

Universidad de las Ciencias Informáticas

Facultad 1



Módulo de gestión de tipos documentales para el ECM Alfresco

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

Autores: Juan Danilo Rodríguez Bocourt

Tutores: Ing. Misael Fonseca Mata

Ing. Annia Surós Vicente

Co-Tutor: Ing. Reinier Elejalde Chacon

Ciudad de la Habana, Junio del 2011

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la UCI los derechos patrimoniales de la misma, con carácter exclusivo.

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

Juan Danilo Rodríguez Bocourt

Ingeniero
Ing. Misael Fonseca Mata

Ingeniero
Ing. Annia Surós Vicente

Ingeniero
Ing. Reinier Elejalde Chacon

Resumen

En el gestor de contenidos empresariales (ECM) Alfresco¹ se generan modelos de contenidos¹ mediante ficheros xml¹, este proceso es bien complejo para los usuarios que no tienen conocimiento en informática, así como lentitud en la elaboración de los ficheros. Por tales razones se desarrolló una aplicación de software (XMLEditor v1.0) presentando una interfaz amigable para la gestión de los modelos de contenidos. A pesar que esta herramienta permitió llevar una gestión de forma cómoda y legible a los usuarios, la misma carece de funcionalidades, dificultando la eficiencia para gestionar los modelos.

El presente trabajo extiende el módulo XMLEditor 1.0 a su versión 2.0, incorporando nuevas funcionalidades. Permitiendo gestionar de manera eficiente los modelos de contenidos en el ECM Alfresco. Se realiza un estado del arte sobre los diferentes software que generan ficheros en formato xml para obtener una mayor visión en las funcionalidades a incorporar; además, un análisis de las herramientas, lenguajes de programación y metodología de software utilizada para extender el desarrollo de la nueva versión.

Finalmente, se diseñaron los casos de pruebas de tipo caja negra para medir la calidad del módulo, permitiendo garantizar que las funcionalidades incorporadas no presenten error en el uso de la aplicación por los usuarios.

Palabras claves: Alfresco, modelos de contenidos, XMLEditor, xml.

¹Consulte el glosario de términos

Índice General

Introducción	1
1. Fundamentación teórica	5
1.1. Introducción	5
1.2. Tipologías documentales	6
1.3. Importancia de Alfresco para eXcriba	6
1.4. Modelos de Contenidos de Alfresco	7
1.5. Estado del arte	8
1.5.1. Conclusiones parciales	11
1.6. Actualidad en Cuba	12
1.7. Metodologías de desarrollo y modelado	12
1.7.1. RUP	12
1.7.2. Conclusiones parciales	15
1.8. Lenguaje de modelado	15
1.8.1. UML	15
1.9. Lenguaje de programación e IDE	16
1.10. Conclusiones del capítulo	17
2. Características del sistema.	19
2.1. Introducción	19
2.2. Especificación de los requisitos de software	20
2.3. Descripción del sistema propuesto	23

2.4. Especificación de los Casos de uso del sistema	24
2.4.1. Gestionar Aspecto	24
2.4.2. Gestionar Tipo de Contenido	28
2.4.3. Gestionar Asociaciones	32
2.4.4. Gestionar Restricciones de los metadatos asociados a tipos documentales	36
2.4.5. Exportar a Alfresco	40
2.4.6. Autenticar en Alfresco	41
2.5. Conclusiones del capítulo	43
3. Análisis y diseño del sistema.	44
3.1. Diagramas de paquetes	45
3.2. Diagramas de clases.	48
3.3. Descripción de las principales clases	50
3.4. Diagramas de colaboración	54
3.5. Aspectos a tener en cuenta	60
3.6. Conclusiones	60
4. Implementación y prueba.	61
4.1. Diagrama de despliegue	61
4.2. Diagrama de componentes	62
4.3. Pruebas de Software	63
4.4. Modelo de prueba	64
4.5. Conclusiones del Capítulo	67
Conclusiones	68
Referencias bibliográficas	72
Bibliografía	74

Introducción

Se entiende por gestión documental al “conjunto de operaciones comprometidas en la búsqueda de la economía y la eficacia en la producción, mantenimiento, uso y destino final de los documentos a lo largo de todo su ciclo de vida; es decir, desde el momento de su concepción en las oficinas administrativas hasta su ingreso en las instituciones de archivos.”[1]

Como resultado de la producción de documentos a través de los años, fue necesario la adopción de políticas, procedimientos, estrategias y tecnologías asociadas a la gestión de los mismos. La gestión de la información implica grandes esfuerzos de consultas, acceso e integración por la gran variedad de tipos de documentos existentes. En la actualidad digitalizar la información es uno de los principales retos que tienen las empresas para poder optimizar su administración y trabajo diario.

Alfresco surge como un proyecto dedicado a la gestión de contenidos empresariales (ECM) desarrollando un producto que permite:[2]

- Un entorno compatible con nuevas aplicaciones comerciales.
- Interoperabilidad entre aplicaciones comerciales gracias a estándares y código abiertos.
- Capacidad para enlazar datos, contenidos, procesos comerciales y contextos.
- Un medio de gestionar los costes corrientes y reducir la inversión de capital.

eXcriba es un Gestor de Documentos Administrativos desarrollado en la Universidad de las Ciencias Informáticas de Cuba, el núcleo para llevar a cabo la gestión de los documentos es el Alfresco.

A partir de este ECM¹ se realizó un análisis para optimizar el desarrollo y la personalización del mismo para obtener una solución de software que le permita a los usuarios desarrollar sus actividades automatizadas sin tener conocimientos avanzados en la informática.

¹gestor de contenidos empresariales

XMLEditor fue uno de los módulos que se desarrolló para que los usuarios no tuvieran que tener alto grado de conocimientos en la informática, pues el mismo se resume en crear y modificar ficheros xml. Este módulo fue desarrollado en el año 2007 como resultado de un trabajo de Diploma del Ingeniero Reinier Elejalde Chacon[3] el cual satisface la necesidad de creación de modelos de contenidos. Es una aplicación informática multiplataforma que permite extender el modelo de contenido de Alfresco con nuevos tipos y relaciones de forma transparente para el usuario final, bajo el concepto de que el usuario necesita estar preparado en su profesión, tener información de los modelos de contenidos que quiere hacer y no preocuparse por la forma en que el módulo construye un xml² con las características que propone el diccionario de datos de Alfresco. Los modelos de contenidos creados a partir de tipologías documentales podrán ser reutilizados cada vez que se necesiten, lo que permite crear en Alfresco un repositorio.

El repositorio de Alfresco provee soporte para el almacenamiento, administración y recuperación del contenido que se gestiona. Para describir la estructura de un contenido cualquiera, soporta un diccionario de datos donde son descritos los metadatos, tipos documentales, propiedades, asociaciones y restricciones que permitirán aumentar la productividad y eficiencia de la empresa en términos de clasificación y categorización del contenido.

Actualmente se cuenta con una herramienta (XMLEditor v1.0) que permite establecer nuevas tipologías documentales así como los metadatos asociados a cada una de estas, sin embargo dicha herramienta no abarca todas las potencialidades que ofrece el diccionario de datos, por citar algunos ejemplos:

- No permite gestionar (crear, modificar, listar y eliminar) los componentes asociados a un modelo de contenido.
- No permite establecer relaciones entre las diferentes tipologías documentales, ni crear tipos documentales compuestos.
- No permite definir las restricciones que se deben o pueden aplicar a los metadatos asociados a una tipología documental.
- No permite exportar los modelos de contenidos de forma automática.

²Diríjase al glosario de términos.

Por lo cual surge el siguiente **problema científico** ¿Cómo mejorar la gestión de tipologías documentales de la herramienta XMLEditor?

Teniendo como **objeto de estudio** el diccionario de datos que provee el repositorio de Alfresco y las diferentes formas en que son descritos los metadatos, tipos documentales, asociaciones, restricciones y delimitando el **campo de acción** a los modelos de contenidos para Alfresco.

El **objetivo general** de esta investigación es incorporar a la herramienta XMLEditor las funcionalidades que permitirán un mejor aprovechamiento del diccionario de datos de Alfresco.

Los **objetivos específicos** son:

1. Identificar y describir las nuevas funcionalidades que se requieren en el módulo ya existente.
2. Implementar las funcionalidades para que usen el diccionario de datos que brinda el repositorio de Alfresco.
3. Exportar el modelo de contenido hacia el Alfresco.
4. Realizar pruebas de caja negra al módulo.

Teniendo los objetivos planteados de esta manera las **tareas a desarrollar** son:

- Realizar estudio sobre el diccionario de datos del Alfresco.
- Realizar estudio de la herramienta XMLEditor v1.0.
- Implementar las nuevas funcionalidades a incorporar a la herramienta XMLEditor v1.0.

Para la correcta realización de las tareas se han usado diferentes métodos de investigación. Dentro de los métodos teóricos se encuentra el **Analítico-Sintético** basándose en los diferentes análisis realizados sobre las teorías y documentos bibliográficos consultados, permitiendo la selección y el filtrado de la información y elementos más importantes para la correcta elaboración del contenido y la obtención de conclusiones. Se utiliza el análisis **Histórico-Lógico** para supervisar la trayectoria y evolución real que va a tener el objeto de estudio de la investigación.

La estructura de este documento consta de un resumen, una introducción, cuatro capítulos, recomendaciones, bibliografía, anexos y glosario de términos.

En el **capítulo 1** se aborda el estado del arte acerca de cómo mejorar el módulo de gestión de tipologías documentales para que cumpla con las nuevas expectativas, así como la realización de un estudio de las principales técnicas que existen a nivel mundial, en Cuba y en la UCI. Se enuncian conceptos, se describen metodologías , procesos y lenguaje de programación utilizados en la actualidad para el desarrollo de este tipo de aplicaciones de escritorio.

En el **capítulo 2** se realiza una descripción de la solución que se propone y el por qué, así como la definición de los requisitos funcionales y no funcionales del sistema.

En el **capítulo 3** se realiza el análisis y diseño del sistema, se hace una descripción de cada una de las clases que serán involucradas en la implementación del sistema y se incorporan diagramas que ayudarán a entender el negocio.

En el **capítulo 4** se describe el proceso de implementación y calidad por el cual pasa el módulo, donde las pruebas de caja negra son fundamentales.

Capítulo 1

Fundamentación teórica

1.1. Introducción

En el mundo existen varias herramientas para edición del lenguaje extensible de maquetado (xml) que pudieran ser de gran utilidad. Esas aplicaciones no están especializadas en elaborar xmls a los Modelos de Contenidos que se necesitan en Alfresco, con la característica de que los xmls serán realizados por personas que tengan conocimiento mínimo de programación, pero que tengan dominio sobre la creación de tipos documentales y tengan la posibilidad de realizar su trabajo sin caer en una profundización de los códigos de programación usados.[4]

En el presente capítulo se abordan temas referentes a los tipos documentales y las diferentes herramientas para la realización del mejoramiento al Módulo de Gestión de Tipos Documentales para el ECM Alfresco.

Además se expondrán conceptos y definiciones que darán una idea general sobre los tipos de datos que se manejan en este módulo. También se expondrá el por qué esta herramienta es necesaria para Alfresco.

Se abordan aspectos que se relacionan con el uso de los medios informáticos. Se realiza una breve explicación de las características de varios lenguajes de programación para desarrollar software de escritorio. Se describen y caracterizan de manera objetiva las metodologías, lenguajes de modelado e IDE ¹ de programación usados para el desarrollo del módulo.

¹Entorno de Desarrollo Integrado

1.2. Tipologías documentales

Un tipo documental es una unidad documental producida por un organismo en el desarrollo de una competencia concreta, regulada por una norma de procedimiento y cuyo formato, contenido informativo y soporte son homogéneos (por ejemplo, el expediente personal). Por tanto se puede decir que son las unidades de información contenidas en los documentos. [5]

Las unidades documentales están clasificadas como simples o complejas[5]:

Simples:

Están formadas por un solo tipo documental, cuyo contenido mantiene una unidad de información. Ejemplos: el oficio, la carta, el memorando, un libro de registro, un libro de caja, recibo, etc.

Complejas:

Se conforma por dos o más tipos documentales que se sustentan entre sí y cuyo contenido mantiene una unidad de información. Se le conoce comúnmente como “expediente”. Ejemplos: el expediente de un estudiante, trámites para licencias, etc.

1.3. Importancia de Alfresco para eXcriba

Alfresco es un gestor de contenidos empresariales basado en software libre, que permite la creación de un repositorio de archivos y contenidos, almacenando toda la información en un mismo repositorio. Los usuarios podrán acceder a la plataforma de forma remota mediante un navegador web.[6]

Alfresco brinda las siguientes funcionalidades:

- Clasificación de los documentos por múltiples criterios.
- Extraer metadatos de los documentos automáticamente.
- Realizar búsquedas complejas, incluyendo el contenido del documento, metadatos, categorías, etc.
- Enviar correo electrónico y notificaciones a otros usuarios incluyendo referencias al documento.
- Crear flujos de trabajo, aprobación, supervisión, etc.

- Control de versiones y auditoría.
- Colaboración en el desarrollo de documentos, bloqueo de documentos en la edición, edición con conexión o sin la misma.
- Seguridad de documentos mediante usuarios, roles y carpetas.
- Ejecutar acciones tales como conversiones, cambios de formato y envíos automáticos.
- Compartir documentos con otros usuarios.
- Incluir documentos en blogs y web de forma sencilla.

