gráficos existentes, teniendo en cuenta además cuándo es idóneo su uso.

- **Gráfico de barras:** representa gráficamente un conjunto de datos o valores en forma de barras rectangulares de longitudes en correspondencia a dichos valores. Se utiliza para comparar las frecuencias o valores de las distintas categorías o grupos. Las barras pueden orientarse vertical u horizontalmente, aunque en sentido horizontal y ordenadas de manera ascendente, facilitan la interpretación de los datos. Además, deben ser más anchas que los espacios entre ellas, las cuales se recomienda no sobrepasen el 40% del ancho de la barra.

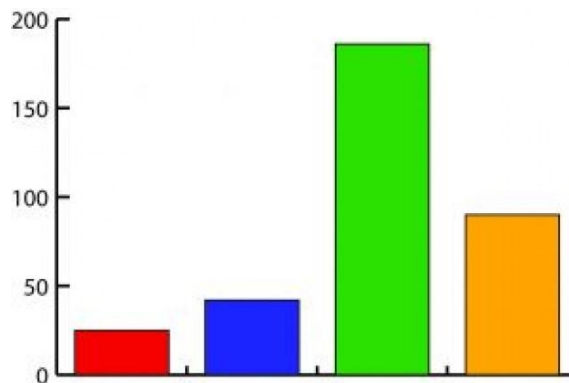


Figura 1. Gráfico de barras.

- **Gráfico de Líneas:** presenta los datos como una serie de puntos conectados por una línea, su principal uso es para representar los datos de un gran número de grupos o visualización de tendencias en los datos a lo largo del tiempo. Sus parámetros pueden ser modificados para mejorar la interpretación, siempre y cuando se preserven los datos.

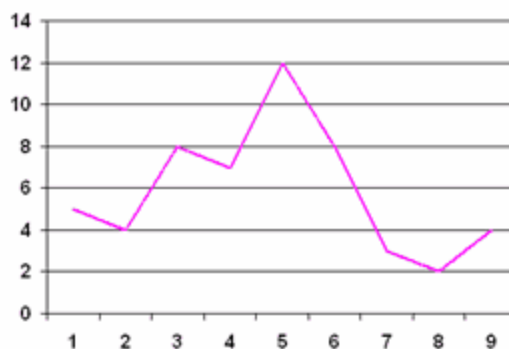


Figura 2. Gráfico de líneas.

- **Gráfico de pastel:** muestra los datos como áreas rellenas de colores o diseños, es usado para representar los datos de un número de grupos limitados (no superior a seis). El uso de este tipo de gráfico no es recomendado por muchos estadísticos, debido a que puede resultar tediosa la

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

comparación entre los diferentes segmentos del pastel y, aún más, cuando se deban comparar datos entre diferentes gráficos de este tipo. Una alternativa que puede usarse es la de etiquetar los segmentos con sus valores reales o nombres de las categorías que representan.

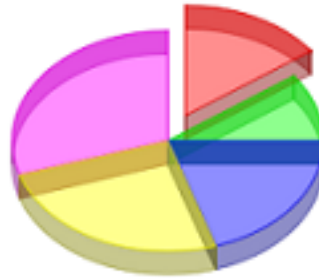


Figura 3. Gráfico de pastel.

En cuanto a la presentación tabular, permite de forma más sintetizada y comprensible mostrar los resultados, sobre todo cuando la información es de tipo repetitivo. Además, permite mostrar frecuencias, relaciones, contrastes, variaciones y tendencias mediante una presentación ordenada de la información.

Existen dos tipos de tablas, las simples, también llamadas de presentación o demostración y las complejas o de referencia como también se les conoce. Las simples se utilizan para presentar los hechos con respecto a uno o más grupos de investigaciones independientes, mientras que las complejas permiten ofrecer la división de las categorías en dos o más subcategorías, es decir, una o varias columnas y/o filas son a su vez subdivididas para representar o resaltar una condición importante del fenómeno que se estudia.

Resulta recomendable al emplear tablas, seguir un mismo diseño, utilizar unas líneas o un sombreado suave para animar a los usuarios a leer en horizontal o en vertical, cuidar que las columnas estén uniformemente espaciadas y no demasiado separadas.

A modo de conclusión, se determina que, para representar los datos en el Módulo estadístico para el SGD de Imágenes Digitales Ecumene Pyxel, se emplearán la representación gráfica para facilitar la comprensión de la información y la representación tabular para sintetizarla.

1.3.3 Generación de datos estadísticos en SGD.

Normalmente, en un SGD, los usuarios suelen ejecutar un gran número de acciones sobre los diferentes contenidos almacenados. Tener un control sobre dichas acciones, resulta sumamente importante debido a que permite tener constancia de la gestión de los contenidos. Por tal razón, diferentes SGD han incorporado de una u otra manera, entre sus funcionalidades, la generación de datos estadísticos.

A continuación se realiza un estudio de algunos de estos sistemas, a fin de analizar la generación o no de datos estadísticos y de este modo, identificar posibles funcionalidades a incorporar al Módulo estadístico para el SGD de Imágenes Digitales Ecumene Pyxel v1.0.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Nuxeo

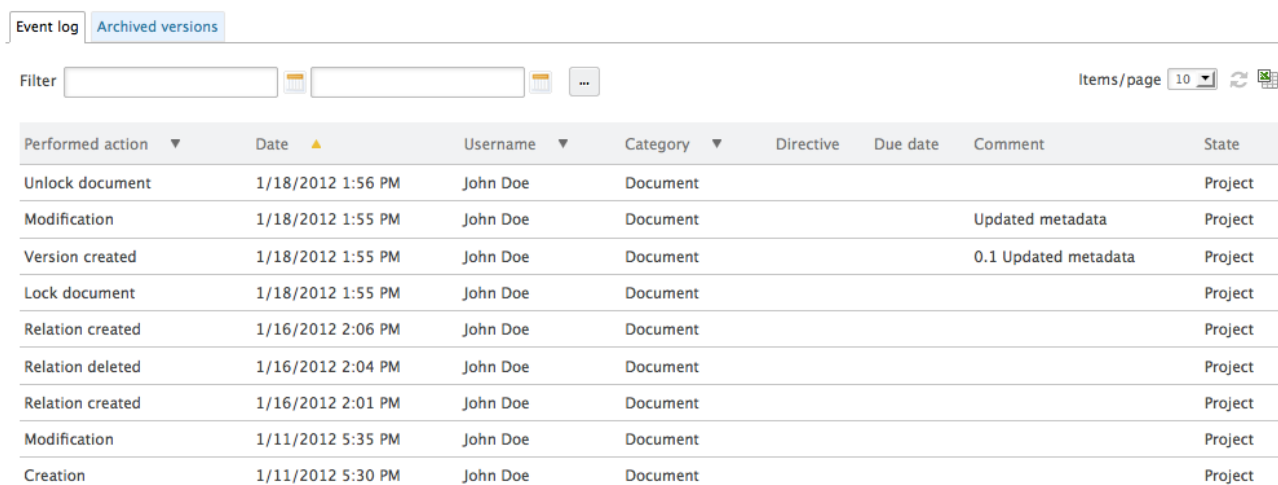
Es un sistema de administración de contenidos libre, basado en estándares abiertos y de escala empresarial para Windows y sistemas operativos similares a Unix. Está diseñado para usuarios que requieren un alto grado de modularidad y rendimiento escalable. Está desarrollado en Java y es utilizado como software de gestión documental para documentos, páginas web, registros, imágenes y desarrollo colaborativo de contenidos. Además permite (Nuxeo, 2012):

- Gestión de documentos.
- Gestión de contenidos web.
- Versionado a nivel de repositorio.
- Gestión de registros.
- Gestión de imágenes.
- Publicación integrada.
- Flujo de trabajo basado en jBPM.
- Búsquedas implementadas con el motor Lucene.
- Servidores descentralizados.
- Soporte multiplataforma (Windows, Linux, Solaris, Mac OS).
- Interfaz gráfica basada en navegadores de Internet.
- Integración de escritorio con Microsoft Office y OpenOffice.Org.

En cuanto a la auditoría a documentos, Nuxeo establece un control sobre la gestión de los mismos, almacenando en un historial la información de todas las acciones que se realizan sobre ellos, tales como la fecha y hora, el usuario, el comentario que se realizó, etc. Dicho historial, muestra además, las versiones anteriores del documento que fueron creadas, permitiendo eliminarlas o modificarlas. (Nuxeo, 2000).

Además, contiene un sistema de alertas que informa a los usuarios que deciden seguir la actividad de un determinado espacio, cuándo se producen cambios en dicho espacio, así como los usuarios responsables de los mismos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA



The screenshot shows the 'Event log' section of the Nuxeo interface. At the top, there are tabs for 'Event log' and 'Archived versions'. Below the tabs is a filter input field and a 'Items/page' dropdown set to 10. The main content is a table with the following columns: Performed action, Date, Username, Category, Directive, Due date, Comment, and State. The table contains 10 rows of log entries.

Performed action	Date	Username	Category	Directive	Due date	Comment	State
Unlock document	1/18/2012 1:56 PM	John Doe	Document				Project
Modification	1/18/2012 1:55 PM	John Doe	Document			Updated metadata	Project
Version created	1/18/2012 1:55 PM	John Doe	Document			0.1 Updated metadata	Project
Lock document	1/18/2012 1:55 PM	John Doe	Document				Project
Relation created	1/16/2012 2:06 PM	John Doe	Document				Project
Relation deleted	1/16/2012 2:04 PM	John Doe	Document				Project
Relation created	1/16/2012 2:01 PM	John Doe	Document				Project
Modification	1/11/2012 5:35 PM	John Doe	Document				Project
Creation	1/11/2012 5:30 PM	John Doe	Document				Project

Figura 4. Historial de documentos en Núxco.

DoCASU

DoCASU es una interfaz de usuario (UI por sus siglas en inglés) para el sistema de código abierto ECM Alfresco. El objetivo del proyecto DoCASU es proporcionar a la Comunidad del ECM Alfresco, una UI personalizada enfocándose principalmente en la experiencia del usuario, es decir que tenga una mayor facilidad de uso, capacidad de respuesta y rapidez.

Es menos confuso para el usuario final, lo que permite una mayor aceptación del sistema. Fue diseñado por la empresa Optaros y liberado bajo la Licencia Pública General (GPL por sus siglas en inglés). Con DoCASU los usuarios de Alfresco, tienen la posibilidad de simplificar y facilitar las soluciones para acceder, agregar, buscar y modificar documentos.

DoCASU puede ser tomado por los desarrolladores como punto de partida para la realización de interfaces de usuario más amplias, o simplemente como la base de aprendizaje para desarrollar aplicaciones basadas en los Web Scripts.

Aviladoc

Aplicación Web desarrollada sobre una plataforma de software libre con base de datos centralizada, destinada a la gestión, tramitación y resguardo de archivos electrónicos. Herramienta de trabajo que gestiona el tramitado interno de los documentos a través de la intranet, sin necesidad de copias o envíos físicos y permite el control de la entrada y salida de documentos, el manejo de anotaciones a los mismos por parte de los usuarios, el tratamiento diferencial para los documentos pendientes, así como la recuperación de la información a través del contenido de la ficha del documento.

Algunas características (Desoft, 2012):

- Constituye un archivo digitalizado, que agiliza las tareas de registro, búsqueda, reproducción y distribución de los documentos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- El tramitado interno de los documentos se realizará sin necesidad de reproducirlos o distribuirlos físicamente a las oficinas involucradas garantizando rapidez en la gestión.
- Permite acceder directamente a las carpetas personales de MS Outlook o a los dispositivos de almacenamiento, para darle entrada al documento, donde quiera que este se encuentre.
- Logra integrar la gestión documental y la gestión de archivo.
- Permite generar reportes por criterios de filtrado y ordenamiento especificado por el usuario.
- Ofrece un historial de los documentos desde que fue creado hasta su eliminación, conservando las acciones que se realizaron sobre el mismo, el usuario que las acometió y las fechas en que lo hizo.

Se encuentra implantado en diferentes instituciones cubanas, tales como: el Ministerio de la Informática y las Comunicaciones (MIC), la Empresa de Telecomunicaciones de Cuba (ETECSA), el Centro de Inmunología Molecular (CIM) y la Oficina Central del Ministerio del Interior (MININT).

eXcriba

eXcriba es un Gestor de Documentos Administrativos, desarrollado en la UCI. Brinda a los clientes la optimización de recursos importantes en cualquier institución, dígame tiempo, organización y tecnología en función del correcto tratamiento de la documentación producida o recibida de las entidades.

Este sistema permite identificar un mercado creciente de necesidades y adaptar el software a las características propias y específicas de los interesados. A continuación se mencionan algunas de las funcionalidades que permite el sistema:

- Descripciones de fondos documentales tanto físicos como digitales.
- Gestión de los plazos de transferencias de las series documentales de manera automática.
- Extrae, importa y exporta metadatos desde el repositorio, de manera singular y múltiple.
- Audita el acceso a los fondos documentales.
- Captura los metadatos de los contenidos desde el momento de su creación, hasta su conservación permanente o destrucción.
- Gestiona el Cuadro de Clasificación funcional de la entidad.
- Brinda seguimiento de la auditoría sobre los documentos electrónicos.

Se encuentra implantado en diferentes instituciones cubanas, tales como: el MIC, la Aduana General de la República, el Archivo Nacional de la República de Cuba (ARNAC) y la Oficina Central del Ministerio del Interior (MININT).

Luego de analizar algunos SGD existentes en Cuba y el mundo, puede afirmarse que existe una tendencia en la mayor parte de los estudiados, de realizar de cierto modo, una auditoría de los documentos por medio del historial que se registra de las acciones llevadas a cabo por los usuarios. Por tanto, sería recomendable que el Módulo estadístico para el SGD de Imágenes Digitales Ecumene Pyxel v1.0, incluyera la funcionalidad: auditar recursos fotográficos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.4 Herramientas y tecnologías asociadas al desarrollo del sistema.

1.4.1 Metodología de desarrollo de software.

El empleo de una metodología durante el desarrollo de un software, le confiere a este y a la investigación que se le asocia, transparencia y calidad, elementos muy perseguidos por el cliente y necesarios para el producto.

La aplicación correcta de una metodología ayuda a mejorar el tiempo de desarrollo del software; definir con exactitud el personal que requiere el proyecto, las herramientas a utilizar, conocimientos concretos sobre el problema a resolver, entre otros aspectos que deben ser chequeados y controlados.

Existen varias metodologías de desarrollo de software que pueden ser clasificadas en ágiles y robustas, dejando al equipo de desarrollo de un producto, tomar la decisión de cuál es la que más se ajusta a sus características. Para guiar el desarrollo del módulo estadístico, se decidió emplear una metodología ágil, debido a que el mismo estará caracterizado por:

- Presentar un equipo de desarrollo muy pequeño y por ende con pocos roles.
- Énfasis en la entrega frecuente de funcionalidades desarrolladas.
- Optimizar el tiempo y evitar la demora en la generación de artefactos.
- Propiciar el trabajo conjunto del cliente con el equipo de desarrollo.

A continuación, se evidencia el estudio realizado de algunas metodologías ágiles a fin de seleccionar la más idónea.

Crystal Clear

Crystal “Clear” es la más ágil de la familia de metodologías Crystal propuestas por Alistair Cockburn y de la que más documentación se dispone. Puede ser usado en proyectos pequeños y como casi todos los otros métodos de Crystal, consiste en valores, técnicas y procesos. Realiza menos énfasis en la documentación exhaustiva y más en versiones funcionales de software que puedan ser probadas.

