

Universidad de las Ciencias Informáticas

Facultad 1



**Solución para la gestión de flujos de trabajo avanzados en el Gestor de Documentos  
Administrativos eXcriba.**

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas

**Autor:** Roberto Allen Polanco

**Tutores:** Ing. Misael Fonseca Mata

Ing. Lizandra Candelario Rodríguez

**La Habana, 20 de Junio del 2012**

## **Declaración de autoría**

Declaro que soy el único autor de este trabajo y autorizo al Centro de Informatización Universitaria de la Universidad de las Ciencias Informáticas, para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 20 días del mes de junio del año 2012.

---

Firma del autor

Roberto Allen Polanco

---

Firma del tutor

Ing. Misael Fonseca Mata

---

Firma del tutor

Ing. Lizandra Candelario Rodríguez

## Resumen

Actualmente en la Universidad de las Ciencias Informáticas (UCI) se desarrolla el Gestor de Documentos Administrativos eXcriba. Esta aplicación permite gestionar los documentos electrónicos a lo largo de su ciclo de vida, desde su creación hasta la disposición o almacenamiento permanente de los mismos. Sin embargo, carece de un medio que permita agilizar el conjunto de tareas y actividades coordinadas por las empresas. Con el objetivo de dar solución a este problema la presente investigación propone el desarrollo de un módulo para la gestión de flujos de trabajo avanzados en el *software* eXcriba. Para ello se describen los antecedentes, aspectos teóricos y conceptuales relacionados con los procesos de negocio y los flujos de trabajo. Se realiza un análisis de las soluciones existentes en la actualidad. Se seleccionan las herramientas, metodología y tecnologías necesarias para el desarrollo del módulo. Se define el diseño de la solución y se describen los artefactos generados en la fase de implementación. Además, se evalúa su correcto funcionamiento mediante pruebas de caja blanca y caja negra.

**Palabras clave:** gestión documental, gestión de flujos de trabajo, procesos de negocio.

# Índice General

<b>Introducción</b>	<b>1</b>
<b>1. Fundamentación teórica del módulo</b>	<b>6</b>
1.1. Gestión Documental . . . . .	6
1.2. Sistema de Gestión Documental . . . . .	7
1.3. Gestión de procesos de negocio . . . . .	8
1.4. Flujo de trabajo . . . . .	8
1.5. Flujo de trabajo en los sistemas de gestión documental . . . . .	10
1.6. Metodología de desarrollo de <i>software</i> . . . . .	14
1.7. Lenguajes . . . . .	15
1.8. Herramienta CASE . . . . .	16
1.9. Tecnologías . . . . .	17
1.9.1. Marco de trabajo . . . . .	18
<b>2. Características del módulo</b>	<b>20</b>
2.1. Problema y situación problemática . . . . .	20
2.2. Propuesta de solución . . . . .	20
2.3. Modelo de dominio . . . . .	21
2.4. Especificación de los requisitos de <i>software</i> . . . . .	23
2.4.1. Técnicas para la captura de requisitos . . . . .	23

2.4.2.	Requerimientos funcionales . . . . .	24
2.4.3.	Requerimientos no funcionales . . . . .	24
2.5.	Definición del actor y los casos de uso del sistema . . . . .	26
2.5.1.	Definición de los actores . . . . .	26
2.5.2.	Definición de los casos de uso del módulo . . . . .	26
2.5.3.	Diagrama de caso de uso del módulo . . . . .	29
2.5.4.	Descripción de los casos de uso del módulo . . . . .	29
<b>3.</b>	<b>Diseño del módulo</b>	<b>33</b>
3.1.	Modelo de diseño . . . . .	33
3.1.1.	Diagrama de clases de diseño . . . . .	33
3.1.2.	Descripción de las clases . . . . .	35
3.1.2.1.	Descripción de los servicios . . . . .	37
3.2.	Diagrama de secuencia del diseño . . . . .	42
3.3.	Descripción de la arquitectura . . . . .	43
3.3.1.	Arquitectura en capas . . . . .	43
3.4.	Patrones de diseño . . . . .	46
<b>4.</b>	<b>Implementación y Prueba del módulo</b>	<b>48</b>
4.1.	Diagrama de despliegue . . . . .	48
4.2.	Diagrama de componentes . . . . .	50
4.3.	Pruebas de <i>software</i> . . . . .	51
4.3.1.	Prueba de caja blanca . . . . .	51
4.3.2.	Pruebas de caja negra . . . . .	54
4.3.2.1.	Casos de prueba de caja negra . . . . .	55
	<b>Conclusiones</b>	<b>61</b>

<b>Recomendaciones</b>	<b>62</b>
<b>Referencias bibliográficas</b>	<b>63</b>
<b>Bibliografía</b>	<b>67</b>
<b>A. Primer apéndice</b>	<b>72</b>
A.1. Descripción de los casos de uso del sistema . . . . .	72
<b>B. Segundo apéndice</b>	<b>76</b>
B.1. Diagrama de clases de diseño . . . . .	76
<b>C. Tercer apéndice</b>	<b>78</b>
C.1. Diagrama de secuencia del diseño . . . . .	78

# Introducción

**E**n el año 1934, uno de los padres de la documentación y responsable de la creación del Instituto Internacional de Bibliografía Paul Otlet<sup>1</sup> consideraba que: *“Para volver accesible la cantidad de informes y artículos dados cada día en la prensa cotidiana, en las revistas, para conservar folletos, informes, prospectos, documentos oficiales; para encontrar materias dispersas en los libros, para hacer un todo homogéneo de esas masas incoherentes, son precisos procedimientos nuevos”*[1].

Este no estuvo tan equivocado, cuando en el año 2010 la International Data Corporation<sup>2</sup> en su informe “El Universo Digital en expansión: una previsión del crecimiento mundial de información hasta 2010”, planteaba que el volumen de información crecería a un ritmo acelerado entre los años 2006 y el 2010. Además, el documento enuncia que la cantidad de información creada, capturada y replicada en el 2010 llegaría a ser semejante a 72 columnas de libros desde la Tierra hasta el Sol. Sin embargo, una actualización del estudio en el mes de junio del 2011 demostró que esta cifra superó todos los cálculos realizados llegando a alcanzar la barrera de los exabytes<sup>3</sup> [2].

Ante este aumento de la información y con el objetivo de ocupar menos espacio se comienza a utilizar el formato electrónico generado por programas informáticos mediante las computadoras para el almacenamiento de la documentación, lo cual ocasionó un cambio de enfoque de la importancia de la información para la sociedad. Sin duda, la aplicación de las tecnologías a la gestión de la información daría lugar al surgimiento,

---

<sup>1</sup>OTLET, Paul. El tratado de documentación... [ed. facs. de la ed. de París: Palais Mundaem, 1934]. Murcia: Universidad, 1996.

<sup>2</sup>International Data Corporation IDC es proveedor líder en el mundo en análisis, inteligencia de mercados y en soporte técnico a fabricantes y usuarios, en el sector de Informática y Telecomunicaciones.

<sup>3</sup>Exabyte (Ex) es el equivalente a 1.000 millones de GB.

redefinición y fusión de numerosas disciplinas en función de las características, condiciones y necesidades actuales de los nuevos usuarios de la información; tal es el caso de la gestión documental de hoy: la gestión electrónica de documentos.

Sobre la base de esta compleja situación en la que se encontraban expuestas las empresas y de la redefinición de la gestión documental, los tecnólogos de la información en conjunto con los especialistas de ciencias de la información han asumido el reto de poner en manos de sus organizaciones las tecnologías, sistemas e infraestructuras adecuadas para almacenar, proteger y administrar la información de manera eficaz y eficiente, así como ejecutar y automatizar los procesos de negocio que utilizan dicha información para crear nuevas fuentes de valor para las organizaciones. Así, los sistemas de gestión documental han dejado de ser una herramienta de uso casual en el apoyo a la actividad administrativa para convertirse en la principal fuente de valor sobre la información.

Cuba no está exenta de este cambio y como parte de la estrategia de la informatización de la sociedad y para integrarse al mercado internacional se han creado soluciones informáticas que aportan una infraestructura sólida para la gestión de documentos, como es el caso del Gestor de Documentos Administrativos (GDA) eXcriba<sup>4</sup> desarrollado en la Universidad de las Ciencias Informáticas (UCI).

Desde sus inicios eXcriba se concibió con el objetivo de cubrir el ciclo de vida de los documentos desde dos componentes esenciales, una estática y otra dinámica. La componente vertical o estática define que todo documento en su ciclo de vida transita por cuatro fases fundamentales: creación o captura, gestión, distribución y almacenamiento o eliminación. Por otra parte, la componente horizontal o dinámica responde a la ejecución de como mínimo un proceso de negocio ya sea operacional o administrativo en el cual colaboran diferentes equipos de trabajo alineados al cumplimiento de las regulaciones o reglamentaciones establecidas [3].

Actualmente la componente dinámica en el Gestor de Documentos Administrativos eXcriba abarca la automatización de reglas de negocio, logrando con ello la organización automática de los documentos y la ejecución de flujos de trabajo simple<sup>5</sup> permitiendo automatizar la aprobación o rechazo de documentos durante

---

<sup>4</sup>eXcriba: *software* para la gestión documental.

<sup>5</sup>Según Munwar Shariff en el libro *Alfresco Enterprise Content Management Implementation* define los flujos de trabajo simple como: el movimiento de documentos a través de diversos espacios.

el proceso de revisión. No obstante, el cliente *web* del eXcriba carece de un medio que permita a los usuarios automatizar un proceso de negocio y asignarlo a un evaluador. Además, no presenta una funcionalidad para realizar un seguimiento de los flujos de trabajo avanzados<sup>6</sup> en el que se encuentra involucrado el usuario, así como no existe una vía para cambiar el estado de los mismos. El *software* no presenta un medio por el cual se pueda enviar notificaciones a las partes involucradas.

Por lo anteriormente expuesto, la presente investigación contribuye a solucionar el siguiente **problema científico**: ¿Cómo gestionar los flujos de trabajo avanzados en el cliente *web* del eXcriba?

Para enmarcar los límites de esta investigación se define como **objeto de estudio**: la gestión de procesos de negocio en aplicaciones informáticas, delimitando el **campo de acción** en la gestión de flujos de trabajo en el Gestor de Contenido Empresarial Alfresco.

La presente investigación tiene como **objetivo general**: Desarrollar un módulo para la gestión de flujos de trabajo avanzados en el Gestor de Documentos Administrativos eXcriba, utilizando el motor de proceso de negocio del Gestor de Contenido Empresarial Alfresco.

Con el propósito de dar cumplimiento a lo anteriormente expuesto se trazan los siguientes **objetivos específicos**:

- Fundamentar los antecedentes y aspectos teóricos de los procesos de negocio en la gestión documental y los flujos de trabajo.
- Diseñar el módulo para gestionar los flujos de trabajo avanzados en el Gestor de Documentos Administrativos eXcriba.
- Implementar el módulo para gestionar los flujos de trabajo avanzados en el Gestor de Documentos Administrativos eXcriba.
- Validar mediante pruebas de caja blanca y caja negra el correcto funcionamiento del módulo desarrollado.

---

<sup>6</sup>La automatización de los procesos de negocio durante el cual “documentos”, “información” y “tareas” son pasados de un participante a otro.

El trabajo de diploma queda sustentado en la siguiente **idea a defender**: El desarrollo de un módulo para la gestión de flujos de trabajo avanzados en el Gestor de Documentos Administrativos eXcriba, permitiría la asignación y la creación de los flujos de trabajo, contribuyendo a que los procesos de negocio se ejecuten de forma rápida y ágil.

Para el desarrollo del siguiente trabajo de diploma se utilizaron los métodos científicos de investigación: Analítico-Sintético, el Histórico-Lógico y el de Modelación como métodos teóricos y como métodos empíricos la Observación.

**Analítico-Sintético:** Se utilizó con el objetivo de dividir el objeto de estudio en conceptos que serán examinados por separados, estudiándose rigurosamente cada uno de ellos de manera independiente y confrontando el criterio de disímiles autores y las correspondencias entre ellos. Además, se realizará una investigación sobre los diferentes sistemas de gestión existentes en el ámbito internacional que utilizan los flujos de trabajo avanzados.

**Histórico-Lógico:** Para la realización de este trabajo investigativo se realizó un estudio con anterioridad sobre los flujos de trabajo en las empresas, así como su automatización en sistema de gestión documental, atendiendo a su progreso y desarrollo del mismo.

**Modelación:** Es empleado para modelar los diagramas que se obtengan durante el diseño del módulo, obteniendo una panorámica del flujo de actividades.

**Observación:** Este método a lo largo del trabajo investigativo es empleado para la exploración del objeto de estudio y poder formular el problema a solucionar, así como también en el diseño de la investigación. Este es utilizado para el desarrollo de las tendencias de los flujos de trabajo.

Para un mejor entendimiento tanto del problema como de la solución que se propone, el trabajo de diploma se ha estructurado de la siguiente manera: introducción, cuatro capítulos que serán descritos a continuación, recomendaciones, bibliografía, referencias bibliográficas y los anexos.

**Capítulo I: Fundamentación teórica del módulo:** El objetivo de este capítulo es abordar conceptos y aspectos generales que permitan comprender los temas relacionados con los flujos de trabajo. Además, se exponen las tecnologías, herramientas, lenguajes y metodología utilizadas en la elaboración del sistema.

**Capítulo II: Características del módulo:** Este capítulo tiene como principal objetivo planificar la solución propuesta haciendo uso de la metodología *RUP (Proceso Rational Unificado)* con nivel 2 de CMMI<sup>7</sup>. Para ellos se especifican los requisitos de *software*, diagramas de modelo de dominio y casos de uso del sistema.

**Capítulo III: Diseño del módulo:** Se exponen a través de un conjunto de artefactos la solución que se le dará al problema en cuestión, dentro de los cuales son fundamentales el diagrama de clase del diseño de cada caso de uso del sistema.

**Capítulo IV: Implementación y Prueba del módulo:** En este capítulo se define la implementación del módulo. La estructura en clases y componentes que garanticen la capacidad, los defectos y pruebas realizadas al mismo a lo largo del ciclo de vida del producto.

---

<sup>7</sup>Modelo de Capacidad y Madurez Integrado. CMMI es un modelo para la mejora de procesos, que proporciona a las organizaciones, los elementos esenciales para procesos eficaces.

# Capítulo 1

## Fundamentación teórica del módulo

### 1.1. Gestión Documental

A mediados del siglo XX fue reconocida de forma oficial la gestión documental. Esta revolucionó toda la práctica archivística que se estaba ejecutando hasta ese momento, debido a que introduce el ciclo vital de los documentos de lo cual no se tenía conocimiento hasta entonces, evidenciando una interconexión entre las diversas etapas o procedimientos que se aplican a los archivos personales o institucionales [4].

La Federal Records Act definió la gestión documental como: *“un área de la gestión administrativa general que se ocupa de garantizar la economía y eficiencia en la creación, mantenimiento, uso y disposición de los documentos archivísticos (records) a lo largo de todo su ciclo de vida”* [5]. Esta tiene como objetivo *“... asegurar una documentación adecuada, evitar lo no esencial, simplificar los sistemas de creación y uso del papeleo, mejorar la forma de cómo se organizan y recuperan los documentos, proporcionar el cuidado adecuado y el almacenamiento a bajo costo de los documentos en los archivos intermedios y asegurar la eliminación adecuada de los documentos que no se necesitan desde hace tiempo en la gestión de los asuntos del momento...”* [5], para garantizar mediante su custodia física, su veracidad.

La gestión de documentos se convierte en un proceso vital en las organizaciones por la magnitud que alcanzan los documentos, provocado además por el amplio alcance y complejidad de las actividades y la transferencia de datos e informaciones [6].

## 1.2. Sistema de Gestión Documental

Debido al enorme volumen de documentación que se maneja en las organizaciones, se ha convertido en todo un reto la administración y control de los documentos, lo cual ha fomentado la adopción de diversas herramientas que favorecen su gestión. Estas aportan mejoras destacadas en el control de los documentos y ofrecen reducciones de tiempos de búsqueda, utilización y ahorro de gastos, muy valioso en los procesos administrativos.

Para Carlota Bustelo, ... *“un SGD<sup>1</sup>, es una parte del sistema de información de la empresa desarrollado con el propósito de almacenar y recuperar documentos, que debe estar diseñado para coordinar y controlar todas aquellas funciones y actividades específicas que afectan a la creación, recepción, almacenamiento, acceso y preservación de los documentos, salvaguardando sus características estructurales, contextuales y garantizando su autenticidad y veracidad”* [7].

Según Mayra Mena [8] los sistemas de gestión documental deben de tener como objetivo:

- Hacer más fácil a las personas trabajar con los documentos. Cada persona sabe qué documentos tiene que guardar, cuándo, cómo y dónde. Además, debe saber realizar búsqueda de información.
- Facilitar que la información se comparta y se aproveche como un recurso colectivo.
- Conservar la memoria de la organización más allá de los individuos que trabajan en ella y poder aprovechar el valor de los contenidos en los que queda plasmada la experiencia, evitando empezar de cero sobre aspectos en los que ya hay experiencia acumulada.

