



Universidad de las Ciencias Informáticas

Facultad 1

**Solución para la ejecución de flujos de trabajo de aprobación
formal de documentos electrónicos en el Gestor de Documentos
Administrativos eXcriba**

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

Autor:

Yanet Sibat Martinez

Tutores:

Ing. Pedro Rodriguez Samon

Lic. Jennisley Verde Acosta

Consultante: Ing Reinier Elejalde Chacon

La Habana, Cuba
Junio 2013



“Tenemos que ayudar todos los días al nacimiento de la conciencia cuando no la haya, y al fortalecimiento de la conciencia cuando ya exista, porque la Revolución se alimenta de la conciencia de un pueblo.”

Hugo Chávez Frías

Declaración de autoría

Declaro que soy la única autora 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 ____ días del mes de _____ del año _____.

Firma del Autor

Yanet Sibat Martinez

Firma del Tutor

Ing. Pedro Rodriguez Samon

Firma del Tutor

Lic. Jenisley Verde Acosta

Agradecimientos

Quisiera agradecer primeramente a esta Revolución y a esta Universidad por haberme dado la oportunidad de ver realizado el mayor de mis sueños.

A mis padres y hermana por ser un ejemplo de sacrificio, confianza, comunicación y sobre todas las cosas de mucho amor. Los amo.

Gracias a mi mami por siempre estar dispuesta a escucharme, por comprenderme, por consentirme y también a regañarme. Por apoyarme en todo momento y confiar en mí, por enseñarme a ser una mujer de bien y por estar orgullosa de mí ante todas las cosas y por ser más que una madre, ser una amiga. Te amo.

A mi papi por confiar siempre en mí, por recordarme todos los días que nada cae del cielo y que las cosas para lograrlas hay que luchar por ellas y poner empeño, y por ser ese hombre inteligente, preocupado y mi inspiración a seguir.

A mi herma por ser mi fuente de inspiración, por ser mi confidente, mi amiga. Te amo.

A mi novio, por estar en las buenas y en las malas, por aguantar mis malos humores en estos días, por enseñarme las cosas buenas de la vida, por quererme, por mimarme y por ser el promotor de nuestros mejores momentos.

A mi abuela Emilia por darme los mejores momentos de mi niñez y cuidarme siempre.

A mi abuela Vicenta que aunque no pudo ver mi sueño hecho realidad sé que estuvo acompañándome en todo momento y debe estar muy orgullosa desde el lugar donde esté.

A toda mi familia materna y paterna por apoyarme siempre y entender todos esos momentos en los que no estuve presente.

A mis suegros y mi cuñado por preocuparse de cada paso que doy en la vida y por acogerme como un miembro más de su familia. Gracias.

Agradecimientos

Agradecerle a una persona en especial a Reinier, por su apoyo incondicional, por su paciencia, por ser un amigo ejemplar, por sacarme siempre una sonrisa y por tener el don tan especial de que todo el mundo lo quiere. Gracias mil veces gracias.

A mis amigas que ya no están Isa, Zoe, Liset, Clau, por todos esos momentos que compartimos juntas, a Jota, Taimi, Victor, gracias por su amistad, por hacerme reír en los turnos de clase y hacer que el profesor de Ética siempre me regañara a mi nada más, a Marbe, Marisol, Migue, Alex, Jose, por brindarme su ayuda y apoyo incondicional en todo momento.

A mis tutores Pedro y Jenisley y al jurado aquí presente, gracias por sus consejos.

A todos los que hicieron que estos cinco años, se convirtieran en los mejores años de mi vida.

Yanet Sibat Martinez

Dedicatoria

A mis padres Ada y Ricardo porque sin ellos no hubiera llegado hasta donde estoy, este es nuestro sueño.

A mi herma, por ser mi fuente de inspiración.

A mi abuelita Emilia, por ser la mejor Abu del mundo.

A mi abuelita Vicenta, por esta siempre presente en mi corazón.

A toda mi familia y a mi futuro sobrino.

Yanet Sibat Martinez

El Gestor de Documentos Administrativos eXcriba ha cobrado un auge extraordinario dentro de las organizaciones, ya que su objetivo principal es automatizar los procesos documentales que se ejecutan dentro de cualquier entidad. Sin embargo, la aprobación formal de documentos electrónicos es un proceso para el cual eXcriba no provee las funcionalidades necesarias para su correcta ejecución, ya sea de forma secuencial o paralela.

Como solución a esta problemática la presente investigación propone una aplicación que a partir de un conjunto de flujos de trabajo contribuya a la ejecución del proceso de aprobación formal de documentos electrónicos.

Para ello se describe un marco conceptual e informativo relacionado con la gestión de procesos de negocio y flujos de trabajo. Se realiza una valoración de las características funcionales de sistemas de gestión documental, que definen y ejecutan flujos de trabajo del proceso de aprobación, y se define un marco de trabajo para el desarrollo de la solución. El presente trabajo da continuidad a la tesis que lleva por título: Infraestructura de servicios para la ejecución de flujos de trabajo de firma digital en el Administrador de Contenidos Empresariales Alfresco sobre la cual se realiza una valoración crítica del diseño propuesto. Se valida la propuesta de solución a través del desarrollo de un conjunto de pruebas realizadas al producto final.

Palabras clave: aprobación formal de documentos, flujos de trabajo, procesos de negocio.

Índice general

Índice de figuras	VI
Índice de tablas.....	VII
Introducción.....	1
Capítulo 1. "Fundamentación teórica de la investigación".....	6
1.1. Proceso de negocio	6
1.2. Gestión de Procesos de Negocio (BPM).....	7
1.3. Flujo de trabajo.....	8
1.4. Descripción general del proceso de aprobación formal de documentos.....	8
1.5. Los flujos de trabajo de aprobación en sistemas de gestión documental.....	10
1.5.1. Análisis de las funcionalidades de los sistemas.....	13
1.6. Proceso de desarrollo de software	13
1.7. Modelo de Capacidad y Madurez Integrado (CMMI).....	14
1.8. Metodología de desarrollo	14
1.9. Entorno de desarrollo	15
1.9.1. Lenguajes	16
1.9.2. Herramientas	18
1.9.3. Tecnologías	19
1.9.3.1. Framework.....	20
1.10. Conclusiones del capítulo	23
Capítulo 2. "Descripción y análisis de la solución propuesta."	24
2.1. Especificación de requisitos	24
2.2. Validación de los requisitos	27
2.3. Especificación de Casos de Uso.....	31
2.3.1. Definición de los actores	31
2.3.2. Análisis del diagrama de caso de uso	32
2.3.3. Descripción de los casos de uso	32
2.4. Diseño del módulo	35
2.4.1. Patrón de arquitectura.....	35
2.4.2. Patrones de flujos de trabajo.....	38
2.4.2.1. Patrones de flujo básico	38
2.4.2.2. Patrones estructurales.....	40
2.4.3. Diagrama de clases del diseño	41
2.4.3.1. Descripción de las clases	42
2.4.4. Estados de los flujos de trabajo.....	44
2.4.5. Diagrama de secuencia del diseño.	46
2.5. Conclusiones del capítulo	46
Capítulo 3. "Implementación y prueba del módulo."	47
3.1. Descripción de los servicios	47
3.2. Estándar de codificación	49
3.3. Diagrama de despliegue	51
3.4. Diagrama de componentes.....	53
3.4.1. Descripción del diagrama de componentes	55

3.5. Pruebas de software	55
3.5.1. Pruebas de caja negra	56
3.5.1.1. Técnicas existentes	56
3.5.1.2. Casos de prueba de caja negra	57
3.4.1.3. Resultados de las pruebas realizadas	61
3.6. Conclusiones del capítulo	62
Conclusiones generales.....	63
Recomendaciones	64
Referencias bibliográficas	65
Bibliografía.....	69
Anexos.....	73

Índice de figuras

Ilustración 1. Aprobación en paralelo.....	9
Ilustración 2. Aprobación en serie.....	10
Ilustración 3. Arquitectura MVC de Alfresco Webscripts.....	23
Ilustración 4. La validación en el proceso de requisitos.....	28
Ilustración 5. Prototipo crear petición de aprobación.....	30
Ilustración 6. Prototipo crear petición de aprobación.....	31
Ilustración 7. Diagrama de Casos de Uso.....	32
Ilustración 8. Arquitectura de la propuesta de solución.....	37
Ilustración 9. Patrón de secuencia.....	39
Ilustración 10. Patrón de separación paralela.....	39
Ilustración 11. Patrón de sincronización.....	39
Ilustración 12 Patrón de ciclos arbitrarios.....	40
Ilustración 13. Diagrama de clases del diseño: Crear petición.....	41
Ilustración 14. Diagrama de secuencia del caso de uso: Gestionar petición.....	46
Ilustración 15. Diagrama de despliegue.....	52
Ilustración 16. Diagrama de componente del módulo flujos de trabajo de aprobación.....	54
Ilustración 17. Crear petición de aprobación en serie.....	73
Ilustración 18. Crear petición de aprobación en paralelo.....	74
Ilustración 19. Bandeja de tareas.....	75
Ilustración 20. Listado de las peticiones.....	76
Ilustración 21. Mostrar petición.....	77
Ilustración 22. Diagrama de clases del diseño: Listar petición.....	79
Ilustración 23. Diagrama de clases del diseño: Consultar el documento asociado a una petición.....	80
Ilustración 24. Diagrama de secuencia del caso de uso: Consultar el documento asociado a una petición.....	81

Índice de tablas

Tabla 1. Requisitos funcionales. 26
Tabla 2. Requisitos no funcionales. 27
Tabla 3. Lista de errores y acciones recomendadas. 28
Tabla 4. Definición de los actores 31
Tabla 5. Descripción textual del caso de uso: Gestionar petición..... 35
Tabla 6. Descripción de los estados de los flujos de trabajo en serie y en paralelo 45
Tabla 7. Descripción del servicio: Crear flujo de trabajo 48
Tabla 8. Descripción del servicio: Mostrar historial del flujo de trabajo..... 48
Tabla 9 Escenario. Crear flujo de trabajo de aprobación 59
Tabla 10 Escenario. Listar petición de aprobación 59
Tabla 11 Escenario. Consultar el documento asociado a la petición 61
Tabla 12 Tabla de resultados 61
Tabla 13. Descripción textual del CU: Consultar el documento asociado a una petición 78
Tabla 14. Descripción del servicio: Modificar petición 82
Tabla 15. Descripción del servicio: Finalizar flujo de trabajo 83

Introducción

En un momento como el presente, en el que la humanidad camina hacia una civilización radicalmente distinta, en que la computadora es la base de todo, parece necesario analizar no solo las repercusiones que el desarrollo de las nuevas tecnologías puede tener en el mundo, sino reflexionar si tenemos resuelto debidamente los múltiples problemas planteados con el aumento de los grandes volúmenes de información que se generan día a día.

Con el constante crecimiento de las empresas, a grandes escalas, los documentos generados en ocasiones son objeto de pérdida, pueden encontrarse duplicados e inclusive diseminados en distintos soportes. Esta situación obstaculiza la efectividad de los procesos de negocios que se realizan a diario, como la búsqueda de información, la recuperación, el control de versiones, la clasificación y la aprobación formal de documentos. Ocasionando una baja productividad y elevados costes para las empresas.

La Gestión Documental ha surgido como respuesta a la gran proliferación de documentos y a las necesidades de acceso a ellos; permitiendo un control y tratamiento uniforme de la documentación administrativa. Si bien se ha incrementado la producción documental, ha cobrado también un auge extraordinario el desarrollo de sistemas para la gestión de documentos, sin embargo, no todos cumplen con las exigencias de las modernas administraciones.

Como parte del desarrollo de estos sistemas, se encuentra en el Centro Cenía, de la Universidad de las Ciencias Informáticas (UCI) el **Gestor de Documentos Administrativos** llamado **eXcriba (GDA eXcriba)**, cuyo objetivo principal es automatizar los procesos documentales que se ejecutan dentro de cualquier entidad, desde la elaboración de un documento en su fase de inicio, hasta su conservación o expurgo¹, incluyendo todas las acciones sobre los documentos, dentro de las que se encuentran: crear, clasificar, control de

¹ Retirar las copias conocidas como inútiles.

versión, definir Tipologías Documentales, gestionar flujos de trabajo², almacenar documentos en diferentes formatos electrónicos, gestionar los trámites de los documentos que se generan o reciben; además está basado en el Gestor de Contenido Empresarial (ECM³ por sus siglas en inglés) Alfresco (FONSECA, 2012).

La gestión de los flujos de trabajo es fundamental, pues ayudan a las organizaciones a ajustarse a procesos empresariales coherentes y mejoran su eficiencia organizativa y productividad al administrar las tareas y los pasos necesarios para los procesos empresariales.

Actualmente, el GDA 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 permitiendo automatizar el movimiento del contenido a través de varios espacios⁴, y avanzados los cuales actualmente se encuentran en proceso de incorporación al eXcriba, ya que su objetivo principal es automatizar los procesos de negocio durante el cual “documentos”, “información y “tareas” son pasados de un participante a otro (SHARIFF, 2006). Sin embargo, eXcriba carece de funcionalidades para que desde aplicaciones externas se puedan ejecutar diferentes actividades de un proceso de negocio definido tanto de forma secuencial como paralela por distintos miembros para la obtención de un mismo objetivo.

El proceso de aprobación formal de documentos, *“acto mediante el cual una o varias personas le dan validez y vigencia a los requisitos que contiene un documento mediante la firma digital o física”* (CALIDAD, 2004), es afectado por esta situación y es un proceso de vital importancia ya que es utilizado para certificar y validar la autoría de la documentación que se gestiona en las empresas. Para ejecutar lo es necesario:

1. Buscar el documento que se desea aprobar en la aplicación para luego imprimirlo.
2. Llevar el documento ante la persona o personas que deben firmarlo.

² Conjunto de actividades de un proceso de negocio ejecutadas de forma secuencial o paralela por distintas personas para la obtención de un mismo objetivo.

³ *Enterprise Content Management.*

⁴ Carpeta con propiedades adicionales, tales como reglas de negocio y de seguridad. Un espacio puede tener cualquier tipo de contenido sobre el cual los usuarios pueden ejecutar distintas acciones.

3. Escanear el documento aprobado para reincorporarlo a la aplicación.

Esta problemática trae consigo un retraso en la ejecución del proceso de aprobación formal de documentos y de los demás procesos y actividades de las entidades que dependen de él; además se convierte en un procedimiento agotador que reduce en gran medida la productividad y el desempeño de los usuarios.

A raíz de lo anteriormente expuesto, surge el siguiente **problema de investigación**: ¿Cómo aprobar formalmente documentos electrónicos en el GDA eXcriba?

Definiendo como **objeto de estudio** a los procesos de negocio. Asumiendo como **objetivo general**: desarrollar un módulo para el GDA eXcriba, que a partir del consumo de un conjunto de servicios de repositorio, contribuya a la ejecución del proceso de aprobación formal de documentos electrónicos, desde aplicaciones externas. Enmarcando su **campo de acción** en los procesos de aprobación formal de documentos electrónicos.

Con vistas al alcance de los objetivos se hace necesario realizar las siguientes **tareas de investigación**:

- ◆ Analizar los antecedentes y aspectos teóricos relacionados con el proceso de aprobación formal de documentos electrónicos.
- ◆ Valorar el diseño del módulo para la ejecución de flujos de trabajo en el GDA eXcriba.
- ◆ Implementar el módulo para la ejecución de flujos de trabajo en el GDA eXcriba.
- ◆ Seleccionar las técnicas de validación para las pruebas de caja negra.
- ◆ Validar el correcto funcionamiento del módulo mediante las pruebas caja negra.

Para realizar la actividad científica-investigativa se han utilizado los métodos siguientes:

Métodos teóricos

- ◆ **Histórico-Lógico**: permitió estudiar el objeto en su desarrollo, mediante el decurso histórico y cronológico de los procesos de negocio. Además, se empleó para determinar la evolución de los diferentes sistemas de gestión documental con respecto al proceso de aprobación formal de documentos electrónicos.

- ◆ **Analítico-Sintético:** se tuvo en cuenta especialmente en todo el proceso de diseño e interpretación de la aprobación formal de documentos, así como en la valoración de aspectos y conceptos básicos como flujos de trabajo y procesos de negocio, los cuales permitieron el desarrollo de la investigación.
- ◆ **Modelación:** se utilizó para modelar los diagramas que se obtuvieron durante el proceso de diseño del módulo para la aprobación formal de documentos electrónicos en el GDA eXcriba.

La propuesta de solución posibilitará:

- ◆ Eliminar la dependencia por medio del papel.
- ◆ Reducir los costos operacionales.
- ◆ Agilizar la ejecución del proceso de aprobación formal de documentos electrónicos desde aplicaciones externas.
- ◆ Incrementar la productividad de los usuarios que interactúan con el GDA eXcriba.
- ◆ Automatizar el proceso de aprobación formal de documentos tanto en paralelo como en serie en el GDA eXcriba.

Para un mejor entendimiento tanto del problema como de la solución que se propone, el trabajo de diploma se ha estructurado en tres capítulos, los cuales serán descritos a continuación:

Capítulo 1: Fundamentación teórica de la investigación: se introducen conceptos y aspectos fundamentales de la gestión de procesos de negocio, flujos de trabajo y la descripción general del proceso de aprobación formal de documentos. Se realiza un estudio de los diferentes sistemas que brindan entre sus funcionalidades la ejecución de flujos de trabajo relacionados con el proceso de aprobación formal de documentos electrónicos. Finalmente, se precisan las herramientas, lenguajes y tecnologías que se utilizan durante el proceso de desarrollo de la propuesta de solución.

Capítulo 2: Propuesta de solución: se realiza la valoración crítica del diseño propuesto proveniente de la tesis que lleva por título: Infraestructura de servicios para la ejecución de flujos de trabajo de firma digital en el Administrador de Contenidos Empresariales Alfresco, se

describen clases, requisitos y operaciones necesarias para darle solución al problema.

Capítulo 3: Implementación y prueba del módulo: se explica el proceso de implementación y se plasman los casos de pruebas a los que fue sometido el módulo en cada una de las iteraciones. Se exponen los resultados obtenidos y se muestran las funcionalidades alcanzadas en el período de desarrollo.

