

Universidad de las Ciencias Informáticas

Facultad 1



**Subsistema de administración 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.

Autores: Marisleidy Gonzalez Licourt

Alfredo Tur Vedey

Tutores: Ing. Elieyis Gómez Padrón

Ing. Susana Gonce Fernández

La Habana, junio de 2013

Declaración de autoría

Declaración de Autoría

Declaramos ser los únicos autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma; así como al Centro de Ideoinformática para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autora: Marisleidy González Licourt

Autor: Alfredo Tur Vedey

Tutor: Ing Susana Gonca Fernández

Tutor: Ing Elieyis Gómez Padrón



Datos de contacto

Datos de contactos

Tutores:

Ing. Elieyis Gómez Padrón

Especialista General del departamento de Soluciones Informáticas para Internet del centro de producción CIDI, Facultad 1.

Email: epadron@uci.cu Teléfono: 837-3186

Ing. Susana Gonce Fernández

Asesor Técnico Docente, Área Dir. Investigaciones.

Email: sgonce@uci.cu Teléfono: 835-8868

Universidad de las Ciencias Informáticas, Habana, Cuba.



Pensamiento



"Si avanzo, seguidme; si me detengo, empujadme; si retrocedo, matadme"

Ernesto Che Guevara.



Agradecimientos

Agradecimientos Marisleidy

En primer lugar a mis padres por darme la confianza y todo el amor necesario durante estos años, por su dedicación y apoyo en todas las situaciones. A mi familia que me ha apoyado muchísimo incondicionalmente en especial a mi tía Adela que siempre estuvo en los momentos más difíciles y confió siempre en mí, a mis primas Yaniuris y Tania le agradezco mucho para ellos mi eterna gratitud.

A mis amistades que he tenido el placer de conocer en esta maravillosa etapa de mi vida en especial mis compañeros de aula que me han acompañado durante estos cinco años y para alguien que ha estado en todos los momentos que la he necesitado mi amiga Leidys, para todos ellos muchas gracias. No puedo dejar de mencionar a personas importantes para mí, mi compañero de tesis Alfredo para él es el máximo reconocimiento, quien me dio su apoyo incondicional en el peor momento. Todos aquellos que hicieron posible de una forma u otra que yo llegara hasta aquí muchas gracias, a Ángel, a Lachi, Leonardo y a mis tutoras Elieyis y Susana por estar ahí cuando las necesitábamos.

En general a todos aquellos que aportaron su granito de arena para la realización de este trabajo.

Agradecimientos

Agradecimientos Alfredo

Les agradezco fundamentalmente a mis padres, por todo el amor, el apoyo de siempre y su confianza en mí, por ayudarme a ser lo que hoy soy, y por mantener la fe de que siempre se puede un poco más.

A mi hermanita que es el otro pedacito de mi vida, gracias por tu amor, tu confianza y tu compañía.

A mis abuelitos que son el amor de mi vida.

A mis tutoras Elieyis Gómez Padrón y Susana Gonca Fernández, por haber sido el motor impulsor en mi trabajo, les agradezco su constancia y empeño, su ayuda incondicional, su esmero y dedicación.

A todos mis amigos y las personas que de modo general permitieron que este día fuera posible y que mi sueño se hiciera realidad. En especial a:

Compañeros:

Joel Ayata Escalona, Lázaro Ricardo Rodríguez Placeres, Leonardo Rojas Rodríguez

Hermanos:

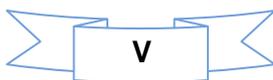
Ángel Echevevarria Zamora, Osvaldo José Aguilera, Oscar Emilio Verdecia Reyes

Hermanas:

Mailen Vaillant Leon, Arlén Herrera Bacallao, Dayamí Palama, Marisleidy González, Elizabeth Vento Caballero, Dianeyis Rivero Chacón, Dunia Espinosa López

Alguien muy especial para mí:

Katisleidy Soto Pavón



Dedicatoria

Dedico esta tesis a mi mamá, mi papá, mi prima Noraiza y toda mi familia que siempre me han apoyado para llegar hasta donde llegué. En especial se la dedico a una persona que no tuvo la oportunidad de verme alcanzar un logro como este a mi Abuela que estoy segura de que esté donde esté estará orgullosa de mí.

Marisleidy

Le dedico el presente trabajo de Diploma a mis padres que sin ellos no sería quien soy ahora, a mi hermana que es el otro pedazo de mi vida, a mis abuelos que están aquí conmigo que son mis amores y a los que no están que siempre me están apoyando y que en cualquier parte que se encuentren sé que estarán orgullosos de mí, a todos mis amigos y compañeros de aula que me brindaron su apoyo de una manera u otra incondicionalmente, a todos muchas gracias.

Alfredo

Resumen

En la Universidad de las Ciencias Informáticas se desarrolló un sistema de gestión documental para las imágenes digitales denominado Ecumene Pyxel con el objetivo de garantizar una correcta organización y documentación de los recursos fotográficos. Este sistema posee una interfaz Web amigable para los usuarios pero aún presenta determinadas desventajas que impiden que las tareas de configuración y administración se realicen con facilidad. Teniendo en cuenta estos inconvenientes se desarrolló un subsistema de administración que permita la gestión de los contenidos de forma rápida y sencilla además de poseer una interfaz de fácil uso para los usuarios. Para su desarrollo se utilizaron herramientas y tecnologías apropiadas establecidas por el Centro de Ideoinformática. Para guiar el proceso de análisis, diseño e implementación y prueba del sistema se utilizó como metodología ágil OpenUp, como tecnología se seleccionaron las siguientes: como marco de trabajo el framework Symfony2, el Visual Paradigm como herramienta para el modelado y como sistema gestor de base de datos PostgreSQL. Se generaron los artefactos que dicta la metodología, fueron implementadas las funcionalidades y realizadas sus correspondientes pruebas. Con el sistema obtenido se garantiza una mejor configuración y administración de los contenidos en Ecumene Pyxel.

Palabras claves: ECM Alfresco, gestión documental, subsistema.

Índice

<i>Introducción</i>	13
1.1 Gestión Documental	18
1.2 Sistema de Gestión Documental.....	18
1.3 Administrador de contenidos empresariales.....	19
1.3.1 Administrador de contenidos empresariales Alfresco.....	20
1.4 Interfaces web desarrolladas para Alfresco	21
1.4.1 Alfresco Explorer	21
1.4.2 lfresco	22
1.4.3 Opsoro.....	23
1.4.4 DoCASU	24
1.4.5 eXcriba.....	25
1.5 Tecnologías asociadas al desarrollo del sistema.....	26
1.5.1 Metodología de Desarrollo	26
1.5.2 Metodología de desarrollo OpenUp	26
1.5.3 Marco de trabajo a utilizar	28
1.5.4 Entornos integrados de desarrollo	29
1.5.5 Sistema Gestor de Bases de Datos.....	30
1.5.6 Herramientas para el Modelado. Visual Paradigm.	30
1.6 Lenguajes utilizados	31
1.6.1 Lenguaje de Modelado. UML	31
1.6.2 PHP 5.3.....	31
1.6.3 JavaScript 1.4	32
1.6.4 CSS 3.....	32
1.6.5 XML.....	33
1.8 Conclusiones parciales	34
<i>Capítulo 2 Propuesta de solución para el SGDID Ecumene Pyxel</i>	35
2.1 Propuesta de solución.....	35
2.2 Modelo de dominio	35
2.3 Listado de los requisitos funcionales y no funcionales	38
2.4 Definición de casos de uso del sistema	42
2.4.1 Descripción de los casos de usos del sistema.....	42
2.5 Propuesta de arquitectura.....	51

Índice

2.5.1 Patrones Arquitectónicos	54
2.5.2 Patrones de Diseño.....	54
2.5.3 Patrones GRASP.....	55
2.5.4 Patrones GOF	55
2.6 Diseño de la interfaz de administración para el SGDID Ecumene Pyxel.....	56
2.6.1 Diagrama de clases del diseño.....	56
2.6.2 Diagramas de Secuencia.....	58
2.7 Diagrama de despliegue.....	60
2.7.2 Descripción de los componentes.....	61
2.8 Conclusiones parciales	62
<i>Capítulo 3 Implementación y Prueba.....</i>	<i>63</i>
3.1 Estructura de implementación del subsistema de administración.....	63
3.1.1 Descripción de los componentes	64
3.2 Estándares de codificación.....	65
3.3 Interfaces principales de la aplicación.....	65
3.4 Pruebas de software	67
3.4.1 Métodos de pruebas.....	67
Criterios de Caja Negra	67
3.4.1 Estrategia de prueba	68
3.4.2 Pruebas funcionales	68
3.4.2 Pruebas de seguridad	73
3.5 Conclusiones parciales	74
Anexo1 Diagrama de clases del diseño	80
Anexo2 Diagrama de secuencia.....	84
Anexo 3 Diagrama de paquetes	87
Anexo4 Prototipos no funcionales	89
Anexo5 Tablas de casos de prueba.....	93
<i>Glosario de Términos</i>	¡Error! Marcador no definido.

Índice de ilustraciones

Figura 1 Esquema de la separación de los contenidos y su presentación	33
Figura 2 Modelo de dominio	37
Figura 3 Diagrama de casos de uso del sistema	42
Figura 4 Arquitectura del ECM Alfresco	52
Figura 5 Arquitectura del Subsistema de Administración	53
Figura 6 Diagrama de clases del diseño Gestionar usuario	58
Figura 7 Diagrama de secuencia Autenticar Usuario	59
Figura 8 Diagrama de secuencia Adicionar Usuario	60
Figura 9 Representación del modelo de desde la aplicación	61
Figura 10 Representación del diagrama de componentes del subsistema de administración	64
Figura 11 Autenticación en el sistema	66
Figura 12 Espacio de usuario	66
Figura 14 Diagrama de clases del diseño gestionar grupo	80
Figura 15 Diagrama de clases del diseño Autenticar usuario	81
Figura 16 Diagrama de clase del diseño Gestionar Contenido	81
Figura 17 Diagrama de clase del diseño gestionar Cuadro de clasificación	82
Figura 18 Diagrama de clase del diseño Gestionar Reglas	83
Figura 19 Diagrama de clase del diseño Gestionar taxonomía	83
Figura 20 Diagrama de secuencia Editar Usuario	84
Figura 21 Diagrama de secuencia Cambiar contraseña	86
Figura 22 Diagrama de secuencia Adicionar taxonomía	86
Figura 23 Diagrama de paquete del bundle usuario	87
Figura 24 Diagrama que representa el Paquete de vistas del usuario	88
Figura 25 Prototipo de interfaz gestionar usuario	89
Figura 26 Prototipo de interfaz gestionar de cuadro de clasificación	90
Figura 27 Prototipo de interfaz editar usuario	91

Índice de tablas

Tabla 1	Listado de requisitos funcionales	39
Tabla 2	Listado de Requisitos no funcionales.	41
Tabla 3	Actores que interactúan en el sistema.	41
Tabla 4	Descripción del CU Autenticar usuario	43
Tabla 5	Descripción del CU Gestionar usuario	46
Tabla 6	Descripción del CU Gestionar grupo	48
Tabla 7	Descripción del CU Gestionar clasificación.	50
Tabla 8	Descripción del CU Realizar búsqueda simple	50
Tabla 9	Descripción del CU Realizar búsqueda avanzada.	51
Tabla 10	Descripción del flujo central del Caso de Uso Gestionar Usuario	69
Tabla 11	Caso de prueba Autenticar usuario	70
Tabla 12	Descripción del Caso de prueba Adicionar Usuario	72
Tabla 13	Número de no conformidades por iteraciones	72
Tabla 14	Principales no conformidades detectadas	73
Tabla 15	Descripción del Caso de prueba Eliminar Usuario	93
Tabla 16	Descripción del Caso de prueba Cambiar contraseña	93
Tabla 17	Caso de prueba Adicionar grupo	94

Introducción

La información se considera un recurso valioso para la sociedad. Desde tiempos muy remotos se ha valorado la necesidad de conservarla y documentarla, pues se ha multiplicado vertiginosamente su volumen. En aras de preservar este recurso en el tiempo el hombre ha desarrollado mecanismos que garanticen su gestión. Actualmente existen organizaciones que trabajan con grandes volúmenes de información, almacenándolas bajo soportes no digitales, ya sea en carpetas o archivos que ocupan espacio en las oficinas y se deterioran con rapidez.

En Cuba los medios de comunicación, específicamente los medios de prensa, han estado inmersos en un profundo y novedoso proceso de cambio tecnológico que garantiza el mejoramiento y optimización de las tareas productivas. Los medios de prensa procesan a diario una gran cantidad de imágenes debido al enorme flujo de noticias que circulan por las redes. Las que constituyen materiales importantes para la representación de la información, pero manipularlas, consultarlas y archivarlas frecuentemente resulta un proceso tedioso.

La Gestión Documental (GD) fue el primer paso para dar inicio a la era de la digitalización de la prensa y en correspondencia con el nivel de desarrollo alcanzado se han creado los Sistemas de Gestión Documental (SGD). Las facilidades que ofrecen estos métodos permiten a los trabajadores realizar su labor con mayor rapidez y calidad.

Como parte del Programa de Informatización para la Prensa Cubana desde la Universidad de las Ciencias Informáticas (UCI) se ha impulsado el estudio y desarrollo de un Sistema de Gestión Documental de Imágenes Digitales denominado SGDID Ecumene Pyxel. Este sistema consta de un núcleo basado en las tecnologías que provee el Administrador de Contenidos Empresariales (ECM, del Inglés *Enterprise Content Management*) Alfresco y una interfaz Web desarrollada con el *framework* Symfony. El núcleo y la interfaz se integran mediante una capa de servicios y en conjunto permiten gestionar, documentar, identificar, describir, catalogar, recuperar, y publicar los recursos fotográficos, además de generar automáticamente diferentes perfiles a partir de la imagen digital adicionada originalmente al sistema. En esta aplicación interactúan varios usuarios que realizan diferentes procesos y los administradores del sistema son los responsables de gestionar los usuarios, grupos, roles y permisos que estos tienen sobre los espacios y contenidos, por lo que su labor resulta imprescindible para que los procesos fluyan de la

manera esperada. Sin embargo, el proceso de administración no se incluyó en la interfaz web de Ecumene Pyxel de manera que la mayoría de las tareas administrativas se realizan en el núcleo a través de la interfaz Web que provee el Alfresco, la cual presenta los siguientes inconvenientes:

- ✓ Tiene una curva de aprendizaje muy elevada.
- ✓ Es poco intuitiva para cualquier usuario.
- ✓ La representación de la información no satisface todas las necesidades del usuario.
- ✓ Las tareas de configuración y administración requieren de un número considerable de pasos por lo que todo el proceso se vuelve tedioso y no resulta sencillo su entendimiento.

Estos inconvenientes traen como consecuencia:

- ✓ Desmotivación por parte de los administradores para trabajar con el sistema.
- ✓ Dificultades en las tareas de configuración y administración del sistema.
- ✓ Errores provocados por el olvido de alguno de los pasos a la hora de realizar determinada acción.
- ✓ Además, la gestión de las taxonomías deben hacerla directamente en la base de datos de la interfaz web de Ecumene Pyxel mediante consultas DQL.
- ✓ Como resultado final el usuario Administrador tiene que hacer uso de dos aplicaciones a la vez para ejecutar sus tareas administrativas (gestionar grupos, gestionar usuario, gestionar cuadro de clasificación).

Teniendo en cuenta lo antes mencionado se identificó el siguiente **problema de investigación** ¿Cómo facilitar el proceso de administración y configuración del sistema de gestión documental de imágenes digitales Ecumene Pyxel V1.0?

El **objeto de estudio** se centra en la administración de sistemas de gestión documental.

El **campo de acción** se basa en la administración del sistema de gestión documental de imágenes digitales Ecumene Pyxel V1.0.

Como **objetivo general** se plantea desarrollar el subsistema administrativo para facilitar el proceso de

administración y configuración del sistema de gestión documental de imágenes digitales Ecumene Pyxel v1.0.

Dando lugar a la siguiente **idea a defender**: El desarrollo de un subsistema de administración permitirá al administrador del SGDID Ecumene Pyxel realizar las tareas de configuración y administración con mayor facilidad.

Objetivos Específicos:

1. Caracterizar el tratamiento administrativo de los sistemas de gestión documental.
2. Diseñar el subsistema de administración para el sistema de gestión documental de imágenes digitales Ecumene Pyxel V1.0.
3. Implementar las funcionalidades del subsistema de administración para el sistema de gestión documental de imágenes digitales Ecumene Pyxel V1.0.
4. Validar las funcionalidades del subsistema de administración para el sistema de gestión documental de imágenes digitales Ecumene Pyxel V1.0.

Tareas de la Investigación:

1. Estudio de las tendencias actuales del tratamiento administrativo de los sistemas de gestión documental.
2. Identificación de los requerimientos del subsistema de administración para el sistema de gestión documental de imágenes digitales Ecumene Pyxel V1.0.
3. Selección de las tecnologías, herramientas, estándares, patrones y metodologías necesarias para el desarrollo del subsistema de administración para el sistema de gestión documental de imágenes digitales Ecumene Pyxel V1.0.
4. Confección del diseño del subsistema de administración para el sistema de gestión documental de imágenes digitales Ecumene Pyxel V1.0.
5. Implementación de las funcionalidades del subsistema de administración para el sistema de gestión documental de imágenes digitales Ecumene Pyxel V1.0.

6. Aplicación de las pruebas de funcionalidad, seguridad e integración al subsistema de administración para el sistema de gestión documental de imágenes digitales Ecumene Pyxel V1.0.

Para darle cumplimiento a los objetivos planteados se utilizaron los siguientes métodos científicos que solventaron el desarrollo de la investigación:

Métodos Teóricos:

Histórico-Lógico: Mediante esta práctica se analizaron las tendencias históricas y actuales del desarrollo de las diferentes aplicaciones web para los SGD y los ECM tanto en el ámbito nacional como internacional.

Análítico-Sintético: Se utilizó para el análisis bibliográfico del tema para realizar una síntesis de los aspectos esenciales relacionados con el desarrollo del subsistema de administración, además se empleó para analizar las herramientas, metodologías y lenguajes utilizados en el desarrollo del sistema.

Métodos Empíricos

La Entrevista: Utilizada en la recopilación de la información para el levantamiento de requisitos. También para el estudio y comprensión entre la comunicación que se establece entre la aplicación web y el núcleo (ECM) Alfresco mediante los servicios web. Además se empleó para obtener toda la información necesaria para diseñar e implementar el sistema.

Aportes prácticos esperados del trabajo

El aporte de la presente investigación radica en la obtención de un subsistema de administración para el SGDID Ecumene Pyxel. Con la utilización de dicha aplicación se obtendrán los siguientes beneficios:

- ✓ Una interfaz de administración amigable para el SGDID Ecumene Pyxel.
- ✓ Lograr una total accesibilidad de la información que le facilite al usuario una mayor comprensión referente al proceso.
- ✓ Agilización en los procesos de administración en el sistema.

El presente trabajo está estructurado en tres capítulos, los cuales guiarán la comprensión del proceso

llevado a cabo para dar cumplimiento a los objetivos que fueron planteados. A continuación se muestra una breve descripción de cada uno. Además de conclusiones, recomendaciones, bibliografía, anexos y glosario de términos.

Capítulo 1 Fundamentación Teórica que sustenta la investigación de la propuesta de solución para el subsistema de administración: En este capítulo se precisan elementos teóricos para el desarrollo de la investigación. Se explican los principales conceptos utilizados, las tendencias sobre el proceso de administración en los sistemas de gestión documental, a través del estudio y análisis de soluciones existentes. Además se realiza un estudio sobre las tecnologías y metodologías para el desarrollo del producto.

Capítulo 2 Propuesta de solución para el SGDID Ecumene Pyxel V1.0: Con el objetivo de obtener el producto final se analiza la solución propuesta, se realiza una descripción del modelo de dominio. Se identifican los requerimientos funcionales y no funcionales. Se hace una descripción de los casos de uso y se muestran los diagramas relacionados con estos. Además se describe la arquitectura y los patrones de diseño propuestos para el desarrollo de la solución.

Capítulo 3 Implementación de la propuesta de solución y validación de los resultados: Se realiza el desarrollo y validación del software. También se muestran las partes principales de la interfaz de usuario (IU) de la aplicación, así como la validación del sistema mediante las pruebas y los resultados de las mismas.

Capítulo 1 Fundamentación Teórica que sustenta la investigación de la propuesta de solución para el subsistema de administración

En el presente capítulo se realiza un estudio de los elementos más importantes para el desarrollo de un subsistema de administración para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V.1.0, se precisan elementos teóricos que sustentan la investigación y el desarrollo del tema propuesto, a través del estudio y análisis de soluciones existentes. Se analizan las principales herramientas, metodología, lenguajes y tecnologías que se utilizarán en el desarrollo del producto.

1.1 Gestión Documental

La GD está conformada por un conjunto de operaciones comprometidas en la búsqueda de la eficiencia en la producción, mantenimiento, uso y destino final de los documentos, es decir, desde el momento de su concepción en las oficinas administrativas hasta su ingreso en las instituciones de archivo. Es un área de la administración general que se encarga de garantizar la economía y disposición de los documentos administrativos (Mena, 2005).

Por otro lado, Valderrama en su libro “Gestión Documental” expresa que la GD es un conjunto de actividades administrativas y técnicas tendientes desde su origen hasta su destino final, con el objeto de facilitar su utilización y conservación.

Por su parte Ana Rosa Brito Valdés en “Gestión Documental para los Proyectos de Investigación del MIC” plantea que la gestión documental es el conjunto de técnicas y prácticas usadas para administrar el flujo de documentos de todo tipo en una organización, permitir la recuperación de información desde ellos, determinar el tiempo que los documentos deben guardarse, eliminar los que ya no se utilizan y asegurar la conservación indefinida de los documentos más valiosos.

En resumen la GD es un proceso que se encarga del tratamiento que se le da a todos los documentos de una organización, desde su concepción hasta su eliminación o conservación permanente, teniendo en cuenta todas las etapas que engloba el ciclo de vida de un documento. Concepto con el que se trabajó durante el desarrollo de esta investigación.

1.2 Sistema de Gestión Documental

Los Sistemas de Gestión Documental (DMS, del Inglés *Document Management System*) son programas

de gestión de bases de datos que disponen de una tecnología idónea para el tratamiento de documentos (Codina, 1993).

Un DMS se conforma como el almacén central de información para toda la organización. Es el punto de referencia y recuperación para la documentación de cualquier índole. Garantiza la integridad de la misma y su acceso consistente desde cualquier punto (Eldar, 1991).