Crystal maneja iteraciones cortas con retroalimentación frecuente por parte de los usuarios/clientes, minimizando de esta forma la necesidad de productos intermedios. Otra de las cuestiones planteadas es la necesidad de disponer de un usuario real aunque sea de forma compartida para realizar validaciones sobre la interfaz del usuario y para participar en la definición de los requerimientos funcionales y no funcionales del software (Amaro, 2007).

En lugar de una interpretación lineal, Cristal Clear enfatiza el proceso como un conjunto de ciclos anidados.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

OpenUP

Dentro de las metodologías ágiles se encuentra **OpenUP**, un proceso iterativo e incremental, mínimo, completo, extensible y unificado que contiene el conjunto mínimo de prácticas que ayudan a los equipos a ser más eficaces en el desarrollo de software. OpenUP abraza una filosofía pragmática y ágil que se centra en la naturaleza colaborativa de desarrollo de software, está basada en RUP, pero es mucho más ligera, debido a que elimina gran parte de las fases del proceso (lo que internamente OpenUP llama Disciplinas) de desarrollo del software.

Se construyó sobre una donación realizada por IBM del Basic Unified Process y fue entregada a Eclipse a finales del 2005 y renombrado como OpenUP en el 2006.

Intenta incluir dentro de su proceso de desarrollo únicamente el contenido imprescindible para garantizar un proceso eficiente y de calidad. Por este motivo, no incluye guías sobre aspectos que suelen aparecer en la gestión de proyectos como puedan ser equipos de desarrollo grandes, negociaciones contractuales, uso de tecnologías específicas y definición de aplicaciones de seguridad o críticas. Esto no quiere decir que no contemple estos aspectos, simplemente los deja sin especificar, pero en cualquier momento se puede extender la metodología para cubrir alguno de ellos.

Está organizada dentro de cuatro áreas principales de contenido: Comunicación y Colaboración, Intención, Solución y Administración.

Esta metodología consta de cuatro fases (Alonso, 2012):

- **Inicio:** el propósito en esta fase es lograr concurrencia entre todos los *stakeholders* sobre los objetivos del ciclo de vida para el proyecto.
- **Elaboración:** establecer la línea base de la arquitectura del sistema y proporcionar una base estable para el gran esfuerzo de desarrollo de la siguiente fase.
- **Construcción:** completar el desarrollo del sistema basado en la arquitectura.
- **Transición:** asegurar que el software está listo para ser entregado a los usuarios.

Programación Extrema (XP)

Programación Extrema (XP del inglés Extreme Programming), *“es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo”* (Canós, 2008).

Esta metodología dirige las actividades de construcción del software mediante un desarrollo incremental, a través de la planificación de distintas iteraciones, cada iteración finaliza con un período de prueba y validación, que garantiza al cliente, la exactitud en las funcionalidades y la satisfacción de sus necesidades. Además, se basa en la realimentación continua entre el cliente y el equipo de desarrollo, así como en la

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. Es recomendada para proyectos con requisitos imprecisos y muy cambiantes. El ciclo de vida de XP consta de 6 fases, Exploración, Planificación de la entrega (*Release*), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos (Alva, 2013):

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

Utiliza como técnica para describir los requisitos de software las historias de usuario, en las cuales, el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. De este modo, el cliente se encuentra mucho más cerca del proceso de desarrollo. Se elimina la fase inicial de recopilación de requerimientos, y se permite que éstos se vayan identificando a lo largo del proyecto, de una manera ordenada.

Como conclusión del estudio realizado, se propone el uso de OpenUP debido a que es una metodología de código abierto, diseñada para pequeños equipos de desarrollo y facilita la detección temprana de errores a través de un ciclo iterativo. Además a diferencia de las otras metodologías estudiadas, divide el ciclo de desarrollo en fases, permitiendo una mayor organización en el mismo.

1.4.2 Herramienta CASE.

Se puede definir a las Herramientas CASE como: un conjunto de programas, métodos, utilidades y técnicas que dan asistencia a los analistas, ingenieros de software y desarrolladores, facilitando la automatización del ciclo de vida del desarrollo de un software, completamente o en alguna de sus fases (INEI, 2003).

Visual Paradigm 8.0

Visual Paradigm 8.0, es una herramienta para la creación de diagramas UML. Su diseño es centrado en casos de uso y enfocado al negocio, generando un software de mayor calidad. Propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Este ha sido concebido para soportar el ciclo de vida completo del proceso de desarrollo del software, a través de la representación de todo tipo de diagramas. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un orientado a objetos.

A continuación se exponen algunas de sus características (León, 2010):

- Cuenta con disponibilidad en múltiples plataformas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Su diseño está centrado en casos de uso y enfocado al negocio.
- Presenta capacidad para realizar ingeniería directa e inversa.
- Permite que el modelo y el código permanezcan sincronizados durante todo el ciclo de desarrollo.
- Es fácil de instalar y actualizar.
- Sus ediciones suelen ser compatibles.

Rational Rose Enterprise Edition 7.0

Rational Rose Enterprise Edition 7.0, es una herramienta de diseño orientada a objetos que permite representar gráficamente el sistema, haciendo énfasis en los detalles más importantes, centrándose en los casos de uso y enfocándose hacia un software de mayor calidad.

“Proporciona un conjunto de prestaciones controladas por modelo para desarrollar muchas aplicaciones de software, incluidas aplicaciones Ada, ANSI C++, C++, CORBA, Java, Java EE, Visual C++ y Visual Basic. El software permite acelerar el desarrollo de estas aplicaciones con código generado a partir de modelos visuales mediante el lenguaje UML (Unified Modeling Language)” (IBM, 2007).

Es funcional únicamente en plataformas Windows y requiere de una licencia para su uso. Ofrece mecanismos para realizar ingeniería inversa y posibilita generar código en diferentes lenguajes a partir de un diseño en UML. Brinda la facilidad de que varias personas trabajen a la vez, permitiendo que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo, manteniendo el control sobre la propagación de los cambios en dicho espacio. Provee cobertura a una gran parte de los flujos de RUP tales como:

- Captura de requisitos (parcialmente).
- Análisis y diseño (completamente).
- Implementación (como ayuda).
- Control de cambios y gestión de configuración (parcialmente).

A partir del estudio realizado, se determina la utilización del Visual Paradigm 8.0 como herramienta de modelado, debido que a diferencia del Rational Rose, es una herramienta multiplataforma. Posee generador de informes en formato PDF y HTML, además de una interfaz agradable e intuitiva. Soporta el ciclo de vida completo del desarrollo de software. Permite dibujar todos los tipos de diagramas de clases y generar código a partir de diagramas.

1.4.3 Lenguaje de Modelado Unificado.

Como consecuencia de la selección de la herramienta Visual Paradigm para el modelado del módulo estadístico, resulta necesario el uso del Lenguaje Unificado de Modelado (UML del inglés Unified Modeling

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Lenguaje). El mismo permite visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

- **Visualizar:** expresar de una forma gráfica un sistema, de manera que otro lo pueda entender.
- **Especificar:** especificar cuáles son las características de un sistema antes de su construcción.
- **Construir:** construir los sistemas diseñados a partir de los modelos especificados.
- **Documentar:** los propios elementos gráficos sirven como documentación del sistema desarrollado.

Además, es un lenguaje consolidado, fácil de aprender y posibilita una comunicación fluida entre los diversos actores.

Un modelo UML describe lo que supuestamente hará el sistema, pero no especifica cómo implementarlo.

1.4.4 Entorno integrado de desarrollo.

Un Entorno Integrado de Desarrollo (IDE por sus siglas en inglés), es un programa compuesto por un conjunto de herramientas destinadas a la implementación de aplicaciones que puede soportar uno o varios lenguajes de programación. El entorno de desarrollo agiliza la producción de un software, pues está provisto de un grupo de funciones y librerías enfocadas en la confección del mismo (Comunidad Netbean, 2012).

NetBeans 7.3

El IDE NetBeans está creado para escribir, compilar, depurar y ejecutar programas. Permite ser extendido mediante un número importante de módulos. Es un producto libre y gratuito sin restricciones de uso, de código abierto, escrito completamente en Java usando la plataforma NetBeans. Soporta el desarrollo de todos los tipos de aplicaciones Java (J2SE, Web, EJB y aplicaciones móviles). NetBeans cuenta con un complemento para integrarlo con Twig (motor que utiliza Symfony2 para el desarrollo de plantillas Web), además soporta PHP 5, CCS 3, HTML 5 y es sencillo de integrar con el marco de trabajo Symfony2.

Zend Studio 8.0

Zend Studio o Zend Development Environment, es un completo IDE para el lenguaje de programación PHP. Es multiplataforma y está escrito en Java. Proporciona depuración completa y pruebas de apoyo que ayudan a analizar y solucionar los problemas rápidamente, así como detección de errores de sintaxis en tiempo real. Se distribuye bajo una licencia comercial y no presenta versiones en español. Posee varias características que facilitan la edición de código, tales como el coloreado de la sintaxis y la terminación automática del código.

Luego de analizar los IDE descritos anteriormente, se decide optar por la utilización de NetBeans 7.3 debido a que se integra satisfactoriamente con el *framework* de desarrollo Symfony y se distribuye por medio de la GPL, a diferencia del Zend Studio que es un IDE privativo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.4.5 Marcos de trabajo (*Framework*)

Un marco de trabajo (*framework*), es una estructura básica utilizada para solucionar problemas comunes de un modo simplificado.

Enfocando este concepto al desarrollo web, se trata de un conjunto de procesos, técnicas y archivos previamente confeccionados, que facilitan y aceleran la producción de sitios y aplicaciones web.

A pesar que existe la tendencia de asociar el concepto de *framework* sólo al desarrollo de software, vale resaltar que también existen algunos que se enfocan a facilitar el diseño web. A continuación, se exponen los marcos de trabajos a utilizar en el desarrollo del módulo.

Symfony 2

Symfony 2, es un marco de desarrollo para PHP, desarrollado completamente con PHP 5, diseñado para optimizar el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web (Modelo - Vista - Controlador). Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Una aplicación desarrollada con el marco de trabajo Symfony 2, contará con código claro y organizado consistentemente. Symfony 2 promueve la reutilización y permite a los nuevos desarrolladores ser productivos en el proyecto con mayor rapidez. El 100% del código que es escrito, es para la aplicación, pues no se necesita desarrollar o mantener servicios públicos de bajo nivel, tales como la carga automática de clases, el enrutado o la reproducción de controladores.

Proporciona acceso a librerías como Doctrine, además a plantillas, seguridad, formularios, validación y traducción. Permite que las URL sean totalmente flexibles gracias al componente *Routing*. La arquitectura centrada en HTTP de Symfony 2 le da acceso a herramientas, tal como la memoria caché HTTP (Cisneros, 2008).

Bootstrap 2.2.2

Bootstrap es un *framework* que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. La mayor ventaja es que posibilita crear interfaces que se adapten a los distintos navegadores, apoyándose en un *framework* potente con numerosos componentes web que ahorrarán mucho esfuerzo y tiempo.

Bootstrap ofrece una serie de plantillas CSS y ficheros JavaScript que permiten integrar el *framework* de forma sencilla y potente en las aplicaciones web (GENBETA:dev, 2012).

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como *tablets* y móviles, a distintas escalas y resoluciones.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Se integra perfectamente con las principales librerías Javascript, por ejemplo JQuery.
- Ofrece un diseño sólido usando estándares como CSS3/HTML5.

La selección de los *framework* anteriormente descritos, se sustenta en las capacidades que presentan de agilizar y facilitar el trabajo de los desarrolladores, así como de reutilizar código. En el caso particular de Symfony, fue elegido debido a que genera un código limpio y fácil de comprender, además que permite un mantenimiento sencillo del mismo. Por otro lado, Bootstrap fue seleccionado por la necesidad de simplificar el proceso de creación del diseño web del módulo, combinando CSS y JavaScript.

1.4.6 Lenguajes a utilizar.

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina. Permite especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias (Turriza, 2010).

La selección de Symfony como *framework* de desarrollo conlleva al empleo de los lenguajes: PHP, HTML, CSS y JavaScript.

PHP 5.3

Es un lenguaje de programación utilizado para la creación de sitios web. Surgió en 1995, desarrollado por PHP Group. Es un lenguaje interpretado en el lado servidor utilizado para la generación de páginas dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. No necesita ser compilado para ejecutarse. Para su funcionamiento, necesita tener instalado Apache o IIS¹ con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de los lenguajes C, Java y Perl con algunas características específicas. PHP está diseñado principalmente, para ser un lenguaje más seguro con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, siguiendo algunas prácticas correctas de programación. A continuación se mencionan algunas de sus características más relevantes. (Centro Ideoinformática, 2012):

- Es multiplataforma.
- Posee funcionalidades de conexión con la mayoría de los SGBD: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otros.
- Soporta la orientación a objeto y sus paradigmas.
- Su potencial se puede expandir utilizando extensiones.

¹ servidor web y conjunto de servicios para el sistema operativo Microsoft Windows.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- No requiere definición de tipos de variables ni manejo detallado de bajo nivel.
- Fácil de aprender.
- Posee una extensa documentación con ejemplos de uso de sus funciones.
- Es distribuido bajo licencia libre.
- La versión más reciente soportada es la 5.3.x sobre la que se basan varios *framework* de desarrollo y sistemas de gestión de contenidos desarrollados sobre este lenguaje.

JavaScript

Lenguaje de scripts desarrollado por Netscape para incrementar las funcionalidades del lenguaje HTML. El navegador del lado cliente se encarga de interpretar las sentencias JavaScript contenidas en una página HTML y ejecutarlas adecuadamente. Es un lenguaje orientado a eventos, es decir, permite el desarrollo de scripts² que ejecuten acciones en respuesta a dichos eventos.

Es similar a Java, aunque no es orientado a objetos y no dispone de herencia, sigue el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base y extendiendo su funcionalidad.

El código JavaScript puede ser integrado dentro de las páginas web, para evitar incompatibilidades el W3C diseñó un estándar denominado Document Object Model (DOM) o Modelo de Objetos de Documento, que es una forma de representar documentos estructurados tales como una página web HTML o un documento XML independiente de cualquier lenguaje.

Es soportado por los diferentes navegadores web mediante la implementación del estándar ECMAScript. (Centro Ideoinformática, 2012)

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* (www). 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, scripts escritos en otros lenguajes como PHP y JavaScript.

Es fácil de entender y utilizar (Centro Ideoinformática, 2012).

CSS 3

Cascading Style Sheets (CSS) u Hojas de Estilo en Cascada, constituyen el estándar para la inserción de estilo (tamaños, colores, tipografías, espacios, bordes) a documentos estructurados, como por ejemplo,

² guión o conjunto de instrucciones que permiten la automatización de tareas creando pequeñas utilidades.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

documentos HTML o XML. El objetivo de la definición de este estándar del W3C es permitir la separación entre las normas de presentación y el propio contenido a mostrar. (Centro Ideoinformática, 2012)

1.4.7 Sistema Gestor de Bases de Datos (SGBD).

PostgreSQL 8.4.

