Universidad de las Ciencias Informáticas

Facultad 1

Diseño del módulo Gestión del control de acceso sobre las tipologías documentales en el Gestor de Documentos Administrativos eXcriba

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

Autor: Liset Laurencio Caballero

Tutores: Ing. Dayelis Blanco Hernández

Ing. Michel David Suárez

La Habana, Junio 2012

Declaración de autoría

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

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

_____

Firma del autor

Liset Laurencio Caballero

_____

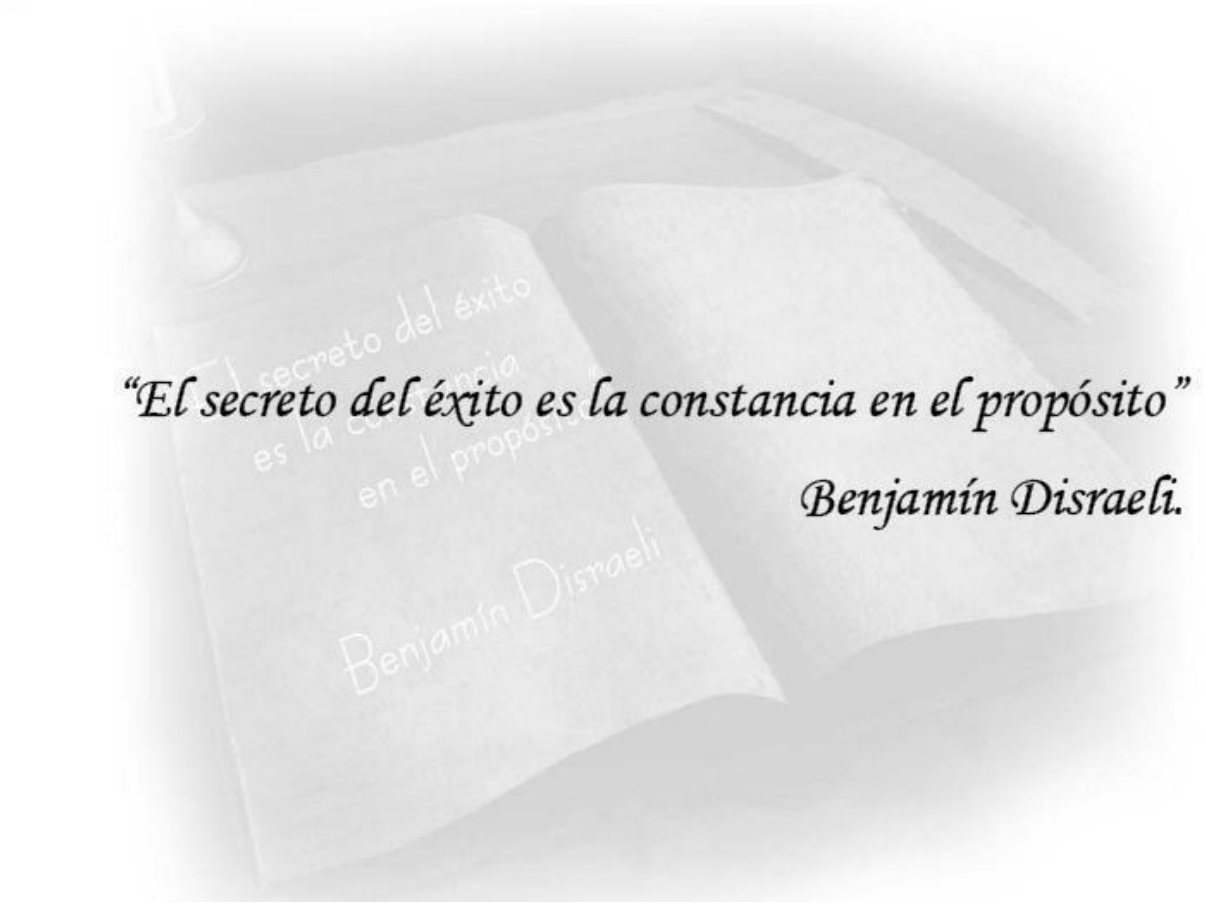
Firma del tutor

Ing. Dayelis Blanco Hernández

_____

Firma del tutor

Ing. Michel David Suárez



"El secreto del éxito es la constancia en el propósito"

Benjamín Disraeli.

Agradecimientos

Un especial agradecimiento a mis padres, por ser un ejemplo de sacrificio, confianza, comunicación y sobre todas las cosas de mucho amor.

Gracias a mi mami por estar siempre presente en cada momento importante de mi vida, por aconsejarme sin imponerme, por constituir una guía para la realización de mis sueños, por enseñarme a ser una mujer de bien y por estar orgullosa de mi ante todas las cosas.

A mi papi por confiar siempre en mí, por recordarme todos los días que nada cae del cielo, por enseñarme el sacrificio de las cosas que se quieren en la vida y por ser ese hombre inteligente, visionario y mi inspiración a seguir.

A mi hermanito por quererme y admirarme tanto, por estar siempre esperándome con ese beso y abrazo tan fuerte.

Un agradecimiento enorme a mi novio, por ser la persona que hizo la mayoría de mis sueños realidad, por demostrarme que el amor es mucho más que las bellas escenas con que lloro en la películas de amor, por ser mucho más que el hombre con quien siempre soñé, por levantarme todos los días con un beso, por apoyarme en cada uno de los momentos difíciles que he pasado, sencillamente por amarme del tamaño de las puntas de los dedos pegaditos.

A mis abuelitas Pilar y Librada, por siempre darme esos sabios consejos que me ayudan a salir adelante.

A toda mi familia materna y paterna por estar siempre apoyándome en cada una de mis decisiones y por entender todos esos momentos en los que no estuve presente.

Agradecerle a una personita que estuvo a mi lado los 5 años de universidad, que me vio llorar, reír, cantar, bailar, discutir, desaprobar, triunfar, engordar, rebajar a mi eterno amigo Reinier, gracias por ser ese apoyo incondicional en las buenas y malas.

A mis amigos de los cinco años, Aliu, Frank y Albert por aquellos momentos únicos que vivimos los primeros años de esta universidad, por todos los conciertos de Buena Fe de los que salíamos roncos, por los espaguetis, arroz frito y panetelas que tantas veces rompieron mi dieta y a Mirley y Susy que aunque llegaron después igual tenemos momentos inolvidables, escribiendo estas líneas me recuerdo mucho de

aquellos días en los que Mirley y yo, íbamos a escribir un libro de psicología con todas nuestras conferencias sobre la situación que nos rodeaba.

A mis suegros que me quieren y malcrían como una hija, por preocuparse de cada paso que doy en la vida.

A mis tutores Dayelis y Michel, por brindarme su ayuda y siempre estar dispuestos en este duro año.

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

Liset Laurencio Caballero

Dedicatoria

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

A mi hermano, por constituir un ejemplo importante para él.

A mi novio Roberto, por amarme cada día, por su paciencia, confianza y entrega.

A mi abuelita Pilar, por ver como le brillan los ojos cuando habla de lo orgullosa que está de su nieta.

A mi abuelita Librada, por tener tanta fe en mí y ser un modelo de mujer a seguir.

A toda mi familia y en especial a mi tío Wilfredo que aunque ya no está entre nosotros, su ausencia cambió mi forma de ver la vida, para ti que yo que estarías muy orgulloso.

Liset Laurencio Caballero

Resumen

El vertiginoso aumento de la información al paso de los años, ha hecho necesario la adopción de nuevos procedimientos para su gestión. En respuesta a esta situación surgen los sistemas de gestión documental. Estos sistemas tienen como objetivo administrar el flujo de documentos en una organización, además de velar por la seguridad de los mismos. En una organización todas las personas no realizan las mismas funciones, así como tampoco gestionan los mismos tipos de documentos. Es por esto por lo que la seguridad en el sistema no solo debe estar enfocada al documento, sino también al tipo de este, contribuyendo a que los documentos sean gestionados por personal autorizado.

El Gestor de Documentos Administrativos (GDA) eXcriba es uno de los sistemas de gestión documental que define los tipos de documentos especificados en una organización, solo que no gestiona el control de acceso sobre los mismos, provocando que el sistema sea incapaz de denegar operaciones sobre la información sensible en el momento de su creación, permitiéndole a cualquier usuario, clasificar documentación con tipos a los cuales no debería tener permisos. Debido a la situación descrita, se define como objetivo de la presente investigación diseñar una solución que permita gestionar el control de acceso de los tipos documentales en el GDA eXcriba.

Culminada la investigación se logrará el diseño de una propuesta para gestionar el control de acceso sobre los tipos de documentos, siendo un aporte al desarrollo del módulo propuesto y así contribuir a una mayor seguridad de los documentos en el GDA eXcriba.

Palabras claves: control de acceso, tipología documental, tipo de documento.

Índice General

Introducción **1**

1. Fundamentación teórica **5**

1.1. Gestión Documental 5

1.1.1. Sistemas de gestión documental 6

1.1.2. Beneficios que aporta un sistema de gestión documental 6

1.2. Tipología documental 7

1.2.1. Caracteres de los documentos 7

1.2.2. Importancia de la tipología documental 8

1.3. Control de acceso 8

1.3.1. Autorización 9

1.3.2. Modelos de autorización 10

1.4. Gestión del control de acceso de las tipologías documentales en los sistemas de gestión documental 12

1.4.1. Knowledge Tree 12

1.4.2. Nuxeo 13

1.4.3. Alfresco 13

1.4.3.1. Modelo de contenidos en Alfresco 14

1.4.4. eXcriba 16

1.4.5. Resultados 17

1.5. Metodología de desarrollo de software . . . 17
1.6. Lenguajes . . . 18
1.6.1. Lenguajes de programación . . . 18
1.6.2. Lenguaje de modelado . . . 19
1.7. Herramientas . . . 20
1.8. Tecnologías . . . 20
1.8.1. Framework . . . 21

2. Propuesta del sistema. 26

2.1. Problema y situación problémica . . . 26
2.2. Propuesta de solución . . . 27
2.3. Modelo de Dominio . . . 27
2.4. Especificación de los requisitos de software . . . 29
2.4.1. Técnicas para la captura de requisitos . . . 29
2.4.2. Requerimientos funcionales . . . 30
2.4.3. Requerimientos no funcionales . . . 31
2.5. Definición de los casos de uso del sistema . . . 32
2.5.1. Definición de los actores . . . 32
2.5.2. Diagrama de casos de uso del sistema . . . 33
2.5.3. Descripción de casos de uso del sistema . . . 33

3. Diseño del sistema. 37

3.1. Modelo de diseño . . . 37
3.1.1. Descripción de la arquitectura. . . . 37
3.1.2. Patrones de diseño . . . 39
3.1.3. Diagramas de clases del diseño. . . . 42
3.1.3.1. Descripción de las clases. . . . 43

3.1.3.2. Descripción de los servicios. 44
3.1.4. Diagramas de interacción del diseño. 47

4. Validación de la propuesta. 49
4.1. Métrica para evaluar los requisitos. 49
4.1.1. Métrica de la calidad de la especificación. 49
4.1.2. Técnica de Construcción de prototipos 53

Conclusiones 54

Recomendaciones 55

Referencias bibliográficas 56

Bibliografía 59

A. Primer apéndice 62
A.1. Descripción de los casos de uso 62

B. Segundo apéndice 88
B.1. Diagramas de clases del diseño 88

C. Tercer apéndice 93
C.1. Descripción de las clases 93

D. Cuarto apéndice 97
D.1. Diagramas de secuencia 97

Introducción

A lo largo de la historia, el proceso de búsqueda, selección, organización y conservación de la información, ha sido de vital importancia en la vida del ser humano. Para el hombre primitivo, la información adquirida a partir de los fenómenos ocurridos a su alrededor, constituía un elemento imprescindible para su supervivencia. Por tanto, sintió la necesidad de que los conocimientos obtenidos por la experiencia, fuesen conservados y transmitidos a las futuras generaciones. Es por ello que, surgieron diferentes métodos, tales como: el lenguaje articulado, el arte rupestre, los jeroglíficos, la escritura cuneiforme, el uso de la tinta y el papiro.

En la actualidad, potenciado por el desarrollo de la informática, las telecomunicaciones y el uso de internet, los usuarios demandan sistemas para el tratamiento de la información, capaces de facilitar el acceso y obtención de esta, de forma rápida, ágil y eficaz. De modo, que las exigencias sociales, han influido en el incremento del volumen de información, así como, en los procedimientos y técnicas para su tratamiento; lo cual ha propiciado pasar de forma progresiva de un concepto bastante simple en su origen a esquemas más complejos, fundamentados por los avances tecnológicos.

Como resultado de la situación descrita surgen los sistemas de gestión documental siendo su función fundamental administrar el flujo de documentos en una organización, además de permitir la recuperación de información, son de gran utilidad para eliminar los documentos inservibles así como asegurar la conservación indefinida de los más valiosos. Cabe destacar que estos sistemas pueden contribuir a una mejora u optimización de los procesos internos de la organización, posibilitando un mejor resultado en los servicios que esta brinda.

Es posible que cuando se menciona sistema de gestión documental surja la interrogante: ¿Qué tan segura está la documentación? Y es que una de las características más importantes de dichos sistemas radica en el control de acceso a los documentos, es decir, la seguridad de la documentación. El control de acceso a la documentación es un tema preocupante para cualquier organización que desea automatizar su gestión documental, ya que se enfrenta a la duda de cómo es manejado en el sistema los permisos para acceder a los documentos durante su gestión.

Actualmente, en la Universidad de las Ciencias Informáticas (UCI), se encuentra el departamento de Gestión Documental y Archivística. En este departamento se desarrolla el Gestor de Documentos Administrativos (GDA) eXcriba, el cual tiene como razón de ser: La gestión electrónica de los documentos administrativos. Al sistema eXcriba se le han incorporado numerosas mejoras que contribuyen a convertirlo en un Gestor de Documentos Electrónicos más completo. Aún así, el control de acceso sobre los documentos como parte de su seguridad, es un aspecto bastante débil, ya que no cuenta con una configuración capaz de restringir las operaciones básicas que puede realizar un usuario como: leer, escribir, editar contenido, en el modelo de datos[1], propiciando que el sistema sea incapaz de denegar acciones sobre la información sensible en el momento de su creación, permitiéndole a cualquier usuario clasificar documentación con tipos a los cuales ni si quiera podría tener conocimiento de que existen. Lo antes descrito implica que en ocasiones un documento no sea válido, producto de que fue creado por un usuario no autorizado para hacerlo, de ahí que se haga necesario emplear algún mecanismo para validar la autenticidad de la documentación creada, incurriendo en gastos innecesarios, además de retrasar el proceso de gestión de los documentos.

Luego de un análisis de la problemática anterior y con el fin de solucionar estas necesidades se propone el siguiente **problema a resolver**: ¿Cómo controlar el acceso a las tipologías documentales en el Gestor de de Documentos Administrativos eXcriba?

Para enmarcar los límites de la investigación se define como **objeto de estudio**: La gestión electrónica de documentos y como **campo de acción** el control de acceso a las tipologías documentales.

[1] Aquella configuración necesaria que tiene cada institución y que es incorporada al sistema en aras de catalogar los tipos documentales lo más conveniente posible. Ejemplos: Libro, Carta, Acta, Memorándum.

El **objetivo general** de la investigación es: Diseñar un módulo para el Gestor de Documentos Administrativos eXcriba que permita gestionar el control de acceso sobre las operaciones del modelo de datos del Administrador de Contenidos Empresariales (ECM) Alfresco.