El proyecto eXcriba, teniendo en cuenta las funcionalidades anteriores, seleccionó al repositorio del Alfresco como núcleo del GDA eXcriba. Sin Alfresco el equipo de desarrollo de eXcriba tendría que dedicar varias horas, días o meses desarrollando un ECM. [7]

1.4. Modelos de Contenidos de Alfresco

Los contenidos poseen la información suficiente para describir un objeto abstracto o real, por ejemplo: ficheros, carpetas, expedientes, etc. Un Modelo de Contenido es una colección de tipos de contenido y aspectos relacionados. Puede referirse a las definiciones dentro de otro modelo. Además se pueden definir tantos espacios de nombre (namespaces) se requieran.

Un tipo de contenido define la estructura de información que sus nodos² tendrán. Esta estructura está compuesta fundamentalmente por tres elementos: propiedades, asociaciones y restricciones.

Diccionario de Datos de Alfresco

Un diccionario de datos es:

- Un conjunto de metadatos que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.[8]

²Consulte glosario de términos

- En él se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema. Los elementos más importantes son: flujos de datos, almacenes de datos y procesos. El diccionario de datos guarda los detalles y descripción de todos estos elementos.[9]

Un diccionario de datos en Alfresco es un metamodelo que puede describir uno o más modelos de contenidos. El metamodelo da soporte a tipos de contenidos (tipos) y aspectos de contenidos. Ambos proporcionan la capacidad para describir una estructura de contenidos específicos, incluyendo las propiedades (metadatos), las asociaciones y restricciones a metadatos de otros contenidos.

¿Cuál es la diferencia entre un tipo de contenido y aspecto de contenido?

- Los tipos (Types) describen la estructura fundamental de los contenidos.
- Un aspecto de contenido (Content Aspect) representa otra forma de encapsulación (conocido como el corte transversal), donde la estructura descrita se puede aplicar a cualquier contenido de cualquier tipo de contenido.
- Cualquier número de aspectos se pueden aplicar a una sola pieza de contenido.

1.5. Estado del arte

A continuación se realizará un análisis de diferentes software editores de xml para dejar en claro la importancia del desarrollo de un editor de xml especializado en Modelos de Contenidos para el ECM Alfresco:

HTML-Kit Build 292[10]

Idioma: Inglés.

Desarrollador: Htmlkit.

Sistema operativo: Win95/98/98SE/Me/2000/NT/XP.

Descripción:

Es un editor de programación con un entorno de trabajo altamente personalizable con todo un extenso conjunto de herramientas que permiten crear, editar, validar, pre visualizar y publicar páginas web. El editor tiene soporte para varios lenguajes.[10]

Serna

Idioma: Inglés.

Desarrollador: Syntext.

Sistema operativo: Win2000/XP/2003/Vista/Ubuntu.

Descripción:

Serna es un editor visual de documentos xml que opta por mostrar el contenido en un árbol jerárquico desplegable. Al mismo tiempo, si el fichero tiene asociada una hoja de estilos, se verá formateado como en un procesador de textos.[10]

XML Notepad

Idioma: Inglés.

Desarrollador: Microsoft.

Sistema operativo: WinXP/2003/Vista.

Descripción:

XML Notepad cuenta con una intuitiva interfaz de doble ventana (texto y estructura de árbol) que proporciona un entorno de trabajo ágil y cómodo para visualizar y editar documentos en xml. Algunas de sus características más destacadas son una práctica herramienta de búsqueda, soporte para “arrastrar y soltar”, la opción de “deshacer” infinito, además permite personalizar fuentes y colores. Trabaja con documentos de gran tamaño.[10]

Exchanger XML Editor V3.2

Licencia: Comercial.

Idioma: Inglés.

Descripción:

El Exchanger XML Editor es una herramienta multiplataforma que incluye un editor de xml que brinda facilidad de edición, exploración, administración y conversión de documentos xml.[10]

Editix

Licencia: Shareware.

Idiomas: Inglés.

Sistema operativo: 98 NT ME 2000 XP.

Descripción:

Editix es un editor xml profesional y multifunción. Editix está escrito en Java y es capaz de crear plantillas. Está acompañado de un sistema de ayuda con guías y documentación.[10]

Peter's XML Editor

Licencia: Freeware.

Idioma: Inglés.

Sistema Operativo: 95, 98, NT, 2000, XP.

Descripción:

Peter's XML Editor permite al programador editar xml, bien en estructura jerárquica de árbol, o bien directamente en la interfaz habitual de cualquier editor. Algunas de sus características se enuncian a continuación: colorea las etiquetas de los comandos con diferentes tonalidades de colores para hacer más legible el código, dispone de una herramienta para chequear el código xml escrito, permite imprimir la estructura de árbol del fichero, incluye plantillas de código ya predefinidas y permite asociar diferentes extensiones de ficheros al editor. [10]

XML Writer

Sistema Operativo: 98, NT, ME, 2000, XP.

Idioma: Inglés.

Licencia: Shareware.

Descripción:

XML Writer es un editor y validador profesional para archivos en formatos xml, XSL, DTD, hojas de estilo (CSS), HTML y texto plano. Una ventana de previsualización integrada permite al usuario dar formato al xml, utilizando tanto hojas de estilo en formato CSS como XSL. Su interfaz es muy intuitiva, altamente personalizable e incluye funciones avanzadas de edición como por ejemplo los marcadores (favoritos), auto-sangrado de texto, buscar y reemplazar. Este programa posee: una ayuda para la gestión de proyectos, coloreado de sintaxis xml en tiempo real, vista con forma de árbol, posibilidad de procesar archivos en línea de comando y vista en esquema expansible. Un editor realmente completo y altamente recomendable en caso de que se deba trabajar con este lenguaje xml. [10]

XML Copy Editor

Sistema Operativo: Ubuntu, Gentoo, Fedora.

Idioma: Inglés, Francés, Italiano.

Licencia: GPL.

Descripción:

Es gratuito y capaz de reconocer la sintaxis de un xml. Posee resaltado de sintaxis, auto-completado de etiquetas, facilidad de importación y exportación de documentos.

Kate

Sistema Operativo: Ubuntu.

Idioma: Inglés.

Licencia: GPL.

Descripción:

Completa y valida código xml. Tiene soporte para búsqueda y reemplazo usando expresiones regulares. Puede ampliar sus funcionalidades a través de plugins y scripts.

Conglomerate

Sistema Operativo: Ubuntu.

Idioma: Inglés.

Licencia: GPL.

Descripción:

Completa y valida código xml. Permite "deshacer/rehacer" los cambios realizados a un documento. Puede almacenar una cantidad arbitraria de "deshacer/rehacer" en dependencia de la memoria de la computadora. Se puede resaltar texto y aplicar formato.

1.5.1. Conclusiones parciales

Las soluciones informáticas analizadas anteriormente se limitan a reconocer la estructura del documento, por lo que no satisfacen la necesidad del Alfresco, ya que no están especializadas para interactuar y reconocer las especificidades que el repositorio documental propone para una correcta estructura de un xml para un modelo de contenido.

1.6. Actualidad en Cuba

Constituye un reto para la Universidad de las Ciencias Informáticas la elaboración y puesta en marcha de una herramienta capaz de gestionar xmls como respuesta a la necesidad del proyecto eXcriba.

En la siguiente tabla se muestra una comparación entre la anterior herramienta XMLEditor (v1.0) y la versión 2.0.

Funcionalidades	XMLEditor 1.0	XMLEditor 2.0
Creación de proyectos	Si	Si
Vista previa del xml	Si	Si
Manejo de Aspectos	Crear Aspecto	CRUD ³ a Aspectos
Manejo de Tipos	Crear Tipo de Contenidos	CRUD a Tipos de Contenido
Manejo de Asociaciones	No	CRUD a Asociaciones
Manejo de Restricciones	No	CRUD a Restricciones
Autenticación en Alfresco	No	Si
Importar a Alfresco	No	Automático

Tabla 1.1: Comparación de funcionalidades del XMLEditor

1.7. Metodologías de desarrollo y modelado

1.7.1. RUP

Conocido como Proceso Unificado Racional, es un proceso para el desarrollo de proyectos que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Tiene tres características esenciales:

- **Dirigido por los casos de uso:** Un caso de uso es un fragmento de funcionalidad del sistema que

³Consulte el Glosario de términos.

proporciona al usuario un resultado importante y representan los requisitos funcionales.

- **Centrado en la arquitectura:** Relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden.
- **Iterativo e incremental:** Se divide el proyecto en pequeños proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

Esta metodología permite la adaptación de los procesos a las características propias del sistema. Permite encontrar un balance de los requerimientos del cliente de modo que se puedan satisfacer sus demandas. Permite que varios grupos de trabajo interactúen entre sí y que exista entre ellos la comunicación necesaria para trabajar. Los proyectos son analizados en etapas iteradas permitiendo retroalimentarse de los clientes y así obtener un producto de máxima calidad.[11]

El ciclo de vida de RUP está dividido en cuatro etapas: Inicio, Elaboración, Construcción y Transición. Para más información ver Anexo 1.

En la fase de inicio se identifican los principales casos de uso, identifican los riesgos y el alcance del proyecto.

La fase elaboración consiste en hacer un plan de proyecto y se completan los casos de uso.

La construcción se concentra en la elaboración del software con todos los requisitos que requiere el cliente y el manual de usuario.

La última fase es la de transición, donde se instala el producto y se le da capacitación a los clientes que lo vayan a usar.

Las disciplinas a realizar son :

- **Modelado de negocio:** es donde se entiende la estructura y la dinámica de la empresa del cliente que solicitó los servicios. También se basa en entender el problema actual del cliente para identificar posibles mejoras a los requisitos del software.
- **Requisitos:** reunir los requisitos de los usuarios.
- **Análisis y Diseño:** diseño de software requisitos en los modelos uml.

- **Aplicación:** se aplica la solución de software en componentes de software y componentes de base de datos.
- **Prueba:** la prueba de componentes construidos.
- **Implementación:** instalación de componentes y usuarios.
- **Configuración y gestión del cambio:** mantener y gestionar los artefactos producidos por el software del equipo.
- **Gestión de proyectos:** gestionar, planificar y distribuir las actividades del equipo, centrándose en la calidad, calendario, presupuesto y el riesgo de ataque.
- **Medio ambiente:** mantener las herramientas de configuración y aspectos ambientales para el equipo de desarrollo.

Uno de los aspectos que diferencian a RUP de las demás metodologías y lo que lo hace único es que los casos de uso no son sólo una herramienta para especificar los requisitos del sistema, sino que también guían su diseño, implementación y prueba. Los casos de uso constituyen un elemento integrador y una guía del trabajo.

Durante todo el ciclo de vida del proyecto es necesario vigilar el cumplimiento de los objetivos, restricciones y gestionar los riesgos.

Estos son algunos artefactos generados al aplicar RUP:

- Documento visión.
- Especificación de requisitos.
- Diagramas de caso de uso.
- Documento arquitectura.
- Diagrama de clases.
- Modelo Entidad-Relación.

- Diagrama de secuencia.
- Diagrama de estados.
- Diagrama de colaboración.
- Vista conceptual.
- Modelo de dominio.
- Mapa de comportamiento a nivel de hardware.

Ejemplo de roles: analista de sistema, arquitecto de software, diseñador de interfaz de usuario y analista de pruebas.

1.7.2. Conclusiones parciales

Aplicando RUP en la realización de la herramienta XMLEditor v2.0 se obtiene una gran cantidad de artefactos que describen el proceso de desarrollo del software. La documentación realizada puede ser de gran utilidad para futuras versiones de la aplicación. Los procesos son más controlados porque deben seguir pasos lógicos. Además esta metodología se utiliza en el proyecto eXcriba.

1.8. Lenguaje de modelado

1.8.1. UML

El lenguaje unificado de modelado (UML), es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Grupo de Administración de Objetos). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. [12]

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos, documentarlos y construirlos. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.[13]

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, pues se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.[14]

1.9. Lenguaje de programación e IDE

Un lenguaje de programación es un idioma artificial diseñado para expresar cálculos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación.[15]

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Java

Java es un lenguaje de programación orientado a objetos, que fue desarrollado por la empresa Sun Microsystems en la década del '90 del siglo XX. Este lenguaje toma muchas características de los lenguajes C y C++ y a su vez mejora la manipulación de los punteros y la memoria. Se creó en base a cinco principios:

debería ser orientado a objetos, multiplataforma, con soporte para trabajo en red y seguridad al ejecutarse de forma remota.

Java es compilado e interpretado. Fue diseñado para crear software altamente fiable, se implementaron barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real, para evitar que los applets⁴ tuvieran acceso total a los recursos del ordenador donde se ejecutan. Se especifican los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas, los cuales están implementados en la máquina virtual java (JVM).[16]

Soporta sincronización multihilos que le permite ejecutar varias acciones a la misma vez. Las clases son definidas a medida de que vayan siendo usadas. Puede ser usado para crear dos tipos de programas: aplicaciones independientes y/o applets. [17]

Lenguaje Extensible de Maquetado: XML

El lenguaje de maquetado extensible, proviene de un lenguaje inventado por IBM⁵ en los años setenta, llamado GML (Generalized Markup Language). Surge por la necesidad de almacenar grandes cantidades de información. Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C) que es una empresa muy destacada en el desarrollo de estándares a usar en la Red de Redes. Es una adaptación bastante simple del SGML⁶ que permite ser usado para definir la gramática de algunos lenguajes. Es muy importante porque permite compartir la información de una manera fácil. Está formado por un encabezado y por el cuerpo del documento así como texto de etiquetas. [18]

Entre las ventajas del uso del xml se puede mencionar:

1. Es extensible: Es posible extender xml con la incorporación de nuevas etiquetas.
2. Es sencillo de entender su estructura y procesarla.