PostgreSQL soporta el almacenamiento de objetos de gran tamaño; se destaca en ejecutar consultas y subconsultas complejas; permite la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. Es escalable tanto en la cantidad de datos que puede manipular como en la cantidad de usuarios concurrentes que puede atender. Puede ajustarse a la cantidad de memoria que posee el sistema de forma óptima, teniendo la posibilidad de soportar una mayor cantidad de peticiones simultáneas de manera correcta. Implementa el uso de subconsultas y transacciones, permitiendo que su funcionamiento sea eficaz.

La velocidad de respuesta de este SGBD es constante, o sea, presenta la misma velocidad al gestionar tanto una base de datos pequeña como una grande. Algunos de los lenguajes que se pueden usar ligados a Postgre son Java, Perl y PHP.

Teniendo en cuenta las características anteriores, además de su uso por el núcleo del sistema, se opta por su utilización en el desarrollo del módulo.

1.4.8 Librerías para el graficado

Open Flash Chart 2.0

Open Flash Chart es una librería de código abierto, distribuida bajo la licencia GPL que permite realizar gráficas estadísticas en Flash usando lenguajes de programación del lado del servidor.

Con esta librería se pueden realizar gráficas interactivas de barras, pastel, líneas y de dispersión, entre otras. Para ello se puede emplear cualquiera de los siguientes lenguajes de programación: PHP, Ruby, Perl, Python, .NET o Java (Carrillo, Germán, 2008). Posibilita además, redimensionar los gráficos, guardarlos como una imagen o resaltar algún punto interesante que aparezca en los mismos.

pChart 2.0

pChart está orientada a la creación de gráficos y diagramas en PHP, recupera la información para generarlos mediante consultas SQL, de un archivo CSV o simplemente introduciendo los datos manualmente en el código fuente de la página. Para el graficado requiere hacer uso de la librería GD de PHP y a diferencia de Open Flash Chart, las gráficas que produce son estáticas, ya que se generan en un archivo de imagen con formato PNG (SourceForge.net).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Proporciona sintaxis de codificación orientada a objetos y está acorde con los nuevos estándares de la web, posee una comunidad de desarrollo que trabaja en posibles mejoras y amplia documentación de ayuda para quienes optan por su uso.

HighCharts 3.0

“Highcharts es una librería escrita puramente en HTML5 y JavaScript, que ofrece gráficos intuitivos e interactivos a un sitio o página web... admite gráficos de línea, áreas, columnas, barras, circulares y dispersión y otros, además brinda la posibilidad de combinarlos en uno solo.” (Highsoft Solutions AS, 2011). Se basa en tecnologías de navegador nativo y no necesita de *plugins* del lado del cliente como Flash o Java. Además, no requiere instalar ningún fichero en el servidor y funciona en todos los navegadores modernos incluyendo los del iPhone y el iPad.

Highcharts maneja el concepto de “temas”, que consiste en un conjunto de opciones predefinidas que se aplican como opciones por defecto a la hora de crear un gráfico, sin embargo, permite a los desarrolladores, crear nuevos temas lo cual suele ser útil para separar el estilo de un gráfico de los datos.

Presenta un módulo que permite exportar los gráficos a formato PNG, JPG, PDF o SVG, o imprimirlos directamente desde la página web.

Tabla 1. Comparación entre algunos tipos de librerías para el graficado.

Librería	Licencia que utiliza	Requerimientos	Diversidad de gráficos	Diseño refinado de los gráficos
Open Flash Chart	GPL	Flash Player 9	X	X
pChart	GPL	Librería GD en el servidor de PHP.	X	-
HighCharts	GPL	-	X	X

Luego de haber analizado las librerías para el graficado descritas con anterioridad, se llega a la conclusión de utilizar Highcharts debido a que está desarrollada bajo la GPL y se basa únicamente en tecnologías de navegador nativo, por lo que no requiere de *plugin* u otro software a diferencia de las librerías pChart y Open Flash Chart. Además, posee diversidad de gráficos, los que pueden ser personalizados a fin de lograr diseños más vistosos.

Además, permite mostrar un texto de ayuda contextual con información sobre cada punto y serie de un determinado gráfico al pasar el ratón él, así como invertir los ejes de los gráficos y rotar sus etiquetas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.5 Conclusiones

El análisis de las tendencias de algunos de los Sistemas de Gestión Documental existentes en Cuba y el mundo, principalmente en cuanto a la generación de reportes estadísticos, permitió concluir que, pese a existir interesantes propuestas, las mismas no resuelven los problemas identificados en la presente investigación. Aunque vale resaltar que, en su mayoría, realizan auditorías a los recursos que gestionan, manteniendo un historial de las acciones que realizan los usuarios sobre los mismos. Valorando la importancia de dicha auditoría, se estima conveniente su incorporación como una funcionalidad del módulo estadístico.

El estudio de las diferentes formas de representar los datos estadísticos, permitió identificar las formas más convenientes de presentar los datos a mostrar en el módulo, de acuerdo a la naturaleza de los mismos, de modo tal que se facilitara su comprensión por parte del administrador del sistema; resultando como las más idóneas: la representación gráfica y la tabular.

Por último, la investigación de las herramientas y tecnologías a utilizarse en el desarrollo del módulo, arrojó como resultado la decisión de seguir la metodología de desarrollo OpenUP, debido a que contiene el conjunto mínimo de prácticas que ayudan a los equipos a ser más eficaces en el desarrollo de software. Se define el uso de los marcos de trabajo: Symfony 2 y Bootstrap 2.2.2. Como lenguajes a utilizar se establecen: PHP5.3 como lenguaje de programación web, HTML5 como lenguaje de marcado, CSS3 para los estilos y JavaScript. El SGBD a usar será PostgreSQL v8.4, y el IDE, NetBeans 7.3 debido a que brinda opciones de autocompletado y se integra con el *framework* seleccionado. La librería para el graficado de los datos será Highcharts 3.0 y el lenguaje y herramienta CASE de modelado los constituirán el UML y el Visual Paradigm 8.0 respectivamente.

CAPÍTULO 2: CONCEPCIÓN Y CARACTERÍSTICAS DEL MÓDULO ESTADÍSTICO.

Capítulo II: Concepción y características del sistema.

2.1 Introducción

El modelado del negocio permite entender los problemas actuales de la compañía, asegurando que los clientes, usuarios, desarrolladores y otros involucrados tengan un mismo entendimiento de la empresa. En este capítulo se describe una propuesta de un Módulo estadístico para el SGD de Imágenes Digitales Ecumene Pyxel. Además, se representa el Modelo de dominio donde se capturan los objetos importantes del entorno donde será empleado el módulo estadístico. Se identifican los requerimientos funcionales y los no funcionales, se elabora el diagrama de Casos de Uso y las descripciones de cada uno de ellos.

2.2 Modelo del dominio

Un Modelo del dominio es una representación de las clases conceptuales o entidades del mundo real, no de componentes de software (Pressman, 2000). Esto ayuda a los usuarios, clientes y desarrolladores a utilizar un lenguaje común para poder entender el contexto en que se emplaza el sistema, captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema y no incluyen las responsabilidades de las personas que ejecutan las actividades.

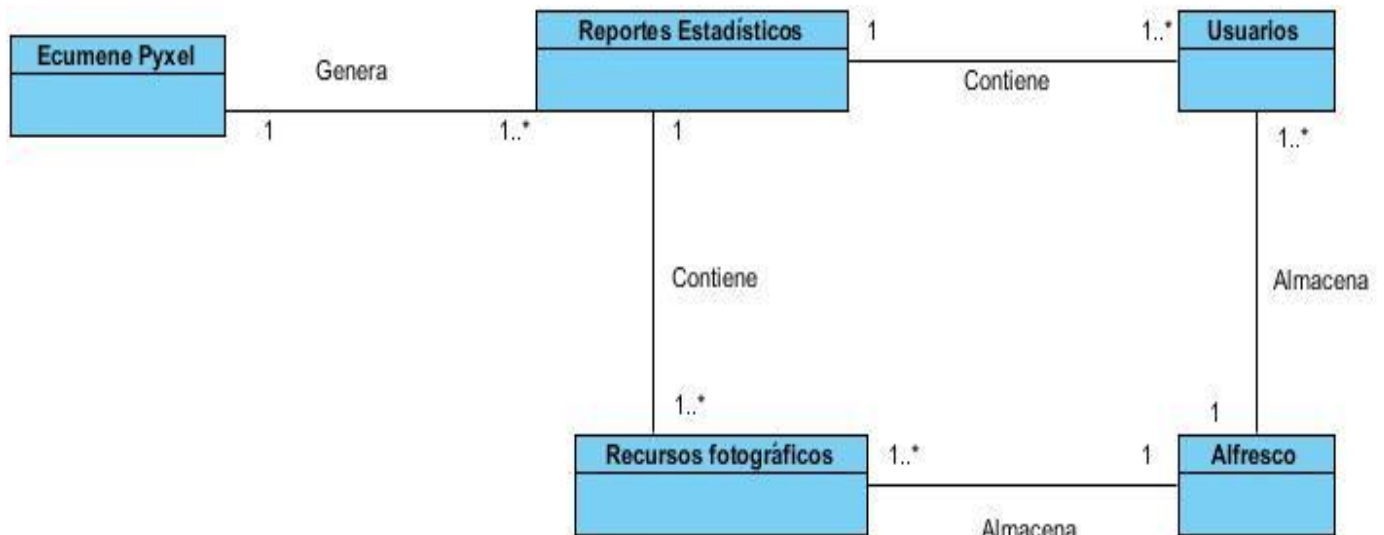


Figura 5. Modelo del dominio.

2.2.1 Definición de las clases del Modelo de dominio

Para un mayor entendimiento del Modelo de dominio del Módulo estadístico del SGD de Imágenes Digitales Ecumene Pyxel, se hace una breve descripción de las clases que lo conforman:

CAPÍTULO 2: CONCEPCIÓN Y CARACTERÍSTICAS DEL MÓDULO ESTADÍSTICO.

- **Alfresco:** se refiere al lugar donde se almacenan los recursos fotográficos.
- **Reportes Estadísticos:** muestra los reportes que se generarán en el módulo, en función de los usuarios y distintos tipos de recursos fotográficos.
- **Ecumene Pyxel:** se refiere al sistema donde serán mostrados los resultados de los reportes estadísticos.
- **Recursos fotográficos:** se refiere a las fotografías, unidades documentales y galerías temáticas contenidas en el Alfresco.
- **Usuarios:** se refiere a los usuarios registrados en el sistema que están en el Alfresco.

2.3 Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. En este caso las funcionalidades que brindará el Módulo estadístico para el SGD de Imágenes Digitales Ecumene Pyxel v1.0 son las siguientes:

RF1. Mostrar cantidad de recursos fotográficos que se han creado por tipo.

RF2. Mostrar los recursos fotográficos más leídos.

RF3. Mostrar los recursos fotográficos más modificados

RF4. Mostrar los usuarios conectados al sistema.

RF5. Mostrar los últimos usuarios creados en el sistema.

RF6. Mostrar los usuarios que no han realizado visitas al sistema

RF7. Mostrar los últimos recursos fotográficos creados en el sistema.

RF8. Mostrar los últimos recursos fotográficos modificados en el sistema.

RF9. Auditar recurso fotográfico.

RF10. Graficar resultados.

RF11. Imprimir gráfica.

RF12. Exportar gráfica.

RF13. Filtrar recursos fotográficos.

RF14. Mostrar detalles de la gráfica.

RF15. Seleccionar tipo de gráfica.

RF16. Eliminar filtro dinámico.

RF17. Actualizar reportes estáticos.

CAPÍTULO 2: CONCEPCIÓN Y CARACTERÍSTICAS DEL MÓDULO ESTADÍSTICO.

2.4 Requisitos no funcionales.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener e imponen restricciones en el diseño o la implementación.

Software.

- **RnF1.** Sistema operativo Ubuntu 10.4 o superior.
- **RnF2.** SGBD PostgreSQL 8.4 o superior.
- **RnF3.** El sistema debe desarrollarse con la tecnología Symfony 2.0
- **RnF4.** El lenguaje de programación utilizado será PHP 5.

Hardware.

- **RnF1.** Servidores con al menos 2 GB de Memoria RAM, procesador Core i3 o superior, HDD entre 250 y 500 Gb.

Seguridad.

- **RnF1.** Los permisos de accesibilidad, lectura y escritura son gestionados en dependencia del rol asignado a cada usuario del sistema.
- **RnF2.** Establecer procedimiento para las salvas periódicas de la información en otros dispositivos.

Legales.

- **RnF1.** Las herramientas seleccionadas para el desarrollo del producto están respaldadas por licencias libres, bajo las condiciones de software libre.

Portabilidad.

- **RnF1.** El producto debe cumplir con los requisitos de integración con el resto de las aplicaciones desarrolladas o por desarrollar en el SGD de Imágenes Digitales Ecumene Pyxel.

Soporte.

- **RnF1.** La documentación resulta un requisito indispensable como evidencia de la evolución del Proyecto.

2.5 Modelo de Casos de Uso del sistema.

Un modelo de Casos de Uso, es un modelo del sistema que contiene actores, Casos de Uso (CU) y sus relaciones. Describe la funcionalidad propuesta del nuevo sistema. Cada usuario se representa mediante uno o más actores (Rodríguez, 2009).

CAPÍTULO 2: CONCEPCIÓN Y CARACTERÍSTICAS DEL MÓDULO ESTADÍSTICO.

2.5.1 Actor del sistema.

En el Módulo estadístico para el SGD de Imágenes Digitales Ecumene Pyxel v1.0, se identificó como único actor el **Administrador** que realiza labores de auditoría y control sobre los recursos fotográficos.

2.5.2 Casos de Uso del sistema.

Los CU son el componente clave del modelado. Su propósito es ilustrar cómo un sistema permite a un actor cumplir una meta, mostrando todos los posibles caminos apropiados que pueden tomar para cumplirla, así como las situaciones que podrían hacerlo fallar (Pressman, 2000).

2.5.3 Diagrama de Casos de Uso del sistema.

Un diagrama de CU del sistema representa gráficamente los procesos y su interacción con los actores. Cada CU debe comunicarse con al menos un actor, lo contrario representa un error en el modelo o en los requerimientos planteados (Pressman, 2000).

CAPÍTULO 2: CONCEPCIÓN Y CARACTERÍSTICAS DEL MÓDULO ESTADÍSTICO.