Las características generales que identifican los DMS son las siguientes (Eldar, 1991):

- ✓ Seguridad: Proporcionando un almacenamiento único, que unifique documentos y la catalogación de los mismos y que simplifique la definición de políticas de mantenimiento.
- ✓ Capacidad de recuperación y acceso: En forma sencilla y proporcionando métodos de búsqueda ágiles y completos que garanticen la reutilización adecuada de cualquier material o parte del mismo.
- ✓ Centralización: Facilitando el mantenimiento de versiones y garantizando la homogeneidad de formatos y contenidos a lo largo de toda la organización. Además, asegurando la consistencia de los criterios de catalogación.
- ✓ Manejo de grandes volúmenes de documentos.
- ✓ Garantía de acceso a la información más actual.
- ✓ Definición de flujos de trabajo en el sistema para la gestión de procesos operativos entre departamentos y empresas externas.
- ✓ Gestión de la información en formato nativo.
- ✓ Control de acceso a la información.
- ✓ Seguridad ante la posible pérdida de documentación.

1.3 Administrador de contenidos empresariales

Los ECM consisten básicamente en usar la tecnología para capturar, administrar, almacenar, preservar, entregar y explotar contenidos y documentos relacionados con los procesos organizacionales. Las estrategias y herramientas de los ECM se encargan de estructurar toda la información de las organizaciones sin importar la fuente de donde provienen (documentos, correos electrónicos, fotos, audio y vídeo, faxes), mediante una plataforma segura, y definiendo niveles de seguridad y de acceso por

usuario o grupos, consiguiendo así optimizar la toma de decisiones en todos los niveles de las organizaciones (Valdes, 2010).

1.3.1 Administrador de contenidos empresariales Alfresco

Alfresco es un conjunto de tecnologías de código abierto que se utiliza para la gestión del ciclo de vida de un contenido: crear, compartir, versionar, aprobar, publicar y archivar, permitiendo la gestión de la información no estructurada de una organización (Capillas, 2011).

Dentro de sus diversas características se encuentran las siguientes (Capillas, 2011):

- ✓ **Gestión de contenido web:** Incluyendo aplicaciones web y virtualización de sesiones administración de archivos y contenidos que están específicamente enfocados a ser publicados en la Web.
- ✓ **Gestión de documentos:** Captura, organiza y comparte archivos generados con herramientas como Microsoft Office u Open Office. Funcionalidad plena a través de una interfaz intuitiva y fácil de usar. Un único almacén unificado para gestionar cualquier contenido: documentos, imágenes, vídeo, audio.
- ✓ **Soporte multiplataforma:** Oficialmente Windows, GNU/Linux y Solaris.
- ✓ **Interfaz gráfica basada en navegadores de Internet:** Oficialmente Internet Explorer y Mozilla Firefox.
- ✓ **Soporte de clustering:** Despliegue en varios servidores.
- ✓ **Gestión de imágenes:** Incluye captura, etiquetamiento y enrutamiento de imágenes de documentos que son originadas desde un escáner.
- ✓ **Gestión de registros:** Está diseñada para facilitar el uso y una rápida adopción. Creada a partir del repositorio de gestión de documentos de Alfresco archiva los registros arrastrando y soltando desde el escritorio o el cliente de correo electrónico, o desde cualquier navegador web.
- ✓ **Seguridad:** Alfresco se ha diseñado para cumplir los máximos estándares de seguridad. Los permisos establecidos para los distintos usuarios o grupos permiten definir con exactitud quién puede ver el contenido, ya sea por carpeta, por archivo. Cualquier contenido en Alfresco se compone de dos elementos básicos, el contenido y la información sobre dicho contenido. El contenido y sus versiones son guardados como archivos binarios, lo que permite un almacenamiento extensivo, acceso aleatorio, y otras opciones para dichos servicios; mientras que la información sobre el contenido se gestiona completamente en la base de datos.

- ✓ **Estándares abiertos:** Alfresco es una plataforma de código abierto creada a partir de estándares abiertos para garantizar la compatibilidad y la capacidad de alcanzar los máximos niveles de integración y aprovechar los conocimientos estándar del sector.

Ventajas

- ✓ Alfresco permite acceder a su repositorio a través de protocolos como CIFS/SMB, FTP, WebDAV.¹
- ✓ La seguridad sobre los espacios de contenido se lleva a cabo asignando roles a usuarios y grupos de usuarios.
- ✓ La corrección de errores y parches tiene más presencia internacional y mayor repercusión.
- ✓ Permite la interoperabilidad con otras aplicaciones.

Alfresco demuestra que se trata de un software con una gran madurez, capaz de implementar satisfactoriamente los requisitos de gestión documental de cualquier organización (Brun, 2008).

1.4 Interfaces web desarrolladas para Alfresco

Actualmente se han desarrollado diferentes aplicaciones que le dan tratamiento documental a un número considerado de documentos entre ellas se encuentran las siguientes.

1.4.1 Alfresco Explorer

Alfresco Explorer es la clásica interfaz web del Alfresco. Está integrado con el núcleo de la aplicación y ofrece algunas funcionalidades de administración del sistema. Entre sus principales características se encuentran (Software, 2009):

- ✓ Es código abierto, liberado bajo la licencia LGPL.
- ✓ Está desarrollado en Java.
- ✓ Consume los servicios directamente del ECM Alfresco.

Algunas de sus limitaciones son las siguientes:

- ✓ Es muy complicada su configuración.
- ✓ Es una interfaz poco intuitiva para el usuario.

¹ Edición y versionado distribuido sobre la Web.

- ✓ No cuenta con todas las funcionalidades necesarias para una fácil configuración y administración del sistema.
- ✓ La última versión del Explorer no es compatible para la versión actual del Alfresco.
- ✓ Escasa información de consulta disponible sobre cómo se realiza el proceso administrativo.

1.4.2 Ifresco

Es un cliente web que ofrece su código abierto para Alfresco tiene todas las características que son necesarios para el trabajo diario con los DMS / ECM y que puede ser operado fácilmente. Funciona como la web basada en navegador que ofrece al consumidor un uso sencillo de listas ordenables, menús contextuales y permite arrastrar y soltar elementos (ifresco.at, 2011).

Arquitectura y características de Ifresco:

- ✓ La integración con Alfresco está basado en la norma servicios RESTful² API³ y Webservice.
- ✓ Implementado en PHP5.
- ✓ Utiliza Doctrine⁴ para la base de datos ORM⁵.
- ✓ Como sistema gestor de base de datos utiliza MySQL para salvar datos administrativos y las sesiones de usuarios.
- ✓ Idiomas compatibles (Inglés, Alemán, Ruso).
- ✓ Maneja ExtJS⁶ para la interfaz de usuario con plug-ins.

Como limitaciones se identificaron:

- ✓ No posee las funcionalidades y características necesarias para una correcta gestión de las imágenes en todo su ciclo de vida.
- ✓ No cuenta con una comunidad que le dé seguimiento y corrección a las deficiencias.

² Transferencia de estado representacional.

³ Interfaz de programación de aplicaciones.

⁴ <http://www.doctrine-project.org/>

⁵ Mapeo Objeto Relacional.

⁶ Es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas (es.wikipedia.org/wiki/ExtJS).

- ✓ Está desarrollado en Symfony 1.4, que es una versión obsoleta por lo cual resulta un poco complicada su modificación.

1.4.3 Opsoro

Opsoro es un cliente web alternativo para la Plataforma del ECM Alfresco. Su objetivo es proporcionar una IU más fácil con una experiencia de usuario más rica, mediante el aprovechamiento de las últimas tecnologías web. Al igual que el ECM Alfresco, Opsoro es un software de código abierto publicado bajo la licencia GPL⁷, está desarrollado en la librería ExtJS, utiliza la tecnología Web Script. Su nombre se debe a la abreviatura de “*Open SourceRocks*” cuya traducción al español es “Rocas de código abierto (Project, 2010).

Características actuales (Opsoro, 2010):

- ✓ Es amigable e intuitivo para el usuario.
- ✓ Permite la búsqueda a medida que se escribe.
- ✓ Contiene categorías y etiquetas de carpetas.
- ✓ Contiene etiquetado de documentos.
- ✓ Permite la vista previa de las imágenes.
- ✓ Permite la visualización en línea de textos, PDF e imágenes.
- ✓ Utiliza principalmente JSON⁸ como formato de respuesta para mantener los fragmentos de datos tan pequeños como sea posible.
- ✓ Proporciona una interfaz de usuario sencilla.

Desventajas de la interfaz.

- ✓ No cuenta con documentación suficiente sobre el desarrollo de la administración en la tecnología.
- ✓ Por el momento no utiliza Java, ni los respaldos Scripts Web: Ext JS 2.0.
- ✓ Opsoro comenzó como un prototipo y aún se encuentra en un estado temprano de su desarrollo.

7 Licencia Pública general.

8 Formato ligero para el intercambio de datos.

1.4.4 DoCASU

DoCASU es una IU desarrollada para el ECM Alfresco. El principal objetivo es obtener un entorno mucho más sencillo y parametrizable que el cliente Alfresco. Esta interfaz se comunica con Alfresco mediante los Web Scripts⁹ externos permitiendo la eficiencia en la comunicación entre los dos sistemas. Es un cliente ligero para Alfresco que permite simplificar el uso de Alfresco para los usuarios finales. Fue desarrollado por Optaros y liberado bajo licencia GPL, es muy sencillo de usar, debido a lo simple que es (López, 2013)

Sus principales características funcionales son (Venizia, 2013):

- ✓ Facilidad de uso: Permite la subida y recuperación de contenido, potenciando la flexibilidad en dichas funcionalidades y omitiendo el resto de las no relevantes al usuario final (no así al administrador).
- ✓ Gestión de permisos sencilla y altamente flexible con posibilidad de modificación del esquema en función a jerarquía de grupos definidos.
- ✓ Acceso a pistas de auditorías para administradores.
- ✓ Configuración sencilla de notificaciones por correo para grupos de usuarios específicos (basados en tipos de documentos).
- ✓ Gestión del versionado de documentos.
- ✓ Distribución autorizada de contenido por tipologías. El usuario final solo visualiza los contenidos basados en la búsqueda realizada, por lo que la navegación por carpeta esta suprimida.
- ✓ Búsqueda avanzada con impresión de informes.

Como limitaciones se identificaron las siguientes

- ✓ Realiza todas las operaciones e intercambio de datos mediante peticiones URL usando los métodos POST, GET y DELETE principalmente, esta vía de solución gana en accesibilidad y confort al usuario pero no es muy segura, ya que los datos enviados por la URL pueden ser intervenidos por un usuario en la red usando un sniffer¹⁰ o cualquier herramienta para ello.

⁹ http://wiki.alfresco.com/wiki/Web_Scripts#Why_Web_Scripts.3F

¹⁰ Es un programa informático que registra la información que envían los periféricos, así como la actividad realizada en un

- ✓ Usa la API Javascript del Alfresco, a través de la librería Javascript Ext. La API de Javascript no es muy amplia y hay funcionalidades que no se encuentran bien definidas, lo que dificulta el trabajo con el repositorio de Alfresco.
- ✓ No se han desarrollado nuevas versiones, la última versión disponible que se encontró es la 1.6 que funciona con Alfresco 3.2.
- ✓ No tiene una amplia comunidad de usuarios para su posterior desarrollo.
- ✓ No posee documentación suficiente sobre el desarrollo de la administración en el sistema.

1.4.5 eXcriba

En el Centro de Informatización Universitaria de la Facultad 1 (CENIA) de la UCI se desarrolló un software para la Gestión de los Documentos Administrativos (GDA) llamado eXcriba. Está compuesto por un núcleo o motor que es el ECM Alfresco y una aplicación Web la cual se integra mediante el API Servicios Web y los Web Scripts externos que brinda el repositorio del ECM Alfresco. El objetivo principal de este producto es automatizar los procesos documentales que se ejecutan dentro de cualquier entidad, desde la elaboración de un documento en su fase de inicio hasta su conservación o expurgo¹¹.

Desventajas

- ✓ No es flexible al cambio de las interfaces de los tipos de usuarios pues todos los usuarios tienen una misma interfaz.
- ✓ Carece de funcionalidades necesarias para que desde aplicaciones externas se puedan ejecutar las actividades de un proceso de negocio definido tanto de forma secuencial como paralela por distintos miembros, para la obtención de un mismo objetivo.
- ✓ No cuenta con todas las funcionalidades necesarias que se requieren en el sistema a desarrollar tales como: No se visualiza la ruta de navegación, No posee la opción de auditar los contenidos.

Después del análisis de cada una de estas aplicaciones se llega a la conclusión que a pesar de que todas dan tratamiento documental a un número considerado de contenidos en una organización, aun así no cumplen con las especificaciones requeridas para el correcto desarrollo del subsistema de administración; debido a las desventajas antes identificadas de cada uno de ellos. Es válido resaltar que se tomarán las

determinado ordenador.

11 Retirar las copias conocidas como inútiles.

buenas prácticas del DoCASU y el eXcriba como punto de partida por los desarrolladores teniendo en cuenta los siguientes aspectos:

- ✓ Conexión mediante servicios Web para aumentar la rapidez de respuesta a las peticiones realizadas por el usuario.
- ✓ Utilización principalmente de JSON como formato de respuesta para mantener los fragmentos de datos pequeños y disminuir la carga.

1.5 Tecnologías asociadas al desarrollo del sistema

El subsistema será desarrollado utilizando las herramientas y tecnologías establecidas por la línea base del Centro, a continuación se describirán las características de cada una de ellas.

1.5.1 Metodología de Desarrollo

Para el desarrollo de un sistema existen diferentes formas, pero para lograr una aplicación con calidad se necesita aplicar una metodología de desarrollo que sea coherente y organizada que garantice un producto eficiente y con ello que satisfaga las necesidades del cliente. Es por ello que los desarrolladores necesitan una guía para crear un producto de calidad.

Debido a que el tiempo de desarrollo del subsistema de administración es corto, el grupo de desarrolladores es pequeño y teniendo en cuenta los objetivos trazados, se determinó utilizar una metodología de desarrollo ágil.

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo (INTECO, 2009).

1.5.2 Metodología de desarrollo OpenUp

Para la selección de la metodología se tuvo en cuenta la establecida por la línea base del Centro de Ideoinformática CIDI y por las ventajas que esta proporciona.

OpenUp es una metodología de desarrollo que permite un proceso ágil y unificado, contiene un conjunto mínimo de prácticas que ayudan a los equipos a ser más poderosos en el desarrollo de software. OpenUp abraza una filosofía pragmática y ágil que se centra en la naturaleza colaborativa de desarrollo de software. Es un proceso iterativo, mínimo, completo y extensible, puede utilizarse tal cual o ampliarse para

tratar una amplia variedad de tipos de proyecto. Se caracteriza por ser iterativo e incremental, centrado en la arquitectura y guiado por los casos de uso. OpenUp está organizado en dos dimensiones diferentes pero interrelacionadas: el método y el proceso (Ideoinformática, 2012).

Es mínimo, porque solo incluye el contenido del proceso fundamental; completo, ya que logra ser manifestado como proceso entero para construir un sistema y es extensible, porque puede ser utilizado como base para agregar o adaptar según las necesidades. Es un proceso que apoya el desarrollo iterativo, ágil, e incremental y es aplicable a un amplio sistema de plataformas y de usos del desarrollo.

OpenUp divide en 4 fases el desarrollo del software:

Inicio:

Los objetivos de esta etapa son:

- ✓ Determinar la visión del proyecto.
- ✓ Identificar la funcionalidad clave del sistema.
- ✓ Determinar al menos una posible solución.
- ✓ Entender el costo, el cronograma y los riesgos asociados al proyecto.

Elaboración: Es el segundo de las 4 fases del ciclo de vida del OpenUP donde se tratan los riesgos significativos para la arquitectura. En esta etapa el objetivo es determinar la arquitectura óptima.

Construcción: El propósito de esta fase es completar el desarrollo del sistema basado en la arquitectura.

Transición: El propósito en esta fase es asegurarse que el software está listo para entregarse a los usuarios y evalúa la funcionalidad y performance del último entregable de la fase de construcción.

Ventajas:

- ✓ Metodología de desarrollo de software de código abierto diseñado para pequeños equipos.
- ✓ Proceso iterativo e incremental que es Mínimo, Completo y Extensible.
- ✓ Practicantes de desarrollo de software (desarrolladores, administradores de proyectos, analistas y probadores) trabajan juntos como un equipo de proyecto.
- ✓ No define un modelo de negocio ni de dominio necesario.
- ✓ Permite detectar errores tempranos a través de un ciclo iterativo.

- ✓ Evita la elaboración de documentación, diagramas e iteraciones innecesarios.
- ✓ Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.

1.5.3 Marco de trabajo a utilizar

Con el objetivo de garantizar una aplicación rápida de código ligero, legible y seguro es recomendable utilizar un marco de trabajo. Para el desarrollo de la interfaz del subsistema de administración del sistema de gestión documental de imágenes digitales Ecumene Pyxel se propone Symfony2.

Symfony es un framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón modelo vista controlador. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web (Potencier, 2009).

Symfony está desarrollado completamente en PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas Unix, Linux, como en plataformas Windows. Su arquitectura interna está completamente desacoplada, lo que permite reemplazar o eliminar fácilmente aquellas partes que no encajan en un proyecto. Utiliza una licencia MIT¹², con la que se puede hacer aplicaciones Web comerciales, gratuitas y/o de software libre (Eguiluz, 2011).

Posee las siguientes ventajas:

- ✓ Fácil de instalar y configurar en la mayoría de las plataformas.
- ✓ Independiente del sistema gestor de base de datos.
- ✓ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✓ Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- ✓ Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.

12 Licencia de código abierto y libre.

- ✓ Es un framework flexible debido a que se ajusta a los tres tipos de framework:
 1. Framework completos: Ofrecen todas las funcionalidades necesarias para construir aplicaciones web.
 2. Framework de componentes: Sus componentes presentan un bajo acoplamiento por lo que pueden ser utilizados por partes.
 3. Micro Framework: Muy pequeños y ligeros. Ofrecen las funcionalidades básicas para desarrollar aplicaciones simples.

1.5.4 Entornos integrados de desarrollo

Un Entorno de Desarrollo Integrado (IDE por sus siglas en Inglés) es un entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Pueden ser aplicaciones por sí solas o ser parte de aplicaciones existentes. Ofrecen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic (Ecured, 2009).

Se determinó que se utilizará NetBeans en su versión 7.2 teniendo en cuenta las características de la aplicación a desarrollar y las ventajas que provee.

Características:

Es un IDE gratuito y de código abierto para el desarrollo de aplicaciones web, de escritorio, disponible en los sistemas operativos Windows, Linux, Mac OS X y Solaris. La base en la que se sustenta su elección es la siguiente:

1. Permite desarrollar aplicaciones utilizando el framework Symfony y ejecutar los comandos del mismo directamente desde la interfaz del IDE.
2. NetBeans IDE es de fácil instalación y multiplataforma ya que se ejecuta en Windows y GNU/Linux.
3. Es un IDE ideal para trabajar con el lenguaje de desarrollo JAVA.
4. Ofrece un excelente entorno para programar en PHP.
5. La IDE de NetBeans es perfecta y muy cómoda para los programadores.
6. Un entorno de desarrollo completo, fácil de usar y gratuito.

1.5.5 Sistema Gestor de Bases de Datos

Los sistemas gestores de bases de datos (SGBD) son programas que proporcionan una interfaz entre los datos almacenados y las aplicaciones que permiten acceder a los mismos. Son herramientas de software que facilitan los procesos de definición, construcción y manipulación de la base de datos para distintas aplicaciones (Pérez, 2010).

Cada SGBD tiene sus características propias que lo distingue, su selección varía teniendo en cuenta las particularidades de la aplicación a desarrollar. El Sistema Gestor de Bases de Datos utilizado por el Núcleo para almacenar los datos es PostgreSQL con versiones superiores a la 8.4.

Características:

- ✓ Instalación ilimitada: Es frecuente que las bases de datos sean instaladas en más servidores de lo que permite la licencia.
- ✓ Sirve de soporte a los lenguajes más populares como PHP, C, C++, Java, Python, Ruby.
- ✓ El número de bases de datos que puede contener es ilimitado.
- ✓ Extensible: El código fuente está disponible para todos sin costo.
- ✓ Multiplataforma: PostgreSQL está disponible en casi cualquier sistema operativo entre ellos: Linux, Windows, Mac SX, Solaris, BSD.
- ✓ Diseñado para ambientes de alto volumen.

1.5.6 Herramientas para el Modelado. Visual Paradigm

Como herramienta para el proceso de análisis y diseño se escogió Visual Paradigm, teniendo en cuenta que es un entorno profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue.

Visual Paradigm es una herramienta CASE de modelación visual que utiliza el lenguaje de modelado unificado (UML, del Inglés *Unified Modeling Language*) como Lenguaje de Modelado, entre sus principales ventajas se encuentran:

- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Capacidades de ingeniería directa e inversa.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.

- ✓ Disponibilidad de múltiples versiones.
- ✓ Disponibilidad de integrarse en los principales IDEs.
- ✓ Es multiplataforma y muy útil para la generación de código fuente en PHP.

1.6 Lenguajes utilizados

En la actualidad son muchos los lenguajes que han surgido, con el objetivo de facilitar la comunicación hombre máquina. Esta evolución de los lenguajes ha venido impulsada por varios motores importantes:

- ✓ Los avances tecnológicos.
- ✓ Los avances conceptuales (de planteamiento).
- ✓ Los avances en cuanto a enfoque de la programación.

A continuación se representa algunos de los diferentes lenguajes de programación para el desarrollo de la aplicación.

1.6.1 Lenguaje de Modelado. UML

El UML es un lenguaje de modelado visual de propósito general orientado a objetos impulsado por el Object Management Group, cuyo vocabulario y reglas se centran en la representación conceptual y física de un sistema. UML es un lenguaje estándar para especificar, visualizar, construir y documentar todos los artefactos de un sistema de software.

Características:

- ✓ UML ofrece vocabulario y reglas para crear y leer modelos bien formados que constituyen los planos de un sistema software.
- ✓ UML es independiente del Proceso de desarrollo.
- ✓ UML cubre las diferentes vistas de la arquitectura de un sistema mientras evoluciona a través del ciclo de vida del desarrollo de software.

1.6.2 PHP 5.3

PHP 5 Es un lenguaje de programación utilizado para la creación de sitios Web. Es un acrónimo recursivo que significa *Hypertext Pre-processor* lenguaje de programación interpretado. Surgió en 1995, desarrollado por PHP Group.

PHP es un lenguaje del lado servidor utilizado para la generación de páginas Web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. No necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl (2009, citado por Isabel María García Hilarión, 2012).

Características específicas

- ✓ Es multiplataforma, puede ser utilizado sobre los sistemas operativos: GNU/Linux, Windows.
- ✓ Funcionalidades de conexión con la mayoría de los sistemas de gestión de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server.
- ✓ 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 frameworks de desarrollo y sistemas de gestión de contenidos desarrollados sobre este lenguaje.
- ✓ Soporta la mayoría de servidores web, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, Oreilly Website Pro server.
- ✓ Es muy fácil de aprender y se caracteriza por ser un lenguaje muy rápido.

1.6.3 JavaScript 1.4

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 HTML¹³ y ejecutarlas adecuadamente.

Java Script es un lenguaje script multiplataforma, pequeño y ligero, con una sintaxis semejante a la del lenguaje Java; no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores Web.

1.6.4 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, documentos HTML o XML. Los CSS son la mejor forma de separar los contenidos y su presentación y es

13 Lenguaje de marcado de hipertexto.

imprescindible para crear páginas Web complejas. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página.