1.10. Conclusiones del capítulo

Con el estudio realizado sobre el tema, se lograron identificar los distintos editores de xmls existentes a nivel internacional. Tras delimitar las fronteras, características y ventajas de estos, se definieron las

⁴Es un componente de una aplicación que se ejecuta en el contexto de otro programa

⁵Consulte el glosario de términos

⁶Estándar de Lenguaje de Marcado Generalizado

metodologías de desarrollo del software y herramientas que se utilizarán en el desarrollo de la solución que se propone, teniendo como resultado las siguientes:

Herramienta CASE: Visual Paradigm.

Metodología de desarrollo de software: Proceso Unificado de Desarrollo (RUP).

Lenguaje de programación: Java, XML.

Después de la realización de un estudio sobre las presentes metodologías y herramientas, quedan sentadas las bases para el desarrollo del módulo.

Capítulo 2

Características del sistema.

2.1. Introducción

En el presente capítulo se realiza un análisis de las características del sistema, así como de los requisitos funcionales y requisitos no funcionales. Se definirán y describirán los casos de usos del sistema, sus actores y los artefactos generados.

Objeto de estudio

Extender una herramienta informática que sea capaz de gestionar los Modelos de Contenidos de forma que incorpore nuevas funcionalidades a la herramienta XMLEditor v1.0 de modo que aproveche todas las potencialidades del diccionario de datos de Alfresco.

Objeto de automatización

Será objeto de automatización la gestión de los Modelos de Contenidos por los usuarios que harán uso del repositorio documental.

Aquellas personas que deseen obtener información sobre el flujo de creación de un modelo de contenidos debe primero leer el documento Manual de Usuario o la ayuda que brinda el propio módulo.

La implementación de las nuevas funcionalidades se encargará de integrar al módulo aquellos aspectos que no se han automatizado y que son necesarios para un correcto aprovechamiento del diccionario de datos de Alfresco.

Información que se maneja

La información que se maneja respecto a un modelo de contenido es para describir las tipologías documentales que se generen en una institución, ejemplo: Actas de Rectoría o Actas de la UJC.

Propuesta del Sistema

Se propone desarrollar el módulo XMLEditor en su versión 2.0 añadiéndole las nuevas funcionalidades: gestionar aspectos, gestionar tipos de contenidos, gestionar asociaciones, gestionar restricciones a los metadatos y exportar Modelos de Contenidos al ECM Alfresco.

2.2. Especificación de los requisitos de software

El software que se propone en este documento es para la creación de Modelos de Contenidos con tipos documentales simples y complejos para Alfresco, aprovechando las ventajas de la importación de Modelos de Contenidos a través de xml.

Requerimientos funcionales

Los requisitos funcionales (RF) son las pautas que el software debe cumplir. Aquí se establece lo que debe hacer el módulo a desarrollar.

RF 1 Gestionar Aspecto.

- **RF 1.1** Adicionar Aspecto. El usuario tendrá la posibilidad de añadir un aspecto al modelo de contenido que está diseñando.
- **RF 1.2** Listar Aspecto. El podrá ver una lista de los diferentes aspectos que ha creado.
- **RF 1.3** Modificar Aspecto. El usuario podrá modificar los aspectos que ha creado y corregir sus datos.
- **RF 1.4** Eliminar Aspecto. El usuario podrá eliminar los aspectos creados.

RF 2 Gestionar Tipo de Contenido.

- **RF 2.1** Adicionar Tipo de Contenido. Para adicionar un nuevo tipo de contenido el usuario podrá usar la opción que el módulo le brinda específicamente para esta acción.
- **RF 2.2** Listar Tipos de Contenido. El usuario podrá ver una lista de los tipos de contenidos existentes en el modelo de contenidos que está realizando.
- **RF 2.3** Modificar Tipo de Contenido. El módulo le permitirá al usuario realizar modificaciones a los tipos de contenidos que ha creado.

- **RF 2.4** Eliminar Tipo de Contenido. El módulo pondrá a disposición del usuario la opción de eliminar los tipos de contenidos que haya creado.

RF 3 Gestionar Asociaciones.

- **RF 3.1** Adicionar Asociaciones. El módulo pondrá a disposición del usuario la opción de agregar una asociación.
- **RF 3.2** Listar Asociaciones. El módulo mostrará una lista de las asociaciones creadas por el usuario.
- **RF 3.3** Modificar Asociaciones. El usuario dispondrá de una opción para modificar las asociaciones creadas por él.
- **RF 3.4** Eliminar Asociaciones. El usuario dispondrá de una opción para la eliminación de las asociaciones.

RF 4 Gestionar Restricciones de metadatos asociados a tipos documentales.

- **RF 4.1** Adicionar Restricción. Para insertar restricciones a los metadatos asociados a tipos documentales por el usuario.
- **RF 4.2** Modificar Restricción. Para modificar las restricciones a los metadatos asociados a tipos documentales.
- **RF 4.3** Listar Restricciones. El módulo hará una lista de las diferentes restricciones a los metadatos asociados a tipos documentales que haya creado el usuario.
- **RF 4.4** Eliminar Restricción. El usuario podrá eliminar cada una de las restricciones a los metadatos asociados a tipos documentales.

RF 5 Exportar a Alfresco. El módulo pondrá a disposición del usuario una opción para Exportar los Modelos de Contenidos creados a Alfresco.

RF 6 Autenticar en Alfresco. Para que el módulo pueda exportar los Modelos de Contenidos necesita que el usuario se autentique en Alfresco.

Requisitos no funcionales

Los requisitos no funcionales son aquellos que garantizan las condiciones óptimas para el uso del producto, por tanto influye de manera directa en el rendimiento y usabilidad.

Software:

- El módulo se implementará con tecnología Java.
- Máquina virtual de Java versión 1.5 o superior.

Soporte:

- El responsable es el grupo de soporte del proyecto eXcriba.

Restricciones de Diseño e Implementación:

- El lenguaje de programación a ser usado para implementar la aplicación será Java.

Apariencia o interfaz:

- La interfaz externa de la aplicación debe ser amigable, sencilla y fácil de usar por usuarios finales, facilitando el control de las operaciones sin necesidad de mucho entrenamiento para su uso.
- La aplicación estará estructurada de forma clara y comprensible, al mismo tiempo permitirá la interpretación correcta e inequívoca de la información.
- El diseño responderá a la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso.

Mantenibilidad:

- Se utilizarán los estándares establecidos para la creación de Modelos de Contenidos, según los tipos de documentos y el formato específico para ello.

Portabilidad:

- El módulo será independiente y multiplataforma, lo que significa que el mismo podrá ser usado sobre cualquier sistema operativo.

Legales:

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

2.3. Descripción del sistema propuesto

El módulo se implementará de forma que el usuario con un nivel básico de conocimientos en informática y que esté especializado en la realización de Modelos de Contenidos pueda utilizar el software para crear los xmls que definen una tipología documental en el Alfresco. El software se desarrollará haciendo uso de las tecnologías y metodologías expuestas en el capítulo anterior.

Descripción de los actores del sistema

Actor	Descripción
Usuario	Persona que desee crear un modelo de contenidos y gestionar las característica que este posee.
Usuario Alfresco	Persona que posea permisos para exportar Modelos de Contenidos a Alfresco.

Tabla 2.1: Definición de actores del sistema

Diagrama de Casos de Uso del Sistema

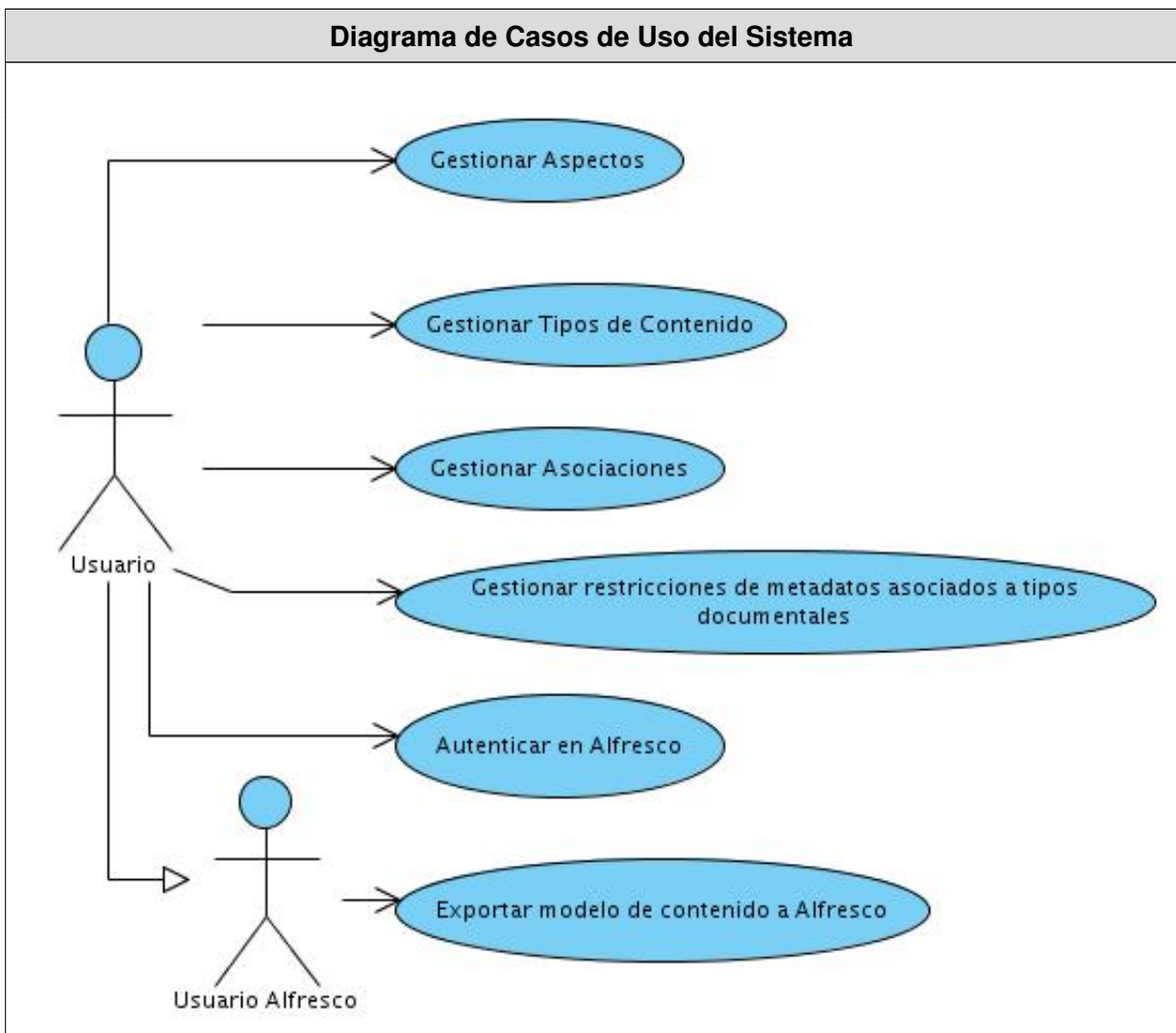


Figura 2.1: Diagrama de Casos de Uso del Sistema

2.4. Especificación de los Casos de uso del sistema

2.4.1. Gestionar Aspecto

Caso de uso	Gestionar Aspecto	
Actor	Usuario	
Resumen	El caso de uso inicia cuando el actor accede a las opciones para gestionar los aspectos. El actor especifica los datos que son solicitados por el módulo de acuerdo a la opción seleccionada. El módulo registra los datos finalizando así el caso de uso.	
Precondiciones	Que el usuario haya creado un proyecto. Que se hayan definido los datos generales del modelo.	
Referencias	RF: 1.1, 1.2, 1.3, 1.4	
Prioridad	Crítico.	
Flujo normal de eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario selecciona el panel "Aspectos".	2. El módulo muestra las opciones que se pueden realizar para gestionar los aspectos.	
3. El usuario seleccionará una de las siguientes opciones: <ul style="list-style-type: none"> ■ Agregar aspecto. Ver sección "Agregar aspecto". ■ Modificar aspecto. Ver sección "Modificar aspecto". ■ Listar aspecto. Ver sección "Listar aspecto". ■ Eliminar aspecto. Ver sección "Eliminar aspecto". 		
Continúa en la próxima página		

Flujo Normal de Eventos	
Sección: “Agregar aspecto”	
Acción del Actor	Respuesta del Sistema
1. El usuario desea crear un aspecto.	2. El módulo muestra los siguientes datos contextuales necesarios para la creación del aspecto: <ul style="list-style-type: none"> ■ Nombre. ■ Título. ■ Padre. ■ Prefijo.
3. El usuario introduce los datos.	4. El módulo verifica la entrada de datos y campos obligatorios.
	5. El módulo crea un nuevo aspecto con los datos que el usuario insertó, de esta forma finaliza el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 El módulo identifica datos erróneos o campos obligatorios vacíos.
	4.2 El módulo muestra un mensaje de error. Remitirse al paso 2 del flujo normal de eventos.
Flujo Normal de Eventos	
Sección: “Modificar aspecto”	
Continúa en la próxima página	

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona el aspecto y elige la acción modificar aspecto.	2. El módulo muestra los siguientes datos contextuales necesarios para la modificación del aspecto: <ul style="list-style-type: none"> ■ Nombre. ■ Título. ■ Padre. ■ Prefijo.
3. El usuario introduce los datos.	4. El módulo verifica la entrada de datos y campos obligatorios.
	5. El módulo modifica el aspecto con los datos que el usuario insertó, de esta forma finaliza el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 El módulo identifica datos erróneos o campos obligatorios vacíos.
	4.2 El módulo muestra un mensaje de error. Remitirse al paso 2 del flujo normal de eventos.
Flujo Normal de Eventos	
Sección: “Listar aspecto”	
Acción del Actor	Respuesta del Sistema
Continúa en la próxima página	

1. El usuario selecciona el panel "Aspectos".	2. El módulo muestra una lista de los aspectos creados por el usuario, de esta forma finaliza el caso de uso.
Flujo Normal de Eventos	
Sección: "Eliminar aspecto"	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona el aspecto y elige la opción eliminar aspecto.	2. El módulo elimina el aspecto seleccionado, de esta forma finaliza el caso de uso.