Capítulo 1. “Fundamentación teórica de la investigación”

Capítulo 1. “Fundamentación teórica de la investigación”

Hoy en día, la tecnología es un factor crítico en el éxito o fracaso de un negocio. Compañías que incorporan sistemas exitosamente en su infraestructura de información se posicionan para lograr ventajas competitivas sostenidas. En el ámbito de negocios de hoy, eso significa implementar sistemas que proveen operaciones simplificadas y de bajo costo. Otras capacidades críticas, incluyen la habilidad de comunicar información a tiempo y certeramente a la creciente fuerza de trabajo ambulante, mejorando continuamente la productividad del empleado y la capacidad de proteger la integridad de los sistemas y la propiedad intelectual.

En el presente capítulo se exponen elementos conceptuales y características que permiten centrarse en el contexto de la investigación. Se realiza un estudio de diferentes sistemas que brindan entre sus funcionalidades la ejecución de flujos de trabajo relacionados con el proceso de aprobación formal de documentos electrónicos, y finalmente, se precisan las herramientas, lenguajes y tecnologías que se utilizarán durante el proceso de desarrollo de la propuesta de solución.

1.1. Proceso de negocio

Thomas Davenport⁵ señala que un proceso de negocio es un “conjunto estructurado y medible de actividades diseñadas para producir un producto especificado para un cliente o mercado específico” (SEVILLANO, 2009).

Las organizaciones se apoyan en sus procesos de negocios para ser guiados en este complejo escenario o para determinar la manera en que la empresa opera, de ahí la importancia de establecerlos. No obstante, en muchas organizaciones, dada la complejidad del mismo, existe una importante diferencia entre los procesos que deberían estar implantados y los procesos que se encuentran operando el negocio en realidad.

⁵ Tom Davenport es profesor distinguido de Tecnología y Gestión de la Información del Babson College y Director de Investigación. Dirige programas de investigación patrocinado en inteligencia analítica, gestión del conocimiento, gestión de procesos, e innovación.

Capítulo 1. “Fundamentación teórica de la investigación”

La aprobación formal de documentos es un ejemplo de proceso de negocio que se desarrolla en diferentes organizaciones, incluye distintos participantes como: los revisores y los solicitantes. Para su desarrollo es necesario realizar un conjunto de tareas como: asignar el documento a la persona correspondiente para su evaluación y revisar el documento.

1.2. Gestión de Procesos de Negocio (BPM)

La gestión por procesos puede definirse como una forma de enfocar el trabajo, donde se persigue el mejoramiento continuo de las actividades de una organización mediante la identificación, selección, descripción, documentación y mejora continua de los procesos. Toda actividad o secuencia de actividades que se llevan a cabo en las diferentes unidades constituye un proceso y como tal, hay que gestionarlo (PEPPER, 2011).

La gestión de procesos ve a los procesos como creaciones humanas, con todas las posibilidades de acción sobre ellos: diseñar, describir, documentar, comparar, eliminar, modificar, alinear o rediseñar. Reconoce que los procesos no pueden estar abandonados a su suerte y establece formas de intervención que tienen por objetivo cumplir la estrategia de la organización y mejorar en múltiples aspectos deseables: eficiencia, atención al cliente, calidad y productividad. Acepta que no tiene finalidad por sí misma, sino que es un medio para lograr grandes metas organizacionales.

Con la gestión de los procesos de negocio las organizaciones se favorecen de los siguientes beneficios:(CARRASCO, 2011).

- ◆ conocer lo que se hace y cómo lo hacen, lo que permite tener consciencia de las fortalezas y carencias de la organización,
- ◆ aplicar métodos de mejora continua y aseguramiento de calidad que permiten aumentar la eficiencia y la eficacia,
- ◆ rediseñar un proceso para obtener rendimientos mayores,
- ◆ innovar a diferentes niveles de profundidad (proceso, actividad y tarea),
- ◆ facilitar el emprendimiento porque todo nuevo negocio debe sustentarse en procesos del negocio que deben estar bien definidos.

Capítulo 1. “Fundamentación teórica de la investigación”

1.3. Flujo de trabajo

La WORKFLOW MANAGEMENT COALITION o WFMC define que “*un flujo de trabajo implica la automatización de procesos organizacionales donde tareas, documentos e información son pasados de un participante a otro de acuerdo a un conjunto definido de reglas para alcanzar o contribuir a alcanzar un objetivo de la organización*” (WORKFLOW MANAGEMENT COALITION, 1999).

En el contexto de la presente investigación se define un flujo de trabajo como una colección de actividades de un proceso de negocio ejecutadas de forma secuencial o paralela por distintos miembros para la obtención de un mismo objetivo.

Los flujos de trabajo permiten modelar los procesos que ocurren en una organización y aportan un grupo de beneficios como (PIXELWARE, 2012):

- ◆ ahorro de tiempo y mejora de la productividad y eficiencia de la empresa debido a la automatización de muchos procesos de negocio,
- ◆ mejora del control de procesos,
- ◆ mayor flexibilidad en los procesos pues permiten su rediseño de acuerdo con las necesidades cambiantes del negocio,
- ◆ optimización de la circulación de información interna entre el personal de la organización,
- ◆ integración de procesos empresariales.

1.4. Descripción general del proceso de aprobación formal de documentos

Se entiende por aprobación formal de un documento al acto mediante el cual una o varias personas le dan validez y vigencia a los requisitos que contiene. La aprobación es competencia de la autoridad que haya sido designada para realizarla. Este hecho se plasma sobre el documento mediante la firma (digital o física), o añadiendo el nombre de la persona que lo aprueba (CALIDAD, 2004). El documento puede no ser aprobado si no se está de acuerdo con lo que en él se especifica.

Capítulo 1. “Fundamentación teórica de la investigación”

Para realizar el proceso se ejecutan los flujos de trabajo de aprobación que siguen el proceso de firmado por el cual pasa el documento que puede ser aprobado por una o más personas. Cuando varios miembros deben realizar la aprobación sobre el mismo documento, el proceso de firmado se divide en dos tipos:

En paralelo: cuando el documento necesita ser firmado por un conjunto de aprobadores que mediante su firma o nombre aprueban el documento de forma independiente (Ilustración 1).

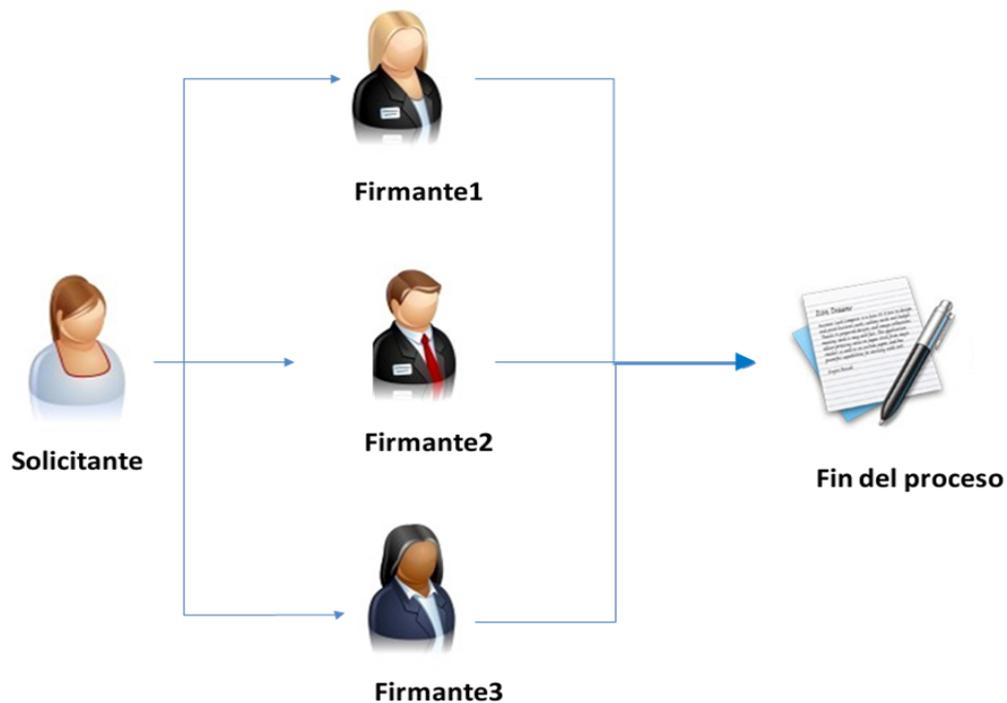


Ilustración 1. Aprobación en paralelo.

En serie: cuando el documento necesita ser firmado por un conjunto de aprobadores de forma única en una secuencia, es decir, el documento será aprobado por un firmante junto con las firmas realizadas por los demás firmantes hasta el momento, el proceso termina cuando el documento es firmado por todas las personas correspondientes (Ilustración 2).

Capítulo 1. “Fundamentación teórica de la investigación”



Ilustración 2. Aprobación en serie.

Este proceso puede ejecutarse de forma independiente o puede formar parte de otros procesos de negocios como el proceso de elaboración de un proyecto. Su correcta ejecución es de vital importancia debido a que certifica la validez de los documentos que circulan en las empresas u organizaciones.

1.5. Los flujos de trabajo de aprobación en sistemas de gestión documental

Nuxeo (BARROCA, 2010)

Nuxeo es una plataforma para la Gestión de Contenido Empresarial. Dispone de algunas de las opciones que facilitan la tarea de gestionar los documentos en las empresas como son: servidor de documentación, espacio de usuario y de grupos, versiones, flujos de trabajo y notificaciones.

Esta herramienta incluye una función de flujo de trabajo que permite a los usuarios configurar tareas sencillas o tareas con varias etapas. El flujo de trabajo ayuda a que el contenido fluya a través de procesos de negocio. Permite establecer flujos de trabajo que son guardados para su posterior reutilización con una plantilla para garantizar la coherencia de los procesos a través de los equipos.

La plataforma tiene definido y permite ejecutar los siguientes flujos de trabajo:

- ◆ flujo de trabajo en paralelo,
- ◆ flujo de trabajo en serie,
- ◆ flujo de trabajo de aprobación.

Los flujos de trabajo en paralelo y en serie son creados por un usuario y se diferencian por el orden de asignación. En el flujo de trabajo en paralelo, todos los usuarios pueden revisar el

Capítulo 1. “Fundamentación teórica de la investigación”

documento tan pronto como el flujo de trabajo sea iniciado. Por otra parte, el flujo de trabajo en serie es ordenado, es decir, que los participantes puedan revisar el documento solo cuando el usuario anterior de la lista haya aprobado el documento.

El flujo de trabajo de aprobación es creado por un usuario quien define las propiedades del flujo de trabajo y los usuarios que participan en el flujo. Cuando los participantes hayan aprobado el documento, el flujo de trabajo queda automáticamente finalizado.

SharePoint (OFFICE, 2012)

SharePoint es una plataforma de colaboración empresarial que permite incrementar la productividad y administrar los contenidos, incluye un conjunto de flujos de trabajo de aprobación que se pueden usar para administrar los procesos empresariales (un flujo de trabajo de aprobación distribuye elementos a personas específicas para su aprobación). Administra y hace un seguimiento de todas las tareas humanas implicadas en el proceso y proporciona un registro del proceso una vez que este se completa.

En SharePoint los flujos de trabajo de aprobación sustentan cualquier proceso empresarial que requiere enviar documentos o elementos a un compañero o jefe para su aprobación. Los usuarios inician el flujo de trabajo y luego seleccionan a quién enviar el documento o elemento para ser aprobado, las fechas de vencimiento y las instrucciones relevantes. El flujo de trabajo asigna automáticamente la tarea al primer participante en el flujo y si las alertas de correo electrónico están habilitadas, se enviará un mensaje de correo electrónico a los participantes.

Si el flujo de trabajo tiene varios participantes las tareas pueden ser asignadas en serie o en paralelo. Cuando los participantes reciben el correo electrónico de las tareas o van directamente a la lista de tareas, pueden ver el elemento o documento y las acciones de aprobar, rechazar, solicitar cambios o reasignar el documento. Mientras se ejecuta el flujo de trabajo, los participantes y el propietario del flujo pueden ver el estado del mismo. Cuando son terminadas las tareas, el flujo de trabajo finaliza y el propietario recibe una notificación informándole de que el flujo se ha completado.

Capítulo 1. “Fundamentación teórica de la investigación”

Documentum (KUMAR PAWAN, 2010)

EMC Documentum o Documentum es una plataforma de gestión de contenidos empresariales que abarca todo el ciclo de vida de la información, desde su captura hasta su almacenamiento, ofrece una solución completa a las necesidades empresariales de Gestión Documental, proporciona capacidades de gestión para todos los tipos de contenido y un conjunto de servicios, tales como la gestión de documentos, colaboración, búsqueda, clasificación de contenidos, *Business Process Management* (BPM), gestión de comunicación con el cliente y de gestión de contenidos Web.

Los flujos de trabajo son un componente clave de las soluciones Documentum, donde las definiciones de actividades se pueden conectar de las siguientes maneras, directa o indirectamente:

- ◆ **serie (en orden):** si una actividad debe ser completada antes de la segunda puede comenzar, se consideran para ser conectado en serie,
- ◆ **paralelo:** si las dos actividades se pueden llevar a cabo simultáneamente, que son considerado para ser conectado en paralelo.

Define que flujo de trabajo pueda estar en cualquiera de los siguientes estados:

- ◆ **Inactivo:** un flujo de trabajo está inactivo cuando se acaba de crear o si tiene ha reiniciado desde un estado suspendido,
- ◆ **Detenido:** detener el flujo de trabajo se ejecuta mueve al estado detenido,
- ◆ **Terminado:** interrumpir el flujo de trabajo en ejecución o detenido lo mueve a estado terminado,
- ◆ **Acabado:** Finalización normal de un flujo de trabajo se ejecuta lo mueve a la estado terminado,

Permite que al iniciar un flujo, se especifique la siguiente información:

- ◆ los archivos que se envían en el flujo rápido,

Capítulo 1. “Fundamentación teórica de la investigación”

- ◆ intérpretes,
- ◆ estructura del rápido flujo secuencial o paralelo: (para secuencial, si rechaza al ejecutante anterior y / o al supervisor se les permite),
- ◆ las instrucciones para completar la tarea,
- ◆ si el iniciador debe ser notificado en la realización de tareas,
- ◆ prioridad (bajo, medio o alto).

1.5.1. Análisis de las funcionalidades de los sistemas

Los sistemas analizados anteriormente presentan funcionalidades que engloban las generalidades del proceso de aprobación formal de documentos y se utilizan como guía para la obtención de los requerimientos de la solución propuesta, ya que:

- ◆ permiten iniciar y ejecutar flujos de trabajo de aprobación,
- ◆ realizan las asignaciones de forma secuencial o paralela según el tipo de proceso,
- ◆ brindan la posibilidad de conocer el estado en el que se encuentra el flujo de trabajo,
- ◆ registran los flujos de trabajo para su posterior utilización,
- ◆ posibilitan la consulta de los documentos asociados al proceso,
- ◆ permiten definir notificaciones sobre el estado del flujo.

Además, se identificaron elementos que pueden ser útiles para la construcción de la solución propuesta como:

- ◆ tipo de flujo de trabajo que se va a realizar: en serie o en paralelo,
- ◆ datos que conforman las tareas asignadas como: las fechas de caducidad,
- ◆ plantillas para el registro de los flujos de trabajo,
- ◆ lista de usuarios que aprobarán el documento,
- ◆ estado de las peticiones: pendientes, en espera, caducas, terminadas.

1.6. Proceso de desarrollo de *software*

Un proceso de desarrollo de *software* es un conjunto de actividades y resultados que originan un producto de *software*. Estas actividades son llevadas a cabo por los ingenieros de

Capítulo 1. “Fundamentación teórica de la investigación”

software. Se especifican cuatro actividades fundamentales que son comunes para todos los procesos de *software*, estas actividades son: (SOMMERVILLE, 2005).

- ◆ **especificación del *software*** – donde los clientes e ingenieros definen el *software* a producir y las restricciones sobre su operación.
- ◆ **desarrollo del *software*** – donde el *software* se diseña y se programa.
- ◆ **validación del *software*** – para asegurar que es lo que el cliente requiere.
- ◆ **evolución del *software*** – donde el *software* se modifica para adaptarlo a los cambios requeridos por el cliente y el mercado.

1.7. Modelo de Capacidad y Madurez Integrado (CMMI)

CMMI (*Capability Maturity Model Integration*) es un enfoque de mejora de procesos que proporciona a las organizaciones los elementos esenciales de procesos efectivos para mejorar su rendimiento. Basada en CMMI la mejora de procesos incluye la identificación de las fortalezas y debilidades de los procesos de la organización y convertir las debilidades del proceso en fortalezas (NIELSEN, 2012).

CMMI está orientado a la garantía de calidad del *software* y a la acreditación de empresas dedicadas al desarrollo de software en función del nivel de madurez de sus procesos de producción. Representa la fusión de un conjunto de modelos orientados a la mejora de procesos de ingeniería del *software*, ingeniería de sistemas, desarrollo de productos y adquisición de aplicaciones. Parte de las prácticas actuales y logra una mejora continua a través de una cultura de procesos. Su implementación aumenta la fiabilidad del *software* producido, la visibilidad de los procesos de producción y soporte, la reusabilidad de componentes y como resultado de la combinación de este tipo de mejoras, disminuyen los costes de producción y mantenimiento de las aplicaciones (NIELSEN, 2012).

1.8. Metodología de desarrollo

Las metodologías de desarrollo de *software* describen los caminos que se deben seguir para la producción de un determinado producto informático, a través de patrones, que surgieron

Capítulo 1. “Fundamentación teórica de la investigación”

mediante la práctica en un determinado campo del desarrollo de *software*, como es el caso de los patrones de diseño, los cuales sugieren una serie de pasos a la hora de realizar el diseño.

1.8.1. Proceso Unificado Rational (RUP) con Nivel 2 de CMMI

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 (JACOBSON, 2004).

RUP presenta características que son imprescindibles para su funcionamiento, entre las cuales se incluyen: la utilización de un único lenguaje de modelado, utilizando el lenguaje UML, siendo único para el desarrollo de todos los modelos, es un proceso integrado en el que se establece una estructura que integra todas sus facetas. RUP mantiene un desarrollo basado en componentes en el que permite que el sistema se vaya creando a medida que se desarrollan sus componentes. Es guiado por casos de uso, centrado en la arquitectura e Iterativo e Incremental (JACOBSON, 2004).