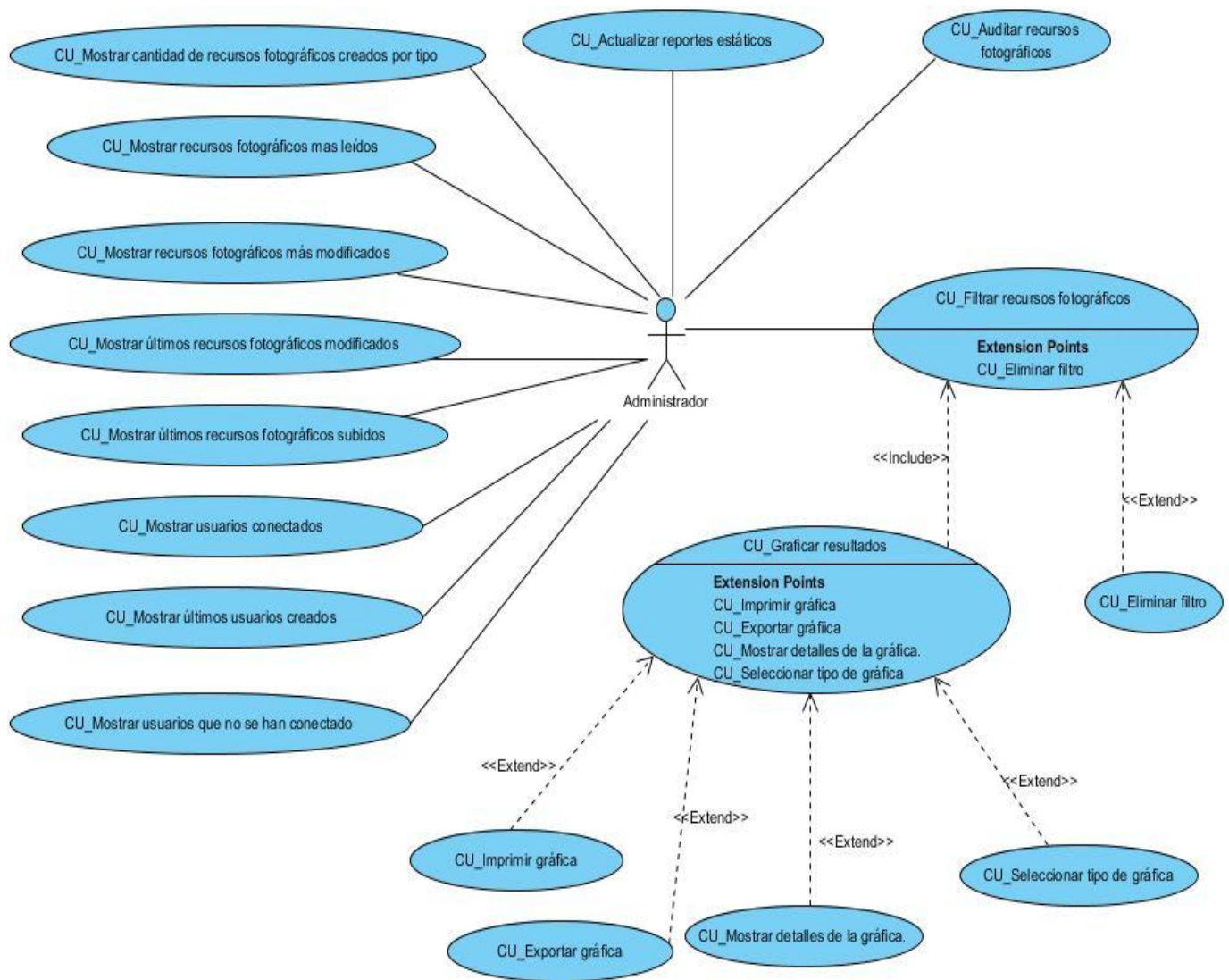


Figura 6. Diagrama de CU del sistema.

2.5.4 Descripción de Casos de Uso del sistema

En esta sección se detallan los CU del sistema, con una descripción del flujo de eventos y la interacción entre actores y el sistema. La descripción contiene además otras informaciones como son resumen del CU, referencias, requisito no funcional asociado y prioridad.

Tabla 2. CU Auditar recurso fotográfico.

Objetivo	Mostrar los campos por lo que está formado un recurso fotográfico.
Actores	Administrador.
Resumen	El CU inicia cuando ya generado un reporte dinámico, el Administrador desea ver las características de uno de los elementos que conforman este reporte, el sistema muestra

CAPÍTULO 2: CONCEPCIÓN Y CARACTERÍSTICAS DEL MÓDULO ESTADÍSTICO.

	dichas características, termina así el CU.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	El usuario debe estar autenticado en el sistema.	
Postcondiciones	Se audita el recurso fotográfico.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1	Selecciona el tipo de recurso fotográfico que desea auditar.	1.1 Muestra los recursos filtrados del tipo seleccionado.
2	Hace clic en el botón <i>Auditar</i> correspondiente al recurso fotográfico deseado.	2.1 Muestra en una nueva interfaz todos los atributos correspondientes al recurso auditado con sus respectivos valores.
3		Finaliza el CU.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos funcionales	no	
Asuntos pendientes		

Tabla 3. CU Filtrar recursos fotográficos.

Objetivo	Obtener los recursos fotográficos dado los campos por los que desea filtrar.	
Actores	Administrador.	
Resumen	El CU se inicia cuando el Administrador desea seleccionar los campos por los que va a filtrar e introduce los valores deseados para generar el reporte, el sistema filtra los recursos fotográficos teniendo en cuenta los campos especificados y le brinda la opción de guardar el filtro realizado, finalizando así el CU.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	El usuario debe estar autenticado en el sistema.	
Postcondiciones	Se genera el reporte.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1	Selecciona los campos por los que desea filtrar el	1.1 Muestra el campo seleccionado y permite

CAPÍTULO 2: CONCEPCIÓN Y CARACTERÍSTICAS DEL MÓDULO ESTADÍSTICO.

	reporte de recursos fotográficos.	introducir su valor. 1.2 Muestra las opciones: <ul style="list-style-type: none"> • <i>Filtrar.</i> • <i>Anular</i> (Ir a sección Anular). • <i>Guardar</i> (Ir a sección Guardar).
2	Introduce el valor deseado en el campo y selecciona la opción <i>Filtrar.</i>	2.1 Muestra el reporte de recursos fotográficos en correspondencia con los valores especificados y permite guardar el filtro creado (<i>Ver Sección Guardar</i>).
Sección "Anular"		
	Actor	Sistema
2	Selecciona la opción <i>Anular.</i>	2.1 Elimina los campos seleccionados por el usuario y oculta las opciones: <ul style="list-style-type: none"> • <i>Filtrar.</i> • <i>Anular.</i> • <i>Guardar.</i>
Sección "Guardar"		
	Actor	Sistema
2	Selecciona la opción <i>Guardar.</i>	2.1 Muestra un campo para introducir el nombre del nuevo filtro.
3	Introduce el nombre con el que desea guardar el filtro y selecciona la opción <i>Aceptar.</i>	3.1 Almacena el filtro en el listado de <i>Filtros dinámicos.</i>
Flujos Alternos		
Nº Evento <no introduce el nombre para el nuevo filtro>		
	Actor	Sistema
3	No introduce el nombre del nuevo filtro y selecciona la opción <i>Aceptar.</i>	3.1 Muestra el mensaje " <i>Debe especificar un nombre para el filtro.</i> "
Sección		
Relaciones	CU Incluidos	CU Graficar resultados
	CU Extendidos	CU Eliminar filtro dinámico.
Requisitos funcionales	no	
Asuntos pendientes		

CAPÍTULO 2: CONCEPCIÓN Y CARACTERÍSTICAS DEL MÓDULO ESTADÍSTICO.

Tabla 4. CU Graficar resultados.

Objetivo	Mostrar de forma gráfica los resultados de los filtros realizados.	
Actores	Administrador.	
Resumen	El CU se inicia cuando el Administrador desea filtrar los recursos fotográficos, el sistema muestra dicho reporte en forma de gráfica, finaliza así el CU.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	El usuario debe estar autenticado en el sistema. Debe haberse desarrollado el <i>CU Filtrar recursos fotográficos</i> .	
Postcondiciones	Se grafican los resultados.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1	Selecciona el tipo de gráfica en la que desea visualizar los datos del reporte y hace clic en el botón <i>Filtrar</i> (Ver <i>CU Filtrar recursos fotográficos</i> y <i>CU Seleccionar tipo de gráfica</i>).	1.1 Muestra lo datos del reporte en forma del tipo de gráfico seleccionado.
2		Termina el CU.
Flujos alternos		
Nº Evento <No Selecciona el tipo de gráfica>		
1	No selecciona el tipo de gráfica y hace clic en el botón <i>Filtrar</i> .	1.1 Muestra los datos del reporte en forma del tipo de gráfico predeterminado (pastel).
2		Termina el CU.
Relaciones	CU Incluidos	
	CU Extendidos	CU Imprimir gráfica. CU Exportar gráfica. CU Mostrar detalles de la gráfica. CU Seleccionar tipo de gráfica.
Requisitos funcionales	no	
Asuntos pendientes		

Para consultar otras descripciones de CU ver *Anexo 1* o el artefacto Especificación de Casos de Uso registrado en el Expediente de Proyecto del proyecto Gestión Documental para la Prensa (GDPrensa).

CAPÍTULO 2: CONCEPCIÓN Y CARACTERÍSTICAS DEL MÓDULO ESTADÍSTICO.

2.6 Conclusiones del capítulo

En el presente capítulo, se lograron identificar los objetos más importantes del entorno donde será empleado el módulo estadístico, permitiendo establecer sus relaciones y de este modo, comprender mejor su funcionamiento. La obtención de los requisitos funcionales y no funcionales contribuyó a la correcta identificación de los casos de uso. La descripción detallada de los mismos, constituirá una guía fundamental para los desarrolladores, haciendo de la construcción una fase más sencilla.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

Capítulo III: Análisis y diseño del sistema.

3.1 Introducción

El análisis tiene como propósito fundamental conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea más fácil de mantener y que ayude a estructurar el sistema completo, incluyendo la arquitectura. El diseño es más que analizar los requerimientos refinándolos y estructurándolos, pues se preocupa en realidad de dar forma al sistema de manera que proporcione vida a todos los requisitos, incluidos todos los no funcionales que incorpora el mismo (Jacobson, 2000). De esta forma, la disciplina de análisis y diseño especifica y describe cómo se va a implementar el sistema, es decir, los diseñadores de software determinan la mejor solución técnica a partir de los requerimientos, de la arquitectura del sistema más adecuada y el diseño detallado necesario previo a las actividades de implementación. En este capítulo, se describe la solución que se propone a partir de los diagramas de clases del análisis y se representan los diagramas de clases del diseño, que reflejan una vista interna del sistema.

3.2 Descripción de estilos arquitectónicos

Symfony está basado en un patrón clásico de arquitectura web conocido como Modelo-Vista-Controlador (MVC). La utilización de este marco de trabajo para el desarrollo del producto conlleva por ende a la implementación de este patrón, que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos:

- **Modelo:** es un conjunto de clases que representan la información del mundo real o que el sistema debe procesar, sin tomar en cuenta la forma en la que esa información va a ser mostrada ni los mecanismos que hacen que esos datos estén dentro del modelo (Bascón, Ernesto Pantoja, 2004). En este se encuentran las clases que son generadas de forma automática según la estructura de la base de datos.
- **Vista:** presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario. La vista es la encargada de originar las páginas que son mostradas como resultado de las acciones.
- **Controlador:** responde a eventos, usualmente acciones del usuario, que son la lógica de la aplicación, e invoca peticiones al modelo y a la vista.

La arquitectura MVC proporciona grandes ventajas, como la organización del código, la reutilización y la flexibilidad. Symfony2 específicamente, toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

3.3 Patrones de diseño

Un patrón de diseño es un conjunto de reglas que describen cómo afrontar tareas y solucionar problemas que surgen durante el proceso de desarrollo de software (Pérez, 2009). En otras palabras, proporcionan una solución adaptable y escalable a problemas de desarrollo de software que están sujetos a contextos similares.

Dos de las familias de patrones de diseño más conocidas son los Patrones Generales de Asignación de Responsabilidad (GRAPS por sus siglas en inglés), los cuales permiten la asignación de responsabilidades y los establecidos por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides conocidos como Gang of Four (GoF). Estos últimos son agrupados en tres grupos (Pérez, 2009):

- Creacionales: crean objetos de manera que ya no se tengan que instanciar directamente, proporcionando mayor flexibilidad para decidir qué objetos usar.
- Estructurales: permiten crear grupos de objetos para contribuir con la realización de tareas complejas.
- Patrones de comportamiento: definen la comunicación entre los objetos del sistema y el flujo de la información.

En el desarrollo del Módulo estadístico para el SGD de Imágenes Digitales Ecumene Pyxel, se utilizan patrones de las familias antes mencionadas.

3.3.1 Decorator

Decorator, también conocido como *Wrapper*, es un patrón estructural que permite añadir responsabilidades extras a objetos concretos de manera dinámica. Proporciona una alternativa flexible a la herencia para extender funcionalidad (Gerra, Esther Sánchez, 2009).

En Symfony2, la vista se separa en un *layout* y en una plantilla. Normalmente, el *layout* es global en toda la aplicación y contiene el código HTML que es común en la mayoría de las páginas. La plantilla sólo se encarga de visualizar las variables definidas en el controlador, esto se puede ver como que el *layout* decora la plantilla lo cual es una implementación del patrón Decorator.

3.3.2 Experto en información

Principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo, se puede obtener un diseño con mayor cohesión y reusabilidad, debido a que la clase es más independiente.

Se apoya en la extensibilidad. Aumenta la reusabilidad, haciendo que las clases sean más independientes (Recabarren, 2009). Los objetos utilizan su propia información para llevar a cabo sus tareas. Se distribuye el comportamiento entre las clases que contienen la información requerida. Son más fáciles de entender y

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

mantener.

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 de 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 que representan.

3.3.3 Controlador

Es el encargado de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. La mayor parte de los sistemas reciben eventos de entrada externa, en cualquiera de los casos que puedan existir, si se recurre a un diseño orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada.

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 realice las operaciones solicitadas, pero siempre manteniendo las clases controladoras sin saturarse.

3.4 Modelo del análisis

El lenguaje que se utilizó en el análisis se basa en un modelo de objetos conceptual llamado modelo de análisis. El modelo de análisis ayuda a estructurar los requisitos y proporciona una estructura basada en aspectos tales como la flexibilidad ante los cambios y la reutilización. Esta estructura no solo es útil para el mantenimiento de los requisitos como tal, sino que también se utiliza como entrada en las actividades de diseño e implementación.

3.4.1 Diagramas de clases del análisis

El diagrama de clases del análisis representa las clases del análisis que participan en las realizaciones de los casos de uso-análisis y las relaciones que se establecen entre ellas. En el análisis se presentan los siguientes estereotipos de clases:

- **Clases de frontera o interfaz:** *“se utilizan para modelar la interacción entre el sistema y sus actores, además, ... modelan las partes del mismo que dependen de dichos actores, lo cual implica que clarifican y reúnen los requisitos en los límites del sistema”* (Jacobson, 2000). Durante el diseño, estas clases son refinadas para tomar en consideración los mecanismos de interfaz seleccionados o implementados, además de facilitar la comunicación con otros sistemas.
- **Clases de entidad:** representan y modelan la información manejada en el CU que posee una larga vida y que a menudo es persistente. Reflejan entidades del mundo real, que resultan necesarias para realizar tareas internas del sistema.
- **Clases de control:** *“representan coordinación, secuencia, transacciones, y control de otros objetos*

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

y se usan con frecuencia para encapsular el control de un CU en concreto” (Jacobson, 2000).

3.4.2 Diagramas de clases del análisis por casos de uso

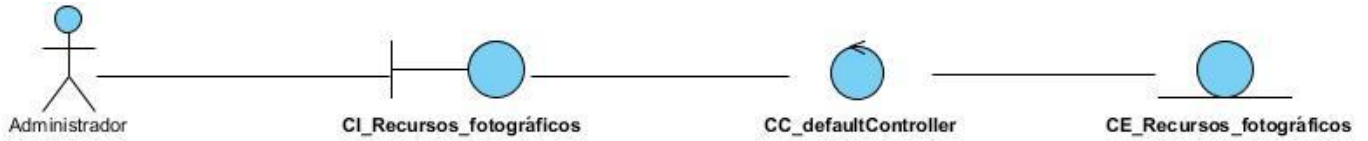


Figura 7. Diagrama de clases del análisis del caso de uso Auditar recursos fotográficos.