Por todo lo antes expuesto, cabe destacar la importancia que tienen las nuevas herramientas informáticas en los sistemas de gestión documental de última generación para la creación, control y gestión de documentos incluidos en los procesos de negocio.

---

<sup>1</sup>Sistema de Gestión Documental

### 1.3. Gestión de procesos de negocio

Para un mayor entendimiento sobre que es gestión de procesos de negocio se verá primeramente que es un proceso de negocio. Este se define como: *“el conjunto de todas las tareas y actividades coordinadas formalmente, dirigidas tanto por personas como por equipos, que lleva a conseguir un objetivo organizativo específico”* [9]. Es decir, es el conjunto de actividades que utilizan entradas transformándolas en salidas, dichas salidas proporcionan valor al cliente y permiten alcanzar las metas de la organización.

Hace unos años nadie había oído hablar acerca de la Gestión de Procesos de Negocio (*Business Process Management*), pero, este se fue convirtiendo en la tendencia de gestión empresarial y tecnología más popular de nuestro tiempo. En la actualidad se conoce como Gestión de Procesos de Negocio (BPM por sus siglas en inglés) *“al conjunto de métodos, herramientas y tecnología utilizados para diseñar, representar, analizar y controlar procesos de negocio. Es un enfoque centrado en los procesos para mejorar el rendimiento que combina las tecnologías de la información con metodologías de procesos y gobierno”* [9].

La gestión de procesos de negocio tiene como objetivo mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio, englobando a todos los procesos que son parte del ciclo de vida de un negocio. Mediante BPM, los procesos de negocio son más efectivos y más ágiles. Los problemas se resuelven antes de que se conviertan en asuntos más delicados. Los procesos producen menos errores y estos se detectan más rápido y se resuelven antes [9].

### 1.4. Flujo de trabajo

Un flujo de trabajo se define como: *“un sistema de secuencia de tareas de un proceso de negocio. Su definición y control puede ser manual, informatizado o mixto. Organiza y controla tareas, recursos y reglas necesarias para completar el proceso de negocio”* [10]. Este se relaciona con la automatización de los procedimientos que se llevan a cabo en una organización, con el fin de culminar una meta común, donde la documentación, información o tareas necesarias para realizar los procedimientos, son enviadas entre sus participantes de acuerdo con un conjunto de reglas previamente establecidas por la organización. El flujo de

trabajo ofrece las facilidades para modelar y gestionar los diversos procesos que ocurren dentro de una institución.

Los flujos de trabajo traen consigo un conjunto de beneficios los cuales son reflejados por uno de los pioneros de la Gestión de Procesos de Negocio (BPM) en España Renato de Laurentiis Gianni [11], estos son los principales:

- Mejora la atención y servicio al cliente.
- Involucran a diferentes personas y departamentos.
- Minimiza el tiempo requerido por los participantes para acceder a la documentación.
- Disminuye el tiempo de transferencia de trabajo, información y documentos entre actividades.
- Disminuye el tiempo que los participantes, supervisores y administradores necesitan para conocer la situación de un trabajo.
- Disponibilidad de mecanismos para una mejor gestión y optimización de procesos.

Según Renato de Laurentiis Gianni [11] las funciones más comunes que proporcionan los flujos de trabajo son:

- Asignar actividades a las personas de forma automática y según cualquier criterio, o según cargas de trabajo.
- Avisar al personal de las actividades pendientes, las cuales son parte de una cola de flujos de trabajo.
- Permitir la colaboración en las tareas comunes.
- Optimizar los recursos humanos y técnicos, alineándolos a la estrategia de la empresa.
- Automatizar las secuencias de los procesos de negocio.
- Agilizar los procesos de negocio, teniendo como resultado un mejor servicio al cliente.

- Asignarle proactivamente a las personas que deben ejecutar las actividades, todos los recursos necesarios (documentos e información) en cada una de ellas.
- Definir y controlar alertas según criterios de tiempo, de evento o de condición, provocando así algún mensaje a un supervisor.

Renato de Laurentiis Gianni [11] plantea que con las tecnologías de los flujos de trabajo:

- El trabajo no queda atascado o extraviado.
- Los jefes pueden enfocarse más en los problemas del negocio y del personal, tal como el rendimiento y capacitación individual, mejoras de procedimientos, y casos especiales, más que en la rutina de asignación de tareas.
- Los procedimientos son formalmente documentados y seguidos de forma exacta y estándar, asegurando que el trabajo es llevado a cabo en la forma planificada, cumpliendo a su vez todos los requerimientos y normas del negocio y externos.
- La persona adecuada, es asignado a cada caso, y los casos más importantes o críticos en el tiempo, son asignados primero. Los usuarios no gastan tiempo escogiendo sobre cual caso trabajar, aplazando quizás aquellos casos más importantes pero de mayor dificultad.
- Se logra el procesamiento paralelo, donde 2 o más actividades no dependientes pueden ser realizadas concurrentemente, generando así beneficios en cuanto a reducción de tiempo de los procesos y mejor servicio al cliente.

## 1.5. Flujo de trabajo en los sistemas de gestión documental

### Nuxeo

Nuxeo es un sistema de gestión de contenidos empresariales. El mismo permite gestionar documentos de forma cómoda, realizando versiones, flujos de trabajo asociados a documentos, publicación remota o búsqueda

avanzada. Este sistema incluye una función de flujo de trabajo intuitiva que permite a los usuarios configurar fácilmente tareas sencillas o tareas con varias etapas. El flujo de trabajo ayuda a que el contenido fluya a través de contenidos estructurados o procesos de negocio. Los flujos de trabajo se pueden establecer rápidamente, guardándose para su posterior reutilización y con una plantilla para garantizar la coherencia de los procesos a través de los equipos. Nuxeo dispone del motor jBPM<sup>2</sup>, el cual es utilizado para la administración de flujos de trabajo [12].

En el Libro Blanco Gestión Documental Open Source [12], se plantea que Nuxeo tiene dos tipos de flujos de trabajo:

- Flujo de trabajo en paralelo.
- Aprobación de flujo de trabajo.

La diferencia entre ellos es el orden de participación. En el flujo de trabajo en paralelo, todos los usuarios pueden revisar el documento tan pronto como el flujo de trabajo se ha iniciado. Por otro lado, el flujo de trabajo de aprobación, es un flujo de trabajo ordenado, esto significa que los participantes puedan revisar el documento solo cuando el usuario anterior de la lista ha aprobado el documento [12].

### **Knowledge Tree**

Es un sistema de gestión electrónica de documentos, este cuenta con un editor de flujos de trabajo y de metadatos lo que facilita la organización y clasificación de los documentos que pueden estar en más de 25 formatos de archivos electrónicos o digitales. Una de sus principales ventajas sobre otros sistemas similares es su indexador de documentos, que busca contenidos dentro de los documentos. Además de presentar otras funcionalidades como control de acceso a carpeta y documentos de acuerdo con roles y permisos y escritorio de usuarios personalizable [12].

El manejo de los flujos de trabajo en Knowledge Tree es gestionado a través del motor ProcessMarker. Este sistema tiene un módulo de flujos de trabajo que permite automatizar los procesos de negocio, mediante el diseño de la ruta para la revisión de los documentos según el proceso de aprobación definido en la

---

<sup>2</sup>Motor para la administración de flujos de trabajo.

organización. El mismo también permite definir alertas cuando los documentos están listos para revisión o recibir una notificación cuando el contenido es creado o modificado en una carpeta [12].

### **Aviladoc**

Es una aplicación *web* para el control de documentos, desarrollada sobre la plataforma de *software* libre con una base de datos centralizada destinada a la gestión, tramitación y resguardo de archivos electrónicos y/o digitales; facilitando la búsqueda o recuperación de información de forma rápida y sencilla. Este permite controlar la entrada y salida de documentos y gestionar el tránsito interno de los mismos, constituyendo un archivo digital que agiliza las tareas de registro, búsqueda, reproducción y distribución de los documentos. Además, ofrece un historial del documento desde que fue creado hasta su eliminación. Este no tiene una funcionalidad para la gestión de flujos de trabajo. Lo que permite es enlazarse a Microsoft Outlook para generar automáticamente tareas a cumplir, notificaciones e información asociada al documento [13].

### **Alfresco**

Gestión de Contenido Empresarial (ECM) Alfresco es considerado el líder en soluciones de código abierto para la *web* y gestión de contenidos. Lo que hace a Alfresco único son sus servicios, estos se pueden ampliar fácilmente con secuencias de comando *web* a través de servicios REST [14].

La plataforma de contenidos de Alfresco, utiliza una arquitectura de estándares abiertos para proporcionar gestión documental, gestión de contenidos web, gestión de registros y *software* colaborativo. Alfresco está construido mediante los últimos componentes de infraestructuras de código abierto, que incluyen: Spring, Hibernate, Lucene y MyFaces. Este se basa en Programación Orientada a Aspectos [15].

Una de las características que brinda Alfresco es que incluye el motor jBPM para el trabajo de procesos de negocio. Este motor de flujo de trabajo es una herramienta muy completa. Está bien integrada en la aplicación y permite llevar a cabo flujos de trabajo profesionales [12].

### **Flujos de trabajo en Alfresco**

En el libro *Alfresco Enterprise Content Management Implementation* se define que un “*flujo de trabajo es una automatización de un proceso de negocio, durante el cual los documentos pasan de una persona a otra para una acción, de acuerdo con un conjunto de procedimientos y reglas. Para algunas empresas un flujo*”

*de trabajo podría ser un simple proceso de aprobación y para otras podría ser una gestión de procesos de negocio complejos. Este proporciona la propiedad y el control sobre el contenido y los procesos” [16].*

En el libro *Alfresco Practical Solutions for Enterprise Content Management* [14] se mencionan los tipos de flujos de trabajo que incluye Alfresco, los cuales son:

- Flujo de trabajo simple: flujo de trabajo que es orientado a los contenidos.
- Flujo de trabajo avanzado: flujo de trabajo que es orientado a las tareas.

Munwar Shariff [16] define como características de los flujos de trabajo:

- Aprovecha la potencia del motor de flujos de trabajo JBoss jBPM integrados en Alfresco.
- Se puede modelar cualquier proceso de negocio incluyendo decisiones, divisiones, uniones, flujos de trabajo paralelos, sub-procesos, estados de espera y temporizadores.
- Puede incluir lógica de negocio escrita en Java o JavaScript, con acceso a la API<sup>3</sup> (Interfaz de Programación de Aplicaciones) de Alfresco

## **Resultados**

Finalizando el análisis realizado a algunos sistemas de gestión documental nacionales e internacionales sobre cómo estos manejan los flujos de trabajo avanzados. Se llega a la conclusión que para la gestión de los mismos se utilizan motores, los sistemas Nuxeo y Alfresco emplean el motor jBPM y Knowledge Tree emplea el motor ProcessMarker. En el caso del Aviladoc este no cuenta con una funcionalidad para poder gestionar los flujos de trabajo avanzados. El análisis realizados a estos sistemas fue un aspecto clave que ayudó a tener una visión de los requisitos que se deben capturar para permitir la correcta gestión de los flujos de trabajo avanzados. Además, aportó el conocimiento adecuado que posibilitará el correcto diseño de los prototipos de las interfaces que serán presentadas a los clientes.

---

<sup>3</sup>Interfaz de programación de aplicaciones, es un conjunto de funciones que ofrece una biblioteca para ser utilizado por otro *software* como una capa de abstracción.

## 1.6. Metodología de desarrollo de *software*

Las metodologías de desarrollo de *software* son un grupo de procedimientos que se deben seguir para desarrollar un *software* o aplicación, en otras palabras, es una guía la cual tiene como objetivo obtener el producto deseado a través de actividades. Estas van orientando cómo separar el proyecto por etapas, las tareas que se deben ir haciendo en estas y las herramientas que se podrán emplear.

### ***RUP (Proceso Rational Unificado)***

RUP es más que un simple conjunto de actividades necesarias para transformar los requerimientos del usuario en el sistema de *software*, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de *software*. Este define claramente quién, cómo, cuándo y qué debe hacerse en un proyecto. El proceso unificado combinado con UML como lenguaje de modelado conforma la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientado a objetos [17].

Estos son las tres características por lo cual está conformado RUP.

**Dirigido por casos de uso:** Tiene los casos de uso como el hilo conductor que orienta las actividades de desarrollo. Se centra en la funcionalidad que el sistema debe poseer para satisfacer las necesidades de un usuario que interactúa con él. Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de entonces los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo) [17].

**Centrado en la arquitectura:** Abarca diferentes vistas del sistema: estructural, funcional, dinámica, la plataforma en que se va a desarrollar y la forma del sistema. La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP propone el desarrollo del *software* mediante iteraciones comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura [17].

**Iterativo e incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencias a pasos en los flujos de trabajo, los incrementos y al crecimiento del producto. Cada iteración se realiza de forma planificada por lo que se dice que son mini-proyectos [17].

RUP divide el proceso de desarrollo del *software* en cuatro fases: inicio, elaboración, construcción y transición, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto [17].

La metodología que se ha seleccionado como guía de desarrollo de la solución propuesta es (*Proceso Rational Unificado*) RUP con Nivel 2 de CMMI. Unas de las características que hacen de RUP una buena elección para utilizar como metodología de desarrollo de *software* es el ser iterativo, lo que permite reducir riesgos y dividir los proyectos en pequeños ciclos o iteraciones a través de cada una de las fases. Además de ser la metodología que se definió y se utiliza para el desarrollo del eXcriba.

## 1.7. Lenguajes

### Lenguajes de programación

#### PHP (*Hypertext Preprocessor*)

Es un lenguaje de código abierto interpretado de alto nivel para la creación de páginas *web* dinámicas que permite la creación de aplicaciones con interfaz gráfica, conexión a servidores de base de datos como Oracle, MySQL y Postgres. Puede ser ejecutado en sistemas operativos como: Windows, Linux y Mac OS. Es utilizado para desarrollar aplicaciones que son montadas en servidores *web*, y puede ser embebido en páginas HTML<sup>4</sup>. Además el mismo tiene abundante documentación [18].

Se decide utilizar PHP en su versión 5.3 ya que es el lenguaje propuesto por la arquitectura del sistema sobre el cual se ha desarrollado el cliente *web* de eXcriba, que es la aplicación con la cual interactúan los

---

<sup>4</sup>HyperText Markup Language, este es un lenguaje de marcado.

usuarios finales, por lo que al desarrollarse las diferentes funcionalidades que le darán cumplimiento a la presente investigación se debe seguir esta restricción de diseño.

### **JavaScript**

JavaScript es un lenguaje basado en objetos e interpretados por los navegadores web. Es utilizado para crear pequeños programas encargados de realizar acciones dentro de una página web. Los códigos escritos en este lenguaje permiten la interactividad entre el usuario y la página, controlar el navegador o crear páginas completas en HTML [19].

En el desarrollo del lado del cliente se implementarán las interfaces de usuarios utilizando la biblioteca jQuery, la cual se explicara más adelante. Para la parte del servidor que es donde se desarrollaran los servicios es factible el uso de las API de JavaScript que provee Alfresco, ya que esta permite a los desarrolladores interactuar con los objetos del repositorio de contenidos. La versión de JavaScript que se utilizará del lado del cliente es la 1.5.

### **Lenguaje de modelado**

#### **UML**

(*Unified Modeling Language*) es un lenguaje gráfico que permite visualizar, construir y documentar los elementos que forman un sistema de *software* orientado a objetos, de ahí que se pueda realizar un análisis en UML indistintamente del lenguaje en el que finalmente sea implementado el sistema. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de *software* reusables [17].

Para el desarrollo de la aplicación se utilizará UML en su versión 2.0, como el lenguaje con que se modelarán los artefactos que se creen en el proceso de desarrollo del *software*. Se elige este lenguaje, ya que el mismo es el que se emplea asociado a la metodología de desarrollo que se seleccionó (RUP).

## **1.8. Herramienta CASE**

### **Visual Paradigm**

Visual Paradigm versión 8.0 para UML, es una herramienta CASE que provee el modelado de procesos de negocio, además de un generador de mapeo de objetos relacionales para los lenguajes de programación Java, .NET y PHP. Es una herramienta que sustenta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Posibilita la importación y exportación de ficheros XML. Este está diseñado para una amplia gama de usuarios, incluidos los ingenieros de *software*, analistas de sistemas y analista de negocios [20].

Para el modelado de diagramas se utilizará Visual Paradigm ya que es una herramienta multiplataforma para modelado UML. Permite la ingeniería inversa, es decir, invertir modelos UML desde el código fuente.