Para el desarrollo del presente trabajo de diploma se utilizará como metodología de desarrollo RUP con nivel 2 de CMMI pues es la metodología que se utiliza en el proyecto eXcriba. Además, permite tener bien documentado el producto así como reducir riesgos que puedan existir en el desarrollo del mismo.

1.9. Entorno de desarrollo

Para realizar el presente estudio se tiene en cuenta que la solución propuesta va a ser una extensión de la API⁶ de servicios de flujos de trabajo del ECM Alfresco, lo que implica que la

⁶ *Application Programming Interface*

Capítulo 1. “Fundamentación teórica de la investigación”

mayoría de los componentes del entorno de desarrollo⁷ son empleados por el gestor para desarrollar distintas funcionalidades que se relacionan con la propuesta. A continuación se presentan los lenguajes, herramientas y tecnologías que conformarán su entorno de trabajo.

1.9.1. Lenguajes

Lenguaje de modelado Unified Modeling Language (UML)

El Lenguaje Unificado para la Construcción de Modelos se define como un lenguaje gráfico utilizado para especificar, visualizar, construir y documentar cada una de los artefactos que comprende el proceso de desarrollo de un sistema de *software*. Es un sistema notacional (que, entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas de modelado que utilizan conceptos orientados a objetos (LARMAN, 2004).

Se decide utilizar UML para el modelado de los artefactos generados durante el proceso de desarrollo.

PHP

PHP (*acrónimo de PHP: Hypertext Preprocessor*), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas; ofrece la integración con varias bibliotecas externas, que permiten al desarrollador tanto generar documentos en pdf hasta analizar código XML (BAKKEN, 2001).

Es por esa razón que para el desarrollo del módulo se decide utilizar el lenguaje de programación PHP, ya que al poder implementarse con independencia de la plataforma y como *software* libre, permite obtener productos en menor tiempo y por consiguiente, con menores costos, además de que la presente investigación es parte del desarrollo del GDA eXcriba, el cual define al lenguaje PHP para la implementación del sistema.

⁷ Es un programa informático compuesto por un conjunto de herramientas de programación.

Capítulo 1. “Fundamentación teórica de la investigación”

JavaScript

Se decide utilizar **JavaScript** para la implementación de los servicios, ya que Alfresco provee diferentes API de JavaScript entre ellas la API de JavaScript, la de flujos de trabajo que permite acceder y gestionar las definiciones de flujos de trabajo, las instancias, los caminos, las tareas y las transiciones. Dicha API se puede utilizar para realizar cualquiera de las siguientes funcionalidades:

- ◆ crear un paquete de flujos de trabajo,
- ◆ iniciar una nueva instancia de flujo con una definición y asignar un paquete a la misma,
- ◆ cancelar una instancia de flujo de trabajo,
- ◆ eliminar una instancia de flujo de trabajo,
- ◆ finalizar una tarea de flujo de trabajo y señalar la ruta de flujo de trabajo asociado para avanzar al siguiente nodo con una transición.

Java

Java es un lenguaje de programación orientado a objetos que se caracteriza por ser distribuido, enfocado en la seguridad, robusto, de arquitectura neutral y portable. Es un lenguaje de propósito general, y como tal es válido para realizar todo tipo de aplicaciones profesionales (ABRAHAM, 2003).

Se decide utilizar Java para la implementación de los servicios que no pudieron ser desarrollados con la API de JavaScript.

Pero, ¿por qué usar Java para implementar las acciones si la API de JavaScript parece cubrir todas las funcionalidades? El hecho es que el uso de Java tiene sentido en este contexto cuando:

- ◆ el acceso a determinados servicios no está disponible desde la API de JavaScript que provee Alfresco,
- ◆ es necesario interactuar con sistemas cuyas API solamente están expuestas vía Java.

A diferencia de la API de JavaScript, con Java sería posible acceder directamente a las API y servicios que ofrece Alfresco para la manipulación de los diferentes objetos del repositorio.

Capítulo 1. “Fundamentación teórica de la investigación”

1.9.2. Herramientas

Visual Paradigm

Visual Paradigm es una herramienta CASE⁸ (*Computer Aided Software Engineering*) que provee el modelado de procesos de negocio, además de un generador de mapeo de objetos relacionales para los lenguajes de programación Java 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 y se puede generar bases de datos. Este está diseñado para una amplia gama de usuarios, incluidos los ingenieros de *software*, Analistas de Sistemas y Analistas de Negocios. Cualquiera de los cambios que se realicen en el código existente puede reflejarse en el modelo (PRESMAN, 2002).

Para el modelado de los artefactos obtenidos durante el proceso de desarrollo se utilizará Visual Paradigm, ya que es una herramienta multiplataforma para modelado UML, facilita la interoperabilidad⁹ con excelentes herramientas CASE, permite la ingeniería inversa, es decir, invertir modelos UML desde el código fuente, genera diferentes tipos de código fuente y captura requisitos mediante el modelado de los casos de uso.

Zend Studio

Zend Studio es un IDE¹⁰ destinado a desarrolladores profesionales. El mismo es compatible con las plataformas GNU/Linux, Mac y Windows e integrado para el lenguaje de programación PHP.

Permite agilizar el desarrollo web y simplificar proyectos complejos. Presenta características como: excelente completamiento de código, administración avanzada de proyectos, múltiples lenguajes, ofrece soporte básico para otros lenguajes Web, como HTML, JavaScript y XML,

⁸ Ingeniería de *Software* Asistida por Computadora. Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de *software* reduciendo el costo de las mismas desde el punto de vista de tiempo y de dinero.

⁹ Habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

¹⁰ *Integrated Development Environment*

Capítulo 1. “Fundamentación teórica de la investigación”

incorpora el Framework de Zend, PHP Documentor y manual de PHP, integración con el gestor de versiones, los navegadores, integración avanzada con FTP y soporte para Web Services, PHP4, PHP5 y SQL (PÉREZ, 2008).

Se decide utilizar Zend Estudio para la implementación del módulo, ya que el código fuente del eXcrista se distribuye en proyectos de este IDE, lo que facilita que los mismos puedan ser importados en esta herramienta.

1.9.3. Tecnologías

Extensible Markup Language (XML)

XML es un lenguaje de marcado¹¹ para documentos que contengan información estructurada, su especificación define una manera estándar de añadir etiquetas a los documentos (O'REILLY, 2010).

Es una tecnología que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con posibilidades mucho mayores. XML, con todas las tecnologías relacionadas representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes (VALDÉS, 2011).

Se decide utilizar XML para la definición de los procesos y demás archivos que se requiere para la ejecución de los flujos de trabajo, ya que provee múltiples vistas para los datos, en efecto, los documentos XML definen el modelo de datos y luego se puede crear más de una vista basado en las necesidades de la aplicación que está desarrollando.

REST (*Representational State Transfer*)

REST se define como una técnica o estilo arquitectónico de *software* orientado a sistemas de hipertexto distribuido, como la WWW (*World Wide Web*) y a su vez es una abstracción de los elementos arquitectónicos dentro de tales sistemas. El mismo ignora los detalles de

¹¹ Un lenguaje de marcas es un mecanismo para identificar estructuras en un documento.

Capítulo 1. “Fundamentación teórica de la investigación”

implementación de los componentes y la sintaxis del protocolo centrándose en las funciones de los componentes, 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 (FIELDING, 2000).

Este estilo o técnica ampliamente aceptada, se utiliza en el sentido más amplio para describir cualquier interfaz web simple que utilice XML (*Extensible Markup Language*) y HTTP el cual siguiendo los principios de REST, posibilita que al realizar una solicitud petición al servicio este no retorne la base de datos completa, sino un tipo de datos interpretable por el cliente (JSON¹², 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 Services (FIELDING, 2000).

Dado que el *software* GDA eXcriba tiene todos sus servicios implementados siguiendo los principios por los cuales se rige REST, se hace necesario implementar los servicios correspondientes al módulo gestión de los flujos de trabajo de aprobación siguiendo estos principios. 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.

1.9.3.1. Framework

jQuery

jQuery es una biblioteca o framework de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, permitiendo manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a las páginas web (SARRION, 2012).

Se selecciona jQuery en su versión 1.3.2 como framework de JavaScript para el desarrollo de las interfaces de usuarios, además por la facilidad de funcionalidades que ya vienen definidas,

¹² JSON: Notación de Objetos de JavaScript, es un formato de texto para el intercambio de datos.

Capítulo 1. “Fundamentación teórica de la investigación”

simplifica código y facilita la implementación, brinda los efectos y eventos de AJAX¹³, las funcionalidades de DOM¹⁴, y se puede trabajar con CSS¹⁵ y con las etiquetas de HTML o XML, permite el intercambio asíncrono de datos entre cliente y servidor de manera sencilla y su biblioteca ha sido definida en la arquitectura del proyecto.

CodeIgniter

CodeIgniter es un entorno de desarrollo abierto que permite crear aplicaciones web dinámicas con PHP. Su principal objetivo es ayudar a que los desarrolladores, puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero. Este es un framework que facilita la escritura de código repetitivo. Además, de ser totalmente extensible y altamente compatible con gran variedad de versiones y configuraciones de PHP (KRIETE, 2006).

Se decide utilizar CodeIgniter versión 1.7.2 del lado de servidor como marco de trabajo para implementar la lógica del negocio de la solución a desarrollar, ya que este fue definido en la arquitectura del proyecto. Además, es un framework que tiene un buen manejo y abstracción con las bases de datos, posee gran documentación, y ofrece flexibilidad en caso de necesitarse un cambio en algún módulo.

FreeMarker

FreeMarker es un motor de plantillas¹⁶ que posee una biblioteca de clases para los programadores de Java. Está diseñado para la generación de páginas web HTML que siguen al patrón de arquitectura MVC (Modelo Vista Controlador). La idea de usar el patrón MVC para las páginas web dinámicas es que separa los autores de contenido de los programadores. Esta separación es útil incluso para proyectos en los que el programador y el autor de la

¹³ *Asynchronous JavaScript and XML* es una técnica de desarrollo web para crear aplicaciones interactivas.

¹⁴ *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.

¹⁵ Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos.

¹⁶ Una plantilla es un documento que se puede aplicar sobre un objeto de datos para producir otro documento. Así, las plantillas se utilizan para presentar datos o el contenido en diferentes estilos y formatos.

Capítulo 1. “Fundamentación teórica de la investigación”

página HTML son la misma persona, ya que ayuda a mantener la aplicación clara y fácil de mantener (PATEL, 2012).

El motor de plantillas FreeMarker se inserta dentro de Alfresco y toma el modelo de datos de Alfresco como entrada y genera texto (HTML o XML) como salida. Los programadores crean la plantilla de presentación con hojas de estilo y código HTML para luego ser generado por el motor FreeMarker integrado en el Alfresco como una página web.

Se decide utilizar FreeMarker para aportar practicidad en la generación de páginas web XML o HTML, ya que es mucho más fácil de aprender y usar, es legible, rápido y sus plantillas son detalladas y mucho más mantenibles.

Webscript

Un **Webscript** es simplemente una URI¹⁷ unido a un servicio utilizando los métodos estándar de HTTP, tales como: *GET*, *POST*, *PUT* o *DELETE*. Alfresco Webscript fue introducido oficialmente en el año 2006 como parte de la arquitectura del sistema Alfresco y desde entonces ha obtenido popularidad entre los desarrolladores e integradores de sistemas que usan Alfresco como Gestor de Contenido Empresarial, implementando nuevos servicios sobre su repositorio. Tanto es así que las últimas implementaciones y proyectos de Alfresco, entre ellas Surf, Share, Web Studio y los servicios CMIS, han sido desarrollados usando webscripts. Alfresco Webscripts implementa la arquitectura Modelo Vista Controlador (Ilustración 3).

Se decide utilizar Webscript para exponer el contenido del repositorio de Alfresco a través de una API RESTful con el objetivo de proveer al repositorio de servicios de contenidos que serán accesibles para todos los usuarios, consultar en el repositorio, extraer contenido y alterar su comportamiento, exponer el repositorio para realizar gestión documental o gestión de contenidos web y proveer de facilidades de búsqueda personalizadas.

¹⁷ *Uniform Resource Identifie*, es una cadena de caracteres corta que identifica inequívocamente un recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia). Normalmente, estos recursos son accesibles en una red o sistema.

Capítulo 1. “Fundamentación teórica de la investigación”

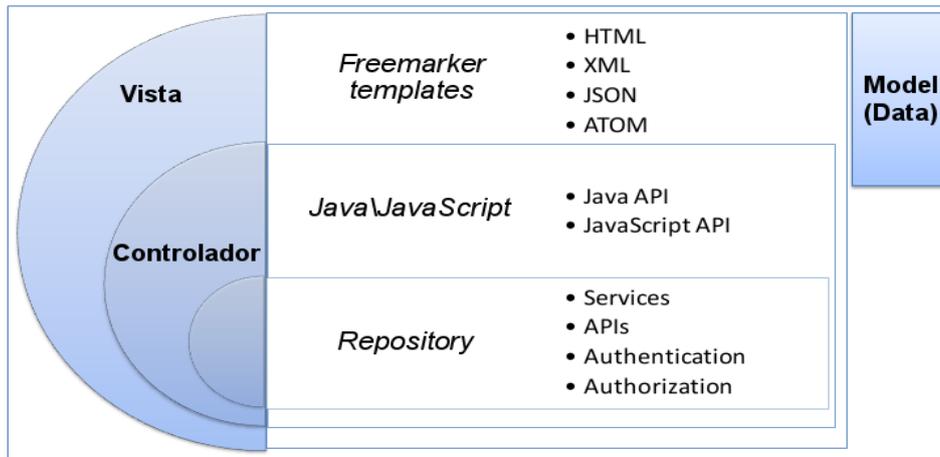


Ilustración 3. Arquitectura MVC de Alfresco Webscripts.

1.10. Conclusiones del capítulo

Como resultado de la investigación en el capítulo se ha expuesto un marco conceptual e informativo que permite tener un mayor entendimiento del contexto donde se desarrolla la propuesta de solución. El análisis de los sistemas de gestión de documental ha sido de gran importancia, porque permite reconocer diferentes funcionalidades de los sistemas que son utilizadas posteriormente como guía para la captura de requerimientos de la propuesta de solución. Los lenguajes, herramientas y tecnologías que se utilizan durante el proceso de desarrollo han permitido conformar un marco de trabajo que reúne las características necesarias, quedando sentadas las bases así, para el desarrollo de la propuesta de solución.

Capítulo 2. “Descripción y análisis de la solución propuesta”

Capítulo 2. “Descripción y análisis de la solución propuesta.”

Con el objetivo de responder a las necesidades planteadas por los clientes y garantizar que la solución propuesta funcione correctamente en el entorno del usuario, se necesita crear una aproximación al negocio y obtener un análisis del mismo. Es fundamental capturar los requerimientos correctos y utilizarlos para conducir el proceso de desarrollo de la solución.

El presente capítulo se centra en la valoración crítica del diseño propuesto por el analista, donde se estudiarán los cambios necesarios para la transición del diseño a la implementación, teniendo en cuenta los requerimientos de la aplicación. Se abordarán cuestiones referentes, entre otras, a la arquitectura, validación de requisitos y especificación de casos de uso. Para finalmente describir las nuevas clases y operaciones necesarias para darle solución a la situación problemática.

2.1. Especificación de requisitos

Los requisitos son una descripción de las necesidades o deseos de un producto. Para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas reflejando las necesidades de los clientes. (SOMMERVILLE, 2005).

Los requisitos funcionales y no funcionales propuestos por el analista coinciden con los requerimientos necesarios para llevar a cabo la implementación del módulo, por ende, se hacen uso en su totalidad. A continuación se muestran:

Requisitos funcionales			
Nº	Nombre	Descripción	Prioridad para el cliente
RF1	Iniciar flujo de trabajo.	El sistema debe permitir iniciar flujos de trabajo según la aprobación que se desee realizar, en serie o en paralelo.	Alta

Capítulo 2. “Descripción y análisis de la solución propuesta”

RF2	Crear petición de aprobación.	El sistema debe permitir crear peticiones de aprobación según el tipo de aprobación que requiera el documento, en serie o en paralelo.	Alta
RF3	Enviar petición de aprobación con el documento asociado.	El sistema debe permitir enviar al destinatario una petición de aprobación.	Alta
RF4	Aceptar petición de aprobación.	El sistema debe permitir a los destinatarios aceptar peticiones de aprobación.	Alta
RF5	Rechazar petición de aprobación.	El sistema debe permitir rechazar peticiones de aprobación, esto ocurre cuando el destinatario decide no aprobar el documento.	Alta
RF6	Cancelar una petición de aprobación.	El sistema debe permitir cancelar peticiones de aprobación.	Alta
RF7	Añadir un documento a la petición.	El sistema debe permitir elegir un documento para que sea añadido a la petición.	Alta
RF8	Listar petición de aprobación	El sistema debe permitir listar todas las peticiones asignadas a los destinatarios.	Alta
RF9	Visualizar el estado de las peticiones realizadas.	El sistema debe permitir mostrar el estado de una petición de aprobación (En espera, En Progreso, Completado,	Media

Capítulo 2. “Descripción y análisis de la solución propuesta”

		Cancelado y Caduco).	
RF10	Mostrar el documento asociado a una petición.	El sistema debe permitir mostrar el documento asociado a una petición	Media
RF11	Listar personas involucradas con la petición.	El sistema debe permitir listar las personas que están involucradas con la petición.	Media
RF12	Consultar el documento asociado a una petición.	El sistema debe permitir consultar el documento asociado a una petición para ser revisado.	Alta
RF13	Finalizar el flujo de trabajo.	El sistema debe permitir finalizar los flujos de trabajo inicializados.	Alta

Tabla 1. Requisitos funcionales.

Requisitos no funcionales	
Usabilidad	
1.	Facilidad de uso por parte de los usuarios: el sistema debe presentar una interfaz de manejo cómodo que posibilite a los usuarios una rápida adaptación. Manejo
2.	Especificación de la terminología utilizada: el sistema debe adaptarse al lenguaje y términos utilizados por los usuarios con vistas a una mayor comprensión por parte del cliente de la herramienta de trabajo.
3.	El sistema podrá ser usado por cualquier persona que posea permisos en la entidad para gestionar los flujos de trabajo.
Restricciones del Diseño	
4.	La herramienta de modelado utilizada será Visual Paradigm.
5.	Se empleará como lenguaje de modelado UML.