El objetivo general planteado se ha desagregado en los siguientes **objetivos específicos**:

- Fundamentar las herramientas, tecnologías, lenguajes, metodología y el estudio de sistemas homólogos para el desarrollo de la solución propuesta.
- Identificar las funcionalidades que permiten la gestión del control de acceso sobre las tipologías documentales en el Gestor de Documentos Administrativos eXcriba.
- Diseñar una solución para gestionar el control de acceso sobre las tipologías documentales en el Gestor de Documentos Administrativos eXcriba.
- Validar la propuesta de diseño para el control de acceso sobre las tipologías documentales en el Gestor de Documentos Administrativos eXcriba.

Para la orientación adecuada en pos de alcanzar los resultados esperados se seleccionaron los siguientes métodos científicos de la investigación.

Métodos teóricos

Histórico-Lógico: Para la adquisición de conocimientos sobre la evolución, desarrollo y funcionamiento de las tipologías documentales, así como del control de acceso a los documentos.

Analítico-Sintético: Para realizar un estudio detallado tanto de las operaciones del modelo de datos de Alfresco como de la gestión del control de acceso a los documentos y a partir de la relación existente entre ambos, sintetizarlos en la solución propuesta.

Modelado: Este método es empleado para estructurar los procesos inherentes al control de acceso sobre las tipologías documentales, posibilitando una mejor comprensión de los mismos.

Métodos empíricos

Observación: Para reconocer la situación problemática existente, permitiendo también la obtención de conocimientos acerca del control de acceso a los tipos documentales en los sistemas de gestión documental.

Se plantea como **justificación de la investigación** que: La realización del diseño de un módulo para gestionar el control de acceso sobre las tipologías documentales que se definen en el sistema mediante el modelo de datos del ECM Alfresco, brindará una mayor seguridad, autenticidad y confidencialidad a los documentos generados en el Gestor de Documentos Administrativos eXcriba, ya que no le permitirá al usuario gestionar documentos con tipos a los cuales no tiene permisos.

Este documento se encuentra estructurado en 4 capítulos organizados de la siguiente forma:

Capítulo I: Fundamentación teórica: Se abordan los diferentes elementos que brindan la base teórico-conceptual para el desarrollo de la solución propuesta. En él se reflejan varios conceptos asociados a la tipología documental y al control de acceso; así como la realización de un estudio a diversos sistemas de gestión documental, para conocer como estos tratan el tema de la gestión del control de acceso sobre los tipos de documentos. Además se hace un análisis y propuesta de las tecnologías, herramientas y metodología de desarrollo de software a utilizar para el diseño e implementación de la solución.

Capítulo II: Propuesta del sistema: Se documenta la solución propuesta desde el punto de vista conceptual, teniendo en cuenta los requisitos funcionales y no funcionales que debe presentar la misma, así como los casos de usos asociados a estos requisitos.

Capítulo III: Diseño del sistema: Se describe a través de los diagramas de clases e interacción del diseño la solución que se dará a la problemática planteada, además de quedar representada y descrita la arquitectura así como los diferentes patrones de diseño que se utilizarán.

Capítulo IV: Validación de la propuesta: Se realiza una validación de los requisitos aplicando algunas métricas en pos de verificar la calidad y efectividad de los mismos, comprobando que estos cumplen con las necesidades del usuario.

Capítulo 1

Fundamentación teórica

En el presente capítulo se abordan los diferentes elementos que brindan la base teórico-conceptual para el desarrollo de la solución propuesta. En él se reflejan varios conceptos asociados a la tipología documental y al control de acceso; así como la realización de un estudio a diversos sistemas de gestión documental, para conocer como estos tratan el tema de la gestión del control de acceso sobre los tipos de documentos. Además se hace un análisis y propuesta de las tecnologías, herramientas y metodología de desarrollo de *software* a utilizar para el diseño e implementación de la solución.

1.1. Gestión Documental

Toda organización con el paso del tiempo va acumulando un gran volumen de información que por su exuberancia y variedad, se convierte en un verdadero reto la administración, almacenamiento y transformación de la misma. Unido a esto, está el alto grado de competitividad al que se enfrentan las entidades actualmente, siendo necesario la adopción de alguna solución que posibilite la mejora de su excelencia operacional, con el fin de lograr reducciones significativas de costos, incrementos en la productividad, optimización en el uso de recursos y fundamentalmente, la disponibilidad de la información. Como una vía de solución adoptada por muchas organizaciones, surge la gestión documental.

La norma UNE-ISO 15489-1 define la gestión documental como el "área de la gestión responsable de un control eficaz y sistemático de la creación, la recepción, el mantenimiento, el uso y la disposición de documentos de archivo, incluidos los procesos para incorporar y mantener en forma de documentos la información y prueba de las actividades y operaciones de la organización" [1]. Dicho de otra manera, la gestión documental es la responsable de controlar el flujo de documentos de todo tipo en una entidad durante su ciclo de vida, desde su creación inicial hasta su eliminación o archivado.

La documentación que es generada y tratada dentro de una entidad tiene diversos niveles de importancia, debido a esto, hay que tener en cuenta la seguridad como un requisito fundamental. ¿Cómo evitar la pérdida de documentos?¿Quién puede crear, modificar, visualizar o eliminar algún documento?¿La información solo es legible para las personas autorizadas? Estas son algunas de las interrogantes que pueden surgir en el seno de cualquier entidad y muchas intentan encontrar respuestas con la aplicación de una adecuada gestión documental.

1.1.1. Sistemas de gestión documental

En el contexto actual, impulsado por el paradigma de las nuevas tecnologías, se aprecia un destacado aumento en la cantidad de documentos en formato digital. Cada día en las instituciones se producen, reciben, exportan y almacenan un mayor número de estos. Lo anterior ha conllevado a la creación de un espacio de trabajo que se encargue esencialmente del manejo de documentos, la transformación de documentos tradicionales al formato electrónico y la gestión de archivos automatizados. Con este propósito surge la industria de los sistemas para la gestión de documentos electrónicos.

Se puede definir como sistema de gestión documental un "sistema de información que incorpora, gestiona y facilita el acceso a los documentos de archivo a lo largo del tiempo" [1].

1.1.2. Beneficios que aporta un sistema de gestión documental

Con el alto grado de competitividad al que están sometidas las entidades de hoy en día, la implantación de un sistema de gestión documental es un arma de vital importancia para mejorar su funcionamiento y resultados, debido a las numerosas ventajas que esto supone, como son:

- Sencillez y accesibilidad a toda la documentación de la entidad.
- La información se encuentra más segura contra pérdidas y accesos por personal no autorizado.
- Ahorro en cuanto a espacio de almacenamiento, además de reducirse el tiempo empleado para hacer búsquedas y en almacenar la información.

- La documentación se encuentra accesible para todos, lo que evita duplicaciones y gastos en copias.
- Aumento de la productividad al tener un acceso más eficiente y rápido a la información.

1.2. Tipología documental

La doctora Vicenta Cortés Alonso definió el tipo documental como "número y disposición de los elementos de la información que corresponden a la actividad que lo ha producido"[2]. Otros autores hacen referencia a este término como la "forma específica o documental en la que se plasma o refleja una función, actividad o tarea de un sujeto productor. Ejemplos: carta, acta, informe, expediente"[3].

Existe una estrecha relación entre el tipo documental y la actividad que lo produce, un ejemplo de esto puede ser “informe” que proviene de la actividad de “informar”. Dada esta relación se puede concluir que los tipos documentales pueden ser infinitos, como lo son las actividades humanas y varían en el tiempo y el espacio.

La reconocida archivera Antonia Heredia ha expuesto que la tipología documental es “la suma de tipología diplomática y tipología jurídico-administrativa. La delimitación de los tipos, su fijación e identificación vendrán determinados por el análisis de los caracteres externos e internos de los documentos y de su mensaje o información ”[4].

1.2.1. Caracteres de los documentos

Los documentos de archivo tienen tanto una estructura física (caracteres externos) como un contenido sustantivo (caracteres internos).

Los caracteres externos son aquellos referidos a la clase (textuales, audiovisuales o electrónicos), el tipo (carta, informe, acta), el formato (papel, cinta, disquete), la cantidad (Número de páginas, folio, metros lineales) y la forma (original, copia).

Los caracteres internos hacen referencia a:

- Entidad productora: persona o institución que produce el documento.

- Origen funcional: es la función, actividad o tarea de una persona o institución que provoca el surgimiento de un documento.
- Fecha y lugar de producción: sitúa en tiempo y espacio el documento.
- Contenido: el asunto del cual trata el documento [3].

1.2.2. Importancia de la tipología documental

La tipología documental es un tema fundamental dentro de la gestión documental, debido a la importancia que adquiere a la hora de clasificar la documentación, ya que reconocer que tipo de documentos se tienen en las manos es primordial para posteriormente saber dónde y cómo archivarlo. Esta clasificación de forma organizada contribuirá a:

- Aceleración y sistematización para organizar los documentos.
- Eficacia de la recuperación de la información para la toma de decisiones.
- Protección de la información administrativa.
- Estabilidad y continuidad administrativa.
- Optimización de los recursos y racionalización de los espacios [5].

1.3. Control de acceso

La norma UNE-ISO 15489-1 define el acceso como derechos, modo y medios de localizar, usar o recuperar información. Teniendo en cuenta la definición anterior se puede concebir el control de acceso como el proceso de conceder o denegar permisos a usuarios para acceder a objetos, información o datos. Cuando se accede a un recurso existen tres actividades que están estrechamente relacionadas: la autenticación (quién soy), la autorización (que puedo hacer), el registro de auditoría (qué he hecho).

- Autenticación: Cuando un usuario requiere acceso a un sistema, debe presentar información personal que permita establecer de forma unívoca su identidad dentro del entorno. El sistema procesa estos datos y determina si ese usuario es realmente quien dice ser.
- Autorización: Este proceso determina lo que un usuario ya autenticado, tiene derecho o no a hacer.
- Auditoría: En muchos sistemas se quiere llevar un control detallado de las operaciones que realizan los usuarios en el mismo, es por esto por lo que existen mecanismos para registrar todas las acciones de los usuarios y posteriormente poder ser usado durante un proceso de auditoría.

En un sistema de gestión documental controlar el acceso a los documentos, estableciendo, por ejemplo, diferentes niveles de seguridad por usuario o por grupos de usuarios es necesario para proteger la confidencialidad, integridad y disponibilidad de la información.

1.3.1. Autorización

"La autorización está estrechamente ligada con la autenticación. Una vez que el usuario ha validado su identidad para acceder a algún recurso, es necesario restringir sus acciones de acuerdo con quién es y qué está tratando de hacer" [6]. De manera general la autorización determina las operaciones que el usuario está autorizado a realizar.

Existen diversos mecanismos para otorgar la autorización de un usuario, puede ser en función de comprobar que el usuario está autenticado correctamente o puede ser mediante atributos que identifican el rol o papel que juega el usuario dentro de la organización.

Una aplicación de gestión documental debe garantizar que los documentos tengan el nivel de seguridad adecuado. Existen documentos con información sensible que no deberían ser accedidos más que por el personal adecuado. Para garantizar estas condiciones es importante tener implantado un adecuado mecanismo de autorización.

1.3.2. Modelos de autorización

Existen fundamentalmente dos tipos básicos de controles de acceso con filosofías diferentes: Control de Acceso Discrecional (DAC) y Control de Acceso Obligatorio (MAC). Posteriormente se propone el modelo de Control de Accesos Basado en Roles (RBAC), como intento de unificar los modelos clásicos DAC y MAC.

Control de Acceso Discrecional (DAC)

"Procedimiento para restringir el acceso a los objetos de un sistema basado en la identidad de los sujetos. El control se denomina discrecional, pues un sujeto con ciertos derechos de acceso puede pasar éstos, quizás indirectamente y siempre que no lo impida un control de acceso obligatorio, a otro sujeto cualquiera. Se instrumenta para aplicar una política de seguridad basada en identidades" [7]. En el modelo DAC, un usuario (típicamente, el creador o "propietario" del recurso) decide cómo protegerlo disponiendo cómo compartirlo, mediante controles de acceso impuestos por el sistema. Lo esencial es que el propietario del recurso puede cederlo a un tercero.

Inicialmente estos sistemas eran bastantes simples, ya que permitían un conjunto limitado de operaciones sobre un recurso, por lo que pronto se le añadieron las Listas de Control de Acceso (ACLs). Estas listas son una estructura básica de autorización que permiten asignar permisos a usuarios o grupos concretos; por ejemplo, se pueden otorgar ciertos permisos a dos usuarios sobre un archivo sin necesidad de incluirlos en el mismo grupo. Estas estructuras contienen los identificadores de los usuarios junto con sus derechos de acceso a un recurso determinado, como leer, escribir, ejecutar. Entre más usuarios soliciten el acceso a un recurso más identificadores contendrá la ACL, lo que dificulta el manejo de estas listas y las hace una alternativa poco escalable.

Control de Acceso Obligatorio (MAC)

"Procedimiento para restringir el acceso a los objetos de un sistema. Está basado en la sensibilidad de la información contenida o tratada en éstos (expresada en una etiqueta de seguridad) y la autorización (denominada habilitación) de los sujetos que pretenden acceder. Se instrumenta para aplicar una política de seguridad basada en reglas. Modelo de seguridad en el que un responsable clasifica los objetos y sujetos

según sus respectivos niveles de seguridad y habilitación, además los compartimenta según el principio de mínimo privilegio" [7].

De manera general en este modelo, es el sistema quién protege los recursos. Todo recurso y usuario del sistema tiene una etiqueta de seguridad, la cual sigue el modelo de clasificación de la información militar, donde la confidencialidad de la información es lo más importante. En este tipo de modelo, todas las decisiones de seguridad las impone el sistema, comparando las etiquetas de acceso frente al recurso accedido.

Control de Acceso Basado en Roles (RBAC)

Es una tecnología para satisfacer las principales necesidades en cuanto a Control de Accesos se refiere. Cuando se tienen grandes sistemas en red la seguridad es un problema principal, debido a lo costoso y compleja que se vuelve su administración, es por ello por lo que surge RBAC, para tratar de minimizar estos problemas.

La administración de la seguridad, consiste en que los roles deben asignarse adecuadamente a los diferentes tipos de personas, según las capacidades y puestos de cada una de ellas. Es el propietario o administrador del sistema quien maneja los datos, para satisfacer las necesidades de la organización.

Para lograr la confiabilidad en un sistema se debe tener en cuenta los caminos para llegar a ella, los niveles que existen en los usuarios (ejemplo: usuario y súper usuario) y los permisos o privilegios que dichos usuarios puedan llegar a tener. En ocasiones los usuarios de una organización pueden cambiar de un puesto a otro, cambiando también sus permisos y responsabilidades, llegando a ser difícil y de alto coste. La situación anterior puede ser evitada mediante RBAC, ya que son los roles de usuario los que posibilitan el acceso al sistema y no la identificación del usuario. Cada rol realiza una función y puede tener asignados uno o más usuarios, además tienen asociado permisos que determinan los datos y aplicaciones a las que el rol tiene acceso.

1.4. Gestión del control de acceso de las tipologías documentales en los sistemas de gestión documental

1.4.1. Knowledge Tree

Es un sistema de gestión electrónica de documentos, cuenta con una versión privativa y una versión libre GNU/GPL [8]. Knowledge Tree posee un completo conjunto de funcionalidades y varios módulos que hacen de él un sistema útil para la gestión de documentos de forma simple y eficaz. Uno de los aspectos positivos con que cuenta el sistema está dado por las funciones de administración, que son accesibles para cualquier administrador, entre las que se encuentra la creación de tipos de documentos.

Los usuarios del sistema se organizan en usuarios, grupos, roles y unidades. Es importante señalar que Knowledge Tree asigna permisos a grupos y roles, pero no a usuarios individuales. De este modo cada usuario se agrega a un grupo y / o un rol. Los grupos contienen típicamente uno o más usuarios y cualquier usuario puede pertenecer a uno o más grupos.