## 1.9. Tecnologías

### **REST (*Representational State Transfer*)**

REST se define como una técnica o estilo arquitectónico de *software* orientado a sistemas de hipermedia distribuido, como la (*World Wid Web*) *www* y a su vez es una abstracción de los elementos arquitectónicos dentro de tales sistemas. El mismo ignora los detalles de implementación de los componentes y la sintaxis del protocolo centrándose en las funciones de los mismos, las limitaciones de la interacción de estos con otros y su interpretación de elementos de datos significativos. Abarca las limitaciones sobre los componentes, los conectores y los datos que definen la base de la arquitectura *web* y por consiguiente la esencia de su comportamiento como una aplicación basada en la red [21].

La implementación de servicios siguiendo los principios de REST utilizando HTTP, posibilita que al realizar una solicitud al servicio este no retorne la base de datos completa, sino un tipo de datos interpretable por el cliente (JSON<sup>5</sup>, XML, HTML). Los sistemas basados en los principios de REST se conocen como RESTful y a los servicios de estos, que se implementan según lo anteriormente expuesto, son conocidos como RESTful *Web Service* [22].

---

<sup>5</sup>JSON: acrónimo de JavaScript Object Notation (en español Notación de Objetos de JavaScript), es un formato de texto para el intercambio de datos.

Dado que el *software* Gestor de Documentos Administrativos eXcriba tiene todos sus servicios implementados en el estilo arquitectónico REST, se hace necesario desarrollar los servicios correspondientes al módulo gestión de flujos de trabajo avanzados en esta misma técnica. El aprovechamiento y uso de la API RESTful que brinda el ECM Alfresco para el cumplimiento del objetivo fundamental del presente trabajo de diploma, se fundamenta en la facilidad de uso de dicha API, la amplia documentación referente al tema, así como una vasta comunidad para el intercambio y la colaboración.

### 1.9.1. Marco de trabajo

#### CodeIgniter

CodeIgniter es un marco de trabajo abierto que permite crear aplicaciones *web* con el lenguaje PHP. Su principal objetivo es ayudar a que los desarrolladores, puedan realizar proyectos mucho más rápido que creando toda la aplicación *web* desde cero. Este es un marco de trabajo que facilita la escritura de código repetitivo. Además, de ser totalmente extensible y altamente compatible con gran variedad de versiones y configuraciones de PHP [23].

Se decide utilizar CodeIgniter en su versión 1.7.2 del lado del servidor como marco de trabajo para implementar la lógica de negocio de la solución a desarrollar, ya que este es una restricción de la arquitectura como el marco de trabajo para la capa de aplicación.

#### jQuery

jQuery es una biblioteca de JavaScript que permite simplificar la manera de interactuar con las páginas HTML, permitiendo manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX<sup>6</sup> a las páginas web. Además, permite el intercambio asíncrono de datos entre cliente y servidor de manera sencilla. Esta biblioteca incluye los efectos y eventos de AJAX, las funcionalidades de DOM<sup>7</sup>, se puede

---

<sup>6</sup>Asynchronous JavaScript and XML.

<sup>7</sup>Document Object Model o DOM ('Modelo de Objetos del Documento') es esencialmente una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

manipular el estilo CSS<sup>8</sup> de las páginas o elementos HTML [24]. La versión que se va a utilizar para la implementación de las interfaces de usuario es la 1.3.2

### **Web Scripts**

Un Web Scripts es simplemente una URI unido a un servicio utilizando los métodos estándar de HTTP, como GET, POST, PUT o DELETE. Los webscript<sup>9</sup> se pueden escribir usando simplemente la API de JavaScript de Alfresco y las plantillas FreeMarker. Alfresco provee el framework de Web Scripts, una API basada en tecnologías RESTful que proporciona una forma de interactuar con el repositorio de contenido y de integrar Alfresco con otros sistemas [25].

### **FreeMarker**

FreeMarker es un motor de plantillas<sup>10</sup> así como una herramienta genérica para generar la salida de texto basado en plantillas. No es una aplicación para los usuarios finales en sí mismo, sino un paquete de Java que los programadores pueden utilizar en sus productos [26].

Este motor es usado por el framework Web Scripts de Alfresco para proveer un mecanismo de generación de respuestas preformateadas, una vez finalizada la ejecución de un webscript, o sea, una vez finalizada la ejecución de la lógica de negocio se emite una respuesta cuya salida queda determinada por la estructura de la plantilla de presentación correspondiente al servicio en ejecución.

Una de las principales ventajas que usa el framework Web Scripts de este motor es que permite generar diferentes vistas y representaciones como resultado de la ejecución de un mismo servicio. De esta manera se logra la separación definitiva entre la lógica de negocio que se ejecuta durante una petición y el formato de salida de la respuesta generada.

---

<sup>8</sup>Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos.

<sup>9</sup>Implementación de un servicio utilizando el framework Web Scripts.

<sup>10</sup>Una plantilla de presentación es un documento de texto que, aplicado a un modelo de datos produce una salida (un nuevo documento) que tiene la estructura definida en la plantilla.

# Capítulo 2

## Características del módulo

### 2.1. Problema y situación problemática

Debido al interés que presenta el Gestor de Documentos Administrativos eXcriba actualmente para gestionar y automatizar los procesos de negocio, así como el reasignar, definir y notificar tareas o asignación de la misma, surge la necesidad de crear un producto informático que pueda satisfacer los problemas que enfrentan diariamente las organizaciones, de modo que se puedan gestionar y que responda a las necesidades de los clientes, cumpliendo con los requerimientos mínimos de calidad.

Como alternativa para resolver esta situación es necesario ampliar las funcionalidades en el *software* eXcriba, considerando que este no solo permita gestionar los documentos a lo largo de su ciclo de vida enfocado al control de la información, sino que también, posibilite administrar los procesos de negocio que se llevan a cabo en las empresas a través de la gestión de flujos de trabajo avanzados en el cliente *web* del eXcriba.

### 2.2. Propuesta de solución

Para dar respuesta al problema planteado anteriormente se propone, el desarrollo de un módulo para gestionar los flujos de trabajo avanzados en el Gestor de Documentos Administrativos eXcriba, con el cual se podrá llevar un mayor control y organización sobre los flujos de trabajo correspondientes a los usuarios. Para la implementación del módulo se dividirá en dos elementos fundamentales.

Primeramente, se tendrá el desarrollo de los servicios que se encargarán de gestionar los flujos de trabajo avanzados a los documentos almacenados en el repositorio de contenidos del *software* eXcriba. Para la

realización de estos servicios se propone emplear el marco de trabajo WebScript que tiene incluido el ECM Alfresco utilizando la API de flujo de trabajo.

El segundo elemento de la propuesta consiste en realizar las interfaces de usuario para gestionar los flujos de trabajo avanzados desde el cliente *web* del eXcriba, así como la implementación de la lógica del negocio, utilizando los marcos de trabajo CodeIgniter y jQuery.

Una vez terminada estos dos elementos vendría la comunicación entre ellos, para lo cual se propone hacer uso del módulo `php5-curl`<sup>1</sup>.

El módulo, en el cliente *web* del eXcriba permitirá al usuario crear los flujos de trabajo avanzados. Además, tendrá otras funcionalidades como notificar y ver los flujos de trabajo avanzados que se le hayan asignado, así como modificar el estado de los mismos de acuerdo con el progreso que se vaya obteniendo y reasignar las tareas a diferentes usuarios.

Con la creación de los flujos de trabajo avanzados se podrán asignar tareas a usuarios brindando los datos siguientes: fecha límite, prioridad, descripción y usuarios asignados. Además, el módulo debe de ser capaz de notificar a los usuarios cuando tengan tareas asignadas.

## 2.3. Modelo de dominio

*“El modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno en el que trabaja el sistema” [17].*

Es decir, el modelo de dominio puede ser tomado como el punto inicial para el diseño del sistema. Este puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño. Es una representación visual de los conceptos u objetos del mundo real, significativos para un problema o área de interés. El modelo de dominio es utilizado por el analista como un medio para comprender el sector de negocios al cual el sistema va a servir.

---

<sup>1</sup>php5-curl es un módulo para obtener archivos desde servidores FTP, GOPHER y HTTP.

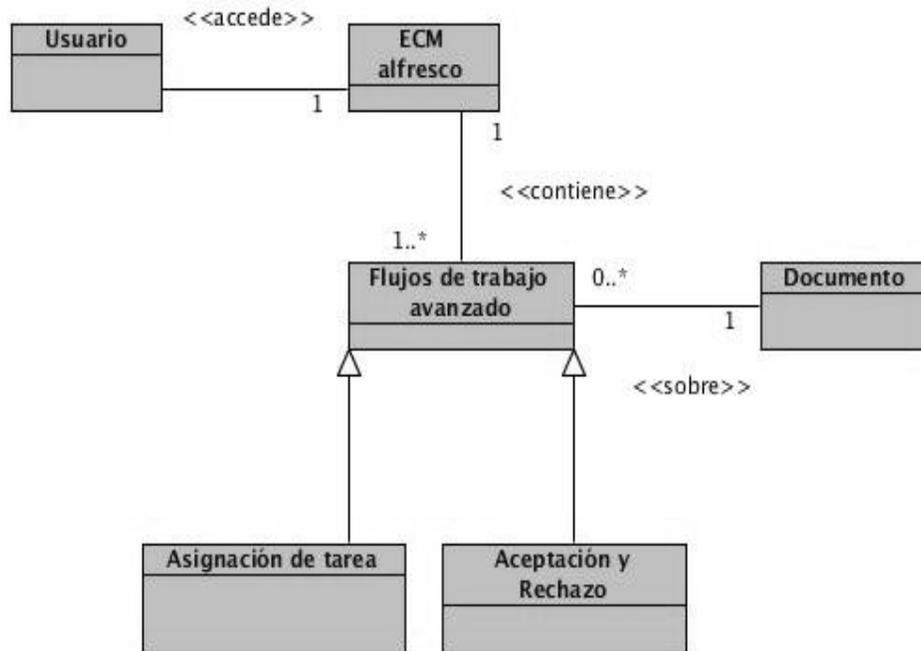


Figura 2.1: Modelo de dominio.

**Usuario:** Es la persona que interactúa con el sistema.

**ECM Alfresco:** Gestor de contenido empresarial que contiene los flujos de trabajo avanzados.

**Flujo de trabajo avanzado:** Permite automatizar una secuencia de actividades o tareas utilizadas para la ejecución de un proceso.

**Documento:** Un documento es la información creada o recibida por un usuario en cualquier formato.

**Asignación de tarea:** Permite automatizar la asignación de una actividad o tarea a un usuario.

**Aceptación y Rechazo:** Acepta o rechaza la automatización de una secuencia de actividades o tareas asignadas al usuario.

## 2.4. Especificación de los requisitos de *software*

Los requerimientos de *software* son condiciones o capacidades que tienen que ser alcanzada por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente [27].

### 2.4.1. Técnicas para la captura de requisitos

En principio, parece bastante simple preguntar al cliente, a los usuarios y a los que están involucrados en los objetivos del sistema o producto y sean expertos, investigar cómo los sistemas o productos se ajustan a las necesidades del negocio y finalmente, cómo el sistema o producto va a ser utilizado en el día a día. Esto que parece simple, es muy complicado [28].

Por esa razón surgen diferentes técnicas que ayudan a comprender el problema, proponer soluciones, negociar diferentes puntos de vista y finalmente especificar un conjunto básico de requisitos de la solución. Ejemplos de estas técnicas pueden ser: Entrevistas, Cuestionarios, Tormentas de ideas, Análisis de sistemas existentes, Arqueología de documentos y Prototipos.

Para la realización del levantamiento de requisitos del módulo se aplicaron las técnicas de Casos de uso, Escenarios y Prototipos.

- **Prototipos:** Un prototipo es un borrador de un producto potencial o de una parte del mismo. Es una simulación de los requisitos. En ocasiones los analistas no pueden continuar su trabajo porque les faltan datos. En esos casos el analista o el resto de las personas involucradas necesitan trabajar con algo más concreto que una lista de requisitos escrito y para esto utilizan un prototipo. Este prototipo también es evaluado por el cliente y es utilizado para refinar los requerimientos del *software* a ser desarrollado [29].
- **Casos de uso:** Los casos de uso identifican el qué y el cómo del comportamiento del producto. Estos describen las iteraciones entre el usuario y el sistema, teniendo como objetivo fundamental que realiza el sistema para el usuario [29].

- **Escenarios:** Los escenarios son una descripción de un caso de uso del negocio, este no tiene tanto detalle. Su objetivo es hacer entender cómo funciona el caso de uso [29].

## 2.4.2. Requerimientos funcionales

*“Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, no alteran la funcionalidad del producto, por lo que se mantienen invariables sin importarle con que propiedades o cualidades se relacionen”* [30], es decir, son las cosas que el sistema puede hacer.

**RF-1** Crear flujo de trabajo avanzado: El módulo debe permitir al usuario crear flujos de trabajo avanzados asociados a documentos.

**RF-2** Modificar flujo de trabajo avanzado: El módulo debe permitir al usuario modificar el estado del flujo de trabajo que ha sido creado a un documento.

**RF-3** Listar los flujos de trabajo avanzados: El módulo debe permitir al usuario listar los flujos de trabajo pendientes que le fueron asignados.

**RF-4** Reasignar flujos de trabajo avanzados: El módulo debe permitir al usuario reasignar los flujos de trabajo que están pendientes.

**RF-5** Notificar asignación de flujo de trabajo avanzado a usuario: El módulo debe notificar al usuario cuando se le ha asignado un flujo de trabajo avanzado.

**RF-6** Listar notificaciones: El módulo debe mostrar las notificaciones que tenga pendientes el usuario.

**RF-7** Modificar estado de notificación: El módulo debe permitir al usuario cambiar el estado de las notificaciones de no leída a leída.

## 2.4.3. Requerimientos no funcionales

*“Los requisitos no funcionales son los requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste, como la fiabilidad y el tiempo de respuesta”* [30], es decir, los requisitos no funcionales representan aquellos atributos que debe exhibir el sistema pero que no son funcionalidad específica.

Requerimientos no funcionales que debe de tener el módulo son:

■ **Usabilidad**

1. Se utilizará el idioma español para los mensajes y texto de la interfaz
2. Actualizar el manual de usuario del *software* eXcriba agregándole los pasos a seguir para realizar las funcionales que ofrece el módulo.

■ **Soporte**

1. Debe tener instalado Mozilla Firefox 6.X o superior.

■ **Portabilidad**

1. Se podrá utilizar la aplicación en todos los sistemas operativos. Se recomienda GNU/Linux.

■ **Legales**

1. Las herramientas seleccionadas para el desarrollo del módulo están respaldadas por licencias libres, con las condiciones de *software* libre. Exceptuando el IDE<sup>2</sup> de desarrollo Zend-Studio.

■ **Interfaz**

1. **Interfaz de comunicación**

- La comunicación con el repositorio Alfresco es mediante el estilo arquitectónico REST.

■ **Restricciones del diseño**

1. Mantener un sistema de codificación estándar siguiendo las pautas establecidas en el documento de Línea Base de la Arquitectura.
2. Utilizar jQuery 1.3.2 como biblioteca fundamental para el diseño de la interfaz de usuario final.

---

<sup>2</sup>Entorno de Desarrollo Integrado.

3. Utilizar servidor *web* Apache 2.2.
4. Utilizar CodeIgniter 1.7.2 como marco de trabajo.
5. Implementar el módulo utilizando el lenguaje de programación PHP versión 5.3.
6. Mantener un sistema de codificación estándar siguiendo las pautas establecidas en el documento de Línea Base de la Arquitectura.
7. Metodología de desarrollo de *software* RUP, usando el lenguaje de modelación UML.

## 2.5. Definición del actor y los casos de uso del sistema

### 2.5.1. Definición de los actores

En el módulo que se está modelando solo actúa un actor denominado "usuario" que es el que interactúa con el *software*.

Actores	Justificación
Usuario	Es la persona que puede realizar todas las operaciones relacionadas con los flujos de trabajo avanzados en el eXcriba.

Tabla 2.1: Definición de los actores.

### 2.5.2. Definición de los casos de uso del módulo

Breve descripción de los casos de uso que tendrá el módulo.

<b>Caso de uso:</b>	<b>Gestionar flujos de trabajo avanzados.</b>
<b>Actor:</b>	Usuario: (Inicia) Crea, modifica y ve los flujos de trabajo.
<b>Descripción:</b>	El usuario accede al sistema para crear un flujo de trabajo.
<b>Referencias:</b>	RF.1, RF.2, RF.3 y RF.5

Tabla 2.2: Definición del caso de uso gestionar flujo de trabajo avanzado.

<b>Caso de uso:</b>	<b>Reasignar flujo de trabajo.</b>
<b>Actor:</b>	Usuario: (Inicia) Reasigna los flujos de trabajo que han sido mandados a un usuario.
<b>Descripción:</b>	El usuario accede al sistema para reasignar un flujo de trabajo hacia otro usuario, ya sea como medio de notificación o para asignar el mismo flujo de trabajo a otro usuario.
<b>Referencias:</b>	RF.4 y RF.5

Tabla 2.3: Definición del caso de uso reasignar flujos de trabajo.