Capítulo 2. “Descripción y análisis de la solución propuesta”

6.	Los lenguajes de desarrollo serán Java, JavaScript y PHP.
7.	Las tecnologías de desarrollo serán jQuery en versión 1.3.2 y CodeIgniter en versión 1.7.2.
	Interfaz
8.	Interfaz Web: La interfaz es sencilla con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo.

Tabla 2. Requisitos no funcionales.

2.2. Validación de los requisitos

La validación de requisitos tiene como objetivo demostrar que estos realmente definen el sistema que el cliente desea. Responde a la pregunta: “¿Estamos construyendo el producto correctamente?” (SOMMERVILLE, 2005). Es el proceso mediante el cual se verifican los requerimientos en cuanto a validez, consistencia, integridad y realismo.

El proceso de validación de requisitos comprende actividades que generalmente se realizan una vez obtenida una primera versión de la documentación de requisitos.

Este proceso tiene por finalidad comprobar que los requisitos del *software* poseen todos los atributos de calidad, en breve: son consistentes, completos, precisos, realistas, verificables y definen lo que el usuario desea del producto final. La realización de estas actividades en este momento pretende evitar los altos costos que significaría el tener que corregir una vez avanzado el desarrollo.

La actividad de validación tiene como entrada el documento de requisitos, los estándares relacionados y el conocimiento de la organización, y como salida se obtiene una lista de problemas y una lista de acciones recomendadas (Ilustración 4).

Capítulo 2. “Descripción y análisis de la solución propuesta”

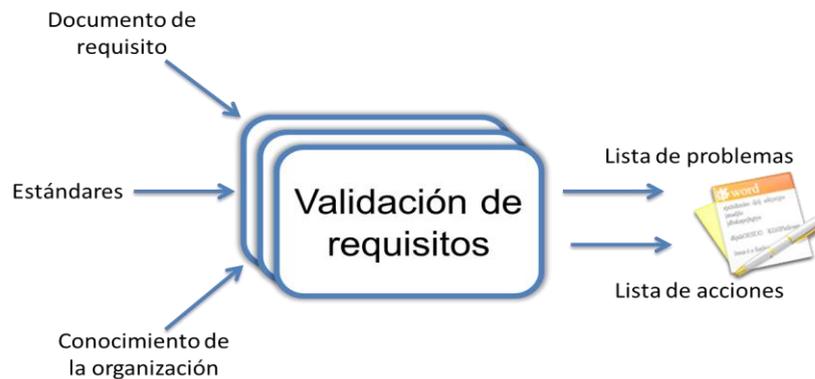


Ilustración 4. La validación en el proceso de requisitos

La validación se realiza a través de diversos métodos. Los dos métodos más habituales son:

Revisiones de requerimientos

Las revisiones de requisitos consisten en reuniones donde un equipo de analistas intenta localizar errores en el documento de especificación (CHAVES, 2005).

Durante la **revisión de los requerimientos** se realizaron verificaciones en los documentos relacionados con las especificaciones de los requisitos para comprobar la integridad, consistencia, verificabilidad y adaptabilidad de los mismos. Durante el desarrollo de la propuesta de solución se efectuaron reuniones entre varios miembros del proyecto fundamentalmente analistas y se analizaron los documentos de especificación de requisitos. Como resultado se encontraron los siguientes errores:

Nº del requisito	Defectos detectados	Acciones recomendadas
RNF7	Omisión del lenguaje de desarrollo PHP.	Agregar a las restricciones del diseño el lenguaje de PHP.
RNF8	Omisión de las principales tecnologías a utilizar.	Agregar a las restricciones del diseño las tecnologías jQuery y CodeIgniter.

Tabla 3. Lista de errores y acciones recomendadas.

Capítulo 2. “Descripción y análisis de la solución propuesta”

Construcción de prototipos

La técnica de prototipado consiste en construir una maqueta del futuro sistema a partir de los requisitos recogidos en la especificación. Esta maqueta será evaluada por usuarios para comprobar su corrección y completitud. La construcción de prototipos puede evitar sorpresas al detectar los requerimientos incompletos o inconsistentes y la falta de funcionalidad en un sistema (CHAVES, 2005).

Para la evaluación de los **prototipos** construidos por el analista se seleccionó un grupo de 4 especialistas con distintos perfiles para validar sus funcionalidades. El prototipo seleccionado corresponde al RF 2 (Crear petición de aprobación) que presenta una alta complejidad y permite además validar los requerimientos funcionales:

- ◆ Añadir el documento a la petición,
- ◆ Mostrar el documento asociado a una petición,
- ◆ Listar personas involucradas con la petición,
- ◆ Consultar el documento asociado a una petición.

Tras un profundo razonamiento se comprobó que el prototipo seleccionado se corresponde con las especificaciones para el cual fue diseñado, permitiendo realizar así la actividad de crear petición. A continuación se muestra:

Capítulo 2. “Descripción y análisis de la solución propuesta”

La imagen muestra un prototipo de una interfaz de usuario para crear una nueva petición. El título de la ventana es "Nueva petición".

En la parte superior, hay un campo de texto "Seleccionar documento:" con el valor "/medi/Datos/Tesis.pdf" y un botón "Examinar" con un ícono de lupa.

Debajo, se indica el "Tipo de firma:" con dos opciones: "Firma en serie" (radio desactivado) y "Firma en paralelo" (radio activado).

El área principal se titula "Seleccionar destinatarios" y está dividida en dos paneles: "Usuarios:" y "Destinatarios:".

- El panel "Usuarios:" contiene una lista de nombres: Pedro Rodríguez, Pedro Rodríguez, Pedro Antonio Rodríguez y Pedro Rosabales. El primer "Pedro Rodríguez" está seleccionado.
- El panel "Destinatarios:" contiene una lista de nombres: Reinier Elejarde, Julio A. Zamora y Pedro Rodríguez. El "Pedro Rodríguez" está seleccionado.
- Entre los paneles hay botones de navegación: << y >>.

Debajo de los paneles, se encuentran los campos "Prioridad:" (con un menú desplegable que muestra "Baja") y "Fecha de caducidad:" (con un calendario que muestra "13/02/2012").

En la parte inferior, hay un campo de texto "Descripción:" con el contenido "Aprobación de los objetivos ...". A la derecha de este campo hay un menú desplegable con tres íconos (una flecha hacia arriba, un menú y una flecha hacia abajo).

Finalmente, en la parte inferior derecha, hay dos botones: "Crear petición" y "Cancelar".

Ilustración 5. Prototipo crear petición de aprobación.

No obstante, se deciden cambiar los prototipos propuestos ya que se quiere familiarizar más al usuario con la interfaz gráfica del eXcriba como se muestra a continuación:

Capítulo 2. “Descripción y análisis de la solución propuesta”

The screenshot shows the 'Crear petición de Aprobación' (Create Approval Request) form in the eXcriba system. The interface includes a header with the eXcriba logo and 'Gestor de Documentos Administrativos', a search bar, and navigation links for 'Accesos Directos', 'Entrantes', 'Salientes', and 'Notificaciones(0)'. The form is divided into several sections:

- Opciones Generales:** Contains fields for 'Usuarios (*)' (with 'ysibat' selected), 'Acciones (*)' (with '>>' and '<<' buttons), and 'Destinatarios (*)' (with 'admin' selected).
- Fecha de caducidad (*):** A date field containing '04/26/2013'.
- Tipo de aprobación (*):** Radio buttons for 'Aprobación en serie' (selected) and 'Aprobación en paralelo'.
- Prioridad:** A dropdown menu showing the value '2'.
- Descripción (*):** A text area containing 'Aprobación de los objetivos'.
- Acciones (*):** 'Enviar' and 'Limpiar' buttons.

Ilustración 6. Prototipo crear petición de aprobación

Ver Anexo A de la versión extendida del documento para ver los restantes prototipos.

2.3. Especificación de Casos de Uso

2.3.1. Definición de los actores

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

Tabla 4. Definición de los actores

Capítulo 2. “Descripción y análisis de la solución propuesta”

2.3.2. Análisis del diagrama de caso de uso

Los diagramas de caso de uso se suelen utilizar en el modelado del sistema desde el punto de vista de sus usuarios, para representar las acciones que realiza cada tipo de usuario.

En el estudio realizado al diagrama de casos de uso propuesto por el analista, se identificó que la funcionalidad buscar peticiones no se encuentra dentro de los requisitos funcionales propuesto con anterioridad, pero sin restarle importancia, se realiza una consulta a los especialistas del proyecto, principalmente analistas, con el objetivo de conocer si es necesario para la implementación del módulo; y se concluye que el mismo no es relevante, por lo tanto, se corrige, quedando como resultado:

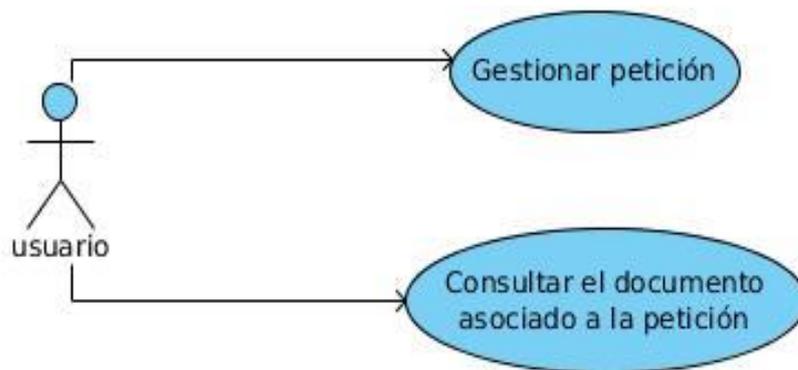


Ilustración 7. Diagrama de Casos de Uso

2.3.3. Descripción de los casos de uso

A raíz de lo anteriormente expuesto, se elimina de la descripción de los casos de uso buscar petición.

Caso de uso	Gestionar petición
Actor	Usuario (inicia) crea, envía, lista, acepta, rechaza, o cancela las peticiones realizadas por un usuario.
Resumen	El caso de uso comienza cuando el usuario desea gestionar una petición. Para ello puede crear, enviar, listar, aceptar, rechazar o cancelar la misma.

Capítulo 2. “Descripción y análisis de la solución propuesta”

Prioridad	Alta	
Complejidad	Alta	
Precondiciones	El cliente ha sido validado y ha accedido a la sesión InBox.	
Poscondiciones	Las peticiones fueron gestionadas.	
Flujo de eventos		
Flujo básico <Gestionar petición>		
Sección 1: “Crear petición”		
Actor	Sistema	
1. Selecciona la opción () “Crear petición de aprobación” 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 una petición de aprobación.	
3. Inserta los datos solicitados.	4. Verifica que los campos obligatorios estén llenos.	
	5. En caso correcto almacena en el repositorio la petición creada y muestra un mensaje de confirmación.	
Flujos alternos		
Actor	Sistema	
3.1. Introduce los datos solicitados en los campos correspondientes.	Identifica datos erróneos e informa de que los datos introducidos no son correctos.	
3.2. Introduce alguno de los datos solicitados (no todos) en los campos correspondientes.	Informa de que faltan campos por llenar.	
Sección 2: “Enviar petición”		
1. El usuario selecciona el botón Enviar	2. Envía la petición al destinatario	

Capítulo 2. “Descripción y análisis de la solución propuesta”

petición.	seleccionado.
Sección 3: “Aceptar petición”	
1. El usuario accede a la bandeja de tareas de su espacio de trabajo.	2. Muestra en la bandeja de tareas las peticiones que le han sido asignadas.
3. Selecciona de la bandeja de tareas la acción  “Mostrar petición”.	4. Muestra en una nueva ventana el formulario con las propiedades de la petición.
5. Introduce los datos solicitados (destinatario, estado y comentario) y selecciona el botón Aceptar petición.	6. Envía la notificación al remitente de que el flujo de trabajo ha sido aceptado. Termina el caso de uso.
Sección 4: “Rechazar petición”	
1. El usuario accede a la bandeja de tareas de su espacio de trabajo.	2. Muestra en la bandeja de tareas las peticiones que le han sido asignadas.
3. Selecciona de la bandeja de tareas la acción  “Mostrar petición”.	4. Muestra en una nueva ventana el formulario con las propiedades de la petición.
5. Introduce los datos solicitados (destinatario, estado y comentario) y selecciona el botón Rechazar petición.	6. Envía la petición al remitente para su valoración.
Sección 5: “Cancelar petición”	
1. El usuario accede a la bandeja de tareas de su espacio de trabajo.	2. Muestra en la bandeja de tareas las peticiones que le han sido asignadas.
3. Selecciona de la bandeja de tareas la acción  “Mostrar petición”.	4. Muestra en una nueva ventana el formulario con las propiedades de la petición.

Capítulo 2. “Descripción y análisis de la solución propuesta”

5. Selecciona el botón Cancelar petición.	6. Termina el caso de uso.
Sección 6: “Listar petición”	
1. Selecciona en la barra de funcionalidad la opción  Bandeja de peticiones que tiene pendiente el usuario.	2. Muestra una interfaz con el listado de las peticiones que tiene pendiente el usuario. Termina el caso de uso.

Tabla 5. Descripción textual del caso de uso: Gestionar petición.

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

2.4. Diseño del módulo

Para comprender el sistema, organizar el desarrollo y fomentar la reutilización del mismo, se necesita una arquitectura correctamente descrita. A continuación se expone la propuesta de solución a través de los diagramas de clases e interacción del diseño, así como la descripción de la arquitectura y de los patrones de diseño utilizados.

2.4.1. Patrón de arquitectura.

“La arquitectura de un sistema es un marco conceptual completo que describe su forma y estructura (sus componentes y la manera en que se integran)” (PRESSMAN, 2002).

Buschmann plantea que los patrones arquitectónicos expresan el esquema de organización estructural fundamental para sistemas de *software*; provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos. La selección de un patrón arquitectónico es, por lo tanto, una decisión fundamental de diseño en el desarrollo de un sistema de *software* (BUSCHMANN, 1996).

Para el desarrollo de la solución se utilizará la arquitectura en capas, ya que simplifica la comprensión y organización del desarrollo del sistema y reduce las dependencias. La

Capítulo 2. “Descripción y análisis de la solución propuesta”

arquitectura propuesta añade una gran flexibilidad al diseño de la aplicación, así como una interoperabilidad en entornos distribuidos con un nivel de abstracción superior.

Las tres capas que se definieron para la arquitectura del módulo son: Presentación, Aplicación y Acceso a Repositorio.

Capítulo 2. “Descripción y análisis de la solución propuesta”

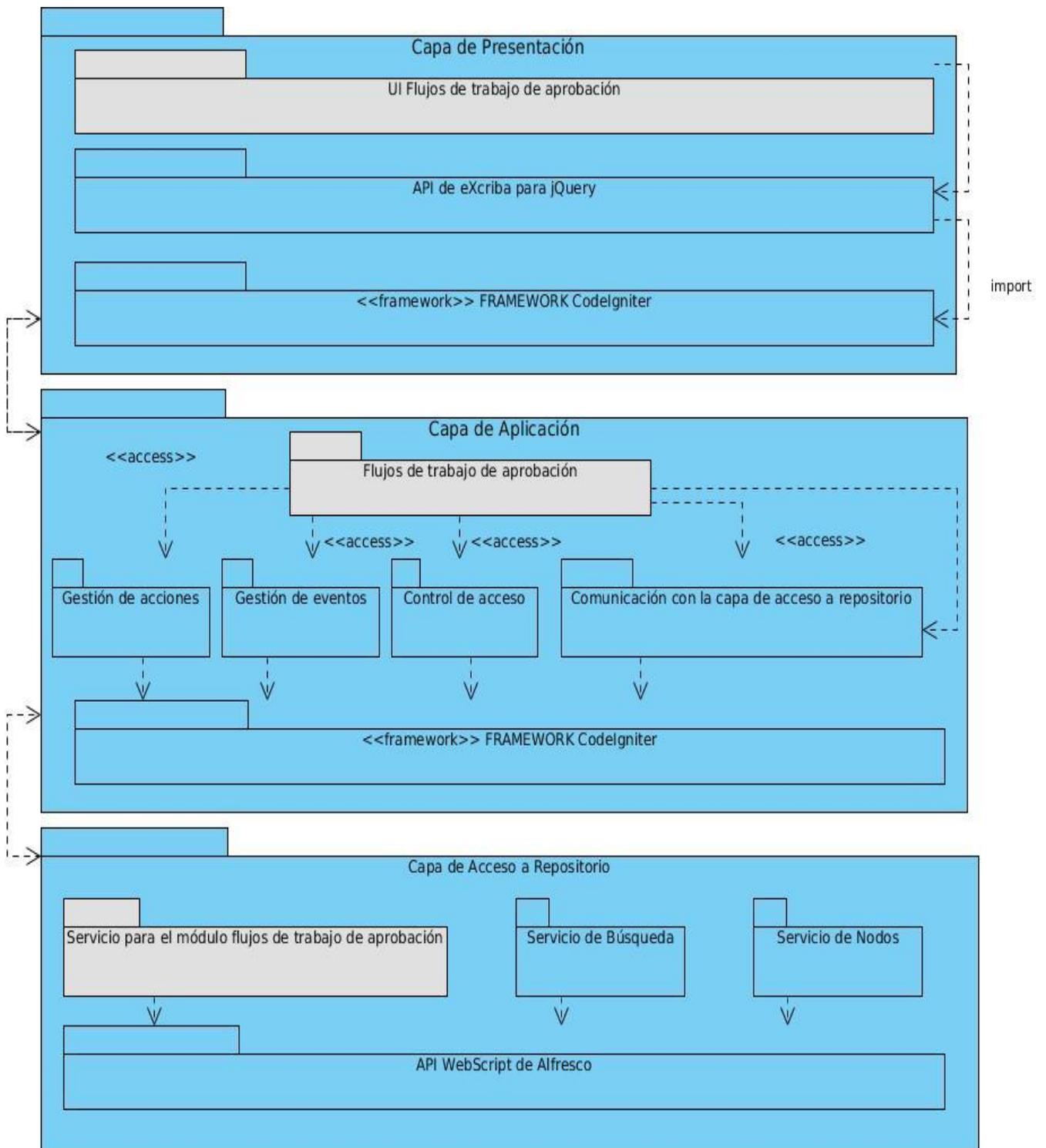


Ilustración 8. Arquitectura de la propuesta de solución

Capítulo 2. “Descripción y análisis de la solución propuesta”

Capa de Presentación: En esta capa se encuentra el conjunto de interfaces de usuario, que les hace posible al cliente y la aplicación establecer la comunicación, manipular los datos, así como representar en función de componentes visuales, toda la información necesaria, consultada y/o generada por el par aplicación - usuario.