CSS es el mecanismo que permite separar los contenidos definidos mediante XHTML y el aspecto que deben presentar esos contenidos.

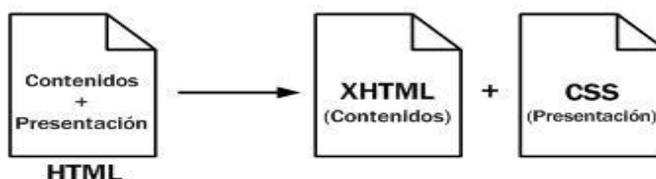


Figura 1 Esquema de la separación de los contenidos y su presentación

Ventajas de utilizar CSS

- ✓ CSS brinda mayor control sobre las presentaciones, ya que su código facilita la modificación del sitio web.
- ✓ Los documentos HTML que emplean CSS son más pequeños, ya que el diseño del estilo es reutilizado continuamente.
- ✓ El documento HTML en sí mismo es más claro de entender al hacer uso de este estilo.

1.6.5 XML

El *Extensible Markup Language* o Lenguaje Extensible de Marcas (XML), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. Es un formato que permite la lectura de datos a través de diferentes aplicaciones. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML¹⁴, SVG¹⁵, MathML¹⁶.

Características específicas

- ✓ Es extensible, después de diseñado y puesto en producción, es posible extender.

¹⁴ Lenguaje marcado de hipertexto extensible.

¹⁵ Scale Vector Graphics.

¹⁶ Mathematical Markup Language es un lenguaje de marcado basado en XML, cuyo objetivo es expresar notación matemática.

- ✓ XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- ✓ Permite proporcionar diferentes vistas sobre los datos (HTML, PDF), dependiendo de quién sea el cliente.
- ✓ Es fácil de distribuir y permite que los datos estructurados sean uniformes e independientes de aplicaciones o compañías.

1.8 Conclusiones parciales

En este capítulo se hizo un análisis de la teoría, herramientas, lenguajes y tecnologías utilizadas en la propuesta de solución, se analizaron diferentes conceptos y las tendencias actuales sobre el tratamiento administrativo en los DMS, pues existe una gran diversidad en el desarrollo de cada una de estas tecnologías. Se fundamentó la elección de cada una de las herramientas tales como: la utilización del marco de trabajo Symfony 2.1, ya que es la versión más actualizada y teniendo en cuenta que es un framework diseñado para el desarrollo de aplicaciones web y como gestor de bases de datos PostgreSQL en su versión 8.4. Se decidió utilizar como metodología de desarrollo de software OpenUp para guiar todo el proceso de desarrollo. Para el lenguaje de modelado se utilizará el UML y Visual Paradigm como herramienta Case. Para apoyar el proceso de implementación se utilizará el IDE de desarrollo NetBeans 7.2 en su versión de PHP por ser un IDE que permite el intercambio con el framework seleccionado y ofrece un entorno adecuado para programar en PHP.

Capítulo 2 Propuesta de solución para el SGDID Ecumene Pyxel

En este capítulo se describe la solución propuesta. Además son enumerados los requisitos funcionales y no funcionales con los cuales debe cumplir el producto final desarrollado, se describen los diagramas de casos de uso y el modelo de dominio.

2.1 Propuesta de solución

En la actualidad existen en todo el mundo diversos programas que garantizan la gestión documental. Son muchas las organizaciones que optan por estos sistemas para adaptarlos a las especificidades y necesidades de cada institución garantizando las facilidades que brindan, para entender cuestiones como la forma de visualizar los contenidos y las operaciones que se pueden realizar sobre estos.

A raíz del desarrollo del Programa de Informatización para la Prensa, la Universidad de las Ciencias Informáticas desarrolló un sistema para la gestión de recursos fotográficos basado en el ECM Alfresco denominado SGDID Ecumene Pyxel. Este sistema tiene ventajas para la gestión de los contenidos fotográficos, pero aún no cuenta con una interfaz individual para la administración del sistema de fácil uso para el usuario, esto trae como consecuencias:

- ✓ Dificil manejo del usuario con el sistema.
- ✓ La información brindada por el sistema no satisfacen todas las necesidades del usuario.
- ✓ Demora en la realización de los procesos de administración.

Luego de analizar la situación antes planteada se propone desarrollar una capa de presentación para la administración que permita la interactividad con el núcleo del sistema, mediante el uso de los servicios Web y que consuma todos los servicios que brinda el ECM Alfresco. El subsistema de administración para el SDDID Ecumene Pyxel será una solución para la Web, de fácil uso y amigable para los usuarios finales, basada en estándares abiertos que permitirá la gestión de usuarios, grupos, cuadro de clasificación y la realización de búsquedas atendiendo a diferentes conceptos entre otras funcionalidades.

2.2 Modelo de dominio

Después de haber realizado un análisis profundo del problema en cuestión, se determinó a través de la presente investigación que no todos los procesos están bien definidos. Por ello se debe realizar un modelo de dominio. Este modelo contribuye posteriormente a identificar algunas clases que se utilizarán en el

sistema.

Un modelo de dominio se realiza para poder representar de una forma u otra los procesos en cuestión, con el objetivo de obtener un buen entendimiento del funcionamiento del sistema y capturar correctamente los requisitos para así poder construir un sistema que cumpla las especificaciones del cliente.

Un modelo del dominio o Modelo Conceptual captura los tipos de objetos más importantes en el contexto del sistema. El mismo tiene como objetivo describir los objetos que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Representa conceptos del mundo real, no de los componentes del software (Jacobson, 2000).

A continuación se muestra el diagrama de dominio, el cual ayuda a comprender los conceptos que utiliza el usuario administrador, los objetos que existen relacionados con el sistema que se va a desarrollar y las relaciones que hay entre ellos.

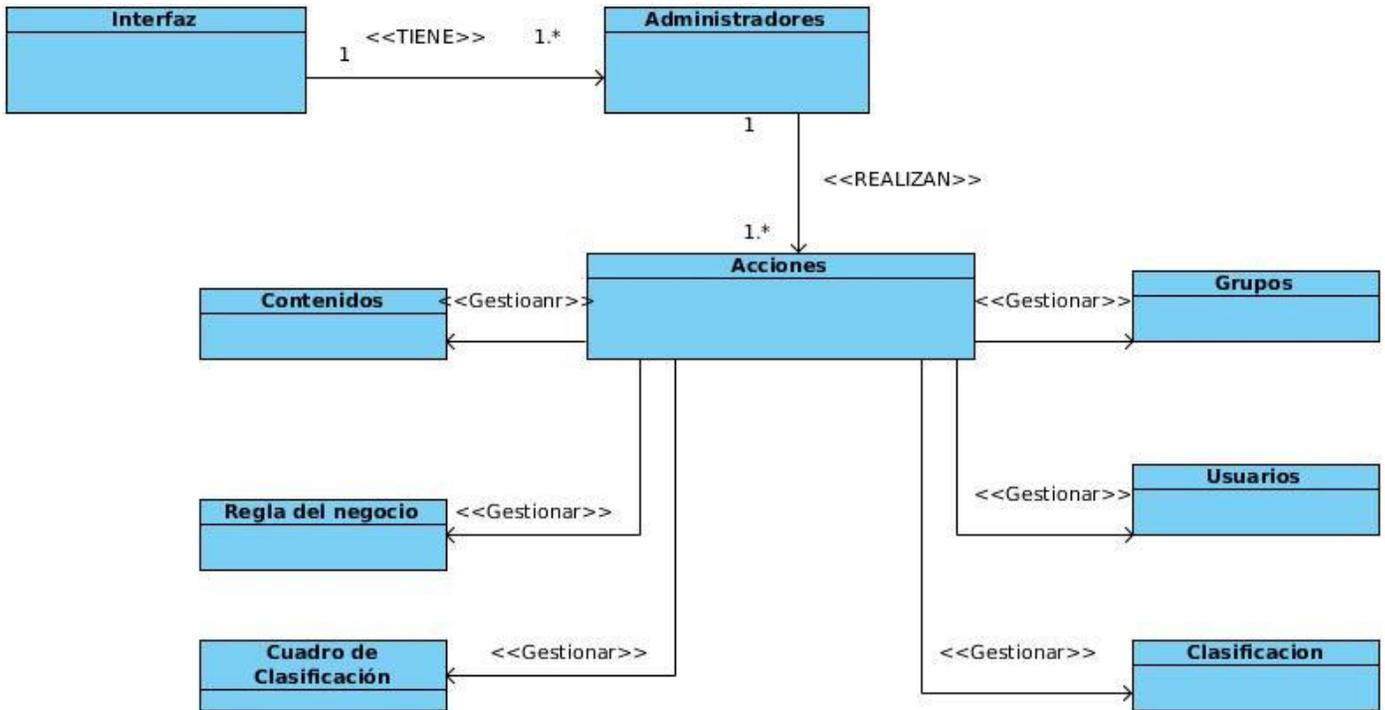


Figura 2 Modelo de dominio

Descripción de clases del modelo del dominio

1. Interfaz: Esta clase representa la vista de la aplicación, es desde donde los usuarios pueden acceder a determinado espacio.
2. Acciones: Esta clase determina las posibles operaciones válidas que se le aplican a los contenidos que se gestionan en la aplicación.
3. Administradores: Representa al personal encargado de realizar todas las operaciones correspondientes dentro del sistema. Es el responsable de tratar y manejar toda la información dentro de la aplicación.
4. Contenidos, Grupos, Cuadro de clasificación, Usuarios y Reglas del negocio, Clasificación o Taxonomías: Son los contenidos que se gestionan, se pueden acceder desde la aplicación y sobre los que se realiza determinada acción.

2.3 Listado de los requisitos funcionales y no funcionales

Los requerimientos funcionales de un sistema describen lo que el sistema debe hacer. Estos requerimientos dependen del tipo de software que se desarrolle, de sus posibles usuarios y del enfoque general tomado por la organización al redactar requerimientos. Son declaraciones de los servicios que debe proporcionar el producto, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en diferentes situaciones (Ángel Gabriel, 2010).

A continuación se puede observar el listado de los requisitos funcionales y no funcionales del sistema:

Requisitos	Complejidad	Prioridad
Autenticar usuario	Alta	Alta
Crear usuario	Alta	Alta
Editar usuario	Alta	Media
Eliminar usuario	Alta	Alta
Listar usuarios	Alta	Alta
Ver detalles del usuario	Alta	Media
Cambiar contraseña de usuario	Alta	Media
Filtrar cuenta de usuario	Media	Media
Crear grupo	Alta	Alta
Eliminar grupo	Alta	Alta
Listar grupos	Alta	Alta
Editar grupo	Alta	Alta
Mostrar usuarios de un grupo	Alta	Alta
Asignar usuarios a un grupo	Alta	Alta
Quitar usuarios de un grupo	Alta	Alta
Auditar contenido	Media	Alta
Crear acceso a directorio	Media	Media

Eliminar acceso directo	Media	Media
Mostrar acceso directo	Media	Media
Realizar búsqueda simple.	Media	Alta
Realizar búsqueda avanzada.	Alta	Media
Crear Regla del negocio	Alta	Media
Editar regla del negocio	Alta	Media
Mostrar regla del negocio	Alta	Media
Eliminar regla del negocio	Alta	Media
Crear clasificación	Alta	Media
Mostrar clasificación	Alta	Media
Editar clasificación	Alta	Media
Eliminar clasificación	Alta	Media
Crear cuadro de clasificación	Media	Media
Editar cuadro de clasificación	Media	Media
Eliminar cuadro de clasificación	Media	Alta
Adicionar espacio a un cuadro de clasificación	Media	Alta
Eliminar espacio de un cuadro de clasificación	Media	Alta
Asignar rol a un usuario en un espacio del cuadro de clasificación	Media	Alta
Quitar rol a un usuario en un espacio de un cuadro de clasificación	Media	Alta
Gestionar contenido	Media	Media
Visualizar ruta de navegación	Media	Media

Tabla 1 Listado de requisitos funcionales

Se determinaron los requisitos no funcionales con el objetivo de especificar o restringir las propiedades del sistema, estos se encuentran descritos en la tabla 2.

Los requerimientos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema (Ángel Gabriel, 2010).

Requisito no Funcionales
Requisitos de Usabilidad
Facilidad de uso por parte de los usuarios: el sistema debe tener una interfaz amigable y cómoda para los usuarios que posean conocimientos básicos en el tratamiento con las computadoras y de un ambiente web.
Posibilidad a los usuarios de llegar de manera rápida y efectiva a la información buscada.
Cerrar automáticamente la sesión en el sistema inactivo.
Requisito de Soporte
La documentación resulta requisito indispensable como evidencia de la evolución del Proyecto.
Especificación de casos de uso.
Especificación de requisitos.
Casos de prueba funcionales.
Arquitectura de software.
Restricción de diseño y la Implementación
El producto debe implementarse hasta el máximo posible, con herramientas Open Source, sobre la filosofía de Software Libre.
Lenguaje php 5.3.10, JavaScript 1.4.
Framework Symfony 2.1.
Visual Paradigm 8.0
Bootstrap
Requisitos de Licencia
Las herramientas seleccionadas para el desarrollo del producto están respaldadas por licencias libres (LGPL).
Software libre

Requisitos de Software
Sistema Operativo Ubuntu 12.04 o superior.
Servidor Apache 2.2.
Sistema Gestor de Bases Datos: PostgreSQL con una versión mayor o igual a la versión 8.4.
Requisitos de Hardware
El servidor de la interfaz debe tener al menos: Memoria RAM 1 GB o superior. HDD (80-160 GB).
El servidor de Base de Datos debe tener al menos: 2 GB RAM, HDD (entre 250 y 500 GB).
El servidor del núcleo del sistema debe tener al menos: 2 GB RAM, HDD (entre 250 y 500 GB).
Requisitos de Seguridad
Protegida de acceso del personal no autorizado (Confidencialidad).
Incluye mecanismos de chequeo de integridad y realización de auditorías (Integridad).
A todos los usuarios autorizados se les garantizan el acceso a la información en el horario que deseen hacer uso de la aplicación (Disponibilidad).
La información manejada por el sistema será objeto de cuidadosa protección, se establecerá la técnica del mínimo privilegio a todo aquel personal que interactúe con el sistema, así se le otorgarán los permisos necesarios a cada uno según su rol.

Tabla 2 Listado de Requisitos no funcionales.

Descripción de los actores del sistema

Seguidamente se muestra el actor que interactúa con el sistema y sus funciones.

Actor	Justificación
Administrador	Este actor al cual está dirigida la aplicación, será quien hará uso de ella, así como de todas las funcionalidades con las que debe cumplir el sistema.

Tabla 3 Actores que interactúan en el sistema.

2.4 Definición de casos de uso del sistema

Los casos de uso del sistema (CUS) se utilizan para capturar los requisitos, describiendo la interacción entre el usuario y el sistema para lograr un objetivo específico. A continuación se muestran los CUS identificados.

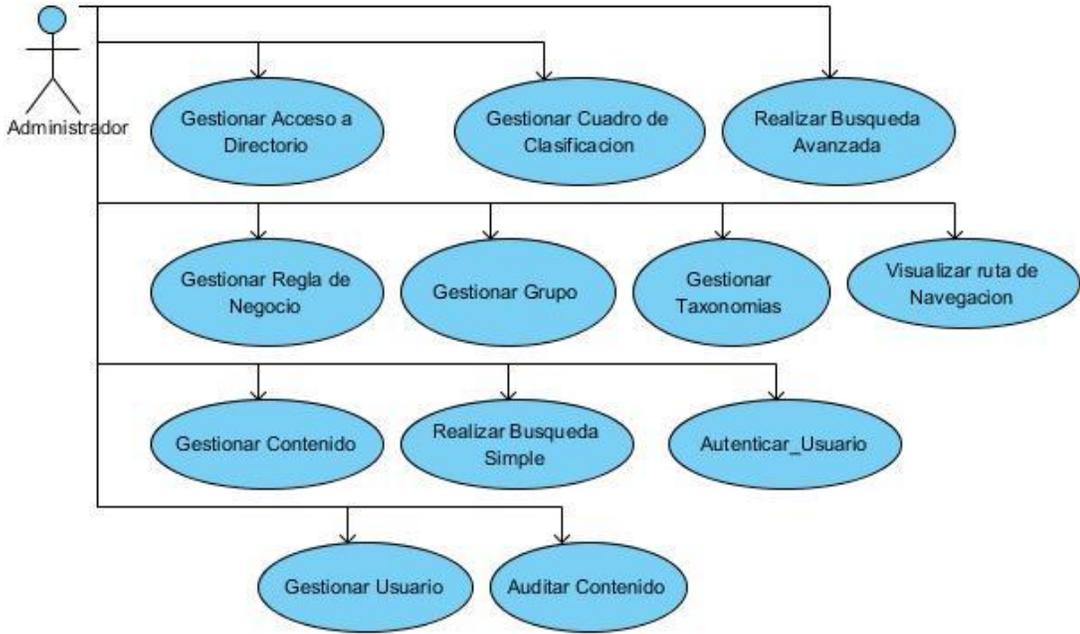


Figura 3 Diagrama de casos de uso del sistema

2.4.1 Descripción de los casos de usos del sistema

En la descripción de los casos de uso se puede percibir la interacción del actor con el sistema, así como las especificidades propias de cada funcionalidad.

En las tablas descritas a continuación se describe la especificación de los casos de uso más significativos.

CU Autenticar Usuario

Objetivo	Autenticación en el sistema.
Actores	Administrador: (Inicia) se autentica con su usuario y contraseña en el sistema.
Resumen	El CU se inicia cuando el usuario desea autenticarse en el sistema, el mismo muestra la interfaz de autenticación donde se deben llenar los campos: nombre de usuario y contraseña.
Complejidad	Alta
Prioridad	Critico

Precondiciones	Los datos introducidos han sido validados.	
Postcondiciones	Usuario autenticado en su espacio personal.	
Flujo de eventos		
Flujo básico Autenticar usuario.		
	Actor	Sistema
1	Para autenticarse en el sistema el usuario debe introducir en su navegador la dirección donde se encuentra el producto http://ip_servidor	Responderá con una interfaz de autenticación, donde debe llenar los siguientes campos: Nombre de Usuario * Contraseña *
2	Introduce el usuario y la contraseña en la interfaz y los envía a la aplicación.	Comprueba que los datos introducidos por el usuario sean correctos
3		Direcciona al usuario hacia la interfaz principal del sistema.
Flujos alternos		
1 Los datos son incorrectos		
	Actor	Sistema
1	Introduce usuario o contraseña incorrecta.	Muestra un mensaje de error donde se le comunica al usuario que no existen el par usuario y contraseña suministrados por él en la base de datos del sistema.
2		
2 Los campos están vacíos.		
	Actor	Sistema
1.	Deja campos vacíos.	Muestra mensaje de error donde le comunica al usuario que no puede dejar campos obligatorios vacíos.
2.		Termina el CU.

Tabla 4 Descripción del CU Autenticar usuario

CU Gestionar Usuario.

Objetivo	Que el administrador pueda agregar, editar, eliminar, habilitar, deshabilitar, listar usuarios del sistema, ver detalles del usuario y agrupar usuarios.
Actores	Administrador: (Inicia)
Resumen	El CU se inicia cuando un usuario ya autenticado en el sistema es capaz de añadir, editar, eliminar, listar, deshabilitar y habilitar varios usuarios del sistema o sea realizar las operaciones básicas de un CRUD sobre los usuarios del sistema.
Complejidad	Alta
Prioridad	Alta

Precondiciones	Administrador del sistema autenticado.	
Postcondiciones	Usuario creado, editado, datos visibles o eliminados.	
Flujo de eventos		
Flujo básico Adicionar usuario		
	Actor	Sistema
1.	El administrador una vez autenticado en el sistema procede a posicionarse en la sección Gestionar Usuario.	Responderá con una interfaz que permite realizar varias acciones con un usuario: <ul style="list-style-type: none"> ✓ Adicionar usuario. Ver Sección1:"Adicionar usuario". ✓ Editar usuario. Ver Sección2:"Editar usuario". ✓ Ver detalles _usuario. Ver Sección3:"Ver detalles_ usuario". ✓ Eliminar usuario. Ver Sección4:"Eliminar usuario". ✓ Cambiar contraseña .Ver Sección5: "Cambiar_ Contraseña usuario" ✓ Listar usuario. Ver: Sección6: "Listar usuario"
Flujos alternos		
1 Gestión de usuario cancelada.		
	Actor	Sistema
1.	Cancela la gestión de usuario.	Retorna a la interfaz de administración.
2.		
Sección 1: "Adicionar usuario"		
Flujo básico Adicionar usuario		
	Actor	Sistema
1.	Selecciona la opción adicionar usuario.	Muestra el formulario para introducir los datos del nuevo usuario.
2.	Llena los datos del usuario.	Valida que los campos estén completos y verifica que los datos sean correctos.
3.		Adiciona el nuevo usuario a la aplicación.
Flujos alternos		
1 Los campos están vacíos		
	Actor	Sistema
1.	Deja campos vacíos	Muestra mensaje de error donde le comunica al usuario que no puede dejar campos obligatorios vacíos
2.		
Sección 2: "Editar usuario"		
Flujo básico Editar usuario		
	Actor	Sistema

1.	Selecciona opción editar usuario que se encuentra a la derecha de la lista de cada usuario.	Muestra formulario con los campos editables activos.
2.	Cambia los datos deseados.	Valida que los campos estén completos.
3.		Guarda los cambios en el sistema.
Flujos alternos		
1 Los campos están vacíos		
	Actor	Sistema
1.	Deja campos vacíos	Muestra mensaje de error.
Sección 3: “Ver detalles_ usuario”		
Flujo básico Visualizar usuario		
	Actor	Sistema
1.	Selecciona opción Ver Detalles del usuario que se encuentra situado al lado derecho de cada usuario que se encuentra en el sistema.	Muestra un formulario con los datos del usuario seleccionado.
2.		
Sección 4: “Eliminar usuario”		
Flujo básico Eliminar usuario		
	Actor	Sistema
1.	Selecciona el usuario a eliminar.	
2.	Selecciona la opción Eliminar, la cual aparece en la parte superior de la aplicación o en un icono posicionado al lado derecho de cada usuario.	Responderá con un mensaje de confirmación de eliminación.
3.	Confirma la eliminación del usuario.	Elimina el usuario del sistema.
Sección 5: “Cambiar_ Contraseña usuario”		
Flujo básico Cambiar_ Contraseña usuario		
	Actor	Sistema
1	El usuario una vez registrado en el sistema selecciona el ícono correspondiente a Cambiar Contraseña, opción particular de cada usuario.	Muestra un formulario con los datos: Usuario, Contraseña nueva, Confirmar contraseña.
2	Llena los datos	Valida que los campos estén completos y correctos.
3		Responderá con un mensaje de confirmación, si desea cambiar la contraseña.
4	Confirma la respuesta del sistema	
Flujos alternos		
1 Los campos están vacíos		
	Actor	Sistema
1	Deja campos vacíos	El sistema muestra mensaje de error.
2		
2 Acción de cambiar contraseña cancelada		
1	Cancela la acción de cambiar contraseña	Retorna a la interfaz de gestión de usuario.