<b>Caso de uso:</b>	<b>Listar notificaciones.</b>
<b>Actor:</b>	Usuario: (Inicia)
<b>Descripción:</b>	El usuario accede al sistema para listar los flujos de trabajo que tiene pendiente.

Continúa en la próxima página

<b>Referencias:</b>	RF.6
---------------------	------

Tabla 2.4: Definición del caso de uso listar notificaciones.

<b>Caso de uso:</b>	<b>Modificar estado de las notificaciones</b>
<b>Actor:</b>	Usuario: (Inicia)
<b>Descripción:</b>	El usuario accede al sistema para listar los flujos de trabajo que tiene pendiente. Luego realiza la acción modificar el estado de los flujos de trabajo.
<b>Referencias:</b>	RF.7

Tabla 2.5: Definición del caso de uso modificar estado de las notificaciones.

### 2.5.3. Diagrama de caso de uso del módulo

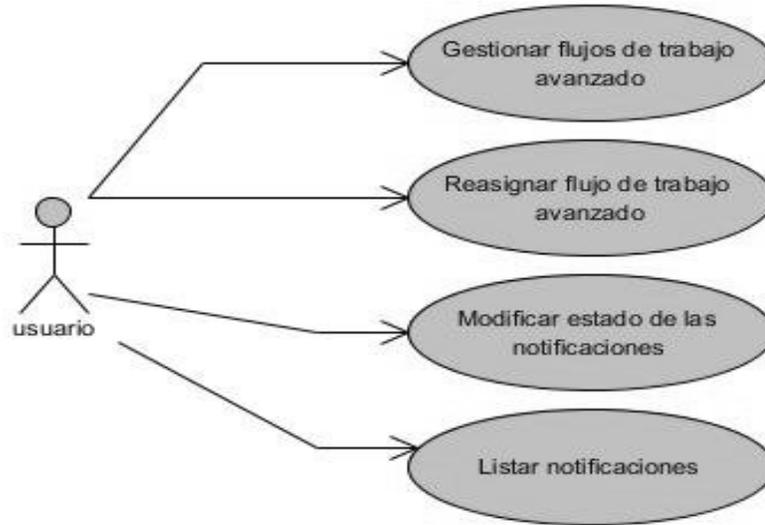


Figura 2.2: Diagrama de Casos de Uso

### 2.5.4. Descripción de los casos de uso del módulo

Ver Anexo A de la versión extendida del documento para ver las otras descripciones de los casos de uso.

<b>Caso de uso</b>	<b>Gestionar flujos de trabajo avanzados</b>
<b>Actor</b>	Usuario (inicia) Crea, modifica y lista los flujos de trabajo avanzados.
<b>Resumen</b>	El caso de uso comienza cuando el usuario desea gestionar un flujo de trabajo avanzado. Para ello puede crear, modificar o listar los mismos. El caso de uso termina cuando realiza alguna de estas acciones.
<b>Prioridad</b>	Alta
<b>Complejidad</b>	Alta
Continúa en la próxima página	

<b>Precondiciones</b>	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder crear un flujo de trabajo avanzado. Para las secciones de modificar y listar debe existir al menos un flujo de trabajo avanzado creado.
<b>Poscondiciones</b>	Al usuario se le envía una notificación cuando se crea un flujo de trabajo avanzado. Se almacenan en el repositorio los datos de los flujos de trabajo avanzados.
<b>Flujo de eventos</b>	
<b>Sección 1: “Crear flujo de trabajo avanzado”</b>	
<b>Actor</b>	<b>Sistema</b>
1. Selecciona el ícono “Crear flujo de trabajo avanzado” del documento sobre el cual va a crear el nuevo flujo de trabajo.	2. Muestra en una nueva ventana el formulario con los campos necesarios para crear un flujo de trabajo avanzado.
3. Inserta los datos necesarios y envía la información presionando un botón.	4. Verifica que los campos obligatorios estén llenos.
5.	6. En caso correcto almacena en el repositorio el flujo de trabajo avanzado creado y muestra un mensaje de confirmación.
	7. Notifica al usuario que se le asignado un flujo de trabajo. Termina el caso de uso.
<b>Flujos alternos</b>	
<b>Actor</b>	<b>Sistema</b>
3.1. Una vez insertados los datos presiona el botón para limpiar el formulario.	Todos los datos insertados en el formulario son eliminados.
Continúa en la próxima página	

4.1.	Muestra un mensaje con el nombre del campo que está vacío.
<b>Sección 2: “Modificar flujo de trabajo avanzado”</b>	
<b>Actor</b>	<b>Sistema</b>
1. Selecciona en la barra de funcionalidad el botón para listar los flujos de trabajo que tiene pendientes.	2. Muestra una interfaz con un listado de los flujos de trabajo que tiene pendiente el usuario.
3. Selecciona el ícono “Modificar flujo de trabajo avanzado” del que se desea modificar.	4. Muestra en una ventana un formulario con algunos datos del flujo los cuales no puede editar y el campo Estado sobre el cual podrá realizar las modificaciones.
5. Modifica el estado del flujo y presiona el botón para enviar los cambios.	6. Almacena los cambios realizados en el repositorio. Termina el caso de uso.
<b>Flujos alternos</b>	
<b>Actor</b>	<b>Sistema</b>
	2.1 Muestra el mensaje: “No existen tareas pendientes”
5.1 Modifica el estado del flujo y cancela el envío.	
	6.1 No se realiza ningún cambio y se muestra la interfaz con el listado de flujos de trabajo pendientes del usuario.
<b>Sección 3: “Listar flujos de trabajo avanzados”</b>	
<b>Actor</b>	<b>Sistema</b>
Continúa en la próxima página	

1. Selecciona en la barra de funcionalidad el botón para listar los flujos de trabajo que tiene pendientes.	2. Muestra una interfaz con un listado de los flujos de trabajo que tiene pendiente el usuario. Termina el caso de uso.
<b>Flujos alternos</b>	
<b>Actor</b>	<b>Sistema</b>
	2.1 Muestra el mensaje: “No existen tareas pendientes””

Tabla 2.6: Descripción textual del caso de uso: 3.1 Gestionar flujos de trabajo avanzados.

# Capítulo 3

## Diseño del módulo

### 3.1. Modelo de diseño

El modelo de diseño es utilizado para modelar los aspectos dinámicos del sistema. Consta de un conjunto de objetos que describen las realizaciones de los casos de uso y sus relaciones. Se centra en los impactos que producen en el sistema los requisitos funcionales y no funcionales. De manera general el modelo de diseño constituye una abstracción al modelo de implementación y del código fuente, o sea, es la entrada o el punto de partida para las posteriores actividades de implementación del sistema que se desea desarrollar [17].

#### 3.1.1. Diagrama de clases de diseño

Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases en sí, muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Los mismos se utilizan para reflejar la vista de diseño estática de un sistema. Son la base para los posteriores diagramas de componentes y los de despliegue. Su importancia radica en que permitirá construir sistemas ejecutables aplicando ingeniería directa e inversa [17].

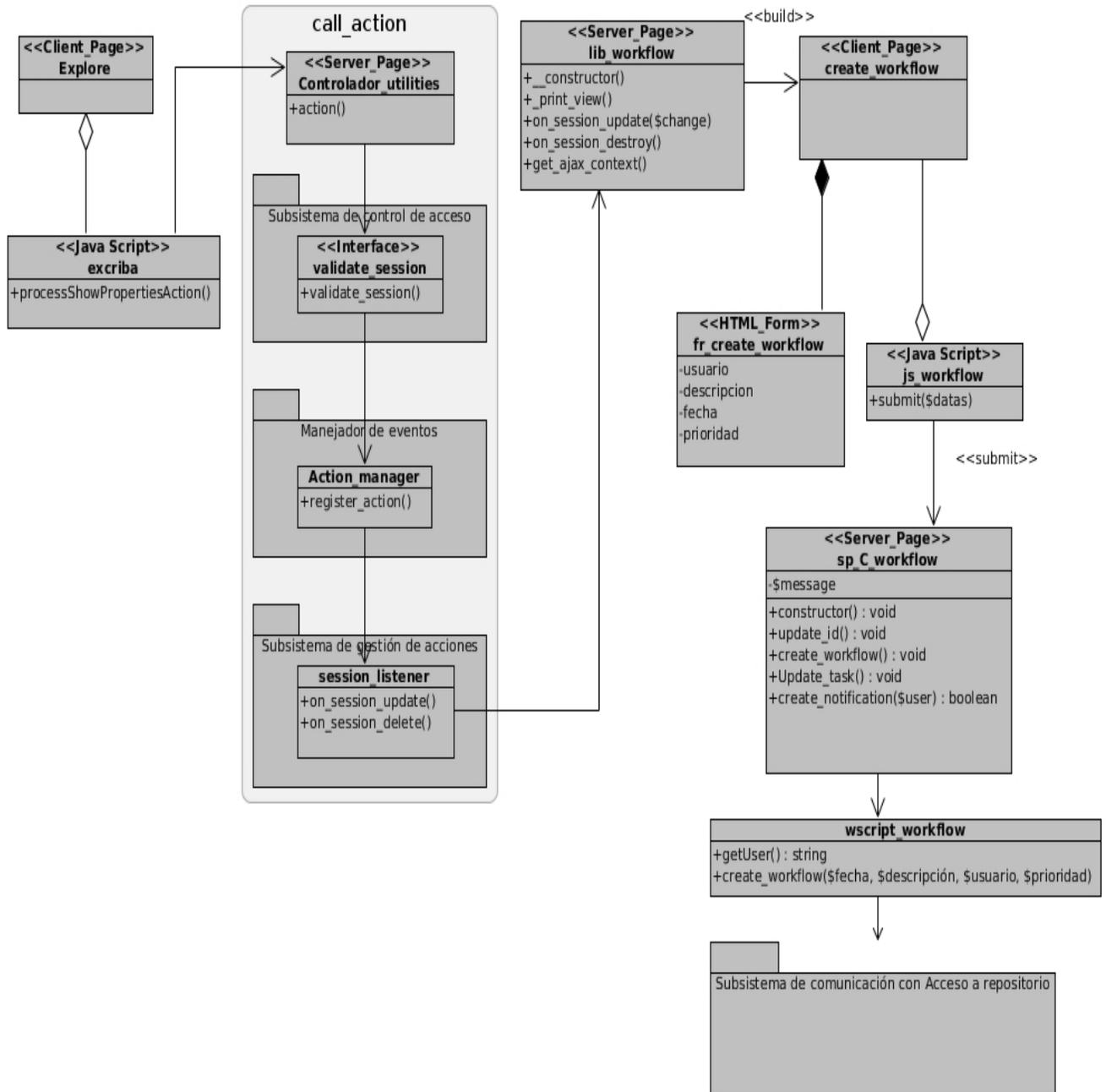


Figura 3.1: Diagrama de clases del diseño crear flujos de trabajo.

### 3.1.2. Descripción de las clases

#### Descripción del Diagrama de Clases del Diseño del CU Gestionar flujos de trabajo avanzados

<b>Nombre:</b> C_workflow	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
\$message	
<b>Para cada responsabilidad:</b>	
Nombre:	update_id ()
Descripción:	Actualiza el ID después de haber realizado la acción Regresar
Nombre:	create_workflow ()
Descripción:	Crear los flujos de trabajo
Nombre:	update_task ()
Descripción:	Modifica los flujos de trabajo
Nombre:	reassignar_workflow ()
Descripción:	Reasigna los flujos de trabajo
Nombre:	create_notification (\$user)
Descripción:	Manda la notificación al usuario luego de haber creado el flujo de trabajo

<b>Nombre:</b> workflow	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
\$descripción	
\$fecha	
\$prioridad	
\$usuario	

**Para cada responsabilidad:**

<b>Nombre:</b> workflow	
<b>Tipo de clase:</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	_on_session_destroy ()
Descripción:	Es el encargado de notificar cuando se destruye algo en la sección
Nombre:	_on_session_delete ()
Descripción:	Es el encargado de notificar cuando se borra algo en la sección
Nombre:	print_view ()
Descripción:	Se declaran los botones y campos que va a tener la vista

<b>Nombre:</b> reasignar	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
\$descripción	
\$fecha	
\$prioridad	
\$usuario	
<b>Para cada responsabilidad:</b>	

<b>Nombre:</b> basic_operation
<b>Tipo de clase:</b> Controladora

Atributo	Tipo
<b>Para cada responsabilidad:</b>	
Nombre:	<code>_on_session_destroy ()</code>
Descripción:	Es el encargado de notificar cuando se destruye algo en la sección
Nombre:	<code>_on_session_delete ()</code>
Descripción:	Es el encargado de notificar cuando se borra algo en la sección
Nombre:	<code>print_view ()</code>
Descripción:	Se declaran los botones y campos que va a tener la vista
Nombre:	<code>get_ajax_context ()</code>
Descripción:	Es el encargado de incluir los ficheros JavaScript

### 3.1.2.1. Descripción de los servicios

Para el desarrollo de las funcionalidades del módulo es necesario implementar algunos servicios en Alfresco, debido a que la API de JavaScript que provee el mismo no expone los servicios necesarios para la gestión de flujos de trabajo desde aplicaciones externas. Estos serán imprescindibles ya que a través de los mismos es que se podrá acceder, modificar y obtener los datos que se encuentran almacenados en el repositorio de contenido.

Servicio: Crear flujos de trabajo avanzado		
Paquete: /cu/uci/excriba/workflow	Plantillas de Respuestas: JSON, XML, HTML	
Descripción: Crear los flujos de trabajo avanzado		
Requerimiento de autenticación: Usuario	Requerimiento de Transacción: Requerida	
Respuesta por defecto:		
URI Absoluta: http://<nombre_servidor>[:<puerto>] /alfresco/service/cu/uci/workflow- instance-create? description={description}&assign={ assign}&date={date?}	Método HTTP: GET	Ruta Relativa: alfresco/service/cu/uci/workflo w-instance-create? description={description}&assi gn={assign}&date={date?}
Dirección del documento de descripción: classpath:alfresco/extension/templates/webscripts/cu/uci/excriba/workflow/create-workflow- instance.get.desc.xml		
Lenguaje de implementación: JavaScript		

Figura 3.2: Descripción del servicio: crear flujos de trabajo avanzados.

Servicio: Listar los flujos de trabajo pendiente		
Paquete: /cu/uci/excriba/workflow	Plantillas de Respuestas: JSON, XML, HTML	
Descripción: Listar todos los flujos de trabajo que tiene el usuario pendiente		
Requerimiento de autenticación: Usuario	Requerimiento de Transacción: Requerida	
Respuesta por defecto:		
URI Absoluta: http://<nombre_servidor>[:<puerto>]/alfresco/service/cu/uci/listar_workflow	Método HTTP: GET	Ruta Relativa: alfresco/service/cu/uci/listar_workflow
Dirección del documento de descripción: classpath:alfresco/extension/templates/webscripts/cu/uci/excriba/workflow/listar_workflow.get.desc.xml		
Lenguaje de implementación: JavaScript		

Figura 3.3: Descripción del servicio: listar flujos de trabajo pendientes.

Servicio: Modificar el estado de los flujos de trabajo		
Paquete: /cu/uci/excriba/workflow	Plantillas de Respuestas: JSON, XML, HTML	
Descripción: Modificar el estado de los flujos de trabajo		
Requerimiento de autenticación: Usuario	Requerimiento de Transacción: Requerida	
Respuesta por defecto:		
URI Absoluta: http://<nombre_servidor>[:<puerto>]/alfresco/service/cu/uci/workflow/properties?id={workflow_id}&status={new_status}	Método HTTP: GET	Ruta Relativa: alfresco/service/cu/uci/workflow/properties?id={workflow_id}&status={new_status}
Dirección del documento de descripción: classpath:alfresco/extension/templates/webscripts/cu/uci/excriba/workflow/properties.get.desc.xml		
Lenguaje de implementación: Java		
Dirección del paquete: /cu/uci/excriba/webscripts		

Figura 3.4: Descripción del servicio: modificar estado del flujo de trabajo avanzado.

Servicio: Mostrar los usuarios del alfresco		
Paquete: /cu/uci/excriba/workflow	Plantillas de Respuestas: JSON, XML, HTML	
Descripción: Muestra todos los usuarios que están en el Alfresco		
Requerimiento de autenticación: Usuario	Requerimiento de Transacción: Requerida	
Respuesta por defecto:		
URI Absoluta: http://<nombre_servidor>[:<puerto>]/ alfresco/service/cu/uci/workflow_get User	Método HTTP: GET	Ruta Relativa: alfresco/service/cu/uci/workflow_getUser
Dirección del documento de descripción: classpath:alfresco/extension/templates/webscripts/cu/uci/excriba/workflow/getUser.get.desc.xml		
Lenguaje de implementación: JavaScript		

Figura 3.5: Descripción del servicio: mostrar usuarios.