Knowledge Tree maneja la seguridad permitiendo la configuración de los niveles de acceso a la información y documentación almacenada en el sistema, de acuerdo con el grupo o área de la compañía a la que pertenezca el usuario autorizándolo según su rol ya sea crear, modificar o eliminar los documentos. De forma general se gestiona el control de acceso a los documentos de acuerdo con roles y permisos, además de realizar auditoría y registro sobre el acceso a los documentos.

El sistema cuenta con una importante gestión sobre los tipos de documentos, permitiendo la visualización, adición y edición de estos. Los tipos de documentos no pueden ser eliminados, en caso de no ser utilizados se pueden desactivar. Cada tipo de documento puede ser asociado con uno o más campos y cada uno de estos a su vez se puede asociar con diversos tipos de documentos.

1.4.2. Nuxeo

Nuxeo es una empresa francesa que ofrece una solución para la gestión documental, Nuxeo Enterprise Platasform. Además cuenta con la herramienta de configuración gráfica Nuxeo Studio, que permite configurar un gran número de opciones, tales como los tipos de documentos, la definición de los ciclos de vida, algunos elementos gráficos de las interfaces o incluso la configuración básica de un proyecto de Nuxeo. Es una solución completa de gestión de contenido empresarial: metadatos, tipos de documentos, *workflow* avanzado, gestión de categorías o funciones de colaboración [9].

En el sistema, los administradores están a cargo de los derechos de acceso, mediante la gestión de usuarios, grupos y permisos. La gestión de derechos de acceso significa la concesión o denegación de acceso en un espacio. Los grupos de usuarios pueden estar compuestos por usuarios y subgrupos, las propiedades de estos se pueden crear y modificar directamente. Existen dos grupos predeterminados, los administradores y los miembros. Además son utilizados para gestionar los derechos de acceso con mayor facilidad.

Nuxeo por defecto posee una gran de variedad de tipos documentales disponibles. Este sistema utiliza el concepto de faceta para añadir nuevos tipos de documentos. Los nuevos tipos de documentos se crean a partir de esquemas XML (XSD) y se añaden a la arquitectura de la solución incorporando nuevos plug-ins (extensiones). El sistema brinda un pequeño nivel de seguridad sobre los tipos de documentos ya que permite restringir la gestión con estos tipos en un espacio dado. Es importante señalar que, esta configuración se puede realizar sin necesidad de cambiar los derechos de acceso aplicados al espacio.

1.4.3. Alfresco

Alfresco es una herramienta que constituye la principal alternativa de código abierto para la gestión de contenidos empresariales. La plataforma de contenidos de Alfresco utiliza una arquitectura flexible de estándares abiertos para proporcionar gestión documental, gestión de contenidos web, gestión de registros y *software* colaborativo. Alfresco está construido mediante los últimos componentes de infraestructuras de código abierto, que incluyen: Spring, Hibernate, Lucene y MyFaces. Se basa en Programación Orientada a Aspectos y no cobra las tradicionales cuotas de licencia [10].

Referente a la protección contra el acceso no autorizado al contenido, Alfresco impone la autorización mediante la asignación de un rol a un usuario o grupo específico de un espacio o contenido determinado. Los permisos definen los derechos de acceso en los espacios y contenidos. Los roles son las colecciones de permisos asignados a un usuario, el sistema define algunos roles como: Consumidor, Editor, Contribuidor, Colaborador y Coordinador. Es importante destacar que los roles y permisos en Alfresco pueden ser extendidos para soportar la necesidades de una institución.

1.4.3.1. Modelo de contenidos en Alfresco

Alfresco cuenta con un repositorio que provee soporte para el almacenamiento, administración y recuperación de contenido. Este soporta un sustancioso diccionario de datos para describir la estructura de un contenido, así como las propiedades, asociaciones y las restricciones de los mismos.

El Diccionario de Datos del repositorio de Alfresco tiene ya predefinido un conjunto de contenidos, dentro de los que se puede citar *"File"*, *"Folder"*, entre otros. Sin embargo, cada aplicación concebida para las organizaciones tiene sus propios requerimientos, por tal motivo se ha diseñado el diccionario de forma que sea extensible, permitiendo así crear nuevos tipos de contenido.

El modelo de contenido soporta dos términos fundamentales: los tipos de contenidos (*Content Types*) y los aspectos (*Aspects*). Estos facilitan la posibilidad de describir la estructura de un contenido en específico, incluyendo las propiedades (metadatos) de los mismos, así como las relaciones o asociaciones con otros tipos de contenidos.

Tipo de contenido

Los tipos de contenido pueden ser vistos como tipos o clases en el mundo de la Programación Orientada a Objetos (POO). Estos son usados para modelar objetos de negocio en una empresa. Tienen propiedades y pueden heredar características y comportamiento de un tipo de contenido padre. En Alfresco se definen algunos tipos como *"Content"*, *"Folder"* y *"Person"*. La definición de tipos de contenido solo está acotada por la imaginación de su creador o por los requerimientos del propio negocio.

En el sistema los tipos de contenido son únicos a través del repositorio por el uso de los *namespaces* (espacios de nombres). Usar los *namespaces* ayuda a prevenir colisiones de nombres a través del repositorio. Alfresco añade los tipos de contenidos mediante configuración XML.

Propiedades

Las propiedades son pedazos de metadatos asociados a un tipo de contenido. Estas pueden ser vistas como los atributos de las clases en la POO, o sea que representan las cualidades o características que poseen los tipos de contenidos.

Aspectos

Los aspectos permiten el enriquecimiento del modelo de contenido con propiedades y asociaciones que pueden adjuntarse a los tipos de contenidos en tiempo de ejecución. Un ejemplo en el que es conveniente el uso de aspectos es cuando se quiere mostrar un subconjunto del contenido del repositorio en el sitio web, en este caso el contenido que será mostrado en la web tendrá que tener una bandera señalizadora que indique cuándo y qué contenido deberá ser mostrado o no.

En resumen un aspecto puede ser visto como un conjunto de metadatos que se puede aplicar a cualquier tipo de contenido, así como condiciones que se pueden definir para asignar determinado comportamiento.

Asociaciones

Las asociaciones definen las relaciones entre los tipos de contenidos. Pueden ser vistas como las relaciones entre clases desde la perspectiva de la POO. Estas se dividen en dos tipos: *"Peer Associations"* y *"Child Associations"*. Las *"Peer Associations"* son definidas en Alfresco como *"Associations"* y definen una relación entre dos objetos, aunque ninguno de los objetos se puede considerar como subordinado del otro. Sin embargo las *"Child Associations"* definen un elemento subordinado a otro, donde la existencia del elemento subordinado depende de la del padre o sea el hijo no existirá una vez que el padre deje de hacerlo. Un ejemplo de *"Child Associations"* es: si una carpeta contiene en su interior determinados elementos, una vez que la carpeta sea eliminada el contenido dentro de la misma también será eliminado.

1.4.4. eXcriba

eXcriba es un Gestor de Documentos Administrativos que surge en el 2007 como una propuesta de la Universidad de las Ciencias Informáticas para la gestión de documentos y la automatización de los procesos documentales que se ejecutan en cualquier organización. Este sistema informático tiene como núcleo el ECM[1] Alfresco. Además está compuesto por una interfaz de usuario sencilla, cómoda y flexible que brinda numerosas funcionalidades entre las que se encuentran:

- Gestión de documentos
- Gestión de carpetas
- Automatización de los flujos documentales
- Control de acceso y permiso.

El sistema posee un subsistema de control de acceso que permite definir las acciones que un usuario puede realizar sobre un espacio de trabajo o documento. Para lograr la gestión del acceso se le asigna un rol a un usuario o a un grupo determinado sobre un espacio de trabajo o documento específico. eXcriba ya trae definido algunos roles como es Consumidor, Editor, Contribuidor, Colaborador y Coordinador.

- Consumidor: Puede leer el contenido
- Editor: Puede leer y editar el contenido
- Contribuidor: Puede leer y agregar contenido
- Colaborador: Puede leer, editar y agregar contenido
- Coordinador: Puede leer, editar, agregar y eliminar contenido.

El eXcriba cuenta con un módulo para la gestión de tipologías documentales. Este módulo es una aplicación de escritorio para la creación de tipologías documentales, con el objetivo de clasificar un documento dentro del repositorio documental.

[1] Administrador de Contenidos Empresariales.

1.4.5. Resultados

Luego de un análisis de algunos sistemas de gestión documental se concluye que casi ninguno gestiona el control de acceso sobre los tipos de documentos que definen, siendo esto una debilidad en la seguridad de dichos sistemas. Nuxeo incluye una pequeña seguridad a los tipos de documentos, ya que restringe la gestión de estos en un espacio dado; o sea que se puede configurar un espacio para que solo se puedan crear, modificar o visualizar documentos de un tipo específico. Esta solución no es la más óptima para dar respuesta a la problemática planteada en la investigación, ya que no incluye la restricción de acceso a usuarios no autorizados a crear, visualizar, modificar o eliminar documentos de un tipo determinado, además tampoco cuenta con la gestión de los aspectos asociados a estos tipos de documentos.

Lo anteriormente expuesto evidencia lo novedoso de la solución propuesta, ya que la mayoría de los sistemas estudiados, incluyendo al Alfresco como núcleo del eXcriba, no gestiona el control de acceso a las tipologías documentales.

1.5. Metodología de desarrollo de software

RUP (*Rational Unified Process*) con Nivel 2 de CMMI

El proceso unificado es un proceso de desarrollo de *software*. Un proceso de desarrollo de *software* es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de *software*. Sin embargo el proceso unificado es mucho más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de *software*, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos [11].

RUP es una metodología de desarrollo que se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Esta metodología define roles, que es el papel que desempeña una persona a lo largo del proceso de desarrollo; además incluye artefactos, que son los productos tangibles que se obtienen en el proceso como por ejemplo, el modelo de caso de uso, el código fuente, etc.

Por lo anteriormente expuesto y por ser la metodología usada en el proyecto eXcriba al cual pertenece la investigación, se define como guía para el desarrollo de la solución propuesta la metodología RUP. Actualmente la Universidad de las Ciencias Informáticas (UCI) como centro productivo está acometiendo un proyecto de mejora de sus procesos basado en el modelo CMMI (*Capability Madurity Model Integration*) con el objetivo de crecerse como organización productora de *software*. Por esta razón se complementará la metodología a utilizar con nivel 2 de CMMI con vistas a mejorar el ciclo de vida dentro del desarrollo de *software* y alcanzar una mayor calidad.

1.6. Lenguajes

1.6.1. Lenguajes de programación

PHP

PHP es un lenguaje de *scripting* de servidor que permite generar páginas web dinámicas. Las páginas PHP pueden contener texto, HTML y bloques de *scripts*. Cuando un explorador solicita una página PHP, se ejecuta el script PHP en el servidor web y el HTML resultante se muestra en el explorador [12]. El código PHP funciona en numerosos sistemas operativos como Linux, Microsoft Windows, Unix y otros. Además soporta la programación orientada a objetos y tiene capacidad de conexión con la mayoría de los sistemas gestores de base de datos.

No existe un lenguaje de programación que se ajuste a todas las necesidades, es por ello por lo que debe seleccionarse el que mejor satisfaga los requerimientos. Por las características que tiene PHP se propone su uso para el desarrollo del módulo propuesto, además de que la presente investigación es parte del GDA eXcriba, el cual define al lenguaje PHP para la implementación del sistema.

JavaScript

Se propone el uso de JavaScript tanto para la implementación de las interfaces de usuario como de los servicios.

Para el desarrollo de las interfaces de usuario se puede usar la biblioteca jQuery escrita en JavaScript, la cual provee un gran cúmulo de funcionalidades que facilitan la implementación de las interfaces del módulo propuesto.

En la implementación de los servicios es factible el uso de la API (*Application Programming Interface*) de JavaScript que provee Alfresco ya que esta le permite a los desarrolladores acceder, modificar o crear objetos del repositorio como usuarios, nodos, grupos, etiquetas o categorías.

Sin embargo, es importante destacar que no se debe confundir el habitual código JavaScript que se escribe para las páginas HTML, donde el código es ejecutado por el navegador (esto significa, en el lado del cliente) con los *script* de la API de JavaScript de Alfresco, los cuales no se ejecutan en el navegador, por el contrario son ejecutados en el servidor.

Java

Java es un lenguaje de programación orientado a objetos, es utilizado en los principales sectores de la industria informática en todo el mundo. Este lenguaje también puede ser usado para la implementación de los servicios ya que Alfresco provee una API de Java. La decisión de usar JavaScript o Java es en dependencia de las necesidades y de los conocimientos del programador, ya que existen muchas funcionalidades que no se pueden implementar usando JavaScript y si es posible mediante Java.

1.6.2. Lenguaje de modelado

UML

El UML (Lenguaje Unificado para la Construcción de Modelos) se define como un "lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas 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 [13]. Este lenguaje permite que personas con poco conocimiento de programación puedan participar en el análisis y diseño de un sistema.

Para la modelación de los artefactos que son generados durante el proceso de desarrollo del *software* se hará uso de este lenguaje de modelado.

1.7. Herramientas

Visual Paradigm

Visual Paradigm es una herramienta CASE (*Computer Aided Software Engineering* o Ingeniería de *Software* Asistida por Ordenador) concebida para soportar el ciclo de vida completo del proceso de desarrollo de *software* a través de la representación de todo tipo de diagramas. La misma proporciona ayudas para el desarrollo de programas informáticos, desde la planificación, el análisis y diseño, hasta la generación de código fuente de los programas.

Teniendo en cuenta las características citadas anteriormente, Visual Paradigm para UML (VP-UML) es la herramienta que se utilizará para la realización del modelado de los diagramas.

1.8. Tecnologías

REST

La Transferencia de Estado Representacional (*Representation State Transfer* - REST) describe un estilo arquitectónico de sistemas en red como, por ejemplo, aplicaciones Web. El término fue utilizado por primera vez en el año 2000 durante una disertación doctoral por Roy Fielding, uno de los principales autores de la especificación HTTP. REST está comprendida por una serie de limitaciones y principios arquitectónicos. Si una aplicación o diseño cumple con esas limitaciones y principios, se considera RESTful [14].

Los principios REST tienen gran importancia para las aplicaciones Web, ejemplos de estos son:

La interacción entre el cliente y el servidor no tiene estado: La solicitud del cliente al servidor debe incluir toda la información necesaria para comprender la solicitud. El cliente no notará si el servidor debe reiniciarse entre las solicitudes. De igual forma, las solicitudes sin estado pueden ser respondidas por cualquier servidor disponible. El cliente puede almacenar los datos en caché para mejorar su rendimiento.

Identificación de recursos: En el servidor, el estado y la funcionalidad de la aplicación se dividen en recursos, como por ejemplo: objetos de aplicación, registros de bases de datos o algoritmos. Cada recurso es de acceso único a través de una URI (*Universal Resource Identifier* – identificador de recursos universal).

Uso de métodos estándar: Para la transferencia de estados entre cliente y servidor los recursos comparten una interfaz uniforme así como el mismo conjunto de métodos. Se usan métodos HTTP como GET, PUT, POST y DELETE que permiten acceder y manejar los diferentes recursos de una forma estándar .

Recurso con múltiples representaciones: ¿Cómo puede un cliente manejar los datos que le devuelve una petición GET o POST? Lo mejor para esto es la separación entre el manejo de los datos y la invocación a las operaciones. El cliente debe especificar el formato de los datos en la petición. Así, la aplicación podrá responderle en un formato que sea capaz de manejar.

Se hace uso de esta tecnología porque la aplicación de sus limitaciones genera una arquitectura simple, escalable, eficiente, segura, confiable y extensible. Mediante la utilización de los servicios web RESTful se aprovecha su capacidad de transmitir datos directamente sobre HTTP. La arquitectura multinivel tanto para servicios web como para aplicaciones Web dinámicas conlleva a la reutilización, simpleza, extensibilidad y a una clara separación de las responsabilidades de los componentes. REST simplifica la implementación tanto para el cliente como para el servidor.