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. Esta capa se comunica con la capa de acceso a repositorio mediante un subsistema de servicios, el cual es el encargado de realizar las llamadas a los servicios.

Capa de Acceso a Repositorio: En esta capa es donde se realiza la implementación de los servicios, los cuales son necesarios para gestionar los datos del repositorio.

2.4.2. Patrones de flujos de trabajo

Los patrones de flujo básico y estructurales definidos por el analista para representar los flujos de trabajo, son reutilizados en la presente investigación debido a que estos patrones no solo están orientados a la representación de los flujos de trabajo, sino también al modo en que estos se ejecutan. A continuación se describen:

2.4.2.1. Patrones de flujo básico

Los patrones de flujo básico son aquellos que tratan aspectos básicos de los procesos de control. Seguidamente se presentan los patrones usados para definir los flujos de trabajo que forman parte de la propuesta de solución.

Secuencia

El patrón de secuencia describe el proceso más simple dentro de un flujo de control. Este patrón expresa el proceso en el que una actividad se inicia después de haberse completado otra en el mismo proceso (Ilustración 9) (FERNÁNDEZ, 2009).

Capítulo 2. “Descripción y análisis de la solución propuesta”



Ilustración 9. Patrón de secuencia

Elección exclusiva

El patrón de elección exclusiva permite a los flujos de trabajo realizar decisiones a la hora de ejecutar actividades en función de las ramas escogidas (FERNÁNDEZ, 2009). En la Ilustración 10 se puede observar un ejemplo de cómo es utilizado el patrón en el flujo de trabajo “En paralelo”, donde se rechaza o se firma el documento.

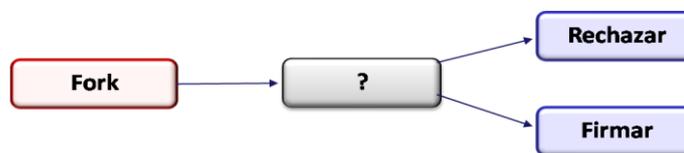


Ilustración 10. Patrón de separación paralela

Sincronización

El patrón de sincronización viene asociado a procesos que tienen en ejecución múltiples actividades ejecutadas en paralelo. En algunos casos es necesario que un subconjunto de las actividades que se están ejecutando en paralelo termine antes de seguir ejecutando el flujo de trabajo. El elemento sincronizador, esperará a que todas las actividades terminen y continuará el proceso con la siguiente o las siguientes actividades (FERNÁNDEZ, 2009). En la Ilustración 11 se evidencia un ejemplo del patrón de sincronización empleado en la representación del flujo de trabajo “En paralelo”.

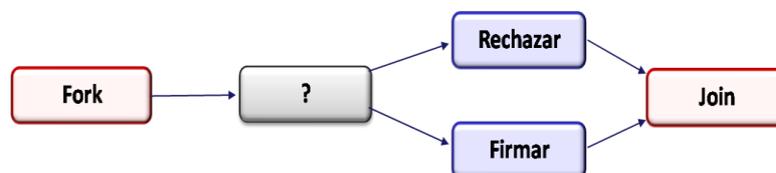


Ilustración 11. Patrón de sincronización

Capítulo 2. “Descripción y análisis de la solución propuesta”

2.4.2.2. Patrones estructurales

Los patrones estructurales son aquellos que tratan el comportamiento repetitivo de un flujo de trabajo (FERNÁNDEZ, 2009). Seguidamente se presenta el patrón estructural utilizado en el flujo de trabajo “En serie” (Ilustración 12).

Patrón de ciclos arbitrarios

Un ciclo arbitrario es un punto donde un conjunto de una o más actividades pueden ser ejecutadas repetidamente (FERNÁNDEZ, 2009).

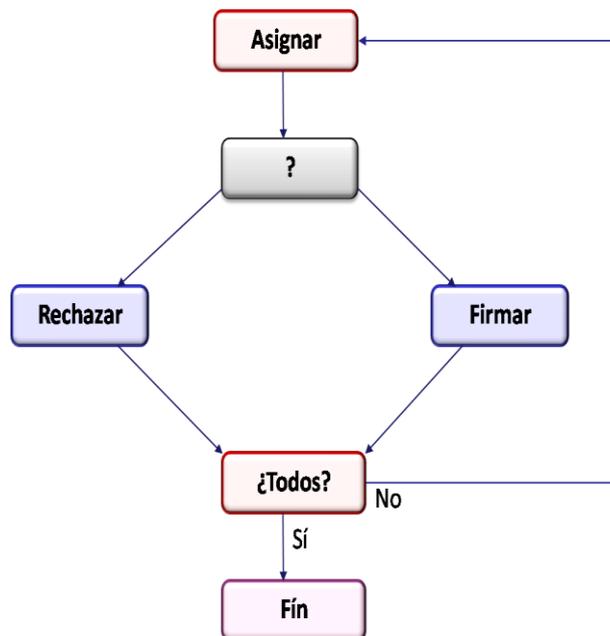


Ilustración 12 Patrón de ciclos arbitrarios

Capítulo 2. “Descripción y análisis de la solución propuesta”

2.4.3. Diagrama de clases del diseño

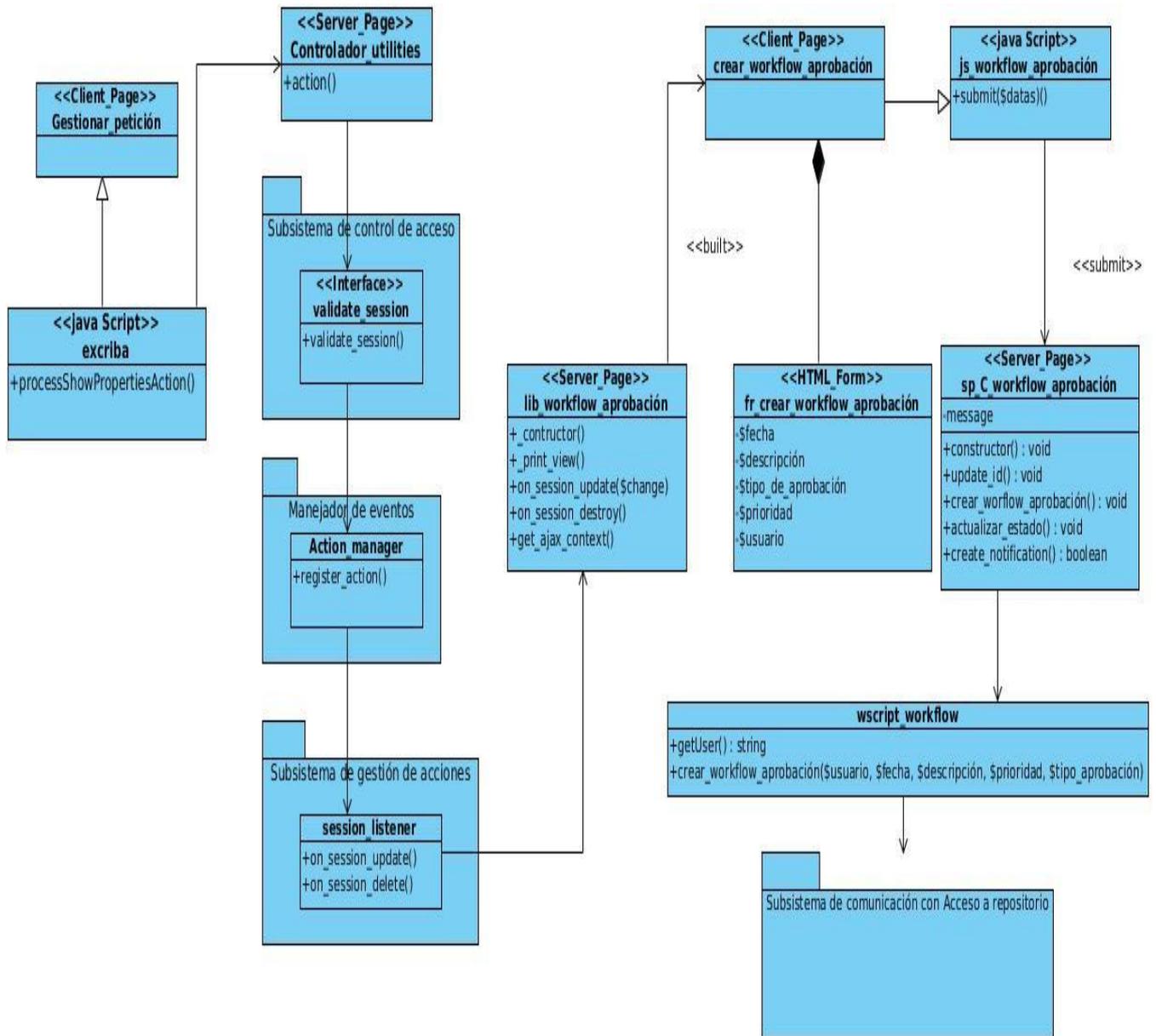


Ilustración 13. Diagrama de clases del diseño: Crear petición

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

Capítulo 2. “Descripción y análisis de la solución propuesta”

2.4.3.1. Descripción de las clases

Descripción del Diagrama de Clases del Diseño del CU Gestionar petición.

Nombre: C_workflow_aprobación	
Tipo de clase: Controladora	
Atributo	Tipo
\$messages	
Para cada responsabilidad:	
Nombre:	update_id ()
Descripción:	Actualiza el ID después de haber realizado la acción Regresar
Nombre:	crear_workflow_aprobación ()
Descripción:	Crea los flujos de trabajo
Nombre:	actualizar estado ()
Descripción:	Modifica las propiedades estado y comentario de un flujo de trabajo
Nombre:	create_notification(\$user)
Descripción:	Manda la notificación al usuario luego de haber creado el flujo de trabajo

Nombre: workflow_aprobación	
Tipo de clase: Interfaz	
Atributo:	Tipo:
\$usuario	
\$fecha	

Capítulo 2. “Descripción y análisis de la solución propuesta”

\$descripción	
\$prioridad	
\$tipo_de_aprobación	

Para cada responsabilidad:

Nombre: workflow_aprobación	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
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

Nombre: actualizar_estado	
Tipo de clase: Interfaz	
Atributo:	Tipo:
\$estado	
\$comentario	

Capítulo 2. “Descripción y análisis de la solución propuesta”

Para cada responsabilidad

Nombre: basic_operation	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
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
Nombre:	get_ajax_context ()
Descripción:	Es el encargado de incluir los ficheros JavaScript

2.4.4. Estados de los flujos de trabajo

Es necesario describir una serie de estados para definir el punto en el que se encuentra la ejecución de un flujo de trabajo. Dado que en la ejecución de una petición siempre hay dos o más usuarios implicados el estado de una petición respecto a cada uno de los usuarios es distinto. Por ello se definen los siguientes estados:

- ◆ **En espera:** define una petición que ha sido enviada por el usuario y que está pendiente de ser firmada o rechazada por los destinatarios,

Capítulo 2. “Descripción y análisis de la solución propuesta”

- ◆ **En progreso:** petición cuya ejecución esté en curso,
- ◆ **Completado:** define una petición cuya ejecución ha finalizado satisfactoriamente¹⁸,
- ◆ **Cancelado:** el usuario que ha creado una petición tiene la posibilidad de cancelarla si ya no le interesa que el documento sea firmado,
- ◆ **Caduco:** para cada flujo de trabajo se define una fecha de caducidad, de modo que cuando esta expira el estado del flujo pasa a ser caduco.

A continuación se distinguen los estados que puede visualizar el usuario dependiendo de su rol dentro del flujo de trabajo (remitente o destinatario):

Estados	Remitente	Destinatario
En espera	X	
En progreso	X	X
Completado	X	X
Cancelado	X	X
Caduco	X	X

Tabla 6. Descripción de los estados de los flujos de trabajo en serie y en paralelo

¹⁸ Que un flujo de trabajo haya finalizado satisfactoriamente no tiene por qué significar que todos los destinatarios hayan firmado el documento, puede que lo hayan rechazado.

Capítulo 2. “Descripción y análisis de la solución propuesta”

2.4.5. Diagrama de secuencia del diseño.

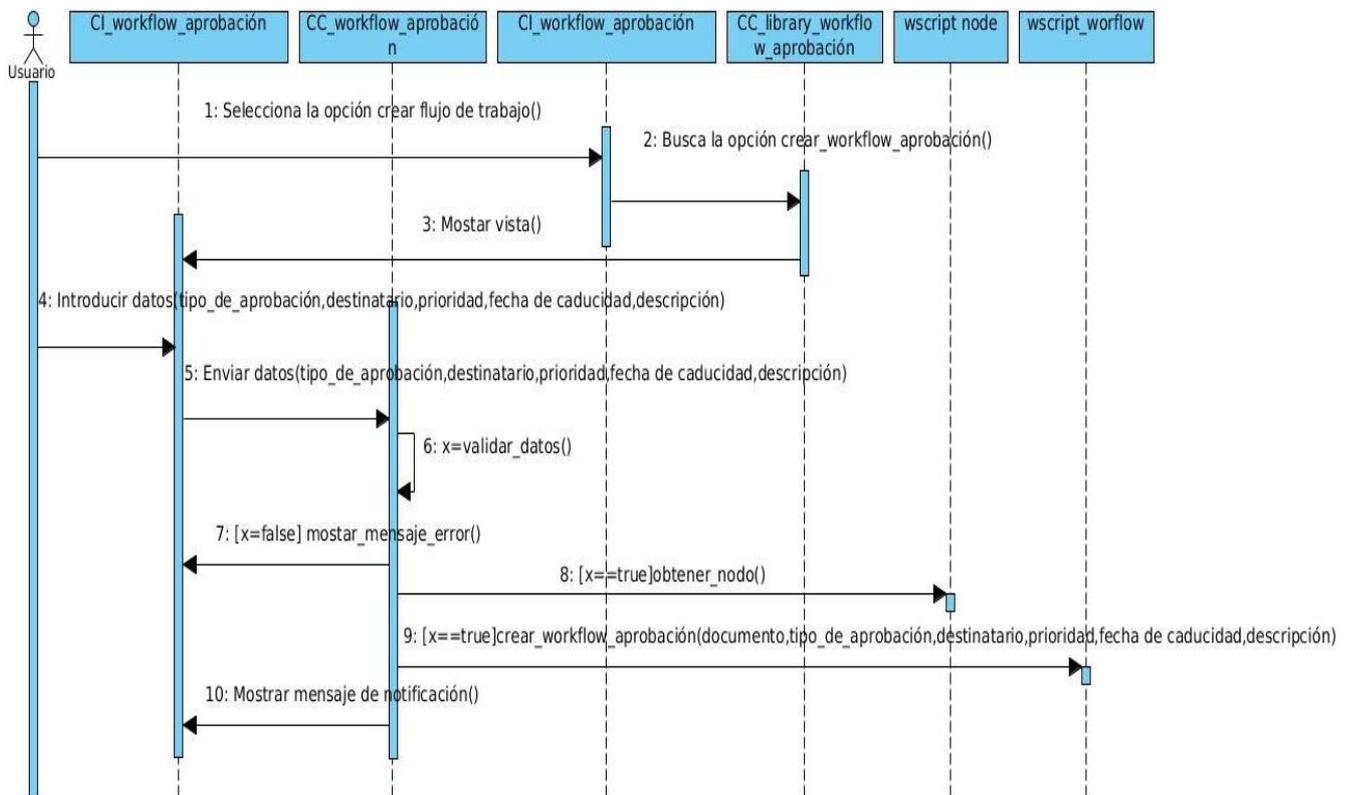


Ilustración 14. Diagrama de secuencia del caso de uso: Gestionar petición

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

2.5. Conclusiones del capítulo

Después de realizar una valoración crítica del diseño propuesto por el analista, se ha logrado arribar a un diseño acorde con las restricciones del sistema, que deben tenerse en consideración a la hora de implementarlo. A dicha propuesta se le realizó un conjunto de cambios adaptativos, estos cambios fueron globales ya que la estructura básica, díganse los diagramas (clases del diseño, diagramas de secuencia y diagrama de caso de uso), la descripción de los casos de uso, la arquitectura y los prototipos funcionales cambiaron radicalmente, todo con el objetivo de implementar un sistema que responda a la problemática planteada.

Capítulo 3. “Implementación y prueba del módulo”

Capítulo 3. “Implementación y prueba del módulo.”

En el presente capítulo quedará definido cómo será la implementación del sistema, los servicios, la representará gráficamente de los diagramas de despliegue y de componentes, así como las pruebas que se le deben realizar al producto para garantizar su correcto funcionamiento durante todo su ciclo de vida.

3.1. Descripción de los servicios

Para el desarrollo de las funcionalidades es necesario implementar algunos servicios en Alfresco, debido a que la API de JavaScript que provee, no expone los servicios para ejecutar los flujos de trabajo del proceso de aprobación formal de documentos electrónicos desde aplicaciones externas que utilizan Alfresco como repositorio de servicios y contenidos. Por ello se definen los siguientes servicios:

Servicio: Crear flujo de trabajo		
Paquete: /cu/uci/excriba/workflow	Plantillas de Respuestas: JSON,XML,HTML	
Descripción: Permite iniciar los flujos de trabajo de aprobación.		
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_cadu={date_cadu}	Método HTTP: GET	Ruta Relativa: alfresco/service/cu/uci/workflow-instance-create?description={description}&assign={assign}&date_cadu={date_cadu}
Dirección del documento de descripción:		

Capítulo 3. “Implementación y prueba del módulo”

classpath:alfresco/extension/templates/webscripts/cu/uci/excriba/workflow/create-workflow-instance.get.desc.xml
Lenguaje de implementación: JavaScript

Tabla 7. Descripción del servicio: Crear flujo de trabajo

Servicio: Mostrar historial del flujo de trabajo		
Paquete: /cu/uci/excriba/workflow	Plantillas de Respuestas: JSON,XML,HTML	
Descripción: Permite conocer el historial del flujo de trabajo, donde se mostrará una descripción con los datos del flujo y el estado en que encuentra.		
Requerimiento de autenticación: Usuario	Requerimiento de Transacción: Requerida	
Respuesta por defecto:		
URI Absoluta: http://<nombre_servidor>[:<puerto>]/alfresco/service/cu/uci/Mostrar_historial_workflow	Método HTTP: GET	Ruta Relativa: alfresco/service/cu/uci/Mostrar_historial_workflow
Dirección del documento de descripción: classpath:alfresco/extension/templates/webscripts/cu/uci/excriba/workflow/Mostrar_historial_workflow.get.desc.xml		
Lenguaje de implementación: JavaScript		

Tabla 8. Descripción del servicio: Mostrar historial del flujo de trabajo

Para ver la descripción de los servicios restantes ver Anexo E de la versión extendida de este documento.