### 3.2. Diagrama de secuencia del diseño

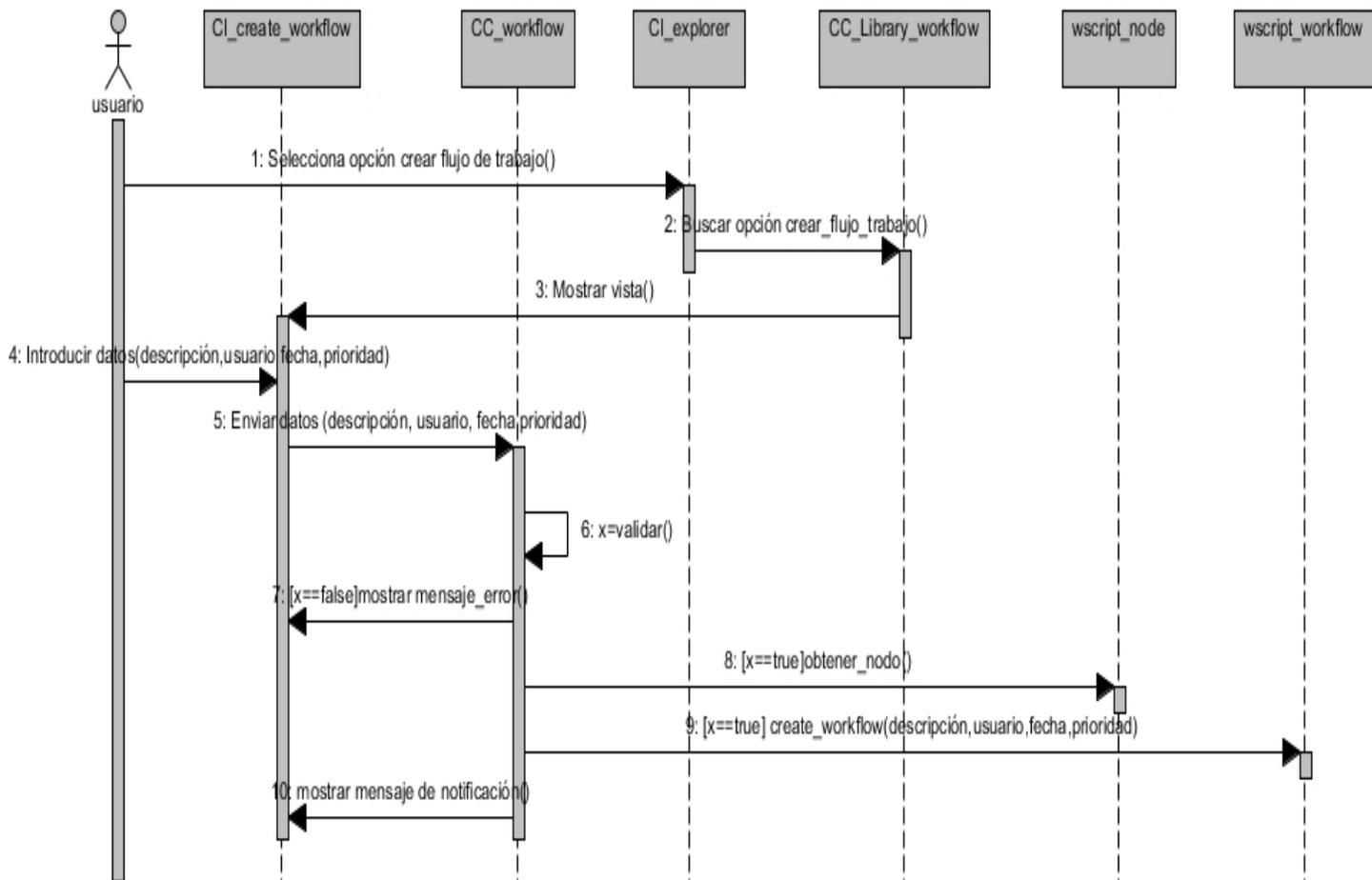


Figura 3.6: Diagrama de secuencia del caso de uso gestionar flujos de trabajo avanzados.

Para ver los diagramas de los restantes casos de uso ver Anexo C de la versión extendida de este documento.

### 3.3. Descripción de la arquitectura

Una arquitectura de *software* consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del *software* para un sistema de información [31].

#### 3.3.1. Arquitectura en capas

Para el desarrollo de la solución se utilizará la arquitectura en capas, debido a que el sistema al cual se le incorporará el módulo ya utiliza la misma, constituyendo esto una restricción del diseño. Esta arquitectura del *software* va a contar con tres capas, estas son: Presentación, Aplicación y Acceso a repositorio.

- **Capa de presentación:** En esta capa es donde estarán el conjunto de interfaces de usuario, que le hará posible establecer la comunicación entre el cliente y la aplicación, así como la manipulación de los datos y representar en términos de componentes visuales, toda la información necesaria, consultada y/o generada por el usuario y la aplicación. Todas las vistas que se diseñarán para el módulo gestión de flujos de trabajo avanzados serán implementadas en la capa de presentación, apoyándonos del Framework jQuery, usando además la API de eXcriba para jQuery la cual brinda un conjunto de funcionalidades que facilitan la implementación.
- **Capa de aplicación:** En esta capa se ejecutan todos los procesos de negocio que han sido previamente implementados, se preparan a su vez las transformaciones de datos, sirviendo como un mediador entre las demandas del cliente y las respuestas de los datos. Controla y dirige el flujo de la aplicación en sentido general. Es además la encargada de comunicarse con los servicios ubicados en la capa inferior. El módulo para la gestión de flujos de trabajo avanzados necesita acceder a un conjunto de subsistemas en esta capa, entre los que se encuentran: Gestión de acciones, Control de acceso, Comunicación con la capa de acceso a repositorio y Gestión de eventos. Estos, son los que validan los permisos que tiene el usuario, las acciones y eventos que pueden ejecutar y los que permiten la comunicación con los servicios

que brinda Alfresco. Además, en esta capa se tendrán las clases controladoras que se encargarán de manejar la lógica del negocio, implementadas a través del Framework CodeIgniter.

- **Capa de acceso a repositorio:** Esta capa es la encargada de gestionar los datos que se encuentran en el repositorio a través de la implementación de los servicios. Los servicios que se implementarán en esta capa estarán relacionados con la API de WebScript de Alfresco.

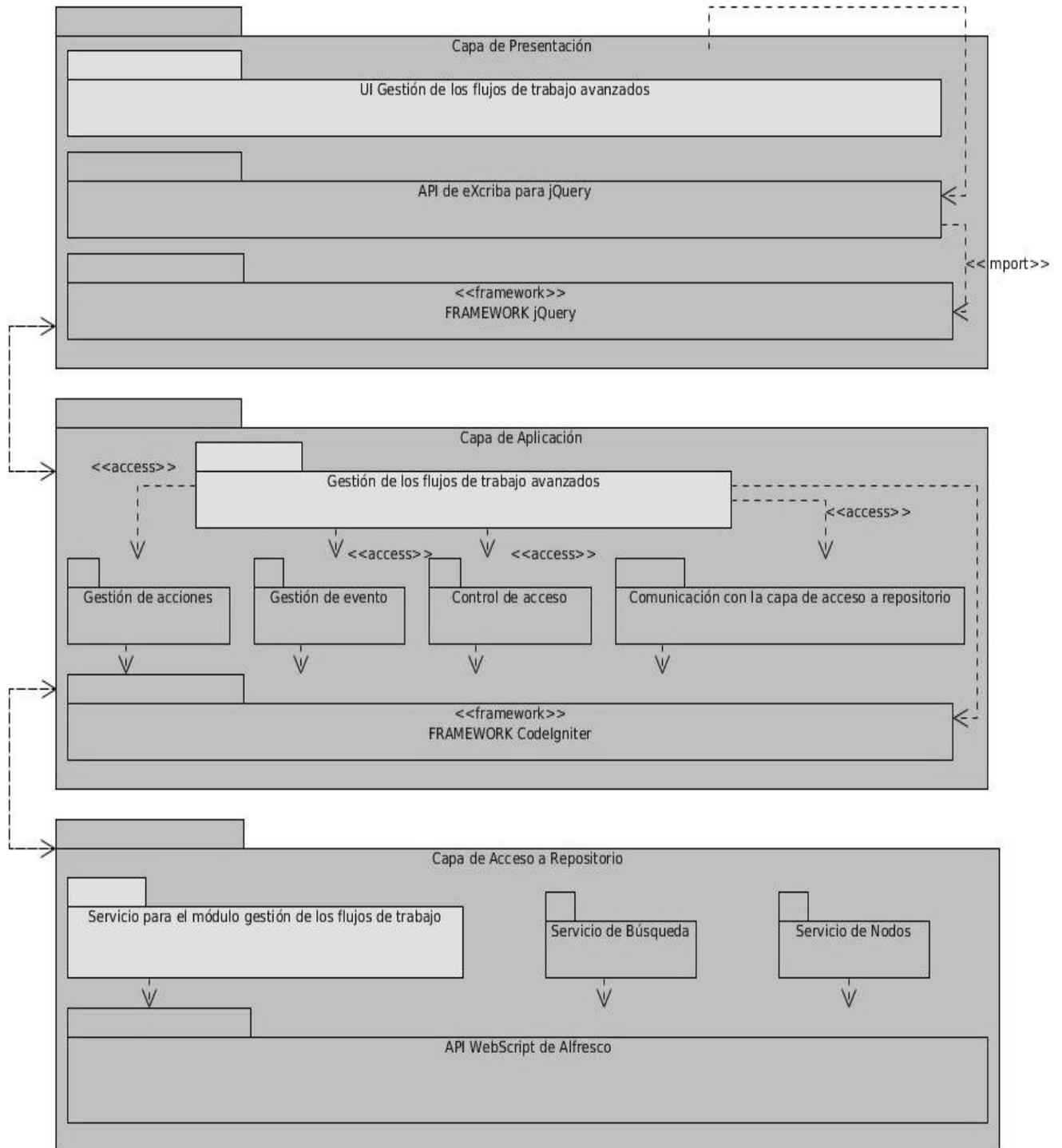


Figura 3.7: Descripción de la arquitectura.

### 3.4. Patrones de diseño

*“Un patrón es una descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos; en teoría, indican la manera de utilizarlo en circunstancias diversas” [32].*

El patrón que se utilizó fue el GRASP (*General Responsibility Assignment Software Patterns*) los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Es importante señalar que muchos de estos patrones están muy relacionados entre sí, ya que por ejemplo, el grado de acoplamiento no puede considerarse aisladamente en otros principios como Experto y Alta Cohesión [32].

- **Experto:** Este patrón es el encargado de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad [32].

Ejemplo: La aplicación de este patrón se evidencia en la clase `application_helper` la cual tiene la información necesaria para hacer la llamada al servicio, acción que también pudiera ser realizada por la clase controladora, pero esta clase solo posee los datos que son necesarios para llamar al método `call_wsbscript` el cual es el que realmente hace la llamada al servicio.

Beneficios: Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide.

- **Bajo Acoplamiento:** La función de este patrón es asignar una responsabilidad para mantener bajo acoplamiento. Una clase con bajo acoplamiento no depende de muchas otras [32].

Ejemplo: Este se evidencia en la creación de una biblioteca para cada vista que se desarrolle, cada biblioteca tendrá las mismas funciones pero estos métodos tendrán diferentes implementaciones. Lo que evidencia la poca dependencia entre las clases.

Beneficios: Las clases no se afectan por cambios de otros componentes, además son fáciles de entender por separado y fáciles de reutilizar.

- **Alta cohesión:** Es el encargado de asignar una responsabilidad, de modo que la cohesión siga siendo alta. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme [32].

Ejemplo: Por la estrecha relación que existe entre este patrón y el de Bajo Acoplamiento se toma como ejemplo el mismo planteado para el anterior patrón.

Beneficios: Mejoran la claridad y la facilidad con que se entiende el diseño. La ventaja de una gran funcionalidad soporta una mayor capacidad de reutilización, porque una clase muy cohesiva puede destinarse a un propósito muy específico.

- **Controlador:** Asignar la responsabilidad de las operaciones del sistema a los objetos situados en la capa del dominio y no en los soportes de la capa de presentación [32].

Beneficios: Mayor potencial de los componentes reutilizables, ya que garantiza que los procesos de dominio sean manejados por la capa de Aplicación y no por la de interfaz, además de tener un mayor control.

# Capítulo 4

## Implementación y Prueba del módulo

### 4.1. Diagrama de despliegue

El diagrama de despliegue es usado para visualizar la distribución de los componentes de *software* en los nodos físicos.

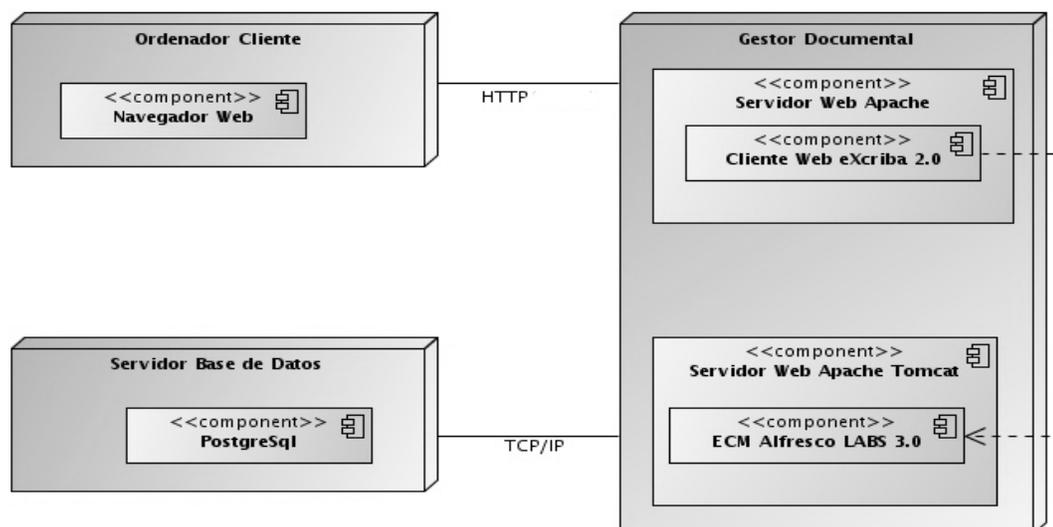


Figura 4.1: Diagrama de despliegue.

El despliegue del *software* eXcriba cuenta con tres nodos. El primero lleva por nombre Ordenador Cliente, en el cual se encontrarán las estaciones de trabajo que el usuario utilizará para acceder a la aplicación. En el ordenador cliente debe residir un navegador *web*. Como segundo nodo se tiene Gestor Documental, este se refiere al núcleo del sistema, donde reside la lógica de aplicación para lograr la conexión del sistema con el Ordenador Cliente se utiliza HTTP como protocolo de comunicación. Dentro del nodo del gestor documental residen dos servidores *web*: Apache que contiene el cliente *web* de eXcriba 2.0 y Apache Tomcat

que contiene el ECM Alfresco Labs 3.0 Stable. Por último está el Servidor de Bases de Datos, este nodo es donde se encuentra el servidor que radica en cada nodo regional donde se van a estar centralizados los datos recopilados. El servidor de base de datos está especificado por el gestor postgresql.