Tabla 2.3: Descripción textual del CU: Gestionar Aspecto

2.4.2. Gestionar Tipo de Contenido

Caso de uso	Gestionar Tipo
Actor	Usuario
Resumen	El caso de uso inicia cuando el actor accede a las opciones para gestionar los tipos de contenidos. El actor especifica los datos que son solicitados por el módulo de acuerdo a la opción seleccionada. El módulo registra los datos finalizando así el caso de uso.
Precondiciones	Que el usuario haya creado un proyecto. Que se hayan definido los datos generales del modelo.
Referencias	RF: 2.1, 2.2, 2.3, 2.4
Prioridad	Crítico.
Flujo normal de eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona el panel "Tipos".	2. El módulo muestra las opciones necesarias para gestionar los tipos de contenidos.
Continúa en la próxima página	

<p>3. El usuario seleccionará una de las siguientes opciones:</p> <ul style="list-style-type: none"> ■ Agregar tipo de contenido. Ver sección "Agregar tipo de contenido". ■ Modificar tipo de contenido. Ver sección "Modificar tipo de contenido". ■ Listar tipo de contenido. Ver sección "Listar tipo de contenido". ■ Eliminar tipo de contenido. Ver sección "Eliminar tipo de contenido". 	
Flujo Normal de Eventos	
Sección: "Agregar tipo de contenido"	
Acción del Actor	Respuesta del Sistema
<p>1. El usuario desea crear un tipo de contenido.</p>	<p>2. El módulo muestra los siguientes datos contextuales necesarios para la creación del tipo de contenido:</p> <ul style="list-style-type: none"> ■ Nombre. ■ Título. ■ Padre. ■ Prefijo.
Continúa en la próxima página	

3. El usuario introduce los datos.	4. El módulo verifica la entrada de datos y campos obligatorios.
	5. El módulo crea un nuevo tipo con los datos que el usuario insertó, de esta forma finaliza el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 El módulo identifica datos erróneos o campos obligatorios vacíos.
	4.2 El módulo muestra un mensaje de error. Remitirse al paso 2 del flujo normal de eventos.
Flujo Normal de Eventos	
Sección: “Modificar tipo”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción: modificar.	2. El módulo muestra un listado de tipos y los siguientes datos contextuales necesarios para la modificación del tipo: <ul style="list-style-type: none"> ■ Nombre. ■ Título. ■ Padre. ■ Prefijo.
3. El usuario selecciona el tipo a modificar	4. El módulo verifica la entrada de datos y campos obligatorios.
Continúa en la próxima página	

	5. El módulo modifica tipo de contenido con los datos que el usuario insertó, de esta forma finaliza el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 El módulo identifica datos erróneos o campos obligatorios vacíos.
	4.2 El módulo muestra un mensaje de error. Remitirse al paso 2 del flujo normal de eventos.
Flujo Normal de Eventos	
Sección: “Listar tipos de contenidos”	
Acción del Actor	Respuesta del Sistema
1. El usuario despliega la caja de opciones (ComboBox) llamada “Padre.”	2. El módulo muestra una lista de los tipos creados por el usuario y las predefinidas, de esta forma finaliza el caso de uso.
Flujo Normal de Eventos	
Sección: “Eliminar tipos de contenido”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción: eliminar.	2. El módulo muestra el listado de tipos de contenidos.
3. El usuario selecciona el tipo de contenido que desea eliminar y pulsa el botón: Eliminar	4. El módulo elimina el tipo de contenido seleccionado, de esta forma finaliza el caso de uso.
Continúa en la próxima página	

Tabla 2.4: Descripción textual del CU: Gestionar Tipos de contenidos

2.4.3. Gestionar Asociaciones

Caso de uso	Gestionar Asociaciones	
Actor	Usuario	
Resumen	El caso de uso inicia cuando el actor accede a las opciones para gestionar las asociaciones. El actor especifica los datos que son solicitados por el módulo de acuerdo a la opción seleccionada. El módulo registra los datos finalizando así el caso de uso.	
Precondiciones	Que el usuario haya creado un proyecto y existan al menos dos tipos de contenidos creados. Que se hayan definido los datos generales del modelo.	
Referencias	RF: 3.1, 3.2, 3.3, 3.4	
Prioridad	Crítico	
Flujo normal de eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario selecciona el panel "Asociaciones".	2. El módulo muestra las opciones que se pueden realizar para gestionar las asociaciones.	
Continúa en la próxima página		

<p>3. El usuario selecciona una de las siguientes opciones:</p> <ul style="list-style-type: none"> ■ Agregar asociación. Ver sección "Agregar asociación". ■ Modificar asociación. Ver sección "Modificar asociación". ■ Listar asociación. Ver sección "Listar asociación". ■ Eliminar asociación. Ver sección "Eliminar asociación". 	
Flujo Normal de Eventos	
Sección: "Agregar asociación"	
Acción del Actor	Respuesta del Sistema
Continúa en la próxima página	

<p>1. El usuario selecciona la opción: agregar asociación.</p>	<p>2. El módulo muestra los siguientes datos contextuales necesarios para la creación de la asociación:</p> <ul style="list-style-type: none"> ■ Nombre. ■ Aplicar a. ■ Asociación padre-hijo. ■ Origen obligatorio. ■ Origen relación 1..*. ■ Clase. ■ Destino obligatorio. ■ Destino relación 1..*.
<p>3. El usuario introduce los datos.</p>	<p>4. El módulo verifica la entrada de datos y campos obligatorios.</p>
	<p>5. El módulo crea un nueva asociación con los datos que el usuario insertó, de esta forma finaliza el caso de uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>4.1 El módulo identifica datos erróneos o campos obligatorios vacíos.</p>
	<p>4.2 El módulo muestra un mensaje de error. Remitirse al paso 2 del flujo normal de eventos.</p>
<p>Continúa en la próxima página</p>	

Flujo Normal de Eventos	
Sección: “Modificar asociación”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la asociación y elige la opción: modificar.	2. El módulo muestra los siguientes datos contextuales necesarios para la modificación de la asociación: <ul style="list-style-type: none"> ■ Nombre. ■ Aplicar a. ■ Asociación padre-hijo. ■ Origen obligatorio. ■ Origen relación 1..*. ■ Clase. ■ Destino obligatorio. ■ Destino relación 1..*.
3. El usuario introduce los datos.	4. El módulo verifica la entrada de datos y campos obligatorios.
	5. El módulo modifica la asociación con los datos introducidos, de esta forma finaliza el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Continúa en la próxima página	

	4.1 El módulo identifica datos erróneos o campos obligatorios vacíos.
	4.2 El módulo muestra un mensaje de error. Remitirse al paso 2 del flujo normal de eventos.
Flujo Normal de Eventos	
Sección: “Listar asociación”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona el panel “Asociaciones”	2. El módulo muestra una lista de las asociaciones creadas por el usuario, de esta forma finaliza el caso de uso.
Flujo Normal de Eventos	
Sección: “Eliminar asociación”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la asociación que quiere eliminar y elige la opción eliminar asociación.	2. El módulo elimina la asociación con todos los datos que este contiene, de esta forma finaliza el caso de uso.

Tabla 2.5: Descripción textual del CU: Gestionar Asociaciones

2.4.4. Gestionar Restricciones de los metadatos asociados a tipos documentales

Caso de uso	Gestionar restricciones de los metadatos asociados a tipos documentales
Actor	Usuario
Continúa en la próxima página	

Resumen	El caso de uso inicia cuando el actor accede a las opciones para gestionar las restricciones asociadas a tipos documentales. El actor especifica los datos que son solicitados por el módulo de acuerdo a la opción seleccionada. El módulo registra los datos finalizando así el caso de uso.	
Precondiciones	Que el usuario haya creado un proyecto. Que se hayan definido los datos generales del modelo.	
Referencias	RF: 4.1, 4.2, 4.3, 4.4	
Prioridad	Crítico	
Flujo normal de eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario selecciona el panel "Restricciones".	2. El módulo muestra las opciones que se pueden realizar para gestionar las restricciones de los metadatos asociados a tipos documentales.	
Continúa en la próxima página		

<p>3. El usuario selecciona una de las siguientes opciones:</p> <ul style="list-style-type: none"> ■ Agregar restricción. Ver sección "Agregar restricción". ■ Modificar restricción. Ver sección "Modificar restricción". ■ Listar restricción. Ver sección "Listar restricción". ■ Eliminar restricción. Ver sección "Eliminar restricción". 	
Flujo Normal de Eventos	
Sección: "Agregar restricción"	
Acción del Actor	Respuesta del Sistema
<p>1. El usuario selecciona la opción agregar restricción.</p>	<p>2. El módulo muestra los siguientes datos contextuales necesarios para la creación de la restricción:</p> <ul style="list-style-type: none"> ■ Nombre de la restricción. ■ Tipo de dato. ■ Valores en correspondencia al Tipo de dato.
<p>3. El usuario introduce los datos.</p>	<p>4. El módulo verifica la entrada de datos y campos obligatorios.</p>
<p>Continúa en la próxima página</p>	

	5. El módulo crea un nueva restricción con los datos que el usuario insertó, de esta forma finaliza el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 El módulo identifica datos erróneos o campos obligatorios vacíos.
	4.2 El módulo muestra un mensaje de error. Remitirse al paso 2 del flujo normal de eventos.
Flujo Normal de Eventos	
Sección: “Modificar restricción”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la restricción y elije la opción: modificar.	2. El módulo muestra los siguientes datos contextuales necesarios para la modificación de la restricción: <ul style="list-style-type: none"> ■ Nombre de la restricción. ■ Tipo de dato de la restricción. ■ Valores en correspondencia al Tipo de Dato.
3. El usuario introduce los datos	4. El módulo verifica la entrada de datos y campos obligatorios.
	5. El módulo modifica la restricción seleccionada, de esta forma finaliza el caso de uso.
Continúa en la próxima página	

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 El módulo identifica datos erróneos o campos obligatorios vacíos.
	4.2 El módulo muestra un mensaje de error. Remitirse al paso 2 del flujo normal de eventos.
Flujo Normal de Eventos	
Sección: “Listar restricción”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona el panel “Restricción”	2. El módulo muestra una lista de las restricciones creadas por el usuario, de esta forma finaliza el caso de uso.
Flujo Normal de Eventos	
Sección: “Eliminar restricción”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la restricción y elige la opción eliminar restricción.	2. El módulo elimina la restricción con todos los datos que este contiene, de esta forma finaliza el caso de uso.

Tabla 2.6: Descripción textual del CU: Gestionar restricciones a los metadatos asociados a tipos documentales.

2.4.5. Exportar a Alfresco

Caso de uso	Exportar a Alfresco
	Continúa en la próxima página

Actor	Usuario Alfresco	
Resumen	El caso de uso inicia cuando el usuario desea exportar uno o varios modelos de contenido a Alfresco, finalizando cuando se ejecuta la acción deseada.	
Precondiciones	Que el usuario haya creado uno o varios proyectos y que esté autenticado en Alfresco. Que se hayan definido los datos generales del modelo.	
Referencias	RF: 5	
Prioridad	Crítico.	
Flujo normal de eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario selecciona la opción: Exportar en el menú del XMLEditor	2. El módulo muestra un listado de los Modelos de Contenidos que se pueden exportar.	
3. El usuario selecciona los Modelos de Contenidos que deseada exportar.	4. El módulo exporta los Modelos de Contenidos seleccionados a Alfresco.	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	4.1 El módulo detecta algún error y retorna al estado inicial del caso de uso.	

Tabla 2.7: Descripción textual del CU: Exportar a Alfresco.