1.8.1. Framework

CodeIgniter

CodeIgniter es un entorno de desarrollo abierto que permite crear webs 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 [15]. Este framework permite enfocarse en el proyecto que se está desarrollando, reduciendo al mínimo la cantidad de código necesario para una tarea determinada.

Se propone el uso de CodeIgniter para la implementación del módulo ya que, desde los inicios eXcriba fue desarrollado con este framenwork, siendo de gran aceptación por los desarrolladores que lo usan, debido a que cuenta con librerías que hacen el trabajo más fácil, permitiendo el ahorro de tiempo, debido a que muchas funcionalidades vienen definidas y solo se reutilizan en la implementación.

jQuery

jQuery es una biblioteca de JavaScript rápida y concisa que simplifica el manejo de eventos, animación y las interacciones Ajax para el desarrollo web rápido. jQuery está diseñado para cambiar la forma en que se escribe JavaScript [16].

Se recomienda el uso de este framework para el desarrollo de las interfaces por la facilidad que brinda las funcionalidades que ya vienen definidas, simplificando la implementación de la propuesta de solución. Además es muy conveniente su uso ya que el eXcriba lo utiliza desde su versión inicial y el equipo de desarrollo tiene dominio del mismo.

Web Script

Un Web Script es simplemente una URI unido a un servicio utilizando los métodos estándar de HTTP, como GET, POST, PUT o DELETE. Los Web Script se pueden escribir usando simplemente la API de JavaScript de Alfresco y las plantillas FreeMarker. Alfresco tiene incluido el framework de Web Script, una API basada en tecnologías RESTful que proporciona una forma fácil, rápida y potente de interactuar con el repositorio de contenido y de integrar Alfresco con otros sistemas. Alfresco Web Scripts implementa la arquitectura MVC (Modelo-Vista-Controlador).

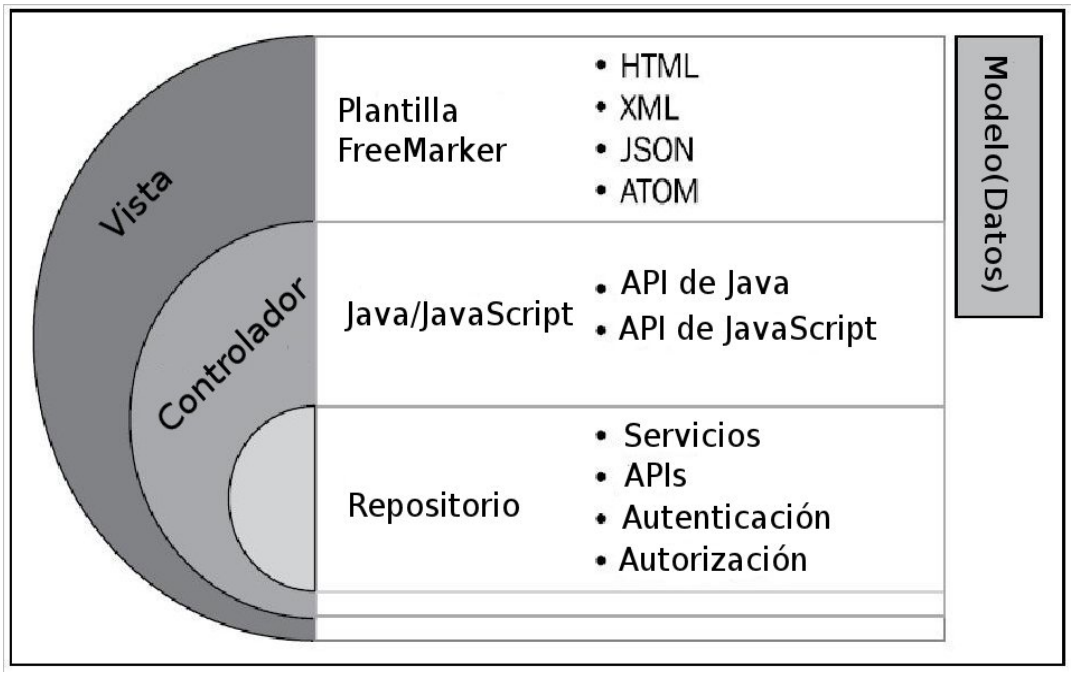


Figura 1.1: Arquitectura de Alfresco Web Scripts [17]

FreeMarker

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.

FreeMarker es un motor de plantillas así como una herramienta genérica para generar la salida de texto basado en plantillas. No es una aplicación para los usuarios finales en sí mismo, sino un paquete de Java que los programadores pueden utilizar para incrustar en sus productos. FreeMarker presenta algunas capacidades de programación, pero aún así no es un verdadero lenguaje de programación. Este no es un framework de aplicaciones Web, aunque es adecuado como un componente de estos, el propio motor FreeMarker no sabe nada acerca de HTTP o servlets, simplemente genera texto.

El motor de plantilla acepta los datos y la plantilla, y se genera un nuevo documento el cual es devuelto según el modelo que se ofrece.

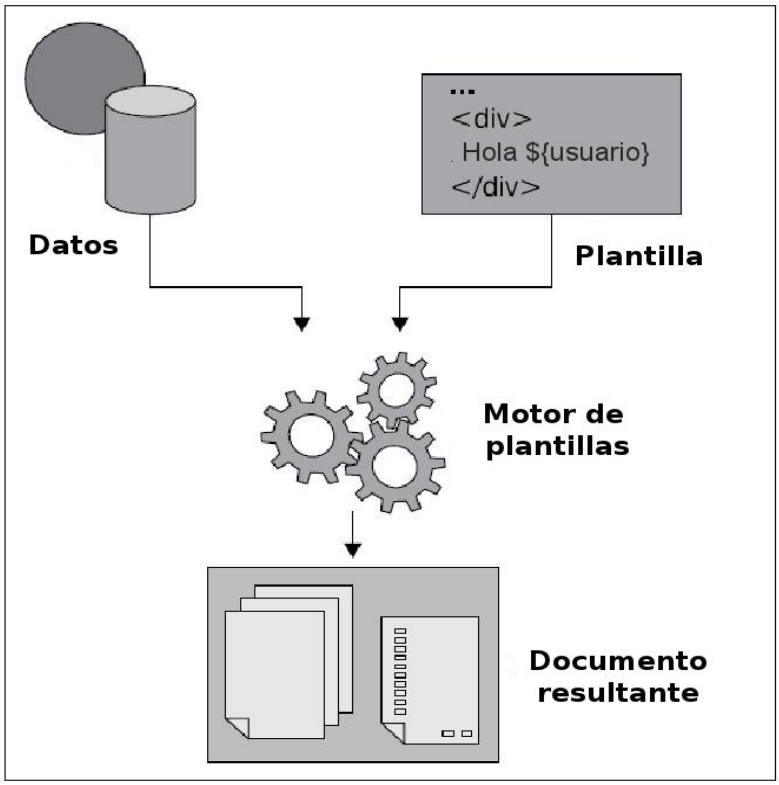


Figura 1.2: Motor de plantillas FreeMarker [17]

API JavaScript

Una API (*Application Programming Interface*) o Interfaz de programación de aplicaciones es un conjunto de funciones que ofrece una biblioteca para ser utilizada por otro *software* como una capa de abstracción.

Alfresco provee una serie de APIs como Alfresco SDK o API de JavaScript. La API de JavaScript es un modelo único para implementar programas y servicios mediante JavaScript. La misma expone todos los objetos del repositorio como objetos de JavaScript que se pueden utilizar. La API sigue el modelo de programación orientado a objetos para los conceptos de Alfresco conocidos como nodos, propiedades, asociaciones y aspectos.

La API de JavaScript es capaz de realizar varias funciones esenciales para el desarrollador de scripts, como por ejemplo:

- Crear y actualizar nodos: Usted puede crear, cargar o actualizar archivos a través de los mismos

- Transformación: Se puede transformar el contenido con esto. Por ejemplo, si desea generar una versión PDF de su documento de MS-Office
- Etiquetado: La API de etiquetado ayudará a crear etiquetas para los contenidos
- Clasificación: Se puede categorizar o clasificar los contenidos que utilizan este
- Personas: Con el uso de esta API, se puede manejar todas las operaciones relacionadas con el usuario y grupo de usuario, como la creación de un nuevo usuario, cambiar la contraseña de un usuario, y así sucesivamente
- Búsqueda: Una de las API más importante y poderosa. Usted puede buscar el contenido haciendo uso de esta API. Puede realizar la búsqueda basada en Lucene o XPath
- Flujo de trabajo: Se puede administrar las tareas y flujos de trabajo en el sistema que utilizan esta API y los servicios
- Operaciones de nodo: Se utiliza esta API para realizar varias funciones relacionadas con el nodo como administrar propiedades, gestionar los aspectos, copiar, eliminar, mover, y así sucesivamente.

En este capítulo han sido tratados aspectos teóricos que constituyen la base fundamental de la investigación. Se abordan con profundidad temas relacionados con los tipos de documentos así como sobre la gestión del control de acceso. Fueron analizados algunos sistemas de gestión documental llegando a conclusiones sobre la seguridad en dichos sistemas. También se identificaron y describieron las principales herramientas, tecnologías y metodología a utilizar, quedando finalmente sentadas las bases para la definición de la propuesta de solución.

Capítulo 2

Propuesta del sistema.

En este capítulo se realizará una descripción de la solución propuesta, proporcionando un mejor entendimiento del sistema. También se especificarán los requisitos funcionales y no funcionales que debe cumplir el módulo así como la descripción y representación de los casos de uso del sistema. Para comprender el entorno en el que trabaja el sistema se incluirá un modelo de dominio y una descripción de los procesos que serán objeto de automatización.

2.1. Problema y situación problémica

Hoy en día la mayoría de las entidades desean tener implementado un sistema de gestión documental con el objetivo de administrar lo mejor posible el flujo de documentos, además de mejorar u optimizar sus procesos. En una entidad todas las personas no realizan las mismas funciones, así como tampoco gestionan los mismos documentos. Existen documentos de un tipo específico como por ejemplo un Acta de advertencia o una Solicitud de traslado que solamente pueden ser creadas por personal autorizado como el director de la entidad. Por lo antes descrito es que los sistemas de gestión documental deben tener en cuenta la seguridad no solo sobre los documentos sino también sobre los tipos de documentos.

El GDA eXcriba posee un subsistema de control de acceso que permite definir las acciones que un usuario puede realizar sobre un espacio de trabajo o documento, pero no sobre las tipologías documentales, lo cual propicia que cualquier usuario pueda clasificar documentos con tipos a los cuales ni si quiera podría tener conocimiento de que existen. No controlar el acceso a los tipos documentales implica que se puedan crear documentos no válidos en el sistema, los cuales pueden estar incluidos en algún proceso de la entidad, provocando un desgaste tanto de recursos como del personal involucrado en el proceso. No restringir el acceso a las tipologías documentales puede afectar de manera indirecta la confidencialidad de la información que se genera en la entidad, ya que se encuentran expuestos todos los tipos de documentos que se manejan en

la misma, siendo una información importante para cualquier persona que desea afectar por alguna vía los intereses de la entidad.

2.2. Propuesta de solución

Para dar solución a la problemática planteada anteriormente, se propone el desarrollo de un módulo para gestionar los permisos sobre los tipos de contenidos y aspectos en el GDA eXcriba. El mismo brindará un conjunto de vistas que le facilitarán al especialista de la entidad gestionar el acceso a los tipos de contenidos y aspectos. El módulo permitirá asignar y denegar permisos sobre un tipo de contenido o aspecto seleccionado, para realizar esta acción el especialista deberá especificar los usuarios y que permisos tienen sobre el mismo. Una vez que han sido asignados los permisos podrán ser listados. El módulo dará la posibilidad al especialista de seleccionar de la lista, los usuarios a los que desea modificar o eliminar sus permisos.

Concluida la implementación del módulo los usuarios solo podrán gestionar los documentos de los tipos a los que tienen permisos. La realización de este módulo será una solución novedosa ya que se concibe la seguridad a los documentos desde otra arista y es a través de los tipos de documentos, dando la posibilidad de restringir la gestión sobre estos.

2.3. Modelo de Dominio

El modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Este modelo se describe mediante diagramas de UML (especialmente mediante diagramas de clases). Estos diagramas muestran a los clientes, usuarios, revisores y a otros desarrolladores las clases del dominio y cómo se relacionan unas con otras mediante asociaciones [18].

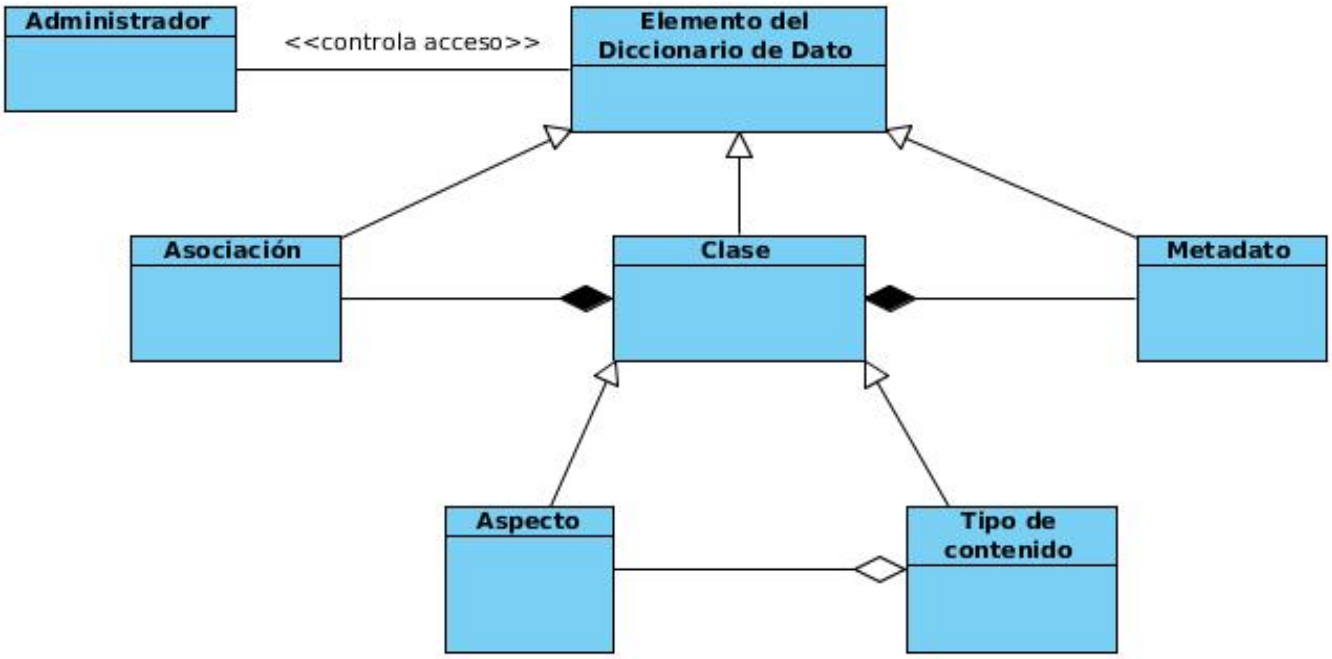


Figura 2.1: Modelo de dominio

El repositorio de Alfresco proporciona soporte para el almacenamiento, manejo y recuperación de los contenidos. Este repositorio soporta un rico diccionario de datos donde se presentan las propiedades, asociaciones y restricciones para describir la estructura de dicho contenido. El diccionario de datos puede ser extendido para describir uno o más modelos de contenidos.

Administrador: Especialista funcional con la responsabilidad de definir quién puede hacer qué sobre cada uno de los elementos del diccionario de datos (tipologías documentales, metadatos y sus relaciones).