Capítulo 3. “Implementación y prueba del módulo”

Los servicios definidos con anterioridad cubrirán todos los requerimientos funcionales establecidos para el sistema y permitirán ejecutar los flujos de trabajo desde aplicaciones externas.

3.2. Estándar de codificación

El propósito fundamental de los estándares de codificación es lograr que el módulo propuesto tenga una arquitectura y un estilo consistente, independiente del autor, con lo cual el sistema resulte fácil de entender y por supuesto fácil de mantener. El mayor problema que trata de enfrentar esta práctica, como fue mencionado al principio, es el tratar de entender el formato y el estilo utilizado en el código escrito por otros desarrolladores.

Los estándares de codificación son un complemento a la programación por pares y no solo es importante tener un estándar, sino tener un buen estándar de codificación, de esta forma, un buen estándar de codificación deberá:

- ◆ Clarificar más que confundir,
- ◆ Promover la intención del código,
- ◆ Permitir que los programas se acerquen lo mejor posible al lenguaje natural,
- ◆ Incorporar las mejores prácticas de la codificación.

A continuación se muestran algunos de los ejemplos utilizados en la implementación del módulo:

◆ Clase y forma de denominación

Los nombres de las clases deben tener siempre la primera letra mayúscula, y el método constructor debe coincidir exactamente. Varias palabras deben estar separadas por un guión y no camelCased. Todos los otros métodos de la clase deben ser enteramente en minúsculas y el nombre que indique claramente su función, que incluye preferentemente un verbo.

Ejemplo:

```
class Super_class {  
function Super_class()
```

Capítulo 3. “Implementación y prueba del módulo”

```
{  
}  
}
```

◆ Nombres de variables

Las variables deben contener solo letras minúsculas, use subrayado separador en caso de ser nombres compuestos, y se llamarán razonable para indicar su propósito y contenido. Las variables muy cortas, no de palabras, sólo deben utilizarse como iteradores en bucles for ().

Ejemplo:

```
for ($j = 0; $j < 10; $j++)
```

```
$str
```

```
$buffer
```

```
$group_id
```

```
$last_city
```

◆ Estructuras de control

Se incluye if, for, foreach, while, switch, entre las estructuras de control y los paréntesis debe de existir un espacio. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales. Esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos.

Ejemplo:

```
foreach ($arr as $key => $val)
```

```
{
```

```
}
```

```
if ($foo == $bar)
```

```
{
```

Capítulo 3. “Implementación y prueba del módulo”

```
}  
else  
{  
}  
for ($i = 0; $i < 10; $i++)  
{  
    for ($j = 0; $j < 10; $j++)  
        {  
        }  
}
```

◆ Funciones:

Siempre comienzan con minúscula y en caso de utilizar parámetros, estos son separados por espacio luego de la coma que los separa.

Ejemplo:

```
function get_file_properties()  
{  
}
```

3.3. Diagrama de despliegue

Es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo (JACOBSON, 2000). Cada nodo representa un recurso de cómputo que poseen relaciones que representan los medios de comunicación entre ellos.

Capítulo 3. “Implementación y prueba del módulo”

El diagrama de despliegue (Ilustración 15) de la propuesta de solución está compuesto por:

◆ PC Cliente

Lugar donde estarán alojados los navegadores web para acceder a las páginas clientes que interactúan con las páginas servidoras mediante el protocolo HTTP/HTTPS.

◆ Servidor Web

Representa la estación de trabajo del GDA eXcriba y su núcleo el ECM Alfresco, así como la capa de servicios y el lugar donde estará alojada la propuesta de solución.

◆ Servidor Base de Datos

Representa el servidor donde estará el sistema gestor de bases de datos que se comunica con el Servidor Web a través del protocolo TCP/IP.

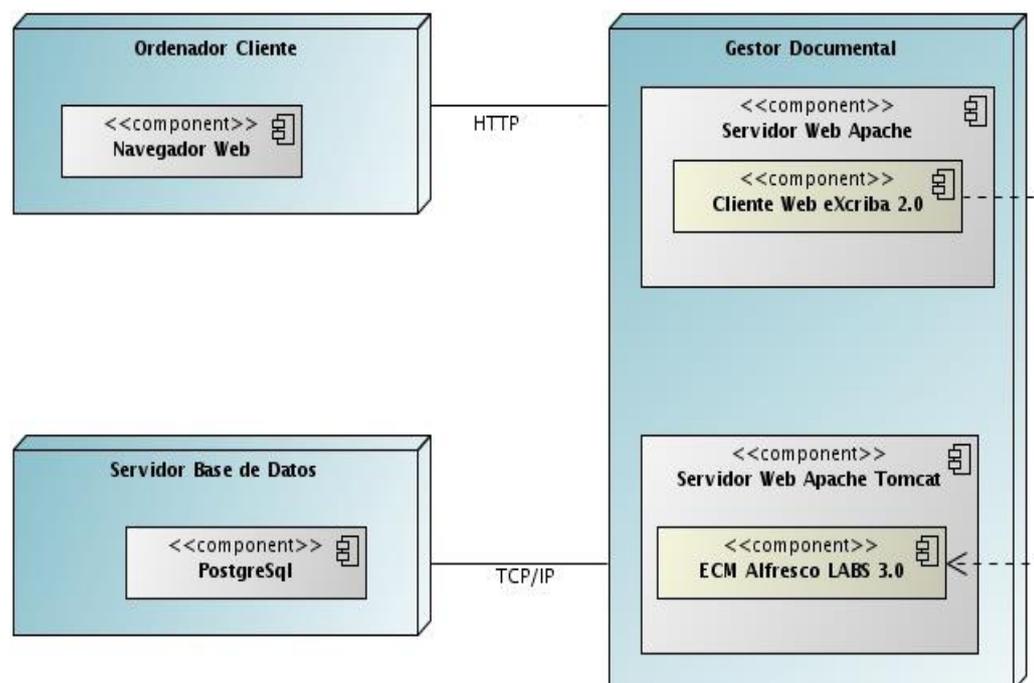


Ilustración 15. Diagrama de despliegue.

Capítulo 3. “Implementación y prueba del módulo”

3.4. Diagrama de componentes

Un diagrama de componentes muestra las dependencias lógicas entre componentes de *software*, sean éstos fuentes, binarios o ejecutables. Los componentes de *software* tienen tipo, que indica si son útiles en tiempo de compilación, enlace o ejecución. Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Estos diagramas representan todos los tipos de elementos de *software* que entran en la fabricación de aplicaciones informáticas. (FERNANDEZ, 2010).

Capítulo 3. “Implementación y prueba del módulo”

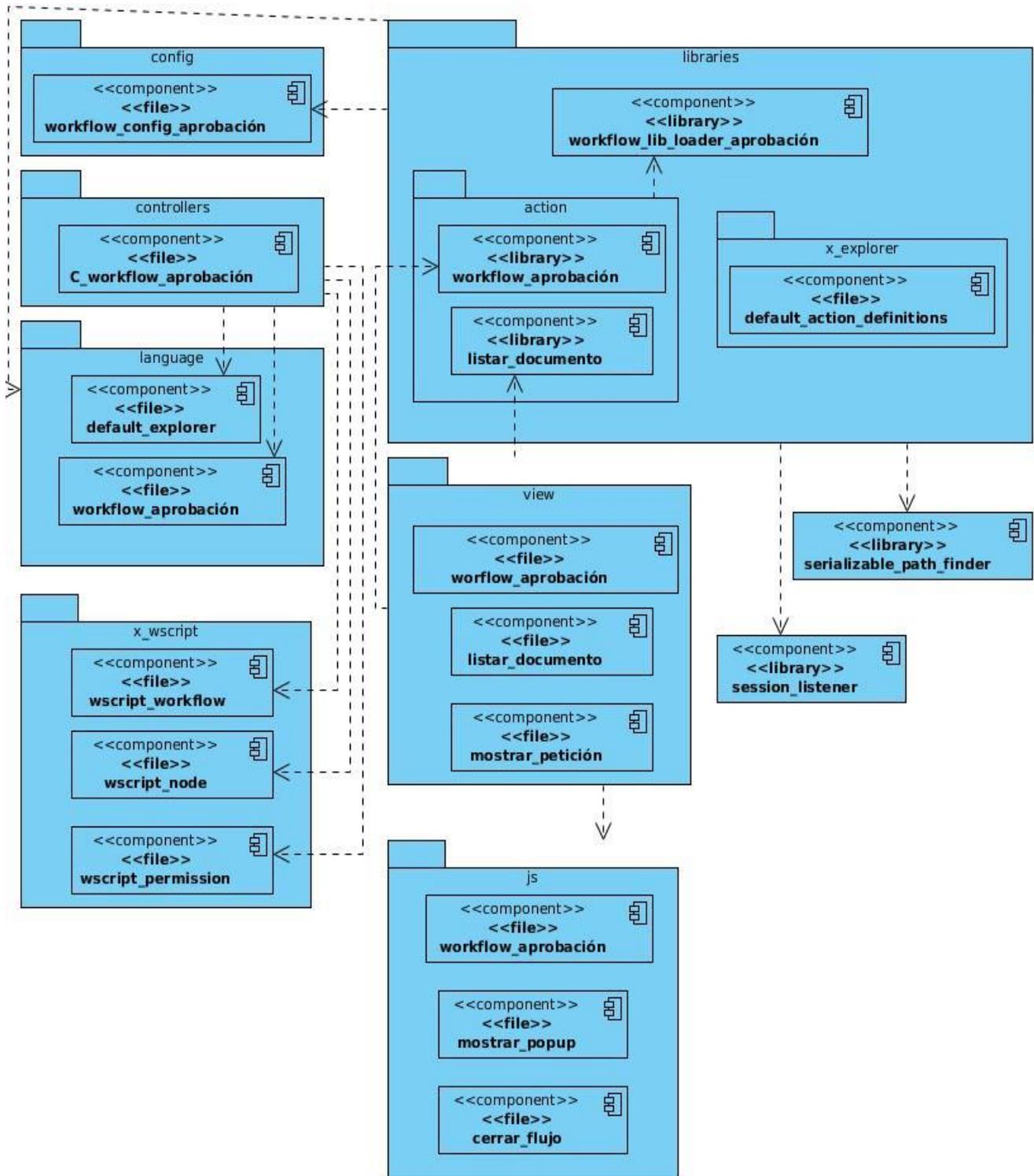


Ilustración 16. Diagrama de componente del módulo flujos de trabajo de aprobación.

Capítulo 3. “Implementación y prueba del módulo”

3.4.1. Descripción del diagrama de componentes

El módulo cuenta con un paquete general llamado *workflow_*aprobación que contiene todos los componentes del módulo agrupados en dos paquetes principales *php* conteniendo toda la implementación del lado del servidor y *resource* que contiene la implementación del lado del cliente.

Dentro del paquete *php* se encuentra el componente *config*, donde se definen las configuraciones para ejecutar las acciones o funcionalidades que brinda el módulo, en el componente *libraries* es donde se construyen las vistas asociadas a las funcionalidades del módulo, preparando para ello los datos que serán mostrados finalmente en la interfaz de usuario, las *libraries* engloban dos componentes *actions* y *x_explorer* en los cuales se construyen las vistas que serán mostradas al usuario y contiene las clases encargadas de la comunicación con el repositorio de contenidos respectivamente, además en el paquete *php* se encuentran los componentes *views*, que contiene las interfaces visuales que se mostrarán al usuario en el navegador y *controllers* que contiene la clase *C_workflow_*aprobación donde se coordina la ejecución de las diferentes operaciones que se pueden llevar a cabo en el módulo.

En el componente *resource* son englobados los paquetes *images*, *javaScript* y *stylesheets*; en *images* se encuentran las imágenes utilizadas para decorar las interfaces del módulo, *javaScript* contiene los ficheros con los *script* que serán ejecutados en el cliente (navegador web) durante la interacción del usuario con las interfaces que brinda el módulo y por último el componente *stylesheets* engloba las hojas de estilo que se le aplican a las interfaces que serán mostradas al usuario.

3.5. 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 (PRESSMAN, 2005).

Capítulo 3. “Implementación y prueba del módulo”

3.5.1. Pruebas de caja negra

El objetivo principal de las pruebas de caja negra es verificar que se cumplan los requisitos funcionales planteados, para ello se deben de crear una serie de juegos de datos los cuales serán las entradas que se le proporcionará al *software*, estos pueden ser datos formulados correcta o incorrectamente debido a que la aplicación debe saber manejar cualquier dato introducido sin importar su índole. Una vez introducido cada juego de datos se debe ser muy observador con las respuestas que brinda el sistema para verificar si son los resultados que realmente se esperan.

3.5.1.1. Técnicas existentes

Para desarrollar las pruebas de caja negra existen es posible aplicar varias técnicas, entre ellas están:

1. **Técnica de la partición de equivalencia:** esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del *software*.
2. **Técnica del análisis de valores límites:** esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
3. **Técnica de grafos de causa-efecto:** es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

La técnica de la partición de equivalencia es considerada una de las técnicas más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el *software*, esta técnica se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Por lo general, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica (PRESSMAN, 2001).

A continuación se describen los casos de prueba realizados utilizando la técnica de partición equivalente, donde se exponen el nombre del caso de uso, la información de entrada, el

Capítulo 3. “Implementación y prueba del módulo”

resultado que se obtiene y las condiciones que se deben cumplir en el momento de su ejecución.

3.5.1.2. Casos de prueba de caja negra

Caso de uso: Gestionar petición de aprobación

Descripción de la funcionalidad: el caso de uso comienza cuando el usuario desea gestionar una petición de aprobación. Para ello puede crear, enviar, listar, aceptar, rechazar o cancelar la misma. El caso de uso termina cuando realiza alguna de estas acciones.

Entrada	Resultado	Condiciones
El usuario crea un flujo de trabajo de aprobación llenando los campos: Destinatarios: "ysibat" Tipo de aprobación: "En serie" Fecha de caducidad: "04/24/2013" Prioridad: "1" Descripción: "Aprobación de los temas de investigación para 4to año"	Se almacena en el repositorio el flujo de trabajo y se muestra el mensaje "El 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 de aprobación.
El usuario crea un flujo de trabajo de aprobación llenando los campos: Destinatarios: "" Tipo de aprobación: "En serie" Fecha de caducidad: "04/24/2013" Prioridad: "1" Descripción: "Aprobación de los temas de investigación para 4to año"	Muestra el mensaje "El campo Destinatarios no debe 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 de aprobación.

Capítulo 3. “Implementación y prueba del módulo”

<p>El usuario crea un flujo de trabajo de aprobación llenando los campos: Destinatarios: "ysibat" Tipo de aprobación: "" Fecha de caducidad: "04/24/2013" Prioridad: "1" Descripción: "Aprobación de los temas de investigación para 4to año"</p>	<p>Muestra el mensaje "El campo Tipo de aprobación no debe estar vacío".</p>	<p>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 de aprobación.</p>
<p>El usuario crea un flujo de trabajo de aprobación llenando los campos: Destinatarios: "ysibat" Tipo de aprobación: "En serie" Fecha de caducidad: "" Prioridad: "1" Descripción: "Aprobación de los temas de investigación para 4to año"</p>	<p>Muestra el mensaje "El campo Fecha no debe estar vacío".</p>	<p>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 de aprobación.</p>
<p>El usuario crea un flujo de trabajo de aprobación llenando los campos: Destinatarios: "ysibat" Tipo de aprobación: "En serie" Fecha de caducidad: "04/24/2013" Prioridad: "1" Descripción: ""</p>	<p>Muestra el mensaje "El campo Descripción no debe estar vacío".</p>	<p>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 de aprobación.</p>
<p>El usuario crea un flujo de trabajo de aprobación llenando los campos: Destinatarios: "ysibat"</p>	<p>Muestra el mensaje "La Fecha de caducidad ya ha pasado".</p>	<p>El usuario tiene que estar autenticado en el sistema. Debe existir al menos un</p>

Capítulo 3. “Implementación y prueba del módulo”

Tipo de aprobación: "En serie" Fecha de caducidad: "04/22/2013" Prioridad: "1" Descripción: "Aprobación de los temas de investigación para 4to año"		documento registrado en el sistema para poder crear un flujo de trabajo de aprobación.
--	--	--

Tabla 9 Escenario. Crear flujo de trabajo de aprobación

Entrada	Resultado	Condiciones
Listar petición de aprobación.	Muestra en una tabla los datos de los flujos de trabajo que tiene pendiente el usuario: Identificador del flujo de trabajo, Estado, Prioridad, Fecha de creado, Fecha de caducidad, Descripción, Documento, Comentario y acciones.	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 de aprobación.
No hay flujos de trabajo.	Muestra el mensaje "No tiene flujos de trabajos pendientes".	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 de aprobación.

Tabla 10 Escenario. Listar petición de aprobación

Capítulo 3. “Implementación y prueba del módulo”

Caso de uso: Consultar el documento asociado a la petición

Descripción de la funcionalidad: el caso de uso comienza cuando el usuario desea consultar los datos referentes a un documento asociado a una petición, terminando el caso de uso una vez realizada la acción.

Entrada	Resultado	Condiciones
Consultar el documento asociado a la petición.	Se almacena 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 crear un flujo de trabajo de aprobación.
El usuario debe llenar los campos: Destinatarios: "admin" Estado: "En progreso" Comentario: "El documento ha sido aprobado"	Se almacena 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 crear un flujo de trabajo de aprobación.
El usuario debe llenar los campos: Destinatarios: "admin" Estado: "En progreso" Comentario: ""	Muestra el mensaje "El campo no debe 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 de aprobación.