Figura 8. Diagrama de clases del análisis del caso de uso Imprimir gráfica.



Figura 9. Diagrama de clases del análisis del caso de uso Exportar gráfica.

3.5 Modelo del diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los CU centrándose en los requisitos, junto con otras restricciones relacionadas con el entorno de implementación. Este tiene impacto en el sistema a considerar y sirve de abstracción a la implementación, y de esa forma, es utilizado como una entrada fundamental a las actividades de implementación (Jacobson, 2000).

3.5.1 Diagramas de clases del diseño

En el siguiente diagrama se representan las clases participantes en las realizaciones de CU, subsistemas y sus relaciones.

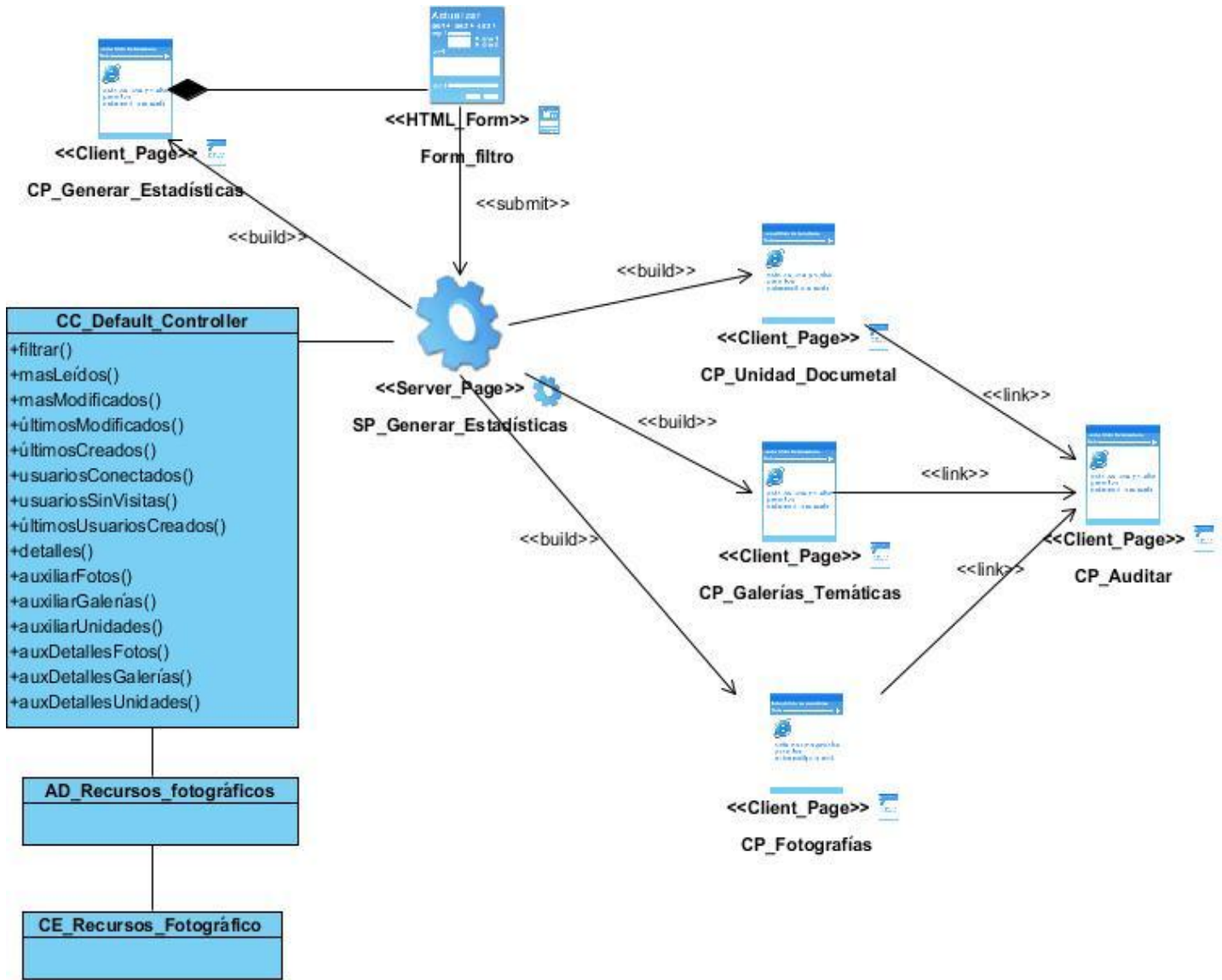


Figura 10. Diagrama de clases del diseño con estereotipos web.

3.5.2 Diagramas de interacción

Los diagramas de interacción constituyen una descripción del modo en que cada operación detectada en los diagramas de colaboración lleva a cabo sus responsabilidades y modifica el estado del sistema, los mismos pueden representarse a través de los Diagramas de Colaboración y los Diagramas de Secuencia (Jacobson, 2000).

El diagrama seleccionado en el presente trabajo, fue el Diagrama de Colaboración, debido a que destaca la organización de los objetos que participan en una interacción, creando enlaces entre ellos y añadiendo mensajes a dichos enlaces.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

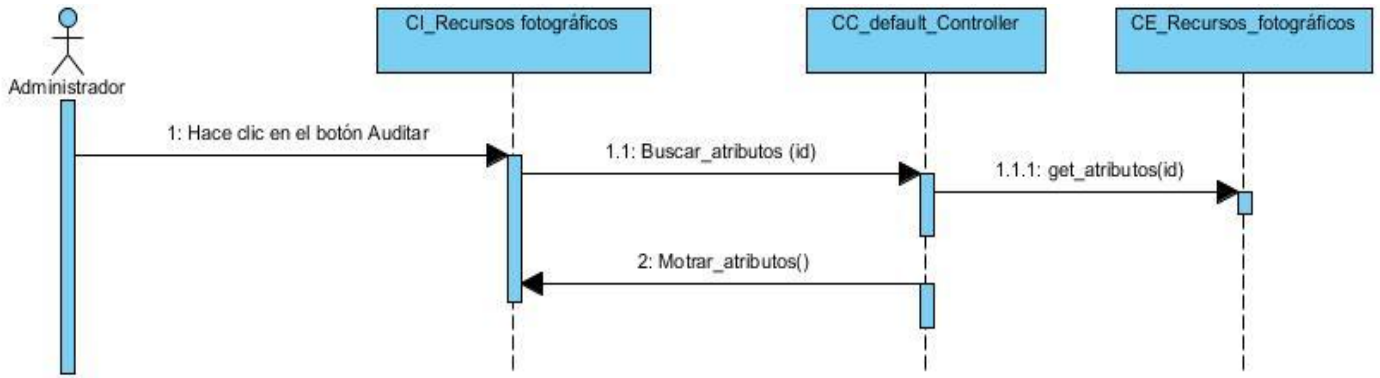


Figura 11. Diagrama de secuencia del CU: Auditar recursos fotográficos.

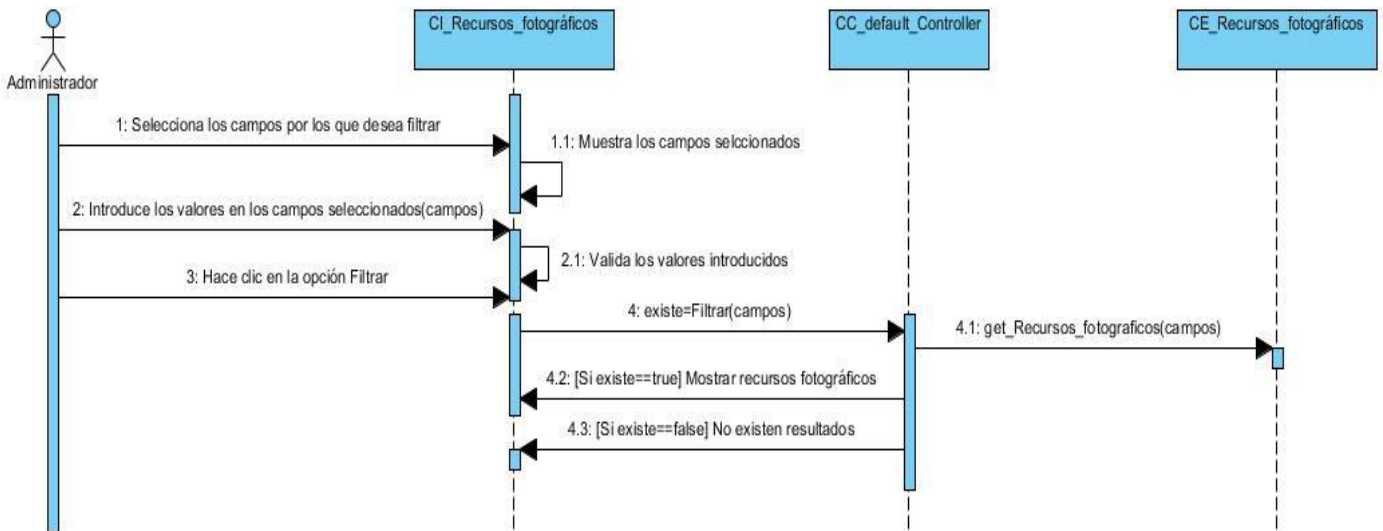


Figura 12. Diagrama de secuencia del CU: Filtrar recursos fotográficos.

3.6 Modelo de despliegue

“El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación, debido a que la distribución del sistema tiene una influencia principal en su diseño.” (Jacobson, 2000).

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

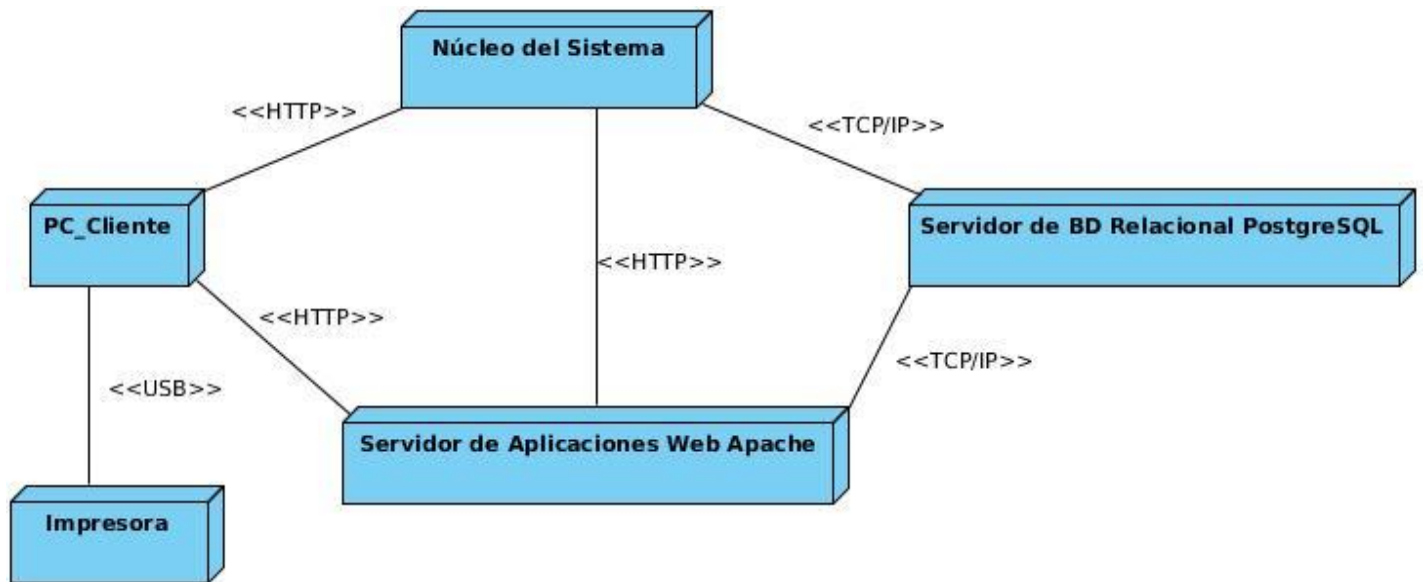


Figura 13. Modelo de despliegue.

3.7 Conclusiones del capítulo

El diseño realizado en el presente capítulo, permitió mostrar una vista interna del funcionamiento del sistema, partiendo del análisis de los requerimientos e incluyendo la arquitectura más adecuada para el módulo.

La definición de patrones de diseño sentaron las bases para la solución de problemas sencillos en la implementación, además de la reutilización y optimización de código. Por último, el modelo de despliegue permitió modelar el hardware a utilizar en la implementación y las relaciones entre sus componentes, representando una correspondencia entre la arquitectura del software y la arquitectura del sistema (*hardware*).

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

Capítulo IV: Implementación y pruebas.

4.1 Introducción

La implementación tiene como propósito primordial desarrollar la arquitectura y el diseño como un todo, para ello, resulta imprescindible: planificar las integraciones de sistema necesarias en cada iteración; distribuir dicho sistema en componentes ejecutables a nodos en el diagrama de despliegue; implementar las clases y subsistemas diseñados y probar los componentes de manera independiente para luego integrarlos al sistema (Jacobson, 2000). Las pruebas, permiten comprobar que el sistema realiza las funciones para las cuales fue diseñado, incidiendo de manera significativa en el control de la calidad del mismo. En el presente capítulo, se genera el diagrama de componentes, reflejando su organización y dependencias. Se reflejan los estándares de codificación a utilizar, así como las clases más significativas. Por último, se hace referencia a las pruebas aplicadas al módulo estadístico incluyendo los resultados arrojados.

4.2 Diagrama de componentes

Los diagramas de componentes son usados para mostrar las relaciones entre los elementos de implementación, las dependencias de compilación de los ficheros de código, relaciones de derivación entre ficheros de código fuente y ficheros que son resultados de la compilación, dependencias entre elementos de implementación y los correspondientes elementos de diseños que son implementados.