2		
Sección 6 "Listar usuario."		
Flujo básico Listar _usuario.		
	Actor	Sistema
1	El administrador selecciona la opción Gestionar usuario.	El sistema muestra una interfaz con todos los usuarios que tiene agregado.
		Termina el caso de uso.

Tabla 5 Descripción del CU Gestionar usuario

CU Gestionar Grupo

Objetivo	Gestionar grupos de usuarios del sistema.	
Actores	Administrador: (Inicia) Adiciona, edita, muestra usuarios de un grupo, asigna usuarios a un grupo, quita usuarios a un grupo y elimina los grupos de usuarios en el sistema.	
Resumen	El CU permite realizar las operaciones básicas de un CRUD sobre los grupos de usuarios del sistema.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Administrador del sistema autenticado.	
Postcondiciones	Grupos de usuarios creados, editados, datos visibles o eliminados.	
Flujo de eventos		
Flujo básico Gestionar grupos.		
	Actor	Sistema
1.	Selecciona opción gestionar grupo.	<p>Permite realizar varias acciones con un grupo:</p> <ul style="list-style-type: none"> ✓ Adicionar grupo. Ver Sección 1:"Adicionar grupo". ✓ Editar grupo. Ver Sección 2:"Editar grupo". ✓ Visualizar grupo. Ver Sección 3:"Listar_ grupo". ✓ Eliminar grupo. Ver Sección 4:"Eliminar grupo". ✓ Mostrar usuarios de un grupo. Ver Sección5 "Mostrar usuarios de un grupo" ✓ Asignar usuarios a un grupo Sección6 "Asignar usuarios a un grupo" ✓ Quitar usuarios a un grupo Sección7 "Quitar usuarios a un grupo"

Flujos alternos		
1 Gestión de grupos cancelada.		
	Actor	Sistema
1.	Cancela la gestión de grupo.	Retorna a la interfaz de administración.
Sección 1: “Adicionar grupo”		
Flujo básico Adicionar grupo		
	Actor	Sistema
1	Selecciona opción Nuevo_ grupo.	Muestra el formulario con el campo Nombre y el del Identificador para introducir los datos del nuevo grupo.
2	Llena los datos del grupo.	Valida que los campos estén completos.
3		Adiciona el nuevo grupo al sistema.
Flujos alternos		
1 Los campos están vacíos		
	Actor	Sistema
1.	Deja campos vacíos	Muestra mensaje de error.
2.		
Sección 2: “Editar grupo”		
Flujo básico Editar grupo		
	Actor	Sistema
1.	Selecciona opción editar grupo.	Muestra formulario con los campos editables activos.
2.	Cambia los datos deseados.	Valida que los campos estén completos.
3.		Guarda los cambios en el sistema.
4.		
Flujos alternos		
1 Los campos están vacíos		
	Actor	Sistema
1.	Deja campos vacíos	Muestra mensaje de error.
2.		
Sección 3: “Listar_ grupo”		
Flujo básico Visualizar grupo		
	Actor	Sistema
1.	Selecciona opción Gestionar Grupo	Muestra el listado de grupos existentes.
2.		
Sección 4: “Eliminar grupo”		
Flujo básico Eliminar grupo		
	Actor	Sistema
1.	Selecciona el grupo a eliminar.	Marca el grupo.
2.	Selecciona la opción eliminar.	Muestra mensaje de confirmación.
3.	Confirma la eliminación del grupo.	Elimina el grupo del sistema.

Sección 5 “Mostrar Usuarios de un grupo”		
Flujo básico Mostrar Usuarios de un grupo		
	Actor	Sistema
1	Accede a la gestión de grupos.	Muestra todos los grupos existentes.
2	Selecciona el grupo deseado.	Muestra el listado de los usuarios del grupo.
Sección 6 Asignar Usuarios a un grupo		
Flujo básico Asignar Usuarios a un grupo		
	Actor	Sistema
1	Accede a la gestión de usuarios.	Muestra todos los usuarios existentes.
2	Selecciona el usuario deseado y elige añadir a un grupo.	Muestra una interfaz para seleccionar el grupo al cual pertenecerá el usuario seleccionado.
3	Selecciona el/los grupos.	Asigna el usuario al grupo.
Sección 7 Quitar Usuarios de un grupo		
Flujo básico Quitar Usuarios de un grupo		
	Actor	Sistema
1	Accede a la gestión de grupos.	Muestra todos los grupos existentes.
2	Selecciona el grupo en que se encuentra el usuario a eliminar, lo elige y selecciona la opción eliminar.	Muestra un mensaje de confirmación
3	Acepta la confirmación	Elimina el usuario seleccionado del grupo.

Tabla 6 Descripción del CU Gestionar grupo

CU Gestionar Clasificaciones

Objetivo	Gestionar clasificaciones de los contenidos del sistema.	
Actores	Administrador: (Inicia) Adiciona, edita, muestra y elimina las clasificaciones de los contenidos en el sistema.	
Resumen	El CU permite realizar las operaciones básicas de un CRUD sobre las clasificaciones de los contenidos del sistema.	
Complejidad	Alta	
Prioridad	Media	
Precondiciones	Administrador del sistema autenticado.	
Postcondiciones	Clasificaciones creadas, editadas, clasificaciones visibles o eliminadas.	
Flujo de eventos		
Flujo básico Gestionar clasificaciones.		
	Actor	Sistema
1	Selecciona opción gestionar clasificaciones.	Permite realizar varias acciones con una clasificación:

		<ul style="list-style-type: none"> ✓ Adicionar clasificación. Ver Sección 1: "Adicionar clasificación". ✓ Editar clasificación. Ver Sección 2: "Editar clasificación". ✓ Mostrar clasificación. Ver Sección 3: "Mostrar clasificación". ✓ Eliminar clasificación. Ver Sección 4: "Eliminar clasificación".
Flujos alternos		
1 Gestión de clasificaciones cancelada.		
	Actor	Sistema
1	Cancela la gestión de clasificación.	Retorna a la interfaz de clasificación.
2		
Sección 1: "Adicionar clasificación o Taxonomía "		
Flujo básico Adicionar grupo		
	Actor	Sistema
1	Selecciona opción adicionar clasificación.	Muestra formulario con los campos correspondientes a la nueva clasificación.
2	Llena los datos	Valida que los campos estén completos.
3		Adiciona la nueva clasificación al sistema
Flujos alternos		
1 Los campos están vacíos		
	Actor	Sistema
1.	Deja campos vacíos	Muestra mensaje de error.
Sección 2: "Editar clasificación"		
Flujo básico Editar clasificación		
	Actor	Sistema
1.	Selecciona opción editar clasificación.	Muestra formulario con los campos editables activos.
2.	Cambia los datos deseados.	Valida que los campos estén completos.
3.		Guarda los cambios en el sistema.
Flujos alternos		
1 Los campos están vacíos		
	Actor	Sistema
1.	Deja campos vacíos	Muestra mensaje de error.
Sección 3: "Mostrar clasificación"		
Flujo básico Mostrar clasificación		
	Actor	Sistema
1.	Selecciona opción Mostrar _clasificación.	Muestra el listado de clasificaciones

		existentes.
Sección 4: "Eliminar clasificación"		
Flujo básico Eliminar clasificación		
	Actor	Sistema
1.	Selecciona la clasificación a eliminar.	Marca la taxonomía.
2.	Selecciona la opción eliminar.	Muestra mensaje de confirmación.
3.	Confirma la eliminación de la taxonomía.	Elimina la clasificación del sistema.

Tabla 7 Descripción del CU Gestionar clasificación.

CU Realizar búsqueda simple

Objetivo	Realizar búsqueda simple.	
Actores	Administrador: (Inicia) Realiza una búsqueda simple.	
Resumen	El CU se inicia cuando un usuario desea realizar una búsqueda, el sistema muestra la interfaz correspondiente con la opción Buscar, finaliza cuando se muestran los resultados.	
Complejidad	Media	
Prioridad	Crítico	
Precondiciones	Que el usuario introduzca un criterio de búsqueda	
Postcondiciones	Se muestra la información de acuerdo al criterio de búsqueda introducido.	
Flujo de eventos		
Flujo básico Realizar búsqueda simple.		
	Actor	Sistema
1.	Indica el criterio de búsqueda por imágenes, galerías temáticas, grupos, usuarios, y espacio documental y oprime el botón Buscar.	Muestra la información correspondiente al criterio seleccionado.
2.		Muestra un paginado cuando se encuentren varios resultados que no se puedan ver todos en la primera página, lo que le permite al usuario acceder a otras páginas de una forma más rápida y eficiente.

Tabla 8 Descripción del CU Realizar búsqueda simple

CU Realizar búsqueda avanzada

Objetivo	Realizar búsquedas avanzadas de información específica en la base de datos del sistema.	
Actores	Administrador: (Inicia) Realiza una búsqueda avanzada.	
Resumen	El CU se inicia cuando un usuario desea realizar una búsqueda avanzada, el sistema muestra la interfaz correspondiente con la opción Búsqueda Avanzada, el usuario realiza la búsqueda y se muestran los resultados obtenidos finaliza así el CU.	
Complejidad	Alta	
Prioridad	Secundario	
Precondiciones		
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.	Selecciona por lo que desea buscar ya sea: imágenes, galerías temáticas, grupos, usuarios, y espacio documental y selecciona la opción Búsqueda Avanzada.	El sistema muestra diferentes campos correspondiente con los criterios por los cuales es posible realizar la búsqueda.
2.	Introduce los datos.	
3.	Presiona el botón Buscar.	El sistema muestra los resultados en un listado paginado de acuerdo a la configuración personal del usuario.

Tabla 9 Descripción del CU Realizar búsqueda avanzada.

2.5 Propuesta de arquitectura

Para el desarrollo de la interfaz del subsistema de administración se propone que se utilice la arquitectura “por capas”. Teniendo en cuenta que el sistema se conectará al ECM Alfresco, el cual utiliza una arquitectura compuesta por una serie de componentes divididos en capas que tienen a su cargo diferentes funciones. Esta arquitectura define cómo organizar el modelo de diseño a través de capas que pueden ser físicamente distribuidas. Además que simplifica la comprensión y la organización del desarrollo del sistema. La arquitectura de Alfresco es la típica de una aplicación web basada en JAVA (ver Figura 3).

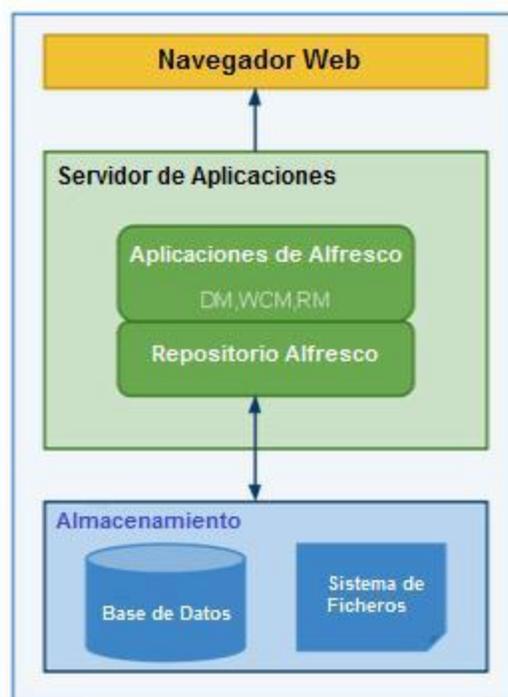


Figura 4 Arquitectura del ECM Alfresco.

Esta arquitectura está dividida en tres capas diferentes:

- ✓ Clientes: Le ofrecen al usuario una interfaz a la solución.
- ✓ Servidores de aplicaciones: Proporciona los servicios de gestión de contenidos y almacenamiento y procesan los datos para los clientes.
- ✓ Servidores de la base de datos: Almacenan los datos para los servidores de aplicación.

Es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y en caso de que ocurra algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado (Ecured, 2013).

La arquitectura del subsistema de administración está conformada por varias capas:

Capa de Presentación: Esta capa es la que el usuario puede ver, a la cual tiene acceso, les permite al cliente y la aplicación establecer la comunicación, manipular los datos, así como representar en términos

de componentes visuales. Reúne todos los aspectos del software que tienen que ver con las interfaces. Además es responsable de presentar información e interactuar con capas inferiores y desde donde se realizan las peticiones al núcleo mediante los servicios web.

Capa de Aplicaciones: En esta capa se encuentra implementado el núcleo, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina también capa de negocio. Desde donde se controla y dirige el flujo de la aplicación.

Capa de acceso a datos: Es donde residen los datos y es la encargada de acceder a los mismos. Recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio. Desde esta etapa se interactúa directamente con los datos contenidos en el núcleo que provee el Alfresco, permitiéndole a la capa de aplicación abstraerse a la forma en que deben recuperar los datos.

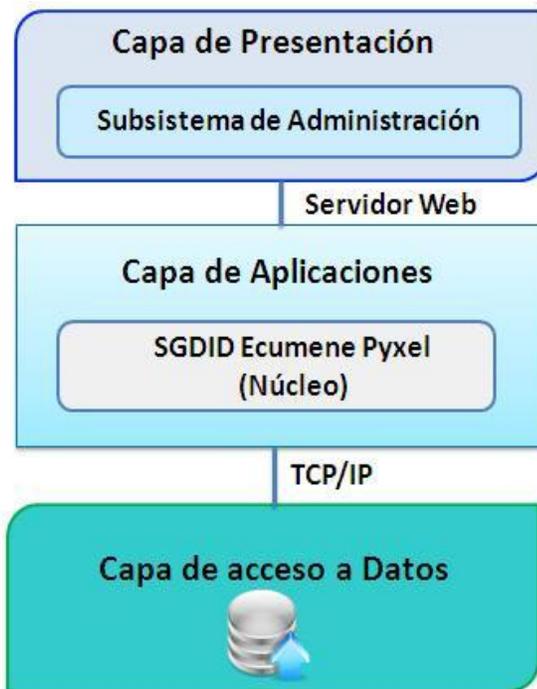


Figura 5 Arquitectura del Subsistema de Administración.

2.5.1 Patrones Arquitectónicos

El patrón arquitectónico seleccionado para el desarrollo de las interfaces de usuarios de la aplicación es el Modelo Vista Controlador (MVC), debido a las ventajas que este posee como la reutilización y la flexibilidad. Además Symfony 2 toma lo mejor de la arquitectura MVC para la implementación y la organización del código, de manera que el desarrollo de la aplicación sea más rápido y sencillo. Este patrón de arquitectura de software separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos, con la finalidad de mejorar la reusabilidad. De esta forma, las modificaciones en las vistas impactan en menor medida en la lógica de negocio o los datos.

Modelo: Esta es la representación específica de la información sobre la cual funciona el sistema. Responde a las peticiones que realizan las vistas sobre su estado y permite su actualización normalmente desde el controlador. Representa cada una de las clases que son generadas según la estructura de la base de datos.

Vista: Representa el modelo en un formato adecuado para interactuar, usualmente interfaz de usuario. Interpreta las acciones del usuario, accediendo a las operaciones de negocio de la aplicación y modificando a partir de sus resultados el estado del modelo y la navegación entre vistas.

Controlador: Este responde a eventos, usualmente acciones del usuario, que son la lógica de la aplicación e invoca peticiones al modelo y a la vista.

2.5.2 Patrones de Diseño

Son aquellos que expresan un esquema para definir estructuras de diseño o sus relaciones con las que construir sistemas de software. Brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.

Se plantean como una buena herramienta para el diseño y la documentación de aplicaciones y *frameworks*. Son descripciones de objetos y clases que se comunican y que son capaces de solucionar un problema de diseño en general, en un contexto en particular (Barotto, 2005).

Symfony sigue las mejores prácticas y patrones de diseño para el desarrollo de aplicaciones Web. Seguidamente se mencionan cuáles fueron los patrones de diseño más significativos que se utilizan.

2.5.3 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. A continuación se describirán los patrones que se evidencian en el desarrollo del sistema.

Experto: Este es uno de los patrones que se utilizan al trabajar con el *framework* de desarrollo Symfony. Genera las clases para la gestión de las entidades con las responsabilidades debidamente asignadas según el patrón Experto, pues cada una de estas clases cuenta con un conjunto de funcionalidades relacionadas directamente con la entidad que representan. El subsistema de administración está estructurado por diferentes componentes y dentro de estos se encuentran las clases correspondientes para las funciones pertinentes, UsuarioController es la responsable de realizar todas las operaciones correspondientes con la entidad Usuario.

Alta cohesión: Una de las características principales de Symfony es la organización del trabajo en cuanto a la estructura del proyecto, lo cual permite crear y trabajar con clases con una alta cohesión. Por ejemplo las clases con sufijo Controller contienen varias funcionalidades estrechamente relacionadas entre ellas, teniendo un sentido común y un propósito único, siendo las encargadas de controlar las acciones de cada entidad y por lo tanto pertenecen a la capa del Controlador dentro de la arquitectura Modelo-Vista-Controlador. Esto hace posible que el software sea flexible a cambios sustanciales con efecto mínimo.

2.5.4 Patrones GOF

Los patrones GOF constituyen un catálogo de 23 patrones de diseño básicos publicados por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides (1995) en el libro *Design Patterns: Elements of Reusable Object-Oriented Software*. GOF significa *Gang of Four* (Banda de los Cuatro), forma en que se conoce al grupo de autores (Peraza, 2012).

Estos patrones tienen las siguientes características (Pérez, 2012).

- Son soluciones concretas. Proponen soluciones a problemas concretos, no son teorías genéricas.
- Se utilizan en situaciones frecuentes debido a que se basan en la experiencia acumulada al resolver problemas reiterativos.

- Favorecen la reutilización de código. Ayudan a construir software basado en la reutilización, a construir clases reutilizables. Los propios patrones se reutilizan cada vez que se vuelven a aplicar.
- Es difícil reutilizar la implementación de un patrón. Al aplicar un patrón aparecen clases concretas que solucionan un problema concreto y que no será aplicable a otros problemas que requieran el mismo patrón.

Los patrones GOF se clasifican en 3 categorías: creacionales, estructurales y de comportamiento.

- **Creacionales:** Abstraen el proceso de creación de instancias y ocultan los detalles de cómo los objetos son creados o inicializados.
- **Estructurales:** Se ocupan de cómo las clases y objetos se combinan para formar grandes estructuras y proporcionar nuevas funcionalidades.
- **Comportamiento:** Están relacionados con los algoritmos y la asignación de responsabilidades entre los objetos. Son utilizados para organizar, manejar y combinar comportamientos.

Solitario: Es un patrón de Creación, propone una sola instancia de la clase controladora para acceder a los datos y modificarlos, al igual que el *framework* usa una sola instancia para manejar las sesiones, los Layouts y cualquier otra información que se requiera.

Decorador: Es un patrón estructural, que permite añadir funcionalidades dinámicamente a una clase con la creación de otra clase sin necesidad de la herencia. En el Symfony, los layouts decoran a las vistas, pues en un momento determinado les confieren propiedades dinámicas que no poseían. Este método pertenece a las clases View, que contiene un decorador para permitir agregar funcionalidades dinámicamente. El archivo nombrado base.html es usual en todas las páginas del sistema para no tener que repetirlo en cada página y se encarga de decorar las plantillas.

2.6 Diseño de la interfaz de administración para el SGDID Ecumene Pyxel

A continuación se muestra el diagrama de clases del diseño correspondiente a la funcionalidad Gestionar usuario y los diagramas de secuencia propios de dicha funcionalidad.

2.6.1 Diagrama de clases del diseño

Los diagramas de clases de diseño simulan una construcción similar a la implementación del sistema, representando gráficamente la jerarquía y las relaciones que se generan a partir de cada una de las

clases.

Para la representación de los diagramas correspondientes a la aplicación se tuvo en cuenta la organización interna de Symfony y la estructura de enrutamiento, la que basa su funcionamiento en la arquitectura MVC.

Esquema simplificado de la arquitectura interna de Symfony2

1. El sistema de enrutamiento determina qué Controlador está asociado con la página de la portada.
2. Symfony2 ejecuta el Controlador asociado a la portada. Un controlador no es más que una clase PHP en la que se puede ejecutar el código deseado.
3. El Controlador solicita al Modelo los datos. El modelo no es más que una clase PHP especializada en obtener información, normalmente de una base de datos.
4. Con los datos devueltos por el Modelo, el Controlador solicita a la Vista que cree una página mediante una plantilla y que inserte los datos del Modelo.
5. El Controlador entrega al servidor la página creada por la Vista.

Una vez descrita la estructura interna del framework se diseñaron los diagramas de clase del diseño guiado por el funcionamiento interno del mismo.

1. La aplicación redirige la petición al script `app.php`. Este archivo se denomina controlador frontal y es el único punto de entrada a la aplicación.
2. El controlador frontal crea el kernel de la aplicación mediante una instancia de la clase `app/AppKernel.php`.
3. El kernel registra todos los bundles activos de la aplicación e importa el archivo de configuración `app/config/config.yml` que carga las rutas del archivo `app/config/routing.yml`.
4. El controlador frontal crea un objeto de tipo `Request` rellenándolo con la información de la petición del usuario y se lo pasa al kernel.
5. Si el controlador devuelve un objeto de tipo `Response`, el kernel lo transforma en la página que se devuelve al usuario. Si el controlador devuelve el nombre de una plantilla, el kernel hace uso de la vista para renderizar la plantilla según el motor de plantillas configurado en la aplicación (Twig o PHP).
6. La ejecución finaliza devolviendo al usuario la página creada a partir de la plantilla.