Capítulo 3. “Implementación y prueba del módulo”

El usuario debe llenar los campos: Destinatarios: "" Estado: "En progreso" Comentario: "El documento ha sido rechazado"	Muestra el mensaje "El campo no debe 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 de aprobación.
---	--	---

Tabla 11 Escenario. Consultar el documento asociado a la petición

3.4.1.3. Resultados de las pruebas realizadas

No Conformidades	Iteración 1	Iteración 2	Iteración 3
Alta	2	-	-
Baja	5	3	-

Tabla 12 Tabla de resultados

Las pruebas realizadas, como se puede apreciar en la tabla anterior, se llevaron a cabo en tres iteraciones, en la primera iteración se detectaron siete no conformidades de las cuales cinco de ellas fueron bajas (no significativas – errores de interfaz) y dos altas (significativas – errores de funcionalidad y validación del módulo); en la segunda iteración se detectaron tres no conformidades todas ellas bajas (no significativas - errores de interfaz) 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.

Capítulo 3. “Implementación y prueba del módulo”

3.6. Conclusiones del capítulo

En el presente capítulo se elaboraron y aplicaron los casos de prueba de partición equivalente, con el objetivo de validar y verificar la calidad de la propuesta desarrollada. Según las pruebas realizadas, se puede concluir que el resultado final de la aplicación se encuentra en condiciones para realizar la ejecución de los flujos de trabajo de aprobación de documentos electrónicos en el GDA eXcriba, logrando contar así con un producto de *software* listo para el despliegue.

Conclusiones generales

Con el desarrollo de la presente investigación que permite realizar la ejecución de flujos de trabajo de aprobación formal de documentos electrónicos en el GDA eXcriba, se arribaron a las siguientes conclusiones:

- ◆ Los flujos de trabajos son esenciales en la Gestión documental, ya que acercan personas, procesos y máquinas, ahorrando tiempo y acelerando la realización del trabajo.
- ◆ El patrón de arquitectura en capas establecido permitió una mejor organización de cada una de las partes del subsistema, contribuyendo a que este sea desarrollado con la calidad requerida.
- ◆ La implementación del módulo permitió automatizar los procesos de negocio en el GDA eXcriba.
- ◆ Con la utilización de las pruebas de caja negra, se logró comprobar el éxito del cumplimiento de las funcionalidades del módulo para el GDA eXcriba.

Recomendaciones

Partiendo del estudio realizado y el conocimiento adquirido durante el desarrollo de la presente investigación se recomienda:

- ◆ Adicionar al módulo una funcionalidad que permita realizar el proceso de aprobación en el que esté involucrado más de un usuario en una misma petición.
- ◆ Adicionar al módulo una funcionalidad que permita una vez que haya pasado la fecha límite, cambiar el estado a caduco.
- ◆ Utilizar el trabajo como material de aprendizaje para aquellas personas que vayan a realizar una aplicación con funcionalidades similares.

Referencias bibliográficas

- ◆ ABRAHAM, Oter. Java². 2003. Este documento puede ser distribuido sólo bajo los términos y condiciones de la licencia de Documentación de javaHispano v1.0. Disponible en: <http://www.javahispano.org/licencias/>).
- ◆ BAKKEN, Stig Sæther. Manual de PHP: ¿Qué es PHP? [en línea]. 2001. Editado por Rafael Martínez. 1062 pp. Este manual es © Copyright 1997, 1998, 1999, 2000, 2001 del Grupo de documentación de PHP.
- ◆ BHAUMIK, Snig. *Alfresco 3 Cookbooks*. 2011. Published by Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK. 436p. ISBN 978 1 849511 08 7.
- ◆ BARROCA, Eric, Nuxeo Enterprise Platform: Manual de Usuario [en línea]. 2010 [Consultado el: 22 Febrero de 2012]. Disponible en: <http://es.scribd.com/doc/91074458/Manual-Usuario-Nuxeo-v1>.
- ◆ BUSCHMANN, F, *et al. Pattern – Oriented Software Architecture. A System of Patterns*. John Wiley & Sons, Inglaterra. 1996. 300 p.
- ◆ CALIDAD. Aprobación de documentos [en línea]. 2004. [Consultado el: 8 noviembre de 2012]. Disponible en: http://www.portalcalidad.com/etiquetas/290-Aprobacion_de_documentos.
- ◆ CARUANA, D.; NEWTON, J, *et al. Alfresco Practical Solutions for Enterprise Content Management* Indianapolis Indiana: Ediciones Wiley Publishing Inc., 2010. 575p. ISBN: 978-0-470-57104-0.
- ◆ CARRASCO, J. B. *et. al. Gestión de procesos*. ISBN 978-956-7604-20-3. 450 p.
- ◆ CEI, P. L. *Alfresco 3 Web Services*. 2010. Published by Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK. 411pp. ISBN 978 1 849511 52 0.

Referencias bibliográficas

- ◆ CHAVES, A. *La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software*. 10 ed. Costa Rica: Universidad de Costa Rica, 2005, vol. VI, 13 p. ISSN 1409 4746.
- ◆ FERNANDEZ, García E. Diagrama de componentes y objetos. <http://ingenieriasoftwaredos.wikispaces.com/Diagrama+de+componentes+y+objetos> [en línea]. 2010. [Consultado el: 13 de febrero de 2013].
- ◆ FIELDING, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.
- ◆ 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 de 2012. ISBN: 987-959-286-019-3.
- ◆ JACOBSON, Ivar, *et al.* *El Proceso Unificado de Desarrollo de Software*. La Habana: Ediciones Félix Varela, 2004. ISBN: 84-7829-036-2.
- ◆ KRIETE, Pascal. Ellislab: What is CodeIgniter? [en línea]. 2006. [Consultado el: 20 de mayo de 2013]. Disponible en: <http://ellislab.com/codeigniter>.
- ◆ KUMAR, Pawan. *Documentum 6.5 Content Management Foundations*. Birmingham Mumbai, 2010. 393 p. ISBN 978-1-849680-22-6.
- ◆ LARMAN, Craig. *UML Y PATRONES: Introducción al análisis y diseño orientado a objetos*. Prentice Hall, Hispanoamericana, S.A. 1999. 2004. ISBN: 9701702611
- ◆ NIELSEN, Paul D. INSTITUTE, S. E. Capability Maturity Model Integration (CMMI) [en línea]. 2012. [Consultado el: 23 Febrero de 2013]. Disponible en: <http://www.sei.cmu.edu/cmmi/>.
- ◆ OFFICE, M. Comprender los flujos de trabajo de aprobación en SharePoint 2010 [en

Referencias bibliográficas

- línea]. 2012. [Consultado el: 20 Febrero de 2012] Disponible en: http://office.microsoft.com/es-es/sharepoint-designer-help/comprender-los-flujos-de-trabajo-de-aprobacion-en-sharepoint-2010-HA101857172.aspx#_Toc264526205.
- ◆ O'REILLY. A Technical Introduction to XML [en línea]. 2010, [Consultado el: 30 Enero de 2013]. Disponible en: <http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN58>.
 - ◆ PATEL, Viral. viralpatel.net: Introduction to FreeMarker Template (FTL) [en línea]. 2012. [Consultado el: 20 mayo de 2013]. Disponible en: <http://viralpatel.net/blogs/introduction-to-freemarker-template-ftl/>.
 - ◆ PEPPER S. Definition of Process Management. *Medwave* [en línea]. nº 5032.2011. [Consultado el: 18 Enero de 2013]. Disponible en: <http://www.mednet.cl/link.cgi/Medwave/Series/GES03-A/>.
 - ◆ PÉREZ, Damián. Maestro de web: Zend Estudio [en línea]. 2008. [Consultado el: 22 de abril de 2013]. Disponible en: <http://www.maestrosdelweb.com/editorial/editores-web-que-facilitan-tu-trabajo/>.
 - ◆ PIXELWARE. Gestión de proceso de negocio [en línea]. [Consultado el: 18 Febrero de 2012]. Disponible en: <http://www.pixelware.com/workflow-flujo-trabajo.htm>.
 - ◆ PRESSMAN, Roger S. *Ingeniería de Software, un enfoque práctico*. 2001 Quinta edición. S.I: McGraw-Hill Companies, 2002. ISBN: 8448132149.
 - ◆ PRESSMAN, Roger S. *Ingeniería del Software Un enfoque práctico*. La Habana: Ediciones Félix Varela, 2005. 601 p. ISBN: 970-105-473-3.
 - ◆ SARRION, Eric. *jQuery.UI. Introduction to jQuery* . Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. 2012.225 pp. ISBN: 978 -1-449 - 31699 - 0.
 - ◆ SEVILLANO, Fernando. REINDUSTRIA. Definición de Proceso de Negocio [en línea]. 2009. [Consultado el: 14 enero de 2013] Disponible en:

Referencias bibliográficas

<http://redindustria.blogspot.com/search?q=proceso+de+negocio>.

- ◆ SOMMERVILLE, Ian. *Ingeniería del software*. Séptima Edición. España, Madrid. Pearson Educación, SA., 2005. 119 pp. ISBN: 978 0 321 31379 9.
- ◆ SHARIFF, Munwar. *Alfresco Enterprise Content Management Implementation*. Birmingham Mumbai: Editor Mike W. Walker, 2006. 353 p. ISBN: 1-904811-11-6.
- ◆ VALDÉS, D. P. Diferentes lenguajes de programación [en línea]. 2010. [Consultado el: 31 Enero de 2013]. Disponible en: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web>.

Bibliografía

- ◆ ABRAHAM, Oter. Java². 2003. Este documento puede ser distribuido sólo bajo los términos y condiciones de la licencia de Documentación de javaHispano v1.0. Disponible en: <http://www.javahispano.org/licencias/>).
- ◆ BAKKEN, Stig Sæther. Manual de PHP: ¿Qué es PHP? [en línea]. 2001. Editado por Rafael Martínez. 1062 pp. Este manual es © Copyright 1997, 1998, 1999, 2000, 2001 del Grupo de documentación de PHP.
- ◆ BHAUMIK, Snig. *Alfresco 3 Cookbooks*. 2011. Published by Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK. 436p. ISBN 978 1 849511 08 7.
- ◆ BARROCA, Eric, Nuxeo Enterprise Platform: Manual de Usuario [en línea]. 2010 [Consultado el: 22 Febrero de 2012]. Disponible en: <http://es.scribd.com/doc/91074458/Manual-Usuario-Nuxeo-v1>.
- ◆ BUSCHMANN, F, *et al. Pattern – Oriented Software Architecture. A System of Patterns*. John Wiley & Sons, Inglaterra. 1996. 300p
- ◆ CALIDAD. Aprobación de documentos [en línea]. 2004. [Consultado el: 8 noviembre del 2012]. Disponible en: http://www.portalcalidad.com/etiquetas/290-Aprobacion_de_documentos.
- ◆ CARUANA, D.; NEWTON, J, *et al. Alfresco Practical Solutions for Enterprise Content Management* Indianapolis Indiana: Ediciones Wiley Publishing Inc., 2010. 575p. ISBN: 978-0-470-57104-0.
- ◆ CARRASCO, J. B. *et. al. Gestión de procesos*. ISBN 978-956-7604-20-3. 450 p
- ◆ CEI, P. L. *Alfresco 3 Web Services*. 2010. Published by Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK. 411pp. ISBN 978 1 849511 52 0.

- ◆ CUPP, Kevin, Estándar de codificación [en línea] 2002. [Consultado el: 3 Marzo de 2013]. Disponible en: <http://www.ellislab.com/codeigniter>.
- ◆ CHAVES, A. *La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software*. 10 ed. Costa Rica: Universidad de Costa Rica, 2005, vol. VI, 13 p. ISSN 1409 4746.
- ◆ FERNANDEZ, García E. Diagrama de componentes y objetos [en línea]. <http://ingenieriasoftwaredos.wikispaces.com/Diagrama+de+componentes+y+objetos>, 2010. [Consultado el: 13 de febrero de 2013].
- ◆ FIELDING, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.
- ◆ 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.
- ◆ JACOBSON, Ivar, *et al.* *El Proceso Unificado de Desarrollo de Software*. La Habana: Ediciones Félix Varela, 2004. ISBN: 84-7829-036-2.
- ◆ KRIETE, Pascal. Ellislab: What is CodeIgniter? [en línea]. 2006. [Consultado el: 20 de mayo de 2013]. Disponible en: <http://ellislab.com/codeigniter>.
- ◆ KUMAR, Pawan. *Documentum 6.5 Content Management Foundations*. Birmingham Mumbai, 2010. 393 p. ISBN 978-1-849680-22-6.
- ◆ LARMAN, Craig. *UML Y PATRONES: Introducción al análisis y diseño orientado a objetos*. Prentice Hall, Hispanoamericana, S.A. 1999. 2004. ISBN: 9701702611
- ◆ NIELSEN, Paul D. INSTITUTE, S. E. Capability Maturity Model Integration (CMMI) [en línea]. 2012. [Consultado el: 23 Febrero de 2013]. Disponible en: <http://www.sei.cmu.edu/cmmi/>.

- ◆ OFFICE, M. Comprender los flujos de trabajo de aprobación en SharePoint 2010 [en línea]. 2012. [Consultado: 20 Febrero de 2012] Disponible en: http://office.microsoft.com/es-es/sharepoint-designer-help/comprender-los-flujos-de-trabajo-de-aprobacion-en-sharepoint-2010-HA101857172.aspx#_Toc264526205.
- ◆ O'REILLY. A Technical Introduction to XML [en línea]. 2010, [Consultado el: 30 Enero de 2013]. Disponible en: <http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN58>.
- ◆ PATEL, Viral. viralpatel.net: Introduction to FreeMarker Template (FTL) [en línea]. 2012. [Consultado el: 20 mayo de 2013]. Disponible en: <http://viralpatel.net/blogs/introduction-to-freemarker-template-ftl/>.
- ◆ PEPPER S. Definition of Process Management. *Medwave* [en línea]. nº 5032.2011. [Consultado el: 18 Enero de 2013]. Disponible en: <http://www.mednet.cl/link.cgi/Medwave/Series/GES03-A/>.
- ◆ PÉREZ, Damián. Maestro de web: Zend Estudio [en línea]. 2008. [Consultado el: 22 de abril del 2013]. Disponible en: <http://www.maestrosdelweb.com/editorial/editores-web-que-facilitan-tu-trabajo/>.
- ◆ PIXELWARE. Gestión de proceso de negocio [en línea]. [Consultado el: 18 Febrero del 2012]. Disponible en: <http://www.pixelware.com/workflow-flujo-trabajo.htm>.
- ◆ PRESSMAN, Roger S. *Ingeniería de Software, un enfoque práctico*. 2001 Quinta edición. S.I: McGraw-Hill Companies, 2002. ISBN: 8448132149.
- ◆ PRESSMAN, Roger S. *Ingeniería del Software Un enfoque práctico*. La Habana: Ediciones Félix Varela, 2005. 601 p. ISBN: 970-105-473-3.
- ◆ SARRION, Eric. *jQuery.UI. Introduction to jQuery* . Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. 2012.225 pp. ISBN: 978 -1-449 – 31699 – 0.
- ◆ SEVILLANO, Fernando. REINDUSTRIA. Definición de Proceso de Negocio [en línea]. 2009. [Consultado el: 14 enero de 2013] Disponible en:

<http://redindustria.blogspot.com/search?q=proceso+de+negocio>.

- ◆ SOMMERVILLE, Ian. *Ingeniería del software*. Séptima Edición. España, Madrid. Pearson Educación, SA., 2005. 119 pp. ISBN: 978 0 321 31379 9.
- ◆ SHARIFF, Munwar. *Alfresco Enterprise Content Management Implementation*. Birmingham Mumbai: Editor Mike W. Walker, 2006. 353 p. ISBN: 1-904811-11-6.
- ◆ VALDÉS, D. P. Diferentes lenguajes de programación [en línea]. 2010. [Consultado el: 31 Enero de 2013]. Disponible en: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web>.

Anexos

Anexo A. "Prototipos"

A.1 Crear petición de aprobación

Crear petición de Aprobación

Accesos Directos Entrantes Salientes Notificaciones(0)

Regresar

Opciones Generales

Usuarios (*): ysibat	Acciones (*): >> <<	Destinatarios (*): admin
Fecha de caducidad (*): 04/26/2013	Tipo de aprobación (*): <input checked="" type="radio"/> Aprobación en serie <input type="radio"/> Aprobación en paralelo	Prioridad : 2
Descripción (*): Aprobación de los objetivos	Acciones (*): Enviar Limpiar	

Ilustración 17. Crear petición de aprobación en serie

Crear petición de Aprobación

Accesos Directos Entrantes Salientes Notificaciones(9)

Regresar

Opciones Generales

Usuarios (*)	Acciones (*)	Destinatarios(*)
ysibat	>> <<	admin

Fecha de caducidad(*)	Tipo de aprobación (*)	Prioridad
05/31/2013	<input type="radio"/> Aprobación en serie <input checked="" type="radio"/> Aprobación en paralelo	2

Descripción (*)	Acciones (*)
Aprobación de los objetivos	Enviar Limpiar

Ilustración 18. Crear petición de aprobación en paralelo

A continuación se describen los campos que el usuario deberá llenar para crear una nueva petición.

1. Tipo de aprobación

El usuario debe seleccionar el tipo de aprobación que se va a realizar, en serie o en paralelo.

2. Seleccionar destinatario

En este campo el remitente indica cual es el usuario que deberá aprobar el documento asociado a la petición. Si la aprobación es en serie, el nombre de los destinatarios tiene que estar en el orden en el que deben aprobar el documento (Ilustración A1).

3. Prioridad

En el campo **Prioridad** se introduce el grado de importancia que posee la petición: 1 (Alta), 2 (Media), o 3 (Baja). La prioridad es meramente informativa.

4. Fecha de caducidad

En el campo se introduce la fecha en que la petición caducará. La caducidad es un valor importante pues permite conocer al destinatario la fecha límite para poder aprobar el documento.

5. Descripción

En el cuadro de descripción el remitente podrá poner los comentarios que desee sobre el documento asociado o de forma general sobre la petición.

A.2 Bandeja de tareas

The screenshot shows the eXcriba web application interface. At the top, there is a header with the logo 'eXcriba Gestor de Documentos Administrativos' and user information 'Administrar Contenido Ayuda | Bienvenido: yanet'. Below the header is a 'Workspace' section with a search bar. A notification banner indicates 'Tiene (1) notificación sin leer'. The main area contains a toolbar with buttons for 'Adicionar Documento', 'Crear Documento', 'Crear Carpeta', and 'Bandeja de Tareas'. Below the toolbar is a table with columns: 'Nombre', 'Título', 'Fecha de creación', 'Autor', and 'Asignar'. A single task is listed: 'Temas de investigación para 4to año' with title 'Temas de investigación para 4to año', creation date '06-may-2013 11:22:05', and author 'ysibat'. A blue arrow points to the 'Bandeja de Tareas' button, and a black box with white text 'Bandeja de Tareas' is overlaid on the task table.