A continuación, se muestra el diagrama de componentes perteneciente al Módulo estadístico para el SGD de Imágenes Digitales Ecumene Pyxel v1.0.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

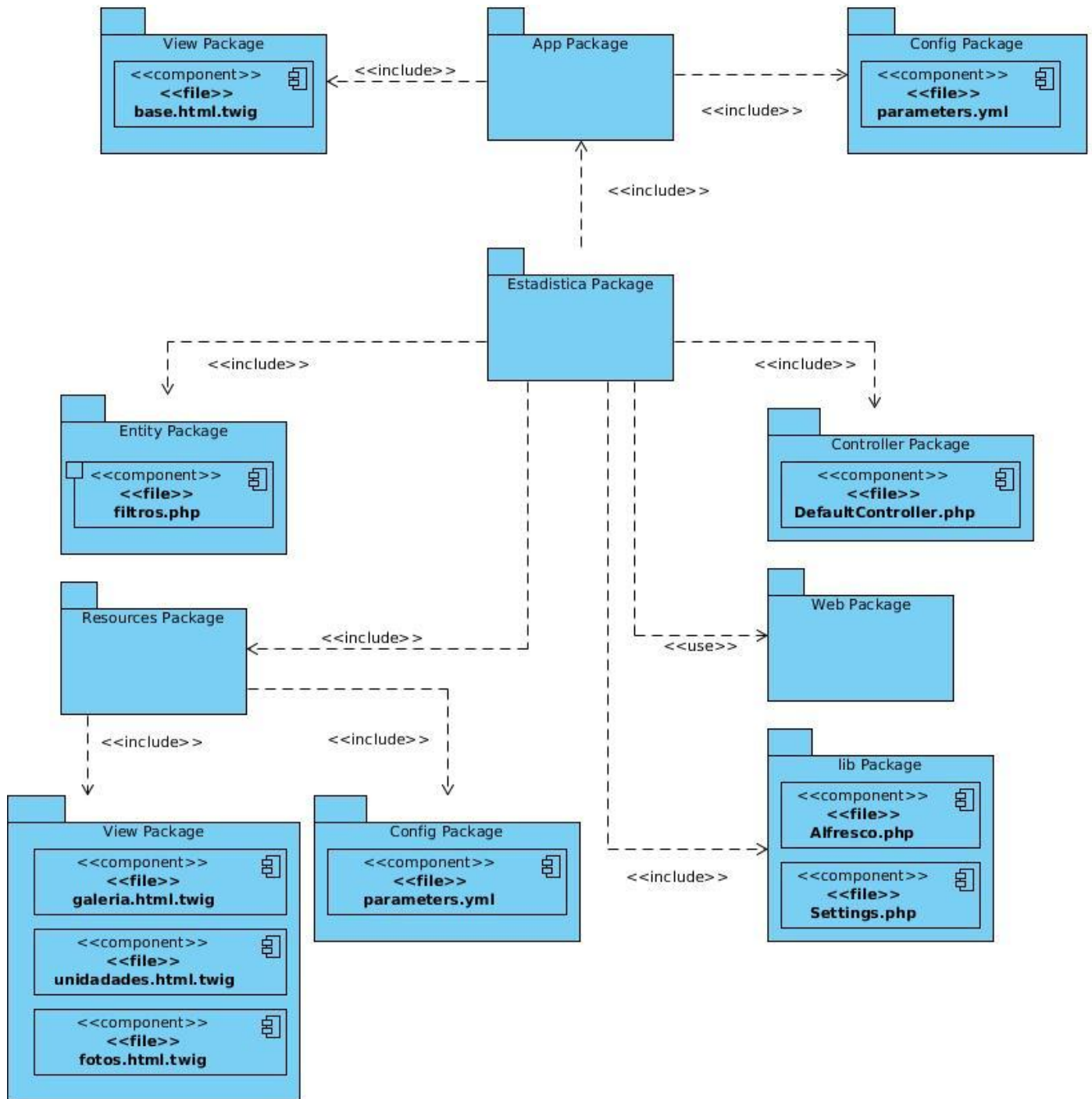


Figura 14. Diagrama de Componentes.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

4.3 Principales aspectos del código fuente del módulo estadístico.

4.3.1 Estándares de codificación.

A la hora de implementar un software, es necesario establecer un estándar de codificación con el objetivo de lograr que el equipo de desarrollo comprenda el código de los demás programadores, además de propiciar que el producto termine con mayor calidad. A continuación, se describen las reglas definidas por el framework utilizado, relacionadas con la nomenclatura y que tributan al estándar de codificación utilizado en el desarrollo del módulo estadístico.

- **Identación³:** la indentación suele estar compuesta por 4 espacios. Las tabulaciones no están permitidas.
- **Tamaño máximo de líneas:** la longitud recomendable para una línea de código es de 140 caracteres. Esto significa que los desarrolladores deberían intentar mantener cada línea de su código por debajo de los 140 caracteres, siempre que sea posible. No obstante, líneas más largas pueden ser aceptables en algunas situaciones. El tamaño máximo de cualquier línea de código PHP es de 160 caracteres.
- **Clases:** los nombres de clases pueden contener sólo caracteres alfanuméricos. Los números están permitidos en los nombres de clase, pero desaconsejados en la mayoría de casos. Las barras bajas (`_`) están permitidas solo como separador de ruta.
Si el nombre de una clase está compuesto por más de una palabra, la primera letra de cada palabra debe aparecer en mayúsculas. Poner en mayúsculas las letras siguientes no está permitido.
- **Clases abstractas:** en general, las clases abstractas siguen las mismas convenciones que las clases, con una regla adicional: Los nombres de las clases abstractas deben acabar con el término, "*Abstract*", y ese término no debe ser precedida por un guión bajo.
- **Interfaces:** los nombres de las interfaces opcionalmente pueden acabar con el término, "Interface", pero el término no debe ser antecedida por un guión bajo. Si bien esta regla no es necesaria, se recomienda encarecidamente su uso, ya que proporciona una buena referencia visual a los desarrolladores, como saber que archivos contienen interfaces en lugar de clases.
- **Nombres de archivo:** para cualquier otro archivo, sólo caracteres alfanuméricos, barras bajas (`_`) y guiones (`-`) están permitidos. Los espacios en blanco están estrictamente prohibidos. Cualquier archivo que contenga código PHP debe terminar con la extensión " `.php` ", con la excepción de los scripts de la vista.

³ Se refiere a la sangría y los espacios en el código.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

- **Funciones y métodos:** los nombres de funciones pueden contener únicamente caracteres alfanuméricos. Los guiones bajos (_) no están permitidos. Los números están permitidos en los nombres de función pero no se aconseja en la mayoría de los casos. Los nombres de funciones deben empezar siempre con una letra minúscula. Cuando un nombre de función consiste en más de una palabra, la primera letra de cada nueva palabra debe estar en mayúsculas. Esto es llamado comúnmente como formato "camelCase". Por norma general, se recomienda la elocuencia. Los nombres de función deben ser lo suficientemente elocuentes como para describir su propósito y comportamiento.
- **Variables:** los nombres de variables pueden contener caracteres alfanuméricos. Las barras bajas (_) no están permitidas. Los números están permitidos en los nombres de variable pero no se aconseja en la mayoría de los casos. Si un nombre está formado por más de una palabra se utiliza el formato "camelCase".
- **Constantes:** las constantes pueden contener tanto caracteres alfanuméricos como barras bajas (_). Los números están permitidos. Todas las letras pertenecientes al nombre de una constante deben aparecer en mayúsculas. Las palabras dentro del nombre de una constante deben
- separarse por barras bajas (_). Las constantes deben ser definidas como miembros de clase con el modificador "const".

4.3.2 Clases más relevantes del módulo estadístico.

El módulo estadístico desarrollado cuenta en su implementación con diferentes clases cada una con objetivos específicos. A continuación se describen las dos más relevantes:

- DefaultController.php: constituye la clase principal, en la que se realiza toda la gestión del filtrado de recursos fotográficos.
- Settings.php: clase que contiene todas las configuraciones del ECM Alfresco.

4.4 Principales pantallas del módulo estadístico.

La interfaz de inicio de cualquier aplicación es uno de los elementos más importantes para que el usuario tenga una experiencia positiva al interactuar con la misma. En la Figura 15, se muestra la interfaz principal del módulo estadístico, la cual está compuesta por un área donde el administrador puede seleccionar el tipo de gráfica y los campos por los que desea filtrar los recursos fotográficos, dándole la posibilidad de guardar dicho filtro creado, para su posterior uso. Muestra además, el resultado de la búsqueda en forma gráfica y tabular. Permite ver una serie de reportes que notifican los recursos fotográficos más leídos, más modificados, últimos modificados y últimos creados. Dichos reportes incluyen al mismo tiempo los usuarios conectados, los últimos usuarios creados y los que no han visitado el sistema.

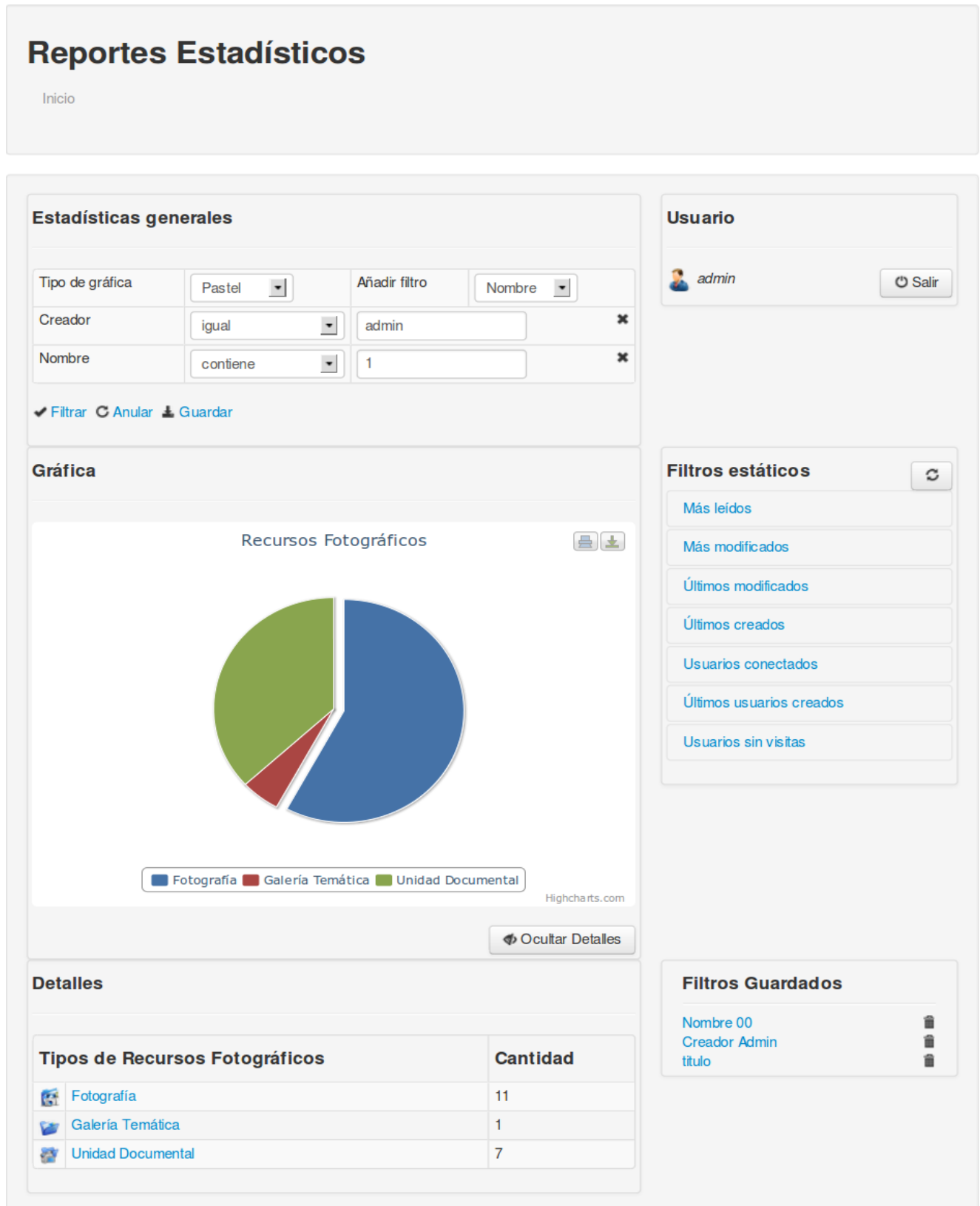


Figura 15. Interfaz principal del módulo estadístico.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

En la Figura 16, se muestra una serie de fotografías como resultado de una búsqueda previa realizada por el administrador. El botón auditar que se muestra en la parte inferior de la imagen permite realizar una auditoría a la misma.

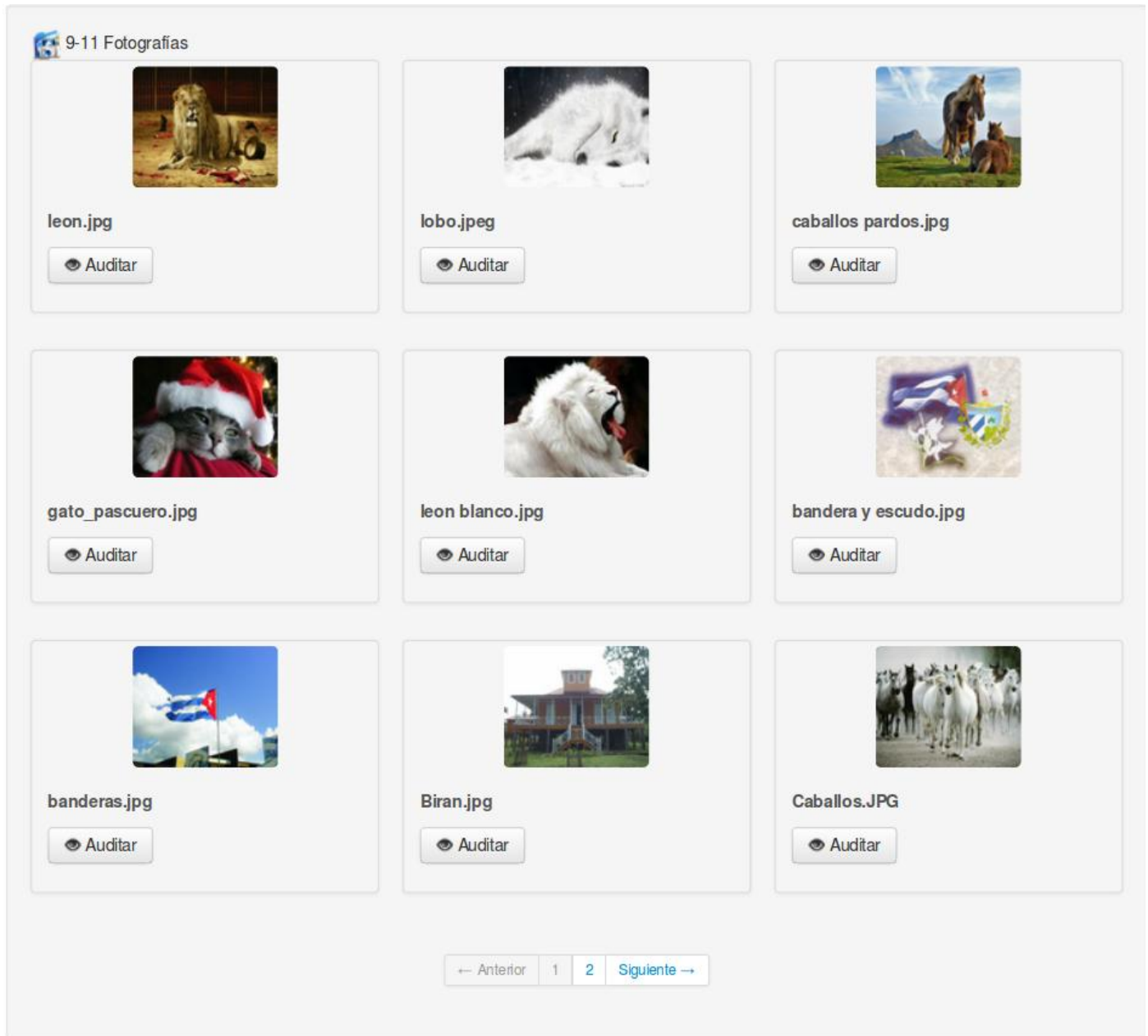


Figura 16. Resultados de una búsqueda de las fotografías.

En la Figura 17, se muestran los datos correspondientes a una fotografía que ha sido auditada.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.