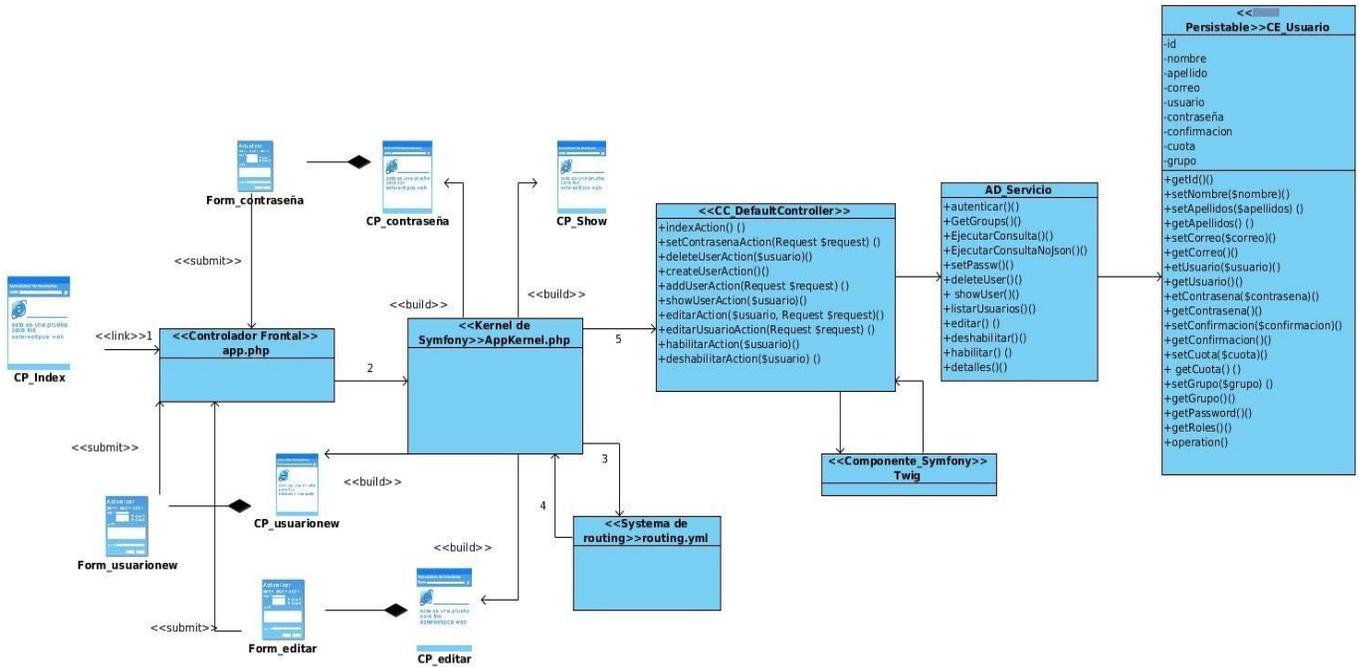


Figura 6 Diagrama de clases del diseño Gestionar usuario

2.6.2 Diagramas de Secuencia

Seguidamente se muestran las descripciones de los principales casos de uso del sistema, mostrando detalladamente cómo se manifestó la implementación de cada uno de ellos. Estos diagramas muestran la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso, mostrando de forma secuencial los envíos de mensajes entre objetos. El diagrama puede así mismo mostrar los flujos de datos intercambiados durante el envío de mensajes.

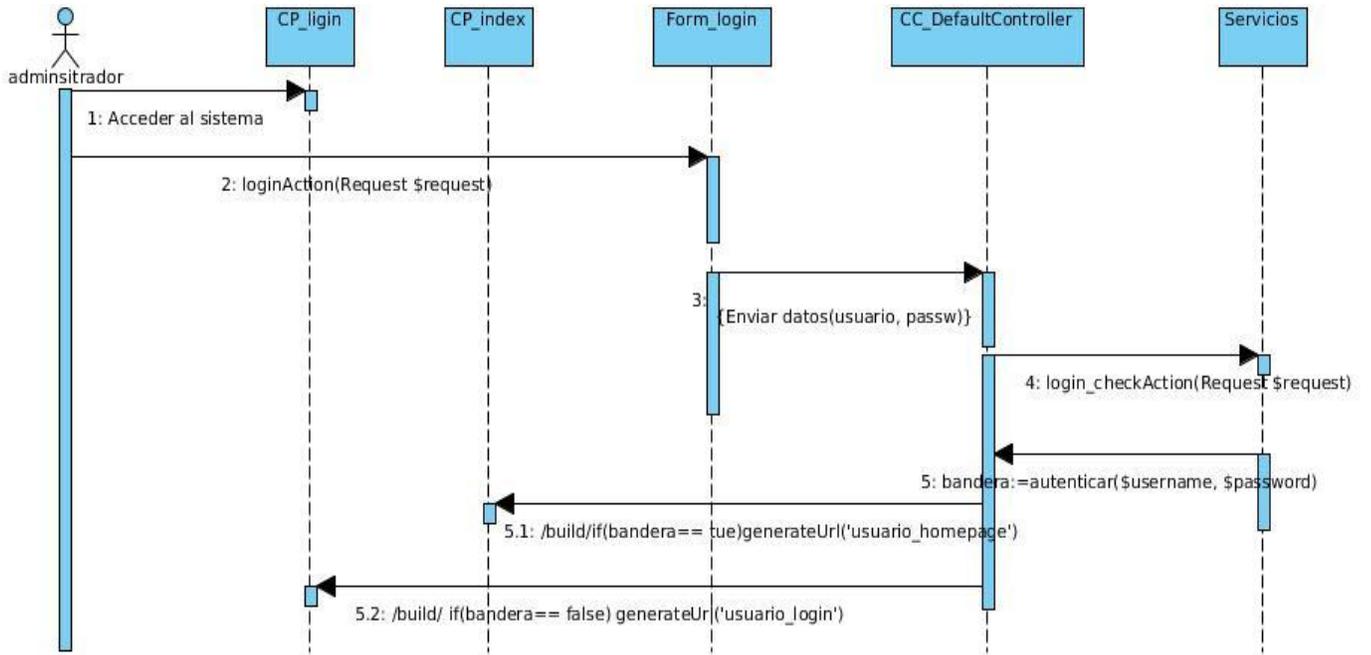


Figura 7 Diagrama de secuencia Autenticar Usuario

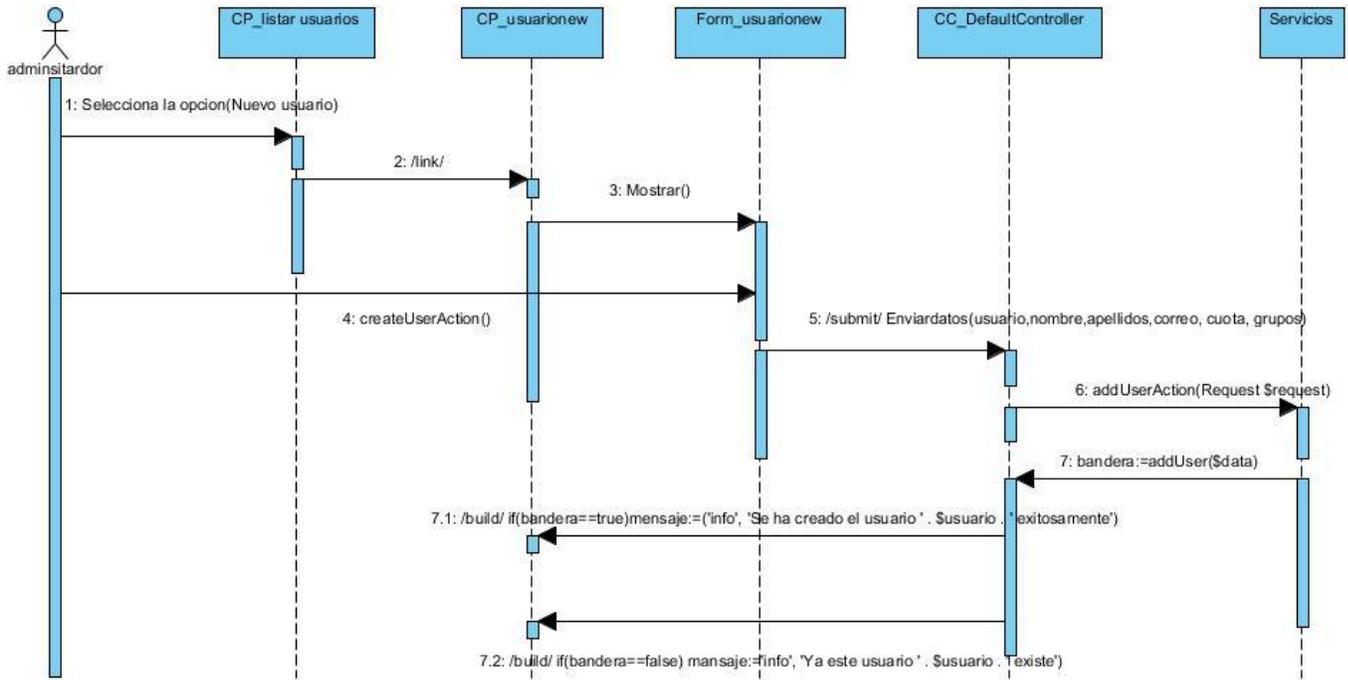


Figura 8 Diagrama de secuencia Agregar Usuario

2.7 Diagrama de despliegue

El modelo de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución. El propósito de este modelo es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema. Se utiliza para visualizar la distribución de los componentes del software en los nodos físicos.

A continuación se muestra el diagrama de despliegue propuesto para el subsistema de administración.

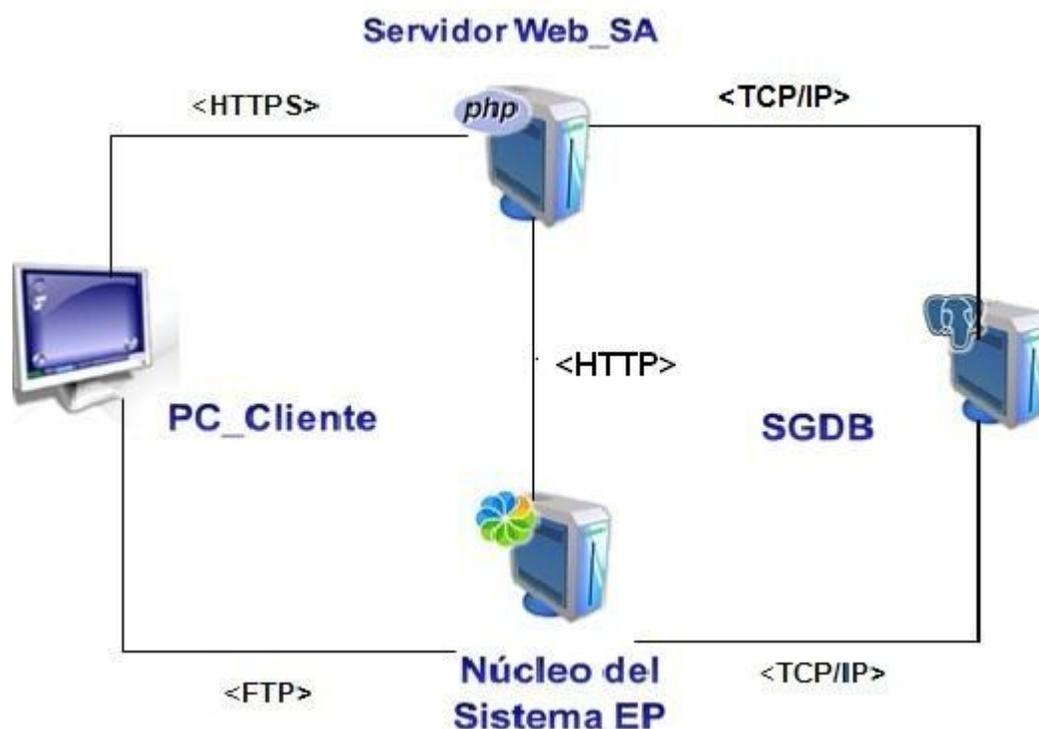


Figura 9 Representación del modelo de desde la aplicación

2.7.2 Descripción de los componentes

La aplicación consta con una PC¹⁷ del lado del cliente, que accederá al servidor web subsistema de administración mediante un navegador web y empleará el protocolo HTTPS¹⁸ y al núcleo del sistema Ecumene Pyxel por el protocolo FTP¹⁹. El servidor web subsistema de administración podrá comunicarse con el núcleo del sistema a través el protocolo HTTP²⁰ y consumir los servicios web que este brinda. Cuenta con un servidor donde se encuentra el sistema gestor de base de datos (SGDB) que contendrá todos los datos necesarios para la aplicación, a este servidor podrán conectarse tanto el núcleo del

¹⁷ Computadora personal.

¹⁸ Protocolo seguro para la transferencia de hipertexto, un protocolo seguro que garantiza una comunicación autenticada y cifrada.

¹⁹ Por sus siglas en Inglés *File Transfer Protocol*, es un protocolo de red (en español llamado Protocolo de Transferencia de Archivos).

²⁰ Por sus siglas en las siglas inglés *HiperText Transfer Protocol* (en español protocolo de transferencia de hipertexto). Es un protocolo de red.

sistema como la interfaz web mediante el protocolo TCP/IP²¹.

2.8 Conclusiones parciales

En este capítulo se desarrollaron artefactos importantes que corresponden con el flujo de trabajo Análisis y Diseño. Además se analizaron las características del sistema a desarrollar y se estableció la arquitectura más adecuada para la representación del proceso. A partir de los casos de uso se modeló el diseño del sistema con los artefactos fundamentales tales como, diagrama de clases del diseño, diagrama de colaboración, diagrama de despliegue y la descripción de cada uno de ellos. Una vez terminada la etapa se propone reunir todos los resultados obtenidos para guiar una futura implementación de la solución propuesta.

²¹ Por sus siglas en inglés, Transmission Control Protocol/Internet Protocol (en español protocolo de Control de Transmisión/Protocolo de Internet). Modelo de descripción de protocolos de red.

Capítulo 3 Implementación y Prueba

En el presente capítulo se describe el diagrama de componentes, con el objetivo de reflejar la estructura del sistema. Además se identifica el estándar de codificación utilizado y se representan algunas de las pantallas principales de la aplicación. Se presentan los casos de prueba y las validaciones realizadas al sistema en cada una de las iteraciones, a fin de asegurar que el sistema cumpla con las funcionalidades que precisa el cliente y que tenga la calidad requerida. Además se reflejan los resultados obtenidos después de realizadas las pruebas.

3.1 Estructura de implementación del subsistema de administración

El diagrama de componentes refleja cómo está dividido el sistema por componentes, así como la relación entre estos. Además representan la organización y las dependencias entre los componentes de un sistema. Un componente es una parte física, en él se representan (librerías, archivos, ejecutables, tablas) es decir, la materialización de una o más clases.

Elementos del diagrama de componentes

- ✓ Componentes
- ✓ Paquetes o subsistemas
- ✓ Relaciones o dependencias
- ✓ Interfaces

A continuación se muestra la representación del diagrama de componentes de la interfaz del subsistema de administración.

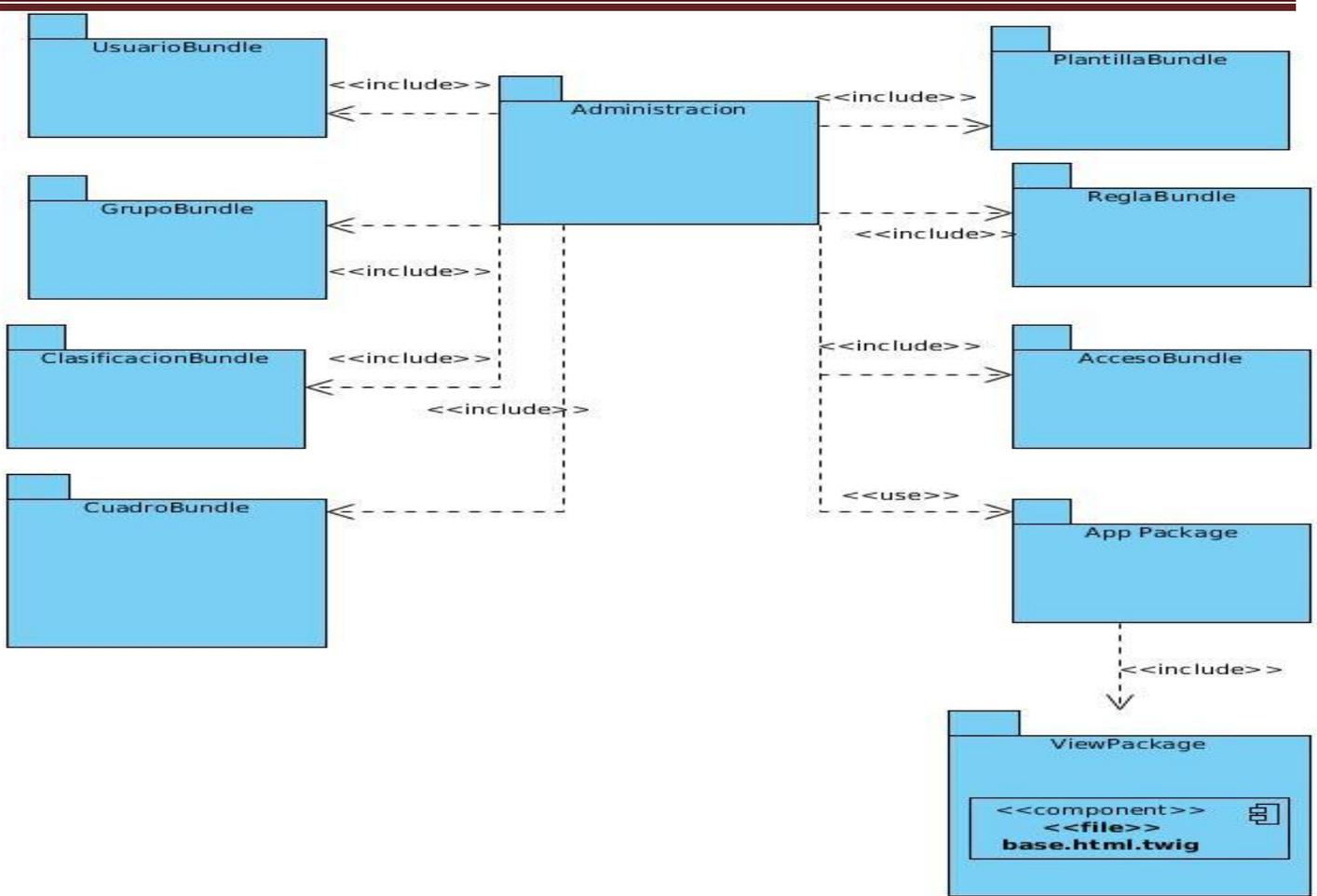


Figura 10 Representación del diagrama de componentes del subsistema de administración

Para ver con más detalles de cómo está organizada la aplicación remítase a los Anexos.

3.1.1 Descripción de los componentes

El diagrama representa los principales componentes que se utilizan para el correcto funcionamiento de la aplicación, se definen los diferentes paquetes que dividen la lógica, la vista, la comunicación con el núcleo y la configuración del sistema.

- ✓ En el paquete Administración se agrupan todos los componentes, para determinar una estructura organizativa del producto.
- ✓ Los paquetes con sufijo Bundles son contenedores de información, dentro de ellos se almacenan todos los datos correspondientes con determinada entidad y dentro de estos existen otros paquetes destinados para determinadas operaciones.

- ✓ El paquete App se denomina controlador frontal y es el único punto de entrada a la aplicación. Incluye varios paquetes este incluye el paquete View donde se almacenan los componentes de diseño de la aplicación, este contiene el componente base.html.twig.

3.2 Estándares de codificación

Los estándares de código resultan importantes en cualquier proyecto de desarrollo, pero son especialmente imprescindibles cuando muchos desarrolladores trabajan en el mismo proyecto. Los estándares de código ayudan a asegurar que el código tenga una alta calidad, menos errores, y pueda ser mantenido fácilmente.

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código óptimo es de gran importancia para la calidad del software y para obtener un buen rendimiento y mantenimiento del mismo.

El estándar de codificación seleccionado fue definido por la dirección del proyecto GD Prensa que lleva por título "Estándares de codificación para PHP" que se encuentra registrado en el expediente de proyecto.

3.3 Interfaces principales de la aplicación

A continuación se muestran algunas de las interfaces principales de la aplicación desarrollada.



Figura 11 Autenticación en el sistema

Después que el usuario se autentica satisfactoriamente podrá acceder a la página principal de la aplicación donde se encuentran los espacios disponibles para el administrador.



Figura 12 Espacio de usuario

3.4 Pruebas de software

Según Roger Pressman en *“Ingeniería del software un enfoque práctico”* plantea que las pruebas de software son un elemento crítico para la garantía de la calidad del sistema. Las pruebas permiten validar y verificar el producto obtenido. Conocidas también como técnicas de evaluación, representan una revisión final de las especificaciones del diseño y de la implementación. Su principal objetivo es diseñar pruebas que, sistemáticamente, saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y de esfuerzo. Las pruebas no pueden asegurar la ausencia de errores; sólo puede demostrar que existen defectos en el software.

3.4.1 Métodos de pruebas

Los métodos de prueba más usados en la validación de un software de manera general tienen dos enfoques diferentes: prueba de caja blanca y prueba de caja negra. La primera de estas pruebas basa su funcionamiento en evaluar la información sobre cómo el software ha sido diseñado y codificado, mientras que el método caja negra actúa sobre la validación de los requisitos funcionales. Se aplican a la interfaz del software para examinar algún aspecto funcional de un sistema.

Criterios de Caja Negra

1. Particiones de Equivalencia.
2. Análisis de Valores Límite.

Partición equivalente: Se dirige a una definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Este método intenta dividir el dominio de entrada de un programa en un número. El diseño de casos de prueba según esta técnica consta de dos pasos:

1. Identificar las clases de equivalencia: Una clase de equivalencia representa un conjunto de estados válidos y no válidos para las condiciones de entrada de un programa.
2. Identificar los casos de prueba finitos de clases de equivalencia: El objetivo es minimizar el número de casos de prueba, así cada caso de prueba debe considerar tantas condiciones de entrada como sea posible.

La técnica de caja negra intenta encontrar errores de las siguientes categorías:

- ✓ Funciones incorrectas o ausentes.

-
- ✓ Errores de interfaz.
 - ✓ Errores en estructuras de datos o en accesos a bases de datos externas.
 - ✓ Errores de rendimiento y errores de inicialización y de terminación

Para desarrollar este tipo de pruebas se hace necesaria la presencia de un buen diseño de casos de prueba. Estos recogen todas las combinaciones de escenarios posibles a ejecutar, contienen la descripción de todas las variables o datos de entrada al sistema. Son un conjunto de condiciones mediante las cuales se comprueba si los requisitos funcionan correctamente (B, 2008).

3.4.1 Estrategia de prueba

Para evaluar dinámicamente el sistema se trazó una estrategia de prueba basado en las pruebas funcionales, de integración y de seguridad, con el objetivo de conocer el nivel de exactitud, seguridad, rapidez y fiabilidad que la aplicación pudiera tener.

Primeramente se analizó el funcionamiento aislado del subsistema sin ser integrado al Núcleo, para comprobar que se cumplieran los requisitos funcionales en su totalidad. Luego se realizaron las pruebas de integración cuya función fue integrar el subsistema al núcleo del SGDID, así las funciones probadas se vuelven a examinar combinadas con el sistema completo para verificar sus interfaces.

3.4.2 Pruebas funcionales

Se denominan pruebas funcionales, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados cumplan con las funciones específicas para los cuales han sido creados, es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales (B, 2008).

Para la realización de las pruebas funcionales del subsistema de administración para el SGDID Ecumene Pyxel se determinó escoger la técnica de “caja negra” guiada por el criterio de partición de equivalencia.

A continuación se muestra como ejemplo los diseños de casos de prueba funcionales realizados a las funcionalidades del subsistema de administración para el SGDID Ecumene Pyxel, específicamente del caso de uso Autenticar usuario y Gestionar usuario. El resto de los casos de prueba se pueden encontrar en los Anexos y en la planilla Diseño de Casos de Prueba basado en caso de uso, la que se encuentra en el Expediente de Proyecto del subsistema de administración para el SGDID Ecumene Pyxel.