Clase: Una clase es un concepto abstracto que se introduce en el diccionario de datos de Alfresco para englobar los elementos comunes entre los tipos de contenido y aspectos.

Tipo de Contenido: Puede ser visto como una clase en la programación orientada a objeto (POO). Son usados para modelar objetos de negocio en una entidad, tienen propiedades y pueden heredar características y comportamientos de un tipo de contenido padre.

Aspecto: Un aspecto puede ser visto como un conjunto de metadatos que se puede aplicar a cualquier tipo de contenido, así como condiciones que se pueden definir para asignar determinado comportamiento.

Metadatos: Pueden ser vistos como los atributos de las clases en la POO, representan las cualidades o características que poseen los tipos de contenidos.

Asociación: Las asociaciones definen las relaciones entre los tipos de contenido. Pueden ser vistas como las relaciones entre clases desde la perspectiva de la POO.

2.4. Especificación de los requisitos de software

Según el estándar 1233 de la IEEE: Guía para el desarrollo de Especificaciones de Requerimientos de Sistemas, un requisito se define como :

- Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
- Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente [19].

Se puede concluir entonces que los requisitos de software son características y funcionalidades que debe cumplir un sistema. Están enfocados hacia todo lo que debe hacer el sistema, el usuario y los miembros del equipo de proyecto. Los requisitos pueden ser clasificados en requisitos funcionales y requisitos no funcionales.

2.4.1. Técnicas para la captura de requisitos

En principio, parece bastante simple preguntar al cliente, a los usuarios y a los que están involucrados en los objetivos del sistema o producto y sean expertos, investigar cómo los sistemas o productos se ajustan a las necesidades del negocio, y finalmente, cómo el sistema o producto va a ser utilizado en el día a día. Esto que parece simple, es muy complicado [20].

Es por ello que surgen diferentes técnicas que ayudan a comprender el problema, proponer soluciones, negociar diferentes puntos de vista y finalmente especificar un conjunto básico de requisitos de la solución.

Ejemplo de estas técnicas pueden ser: entrevistas, cuestionarios, tormentas de ideas, análisis de sistemas existentes, arqueología de documentos, prototipos, entre otras.

Para la realización del levantamiento de requisitos del módulo se aplicaron las técnicas de Tormentas de ideas y Prototipos.

Tormenta de ideas: Técnica basada en lluvias de ideas, implica tanto la generación como la reducción de ideas. Básicamente se busca que los involucrados en un proyecto desarrollen su creatividad, promoviendo la introducción de los principios creativos. Las ideas más creativas e innovadoras resultan con frecuencia de la combinación de ideas aparentemente sin relación. El uso de esta técnica fue muy efectivo pues tanto los clientes como personal del proyecto aportaron ideas para la definición de las funcionalidades básicas que debe tener el módulo.

Prototipos: Un prototipo es un borrador de un producto potencial o de una parte del mismo. Es una simulación de los requisitos. En ocasiones los analistas no pueden continuar su trabajo porque les faltan datos. En esos casos el analista o el resto de las personas involucradas necesitan trabajar con algo más concreto que una lista de requisitos escritos y para esto utilizan un prototipo. El empleo de esta técnica se evidencia en el conjunto de prototipos que fueron presentados al cliente para realizar una evaluación de los mismos, también fueron utilizados para refinar los requerimientos del software a ser desarrollado.

2.4.2. Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, no alteran la funcionalidad del producto, por lo que se mantienen invariables sin importarle con que propiedades o cualidades se relacionan [21].

RF 1. Gestionar control de acceso sobre tipos de contenido.

RF 1.1 Asignar permiso a un tipo de contenido.

RF 1.2 Denegar permiso a un tipo de contenido.

RF 1.3 Modificar permiso a un tipo de contenido.

RF 1.4 Eliminar permiso a un tipo de contenido.

RF 1.5 Listar permisos de un tipo de contenido.

RF 2. Gestionar control de acceso sobre aspectos.

RF 2.1 Asignar permiso a un aspecto.

RF 2.2 Denegar permiso a un aspecto.

RF 2.3 Modificar permiso a un aspecto.

RF 2.4 Eliminar permiso a un aspecto.

RF 2.5 Listar permisos de un aspecto.

RF 3. Buscar tipo de contenido.

RF 4. Buscar usuario.

2.4.3. Requerimientos no funcionales

Los requisitos no funcionales son los requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste, como la fiabilidad y el tiempo de respuesta [21].

- **Usabilidad**

  1. Utilizar el idioma español para los mensajes y textos de la interfaz.
  2. Permitir auto-completado de algunos campos en la interfaz.

- **Fiabilidad**

  1. La precisión y exactitud de las salidas del sistema se corresponden con la calidad y exactitud de la información contenida en las base de datos y de la información introducida por los usuarios del sistema.

- **Portabilidad**

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

- **Soporte**

  1. La estación de trabajo cliente debe tener instalado el navegador Mozilla Firefox 6.X o superior.

- **Legales**

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

- **Restricciones de diseño**

  1. Mantener un sistema de codificación estándar siguiendo las pautas establecidas en el documento de Línea Base de la Arquitectura.
  2. Utilizar servidor web Apache 2.2.
  3. Implementar el módulo en el lenguaje de programación PHP 5.3.
  4. Utilizar CodeIgniter 1.7.2 como marco de trabajo.
  5. Utilizar jQuery 1.3.2 como biblioteca fundamental para el diseño de la interfaz de usuario final.

2.5. Definición de los casos de uso del sistema

Un caso de uso contempla una secuencia de transacciones desarrolladas por un sistema en respuesta a una acción que realiza el actor, de esta forma ayudan a describir lo que el sistema debe hacer.

2.5.1. Definición de los actores

Un actor en un conjunto coherente de roles[1] que los usuarios de casos se uso desempeñan cuando interaccionan con estos casos de uso [22].

[1] El comportamiento específico de una entidad que participa en un contexto particular.

Actores	**Justificación**
Administrador	Especialista funcional con la responsabilidad de definir quién puede hacer qué sobre cada uno de los elementos del diccionario de datos (tipologías documentales, metadatos y sus relaciones).

Tabla 2.1: Definición de los actores

2.5.2. Diagrama de casos de uso del sistema

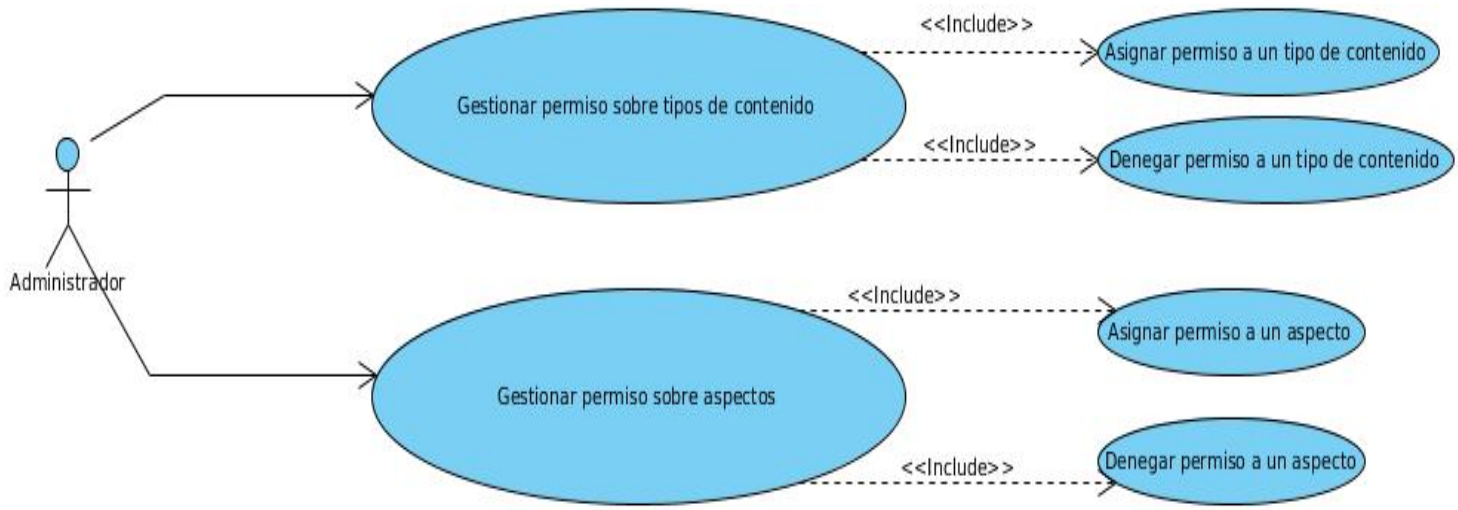


Figura 2.2: Diagrama de casos de uso.

2.5.3. Descripción de casos de uso del sistema

Caso de uso:	**Gestionar permiso sobre tipos de contenido.**
Actor:	Administrador: (Inicia) Asigna, deniega, modifica, ve y elimina los permisos asignados a un usuario o grupo de usuarios.
Descripción:	El caso de uso inicia cuando el administrador desea asignar, denegar, modificar, listar o eliminar permisos a usuarios y grupos sobre un tipo de contenido.
Referencias:	RF 1.3, RF 1.4, RF 1.5, RF 5, RF 6

Tabla 2.2: Definición del caso de uso: Gestionar permiso sobre tipos de contenido.

Caso de uso:	**Gestionar permiso sobre aspectos.**
Actor:	Administrador: (Inicia) Asigna, deniega, modifica, ve y elimina los permisos asignados a un usuario o grupo de usuarios.
Descripción:	El caso de uso inicia cuando el administrador desea asignar, denegar, modificar, listar o eliminar permisos a usuarios y grupos sobre un aspecto.
Referencias:	RF 2.3, RF 2.4, RF 2.5, RF 5, RF 6

Tabla 2.3: Definición del caso de uso: Gestionar permiso sobre aspectos.

Caso de uso:	**Asignar permiso a un tipo de contenido.**
Actor:	Administrador: (Inicia) Asigna permisos a un usuario o grupo de usuarios.
Descripción:	El caso de uso inicia cuando el administrador desea asignar permisos a usuarios y grupos sobre un tipo de contenido.
Referencias:	RF 1.1, RF 5

Tabla 2.4: Definición del caso de uso: Asignar permiso a un tipo de contenido.

Caso de uso:	**Asignar permiso a un aspecto.**
Actor:	Administrador: (Inicia) Asigna permisos a un usuario o grupo de usuarios.
Descripción:	El caso de uso inicia cuando el administrador desea asignar permisos a usuarios y grupos sobre un aspecto.
Referencias:	RF 2.1, RF 5

Tabla 2.5: Definición del caso de uso: Asignar permiso a un aspecto.

Caso de uso:	**Denegar permiso a un tipo de contenido.**
Continúa en la próxima página	

Actor:	Administrador: (Inicia) Deniega permisos a un usuario o grupo de usuarios.
Descripción:	El caso de uso inicia cuando el administrador desea denegar permisos a usuarios y grupos sobre un tipo de contenido.
Referencias:	RF 1.2, RF 5

Tabla 2.6: Definición del caso de uso: Denegar permiso a un tipo de contenido.

Caso de uso:	**Denegar permiso a un aspecto.**
Actor:	Administrador: (Inicia) Deniega permisos a un usuario o grupo de usuarios.
Descripción:	El caso de uso inicia cuando el administrador desea denegar permisos a usuarios y grupos sobre un aspecto.
Referencias:	RF 2.2, RF 5

Tabla 2.7: Definición del caso de uso: Denegar permiso a un aspecto.

Para consultar las descripciones textuales más detalladas de los casos de uso, ver Anexo A de la versión extendida de este documento.

En este capítulo se realizó una descripción de la propuesta de solución, además se definieron los requisitos funcionales y no funcionales que dan respuesta a la problemática existente. Se representó el modelo de dominio y fueron identificados y descritos los usuarios y casos de uso del sistema. Al concluir el capítulo quedaron establecidos los cimientos para la realización posterior de un diseño exitoso.

Capítulo 3

Diseño del sistema.

En el presente capítulo se realiza el modelado del diseño del sistema. Se expone la solución propuesta a través de los diagramas de clases e interacción del diseño, así como una descripción de la arquitectura y de los patrones de diseño utilizados.

3.1. Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Este modelo sirve de abstracción de la implementación del sistema [23].

3.1.1. Descripción de la arquitectura.

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema puesto que establece la estructura, funcionamiento e interacción entre las partes del software [24].

Arquitectura por Capas

Para el desarrollo del módulo se propone el uso de una arquitectura en capas, la cual simplifica la comprensión y la organización del desarrollo del sistema. Este patrón reduce las dependencias, ya que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. La 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.

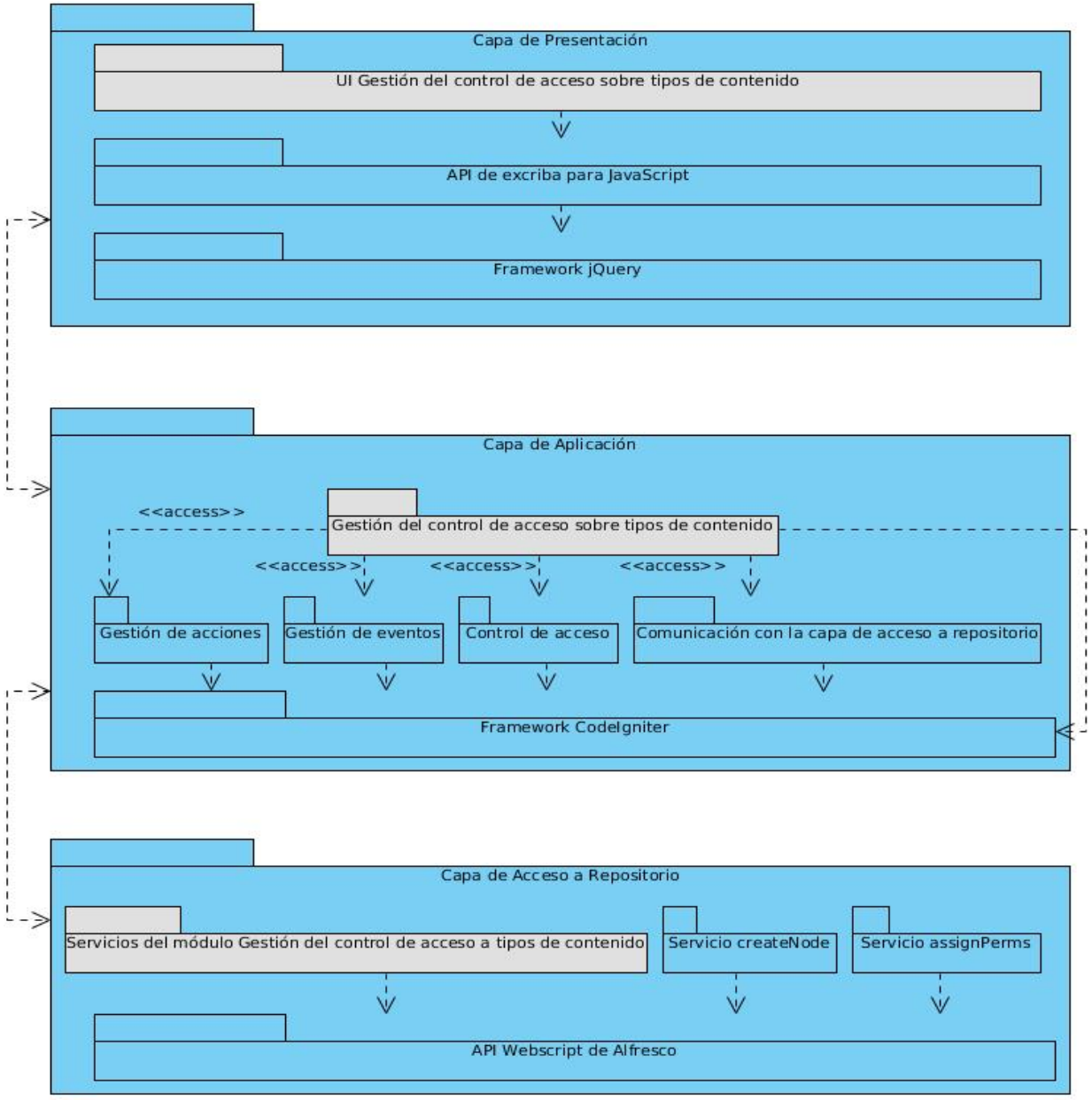


Figura 3.1: Descripción de la arquitectura.