Nombre:	bandera y escudo.jpg
Título:	bandera y escudo.jpg
Descripción:	la bandera y el escudo nacional de Cuba
Autor:	
Creador:	fotografo
Creado:	09-05-2013
Editor:	admin
Modificado:	09-05-2013
Ruta en Alfresco:	/Espacio raíz/Fototeca/Proceso Editorial/Seleccion/bandera y escudo.jpg

Figura 17. Auditoría a una fotografía.

4.5 Validación del módulo estadístico.

Aplicar pruebas a un software, permite verificar el resultado de la implementación, comprobando cada construcción, desde las internas, pasando por las intermedias, así como las versiones finales a entregar a los clientes (Jacobson, 2000). De manera más específica, se refiere a ejecutar un sistema o componente bajo condiciones específicas, observando o almacenando los resultados y realizando además, una evaluación de algún aspecto del mismo. (IEEE, 1990).

Existen diferentes tipos de pruebas, cada uno con objetivos específicos que posibilitan a los probadores, validar el software desde diferentes aristas, a estas pruebas se les conoce como **Pruebas de Sistema**, las cuales tras un estudio realizado, resultaron ser las idóneas a aplicar al módulo desarrollado. Esto se decidió principalmente por las características del módulo, el cual brinda una interfaz al administrador del SGD de Imágenes Digitales Ecumene Pyxel en la que se podrán visualizar un conjunto de reportes estadísticos del uso de los recursos fotográficos almacenados en el mismo, por lo que las pruebas estarán encaminadas a verificar que se integre adecuadamente al SGD. Además, estarán dirigidas a comprobar que el módulo realiza todas las funcionalidades requeridas. A continuación se hace referencia a los tipos de pruebas aplicados.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

4.5.1 Pruebas funcionales. Método y Técnica utilizados.

Las pruebas funcionales, se orientan a verificar si el software desarrollado implementa correctamente las funcionalidades descritas en los requisitos.

Para la realización de dichas pruebas, se hizo uso del método: Pruebas de Caja Negra, debido a que el mismo, permite centrarse en los requisitos funcionales y por medio de los Casos de Prueba, demostrar que: las funcionalidades son operativas, se aceptan las entradas correctamente, se producen las salidas esperadas y se preserva la integridad de la información.

Para la aplicación del método descrito con anterioridad, se empleó la técnica de Partición de Equivalencia, partiendo de la información del tipo (desde el punto de vista del lenguaje de programación) de las variables que constituyen entradas de datos en el módulo, y del conjunto de valores posibles a tomar, dividiendo el mismo, en subconjuntos de valores válidos e inválidos.

A continuación, se exponen algunos de los Diseños de Casos de Pruebas (DCP) elaborados, el resto pueden ser consultados en la planilla Diseño de Casos de Prueba basado en Caso de Uso, la cual se encuentra en el Expediente de Proyecto SGD de Imágenes Digitales Ecumene Pyxel o en el Anexo 2.

Diseños de Casos de Prueba basados en CU.

Tabla 5. Exportar gráfica.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Hacer clic en el botón correspondiente al menú de la gráfica generada.	Muestra una lista de las posibles acciones que puede realizar el usuario sobre la gráfica.	Muestra el listado de opciones que pueden realizarse: <ul style="list-style-type: none">• <i>Download PNG image</i>• <i>Download JPEG image</i>• <i>Download PDF document.</i>• <i>Download SVG vector image</i>	<i>Principal/Menú de la gráfica</i>
EC 1.2 Seleccionar la opción <i>Download PNG image</i>	Selecciona la opción <i>Download PNG image</i> para descargar la imagen de la gráfica en formato .png.	Muestra una ventana que permite abrir o guardar la gráfica en formato PNG en algún directorio del ordenador.	<i>Principal/Menú de la gráfica/Download PNG image.</i>
EC 1.3 Seleccionar la opción <i>Download JPEG image</i>	Selecciona la opción <i>Download JPEG image</i> para descargar la imagen de la gráfica en formato .jpeg.	Muestra una ventana que permite abrir o guardar la gráfica en formato JPEG en algún directorio del ordenador.	<i>Principal/Menú de la gráfica/Download JPEG image</i>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

EC 1.4 Seleccionar la opción <i>Download PDF document</i>	Selecciona la opción <i>Download PDF document</i> para descargar la imagen de la gráfica en formato .pdf.	Muestra una ventana que permite abrir o guardar la gráfica en formato PDF en algún directorio del ordenador.	<i>Principal/Menú de la gráfica/Download PDF document</i>
EC 1.5 Seleccionar la opción <i>Download SVG vector image</i>	Selecciona la opción <i>Download SVG vector image</i> para descargar la imagen de la gráfica en formato .pdf.	Muestra una ventana que permite abrir o guardar la gráfica en formato SVG en algún directorio del ordenador.	<i>Principal/Menú de la gráfica/Download SVG vector image</i>

Tabla 6. DCP: Auditar recurso fotográfico.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar tipo de recurso fotográfico.	El usuario selecciona el tipo de recurso fotográfico que desea auditar: unidad documental, fotografía o galería temática.	Muestra los recursos existentes del tipo seleccionado.	<i>Principal/Recursos fotográficos/Tipo de recurso</i>
EC 1.2 Auditar recurso	El usuario hace clic en el botón <i>Auditar</i> correspondiente al recurso fotográfico que desea auditar.	Muestra en una nueva interfaz todos los atributos correspondientes al recurso auditado con sus respectivos valores.	<i>Principal/Recursos fotográficos/Tipo de recurso/Auditar</i>

Tabla 7. DCP: Graficar resultados.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Selecciona el tipo de gráfica en la que desea visualizar los datos del reporte y hace clic en el botón <i>Filtrar</i> (Ver CP <i>Filtrar recursos fotográficos</i> y CP <i>Seleccionar tipo de gráfica</i>).	Muestra lo datos del reporte en forma del tipo de gráfico seleccionado.	Muestra lo datos del reporte en forma del tipo de gráfico seleccionado.	<i>Principal/Tipo de gráfica/Graficar</i>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

EC 1.2 No selecciona el tipo de gráfica y hace clic en el botón <i>Filtrar</i> .	Muestra los datos del reporte en forma del tipo de gráfico predeterminado (pastel).	Muestra los datos del reporte en forma del tipo de gráfico predeterminado (pastel).	<i>Principal/Graficar</i>
--	---	---	---------------------------

Resultados de las pruebas funcionales.

Para la evaluación de los requisitos del sistema, se realizaron tres iteraciones, en las que se detectaron inicialmente un total de 13 No Conformidades (NC), de ellas 8, 3 y 2 de tipo significativas, no significativas y recomendaciones respectivamente. Luego de darle solución a las mismas, se llevó a cabo una segunda iteración, detectándose 4 NC, de ellas 2, 1 y 1 de tipo significativas, no significativas y recomendaciones respectivamente; por último tras desarrollar una tercera iteración de las pruebas se detectó únicamente una NC de tipo no significativa (*Ver tabla 8*).

Tabla 8. Número de NC detectadas por iteraciones.

Complejidad	1ra Iteración	2da Iteración	3ra Iteración
Significativas	8	2	-
No Significativas	3	1	1
Recomendaciones	2	1	-

Como puede apreciarse en la gráfica de la figura 17, el número de NC de una iteración a otra disminuyó, permitiendo refinar el módulo hasta obtener un producto libre de errores.

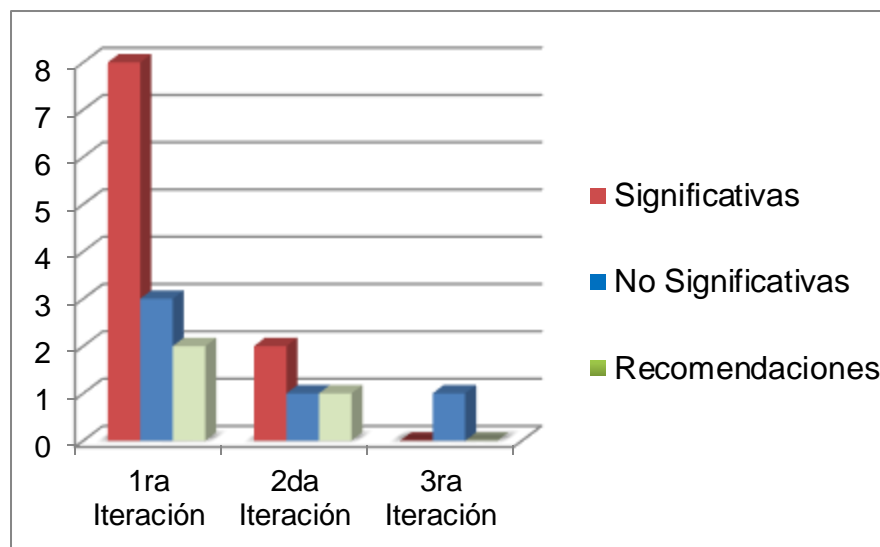


Figura 18. Resultados de las pruebas.

A continuación se hace referencia a algunas NC detectadas durante la aplicación de las pruebas funcionales.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

Tabla 9. Resumen de las NC detectadas en las pruebas funcionales.

Clasificación	Descripción
Significativas	No se valida que los campos de entrada seleccionados a la hora de filtrar los recursos fotográficos estén completos.
	No se valida que los campos de entrada seleccionados a la hora de filtrar los recursos fotográficos, solo admitan los valores correctos.
	Al filtrar los recursos fotográficos por el campo de entrada <i>Nombre</i> , seleccionando la opción <i>Contiene</i> , el sistema no devuelve los valores correctos.
No Significativas	La fuente de los vínculos a otras interfaces posee tamaños de letras diferentes.
	Abuso del uso de mayúsculas al nombrar vínculos, botones y campos de entrada de datos.
	Existen palabras con errores ortográficos, tales como cambio de consonantes, palabras sin tildar, consonantes o vocales repetidas.

4.5.2 Pruebas de unidad e integración.

“Las pruebas de unidad centran el proceso de verificación en la menor unidad del diseño del software: el módulo” (Pressman, 2000). Con la aplicación de estas pruebas, se probó la interfaz del módulo, asegurando que la información fluye de forma adecuada en ambos sentidos (usuario-sistema). Se ejercitaron además, todos los caminos básicos, garantizándose que todas las sentencias se ejecutan al menos una vez.

“Las pruebas de integración son una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es coger los módulos probados en unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño” (Pressman, 2000).

Con la realización de las pruebas anteriormente descritas, se verificó la operación conjunta del módulo desarrollado con cada uno de los componentes SGD de Imágenes Digitales Ecumene Pyxel, haciendo énfasis en su interacción, lo que posibilitó la detección de incoherencias en el funcionamiento de la aplicación.

4.5.3 Pruebas de seguridad.

Las pruebas de seguridad intentan verificar que los mecanismos de protección incorporados en el sistema, sean capaces de protegerlo ante accesos impropios o ilegales (Pressman, 2000). Para llevar a cabo estas pruebas en un primer nivel, se utilizó una lista de chequeo emitida por CALISOFT⁴, la cual abarca 15

⁴ Centro Nacional de Calidad de Software

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

indicadores pertenecientes a cuatro tipos de pruebas de seguridad: pruebas de autorización, pruebas de gestión de sesiones, comprobación del sistema de autenticación y validación de datos. Como resultado de la aplicación de la lista de chequeo antes mencionada, fue posible determinar deficiencias en dos indicadores de impacto crítico en la seguridad, los cuales fueron corregidas luego de ser detectadas. A continuación se muestran dichos indicadores:

- 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.
- Al cerrar la sesión de un usuario y dar clic en el botón del navegador “Atrás” la aplicación vuelve entrar a la sesión autenticada.

4.6 Conclusiones del capítulo.

La definición y utilización del estándar de codificación en la implementación, dotó al código de una mayor legibilidad, uniformidad y limpieza. La utilización de las herramientas y lenguajes identificados en el primer capítulo, contribuyó en gran medida a los resultados obtenidos en esta fase, permitiendo satisfacer cada uno de los requerimientos establecidos en la propuesta de solución. La realización de las pruebas funcionales, unitarias, de integración y seguridad, posibilitó la detección de errores y problemas para su posterior solución en las diferentes etapas de la implementación, así como la ejecución del módulo bajo condiciones específicas.

Conclusiones generales

- El estudio de diferentes Sistemas de Gestión Documental en Cuba y el mundo, así como la generación de reportes estadísticos y las diferentes formas de representar dichos reportes, permitió caracterizar el tratamiento estadístico en los Sistemas de Gestión Documental, concluyendo que, pese a existir interesantes y poderosas propuestas, estas no incluyen módulos dedicados al control y auditoría de recursos fotográficos.
- La generación de los modelos y artefactos correspondientes a las diferentes fases de la metodología empleada, permitió elaborar una propuesta de solución que permitió obtener un mayor entendimiento de los procesos a implementar.
- Se obtuvo un módulo estadístico que constituye un apoyo significativo al administrador del Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0, permitiéndoles establecer un control sobre los recursos fotográficos almacenados en el sistema, así como los usuarios que interactúan con dichos recursos mediante la generación de diferentes reportes.
- La resolución de las no conformidades detectadas durante la aplicación de diferentes tipos de pruebas para comprobar la calidad del sistema, permitió asegurar su correcto funcionamiento y seguridad.

Recomendaciones

- Como primera recomendación se sugiere ampliar las funcionalidades módulo estadístico, y a corto plazo expandir las opciones por las que se pueden filtrar los recursos fotográficos a fin de obtener reportes más precisos.
- Diversificar las formas de graficar los resultados con el objetivo de facilitar al administrador una mayor comprensión de los mismos.

REFERENCIAS BIBLIOGRÁFICAS

Referencias Bibliográficas

Amaro Calderón, Sarah Dámaris y Valverde Rebaza, Jorge Carlos. 2007. *Metodologías Ágiles*. Perú : s.n., 2007.

Alva Zamora, David Enmanuel. 2013. *Metodología de desarrollo de software*. Celaya : s.n., 2013.

Bascón Pantoja, Ernesto. 2004. *El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing*. s.l. : Acta Nova, 2004.

Bustelo. 2000. *Gestión Documental en las Empresas: Una Aproximación Práctica*. 2000.

Campillo Torres, Irima. 2010. *Sistema de Gestión Integral de Documentos de archivo*. 2010.

ISO 15489-1:2006 Information and Documentation - Records Management.

Canós, José H , Letelier, Patricio y Penadé, María Carmen . 2008. *Métodologías Ágiles en el Desarrollo de Software*. Valencia : s.n., 2008.

Gerra Sánchez, Esther;. 2009. *Patrones de Diseño, Patrón estructural Decorator*. 2009.

Ginebra, Naciones Unidas. 2009. *Un agúa para representar estadísticas Parte2 Cómo hacer comprensibles los datos*. 2009.

Highsoft Solutions AS . 2011. Highcharts JS. [En línea] Highsoft Solutions AS , 2011. [Citado el: 5 de abril de 2013.] <http://www.highcharts.com/>.

IBM. 2007. IBM.com. [En línea] IBM , 2007. [Citado el: 30 de mayo de 2013.] <http://www-03.ibm.com/software/products/es/es/enterprise/>.

IEEE. 1990. *Computer Dictionary*. s.l. : Computer Society, 1990.

INEI. 2003. *Herramientas Cases*. Perú : s.n., 2003.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. *El Proceso Unificado de Desarrollo de Software*. Madrid : Addison Wesley, 2000.

Mariñán Pérez, Martín. 2009. *Patrones de Diseño*. 2009.