Nombre del Caso de uso	Descripción general	Escenarios	Flujo central
Gestionar usuario	Permite adicionar, eliminar, editar y cambiar la contraseña de un usuario determinado.	SC1Adicionar usuario	El usuario entra todos los datos correctamente y presiona el botón Adicionar.
		SC2Editar usuario	Se selecciona el usuario que se desea editar y se presiona la opción editar usuario y luego se procede a cambiar los campos necesarios y presionar el botón Editar o Cancelar.
		SC3Eliminar usuario	Se selecciona el usuario que se desea eliminar y se presiona la opción eliminar usuario y presionar el botón Aceptar para confirmar la acción o Cancelar en caso contrario.
		SC4Cambiar contraseña	Se selecciona el usuario al cual se le desea cambiar la contraseña y se presiona la opción Cambiar contraseña y luego se procede a cambiar los campos necesarios y presionar el botón Cambiar o Volver.

Tabla 10 Descripción del flujo central del Caso de Uso Gestionar Usuario

Descripción General: Le permite al usuario administrador registrarse en el sistema, para ello se muestra una interfaz con los datos para autenticarse.					
Condiciones de ejecución: Para autenticarse en el sistema el usuario debe introducir en el navegador la dirección de la ruta del sistema					
CU1 Autenticar Usuario					
	Descripción	Usuario	contraseña	Respuesta del sistema	Flujo central
EC1.1 Datos Correctos	Se muestra la interfaz de autenticación para que el usuario se autentique en el sistema con los datos correctamente	V	V	El sistema muestra el espacio del usuario.	El usuario escribe en su navegador la dirección donde se encuentra el producto http://ip_servidor, e introduce los datos correctamente y presiona el botón Aceptar
		Admin	admincip		
EC1.2 Campos Vacíos	Se muestra la interfaz de autenticación para que el usuario se autentique en el sistema, pero se dejan campos vacíos	I	I	El sistema muestra un mensaje de error: Debe llenar el campo usuario o : Debe introducir la contraseña	El usuario escribe en su navegador la dirección donde se encuentra el producto http://ip_servidor, e introduce los datos dejando campos en blanco y presiona el botón Aceptar
		" "	" "		
EC 1.3 No hay comunicación con el núcleo	El usuario intenta autenticarse, pero el servidor del núcleo está fuera de servicio	V	V	El sistema muestra un mensaje de error: No hay comunicación con el Alfresco	El usuario escribe en su navegador la dirección donde se encuentra el producto http://ip_servidor, e introduce los datos correctamente y presiona el botón Aceptar
		Admin	epyxelcip		
EC1.4 Datos incorrectos	El usuario intenta autenticarse en el sistema e introduce datos incorrectos	I	V	El sistema no redirecciona a ninguna dirección y muestra el mensaje de error: Usuario incorrecto	El usuario escribe en su navegador la dirección donde se encuentra el producto http://ip_servidor, e introduce los datos correctamente y presiona el botón Aceptar
		Usuario	admincip		
		V	I	El sistema no redirecciona a ninguna dirección y muestra el mensaje de error: Usuario incorrecto	El usuario escribe en su navegador la dirección donde se encuentra el producto http://ip_servidor, e introduce los datos correctamente y presiona el botón Aceptar
		Admin	incorrecta		

Tabla 11 Caso de prueba Autenticar usuario

Descripción General: Permite importar un contenido determinado										
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema con el rol de “administrador”										
SC 1 Adicionar Usuario										
Escenario	Descripción	Nombre	Apellido	Correo	Usuario	Contraseña	Confirmar	Cuota	Grupo	Respuesta del sistema
EC 1.1 Datos correctos	Se muestra una interfaz con todos los datos de un usuario y se llenan correctamente	V	V	V	V	V	V	V	V	V
		alfredo	tur	atur@est.cu	atur	estesta	estesta	10MB	administrador	El sistema adiciona correctamente el usuario nuevo y muestra un mensaje de confirmación: Se ha creado el usuario mhernandez exitosamente
EC 1.2 Campos vacíos	El usuario deja alguno o todos los campos sin llenar	I	I	I	I	I	I	I	I	I
		vacío	vacío	vacío	vacío	vacío	vacío	vacío	vacío	vacío
Ec1.3 Datos incorrectos	El usuario introduce algún valor incorrecto en el sistema	I	V	V	V	V	V	V	V	V
		alfredo2	tur	atur@est.cu	atur	estesta	estesta	4MB	fotógrafo	El sistema muestra el mensaje de error: El nombre debe ser sólo letras
		V	I	V	V	V	V	V	V	V
		mary	23ms	mlic@est.cu	mlic	yosi	yosi	5MB	editor	El sistema responde un mensaje de error: El apellido debe ser sólo letras
		V	V	I	V	V	V	V	V	V
		mary	licourt	@12es.cu	mlic	yosi	yosi	5MB	editor	Responde un mensaje de error: Formato del correo incorrecto
		V	V	V	V	I	V	V	V	V
mary	licourt	mlic@est.cu	mlic	mmm	admin	10MB	administrador	Responde con el mensaje de error : la contraseña no		

			cu							coincide
	V	V	V	V	V	I	V	V		Responde con el mensaje de error : la contraseña no coincide
	mary	licourt	mlic@est.cu	mlic	estesta	xxxx	10MB	editor		

Tabla 12 Descripción del Caso de prueba Adicionar Usuario

Las celdas de la tabla contienen V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

Para la evaluación de los requisitos del sistema se realizaron tres iteraciones de las pruebas no funcionales, con el objetivo de detectar las no conformidades de las interfaces del subsistema de administración, obteniendo como resultados los siguientes (Ver tabla 13):

No conformidades	Primera iteración	Segunda iteración	Tercera iteración
Detectadas	20	10	8
Resueltas	15	5	8
Pendientes	5	5	0

Tabla 13 Número de no conformidades por iteraciones

En el Gráfico1 se muestra el comportamiento de las no conformidades durante las tres iteraciones realizadas (Ver grafico1).

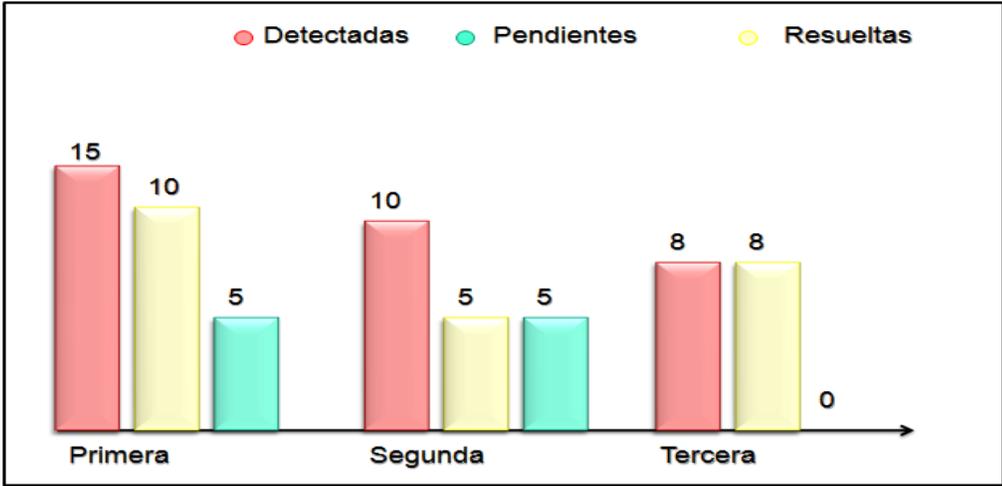


Grafico 1 Comportamiento de las no conformidades por iteraciones

Las principales no conformidades encontradas durante las pruebas funcionales se encuentran registradas en la siguiente tabla (Ver tabla 14).

Iteraciones	No conformidades
Primera	No se muestran los mensajes de error en la autenticación de sistema. Los mensajes no coinciden con los errores identificados.
Segunda	No muestra un mensaje de error si no hay conexión con el núcleo. No se realizan correctamente las validaciones de los datos.

Tabla 14 Principales no conformidades detectadas

3.4.2 Pruebas de seguridad

Las pruebas de seguridad garantizan la protección del software, asegurando la integridad de los datos, de forma tal que solamente los usuarios definidos puedan acceder a la información almacenada según los niveles de acceso. El objetivo de la seguridad es evaluar la confidencialidad, integridad y disponibilidad de los datos. A continuación se muestra una breve descripción de cómo se establece la seguridad del subsistema de administración del SGDID Ecumene Pyxel.

Para el subsistema de administración la seguridad se establece mediante la asignación de roles y la autenticación con el uso de contraseña. En este sentido Symfony implementa una serie de mecanismos para garantizar aplicaciones seguras. Además de la seguridad que proporciona el Symfony 2, se estableció el mecanismo de autenticación al sistema basado en usuario y contraseña. El usuario con el rol de administrador puede realizar las funcionalidades que se determinaron, tendrá todos los permisos para poder eliminar, editar o adicionar otro usuario en el sistema y así será con las demás operaciones. Además para verificar los errores más comunes de seguridad se utilizó la lista de chequeos para pruebas de seguridad cuyo documento contiene información propietaria de ALBET Ingeniería y Sistemas. Después de aplicar este chequeo se detectaron las siguientes no conformidades para una primera iteración de estas pruebas.

Comprobación del Sistema de Autenticación

- ✓ No se bloquea la sesión del usuario después de un tiempo X (establecido por la aplicación) sin usar la misma.
- ✓ No se bloquea la cuenta del usuario los datos después de un número X (establecidos por la aplicación) con intento fallido por el usuario de acceder al sistema.

Pruebas de Gestión de Sesiones:

- ✓ 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.

3.5 Conclusiones parciales

Durante este capítulo se generaron los artefactos correspondientes a los flujos de trabajo de implementación y prueba, tras estos flujos el sistema quedó desarrollado. En los diagramas descritos pudo ilustrarse la relación entre los principales componentes del sistema, lo que permitió una mejor organización y visión del producto. Las pruebas realizadas permitieron aumentar la calidad final de la solución. Además la aplicación final aprovechó las ventajas de Symfony y quedó lista para integrarse al SGDID Ecumene Pyxel.

Conclusiones

A partir de la investigación realizada se arribaron a las siguientes conclusiones:

- ✓ El estudio bibliográfico durante la investigación permitió obtener un conocimiento más profundo de la situación actual y las tendencias sobre los DMS. A través del mismo se demostró la necesidad de desarrollar el subsistema de administración para el SGDID Ecumene Pyxel.
- ✓ La generación de los artefactos y esquemas relacionados con el flujo de análisis y diseño teniendo en cuenta la arquitectura MVC que Symfony establece, permitió definir la arquitectura de información requerida. Además facilitó obtener una mayor comprensión de la propuesta de solución adecuada, lo que permitió alcanzar los principios que guiaron la implementación del sistema.
- ✓ Partiendo del diseño y la aplicación de los casos de prueba al subsistema, se comprobó que las funcionalidades descritas, satisfacen las necesidades en cuanto al proceso de administración en el SGDID Ecumene Pyxel, obteniéndose los resultados esperados.
- ✓ Se logró dar cumplimiento de forma satisfactoria a los objetivos y tareas trazadas al inicio de la investigación.

Recomendaciones

Una vez terminada la investigación y el desarrollo de la aplicación, para tratar de aumentar las ventajas competitivas del SGDID Ecumene Pyxel y la interfaz de administración como parte importante de este, se recomienda:

- ✓ Agregar nuevas funcionalidades a la Interfaz de Administración para el SGDID Ecumene Pyxel tales como:
 - Configurar correo electrónico.
 - Gestionar Plantillas.
- ✓ Perfeccionar el estilo de la aplicación.
- ✓ Seguir con las actualizaciones en el mundo de las tecnologías implementadas que faciliten la interoperabilidad con ECMAIfresco.

Bibliografía

- A, Ernesto Quiñones.** Introducción a PostgreSQL. *Técnica de evaluación de software.*
- B, Ing. Alexander Oré. 2008.** Calidad Software.com. *Calidad Software.com.* [En línea] Nazcasoft.com, 2008. [Citado el: 2 de 4 de 2013.] <http://www.calidadyssoftware.com/testing.php>.
- Brun, Ricardo Brito. 2008.** *Sobre la viabilidad del código abierto. El caso de Alfresco23.* 2008.
- Barotto Vanina, Demonte Mauricio** DEFINICIÓN DE PATRONES DE DISEÑO iDEFINICIÓN DE PATRONES DE DISEÑO [Informe]. - Río Cuarto : [s.n.], 2005.
- Codina, Lluís. 1993.** *El profesional de la información* . [En línea] ISI Social Sciences Citación Índex, Mayo de 1993. [Citado el: 1 de diciembre de 2012.]
- Chapell, David. 2002.** *Java Web Servicios.* 2002.
- ecured. 2009.** IDE de Programación [En línea] ecured, 2009. [Citado el: 8 de 2 de 2013.] (http://www.ecured.cu/index.php/IDE_de_Programación).
- Eguiluz, Javier. 2011.** *Desarrollo web ágil con Symfony2.* 2011.
- Concepto de JavaScript.** *Concepto de JavaScript.* [En línea] [Citado el 01 de 03 de 2013]. https://developer.mozilla.org/index.php/Concepto_de_JavaScript.
- Eldar.** *Sistema de Gestión Documental.* Madrid : s.n., 1991.
- Gallardo Palay, Adrian. 2009.** *Interfaz Web para el sistema de análisis Airesweb.* La Habana : s.n., 2009.
- Hernández León, Rolando Alfredo, Coello González Sayda. 2011.** *El proceso de la investigación científica.* El Vedado, Ciudad de La Habana, : Universitaria del Ministerio de Educación Superior, 2011, 2011. ISBN 978-959-16-.
- Ifresco. iFresco. 2011.** *Ifresco.* [En línea] [Citado el: 2 de diciembre de 2012.] <http://www.ifresco.at/de/products/client/overview.html>.
- Isabel María García Hilarión, Sergio Antonio Segura Fernández. 2012.** *Interfaz Web para el Sistema de Gestión Documental Interfaz* . La Habana : s.n., 2012.
- INTECO, Laboratorio Nacional de Calidad de Software. 2009.** *Ingeniería del Software: Metodologías y Ciclos de Vida.* España : s.n., 2009.
- Ideoinformática, Centro. 2012.** *Configuración de la metodología OpenUP.* Ciudad de La Habana. Cuba : s.n., 2012.
- ifresco.at. 2011.** ifresco_Client. [En línea] ifresco.at, 2011. [Citado el: 25 de 1 de 2013.] www.ifresco.at.
- Juristo Natalia, Ana M. Moreno, Sira Vegas. 2005.** *Tecnnica de evaluación de software.* 2005.

- Jacobson, Ivar. 2000.** *El proceso unificado de desarrollo de software*. Madrid : Addison Wesley, 2000.
- Lien Burgos Pérez, Maylevis Morejón Valdés. 2012.** *Versión 2.0 del sistema informático para la gestión de las solicitudes del Despacho Comercial en la Aduana General de la República de Cuba*. La Habana : s.n., 2012.
- López, Jose Antonio. 2013.** rodeadosdetecnologia.. [En línea] 23 de 1 de 2013. [Citado el: 15 de 2 de 2013.] <http://rodeadosdetecnologias.blogspot.com>.
- MagmaSoft.** *Gestión Documental con Alfresco*. [En línea] [Citado el: 2 de diciembre de 2012.] http://www.imaginar.org/dgd/pres/4_Alfresco.pdf.
- Moreno-Torres, Fernando.** *Gestión documental para gente normal*.. [En línea] [Citado el: 1 de diciembre de 2012.] <http://gestiondocumentalparagentenormal.com/>.
- Mugica Mena, Mayra. 2005.** *La Gestión Documental y Organización de Archivos*. La Habana : Félix Varela, 2005.
- Natalia Juristo, Ana M. Moreno, Sira Vegas. 2005.** 2005.
- Opsoro. 2010.** *Opsoro*. [En línea] 2010. <http://www.opsoro.org/home>.
- Potencier Fabien, Zanionotto François.** *Symfony la Guía definitiva*.
- Pérez, Tereza Garzon. 2010.** *Sistemas Gestores de Bases de Datos*. 2010. 1988 6047.
- Peraza, Danelis Llera. 2012.** *Módulo de la Dirección de Cultura de la Administración Provincial de Artemisa*. Artemisa, Cuba : s.n., 2012.
- Pressman, Roger S.** *Ingeniería del software un enfoque práctico*, 2005.
- Peña, Ing. Yadira Machado.** *Planificación, Diseño y Ejecución de Pruebas Funcionales*. Habana: Calisoft, marzo-2011.
- Potenciar Fabien, Zaninotto Francois. 2008.** *Symfony la guía definitiva*. 2008.
- Pérez Lien Burgos, Valdés Maylevis Morejón. 2012.** *Versión 2.0 del sistema informático para la gestión de las solicitudes del Despacho Comercial en la Aduana General de la República de Cuba*. La Habana : s.n., 2012.
- Project, Alfresco Forge. 2010.** Opsoro. [En línea] 2010. [Citado el: 2 de diciembre de 2012.] <http://www.opsoro.org/architecture>.
- Romero, Gladys. 2008.** *Metodología ágil para proyectos de software libre*. La Habana: UCI, 2008.
- Rotz, B. 2008.** DoCASU. DoCASU. [En línea] EOSDirectory, 2008. [Citado el: 05 de 12 de 2011.] <http://www.eosdirectory.com/blogs/2008/07/15/>
- Software, Alfresco. 2012.** *Alfresco Enterprise 4: la plataforma de contenido conectada en la nube*. *Alfresco Enterprise 4: la plataforma de contenido conectada en la nube*. [En línea] 2012. [Citado el: 3 de

diciembre de 2012.] <http://www.alfresco.com/es/la-plataforma-alfresco>.

Software, Alfresco. 2009. Alfresco.org. [En línea] Alfresco Software, 5 de 3 de 2009. [Citado el: 20 de 2 de 2013.] wiki.alfresco.com/wiki/Alfresco_explore.

Unidad Docente de Ingeniería de Software. *Patrones del “Gang of Four”*. Madrid : s.n.

Valderrama, Luis David Fernández. 2001. *Gestión documental*. Caracas-Venezuela : s.n., 2001. 1578-326x.

Valdés, Ana Rosa Brito, Fernández, Lianet Bárbara Zaldívar. 2010. *Gestión Documental para los Proyectos de Investigación del MIC*. La Habana : s.n., 2010.

Venzia. 2013. Extensión DoCASU y Alfresco. [En línea] Venzia, 2013. [Citado el: 19 de 2 de 2013.]