2.4.6. Autenticar en Alfresco

Caso de uso	Autenticar en Alfresco	
Actor	Usuario	
Resumen	El caso de uso comienza cuando el Usuario Alfresco decide exportar algún modelo de contenido y finaliza cuando el módulo verifica los datos de autenticación.	
Referencias		
Precondiciones	La computadora del Usuario Alfresco tenga conexión al servidor de Alfresco.	
Flujo normal de eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario elige la opción: Exportar, en el menú del XMLEditor.	2. El módulo muestra los datos necesarios para la autenticación en Alfresco: <ul style="list-style-type: none"> ▪ Usuario. ▪ Contraseña. ▪ URL. ▪ Puerto. 	
3. El usuario introduce los datos necesarios para autenticarse.	4. El módulo verifica los datos e intenta conectarse a Alfresco.	
Flujo Alternativo		
Acción del Actor	Respuesta del Sistema	
	4.1 El módulo identifica datos erróneos y muestra un mensaje de error al usuario dándole una sugerencia del fallo. Remitirse al paso 2 del flujo normal de eventos.	
Continúa en la próxima página		

Prioridad	Crítico
------------------	---------

Tabla 2.8: Descripción textual del caso de uso: Autenticar en Alfresco.

2.5. Conclusiones del capítulo

En este capítulo se llevaron a cabo acciones que facilitarán el desarrollo de un módulo que satisfaga las necesidades y exigencias del cliente. Se realizó la modelación del negocio, el levantamiento de requisitos y la modelación del módulo, incluyendo los diagramas de casos de usos y las descripciones textuales de los casos de uso del sistema, los cuales permitirán una mayor visibilidad de la solución propuesta.

Capítulo 3

Análisis y diseño del sistema.

Introducción

En el presente capítulo se modelará el sistema. Para el cumplimiento de dicha tarea se expondrán diagramas de clases de diseño, así como la realización de los casos de uso correspondientes al flujo de trabajo de diseño.

Los objetivos que se pretenden alcanzar en este capítulo son:

- Adquirir una comprensión detallada de los requisitos funcionales.
- Dividir el trabajo de implementación en partes que puedan ser realizadas por diferentes personas.

3.1. Diagramas de paquetes

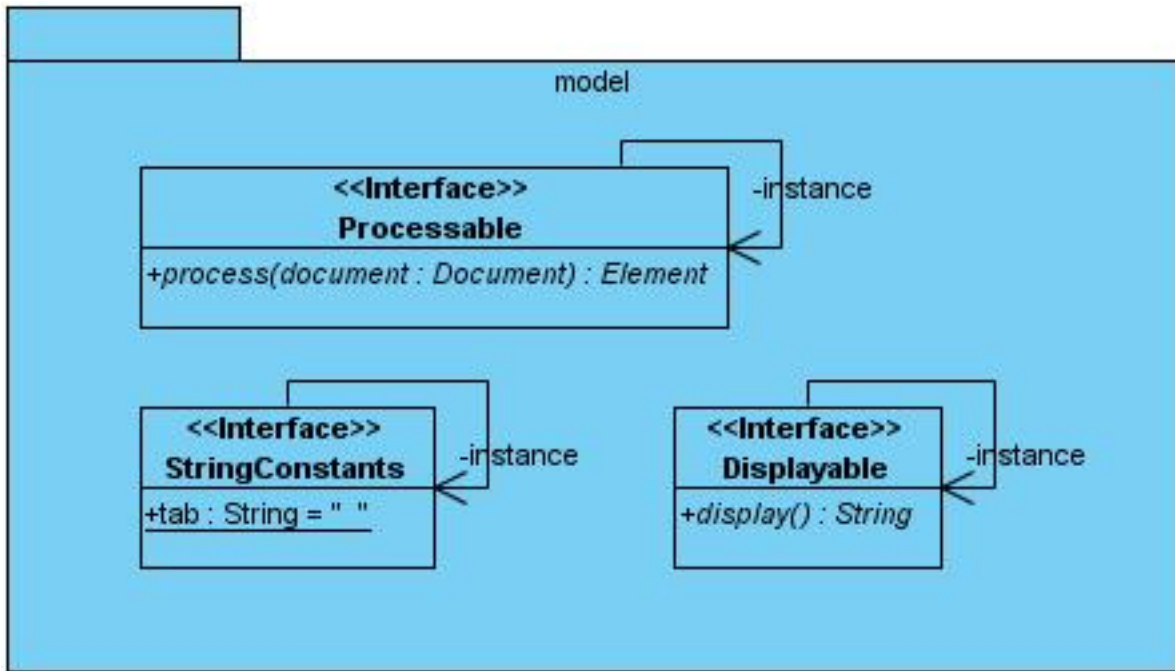


Figura 3.1: Paquete model

En el paquete se encuentran agrupadas varias clases que servirán de apoyo a las clases entidades del sistema. Las interfaces Displayable y Processable contienen los métodos necesarios para poder mostrar y procesar los tipos de contenidos.

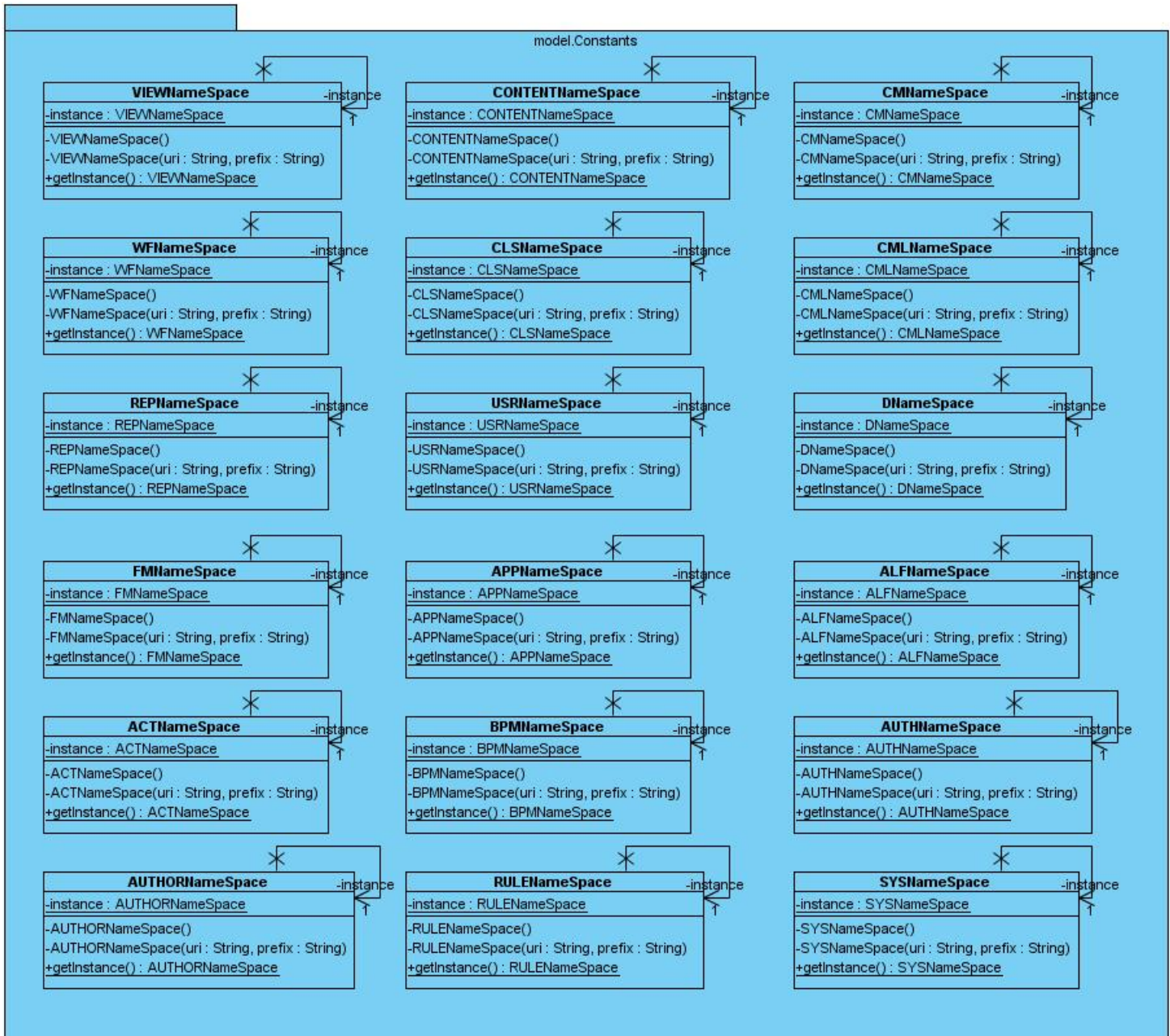


Figura 3.2: Paquete model.Constants

Este paquete contiene las clases necesarias para poder simular los NameSpaces definidos por Alfresco

y otros definidos por el Usuario, los cuales permitirán que los nuevos tipos de contenidos hereden las características y comportamientos de los tipos definidos en cada uno de dichos NameSpaces.

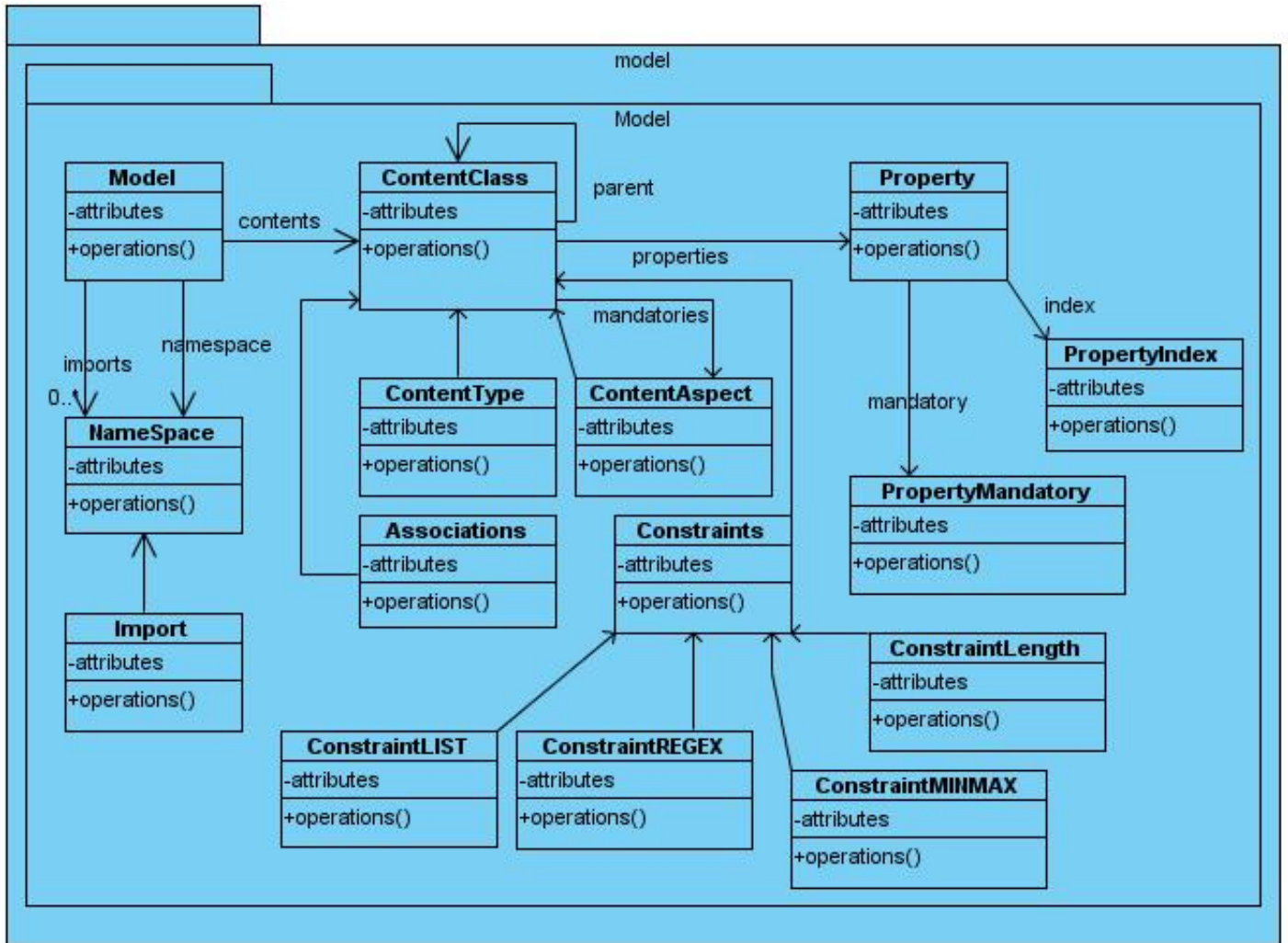


Figura 3.3: Paquete model.Model

El paquete contiene las clases necesarias para modelar realmente un modelo de contenido al estilo de Alfresco. Dentro del mismo, las clases fundamentales son ContentClass, Property, Model, Associations y Constraints, las cuales permitirán crear los tipos de contenido, aspectos, propiedades, asociaciones,

restricciones y el modelo de contenido en sí. Además se incluyen clases que permitirán un manejo más fácil para lograr la creación de dichos Modelos de Contenidos.

3.2. Diagramas de clases.

Un diagrama de clases es una colección estática de los elementos declaratorios del modelo, como clases, tipos y sus relaciones, conectados unos a otros y a sus contenidos. Modela los conceptos del dominio de la aplicación, así como los conceptos internos generados siendo parte de la implementación de la aplicación. Es un paso esencial en el análisis orientado a objetos.