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 términos 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.

3.1.2. Patrones de diseño

Un patrón es una descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas.

Un ejemplo de patrones son los GRASP (*General Responsibility Asignment Software Patterns*), los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Es importante destacar que muchos de estos patrones están estrechamente relacionados ya que por ejemplo, el grado de acoplamiento no puede considerarse aisladamente de otros principios como Experto y Alta Cohesión.

Experto

Este patrón es el encargado de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad [23].

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

Beneficios: Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento. Además alienta la definición de clases "sencillas" y más cohesivas que son más fáciles de comprender y manejar.

Bajo Acoplamiento

La función de este patrón es asignar una responsabilidad para mantener bajo acoplamiento [23]. Una clase con bajo acoplamiento no depende de muchas otras.

Ejemplo: La utilización de este patrón se evidencia en el uso una librería específica para la construcción de cada una de la vistas, lo que evidencia la poca dependencia entre las clases.

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

Alta Cohesión

Es el encargado de asignar una responsabilidad de modo que la cohesión siga siendo alta [23]. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

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

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

Controlador

Asignar la responsabilidad de las operaciones del sistema a 1os objetos situados en la capa del dominio y no en 1os soportes de la capa de presentación [23].

Ejemplo: El empleo de este patrón se evidencia cuando la clase js_asignar_permiso_tipo_contenido, que forma parte de la capa de Presentación, envía los datos para la clase controladora Gestionar_permiso, donde se realiza el procesamiento de la operación, la cual se encuentra en la capa de Aplicación que es la que maneja la lógica del negocio.

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

3.1.3. Diagramas de clases del diseño.

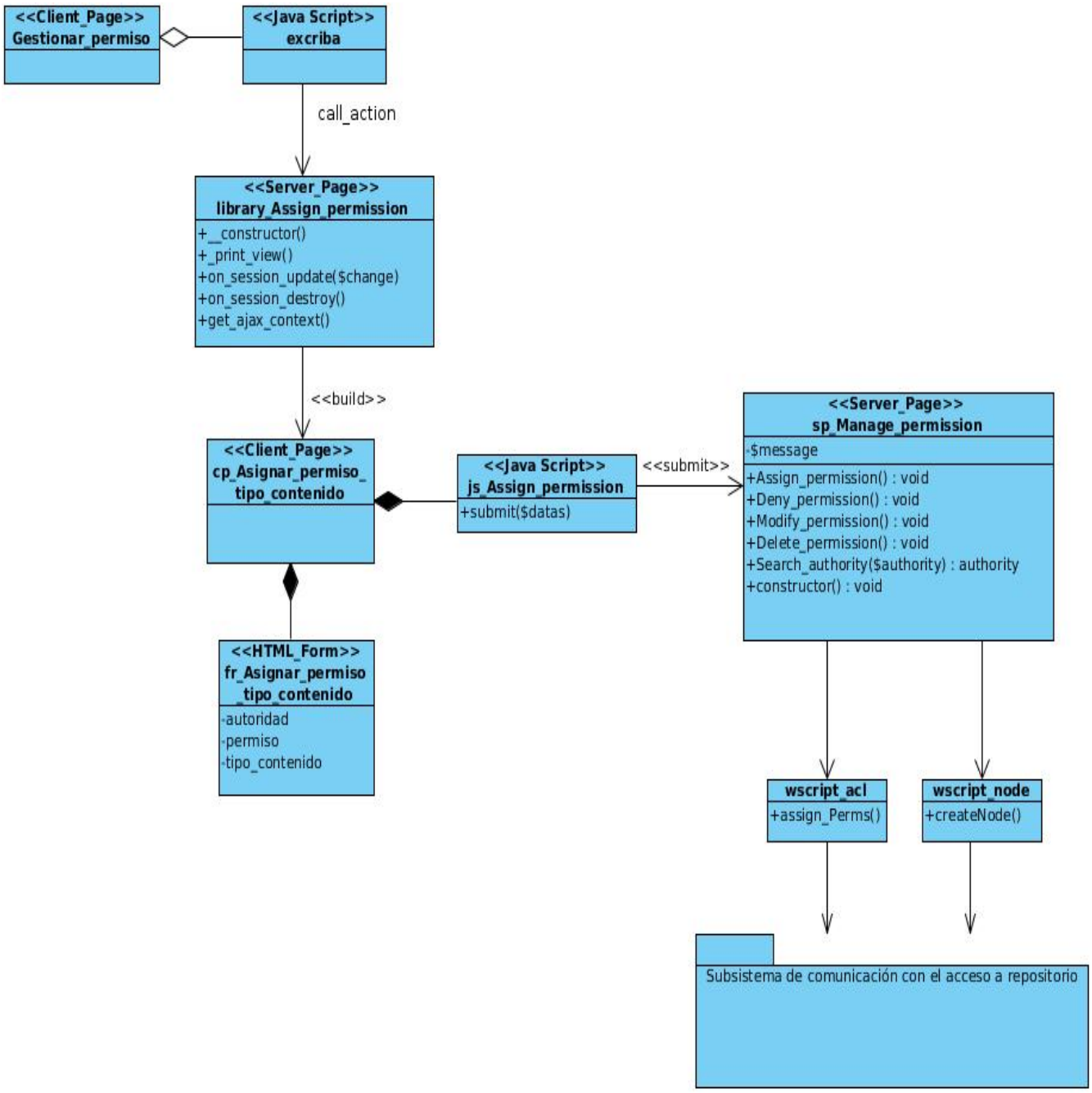


Figura 3.2: Diagrama de clases del CU Asignar permiso a un tipo de contenido.

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

3.1.3.1. Descripción de las clases.

Descripción de las clases del CU Asignar permiso a un tipo de contenido

Nombre: Manage_permission	
Tipo de clase: Controladora	
Atributo	**Tipo**
$message	
Para cada responsabilidad:	
Nombre:	Assign_permission()
Descripción:	Asigna los permisos a un tipo de contenido o aspecto
Nombre:	Deny_permission()
Descripción:	Deniega los permisos a un tipo de contenido o aspecto
Nombre:	Modify_permission()
Descripción:	Modifica los permisos asignados a un tipo de contenido o aspecto
Nombre:	Delete_permission()
Descripción:	Elimina los permisos asignados a un tipo de contenido o aspecto
Nombre:	Search_autority()
Descripción:	Busca una autoridad en el sistema

Nombre: library_Assing_permission	
Tipo de clase: Controladora	
Atributo	**Tipo**
&	
Para cada responsabilidad:	

Nombre:	_print_view()
Descripción:	Muestra la vista para asignar los permisos a un tipo de contenido o aspecto
Nombre:	on_session_update($change)
Descripción:	Ejecuta la acción de asignar permiso
Nombre:	constructor()
Descripción:	Constructor

Nombre: Asignar permiso tipo contenido	
Tipo de clase: Interfaz	
Atributo	**Tipo**
$autoridad	
$permiso	
$tipo_contenido	
Para cada responsabilidad:	
Nombre:	
Descripción:	

Para ver las descripciones de las restantes clases consultar en Anexo C de la versión extendida de este documento.

3.1.3.2. Descripción de los servicios.

Antes de comenzar la explicación de la propuesta de servicios a usar e implementar para el desarrollo del módulo es necesario hacer algunas aclaraciones referentes a la propuesta de solución.

La propuesta incluye la gestión de permisos a los tipos de contenidos base que usa eXcriba para la gestión de contenidos y carpetas, exm: base_content y exm: base_folder respectivamente. Para poder realizar la gestión de permisos sobre los aspectos se propone la creación del aspecto exm: base_aspect dentro del

modelo de contenido del cual heredarían los aspectos. El funcionamiento básico del módulo consiste en que cuando se le asigne permisos a un tipo de contenido seleccionado se cree un nodo asociado a ese tipo de contenido y posteriormente sobre ese nodo se definen que usuarios tienen permiso. El funcionamiento de este módulo en el sistema sería: un usuario ya autenticado desea crear un documento (por ejemplo), cuando el usuario especifique que tipo de contenido tiene el mismo, el sistema busca en el directorio donde se encuentran los nodos a los que ya se le han asignado permiso y verifica en cuales se encuentra ese usuario con ese permiso, devolviendo los nodos encontrados, los cuales son mostrados al usuario para que escoja con cual clasifica al documento.

El primer requisito que es imprescindible desarrollar para poder realizar el resto de las operaciones del módulo es: Buscar tipo de contenido o aspecto.

Para la implementación del mismo se propone la creación de un servicio que devuelva los tipos de contenidos y aspectos. La descripción de este servicio se muestra a continuación.

Servicio: Obtener tipos de contenido		
Paquete: /cu/uci/excriba/models/classes	Plantillas de Respuesta: JSON, XML, HTML	
Descripción: Devolver todos los tipos de contenidos y aspectos.		
Requerimiento de Autenticación: Usuario	Requerimiento de Transacción: Requerida	
Respuesta por defecto:		
URI Absoluta: *http://<nombre_servidor>[:<puerto>]/alfresco/service/cu/uci/excriba/models/classes*	Método HTTP: GET	Ruta Relativa: *alfresco/servicio/cu/uci/excriba/models/classes/*
Dirección del documento de descripción: *classpath:alfresco/extension/templates/webscripts/cu/uci/excriba/models/classes* classes.get.desc.xml		

Figura 3.3: Descripción del servicio: Obtener tipos de contenido.

Descripción de los servicios para la funcionalidad: Asignar permiso a un tipo de contenido.

Para la realización de esta funcionalidad se propone el uso de 3 servicios ya implementados en el Alfresco.

- **create:** Para la creación de un nodo. El cual hace uso de la API Script Node.
- **assignPerms:** Para asignar determinados permisos a usuarios sobre un nodo, haciendo uso para ello de la Permission and Security API.
- **searchAuths:** Devuelve las autoridades según un criterio de búsqueda.

Para el desarrollo de esta funcionalidad es recomendable contar con una implementación que te permita ejecutar dos o más servicios en una misma transacción, evitando así posibles problemas que puedan ocasionar la ejecución de dos servicios por separado.

Descripción de los servicios para la funcionalidad: Listar permiso de un tipo de contenido.

Para la implementación de esta funcionalidad se propone el uso del servicio getPerms.

- **getPerms:** Devuelve un listado con los permisos asociados a un nodo.

Luego de un análisis realizado al webscript, se propone una reimplementación del mismo de manera que devuelva los permisos que han sido asignados y denegados, ya que actualmente solo devuelve los asignados.

Descripción de los servicios para la funcionalidad: Modificar los permisos a un tipo de contenido.

Se expone para el desarrollo de la funcionalidad, utilizar nuevamente el webscript assignPerms ya definido en el Alfresco.

- **assignPerms:** Es preciso especificarle el nodo, la lista de autoridades y la acción (puede ser asignar o eliminar).

Descripción de los servicios para la funcionalidad: Eliminar permiso a un tipo de contenido.

Se propone para dar solución a esta funcionalidad la implementación de un nuevo servicio, el cual se describe a continuación.

Servicio: Eliminar permiso		
Paquete: /cu/uci/excriba/acl/	Plantillas de Respuesta: JSON, XML, HTML	
Descripción: Eliminar los permisos de una autoridad sobre un tipo de contenido.		
Requerimiento de Autenticación: Usuario	Requerimiento de Transacción: Requerida	
Respuesta por defecto:		
URI Absoluta: *http://<nombre_servidor>[:<puerto>]/alfresco/service/cu/uci/excriba/acl/*	Método HTTP: GET	Ruta Relativa: *alfresco/servicio/cu/uci/excriba/acl/*
Dirección del documento de descripción: *classpath:alfresco/extension/templates/webscripts/cu/uci/excriba/acl* elimPerm.get.desc.xml		

Figura 3.4: Descripción del servicio: Eliminar permiso.

3.1.4. Diagramas de interacción del diseño.

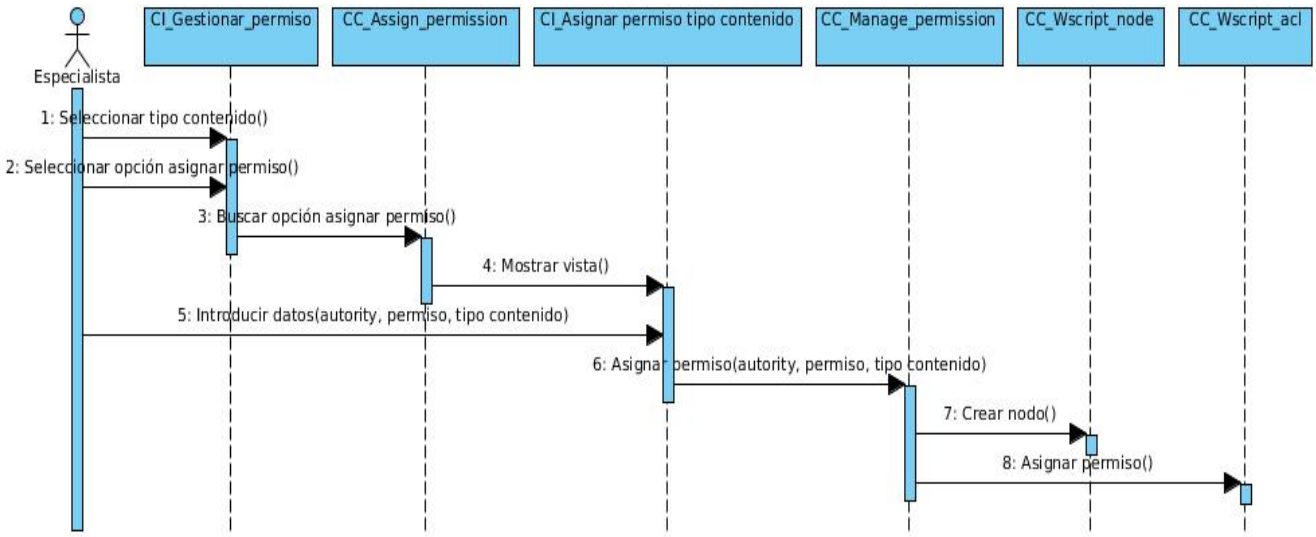


Figura 3.5: Diagrama de secuencia del CU Asignar permiso.

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

En este capítulo se realizó el diseño de la propuesta de solución mediante los diagramas de clases y de interacción del diseño, se describió detalladamente las clases y los servicios propuestos, para la estructuración de los diagramas se especificó la arquitectura, así como se definieron un conjunto de patrones de diseño para obtener modelos más eficientes y reutilizables.

Capítulo 4

Validación de la propuesta.

Un elemento clave de cualquier proceso de ingeniería es la medición. Se emplean medidas para entender mejor los atributos de los modelos que se crean. Pero, fundamentalmente, se emplean las medidas para valorar la calidad de los productos de ingeniería o de los sistemas que son construidos [25]. Durante la investigación se obtuvieron numerosos artefactos, producto de la metodología de software usada. Para validar estos artefactos, en el presente capítulo se aplican distintas métricas en pos de verificar la calidad y efectividad de los mismos.