Anexos

Anexo1 Diagrama de clases del diseño

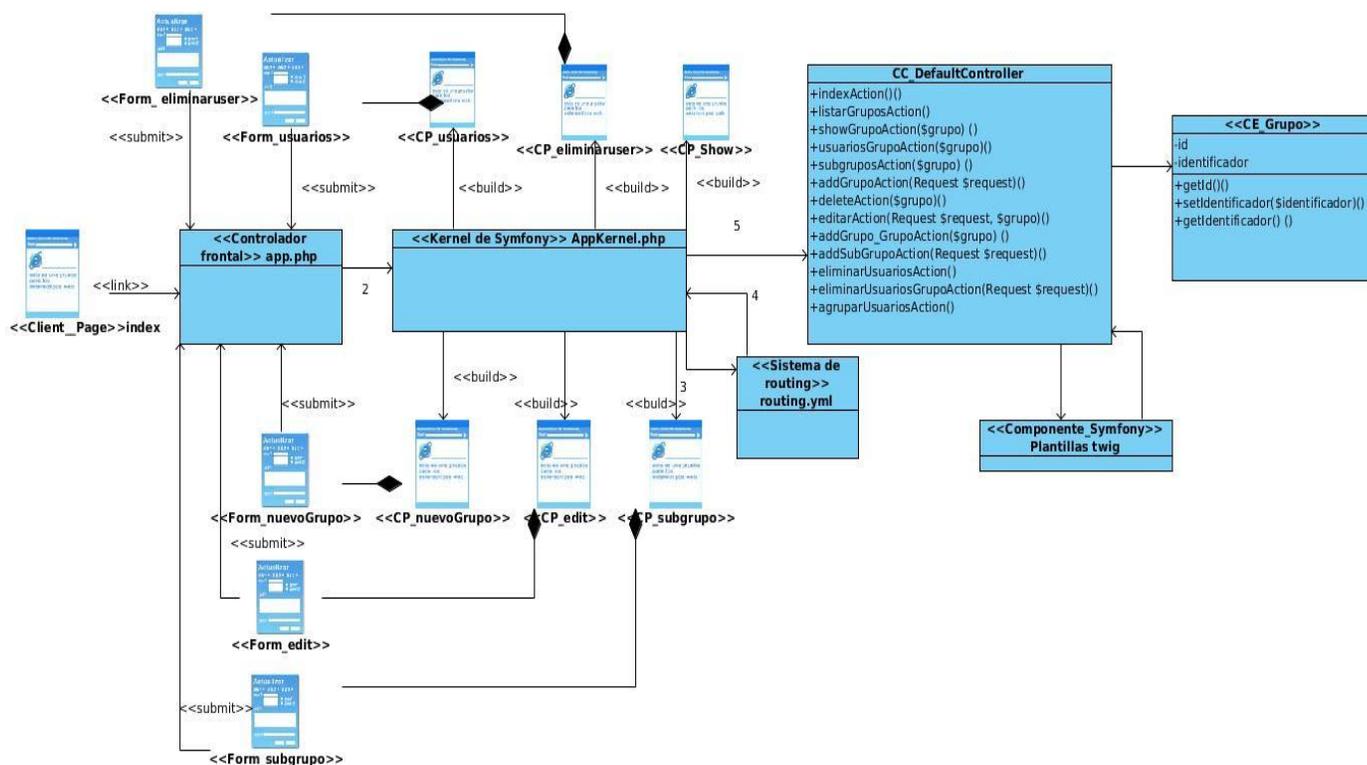


Figura 13 Diagrama de clases del diseño gestionar grupo

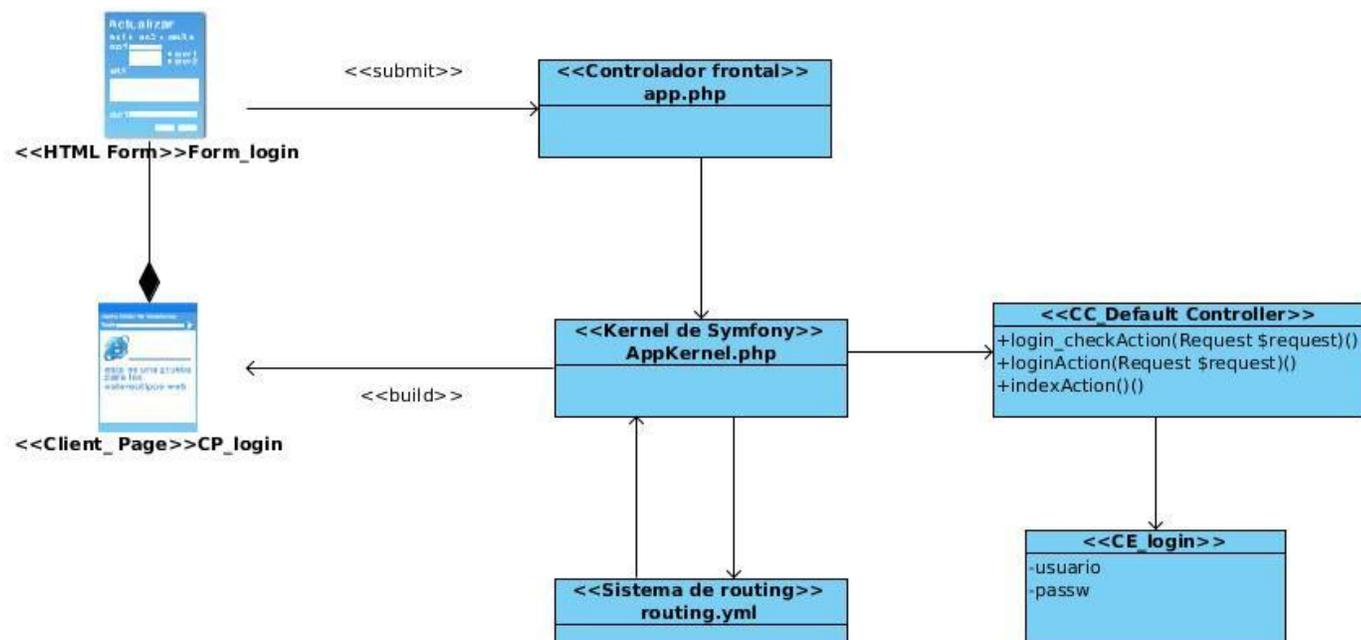


Figura 14 Diagrama de clases del diseño Autenticar usuario

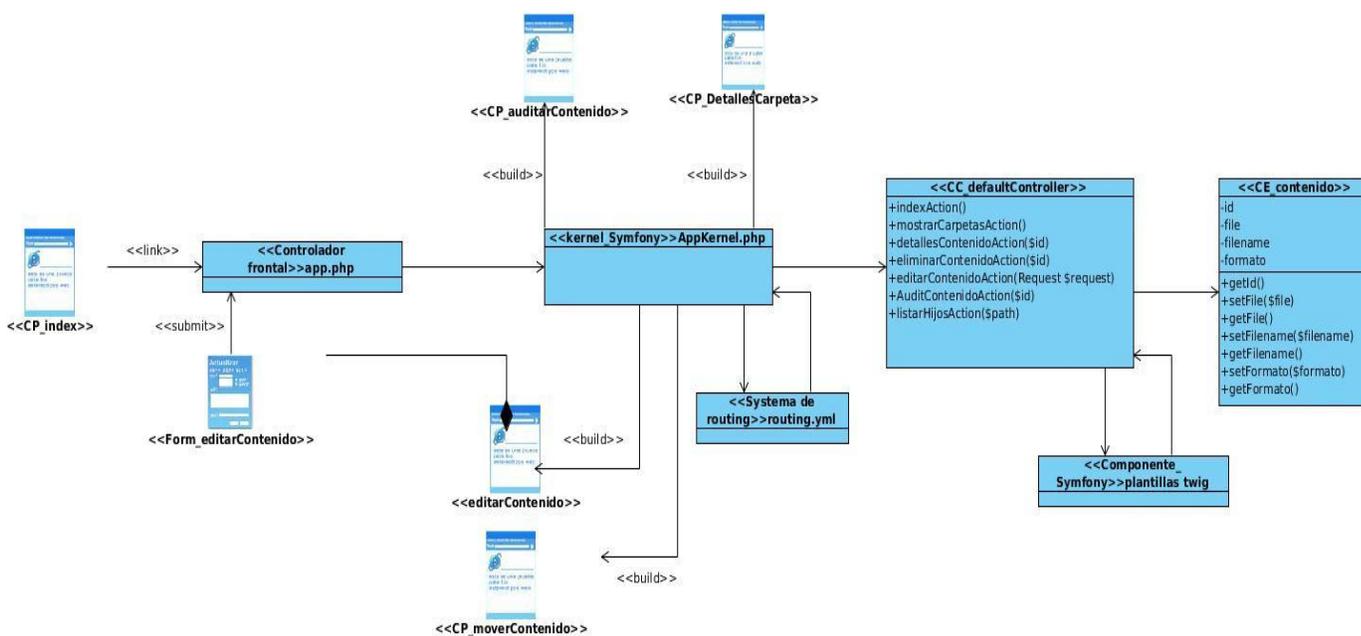


Figura 15 Diagrama de clase del diseño Gestionar Contenido

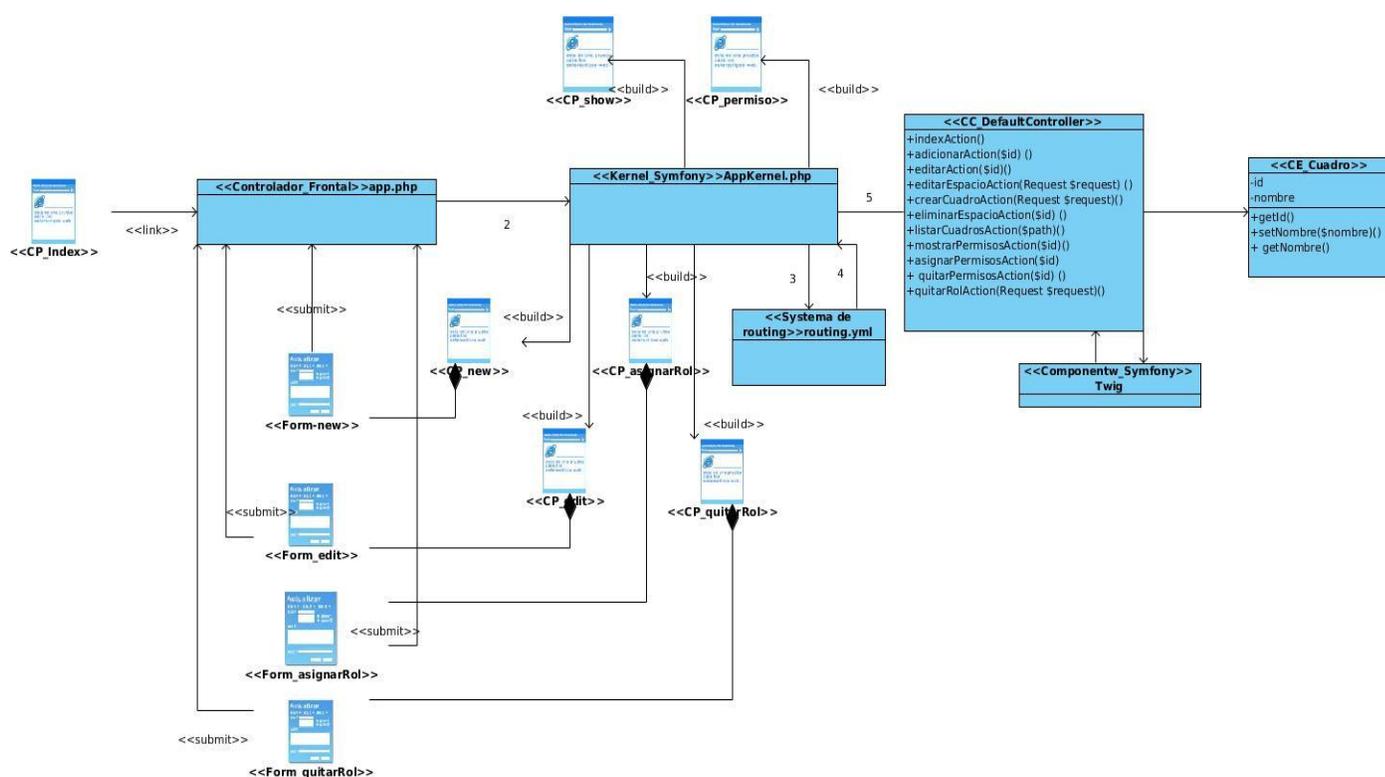


Figura 16 Diagrama de clase del diseño gestionar Cuadro de clasificación

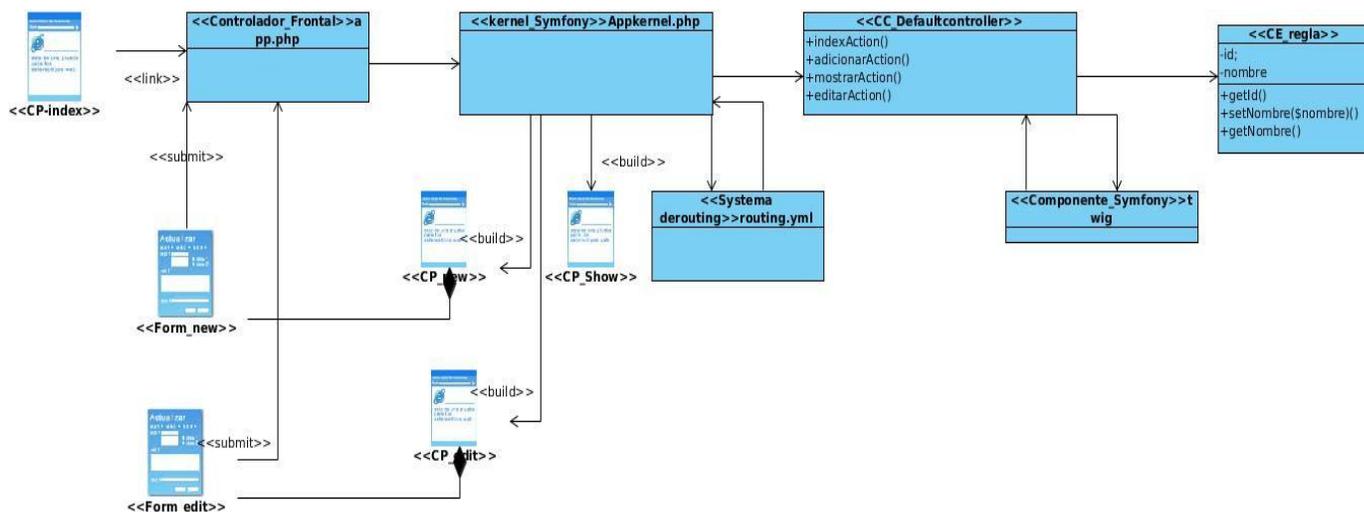


Figura 17 Diagrama de clase del diseño Gestionar Reglas

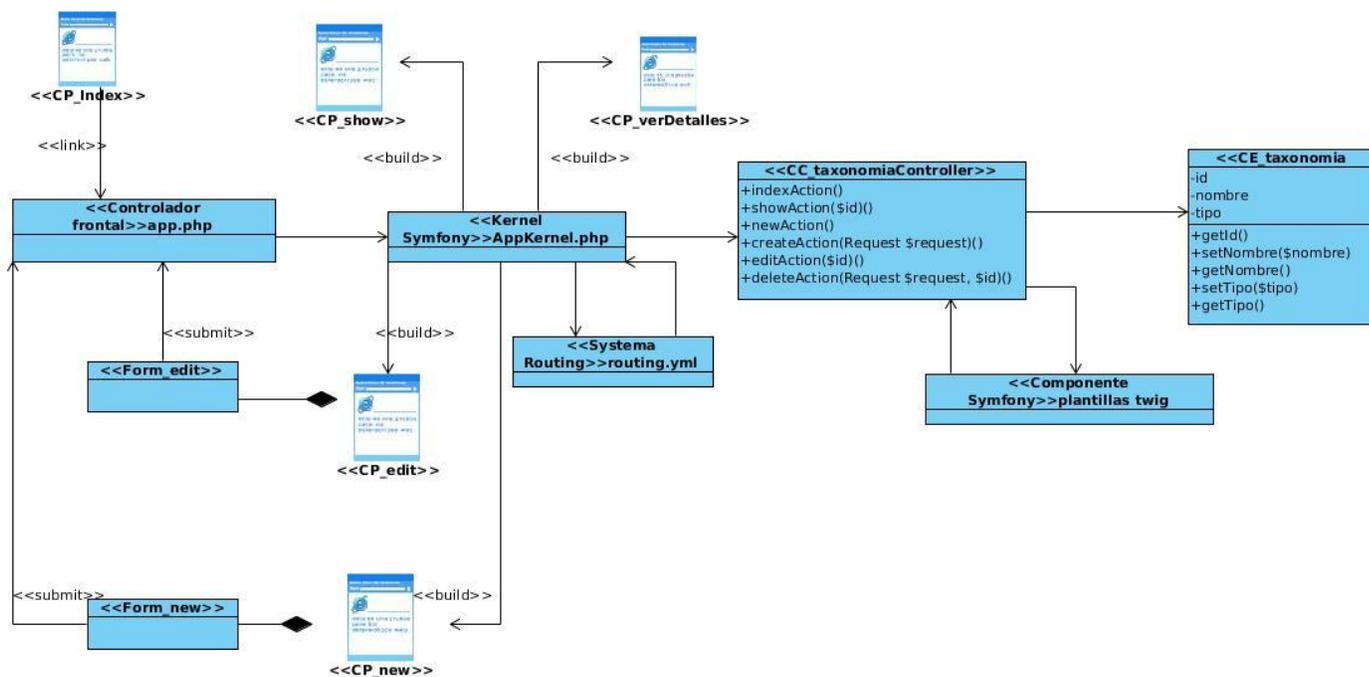


Figura 18 Diagrama de clase del diseño Gestionar taxonomía

Anexo2 Diagrama de secuencia

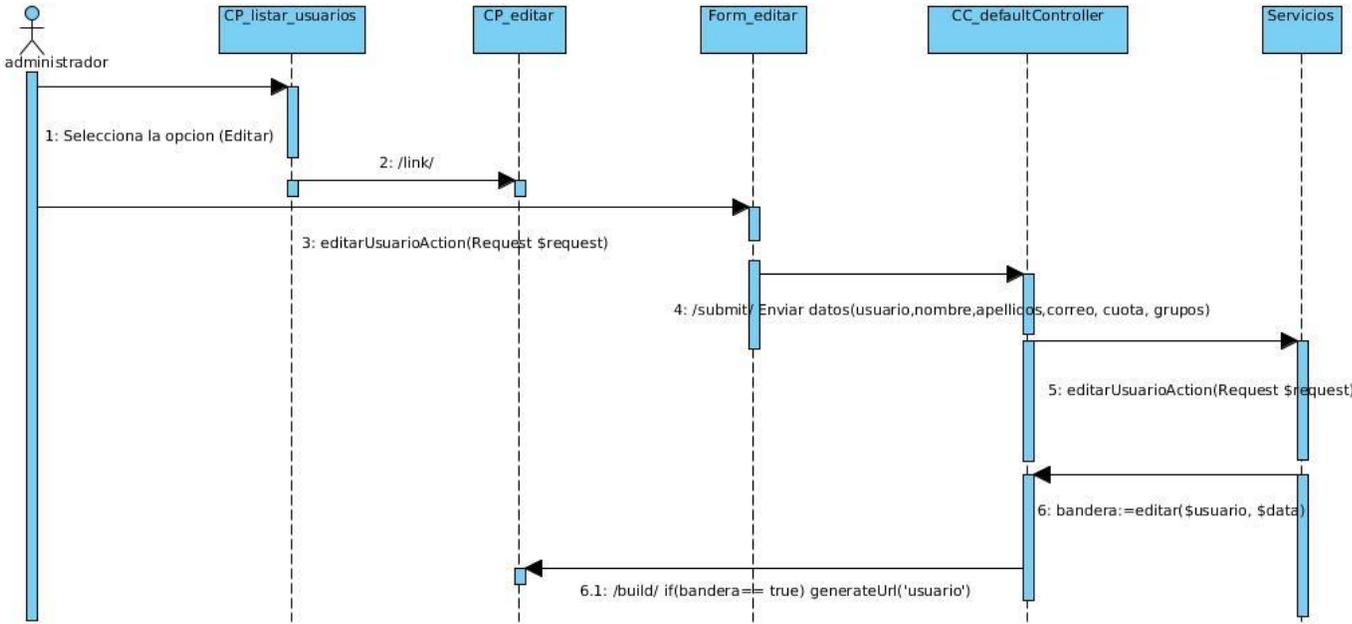


Figura 19 Diagrama de secuencia Editar Usuario

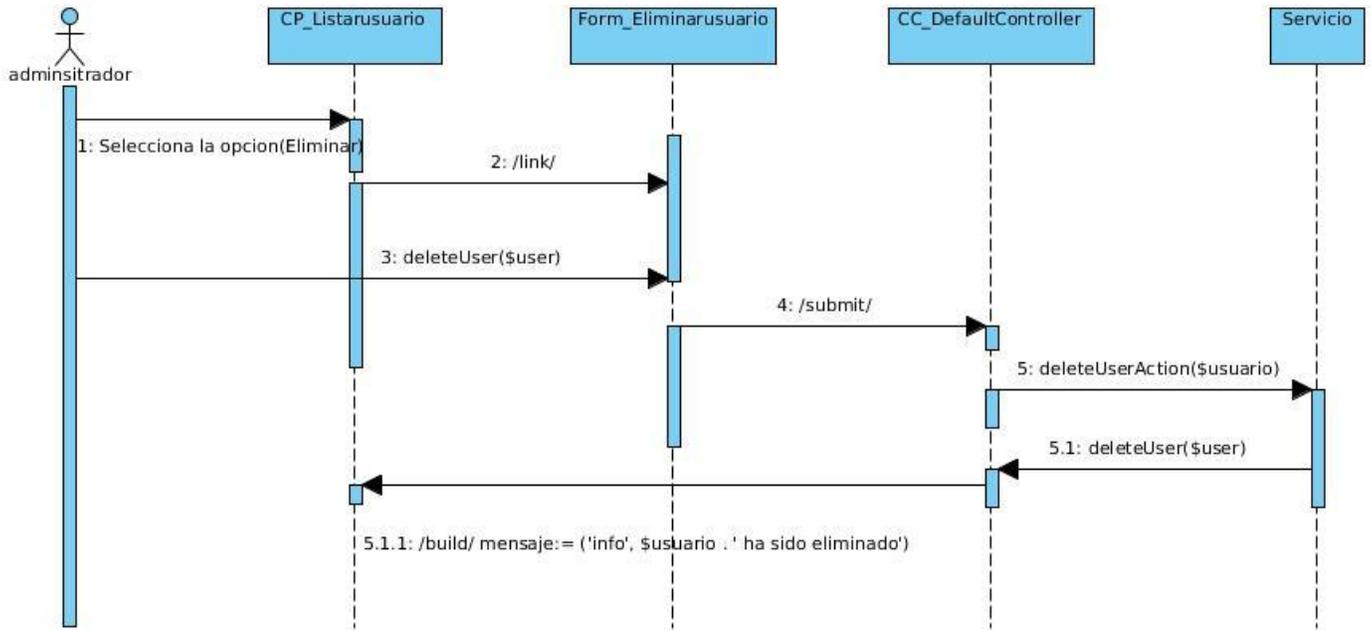


Figura 21 Diagrama de secuencia Eliminar Usuario

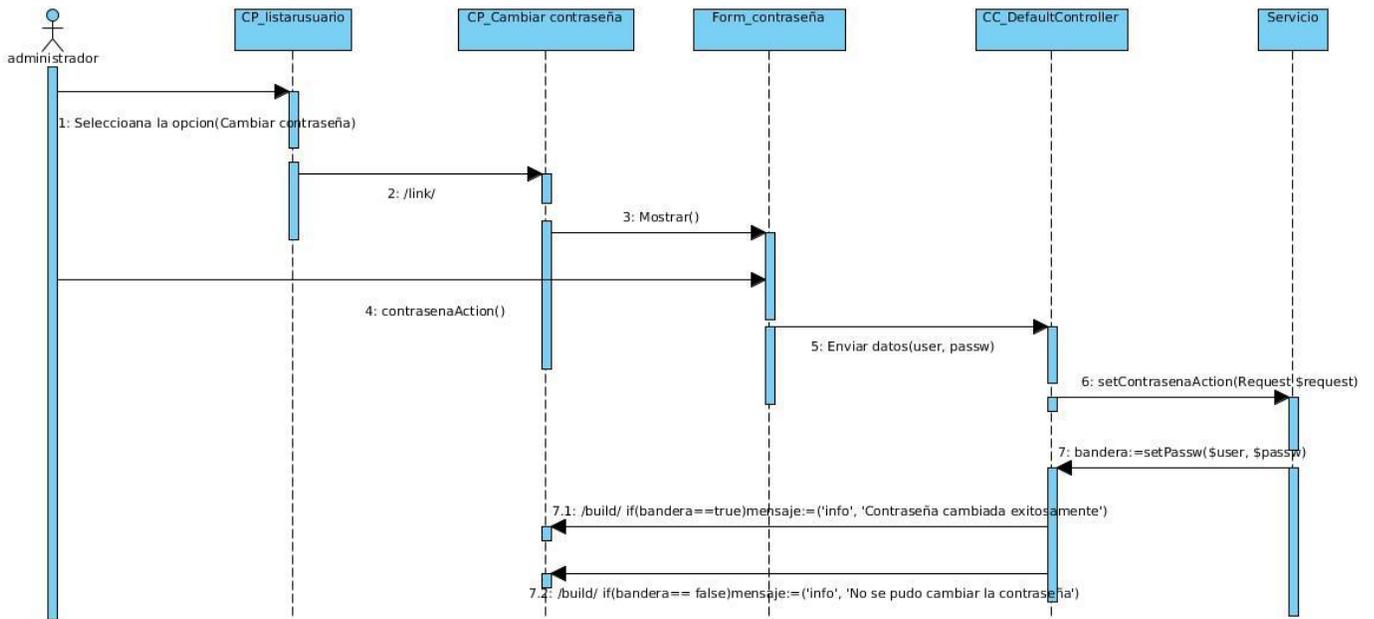


Figura 20 Diagrama de secuencia Cambiar contraseña

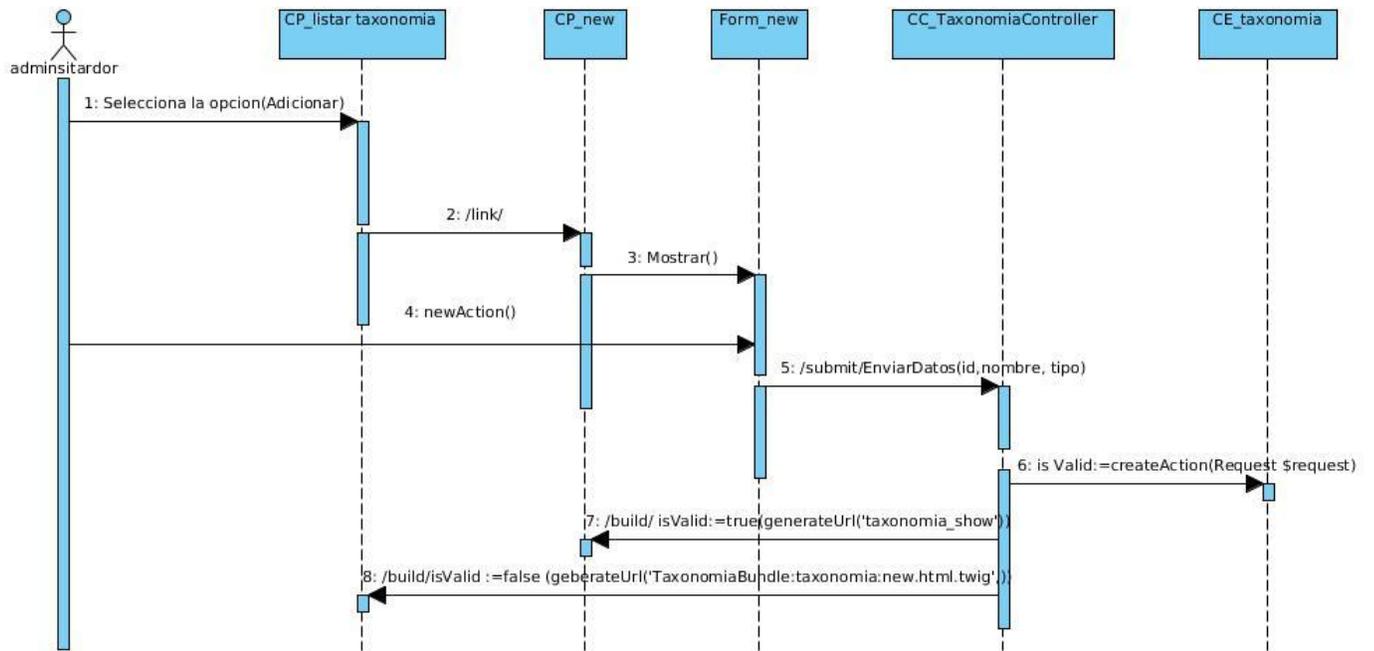


Figura 21 Diagrama de secuencia Adicionar taxonomía

Anexo 3 Diagrama de paquetes

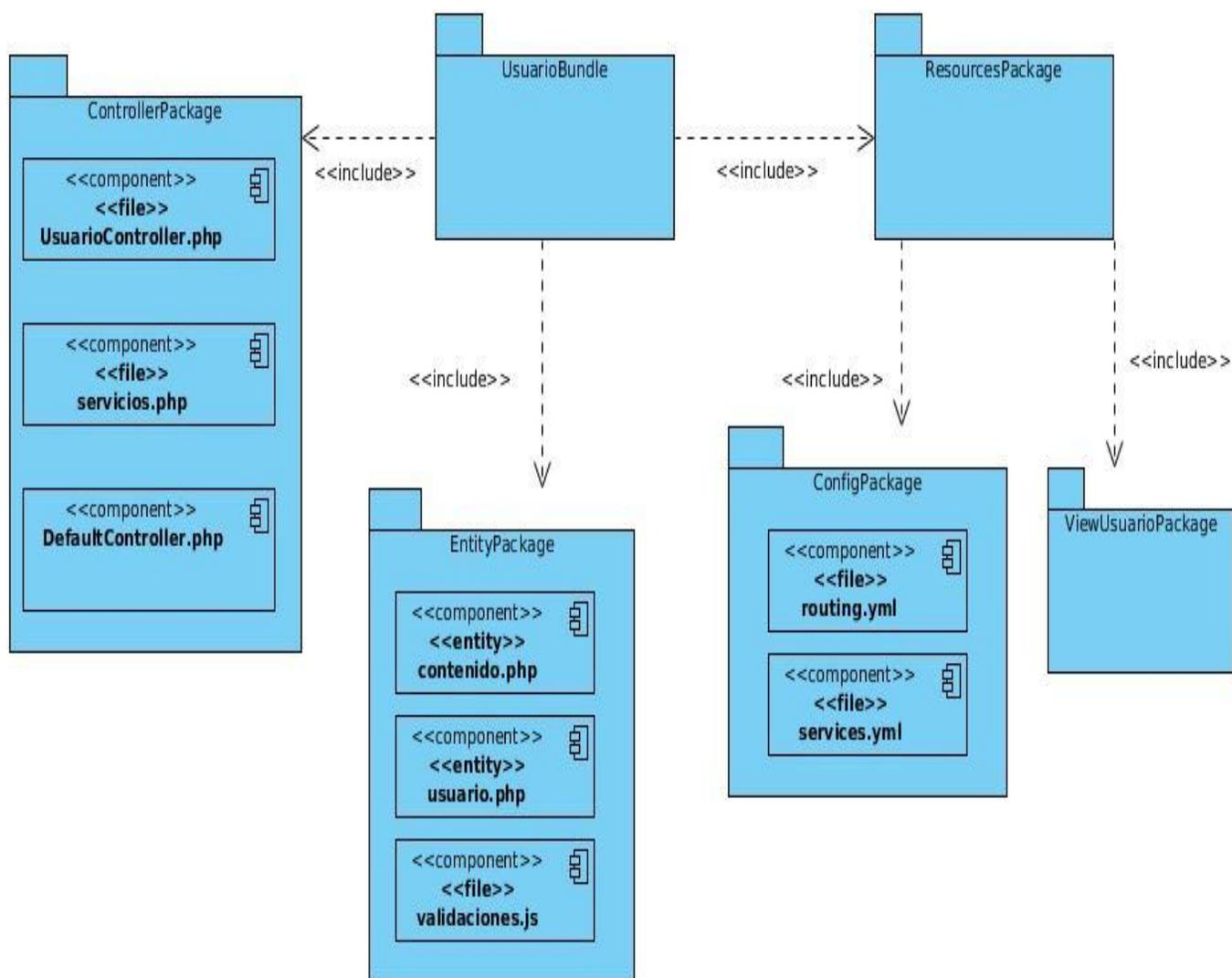


Figura 22 Diagrama de paquete del bundle usuario

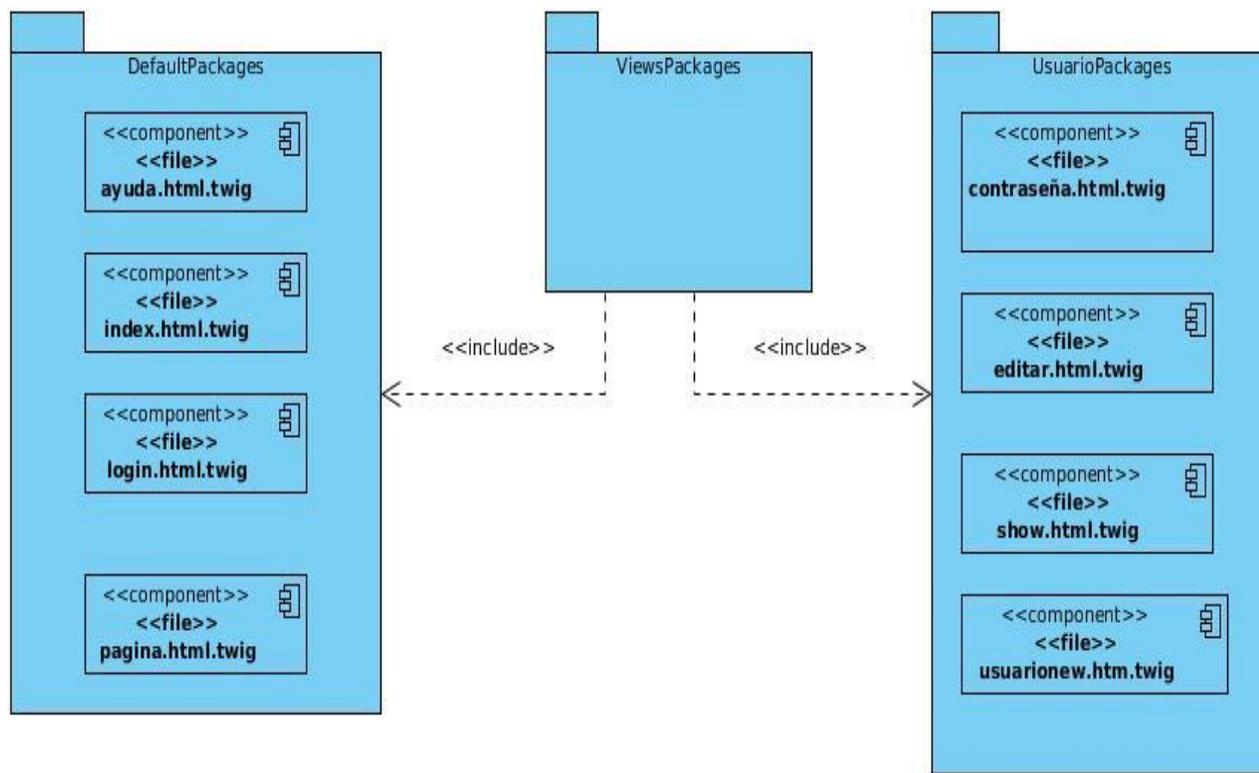


Figura 23 Diagrama que representa el Paquete de vistas del usuario

Anexo4 Prototipos no funcionales

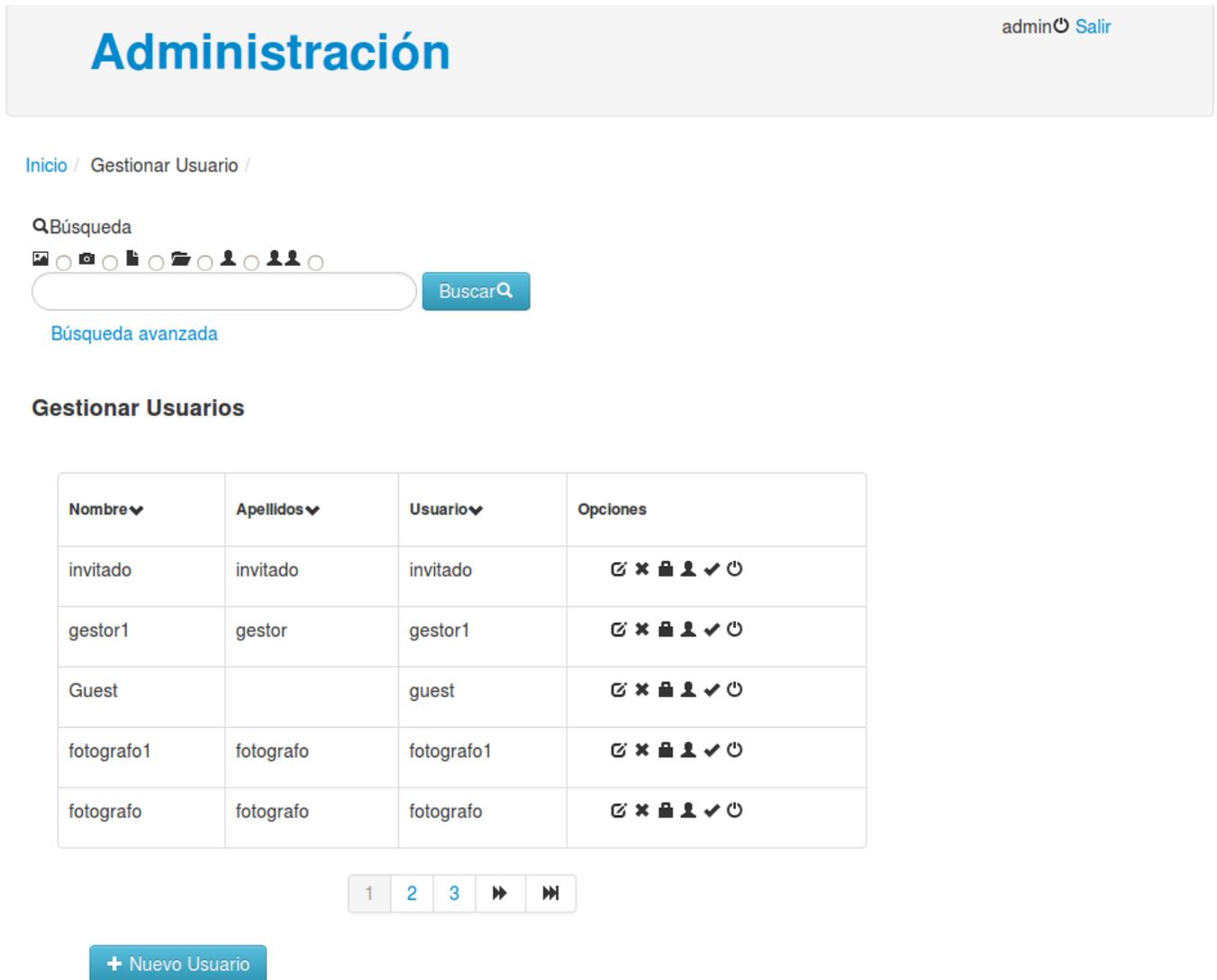


Figura 24 Prototipo de interfaz gestionar usuario

Administración

admin [Salir](#)

[Inicio](#) / [Gestionar Cuadro de Clasificación](#) /

Búsqueda

[Buscar](#)

[Búsqueda avanzada](#)

Gestionar Cuadro de Clasificaciones

Nombre	Opciones
Proceso Documental	📄 ✍️ 🗑️ + 🔒 ✓ - 🗑️ +
Proceso Editorial	📄 ✍️ 🗑️ + 🔒 ✓ - 🗑️ +
Proceso Identificación	📄 ✍️ 🗑️ + 🔒 ✓ - 🗑️ +
Tesis_GDPrensa	📄 ✍️ 🗑️ + 🔒 ✓ - 🗑️ +

[+ Nuevo](#)

UCI Copyleft © 2013 - Ecumene Pyxel | Todos los derechos reservados.

Figura 25 Prototipo de interfaz gestionar de cuadro de clasificación

Administración

Sair

Inicio / Gestionar Usuario / Editar Usuario /

Búsqueda

[Búsqueda avanzada](#)

Editar usuario

Nombre	Usuario
<input type="text" value="invitado"/>	<input type="text" value="invitado"/>
Apellidos	Cuota
<input type="text" value="invitado"/>	<input type="text" value="-1"/>
Correo	Grupo
<input type="text" value="invitado"/>	<input type="text" value="-selecione un grupo-"/>

Figura 26 Prototipo de interfaz editar usuario

Inicio /

Búsqueda



Buscar

Búsqueda avanzada

Autor

Fecha de inicio

Fecha Fin

Usuarios

Espacios

Colección

Subsistema de Administración para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel

Accesos Directos	
3d-67.jpg	
docente 2.jpg	
doc 4.JPG	
IMG_1229.JPG	
3d-69.jpg	
Tesis_GDPrensa	
Seleccion	
Proceso Documental	

Gestionar Usuarios

Gestionar Grupo

Gestionar Cuadro de Clasificación

Gestionar Clasificaciones

Gestionar Contenido

Figura 28 Prototipo de interfaz de usuario realizar búsqueda

Anexo5 Tablas de casos de prueba

Descripción General: Permite eliminar un usuario seleccionado		
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. El usuario debe estar autenticado con el rol de "administrador"		
SC3 Eliminar Usuario		
Escenario	Descripción	Respuesta del sistema
EC 3.1 Se selecciona la opción eliminar, de un usuario determinado	Se procede a eliminar el usuario	Mostrar un mensaje informando "el usuario X ha sido eliminado".

Tabla 15 Descripción del Caso de prueba Eliminar Usuario

Descripción General: Permite cambiar la contraseña de un usuario determinado				
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. El usuario debe estar autenticado con el rol de "administrador"				
SC 4 Cambiar contraseña del usuario				
Escenario	Descripción	Usuario	Contraseña nueva	Respuesta del sistema
EC 2.1 Datos correctos	Se muestra una interfaz con todos los datos usuario y contraseña nueva, el usuario ingresa los datos correctamente.	V	V	Si el usuario Acepta el sistema muestra un mensaje confirmado el cambio, si presiona Volver el sistema regresa a la página inicial del usuario.
		mary	ejemplo12	
EC 2.2 Campos vacíos	El usuario deja alguno o todos los campos sin llenar	I	I	Muestra un mensaje de error que especifica que "Debe introducir el usuario"
		vacío	vacío	
EC 2.3 Datos incorrectos	El usuario introduce algún valor incorrecto en el sistema	V	V	Muestra un mensaje de error que especifica que "El usuario no existe"
		xxx	ejemplo1	

Tabla 16 Descripción del Caso de prueba Cambiar contraseña

Descripción General: Permite adicionar un nuevo grupo en el sistema.

Condiciones de Ejecución: El usuario debe estar autenticado en el sistema.
El usuario debe estar autenticado con el rol de "administrador"

SC 1 Adicionar Grupo

Escenario	Descripción	Nombre	Identificador	Respuesta del sistema	Flujo central
EC 3.1 Datos correctos	Se muestra una interfaz con todos los datos del grupo y se llenan correctamente	V	V	Si el usuario Acepta el sistema muestra un mensaje El grupo FOTOTECA_EDITOR ES ha sido creado	El usuario selecciona la opción Nuevo Grupo. Introduce los datos del grupo y presiona la opción Adicionar o Grupos para regresar
		FOTOTECA_EDITORES	FOTOTECA_EDITORES		
EC 2.2 Campos vacíos	El usuario deja alguno o todos los campos sin llenar	I	I	Muestra un mensaje de error que especifica que "Debe introducir el nombre del grupo"	El usuario selecciona la opción Nuevo Grupo. Introduce los datos del grupo y presiona la opción Adicionar o Grupos para regresar

Tabla 17 Caso de prueba Adicionar grupo

Descripción General: Permite eliminar un grupo seleccionado			
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. El usuario debe estar autenticado con el rol de “ administrador”			