## 4.2. Diagrama de componentes

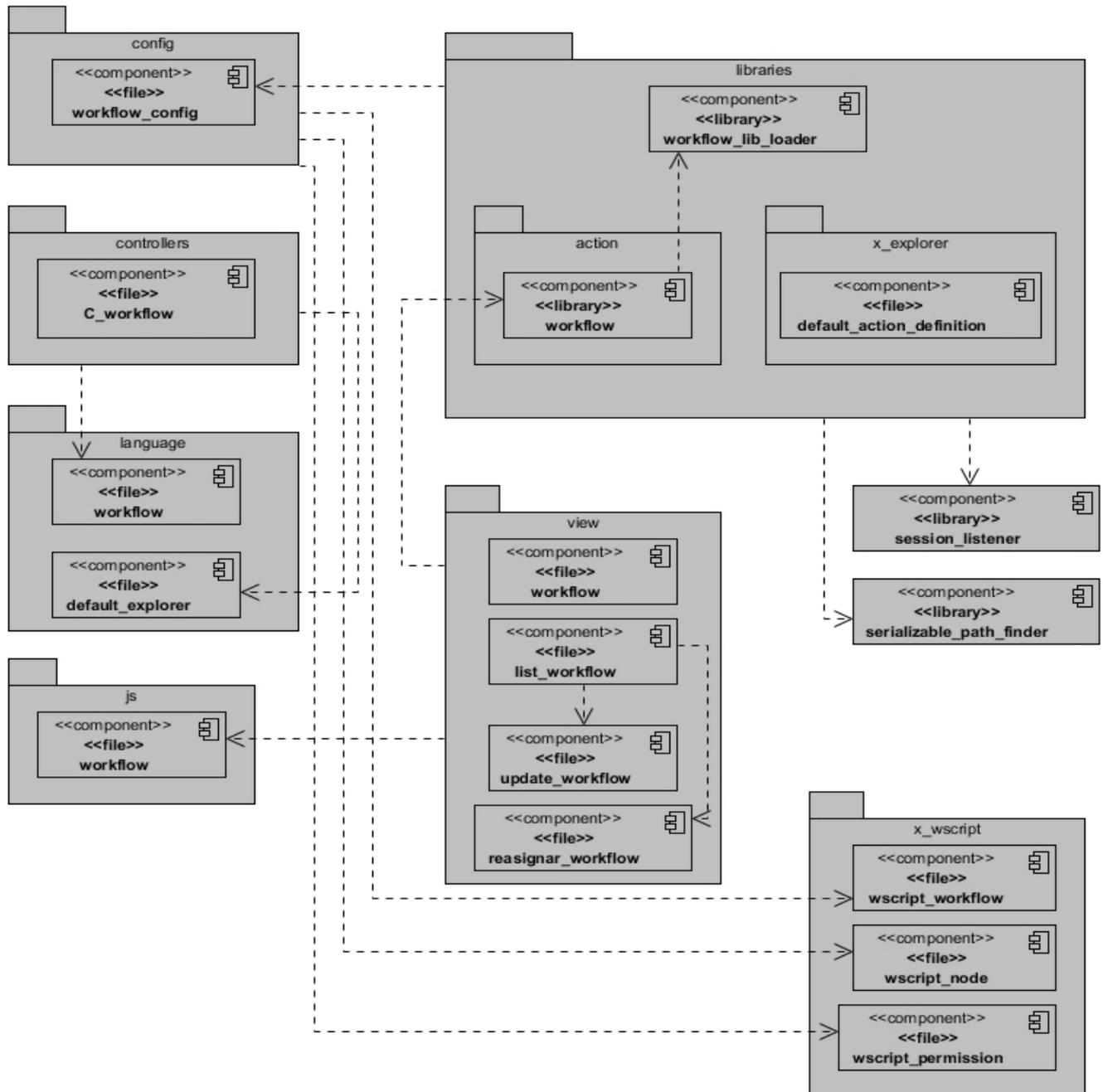


Figura 4.2: Diagrama de componente del módulo gestión de flujos de trabajo avanzados.

## 4.3. Pruebas de *software*

Las pruebas del *software* son un elemento fundamental para la garantía de calidad del sistema. Estas pruebas representan una revisión final de las especificaciones del diseño y de la codificación, es decir, las pruebas verifican que el *software* funcione como se diseñó y que los requerimientos son satisfechos, además de brindar soporte para encontrar y documentar defectos del sistema [28].

### 4.3.1. Prueba de caja blanca

“La prueba de caja blanca es un método de diseño de casos de prueba que usa la estructura de control de diseño procedimental para obtener los casos de prueba” [28]. Mediante los métodos de caja blanca se garantiza que se ejerciten al menos una vez todos los caminos independientes de cada módulo y todas las decisiones lógicas en sus vertientes verdaderas y falsa. Además de que se ejecuten todos los bucles en sus límites y se ejerciten las estructuras internas de datos para asegurar su validez [28].

A continuación se especifican algunos tipos de prueba de caja blanca:

- Prueba de condición: es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa.
- Prueba de flujo de datos: se seleccionan caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.
- Prueba de bucles: comprueba la validez de las construcciones de los bucles.
- Prueba del camino básico: permite obtener una medida de la complejidad lógica de un diseño.

Para comprobar la precisión del código se seleccionó la prueba de camino básico. Esta permitirá permitió comprobar la complejidad ciclomática de los métodos del módulo y verificar que todos los caminos se ejecutaban de forma correcta. A continuación se muestran los pasos realizados para validar el correcto funcionamiento del método `create_workflow`.

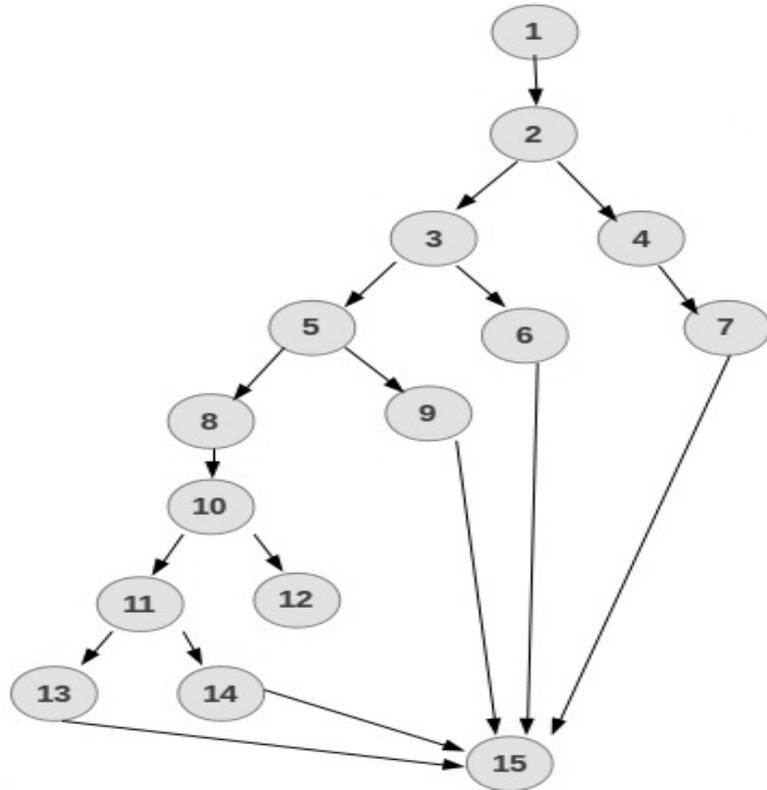


Figura 4.3: Diagrama del flujo asociado al algoritmo create\_workflow.

Fórmula para calcular la complejidad ciclomática:

$$V(G) = (A - N) + 2$$

$$V(G) = (19 - 15) + 2$$

$$V(G) = 6$$

Siendo “A” la cantidad total de aristas y “N” la cantidad total de nodos.

Otra forma de calcular la complejidad ciclomática:

$$V(G) = P + 1$$

$$V(G) = 5 + 1$$

$$V(G) = 6$$

Siendo “P” la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

Se puede usar además:

$$V(G) = R$$

Siendo “R” la cantidad total de regiones incluyendo la del exterior, para cada fórmula “V(G)” representa el valor del cálculo. El resultado obtenido mediante las fórmulas anteriores representan los posibles caminos por los que transitará el flujo, así como la cantidad mínima de casos de pruebas que se deben realizar para el procedimiento escogido. A continuación se representa los caminos básicos que se identificaron en el flujo:

Camino 1	1-2-3-5-8-10-11-13-15
Camino 2	1-2-4-7-15
Camino 3	1-2-3-6-15
Camino 4	1-2-3-5-9-15
Camino 5	1-2-3-5-8-10-12-15
Camino 6	1-2-3-5-8-10-11-14-15

Figura 4.4: Tabla de caminos.

Se procede a realizar los casos de pruebas, teniendo en cuenta que se debe realizar al menos un caso de prueba por cada camino básico identificado:

**Casos de prueba para el camino básico 1**

Camino 1: [ 1-2-3-5-8-10-11-13-15 ]

Descripción: los datos entrantes permiten la creación del flujo de trabajo.

Entrada: Descripción= ” Prueba ”, Fecha= ” 08/05/2012 ” , Usuario= ” admin ” y Prioridad = “ 1 ”.

Resultados esperados: se muestra el mensaje “ El flujo de trabajo fue creado correctamente ” .

**Casos de prueba para el camino básico 2**

Camino 2: [ 1-2-4-7-15 ]

Descripción: los datos entrantes permiten la creación del flujo de trabajo.

Entrada: Descripción= ” ”, Fecha= ” 08/05/2012 ” , Usuario= ” admin ” y Prioridad = “ 1 ”.

Resultados esperados: se muestra el mensaje “ El campo descripción no puede estar vacío ”.

**Casos de prueba para el camino básico 3**

Camino 3: [ 1-2-3-6-15 ]

Descripción: los datos entrantes permiten la creación del flujo de trabajo.

Entrada: Descripción= " Prueba ", Fecha= " " , Usuario= " admin " y Prioridad = " 1 ".

Resultados esperados: se muestra el mensaje " El campo fecha no puede estar vacío ".

#### **Casos de prueba para el camino básico 4**

Camino 4: [ 1-2-3-5-9-15 ]

Descripción: los datos entrantes permiten la creación del flujo de trabajo.

Entrada: Descripción= " Prueba ", Fecha= " 08/05/2012 " , Usuario= " admin " y Prioridad = " 1 ".

Resultados esperados: se muestra el mensaje " La fecha puesta es menor que la de la actual ".

#### **Casos de prueba para el camino básico 5**

Camino 5: [ 1-2-3-5-8-10-12-15 ]

Descripción: los datos entrantes permiten la creación del flujo de trabajo.

Entrada: Descripción= " Prueba ", Fecha= " 08/05/2012 " , Usuario= " admin " y Prioridad = " 1 ".

Resultados esperados: se muestra el mensaje " No se pudo crear el flujo de trabajo no se encuentra el webscript ".

#### **Casos de prueba para el camino básico 6**

Camino 6: [ 1-2-3-5-8-10-11-14-15 ]

Descripción: los datos entrantes permiten la creación del flujo de trabajo.

Entrada: Descripción= " Prueba ", Fecha= " 08/05/2012 " , Usuario= " admin " y Prioridad = " 1 ".

Resultados esperados: se muestra el mensaje " No se pudo crear la notificación ".

### **4.3.2. Pruebas de caja negra**

La prueba de caja negra se centra principalmente en los requisitos funcionales del *software*. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos. Intentando encontrar errores de las siguientes categorías: funciones

incorrectas o ausentes; errores de interfaz, en estructuras de datos o en acceso a bases de datos externas; errores de rendimiento, de inicialización y de terminación [28].

Para comprobar la calidad de la solución planteada, se empleó el método de caja negra aplicándose la técnica de partición equivalente, pues permite examinar los valores válidos e inválidos de las entradas existentes en el *software*, además se descubre de forma inmediata los errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico, y con la utilización de esta técnica nos permitiría reducir el número de clases de prueba que hay que desarrollar [28].

A continuación se presentan y describen los casos de prueba desarrollados para cada caso de uso definido, especificando la información de entrada, los resultados obtenidos una vez ejecutado el caso de prueba y las condiciones que deben cumplirse para que este se ejecute.

#### **4.3.2.1. Casos de prueba de caja negra**

Caso de uso Gestionar flujos de trabajo avanzados.

Descripción de la funcionalidad: el caso de uso comienza cuando el usuario desea gestionar un flujo de trabajo avanzado. Para ello puede crear, modificar o listar los mismos. El caso de uso termina cuando realiza alguna de estas acciones.

Condiciones de ejecución: el usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder crear un flujo de trabajo avanzado. Para las secciones de modificar y listar debe existir al menos un flujo de trabajo avanzado creado.

Entrada	Resultado	Condiciones
El usuario crea un flujo de trabajo llenando los campos: Descripción: "Prueba" Fecha límite: "05/05/2012" Prioridad: "1" Usuario: "rallen"	Se almacena en el repositorio el flujo de trabajo y se muestra el mensaje "Flujo de trabajo fue creado correctamente ".	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder crear un flujo de trabajo avanzado.
El usuario crea un flujo de trabajo llenando los campos: Descripción: " " Fecha límite: "05/05/2012" Prioridad: "1" Usuario: "rallen"	Se muestra el mensaje "El campo descripción no puede estar vacío "	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder crear un flujo de trabajo avanzado.
El usuario crea un flujo de trabajo llenando los campos: Descripción: "Prueba" Fecha límite: "" Prioridad: "1" Usuario: "rallen"	Se muestra el mensaje "El campo Fecha no puede estar vacío "	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder crear un flujo de trabajo avanzado.
El usuario crea un flujo de trabajo llenando los campos: Descripción: "Prueba" Fecha límite: "01/01/2011" Prioridad: "1" Usuario: "rallen"	Se muestra el mensaje "La fecha límite no puede ser menor que la actual"	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder crear un flujo de trabajo avanzado.
Limpiar formulario de creación de un flujo de trabajo avanzado.	Se eliminan todos los datos de los datos insertados por el usuario.	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder crear un flujo de trabajo avanzado.

Figura 4.5: Escenario crear flujo de trabajo avanzado.

Entrada	Resultado	Condiciones
Modificar flujo de trabajo avanzado correctamente	Se guardan los cambios en el repositorio.	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder modificar un flujo de trabajo avanzado.
No hay flujos de trabajo	Muestra el mensaje "No tiene flujos de trabajo avanzados pendiente"	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder modificar un flujo de trabajo avanzado.
Cancela la acción.	No se guardan los cambios efectuados y se muestra la lista de flujos de trabajo pendientes	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder modificar un flujo de trabajo avanzado.

Figura 4.6: Escenario modificar flujo de trabajo avanzado.

Entrada	Resultado	Condiciones
Listar flujos de trabajo avanzado correctamente	Muestra en una tabla los siguientes datos de los flujos de trabajo avanzado que tiene pendiente el usuario: Id, Estado, Prioridad, Descripción, Documento, Fecha y Acciones	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder listar un flujo de trabajo avanzado.
No hay flujos de trabajo	Muestra el mensaje "No tiene flujos de trabajo avanzados pendiente"	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder listar un flujo de trabajo avanzado.

Figura 4.7: Escenario listar flujos de trabajo avanzados.

Caso de uso Reasignar flujos de trabajo avanzados.

Descripción de la funcionalidad: el caso de uso comienza cuando el usuario desea reasignar un flujo de trabajo avanzado que tiene pendiente, terminando el caso de uso una vez realizada la acción.

Condiciones de ejecución: el usuario tiene que estar autenticado en el sistema y debe existir al menos un flujo de trabajo avanzado asignado a él.

Entrada	Resultado	Condiciones
Reasignar flujo de trabajo avanzado correctamente	Se almacenan los cambios en el repositorio y se muestra el mensaje "El flujo de trabajo ha sido reasignado correctamente".	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder reasignar un flujo de trabajo avanzado.
Reasignar flujo de trabajo avanzado dejando campos obligatorios vacios	Se muestra el mensaje "El campo descripción no puede estar vacío "	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder reasignar un flujo de trabajo avanzado.
El usuario reasigna un flujo de trabajo llenando los campos: Descripción: "Prueba" Fecha límite: "05/05/2012" Prioridad: "1" Usuario: "rallen"	Se almacena en el repositorio el flujo de trabajo y se muestra el mensaje "Flujo de trabajo fue creado correctamente".	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder reasignar un flujo de trabajo avanzado.
El usuario reasigna un flujo de trabajo llenando los campos: Descripción: "" Fecha límite: "05/05/2012" Prioridad: "1" Usuario: "rallen"	Se muestra el mensaje "El campo descripción no puede estar vacío "	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder reasignar un flujo de trabajo avanzado.
El usuario reasigna un flujo de trabajo llenando los campos: Descripción: "Prueba" Fecha límite: "" Prioridad: "1" Usuario: "rallen"	Se muestra el mensaje "El campo Fecha no puede estar vacío "	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder reasignar un flujo de trabajo avanzado.
El usuario reasigna un flujo de trabajo llenando los campos: Descripción: "Prueba" Fecha límite: "01/01/2011" Prioridad: "1" Usuario: "rallen"	Se muestra el mensaje "La fecha límite no puede ser menor que la actual"	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder reasignar un flujo de trabajo avanzado.
Limpiar formulario de creación de un flujo de trabajo avanzado.	Se eliminan todos los datos de los datos insertados por el usuario.	El usuario tiene que estar autenticado en el sistema. Debe existir al menos un documento registrado en el sistema para poder reasignar un flujo de trabajo avanzado.

Figura 4.8: Escenario reasignar flujo de trabajo.

Una vez que se realizan la prueba de caja negra, se obtienen los resultados siguientes:

No conformidades	Iteración 1	Iteración 2	Iteración 3
Alta	1	.	.
Baja	5	3	.

Figura 4.9: Tabla de resultados.

En el flujo de prueba se realizaron las descripciones de los casos de prueba pertenecientes a los casos de uso documentados en el trabajo, para comprobar las funcionalidades del módulo; las pruebas realizadas se llevaron a cabo en tres iteraciones, en la primera iteración se detectaron seis no conformidades de las cuales cinco de ellas fueron bajas (no significativas) y una alta (significativa); en la segunda iteración se detectaron tres no conformidad todas ellas baja(no significativas) que fueron resueltas inmediatamente, en la última iteración ya no se encontraron no conformidades, por lo tanto se puede concluir que las pruebas de funcionalidad en el módulo fueron realizadas satisfactoriamente.