Diagramas de clases del diseño

Los diagramas de clases del diseño describen la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

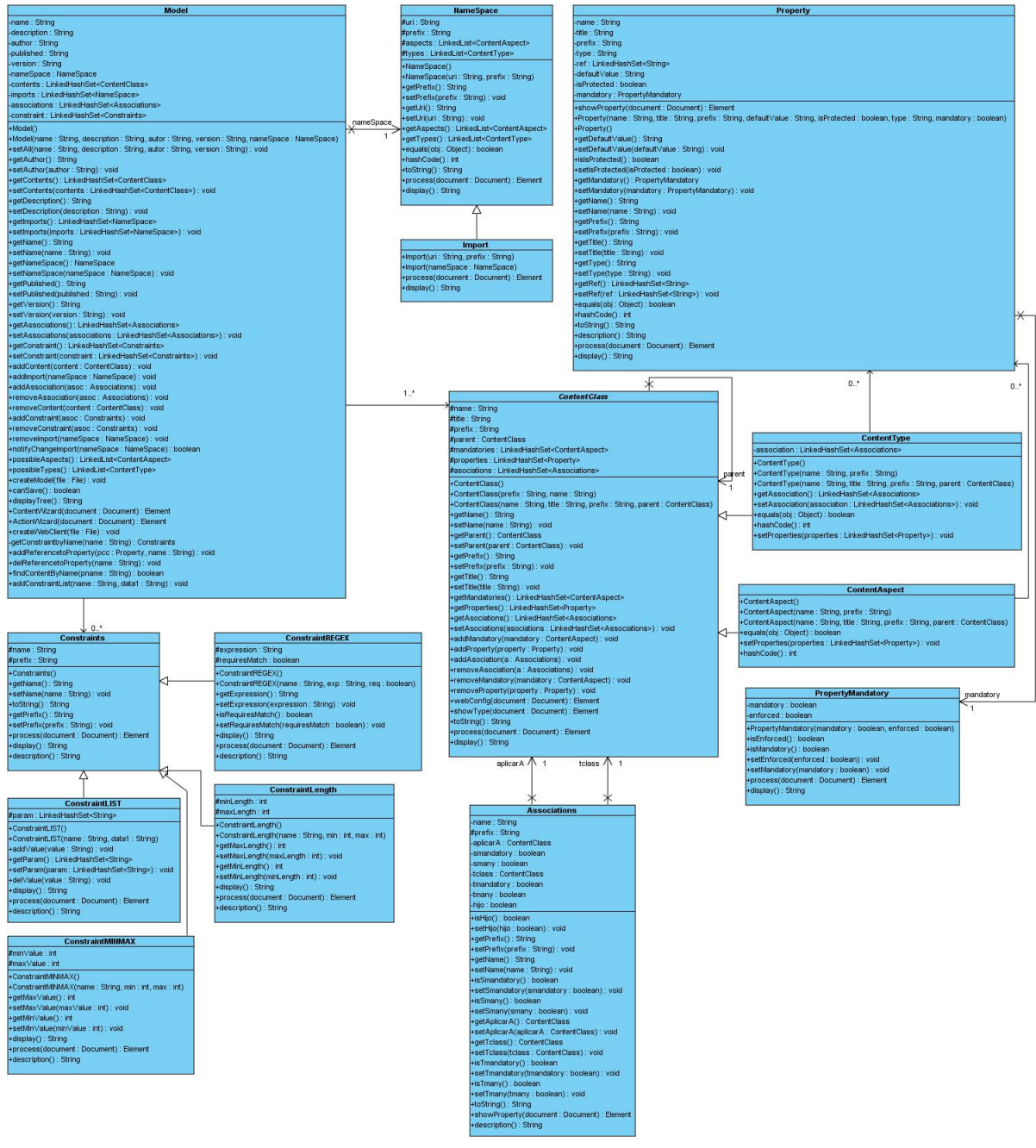


Figura 3.4: Diagrama de clases del sistema

3.3. Descripción de las principales clases

A continuación se describen las principales clases que modelan el sistema.

Nombre: Model	
Tipo de Clase: Entidad	
Atributo	Tipo
name	String
description	String
author	String
published	String
version	String
nameSpace	NameSpace
contents	LinkedHashSet<ContentClass>
imports	LinkedHashSet<NameSpace>
associations	LinkedHashSet<Associations>
constraint	LinkedHashSet<Constraints>
Principales responsabilidades	
Nombre:	<code>public Model(String name, String description, String autor, String version, NameSpace nameSpace)</code>
Descripción:	Constructor de la clase, crea una instancia de la clase a partir de los datos especificados.
Nombre:	<code>addContent(ContentClass content)</code>
Descripción:	Adiciona un nuevo contenido o aspecto al modelo de contenido que se está creando.
Nombre:	<code>addImport(NameSpace nameSpace)</code>
Continúa en la próxima página	

Descripción:	Adiciona un nuevo nameSpace al modelo contenido para importar sus contenidos y aspectos.
Nombre:	removeContent(ContentClass content)
Descripción:	Elimina un contenido o aspecto del modelo de contenido que se está creando.
Nombre:	createModel(File file)
Descripción:	Crea el modelo de contenido en la dirección especificada por el archivo seleccionado como parámetro.
Nombre:	canSave()
Descripción:	Determina si el modelo de contenido está listo para ser guardado.
Nombre:	displayTree()
Descripción:	Devuelve en una cadena el modelo de contenido que se está creando.
Nombre:	createWebClient(File file)
Descripción:	Crea en el archivo especificado como parámetro la descripción del cliente web.
Nombre:	addAssociation(Associations asoc)
Descripción:	Adiciona una nueva asociación al modelo de contenido que se está creando.
Nombre:	removeAssociation(Associations asoc)
Descripción:	Elimina una asociación del modelo de contenido que se está creando.
Nombre:	addConstraint(Constraints asoc)
Descripción:	Adiciona una nueva restricción en el modelo de contenido que se está creando.
Nombre:	removeConstraint(Constraints asoc)
Descripción:	Elimina una restricción del modelo de contenido que se está creando.

Tabla 3.1: Descripción de la clase Model

Nombre: Associations
Continúa en la próxima página

Tipo de Clase: Entidad	
Atributo	Tipo
name	String
prefix	String
aplicarA	ContentClass
smandatory	boolean
smany	boolean
tclass	ContentClass
tmandatory	boolean
tmany	boolean
hijo	boolean
Principales responsabilidades	
Nombre:	showProperty(Document document)
Descripción:	Retorna el tipo de la asociación implícita en ésta clase.
Nombre:	description()
Descripción:	Retorna la descripción de la asociación implícita en esta clase.

Tabla 3.2: Descripción de la clase Associations

Nombre: Constraints	
Tipo de Clase: Entidad	
Atributo	Tipo
name	String
prefix	String
Principales responsabilidades	
Continúa en la próxima página	

Nombre:	process(Document document)
Descripción:	Devuelve el elemento en representación XML para ser adicionado al modelo de contenido que se esté creando. Este método es reemplazado por su equivalente en las clases que heredan de él.
Nombre:	display()
Descripción:	Devuelve una cadena con la representación XML del objeto en cuestión . Este método es reemplazado por su equivalente en las clases que heredan del él.
Nombre:	description()
Descripción:	Retorna la descripción de la restricción implícita en esta clase. Este método es reemplazado por su equivalente en las clases que heredan del él.

Tabla 3.3: Descripción de la clase Constraints

Nombre: Property	
Tipo de Clase: Entidad	
Atributo	Tipo
name	String
prefix	String
title	String
type	String
ref	LinkedHashSet<String>
defaultValue	String
isProtected	boolean
mandatory	PropertyMandatory
Principales responsabilidades	
Continúa en la próxima página	

Nombre:	process(Document document)
Descripción:	Devuelve el elemento en representación XML para ser adicionado al modelo de contenido que se esté creando.
Nombre:	display()
Descripción:	Devuelve una cadena con la representación XML del objeto en cuestión.
Nombre:	description()
Descripción:	Retorna la descripción de la restricción implícita en esta clase.

Tabla 3.4: Descripción de la clase Property

3.4. Diagramas de colaboración

Los diagramas de colaboración muestran las interacciones que ocurren entre los objetos que participan en una situación determinada donde se enfatizan las relaciones entre los objetos y su topología. Los mensajes enviados de un objeto a otro se representan mediante flechas, mostrando el nombre del mensaje, los parámetros y la secuencia del mensaje. Los diagramas de colaboración están indicados para mostrar una situación o flujo de un programa en específico, demostrar y explicar rápidamente un proceso dentro de la lógica del programa.

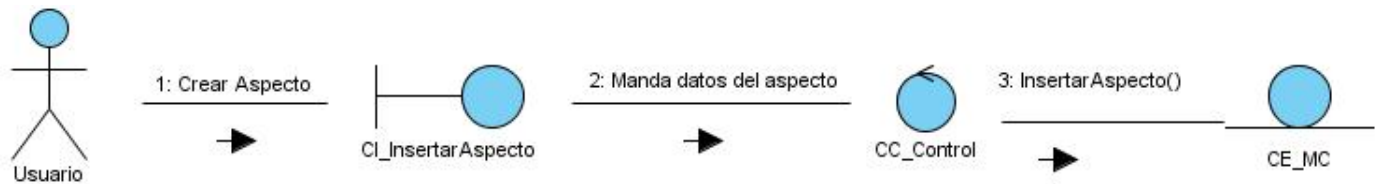


Figura 3.5: Diagrama de Colaboración. Insertar Aspecto.

El diagrama muestra cómo el sistema crea un aspecto dándole respuesta a la acción realizada por el usuario.



Figura 3.6: Diagrama de Colaboración. Modificar Aspecto.

En este diagrama se ilustra cómo el sistema modifica el aspecto después de la acción realizada por el usuario.

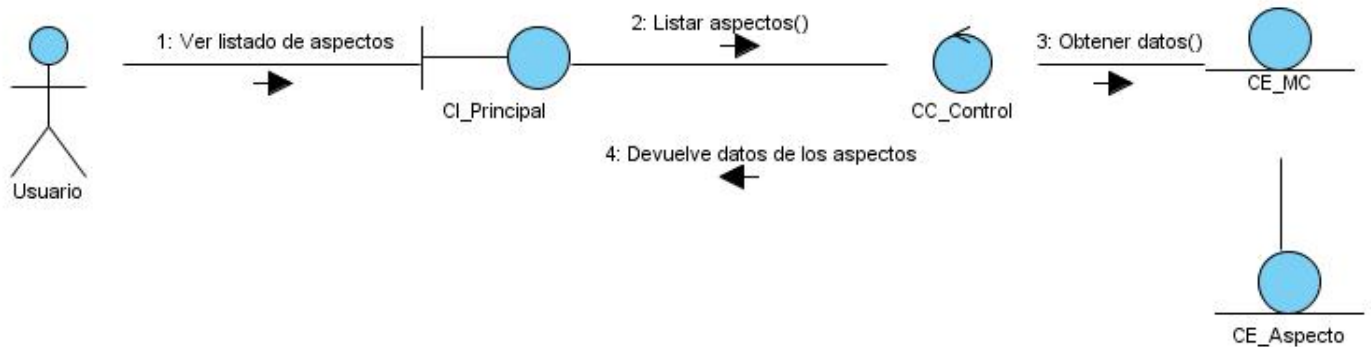


Figura 3.7: Diagrama de Colaboración. Listar Aspecto.

Se muestra el flujo de información cuando el usuario decide listar los aspectos que ha creado y cómo el sistema obtiene y devuelve los datos solicitados.



Figura 3.8: Diagrama de Colaboración. Eliminar Aspecto.

En el anterior diagrama se representa cómo el sistema elimina un aspecto después que el usuario realiza la acción pertinente.



Figura 3.9: Diagrama de Colaboración. Insertar Asociación.

Se ilustra cómo el usuario crea una asociación y la respuesta del sistema.



Figura 3.10: Diagrama de Colaboración. Modificar Asociación.

Se ilustra cómo el usuario modifica una asociación y la respuesta del sistema.

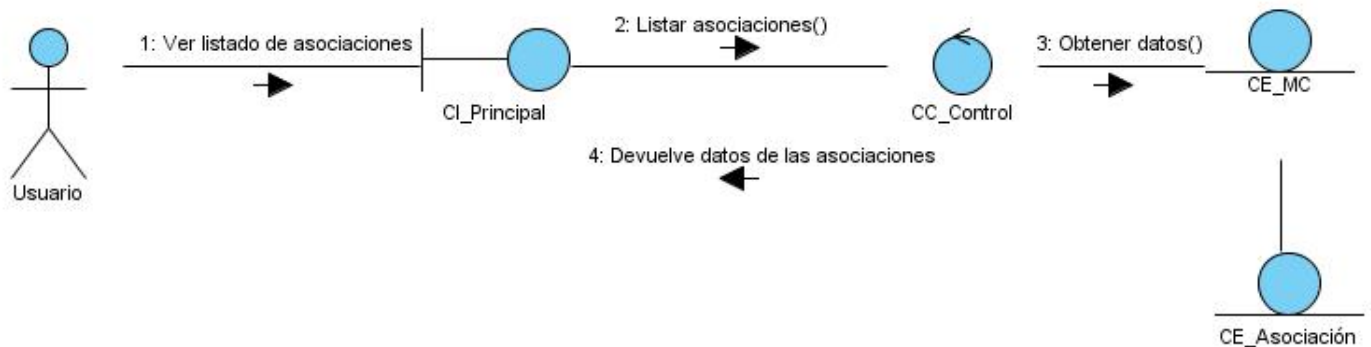


Figura 3.11: Diagrama de Colaboración. Listar Asociación.