4.1. Métrica para evaluar los requisitos.

4.1.1. Métrica de la calidad de la especificación.

Para realizar la validación de los requisitos existe toda una lista de características que sugieren el uso de una o más métricas como son: especificidad (ausencia de ambigüedad), corrección, compleción, comprensión, capacidad de verificación, consistencia externa e interna, capacidad de logro, concisión, trazabilidad, capacidad de modificación, exactitud y capacidad de reutilización [26].

Es importante tener en cuenta que para medir las características de la especificación, es necesario conseguir profundizar cuantitativamente en la especificidad y en la completitud.

Especificidad

Para llevar a cabo este proceso se tiene que: n_r representa el número de requisitos del sistema.

$$n_r = n_f + n_{nf}$$
$$n_r = 12 + 11$$
$$n_r = 23$$

Donde n_f es el número de requisitos funcionales y n_{nf} es el número de requisitos no funcionales. Para determinar la especificidad (ausencia de ambigüedad) de los requisitos se sugiere una métrica basada en la consistencia de la interpretación de los revisores para cada requisito.

$$Q = n_{ui}/n_r$$

Donde n_{ui} es el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas. El valor de Q a medida que se acerca a 1, se va disminuyendo la ambigüedad de la especificación.

Con el objetivo de obtener la menor ambigüedad posible, para que los requisitos tengan una mayor claridad de modo que satisfagan las necesidades de los clientes, se llevaron a cabo 2 revisiones con 3 revisores consultados.

En la primera revisión se identificaron algunos requisitos funcionales y no funcionales que presentaban problemas en la redacción y de ambigüedad. Para un total de 23 requerimientos, los revisores tuvieron la misma interpretación para 19 de ellos.

$$Q = 19/23$$
$$Q = 0{,}82$$

En la segunda revisión, ya corregidos los errores identificados en la primera, los revisores tuvieron la misma interpretación para el total de requisitos.

$$Q = 23/23$$
$$Q = 1$$

Figura 4.1: Gráfica de comparación entre la primera y la segunda revisión en la especificidad

Al concluir el estudio cuantitativo realizado a los resultados obtenidos en cada una de las revisiones, se evidencia que los requisitos presentan un grado bajo de ambigüedad.

Compleción

La aplicación de esta métrica siempre devuelve un resultado entre 0 y 1. Mientras más cercano a 1 se encuentre el resultado, indica un alto nivel de completitud en la definición de los requisitos. Este valor se calcula de la siguiente forma:

$$Q_1 = n_a / n_a + n_b$$

n_a: Número de requisitos completos.

n_b: Número de requisitos pobremente especificado.

Para la aplicación de esta métrica se procedió de igual forma que la anterior, arrojando los siguientes resultados:

En la primera revisión:

$$Q_1 = 18/23$$

$$Q_1 = 0.78$$

En la segunda revisión:

$$Q_1 = 22/23$$

$$Q_1 = 0.95$$

Figura 4.2: Gráfica de comparación entre la primera y la segunda revisión en la compleción

Culminada las revisiones los datos arrojados muestran que la mayoría de los requisitos están completamente especificados.

Según los resultados obtenidos con la aplicación de las métricas de Especificidad y Compleción para evaluar la calidad de la Especificación de Requisitos, se evidencia que los mismos están completamente especificados y son de claro entendimiento por el lector.

La siguiente gráfica muestra lo mencionado anteriormente.

Figura 4.3: Gráfica con los datos finales de la aplicación de las 2 métricas

4.1.2. Técnica de Construcción de prototipos

Un prototipo puede considerarse a la implementación concreta de un sistema que se crea para explotar cuestiones sobre aspectos muy diversos durante el desarrollo de un sistema. Estos prototipos permiten la comunicación y participación entre el equipo de desarrollo y el cliente, haciendo de las especificaciones una herramienta fundamental para comprobar los requisitos del software.

Un prototipo no funcional es el diseño de la posible interfaz del software a construir, esta es utilizada para tener un mejor entendimiento del problema y validar los requisitos que solicitó el usuario. La construcción de los prototipos es una alternativa para validar los requerimientos funcionales que fueron capturados durante la etapa de Requerimientos. Para la construcción de la interfaz gráfica se hizo uso de la herramienta Pencil 1.3.4, modelo que satisfizo la mayoría de las expectativas y necesidades de los clientes.

Las interfaces de usuario tienen como ventaja la de reflejar la presentación e interacción de las necesidades de usuario en un entorno amigable y fácil de entender.

En este capítulo se evaluaron algunos de los artefactos obtenidos durante el desarrollo de la investigación. La aplicación de las métricas para evaluar la calidad de la Especificación de requisitos demostró que estas especificaciones se corresponden con las necesidades del cliente y que tienen correcta interpretación por parte del equipo de desarrollo de software.

Conclusiones

Durante el desarrollo de esta investigación se expuso la necesidad de realizar el diseño de un módulo para gestionar el control de acceso sobre los tipos de contenidos y aspectos, arribando a las siguientes conclusiones:

- La gestión del control de acceso sobre las tipologías documentales es un aspecto no incluido en varios sistemas de gestión documental, por lo que constituye una funcionalidad novedosa para el Gestor de Documentos Administrativos eXcriba.
- El uso de técnicas para realizar la captura de requisitos permitió identificar las funcionalidades y cualidades que debe tener el sistema, propiciando un mejor entendimiento del diseño de la propuesta de solución.
- El diseño de la gestión del control de acceso sobre las tipologías documentales contribuirá a la seguridad y autenticidad de los documentos en el Gestor de Documentos Administrativos eXcriba.
- Las métricas aplicadas a la propuesta de solución propiciaron una correcta validación de las funcionalidades y el diseño de la gestión del control de acceso de las tipologías documentales en el Gestor de Documentos Administrativos eXcriba.

Recomendaciones

Para una gestión de control de acceso más completa sobre los tipos de contenidos y aspectos se recomienda lo siguiente:

1. Realizar una implementación que permita ejecutar dos o más servicios en una misma transacción.
2. Hacer más extensiva la propuesta del módulo, incluyendo la gestión del control de acceso sobre las propiedades de los tipos de contenidos.

Referencias bibliográficas

[1] UNE-ISO. *Norma iso 15489. Información y documentación. Gestión de documentos*. 2000. [Consultado: Enero 2012].

[2] Alonso, Vicenta Cortés. *Manual de archivos municipales*. ANABAD, Madrid, 2da edition edition, 1989. ISBN 84-300-81380. [Consultado: Enero 2012].

[3] Mugica, Mayra Mena. *Gestión documental y organización de archivos*. Félix Varela, La Habana, 1st edition edition, 2005. ISBN 959-258-950-X. [Consultado: Enero 2012].

[4] Herrera, Antonia Heredia. *Archivística General. Teoría y práctica*. Sevilla, 5ta edición edition, 1991. ISBN 84-7798-056-X. [Consultado: Enero 2012].

[5] Sistema archivístico de la universidad de almería. http://www.ual.es/Universidad/Biblioteca/servicios/claarch.htm. [Consultado: Enero 2012].

[6] Stell, A. J. and Sinnott, R. O. and Watt, J. P. *Comparison of Advanced Authorisation Infrastructures for Grid Computing*. IEEE Computer Society Washington, DC, USA, 2005. ISBN 0-7695-2343-9. [Consultado: Enero 2012].

[7] Ribagorda, Arturo. *Glosario de Términos de Seguridad de las T.I.* CODA, Madrid, 1997. [Consultado: Enero 2012].

[8] Knowledge tree. http://radar.com.co/productos-y-servicios/gestion-tecnologica/knowledge-tree.html. [Consultado: Marzo 2012].

[9] Smile open source solutions. *Smile, Libro Blaco: Gestion Documental Open Source*. 2008. [Consultado: Febrero 2012].

[10] Sobre alfresco - la alternativa para la gestión de contenidos empresariales de código libre. http://www.alfresco.com/es/about/. [Consultado: Febrero 2012].

[11] Jacobson, Ivar and Booch, Grady and Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. Pearson Educación, S.A, Madrid, 2000. ISBN 84-7829-036-2. 4 pp. [Consultado: Febrero 2012].

[12] Información general de PHP. http://msdn.microsoft.com/es-es/library/cc294979.aspx. [Consultado: Febrero 2012].

[13] Larman, Craig. *UML y PATRONES Introducción al análisis y diseño orientado a objetos*. Prentice Hall , Inc., México, 1999. ISBN 970-17-0261-1. 15 pp. [Consultado: Febrero 2012].

[14] Sun, Bruce. Arquitectura multinivel para la construcción de servicios web RESTful. http://www.ibm.com/developerworks/ssa/library/wa-aj-multitier/index.html. [Consultado: Febrero 2012].

[15] CodeIgniter - open source PHP web application framework. http://codeigniter.com/. [Consultado: Febrero 2012].

[16] jQuery project. http://jquery.org/. [Consultado: Febrero 2012].

[17] , Snig Bhaumik. *Alfresco 3 Cookbook*. Packt Publishing Ltd., Birmingham, July 2011. ISBN 978-1-849511-08-7. [Consultado: Abril 2012].

[18] Jacobson, Ivar and Booch, Grady and Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. Pearson Educación, S.A, Madrid, 2000. ISBN 84-7829-036-2. 112 pp. [Consultado: Febrero 2012].

[19] IEEE. *IEEE:Guide for Developing System Requirements Specification*. 1998. ISBN 1-55937-716-X. [Consultado: Febrero 2012].

[20] Pressman, Roger S. *Ingeniería del Software. Un enfoque práctico*. Mc Graw Hill, España, 5ta edition, 2001. ISBN 8448132149. 172 pp. [Consultado: Febrero 2012].

[21] Sommerville, Ian. *Software Engineering Eighth Edition*. Pearson Education Limited, China, 2006. ISBN 978-0-321-31379-9. 119 pp. [Consultado: Marzo 2012].

[22] Jacobson, Ivar and Booch, Grady and Rumbaugh, James. *El proceso Unificado de Desarrollo de Software*. Pearson Educación, S.A, Madrid, 2000. ISBN 84-7829-036-2. 412 pp. [Consultado: Marzo 2012].

[23] Larman, Craig. *UML y PATRONES : Introducción al análisis y diseño orientado a objetos*. Editorial Félix Varela, La Habana, 2004. ISBN 970-17-0261-1. [Consultado: Marzo 2012].

[24] Arquitecturas de software - EcuRed. http://www.ecured.cu/index.php/Arquitecturas_de_software. [Consultado: Marzo 2012].

[25] Pressman, Roger S. *Ingeniería del Software. Un enfoque práctico*. Mc Graw Hill, España, 5ta edition, 2001. ISBN 8448132149. 323 pp. [Consultado: Abril 2012].

[26] Pressman, Roger S. *Ingeniería del Software. Un enfoque práctico*. Mc Graw Hill, España, 5ta edition, 2001. ISBN 8448132149. 331 pp. [Consultado: Abril 2012].

Bibliografía

- *Los estudios de Tipología Documental Municipal*. Disponible en: *http://www.ucm.es/info/mabillon/articulos/estados/tipologia.htm*. [Consultado: Enero 2012].
- UNE-ISO. *Norma iso 15489. Información y documentación. Gestión de documentos*. 2000. [Consultado: Enero 2012].
- ALONSO, V CORTES. *Manual de archivos municipales*. ANABAD, Madrid, 2da edition, 1989.[Consultado: Enero 2012].
- Mayra Mena Mugica. *Gestión documental y organización de archivos*. La Habana, 2005. [Consultado: Enero 2012].
- Antonia Heredia Herrera. *Archivística General. Teoría y práctica*. Sevilla, 5ta edition, 1991.[Consultado: Enero 2012].
- A. J STELL and D. R. O SINNOTT. *Comparison of Advanced Authorisation Infrastructures for Grid Computing*. IEEE, 2005. [Consultado: Enero 2012].
- A. Ribagorda. *Glosario de Términos de Seguridad de las T.I.* Ediciones CODA, Madrid, 1997.[Consultado: Enero 2012].
- Ivar Jacobson, Grady Booch, and James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*.Pearson Educación, S.A, Madrid, 2000. [Consultado: Febrero 2012].

- Craig Larman. *UML y PATRONES Introducción al análisis y diseño orientado a objetos*. Prentice Hall,Inc., Mexico, 1999. [Consultado: Febrero 2012].
- IEEE. *IEEE:Guide for Developing System Requirements Specification*. 1998. [Consultado: Febrero2012].
- Roger S Pressman. *Ingeniería del Software. Un enfoque práctico*. Mc Graw Hill, Quinta Edición.[Consultado: Marzo 2012].
- Ian Sommerville. *Software Engineering Eighth Edition*. Pearson Education Limited, China, 2006.[Consultado: Febrero 2012].
- Snig Bhaumik. *Alfresco 3 Cookbook*. Packt Publishing, Birmingham, 2011. [Consultado: Febrero 2012].
- Ugo Cei, Piergiorgio Lucidi. *Alfresco 3 Web Services*. Packt Publishing, Birmingham, 2010. [Consultado: Febrero 2012].
- Munwar Shariff, Amita Bhandari, Pallika Majmudar, Vinita Choudhary. *Alfresco 3 Web Content Management*. Packt Publishing, Birmingham, 2010. [Consultado: Febrero 2012].
- Jeff Potts. *Alfresco Developer Guide*. Packt Publishing, Birmingham, 2010. [Consultado: Febrero 2012].
- Munwar Shariff. *Alfresco Enterprise Content Management Implementation*. Packt Publishing, Birmingham, 2006. [Consultado: Febrero 2012].
- David Caruana, John Newton, Michael Farman, Michael G. Uzquiano, Kevin Roast. *Professional Alfresco.Practical Solutions for Enterprise Content Management*. Wiley Publishing, Inc, Indianapolis, 2010. [Consultado: Febrero 2012].
- *Visual Paradigm for UML - UML tool for software application development*. Disponible en: *http://www.visual-paradigm.com/product/vpuml/*. [Consultado: Enero 2012].
- *FreeMarker*. Disponible en: *http://freemarker.sourceforge.net/index.html*. [Consultado: Enero 2012].

- *CodeIgniter User Guide Version 2.1.0*. Disponible en: *http://codeigniter.com/user_guide/*. [Consultado: Enero 2012].
- Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. Irvine, Birmingham, 2000. [Consultado: Febrero 2012].
- *KnowledgeTree 3.7 Administrator Manual*. KnowledgeTree Inc., 2009. [Consultado: Febrero 2012].
- *Smile, Libro Blaco: Gestion Documental Open Source*, Smile open source solutions, 2008.[Consultado: Febrero 2012]
- Nuxeo. *Nuxeo Document Management 5.5 User Guide*, 2010-2012.[Consultado: Febrero 2012].