# Conclusiones

Durante el desarrollo de esta investigación se expuso la necesidad de implementar un módulo para gestionar los flujos de trabajo avanzado en el cliente *web* del eXcriba, arribando a las siguientes conclusiones:

- Los flujos de trabajo constituyen un aspecto importante dentro de la gestión documental, ya que permiten el ahorro de tiempo en la creación de procesos de negocio, mejorando la productividad debido a la automatización de los mismos.
- El diseño del módulo gestión de flujos de trabajo avanzado permitió establecer un punto de partida para la actividades de implementación y facilitó la descomposición en partes de las funcionalidades a desarrollar.
- La implementación del módulo permitió automatizar los procesos de negocio en el cliente *web* del eXcriba.
- La utilización de la técnica partición equivalente y la prueba de camino básico permitieron validar el correcto funcionamiento del módulo.

# Recomendaciones

Adicionar al módulo de gestión de flujos de trabajo avanzado una funcionalidad que permita notificar a los usuarios sobre los flujos de trabajo pendientes que tienen cerca su fecha límite.

## Referencias bibliográficas

- [1] OTLET, Paul. El Tratado de documentación. El libro sobre el libro [en línea]. Bruselas: Ediciones Mundeandum, 1934, enero 2008. [citado noviembre 2011]. Disponible en Internet: <http://books.google.es>.ISSN: 978-84-8371-703-5.
- [2] MARTELL, Javier. *El Universo Digital en Expansión. En: El Estándar Social* [en línea], julio 2007. [citado enero 2012]. Disponible en Internet: <http://www.elestandarsocial.com/2007/07/el-universo-digital-en-expansion.html>.
- [3] FONSECA, Misael, *et al.* eXcriba, gestor de documentos administrativos. En II Taller de Sistemas de Gestión de la Información y el Conocimiento. Memorias de la VI Conferencia Universidad de las Ciencias Informáticas, UCIENCIA 2012. Universidad de las Ciencias Informáticas, 20-22 febrero 2012. ISBN: 987-959-286-019-3.
- [4] FERNÁNDEZ, Luis David. *Gestión Documental. En Sociedad de la Información* [en línea], septiembre 2007, no. 12. [citado marzo 2012]. Disponible en: <http://www.sociedadelainformacion.com/12/Gestion%20Documental.pdf>.
- [5] WALNE, P. Dictionary of archival terminology. English and French, with equivalents in Dutch, German, Italian, Russian and Spanish. *Dictionnaire de terminologie archivistique*. München: Saur, 1988. [citado Enero 2012].

- [6] CAMPILLO\_TORRES, Irima. *Sistema de Gestión Integral de Documentos de archivo para empresas de la construcción del territorio de Camagüey*. Tesis doctoral, Universidad de Granada, Octubre 2010. ISBN: 978-84-694-0954-1. .
- [7] BUSTELO\_RUESTA, Carlota. *Gestión Documental en las Empresas: Una aproximación práctica*. En VII Jornadas Españolas de Documentación, FESABID 2000. Bilbao, 19-21 octubre 2000.
- [8] MENA, M. *Gestión documental y organización de archivos*. La Habana: Félix Varela, 2005.
- [9] GARIMELLA, Kiran, LEES, Michael, and WILLIAMS, Bruce. *Introducción a BPM para Dummies*. Edición especial de Software AG. Indiana: Wiley Publishing, Inc. ISBN: 978-0-470-37359-0 .
- [10] ¿Qué es workflow y control del flujo de trabajo?[en línea]. Barcelona, 2011, [citado enero 2012]. Disponible en: <http://www.pixelware.com/workflow-flujo-trabajo.htm> .
- [11] LAURENTIIS\_GIANNI, Renato. *WorkFlow-Tecnología para la Integración y Orquestación de Procesos, Sistemas y Organización*. IBERICA IT Group, Madrid-España. Disponible en: <ftp://10.0.0.22/documentacion/Computer%20Science/Workflow/books/>.
- [12] DE\_ÁVILA, Sancho. *Libro Blanco Gestión Documental Open Source* [en línea]. Barcelona. [citado enero 2012]. Disponible en:<http://www.smile-iberia.com>.
- [13] Portal de Matanzas, AvilaDoc [en línea]. Cuba, 2011, [citado enero 2012]. Disponible en: <http://www.expomatanzas.cu/empresa.php?emp=151&prd=253> .
- [14] CARUANA, David, *et al.* *Alfresco Practical Solutions for Enterprise Content Management*.Indianapolis Indiana: Ediciones Wiley Publishing Inc., 2010. 575p. ISBN: 978-0-470-57104-0.
- [15] Alfresco, *Sobre Alfresco*[en línea]. Estados Unidos, 2012, [citado marzo 2012]. Disponible en: <http://www.alfresco.com/es/about/>.
- [16] SHARIFF, Munwar. *Alfresco Enterprise Content Management Implementatio*. Birmingham Mumbai: Editor Mike W. Walker, 2006. 353 p. ISBN: 1-904811-11-6.

- [17] Jacobson, Ivar, *et al.* *El Proceso Unificado de Desarrollo de Software*. La Habana: Ediciones Félix Varela, 2004. ISBN: 84-7829-036-2.
- [18] SAETHER, S, *et al.* Manual de php. Editores: Rafael Martínez, Víctor Fernández, Leonardo Boshell. [en línea]. noviembre 2002 Disponible en: <http://www.calitae.com/manuales/manual-php.pdf> [citado Marzo 2012] .
- [19] ISSI\_CAMY, Lázaro. *JavaScript*. Madrid: Ediciones Anaya Multimedia, 2002. 1005 p. ISBN: 84-415-1384-8.
- [20] Visual Paradigm - EcuRed. Disponible en: [http://www.ecured.cu/index.php/Visual\\_Paradigm](http://www.ecured.cu/index.php/Visual_Paradigm). [citado Febrero 2012]. .
- [21] FIELDING, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures*. Tesis doctoral, University of California, Irvine, 2000.
- [22] RESTful Web services: The basics [en línea]. noviembre 2008. Disponible en: <http://www.ibm.com/developerworks/webservices/library/ws-restful/> [citado Mayo 2012].
- [23] CodeIgniter[en línea].2012,[citado abril 2012]. Disponible en: <http://codeigniter.com/>.
- [24] The jQuery Foundation, JavaScript library. [en línea]. 2012, [citado febrero 2012]. Disponible en: <http://jquery.com/>.
- [25] Web Scripts. [en línea]. febrero 2011. Disponible en: [http://wiki.alfresco.com/wiki/Web\\_Scripts](http://wiki.alfresco.com/wiki/Web_Scripts) .
- [26] FreeMarker: java template engine library - overview. [en línea]. Disponible en: <http://freemarker.org/>. [citado Abril 2012].
- [27] IEEE: Guide for developing system requirements specification., 1998. [citado Abril 2012].
- [28] PRESSMAN, Roger S. *Ingeniería del Software Un enfoque práctico*. La Habana: Ediciones Félix Varela, 2005. 601 p. ISBN: 970-105-473-3.

- [29] Laboratorio Nacional de Calidad del Software de INTECO, *Guía de Desarrollo y Gestión de Requisitos de la Adquisición*. España: Instituto Nacional de las Tecnologías de la Comunicación (INTECO), noviembre 2009. Disponible en: [http://www.inteco.es/calidad\\_TIC/descargas/guias/guia\\_requisitos\\_adquisicion](http://www.inteco.es/calidad_TIC/descargas/guias/guia_requisitos_adquisicion).
- [30] SOMMERVILLE, Ian. *Ingeniería de Software*. La Habana: Ediciones Félix Varela, enero 2005. 687 p. ISBN: 84-7829-074-5.
- [31] Arquitectura de software [en línea]. 2011. Disponible en: [http://www.ecured.cu/index.php/Arquitecturas\\_de\\_sof](http://www.ecured.cu/index.php/Arquitecturas_de_sof) [citado: Febrero 2012]. .
- [32] LARMAN, Craig. *UML y Patrones Introducción al análisis y diseño orientado a objetos*. La Habana: Ediciones Félix Varela, 2004. 492 p. ISBN: 842-053-438-2.

# Bibliografía

- OTLET, Paul. *El Tratado de documentación. El libro sobre el libro*. [en línea]. Bruselas: Ediciones Mundeanum, 1934, enero 2008. [citado noviembre 2011]. Disponible en Internet: <http://books.google.es>. ISSN: 978-84-8371-703-5.
- MARTELL, Javier. *El universo digital en expansión. en: El estándar social* [en línea]. julio 2007. [citado enero 2012]. Disponible en Internet: <http://www.elestandarsocial.com/2007/07/el-universo-digital-en-expansion.html>. Julio 2007 [citado: Enero 2012].
- FONSECA, Misael, *et al* . eXcriba, gestor de documentos administrativos. En II Taller de Sistemas de Gestión de la Información y el Conocimiento. Memorias de la VI Conferencia Universidad de las Ciencias Informáticas, UCIENCIA 2012. Universidad de las Ciencias Informáticas, 20-22 febrero 2012. ISBN: 987-959-286-019-3.
- FERNÁNDEZ, Luis David. *Gestión Documental. En Sociedad de la Información* [en línea], septiembre 2007, no. 12. [citado marzo 2012]. Disponible en: <http://www.sociedadelainformacion.com/12/Gestion%20Documental.pdf>
- WALNE, P. Dictionary of archival terminology. English and French, with equivalents in Dutch, German, Italian, Russian and Spanish. *Dictionnaire de terminologie archivistique*. München: Saur, 1988.

- CAMPILLO\_TORRES, Irima. Sistema de Gestión Integral de Documentos de archivo para empresas de la construcción del territorio de Camagüey. Tesis doctoral, Universidad de Granada, Octubre 2010. ISBN: 978-84-694-0954-1.
- BUSTELO\_RUESTA, Carlota. Gestión Documental en las Empresas: Una aproximación práctica. En VII Jornadas Españolas de Documentación, FESABID 2000. Bilbao, 19-21 octubre 2000.
- GARIMELLA, Kiran, LEES, Michael, and WILLIAMS, Bruce. *Introducción a BPM para Dummies*. Edición especial de Software AG. Indiana: Wiley Publishing, Inc. ISBN: 978-0-470-37359-0
- ¿Qué es workflow y control del flujo de trabajo? [en línea]. Barcelona, 2011, [citado enero 2012]. Disponible en: <http://www.pixelware.com/workflow-flujo-trabajo.htm>
- LAURENTIIS\_GIANNI, Renato. Workflow-Tecnología para la Integración y Orquestación de Procesos, Sistemas y Organización. IBERICA IT Group, Madrid-España. Disponible en: <ftp://10.0.0.22/documentacion/Computer%20Science/Workflow/books/>
- DE\_ÁVILA, Sancho. Libro Blanco Gestión Documental Open Source [en línea]. Barcelona. [citado enero 2012]. Disponible en: <http://www.smile-iberia.com>
- Portal de Matanzas, AvilaDoc [en línea]. Cuba, 2011, [citado enero 2012]. Disponible en: <http://www.expomatanzas.cu/empresa.php?emp=151&prd=253>
- CARUANA, David, *et al.* *Alfresco Practical Solutions for Enterprise Content Management*. Indianapolis Indiana: Ediciones Wiley Publishing Inc., 2010. 575p. ISBN: 978-0-470-57104-0
- Alfresco, Sobre Alfresco [en línea]. Estados Unidos, 2012, [citado marzo 2012]. Disponible en: <http://www.alfresco.com/es/about/>
- SHARIFF, Munwar. *Alfresco Enterprise Content Management Implementatio*. Birmingham Mumbai: Editor Mike W. Walker, 2006. 353 p. ISBN: 1-904811-11-6

- FONSECA\_MATA, Misael, *et al* . Sistema de Gestión Integral de Documento y Archivos. En XVI Fórum de Ciencia y Técnica. Universidad de las Ciencias Informáticas, 16 mayo 2012
- Jacobson, Ivar, *et al*. *El Proceso Unificado de Desarrollo de Software*. La Habana: Ediciones Félix Varela, 2004. ISBN: 84-7829-036-2
- The PHP Group, PHP [en línea]. junio 2012, [citado marzo 2012]. Disponible en: <http://www.php.net/>
- PETROVSKY, Michele. *Manual de Dynamic HTML*. España, Madrid: Ediciones McGraw-Hill, 1998. 365 p.
- ISSI\_CAMY, Lázaro. *JavaScript*. Madrid: Ediciones Anaya Multimedia, 2002. 1005 p. ISBN: 84-415-1384-8
- FIELDING, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.
- CodeIgniter [en línea]. 2012, [citado abril 2012]. Disponible en: <http://codeigniter.com/>
- The jQuery Foundation, JavaScript library. [en línea]. 2012, [citado febrero 2012]. Disponible en: <http://jquery.com/>
- FreeMarker: java template engine library - overview. [en línea] Disponible en: <http://freemarker.org/>. [citado: Abril 2012].
- IEEE: Guide for developing system requirements specification., 1998. [citado: Abril 2012].
- PRESSMAN, Roger S. *Ingeniería del Software Un enfoque práctico*. La Habana: Ediciones Félix Varela, 2005. 601 p. ISBN: 970-105-473-3
- *Laboratorio Nacional de Calidad del Software de INTECO, Guía de Desarrollo y Gestión de Requisitos de la Adquisición* .España:Instituto Nacional de las Tecnologías de la Comunicación

- (INTECO), noviembre 2009. Disponible en:[http://www.inteco.es/calidad\\_TIC/descargas/guias/guia\\_requisitos\\_adquisicion](http://www.inteco.es/calidad_TIC/descargas/guias/guia_requisitos_adquisicion)
- SOMMERVILLE, Ian. *Ingeniería de Software*. La Habana: Ediciones Félix Varela, enero 2005. 687 p. ISBN: 84-7829-074-5
  - LARMAN, Craig. *UML y Patrones Introducción al análisis y diseño orientado a objetos*. La Habana: Ediciones Félix Varela, 2004. 492 p. ISBN: 842-053-438-2
  - UNE ISO. Información y documentación: Gestión de documentos Parte 1 Generalidades, 2006. (ISO 15489-1)
  - Sistema de gestión documental y beneficios de su implantación en los procesos de negocio. [en línea]. Barcelona, 2011, [citado febrero 2012]. Disponible en: <http://www.pixelware.com/sistemas-gestion-documental.htm>
  - Ecured, Flujo de trabajo análisis y diseño. [en línea].[citado febrero 2012]. Disponible en: [http://www.ecured.cu/index.php/Flujo\\_de\\_Trabajo\\_Análisis\\_y\\_Diseño#Objetivos\\_del\\_análisis\\_y\\_diseño](http://www.ecured.cu/index.php/Flujo_de_Trabajo_Análisis_y_Diseño#Objetivos_del_análisis_y_diseño)
  - Arquitectura 3 Capas. Disponible en: <http://www.slideshare.net/Decimo/arquitectura-3-capas> [citado marzo 2012].
  - Arquitectura de software Disponible en: <http://es.scribd.com/doc/51339509/Arquitecturadesoftware> [citado 10 marzo 2012].
  - W. Jason Gilmore and Bob Bryla *Beginning PHP and Oracle :From Novice to Professional*, Apress. ISBN 978-1-59059-770-5. Disponible en: <http://catalogoenlinea.uci.cu>.
  - Sistemas de Gestión Documental Disponible en: <http://www.pixelware.com/sistemas-gestion-documental.htm> [citado noviembre del 2011]
  - Visual paradigm for UML user's guide. <http://www.visual-paradigm.com/support/documents/vpumluserguide.jsp> [citado Enero 2012].

- Motor jbpn Disponible en: <http://docs.jboss.com/jbpm/v3.2/userguide/html/ch01s06.html> [citado Febrero 2012].
- GOODMAN Danny. *Dynamic HTML The Definitive Reference*, O'REILLY, EEUU, Julio 1998.
- SAETHER, S, et al. Manual de php. Editores: Rafael Martínez, Víctor Fernández, Leonardo Boshell. Disponible en: [www.calitae.com/manuales/manual-php.pdf](http://www.calitae.com/manuales/manual-php.pdf) [citado Marzo 2012] .