Se ilustra cómo el usuario a través de la interfaz principal puede ver el listado de las asociaciones creadas por él.



Figura 3.12: Diagrama de Colaboración. Eliminar Asociación.

Se ilustra cómo el usuario elimina una asociación y la respuesta del sistema.



Figura 3.13: Diagrama de Colaboración. Insertar Tipo de Contenido.

Se ilustra cómo el usuario crea un tipo de contenido y la respuesta del sistema.



Figura 3.14: Diagrama de Colaboración. Modificar Tipo de Contenido.

Se ilustra cómo el usuario modifica un tipo de contenido y la respuesta del sistema.

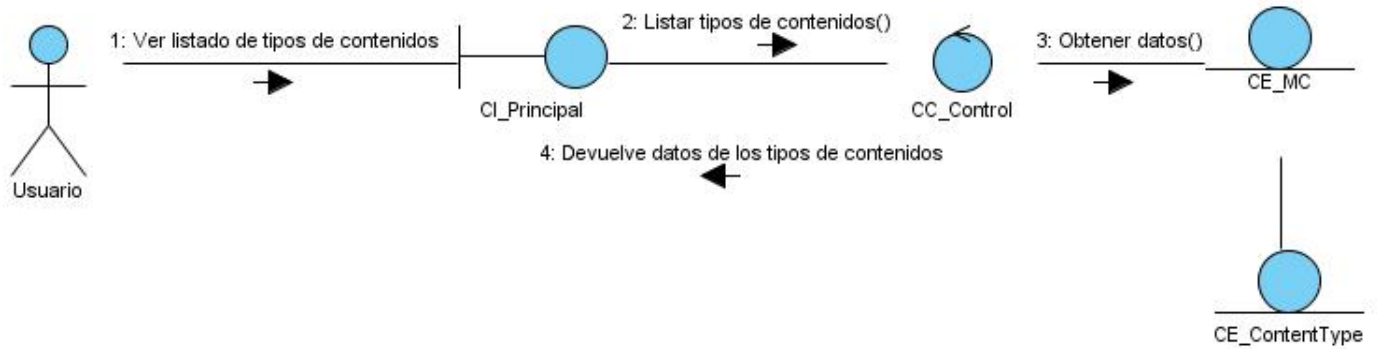


Figura 3.15: Diagrama de Colaboración. Listar Tipo de Contenido.

Se ilustra cómo el usuario a través de la interfaz principal puede ver el listado de los tipos de contenido creados por él.

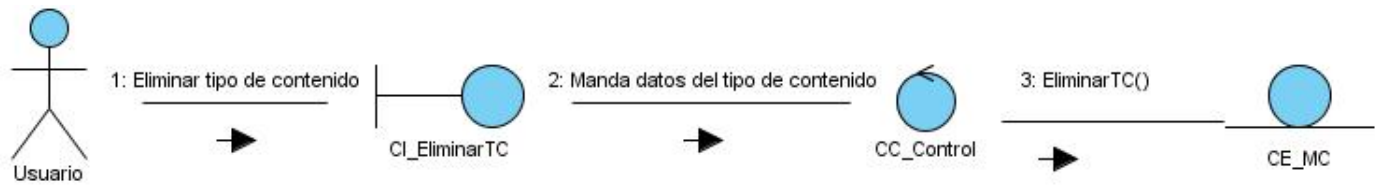


Figura 3.16: Diagrama de Colaboración. Eliminar Tipo de Contenido.

Se ilustra cómo el usuario elimina un tipo de contenido y la respuesta del sistema.



Figura 3.17: Diagrama de Colaboración. Insertar Restricción.

Se ilustra cómo el usuario crea una restricción y la respuesta del sistema.



Figura 3.18: Diagrama de Colaboración. Modificar Restricción.

Se ilustra cómo el usuario modifica una restricción y la respuesta del sistema.

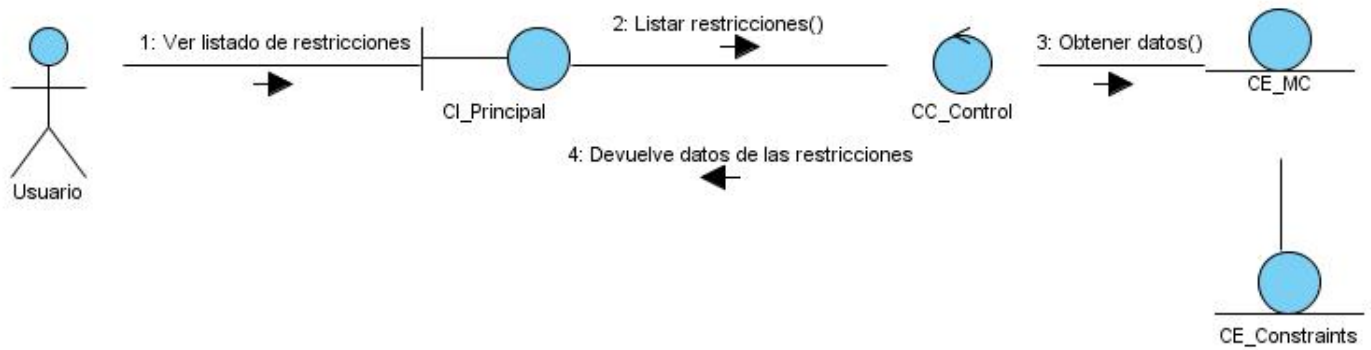


Figura 3.19: Diagrama de Colaboración. Listar Restricción.

Se ilustra cómo el usuario a través de la interfaz principal puede ver el listado de las restricciones creadas por él.



Figura 3.20: Diagrama de Colaboración. Eliminar Restricción.

Se ilustra cómo el usuario elimina una restricción y la respuesta del sistema.

3.5. Aspectos a tener en cuenta

Tratamiento de errores

Los errores críticos del sistema han sido tratados para que sean mejor entendidos por los usuarios.

Seguridad

Los datos manejados en esta aplicación son transparentes al usuario y el código fuente está disponible para futuras consultas.

Interfaz

La interfaz se diseñó teniendo en cuenta las necesidades del cliente.

Concepción de la ayuda.

La ayuda que se brinda en la aplicación realizada es brindada de varias formas y estilos:

- Textos explicativos.
- Mensajes aclaratorios.
- Mensajes de error.

3.6. Conclusiones

En este capítulo se han cumplido con los requisitos necesarios para comenzar con la implementación del sistema. Se expuso una idea de cómo el sistema debe funcionar para dar solución a los requisitos funcionales y no funcionales propuestos. Se ha logrado definir la arquitectura del sistema lo que permitirá establecer pautas para la implementación del software.

Capítulo 4

Implementación y prueba.

El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. En este flujo de trabajo se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. Además, se deben hacer las pruebas de unidad: cada implementador es responsable de probar las unidades que produzca. El resultado final de este flujo de trabajo es un sistema ejecutable.[19]

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.[20]

4.1. Diagrama de despliegue

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. En el contexto del diagrama de despliegue, una asociación representa una ruta de comunicación entre los nodos.

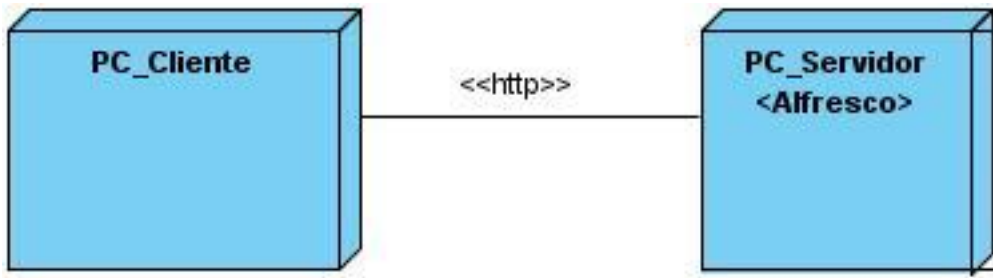


Figura 4.1: Diagrama de Despliegue.

PC_Cliente: Denota la computadora en la cual el usuario, con la ayuda del XMLEditor crea y/o exporta los Modelos de Contenidos.

PC_Servidor: Se refiere a la computadora en donde se aloja el gestor de contenidos empresariales Alfresco.

http: Protocolo usado por el XMLEditor para acceder al ECM Alfresco.

Como se muestra en la figura anterior el diagrama de despliegue solo cuenta con dos nodos que representan a la PC del cliente y a la PC del servidor los cuales están asociados.

4.2. Diagrama de componentes

Los diagramas de componentes ilustran las piezas del software y controladores embebidos que conformarán un sistema. Tiene un nivel más alto de abstracción que un diagrama de clases, usualmente un componente se implementa por una o más clases. Los componentes se representan como un clasificador rectangular con la clave «componente», opcionalmente el se puede mostrar como un rectángulo con un icono representativo en la esquina derecha arriba.

A continuación se exponen los Diagramas de Componentes que modelan el software que se está proponiendo:

Seguidamente se muestra el diagrama de componentes asociado a los subsistemas “model” y “controller” respectivamente.

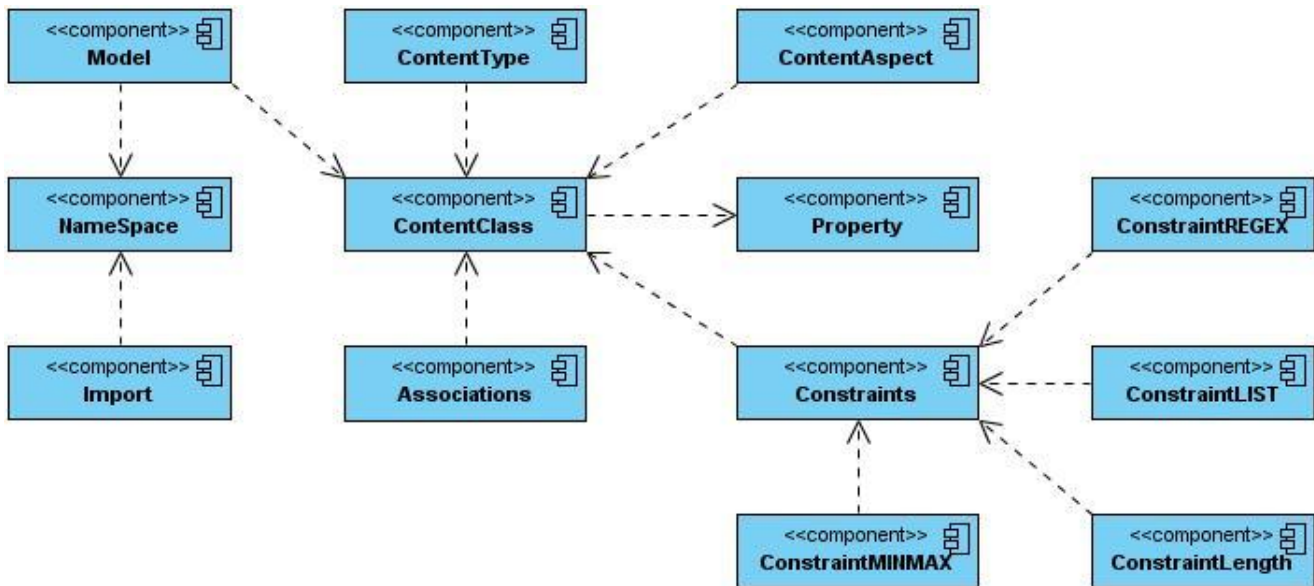


Figura 4.2: Subsistema model.

4.3. Pruebas de Software

Los objetivos del flujo trabajo de prueba:

- Encontrar y documentar los defectos que puedan afectar la calidad del software.
- Validar que el software funcione como fue diseñado.
- Validar y probar los requisitos que debe cumplir el software.
- Validar que los requisitos fueron implementados correctamente.

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software para hallar errores tales como: funciones incorrectas o ausentes; errores de interfaz; errores de rendimiento y errores de inicialización o terminación.[21] Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado

correcto, así como que la integridad de la información externa se mantiene. En la prueba de la caja blanca del software se comprueba los caminos lógicos del software proponiendo casos de prueba que se ejerciten conjuntos específicos de condiciones y/o ciclos. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado.

Pruebas de la caja negra:

- Verifican las especificaciones funcionales y no consideran la estructura interna del programa.
- Es hecha sin el conocimiento interno del producto.
- No validan funciones ocultas (por ejemplo funciones implementadas pero no descritas en las especificaciones funcionales del diseño) por tanto los errores asociados a ellas no serán encontrados.