Nombre	Título	Fecha de creación	Autor	Asignar
Temas de investigación para 4to año	Temas de investigación para 4to año	06-may-2013 11:22:05	ysibat	

Ilustración 19. Bandeja de tareas

A.3 Listado de las peticiones

eXcriba Gestor de Documentos Administrativos

Administrar Contenido | Ayuda | Bienvenido: yanet

Workspace

Listar petición de Aprobación

▲ Tiene (1) notificación sin leer

[Regresar](#)

Flujo de trabajo	Prioridad	Estado	Fecha de creado	Fecha de caducidad	Descripción	Documento	Comentario	Acciones
1081345	1	No está iniciado	17-may-2013 15:29:02	31-may-2013 0:00:00	Aprobacion de los temas de investigacion	Temas de investigación de 4to año.odt		Mostrar petición

Ilustración 20. Listado de las peticiones

Desde la bandeja de tareas los usuarios podrán ver las peticiones que le han sido asignadas con los siguientes datos:(Identificador del flujo de trabajo, prioridad, estado, fecha de creado y de caducidad, descripción y el documento asociado, que puede ser consultado). Además, se representa la acción que se puede realizar sobre las peticiones (**Mostrar petición**).

A.4 Mostrar petición

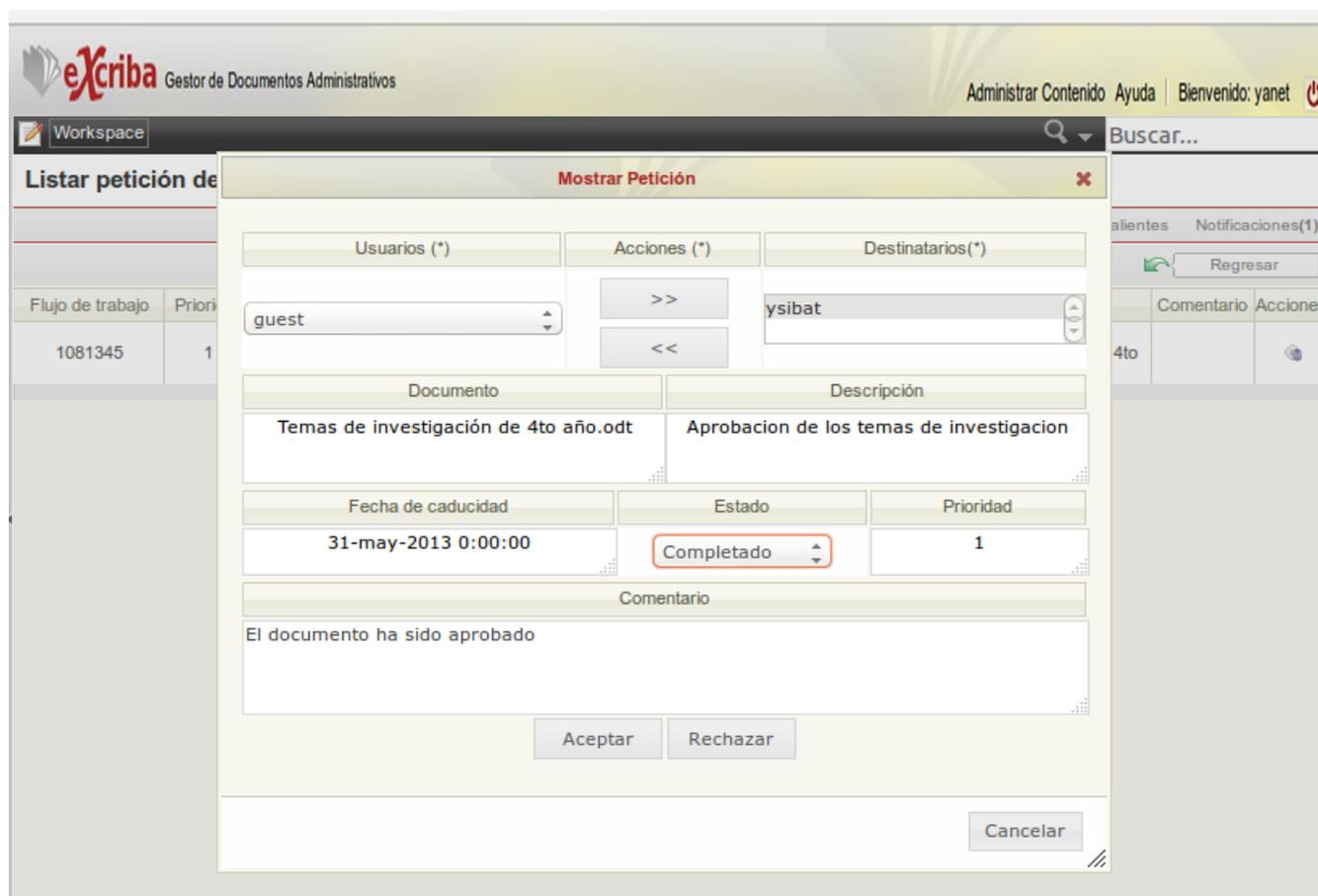


Ilustración 21. Mostrar petición

Las peticiones son mostradas a los destinatarios como se observa en la Ilustración 21. El cuadro de las Propiedades de la petición es informativo, en él aparecen el nombre del documento, la fecha de caducidad, la prioridad y la descripción; una vez que el destinatario examine las propiedades de la petición y haya consultado previamente el documento asociado, podrá cambiar el estado, realizar algún comentario al respecto y deberá seleccionar el destinatario al cual le llegaran las notificaciones asociadas a las acciones que el usuario haya realizado.

Anexo B. “Descripción de los casos de uso”

B1. Consultar el documento asociado a una petición

Caso de uso	Consultar el documento asociado a una petición	
Actor	Usuario (inicia)	
Resumen	El caso de uso comienza cuando el usuario desea consultar los datos referentes a un documento asociado a una petición, terminando el caso de uso una vez realizada la acción.	
Prioridad	Alta	
Complejidad	Alta	
Precondiciones	Debe de existir peticiones con los documentos correspondientes asociados.	
Poscondiciones	Los datos fueron consultados.	
Flujo de eventos		
Flujo básico <Consultar documento >		
	Actor	Sistema
	1. El usuario accede a la bandeja de tareas de su espacio de trabajo.	2. Muestra en la bandeja de tareas las peticiones que le han sido asignadas.
	3. Selecciona la acción () “Mostrar petición”.	4. Muestra una interfaz informativa de la petición sobre la cual el usuario deberá seleccionar el destinatario, actualizar el estado y realizar algún comentario al respecto, una vez que haya consultado el documento previamente.
	5. El usuario tendrá la opción de elegir Aceptar, Rechazar o Cancelar. (Ver descripción textual en el CU Gestionar petición).	6. Almacena los cambios realizados en el repositorio. Termina el caso de uso.

Tabla 13. Descripción textual del CU: Consultar el documento asociado a una petición

Anexo C. "Diagrama de clases del diseño"

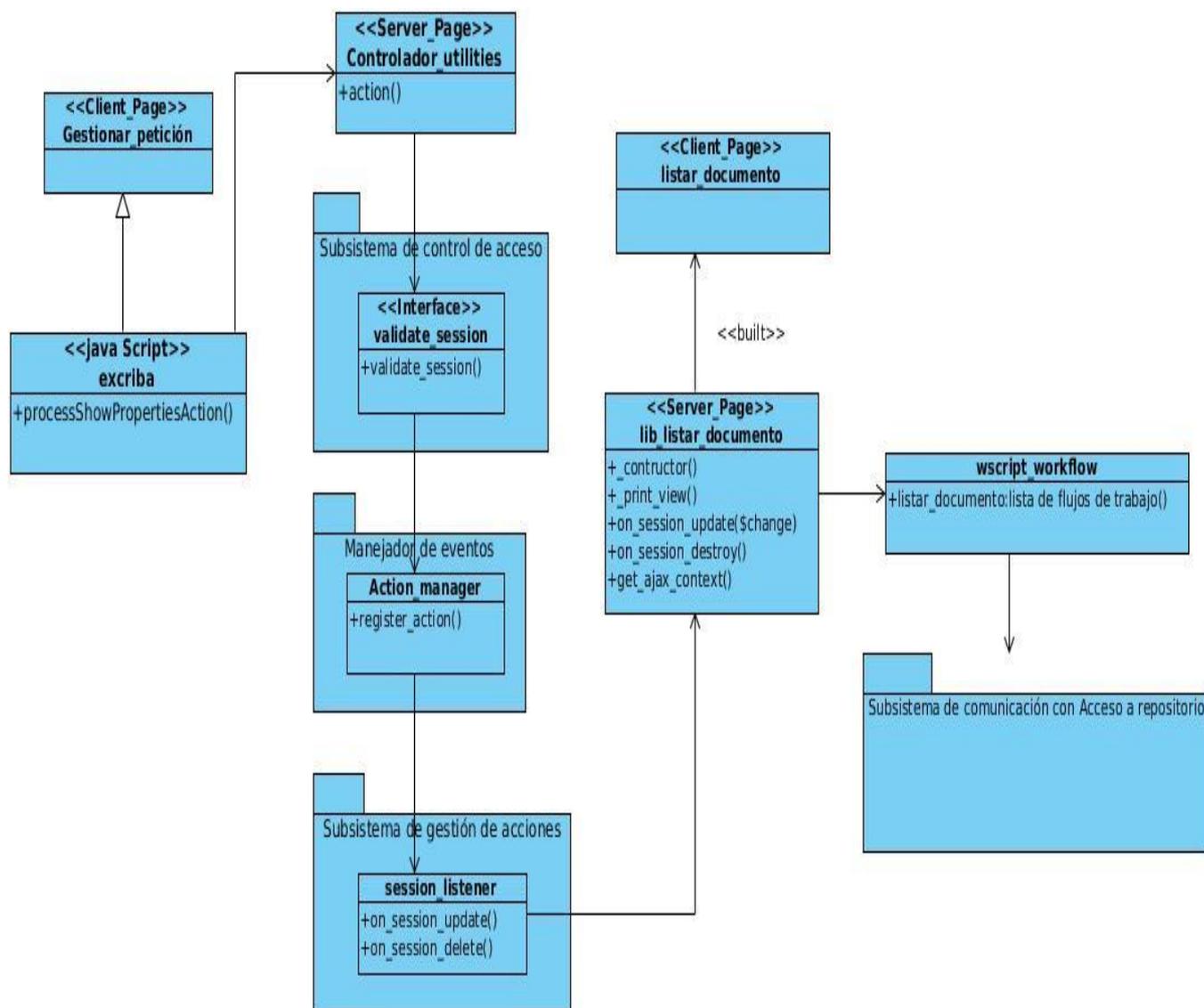


Ilustración 22. Diagrama de clases del diseño: Listar petición

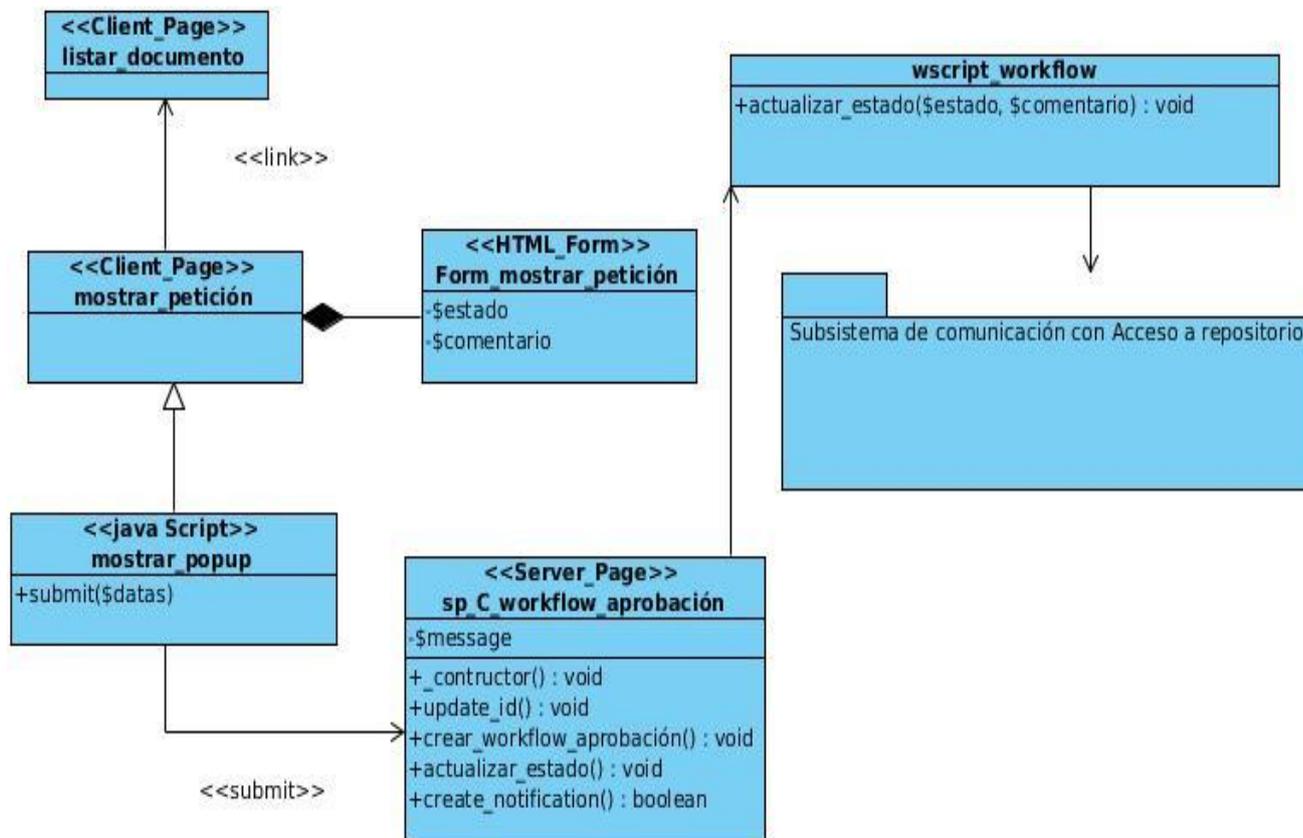


Ilustración 23. Diagrama de clases del diseño: Consultar el documento asociado a una petición

Anexo D. “Diagrama de secuencia del diseño”

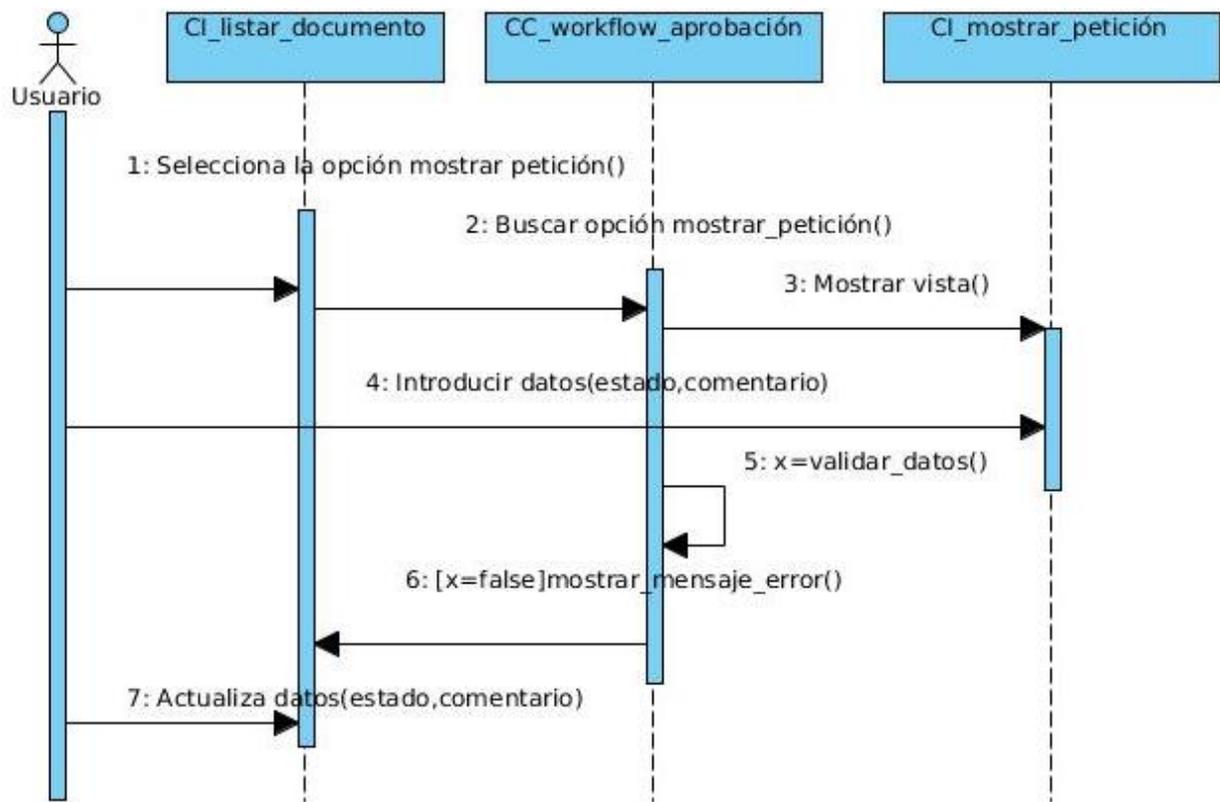


Ilustración 24. Diagrama de secuencia del caso de uso: Consultar el documento asociado a una petición

Anexo E. “Descripción de los servicios”

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

Tabla 14. Descripción del servicio: Modificar petición

Servicio: Finalizar flujo de trabajo		
Paquete: /cu/uci/excriba/workflow	Plantillas de Respuestas: JSON,XML,HTML	
Descripción: Permite finalizar 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/excriba/workflow/end_Task?id={id}	Método HTTP: GET	Ruta Relativa: alfresco/service/cu/uci/excriba/workflow/end_Task?id={id}
Dirección del documento de descripción: classpath:alfresco/extension/templates/webscripts/cu/uci/excriba/workflow/end_Task.get.desc.xml		
Lenguaje de implementación: Javascript		
Dirección del paquete: /cu/uci/excriba/webscripts		

Tabla 15. Descripción del servicio: Finalizar flujo de trabajo