Medina., Msc. Hilda María Saladrigas. 2005. *Coordenadas cubanas para un fenómeno complejo: Fundamentos para un enfoque teórico- metodológico de la investigación de la Comunicación Organizacional*. La Habana : s.n., 2005.

Mercado Martinic, Beatriz. 2011. *Guía para el Diseño e Implementación de un Sistema de Gestión de Archivos*. Chile : s.n., 2011.

Ministerio de Fomento. 2009. *Gestión Documental (Nivel 3)*. 2009.

Mosquera, Jenny C. 2010. *DISEÑO RECOMENDADO DE UN MODELO DE GESTIÓN DOCUMENTAL PARA LA AUTOMATIZACIÓN DE EXPEDIENTES DE CLIENTES EN UNA INSTITUCIÓN FINANCIERA*.

REFERENCIAS BIBLIOGRÁFICAS

Caracas : s.n., 2010.

Nuxeo. 2000. nuxeo/documentation center. [En línea] 2000. [Citado el: 7 de febrero de 2013.] <http://doc.nuxeo.com/display/USERDOC/Document%27s+history>.

Pressman, Roger S. 2000. Ingeniería del Software un Enfoque Práctico. s.l. : Mc Graw Hill, 2000.

Recabarren, Matías. 2009. *Ingeniería de Software, Diseño Patrones GRASP*. Chile : s.n., 2009.

Sánchez Crespo, Gonzalo; Benitez, Vicente. 2008. *SOBRE LA DEFINICIÓN DE ESTADÍSTICA*. 2008.

Tuya, Javier. 2009. *Nuevo estándar de pruebas del software ISO IEC 29119*. Universidad de Oviedo. Oviedo : s.n., 2009.

Unidad Educativa Colegio Loyola–Gumilla. 2001. *decargar_conceptos-*. Guayana : s.n., 2001.

Walne, Peter. 1988. *Dictionary of archival terminology. English and French, with equivalents in Dutch, German, Italian, Russian and Spanish. Dictionnaire de terminologie archivistique*. 1988.

Bibliografía consultada

- Alfresco Software. 2012.** Alfresco. [En línea] Alfresco Software, 2012. <https://www.alfresco.com/es/recorrido-por-alfresco>.
- Alonso, Yulainne. 2012.** *Configuración de la Metodología OpenUP v1.0*. La Habana : s.n., 2012.
- Alva Zamora, David Enmanuel. 2013.** *Metodología de desarrollo de software*. Celaya : s.n., 2013.
- Amaro Calderón, Sarah Dámaris y Valverde Rebaza, Jorge Carlos. 2007.** *Metodologías Ágiles*. Perú : s.n., 2007.
- Arranz Otero, Luis José; Martín Galán, Bonifacio. 2006.** Software Libre: ¿una alternativa de evolución tecnológica para la gestión y el servicio público en los archivos?). *El archivo, ¿un servicio público?* s.l. : Dialnet, 2006.
- Bascón Pantoja, Ernesto. 2004.** *El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing*. s.l. : Acta Nova, 2004.
- Blank, Isabel, Herrera, Larissa y Ortiz, Miguel. 2005.** *Pruebas de Funcionalidad*. 2005.
- Mercado Martinic, Beatriz. 2011.** *Guía para el Diseño e Implementación de un Sistema de Gestión de Archivos*. Chile : s.n., 2011.
- Canós, José H , Letelier, Patricio y Penadé, María Carmen . 2008.** *Métodologías Ágiles en el Desarrollo de Software*. Valencia : s.n., 2008.
- Carrillo, German. 2008.** GeoTux, Soluciones Informáticas Libres. [En línea] 2 de Diciembre de 2008. [Citado el: 25 de Noviembre de 2012.] <http://geotux.tuxfamily.org/index.php/fr/component/k2/item/184-graficas-estadisticas-de-geotux-con-la-libreria-open-flash-chart>.
- Centro Ideoinformática. 2012.** *Base Tecnológica*. La Habana : s.n., 2012.
- CMS-Latino.com*. (s.f.). Recuperado el 25 de enero de 2013, de <http://www.cms-spain.com/articulo/11315>
- S.C.A., Y. (s.f.). *Athento*. Recuperado el 25 de enero de 2013, de <http://www.athento.com>
- Cisneros, Cirley; Tupe, Juan. 2008.** *Evaluación y selección de Framework de Desarrollo PHP*. 2008.
- Comunicación, Instituto Nacional de Tecnologías de la. 2012.** *Gestión de Sesiones Web: Ataques y medidas de seguridad*. España : s.n., 2012.
- Comunidad Netbean. 2012.** NetBeans. [En línea] 2012. [Citado el: 5 de Noviembre de 2012.] <http://netbeans.org/>.
- Desoft. 2012.** Desoft. [En línea] 2012. [Citado el: 29 de Octubre de 2012.] <http://www.desoft.cu/Productos1/AvilaDoc/tabid/430/Default.aspx>.
- Dr. C. Israel A.** *Propuesta de clasificación de las herramientas - software para la gestión del conocimiento*.

BIBLIOGRAFÍA CONSULTADA

- Fugueras, Alberch. 2003.** *Los archivos entre la memoria histórica y la sociedad del conocimiento.* Barcelona : UOC, 2003.
- GENBETA:dev. 2012.** GENBETA. [En línea] 16 de Junio de 2012. [Citado el: 15 de Noviembre de 2012.] <http://www.genbetadev.com/frameworks/bootstrap>.
- Ginebra, Naciones Unidas. 2009.** *Un agüía para representar estadísticas Parte2 Cómo hacer comprensibles los datos.* 2009.
- Gerra Sánchez, Esther;. 2009.** *Patrones de Diseño, Patrón estructural Decorator.* 2009.
- Highsoft Solutions AS . 2011.** Highcharts JS. [En línea] Highsoft Solutions AS , 2011. [Citado el: 5 de abril de 2013.] <http://www.highcharts.com/>.
- IBM. 2007.** IBM.com. [En línea] IBM , 2007. [Citado el: 30 de mayo de 2013.] <http://www-03.ibm.com/software/products/es/es/enterprise/>.
- IEEE. 1990.** *Computer Dictionary.* s.l. : Computer Society, 1990.
- INEI. 2003.** *Herramientas Cases.* Perú : s.n., 2003.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison Wesley, 2000.
- León, Eduardo. 2010.** *Tutorial Visual Paradigm for UML.* 2010.
- Lira Turriza, José Luis. 2010.** *Algoritmos y lenguajes de programación.* Campeche : s.n., 2010.
- Mariñán Pérez, Martín. 2009.** *Patrones de Diseño.* 2009.
- Ministerio de Fomento. 2009.** *Gestión Documental (Nivel 3).* 2009.
- Mujica, Mena. 2005.** *M. Archivos históricos, objetivos y funciones. Gestión documental y organización de archivos.* La Habana : s.n., 2005.
- Nuxeo. 2012.** Nuxeo Content Magnagement Platform for Business Aplications. [En línea] 2012. <http://www.recbib.es/book/nuxeo>.
- Nuxeo. 2000.** nuxeo/documentation center. [En línea] 2000. [Citado el: 7 de febrero de 2013.] <http://doc.nuxeo.com/display/USERDOC/Document%27s+history>.
- Núñez, Paula; Núñez, Yiny Govín. 2005.** 2, La Habana : s.n., 2005, Vol. 13.
- Ponjuan, Gloria Dante. 2005.** 3, La Habana : Ciencias de la Información, 2005, Vol. 36. ISSN 0864-4659.
- Pressman, Roger S. 2000.** *Ingeniería del Software, un Enfoque Práctico.* s.l. : Mc Graw Hill, 2000.
- Pressman, Roger. 2005.** *ingeniería del Software, un Enfoque práctico, Sexta edición.* s.l. : McGraw Hill interamericana, 2005.
- Procedimiento para la realización de pruebas de unidad de software orientado porobjetos a nivel de clases.*
- Castro, Alba. 2011.** 2, Medellín, Colombia : s.n., 2 de Julio de 2011, Vol. 8. 1657-7663.

BIBLIOGRAFÍA CONSULTADA

Recabarren, Matías. 2009. *Ingeniería de Software, Diseño Patrones GRASP*. Chile : s.n., 2009.

S.C.A, YSENGINEERS. Athento. [En línea] [Citado el: 25 de enero de 2013.] <http://www.athento.com>.

SourceForge.net. SourceForge.net. *SourceForge.net*. [En línea] [Citado el: 5 de abril de 2013.] <http://pchart.sourceforge.net/>.

Tuya, Javier. 2009. *Nuevo estándar de pruebas del software ISO IEC 29119*. Universidad de Oviedo. Oviedo : s.n., 2009.

Unidad Educativa Colegio Loyola–Gumilla. 2001. *decargar_conceptos-*. Guayana : s.n., 2001.

Vega, A. M.; Rodríguez, K. *Análisis y Diseño del módulo Visor de un sistema para la generación de reportes*. La Habana, 2009.

Walne, Peter. 1988. *Dictionary of archival terminology. English and French, with equivalents in Dutch, German, Italian, Russian and Spanish. Dictionnaire de terminologie archivistique*. 1988.

Glosario de términos

- **Gestión documental:** Conjunto de normas, técnicas y prácticas usadas para administrar el flujo de documentos de todo tipo en una organización.
- **Galería temática:** agrupación lógica de imágenes que se encuentran almacenadas en las colecciones Difusión y Conservada.
- **Librería de graficado:** Software que genera imágenes en base a modelos matemáticos y patrones de iluminación y texturas.
- **Módulo estadístico:** Permite generar reportes estadísticos para la toma de decisiones.
- **Reporte:** En el ámbito de la informática, los reportes son informes que organizan y exhiben la información contenida en una base de datos.
- **SGD de Imágenes de Digitales:** Sistema de archivo, digitalización y almacenaje informatizado de imágenes digitales, tanto obtenidos de documentos originalmente soportados en papel como de documentos que ya se encuentran en formato digital.

Anexo 1. Descripción de CU

Tabla 10. CU Exportar gráfica.

Objetivo	Exportar a un determinado formato la gráfica resultante del reporte dinámico generado.	
Actores	Administrador.	
Resumen	El CU se inicia cuando el Administrador desea exportar la gráfica resultante del reporte generado, el sistema permite exportar dicha gráfica, finaliza así el CU.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	El usuario debe estar autenticado en el sistema. Debe haberse ejecutado el CU Graficar resultado.	
Postcondiciones	Se exporta la gráfica.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1	Hace clic en el botón correspondiente al menú de la gráfica generada.	1.1 Muestra el listado de opciones que pueden realizarse: <ul style="list-style-type: none"> • Dowload PNG image • Dowload JPEG image • Dowload PDF document • Dowload SVG vector image
2	Selecciona la opción <i>Dowload PNG image</i>	2.1 Muestra una ventana que permite abrir o guardar la gráfica en formato PNG en algún directorio del ordenador.
3		Termina el CU.
Flujos alternos		
Nº Evento <Selecciona la opción Dowload JPEG image>		
	Actor	Sistema
2	Selecciona la opción <i>Dowload JPEG image</i>	2.1 Muestra una ventana que permite abrir o guardar la gráfica en formato JPEG en algún directorio del ordenador.
3		Termina el CU.
Flujos alternos		
Nº Evento <Selecciona la opción Dowload PDF document>		
	Actor	Sistema
2	Selecciona la opción <i>Dowload PDF document</i>	2.1 Muestra una ventana que permite abrir o guardar la gráfica en formato PDF en algún directorio del ordenador.

3		Termina el CU.
Flujos alternos		
Nº Evento <Selecciona la opción Dowload SVG vector image >		
	Actor	Sistema
2	Selecciona la opción <i>Dowload SVG vector image</i>	2.1 Muestra una ventana que permite abrir o guardar la gráfica en formato SVG en algún directorio del ordenador.
3		Termina el CU.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 11. CU Eliminar filtro dinámico.

Objetivo	Eliminar un filtro dinámico creado previamente.	
Actores	Administrador.	
Resumen	El CU se inicia cuando el Administrador desea eliminar un filtro dinámico creado con anterioridad y el sistema lo elimina del listado de filtros dinámicos guardados, finalizando así el CU.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	El usuario debe estar autenticado en el sistema.	
Postcondiciones	Se genera el reporte.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1	Hace clic en el vínculo <i>Eliminar</i> correspondiente al filtro dinámico que desea eliminar.	1.1 Muestra una ventana de confirmación con el mensaje “¿Está seguro que desea eliminar el filtro”?
2	Hace clic en el botón <i>Aceptar</i> .	2.1 Elimina el filtro dinámico.
3		Termina el CU
Flujos alternos		
Nº Evento <Condición que dio lugar a la extensión>		

	Actor	Sistema
2	Hace clic en el botón <i>Cancelar</i> .	2.1 No elimina el filtro dinámico.
3		Termina el CU.
Relaciones		
	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 12. CU Mostrar últimos usuarios creados

Objetivo	Mostrar un reporte de los usuarios creados en el sistema.	
Actores	Administrador.	
Resumen	El CU se inicia cuando el Administrador desea ver un reporte de los últimos usuarios creados en el sistema, el sistema muestra el reporte con los usuarios creados, finaliza el CU.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	El usuario debe estar autenticado en el sistema.	
Postcondiciones	Se genera el reporte	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1	Selecciona la opción <i>Ultimos usuarios creados</i> .	1.1 Muestra un reporte con los últimos usuarios creados en el sistema. De cada uno muestra: <ul style="list-style-type: none"> • Usuario • Nombre y apellidos • Fecha de creación.
2		Termina el CU.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Anexo 2. Diseños de CP

Tabla 13. DCP: Seleccionar tipo de gráfica.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar la opción <i>Tipo de gráfica</i> .	Selecciona la opción <i>Tipo de gráfica</i> .	Muestra los tipos de gráfica en la que se pueden mostrar los reportes: <ul style="list-style-type: none"> • Pastel • Barras 	<i>Principal/Tipo de gráfica.</i>
EC 1.2 Selecciona el tipo de gráfica de <i>Barras</i>	Selecciona el tipo de gráfica de <i>Barras</i>	Muestra los resultados según la gráfica seleccionada por el usuario	<i>Principal/Tipo de gráfica/Barras</i>
EC 1.3 Selecciona el tipo de gráfica <i>Pastel</i>	Selecciona el tipo de gráfica de <i>Pastel</i>	Muestra los resultados según la gráfica seleccionada por el usuario	<i>Principal/Tipo de gráfica/Pastel</i>

Tabla 14. DCP: Eliminar filtro dinámico.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Hace clic en el vínculo <i>Eliminar</i> .	Hace clic en el vínculo <i>Eliminar</i> para eliminar los filtros dinámicos guardados.	Muestra una ventana de confirmación con el mensaje “¿Está seguro que desea eliminar el filtro?”	<i>Principal/Eliminar</i>
EC 1.2 Hacer clic en el botón <i>Aceptar</i> .	El usuario hace clic en el botón <i>Aceptar</i> , si realmente desea eliminar el filtro dinámico seleccionado.	Elimina el filtro dinámico.	<i>Principal/Eliminar/Aceptar</i>
EC 1.3 Hace clic en el botón <i>Cancelar</i> .	El usuario hace clic en el botón <i>Cancelar</i> , si no desea eliminar el filtro dinámico seleccionado.	No elimina el filtro dinámico.	<i>Principal/Eliminar/Cancelar</i>