Para confeccionar los casos de prueba de Caja Negra existen distintos criterios. Algunos de ellos son:

Particiones de equivalencia: La partición equivalente es un método que divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de error (por ejemplo procesamiento incorrecto de todos los datos de carácter) que de otro modo requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente 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.[21]

4.4. Modelo de prueba

Caso de Prueba “Agregar Aspecto“

Entrada	Resultado	Condiciones
No está seleccionado el tipo a eliminar.	El sistema muestra un mensaje aclarando que no se ha seleccionado el tipo a borrar.	No aplica.
Continúa en la próxima página		

Está seleccionado el tipo a eliminar.	El sistema elimina el aspecto seleccionado.	No aplica.
---------------------------------------	---	------------

Caso de Prueba “Modificar Aspecto”

Entrada	Resultado	Condiciones
Datos de: nombre y título	El sistema muestra un mensaje indicando que los datos son incorrectos.	Dato de nombre o título erróneos.
Datos de: nombre	El sistema muestra un mensaje indicando que ya existe un contenido con ese nombre.	El nombre coincide con otro contenido ya existente.
Datos de: nombre y título	El sistema modifica el aspecto seleccionado.	Dato de nombre o título correctos.

Caso de Prueba “Eliminar Aspecto”

Entrada	Resultado	Condiciones
No está seleccionado el aspecto a eliminar.	El sistema muestra un mensaje aclarando que no se ha seleccionado el aspecto a borrar.	No aplica.
Está seleccionado el aspecto a eliminar.	El sistema elimina el aspecto seleccionado.	No aplica.

Caso de Prueba “Agregar Tipo”

Entrada	Resultado	Condiciones
Adicionar un contenido con algún campo obligatorio vacío.	El sistema muestra en el campo erróneo un mensaje aclarando que el mismo está incorrecto.	Los campos no pueden estar vacíos.
Adicionar un contenido con todos los campos correctos.	El sistema crea el contenido con los datos especificados y se lo adjunta al modelo de contenido seleccionado.	No aplica.

Caso de Prueba “Modificar Tipo”

Entrada	Resultado	Condiciones
Datos de: nombre y título.	El sistema muestra un mensaje indicando que los datos son incorrectos.	Dato de nombre o título erróneos.
Datos de: nombre.	El sistema muestra un mensaje indicando que ya existe un contenido con ese nombre.	El nombre coincide con otro tipo ya existente.
Datos de: nombre y título.	El sistema modifica el aspecto seleccionado.	Dato de nombre o título correctos.
		Continúa en la próxima página

Datos de: nombre y título.	El sistema muestra un mensaje indicando que debe seleccionar un tipo para modificar.	No se ha seleccionado el tipo a modificar.
----------------------------	--	--

Caso de Prueba “Exportar a Alfresco”

Entrada	Resultado	Condiciones
No existen proyectos para exportar.	El sistema muestra un mensaje informando que no existen proyectos para exportar.	No existen proyectos guardados.
No selecciona los proyectos a exportar.	El sistema muestra un mensaje informando que no seleccionó los proyectos para exportar.	No haber seleccionado algún proyecto a exportar.
Datos de: Usuario, Contraseña, URL y Puerto.	El sistema muestra un mensaje informando que falló la conexión a Alfresco.	Dato de usuario, password, url o puerto erróneos.

4.5. Conclusiones del Capítulo

De modo general se realizaron los principales diagramas y los modelos de pruebas correspondientes a los flujos de trabajo implementación y prueba que alcanzan mayor desarrollo durante la fase de construcción del proyecto. La información anteriormente expuesta es la necesaria para continuar el ciclo que la metodología RUP propone.

Conclusiones

Al término de la investigación se pueden arribar a las siguientes conclusiones:

1. Se realizó un estudio sobre la importancia de los Modelos de Contenidos a nivel internacional, lo que permitió obtener el conocimiento para la implementación de la solución propuesta. Se fundamentaron las tecnologías usadas para el desarrollo de la aplicación sentando las bases en las ventajas que ofrecían al desarrollo.
2. Se realizó la identificación de los requerimientos del módulo a desarrollar, los casos de uso y los actores que interactúan con el sistema, quedando plasmados en las descripciones de los casos de uso, permitiendo así el análisis profundo de las funcionalidades que los requisitos deben cumplimentar, así como también se definieron los requisitos no funcionales que definen comportamientos específicos del sistema.
3. Se logró realizar el análisis y el diseño del módulo de gestión de tipos documentales, permitiendo un mejor entendimiento para la implementación del mismo.
4. Se describieron los diagramas de componentes y de despliegue, se logró la implementación de las funcionalidades así como se probaron las mismas a través del método de caja negra aplicando la técnica de particiones equivalentes, obteniendo resultados satisfactorios.
5. Se obtuvo el módulo de gestión de tipos documentales desarrollado con herramientas libres además de los artefactos generados por la metodología usada que sirven como documentación de la solución, facilitando la realización de posteriores cambios.

Glosario de términos

SGDA Sistema Gestor de Documentos Administrativos.

XMLEditor Software para la creación de xml de modelos de contenidos.

IBM International Business Machines, empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática.

Alfresco Sistema de gestión de contenido empresarial.

Framework Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

metamodelo Un modelo que define otros modelos.

IDE Entorno de Desarrollo Integrado.

CASE Ingeniería de Software Asistida por Computadora.

CRUD Acrónimo de Crear, Obtener, Actualizar y Borrar.

Software libre Se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

UCI Universidad de las Ciencias Informáticas.

XML Lenguaje extensible de maquetado.

DTD Definición de tipo de documento.

XSL Lenguaje extensible de hojas de estilos.

CSS Lenguaje de hojas de estilos.

HTML Estas siglas se corresponden con la definición "Lenguaje para marcado de hipertexto". Más claro aún, se trata de un lenguaje para estructurar documentos. Este lenguaje se basa en tags (instrucciones que le dicen al texto como deben mostrarse) y atributos (parámetros que dan valor al tag).

HTTP Es el protocolo de transferencia de hipertexto, o sea, el protocolo que los servidores de World Wide Web utilizan para mandar documentos HTML a través de Internet.

URL Es el Localizador Uniforme de Recursos, o dicho mas claramente, es la dirección que localiza una información dentro de Internet.

CSS Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML.

C, C++, Ruby, Python Lenguajes de programación.

PHP Preprocesador de hipertexto.

metadatos Son conjuntos de datos altamente estructurados que describen información, el contenido, la calidad, condición y otras características de los datos.

nodos Término usado en Alfresco para referirse a la forma en que están organizados los datos que contiene.

tipos de contenidos Término usado en Alfresco para representar un objeto de un modelo de negocio.

tipos de aspectos Término usado en Alfresco para representar un objeto de un modelo de negocio y permite crear reglas.

asociaciones Término usado en Alfresco para representar relaciones entre dos tipos de contenidos.

propiedades Término usado en Alfresco para representar uno o varios atributos pertenecientes a los tipos de aspectos o contenidos.

restricciones Término usado en Alfresco para representar reglas que se pueden aplicar a las propiedades.

modelo de contenido Describe los datos que serán almacenados en el repositorio de Alfresco.

diccionario de datos Forma parte del repositorio de datos de Alfresco, se encarga de describir las propiedades, asociaciones y restricciones de los contenidos.

Referencias bibliográficas

- [1] M. M. Mayra, *Gestión documental y organización de archivos*. Editorial Félix Varela, 2005.
- [2] “Alfresco ofrece una plataforma ECM de próxima generación para composite content applications. [en línea].” <http://winred.com/management/alfresco-ofrece-una-plataforma-ecm-de-proxima-generacion-para-composite-content-applications/gmx-niv116-con19634.htm>. [Consultado: Mayo 2011].
- [3] R. E. Chacon, “Módulo para la creación de modelos de contenidos para alfresco,” 2009.
- [4] “Editores XML: para crear o modificar documentos cómodamente · weblog · canal tecnología · terra. [en línea].” <http://www.terra.es/tecnologia/articulo/html/tec11123.htm>. [Consultado: Mayo 2011].
- [5] “ARTICULOS. [en línea].” http://www.adabi-ac.org/investigacion_arch_civil/memorias/paginas/articulo_707.htm. [Consultado: Mayo 2011].
- [6] “Alfresco ECM. [en línea].” <http://openred.es/index.php/gestion-documental>. [Consultado: Mayo 2011].
- [7] “Gestión del contenido corporativo con alfresco | quieres tecnologías, la gestión documental como un servicio. [en línea].” <http://www.queres.es/cms/soluciones/gestiondocumental/alfresco>. [Consultado: Mayo 2011].
- [8] “Diccionario de datos. [en línea].” <https://www.slideboom.com/presentations/181014/Diccionario-de-Datos>. [Consultado: Mayo 2011].
- [9] “Qué es un diccionario de datos - nocturnar.com. [en línea].” <http://www.nocturnar.com/forum/programacion/188323-que-diccionario-de-datos.html>. [Consultado: Mayo 2011].

- [10] “Descargar editor xml. [en línea].” <http://www.softonic.com/s/editor-xml>. [Consultado: Mayo 2011].
- [11] M. E. Roger O., Leslee P., “Applying requirements management with use case. [en línea],” [Consultado: Mayo 2011].
- [12] “Mastering UML with rational rose 2002. [en línea].” [ftp://ucistore.uci.cu/documentacion/Ingenieria Software/UML/Mastering UML with Rational Rose 2002.pdf](ftp://ucistore.uci.cu/documentacion/Ingenieria%20Software/UML/Mastering%20UML%20with%20Rational%20Rose%202002.pdf). [Consultado: Mayo 2011].
- [13] “Rational - UML for data modeling. [en línea].” [ftp://ucistore.uci.cu/documentacion/Ingenieria Software/UML/Rational - UML for data modeling.pdf](ftp://ucistore.uci.cu/documentacion/Ingenieria%20Software/UML/Rational%20-%20UML%20for%20data%20modeling.pdf). [Consultado: Mayo 2011].
- [14] “UML- guia visual. [en línea].” [ftp://ucistore.uci.cu/documentacion/Ingenieria Software/UML/UML- Guia Visual.pdf](ftp://ucistore.uci.cu/documentacion/Ingenieria%20Software/UML/UML-%20Guia%20Visual.pdf). [Consultado: Mayo 2011].
- [15] L. B. Wilson, *Comparative Programming Languages*. Addison-Wesley, second ed., 1993.
- [16] “Learn about java technology. [en línea].” <http://www.java.com/en/about/>. [Consultado: Mayo 2011].
- [17] “Ayuda para programación java. [en línea].” <http://ayuda-java.blogspot.com/2007/07/qu-es-netbeans.html>. [Consultado: Mayo 2011].
- [18] “Extensible markup language (XML). [en línea].” <http://www.w3.org/XML/>. [Consultado: Mayo 2011].
- [19] “Flujo de trabajo de implementación - EcuRed. [en línea].” [http://www.ecured.cu/index.php/Flujo_de _Trabajo_de_Implementaci%C3%B3n](http://www.ecured.cu/index.php/Flujo_de_Trabajo_de_Implementaci%C3%B3n). [Consultado: Mayo 2011].
- [20] “Prueba de RUP - EcuRed. [en línea].” [http://www.ecured.cu/index.php?title=Importancia_y_objetivos_del _flujo_de _trabajo_Prueba_de_RUP&oldid=605585](http://www.ecured.cu/index.php?title=Importancia_y_objetivos_del_flujo_de_trabajo_Prueba_de_RUP&oldid=605585). [Consultado: Mayo 2011].
- [21] P. Roger S, *Ingeniería del software*. McGraw-Hill, sixth ed., 2005.

Bibliografía

1. Potts, Jeff. Alfresco Developer Guide. 2009.
2. Shariff, Munwar. Alfresco Enterprise Content Management Implementation. 2006.
3. Pressman, Roger S. Ingeniería del software. 6th Edition. 2005.
4. Mena Mujica, Mayra. Gestión documental y organización de archivos. 2005.
5. Akif, Móhammad; Brodhead, Steven; Cioraianu, Andei; Jung, Eric; & Wdtz, Dave. Java y XML. 2002.
6. Lafore, Robert. Data Structures & Algorithms in Java. 1998.
7. McLaughlin, Brett. Java & XML. 2nd Edition. 2001.
8. Kruchten, Philippe. The Rational Unified Process, An Introduction, Second Edition, Addison-Wesley, 2000.
9. Jacobson, Ivar; Booch, Grady; Rumbaugh, James, "El Proceso Unificado de Desarrollo de Software".2000. Addison Wesley.
10. Larman, Craig. UML y Patrones, Second Edition, 2001
11. Extensible Markup Language. [Citado: Junio 2011]. Disponible en la dirección web:
<<http://www.w3.org/XML/>>
12. Learn about Java Technology. [Citado: Junio 2011]. Disponible en la dirección web:
<<http://www.java.com/en/about/>>

13. Applying Requirements Management with Use Case. [Citado: Junio 2011]. Disponible en la dirección web:
<<http://www.rational.com/media/whitepapers/aprmuc.pdf>>
14. UML- Guia Visual. [Citado: Junio 2011]. Disponible en la dirección web:
<<ftp://ucistore.uci.cu/documentacion/Ingenieria Software/UML/UML- Guia Visual.pdf>>
15. Extreme Programming: A Gentle Introduction. [Citado: Junio 2011]. Disponible en la dirección web:
<<http://www.extremeprogramming.org/>>
16. Manual de usuario de alfresco. [Citado: Junio 2011]. Disponible en la dirección web:
<<http://es.scribd.com/doc/54687770/Manual-de-usuario-de-alfresco>>
17. Sobre Alfresco - La alternativa para la Gestión de Contenidos Empresariales de Código Libre. [Citado: Junio 2011]. Disponible en la dirección web:
<<http://www.alfresco.com/es/about/>>