SC2 Eliminar Grupo			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 3.2 Se selecciona la opción eliminar, de un grupo determinado	Se procede a eliminar el grupo en específico	Mostrar un mensaje informando "el grupo X ha sido eliminado".	Se selecciona la Opción eliminar grupo, que se encuentra a la derecha de cada grupo.

Tabla 18 Caso de prueba Eliminar grupo

Descripción General: Permite crear un subgrupo dentro de un grupo determinado.					
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. El usuario debe estar autenticado con el rol de “ administrador”					
SC 6 Adicionar Subgrupo					
Escenario	Descripción	Grupo	Subgrupo	Respuesta del sistema	Flujo central
EC 3.1 Datos correctos	Se muestra una interfaz con todos los datos del grupo y se llenan correctamente	V	V	Si el usuario Acepta el sistema crea el nuevo grupo perteneciente al grupo seleccionado.	El usuario selecciona la opción Adicionar Subgrupo. Introduce los datos del grupo y presiona la opción Adicionar o Grupos para regresar
		FOTOTECA_E DITORES	GROUP_nuevo		
EC 2.2 Campos vacíos	El usuario deja alguno o todos	I	I	Muestra un mensaje de error que especifica que	El usuario selecciona la
		vacío	vacío		

	los campos sin llenar			"Debe introducir el nombre del grupo"	opción Nuevo Subgrupo. Introduce los datos del grupo y presiona la opción Adicionar o Grupos para regresar
EC 2.3 Datos incorrectos	El usuario introduce algún valor incorrecto en el sistema	V	V	Muestra un mensaje de error que especifica: El nombre del grupo debe comenzar con GROUP_	El usuario selecciona la opción Nuevo Subgrupo. Introduce los datos del grupo y presiona la opción Adicionar o Grupos para regresar
		FOTOTECA_E DITORES	incorrecto		

Tabla 19 Caso de prueba Crear un Subgrupo

Glosario de Términos

Término	Definición
API	(Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.
CSS	Hojas de Estilos en Cascadas por sus siglas en Inglés <i>Cascading Style Sheets</i> Tecnología empleada en la creación de páginas Web, que permite un mayor control sobre el lenguaje HTML. Permite hojas de estilo que definen como cada elemento, se tiene que mostrar. CSS ha sido desarrollado por W3C.
CUS	Casos de uso del Sistema.
DMS	Sistemas de Gestión Documental por sus siglas en Inglés <i>Document Management System</i> , son programas de gestión de bases de datos que disponen de una tecnología idónea para el tratamiento de documentos.
ECM	Gestor de Contenido Empresarial, el cual identifica a los sistemas informáticos que manejan la captura, almacenamiento, seguridad, control de versiones, recuperación, distribución, conservación y destrucción de documentos y contenido a nivel empresarial.
ECM Alfresco	Es la alternativa de Código Abierto para la gestión de contenido empresarial (ECM), este proporciona gestión documental, colaboración, gestión de registros, gestión de registros, gestión de información, gestión de contenidos web e imágenes.
Framework	En el desarrollo de software es una estructura conceptual y tecnológica de soporte definida, normalmente, con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.
FTP	Por sus siglas en Inglés <i>File Transfer Protocol</i> , este protocolo de red llamado Protocolo de Transferencia de Archivos es como su nombre lo indica una de las formas en la cual se puede enviar archivos hacia una Red TCP.

GD	Gestión Documental.
GPL	Licencia pública general.
GDA	Gestión de los Documentos Administrativos.
GOF	Banda de los cuatro por sus siglas en inglés Gang of Four. Los patrones GOF constituyen un catálogo de 23 patrones de diseño básicos publicados por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides.
HTML	Lenguaje de marcado de hipertexto.
HTTP	Por sus siglas en Inglés <i>HyperText Transfer Protocol</i> . Es el protocolo que emplea la WWW. Define como se tienen que crear y enviar los mensajes y que opciones debe tener el servidor y el navegador en respuesta a un comando. Es un protocolo de red, funciona a través de solicitudes y respuestas entre un cliente.
HTTPS	Por sus siglas en Inglés <i>HyperText Transfer Protocol Security</i> . Protocolo seguro para la transferencia de hipertexto"), un protocolo seguro que garantiza una comunicación autenticada y cifrada.
IDE	Entorno de Desarrollo Integrado.
JSON	Formato ligero para el intercambio de datos.
MIC	Metodología de la Investigación y las Comunicaciones.
ORM	Mapeo Objeto Relacional.
PHP	Lenguaje de código abierto muy popular especialmente adecuado para el desarrollo Web y que puede ser incrustado en HTML. Lenguaje de programación interpretado.
Symfony	Es un Framework para programar aplicaciones Web en PHP.
SQL	<i>Structured Query Language</i> .
SGDID	Sistema de Gestión Documental de Imágenes Digitales.
SGBD	Sistema Gestor de Base de Datos.
Servicio WEB	Un servicio web por sus siglas en Inglés <i>Web service</i> , es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
UML	Lenguaje Unificado de Modelado por sus siglas en inglés <i>Unified Modeling Language</i> es un conjunto de herramientas, que permite modelar (analizar y diseñar) sistemas orientados a

	objetos.
UCI	Son las siglas de Universidad de las Ciencias Informáticas la cual fue creada, por el comandante en jefe de la Revolución cubana, al calor de la batalla de ideas; en el año 2002.
URL	Localizador de Recurso Uniforme por sus siglas en inglés <i>Uniform Resource Locator</i> , la dirección global de documentos y de otros recursos en la Web.
W3E	<i>World Wide Web Consortium</i> . Consorcio internacional de compañías y organizaciones involucradas en el desarrollo de Internet y en especial de la World Wide Web. Su propósito es desarrollar estándares y “poner orden “en Internet.
XML	Lenguaje Extensible de Marcas por sus siglas en inglés <i>eXtensible Markup Language</i> . Metalenguaje de etiquetado en SGML. Diseñado específicamente para la WWW por W3C. Permite que un usuario diseñe sus propias etiquetas, con sus atributos y reglas de construcción de documentos.