## Anexo A

### Primer apéndice

#### A.1. Descripción de los casos de uso del sistema

<b>Caso de uso</b>	<b>Reasignar flujo de trabajo avanzado.</b>	
<b>Actor</b>	Usuario	
<b>Resumen</b>	El caso de uso comienza cuando el usuario desea reasignar un flujo de trabajo avanzado que tiene pendiente, terminando el caso de uso una vez realizada la acción.	
<b>Prioridad</b>	Alta	
<b>Complejidad</b>	Media	
<b>Precondiciones</b>	El usuario tiene que estar autenticado en el sistema y debe existir al menos un flujo de trabajo avanzado asignado a él.	
<b>Poscondiciones</b>	El flujo de trabajo avanzado es reasignado.	
<b>Flujo de eventos</b>		
<b>Flujo básico “Reasignar flujo de trabajo avanzado”</b>		
<b>Actor</b>	<b>Sistema</b>	
1. Selecciona en la barra de funcionalidad el botón para listar los flujos de trabajo que tiene pendientes.	2. Muestra una interfaz con un listado de los flujos de trabajo que tiene pendiente el usuario.	
Continúa en la próxima página		

3. Selecciona el ícono “Reasignar flujo de trabajo avanzado” del que se desea modificar.	4. Muestra en una nueva ventana el formulario con los campos necesarios para reasignar el flujo de trabajo avanzado seleccionado.
5. Inserta los campos necesarios y envía los datos presionando un botón.	6. Verifica que los campos obligatorios estén llenos.
	7. En caso correcto actualiza en el repositorio el flujo de trabajo avanzado reasignado y muestra un mensaje de confirmación.
	8. Notifica al usuario que se le asignado un flujo de trabajo. Termina el caso de uso.
<b>Flujos alternos</b>	
<b>Actor</b>	<b>Sistema</b>
2.1.	Muestra el mensaje: “No existen tareas pendientes”.
5.1. Una vez insertados los datos cancela el envío.	No se realiza ningún cambio y se muestra la interfaz con el listado de flujos de trabajo pendientes del usuario.
6.1. Muestra un mensaje con el nombre del campo que está vacío.	

Tabla A.1: Descripción textual del CU: Reasignar flujo de trabajo avanzado.

<b>Caso de uso</b>	<b>Mostrar notificaciones</b>
Continúa en la próxima página	

<b>Actor</b>	Usuario
<b>Resumen</b>	El caso de uso comienza cuando el usuario desea visualizar sus notificaciones, terminando el caso de uso una vez realizada la acción.
<b>Prioridad</b>	Baja
<b>Complejidad</b>	Media
<b>Precondiciones</b>	El usuario tiene que estar autenticado en el sistema y debe haberse creado o reasignado un flujo de trabajo al usuario.
<b>Poscondiciones</b>	
<b>Flujo de eventos</b>	
<b>Flujo básico “Mostrar notificaciones”</b>	
<b>Actor</b>	<b>Sistema</b>
1. Selecciona en la barra de acciones globales el ícono para listar las notificaciones.	2. Muestra una interfaz con un listado de las notificaciones pertenecientes al usuario.
<b>Flujos alternos</b>	
<b>Actor</b>	<b>Sistema</b>

Tabla A.2: Descripción textual del CU: Mostrar notificaciones.

<b>Caso de uso</b>	<b>Modificar estado de la notificación</b>
<b>Actor</b>	Usuario
Continúa en la próxima página	

<b>Resumen</b>	El caso de uso comienza cuando el usuario desea cambiar el estado de la notificación de no leída a leída, terminando el caso de uso una vez realizada la acción.	
<b>Prioridad</b>	Baja	
<b>Complejidad</b>	Media	
<b>Precondiciones</b>	El usuario tiene que estar autenticado en el sistema y debe existir a menos una notificación.	
<b>Poscondiciones</b>	Se elimina la notificación	
<b>Flujo de eventos</b>		
<b>Flujo básico “Modificar estado de la notificación”</b>		
<b>Actor</b>	<b>Sistema</b>	
1. Selecciona en la barra de acciones globales el ícono para listar las notificaciones.	2. Muestra una interfaz con un listado de las notificaciones pertenecientes al usuario.	
3. Selecciona el ícono Aceptar de la notificación.	4. Muestra un mensaje confirmando el éxito de la operación.	
5.	Se elimina del repositorio de contenido la notificación.	
<b>Flujos alternos</b>		
<b>Actor</b>	<b>Sistema</b>	

Tabla A.3: Descripción textual del CU: Modificar estado de la notificación.

## Anexo B

### Segundo apéndice

#### B.1. Diagrama de clases de diseño

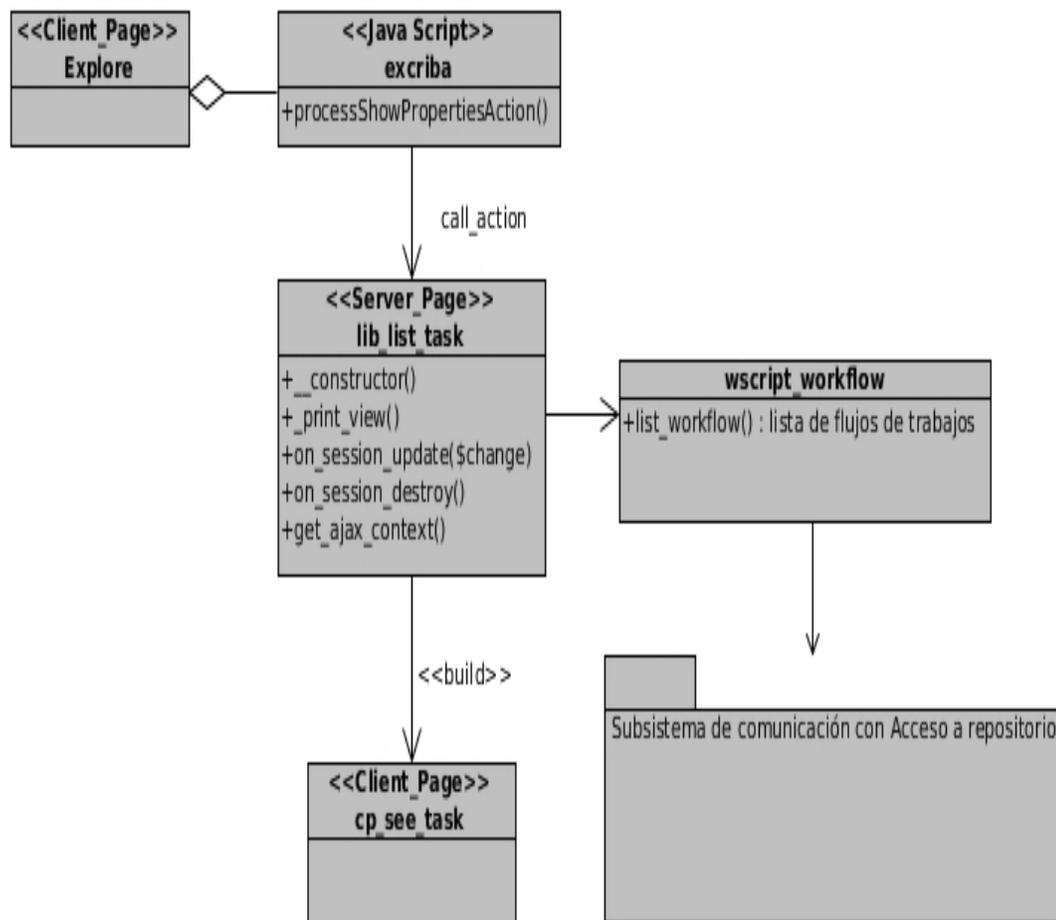


Figura B.1: Diagrama de clases del diseño listar flujos de trabajo

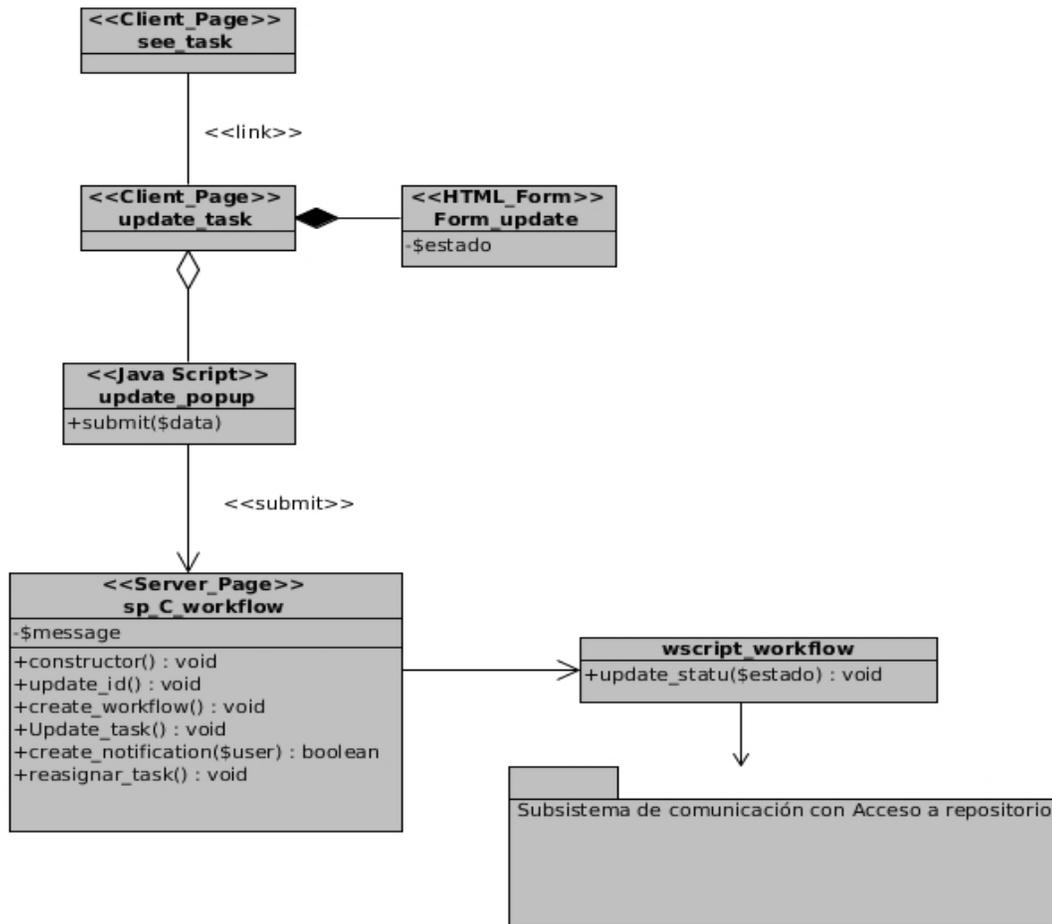


Figura B.2: Diagrama de clases del diseño listar flujos de trabajo

## Anexo C

### Tercer apéndice

#### C.1. Diagrama de secuencia del diseño

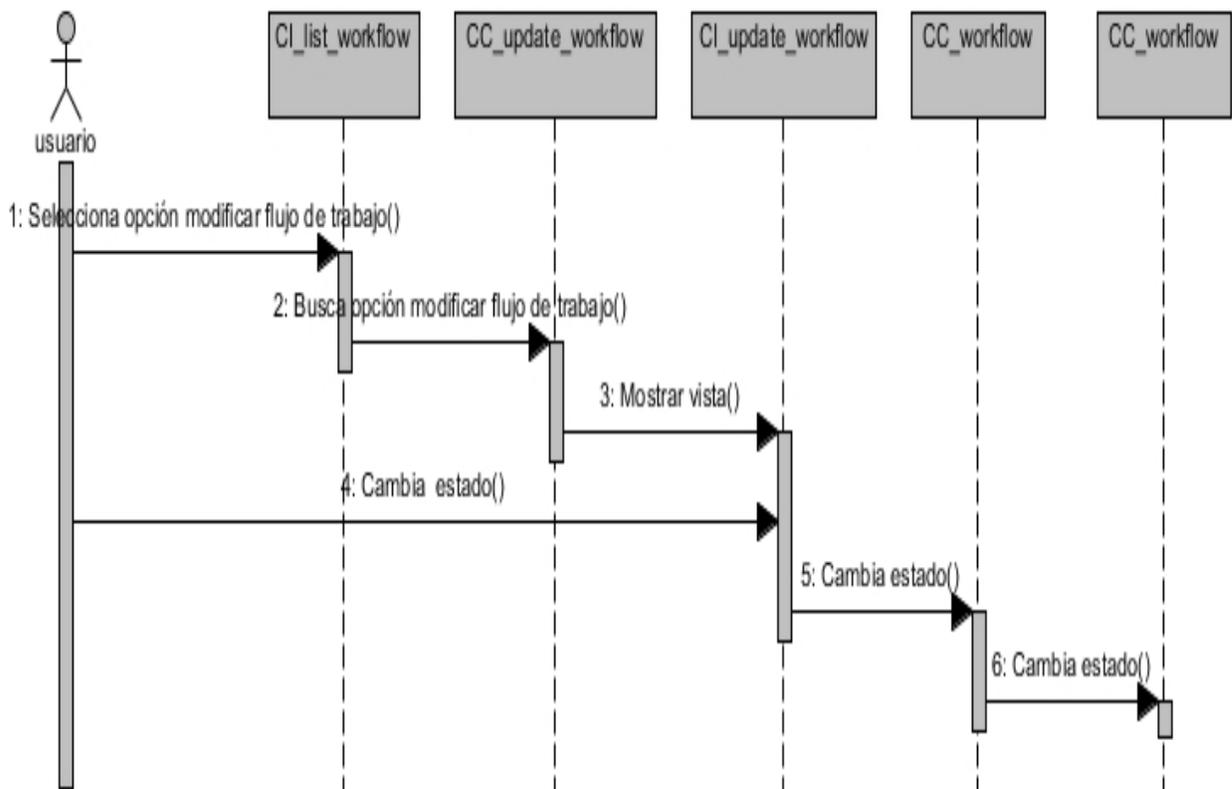


Figura C.1: Diagrama de secuencia del modificar flujo de trabajo pendiente

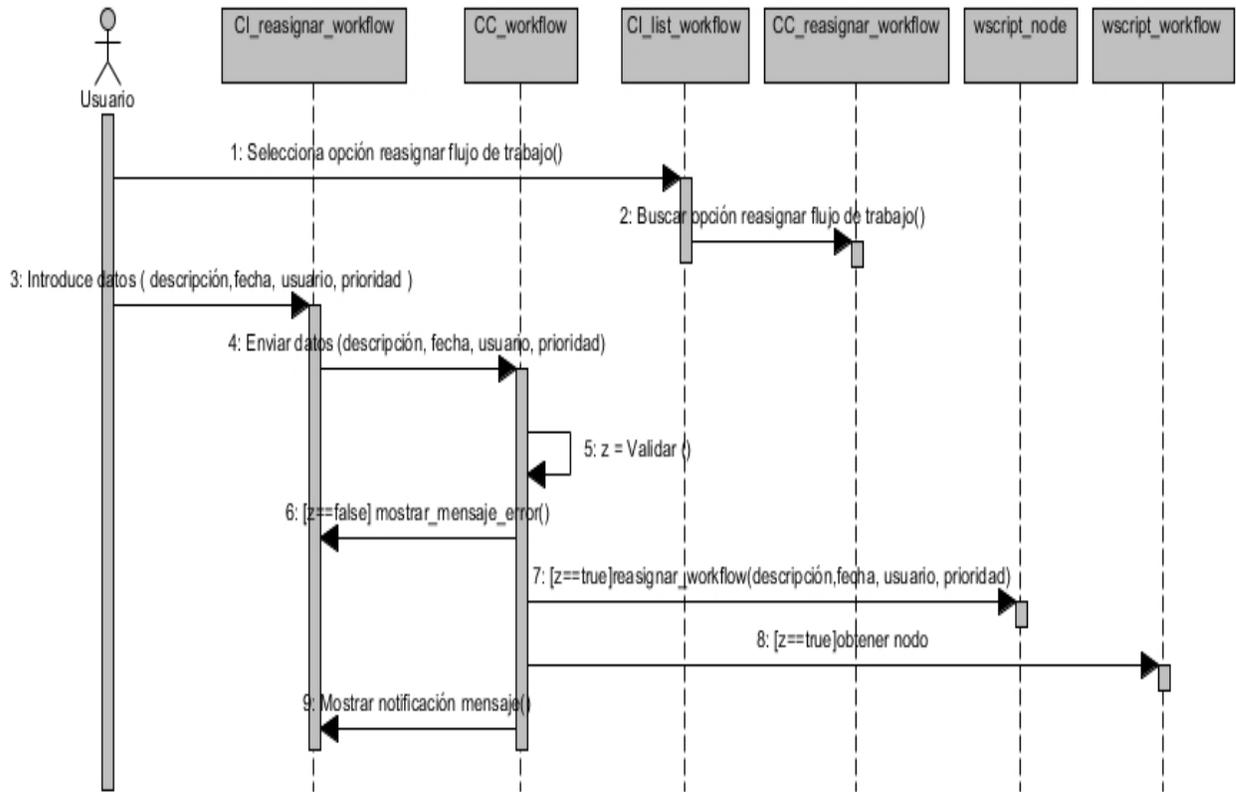


Figura C.2: Diagrama de secuencia del reasignar flujo de trabajo pendiente