Anexo A

Primer apéndice

A.1. Descripción de los casos de uso

Caso de uso	**Gestionar permiso sobre tipos de contenido.**
Actor	Administrador: (Inicia) Asigna, deniega, modifica, ve y elimina los permisos asignados a un usuario o grupo de usuarios.
Resumen	El caso de uso inicia cuando el administrador desea asignar, denegar, modificar, listar o eliminar permisos a usuarios y grupos sobre un tipo de contenido.
Prioridad	Media
Complejidad	Media
Referencias	RF 1.3, RF 1.4, RF 1.5, RF 5, RF 6
Precondiciones	El usuario se ha autenticado en el sistema. El usuario tiene los permisos de administrador para poder realizar esta acción. El usuario ha asignado un permiso a algún usuario sobre un tipo de contenido.
Poscondiciones	Se gestionan los permisos a los tipos de contenido.
Flujo de eventos	
Flujo básico <Gestionar permiso sobre tipos de contenido.>	
Actor	**Sistema**
Continúa en la próxima página	

1. Selecciona la opción de gestionar permisos del modelo de contenido.	2. Muestra una interfaz con un buscar, para especificar el tipo de contenido a buscar. Además permite realizar las siguientes acciones sobre el tipo de documento escogido. ▪ Asignar permiso. Ver CU Asignar permiso a un tipo de contenido. ▪ Denegar permiso. Ver CU Denegar permiso a un tipo de contenido. ▪ Modificar permiso. Ver Sección 1: "Modificar permiso a un tipo de contenido". ▪ Eliminar permiso. Ver Sección 2: "Eliminar permiso a un tipo de contenido". ▪ Listar permisos. Ver Sección 3: "Listar permisos de un tipo de contenido".
	3. Termina el caso de uso.
Flujos alternos	
Sección 1: "Modificar permiso a un tipo de contenido"	
Flujo básico <Modificar permiso a un tipo de contenido>	
Actor	**Sistema**
Continúa en la próxima página	

1. Especifica un tipo de contenido a buscar.	2. Muestra el tipo de contenido con un listado de sus metadatos, aspectos y asociaciones.
3. Selecciona el tipo de contenido mostrado.	4. Muestra una tabla con los usuarios y grupos a los que se le han asignado permisos sobre ese tipo de contenido, así como los permisos que fueron asignados y el estado.
5. Selecciona los usuarios o grupos a los que desea modificar los permisos y presiona el botón Modificar.	6. Muestra una ventana donde aparece un listado con los usuarios y grupos seleccionados y da la posibilidad de añadir o quitar algún permiso.
7. Presiona el botón Aceptar.	8. Termina el caso de uso.
Flujos alternos	
4.a No se especificó ningún tipo de contenido	
	4.a.1 Muestra el mensaje de error: "Debe especificar algún tipo de contenido".
	4.a.2 Finaliza el caso de uso.
Sección 2: "Eliminar permiso a un tipo de contenido"	
Flujo básico <Eliminar permiso a un tipo de contenido>	
Actor	**Sistema**
1. Especifica un tipo de contenido a buscar.	2. Muestra el tipo de contenido con un listado de sus metadatos, aspectos y asociaciones.

Continúa en la próxima página

3. Selecciona el tipo de contenido mostrado.	4. Muestra una tabla con los usuarios y grupos a los que se le han asignado permisos sobre ese tipo de contenido, así como los permisos que fueron asignados y el estado.
5. Selecciona los usuarios o grupos a los que desea eliminar los permisos y presiona el botón Eliminar.	6. Muestra el mensaje: “Los permisos han sido eliminados satisfactoriamente”.
	7. Termina el caso de uso.
Flujos alternos	
5.a No seleccionó un usuario o grupo a eliminar	
	5.a.1 Muestra el mensaje: “Debe seleccionar algún usuario o grupo”.
	5.a.2 Finaliza el caso de uso.
Sección 3: “Listar permisos de un tipo de contenido”	
Flujo básico <Listar permisos de un tipo de contenido>	
Actor	**Sistema**
1. Especifica un tipo de contenido a buscar.	2. Muestra el tipo de contenido con un listado de sus metadatos, aspectos y asociaciones.
3. Selecciona el tipo de contenido mostrado.	4. Muestra una tabla con los usuarios y grupos a los que se le han asignado permisos sobre ese tipo de contenido, así como los permisos que fueron asignados y el estado.
	5. Termina el caso de uso.
Flujos alternos	
Continúa en la próxima página	

Tabla A.1: Descripción textual del CU: Gestionar permiso sobre tipos de contenido.

170

Caso de uso	**Asignar permiso a un tipo de contenido.**
Actor	Administrador: (Inicia) Asigna permisos a un usuario o grupo de usuarios.
Resumen	El caso de uso inicia cuando el administrador desea asignar permisos a usuarios y grupos sobre un tipo de contenido.
Prioridad	Alta
Complejidad	Alta
Referencias	RF 1.1, RF 5
Precondiciones	El usuario se ha autenticado en el sistema. El usuario tiene los permisos de administrador para poder realizar esta acción.
Poscondiciones	Se asignan los permisos a los tipos de contenido.
Flujo de eventos	
Flujo básico <Asignar permiso a un tipo de contenido>	
Actor	**Sistema**
1. Especifica un tipo de contenido a buscar.	2. Muestra el tipo de contenido con un listado de sus metadatos, aspectos y asociaciones.
Continúa en la próxima página	

3. Selecciona el tipo de contenido mostrado y presiona el botón Nuevo permiso.	3. Muestra un menú desplegable con diversas opciones para realizar la asignación de permisos. ▪ Asignar permiso. Ver Sección 1: “Asignar permiso”. ▪ Asignar múltiples permisos. Ver Sección 2: “Asignar múltiples permisos”. ▪ Asignar permiso a múltiples usuarios/grupos. Ver Sección 3: “Asignar permiso a múltiples usuarios/grupos”. ▪ Asignar permiso avanzado. Ver Sección 4: “Asignar permiso avanzado”.
Flujos alternos	
1.a No se especificó ningún tipo de contenido	
	1.a.1 Muestra el mensaje de error: “Debe especificar algún tipo de contenido”.
	1.a.2 Finaliza el caso de uso.
Sección 1: “Asignar permiso”	
Flujo básico <Asignar permiso>	
Actor	**Sistema**
Continúa en la próxima página	

1. Selecciona la opción Asignar permiso.	2. Muestra una interfaz con un campo para seleccionar el usuario o grupo al que se le va asignar el permiso, y un campo para especificar el permiso.
3. Inserta el usuario o grupo, el permiso y presiona el botón Aceptar.	4. Termina el caso de uso.
Flujos alternos	
Sección 2: "Asignar múltiples permisos"	
Flujo básico <Asignar múltiples permisos>	
Actor	**Sistema**
1. Selecciona la opción Asignar múltiples permisos.	2. Muestra una interfaz con un campo para seleccionar el usuario o grupo al que se le va asignar el permiso, un campo para especificar el permiso y otro donde se van agregando los permisos especificados.
3. Inserta el usuario o grupo y los permisos.	4. Va agregando los permisos especificados en una lista con una cruz roja al lado por si el usuario desea eliminar el permiso de la lista.
5. Presiona el botón Aceptar.	6. Termina el caso de uso.
Flujos alternos	
5.a Selecciona la cruz roja al lado de un permiso	
	5.a.1 Elimina el permiso de la lista.
Sección 3: "Asignar permiso a múltiples usuarios/grupos"	
Flujo básico <Asignar permiso a múltiples usuarios/grupos>	
Continúa en la próxima página	

Actor	**Sistema**
1. Selecciona la opción Asignar permiso a múltiples usuarios/grupos.	2. Muestra una interfaz con un campo para seleccionar el permiso, un campo para especificar el usuario/grupo, otro donde se van agregando los usuarios/grupos especificados.
3. Inserta el permiso, los usuarios o grupos.	4. Va agregando los usuarios o grupos especificados en una lista con una cruz roja al lado por si el usuario desea eliminar el usuario o grupo de la lista.
5. Presiona el botón Aceptar	6. Termina el caso de uso.
Flujos alternos	
5.a Selecciona la cruz roja al lado de un permiso	
	5.a.1 Elimina el permiso de la lista.
Sección 4: “Asignar permiso avanzado”	
Flujo básico <Asignar permiso avanzado>	
Actor	**Sistema**
1. Selecciona la opción Asignar permiso avanzado.	2. Muestra una interfaz con un campo para especificar un usuario o grupo y otro donde se van añadiendo los usuarios o grupos a los que se les va a añadir los permisos, de igual forma se estructura para los permisos.

Continúa en la próxima página

3. Inserta los usuarios o grupos y los permisos.	4. Va agregando los usuarios o grupos y los permisos especificados en sus respectivas listas con una cruz roja al lado por si el usuario desea eliminar el usuario, grupo o el permiso de la lista.
5. Presiona el botón Aceptar.	6. Termina el caso de uso.
Flujos alternos	
5.a Selecciona la cruz roja al lado de un usuario, grupo o permiso.	
	5.a.1 Elimina el usuario, grupo o permiso de la lista.

Tabla A.2: Descripción textual del CU: Asignar permiso a un tipo de contenido.

Caso de uso	**Denegar permiso a un tipo de contenido.**
Actor	Administrador: (Inicia) Deniega permisos a un usuario o grupo de usuarios.
Resumen	El caso de uso inicia cuando el administrador desea denegar permisos a usuarios y grupos sobre un tipo de contenido.
Prioridad	Alta
Complejidad	Alta
Referencias	RF 1.2, RF 5
Precondiciones	El usuario se ha autenticado en el sistema. El usuario tiene los permisos de administrador para poder realizar esta acción.
Continúa en la próxima página	

Poscondiciones	Se deniegan los permisos a los tipos de contenido.
Flujo de eventos	
Flujo básico <Denegar permiso a un tipo de contenido.>	
Actor	**Sistema**
1. Especifica un tipo de contenido a buscar.	2. Muestra el tipo de contenido con un listado de sus metadatos, aspectos y asociaciones.
3. Selecciona el tipo de contenido mostrado y presiona el botón Nuevo permiso.	3. Muestra un menú desplegable con diversas opciones para realizar la denegación de permisos. ▪ Denegar permiso. Ver Sección 1: "Denegar permiso". ▪ Denegar múltiples permisos. Ver Sección 2: "Denegar múltiples permisos". ▪ Denegar permiso a múltiples usuarios/grupos. Ver Sección 3: "Denegar permiso a múltiples usuarios/grupos". ▪ Denegar permiso avanzado. Ver Sección 4: "Denegar permiso avanzado".
Flujos alternos	
1.a No se especificó ningún tipo de contenido	
	1.a.1 Muestra el mensaje de error: "Debe especificar algún tipo de contenido".
	Continúa en la próxima página

	1.a.2 Termina el caso de uso.
Sección 1: “Denegar permiso”	
Flujo básico <Denegar permiso>	
Actor	**Sistema**
1. Selecciona la opción Denegar permiso.	2. Muestra una interfaz con un campo para seleccionar el usuario o grupo al que se le va denegar el permiso, y un campo para especificar el permiso.
3. Inserta el usuario o grupo, el permiso y presiona el botón Aceptar.	4. Termina el caso de uso.
Flujos alternos	
Sección 2: “Denegar múltiples permisos”	
Flujo básico <Denegar múltiples permisos>	
Actor	**Sistema**
1. Selecciona la opción Denegar múltiples permisos.	2. Muestra una interfaz con un campo para seleccionar el usuario o grupo al que se le va denegar el permiso, un campo para especificar el permiso y otro donde se van agregando los permisos especificados.
3. Inserta el usuario o grupo y los permisos.	4. Va agregando los permisos especificados en una lista con una cruz roja al lado por si el usuario desea eliminar el permiso de la lista.
5. Presiona el botón Aceptar.	6. Termina el caso de uso.
Flujos alternos	
Continúa en la próxima página	

5.a Selecciona la cruz roja al lado de un permiso	
	5.a.1 Elimina el permiso de la lista.
Sección 3: "Denegar permiso a múltiples usuarios/grupos"	
Flujo básico <Denegar permiso a múltiples usuarios/grupos>	
Actor	**Sistema**
1. Selecciona la opción Denegar permiso a múltiples usuarios/grupos.	2. Muestra una interfaz con un campo para seleccionar el permiso, un campo para especificar el usuario/grupo, otro donde se van agregando los usuarios/grupos especificados.
3. Inserta el permiso y los usuarios o grupos.	4. Va agregando los usuarios o grupos especificados en una lista con una cruz roja al lado por si el administrador desea eliminar el usuario o grupo de la lista.
5. Presiona el botón Aceptar.	6. Termina el caso de uso.
Flujos alternos	
5.a Selecciona la cruz roja al lado de un permiso	
	5.a.1 Selecciona la cruz roja al lado de un permiso.
Sección 4: "Denegar permiso avanzado"	
Flujo básico <Denegar permiso avanzado>	
Actor	**Sistema**
Continúa en la próxima página	

1. Selecciona la opción Denegar permiso avanzado.	2. Muestra una interfaz con un campo para especificar un usuario o grupo y otro donde se van añadiendo los usuarios o grupos a los que se les van a denegar los permisos, de igual forma se estructura para los permisos.
3. Inserta los usuarios o grupos y los permisos.	4. Va agregando los usuarios o grupos y los permisos especificados en sus respectivas listas con una cruz roja al lado por si el usuario desea eliminar el usuario, grupo o el permiso de la lista.
5. Presiona el botón Aceptar.	6. Termina el caso de uso.
Flujos alternos	
5.a Selecciona la cruz roja al lado de un usuario, grupo o permiso.	
	5.a.1 Elimina el usuario, grupo o permiso de la lista.

Tabla A.3: Descripción textual del CU: Denegar permiso a un tipo de contenido.

Caso de uso	**Gestionar permiso sobre aspectos.**
Actor	Administrador: (Inicia) Asigna, deniega, modifica, ve y elimina los permisos asignados a un usuario o grupo de usuarios.
Resumen	El caso de uso inicia cuando el administrador desea asignar, denegar, modificar, listar o eliminar permisos a usuarios y grupos sobre un aspecto.
Prioridad	Media

Continúa en la próxima página

Complejidad	Media
Referencias	RF 2.3, RF 2.4, RF 2.5, RF 5, RF 6
Precondiciones	El usuario se ha autenticado en el sistema. El usuario tiene los permisos de administrador para poder realizar esta acción. El usuario ha asignado un permiso a algún usuario sobre un aspecto.
Poscondiciones	Se gestionan los permisos a los aspectos.
Flujo de eventos	
Flujo básico <Gestionar permiso sobre aspectos>	
Actor	**Sistema**

Continúa en la próxima página

1. Selecciona la opción de gestionar permisos del modelo de contenido.	2. Muestra una interfaz con un buscar, para especificar el tipo de contenido a buscar. Además permite realizar las siguientes acciones sobre el tipo de documento escogido. ▪ Asignar permiso. Ver CU Asignar permiso a un aspecto. ▪ Denegar permiso. Ver CU Denegar permiso a un aspecto. ▪ Modificar permiso. Ver Sección 1: "Modificar permiso a un aspecto". ▪ Eliminar permiso. Ver Sección 2: "Eliminar permiso a un aspecto". ▪ Listar permisos. Ver Sección 3: "Listar permisos de un aspecto".
	3. Termina el caso de uso.
Flujos alternos	
Sección 1: "Modificar permiso a un aspecto"	
Flujo básico <Modificar permiso a un aspecto>	
Actor	**Sistema**
1. Especifica un aspecto a buscar.	2. Muestra el aspecto con un listado de sus metadatos.
Continúa en la próxima página	

3. Selecciona el aspecto mostrado.	4. Muestra una tabla con los usuarios y grupos a los que se le han asignado permisos sobre ese aspecto, así como los permisos que fueron asignados y el estado.
5. Selecciona los usuarios o grupos a los que desea modificar los permisos y presiona el botón Modificar.	6. Muestra una ventana donde aparece un listado con los usuarios y grupos seleccionados y da la posibilidad de añadir o quitar algún permiso.
7. Presiona el botón Aceptar.	8. Termina el caso de uso.
Flujos alternos	
4.a No se especificó ningún aspecto	
	4.a.1 Muestra el mensaje de error: "Debe especificar algún aspecto".
	4.a.2 Finaliza el caso de uso.
Sección 2: "Eliminar permiso a un aspecto"	
Flujo básico <Eliminar permiso a un aspecto>	
Actor	**Sistema**
1. Especifica un aspecto a buscar.	2. Muestra el aspecto con un listado de sus metadatos.
3. Selecciona el aspecto mostrado.	4. Muestra una tabla con los usuarios y grupos a los que se le han asignado permisos sobre ese aspecto, así como los permisos que fueron asignados y el estado.

Continúa en la próxima página

5. Selecciona los usuarios o grupos a los que desea eliminar los permisos y presiona el botón Eliminar.	6. Muestra el mensaje: “Los permisos han sido eliminados satisfactoriamente”.
unsrt	7. Termina el caso de uso.
Flujos alternos	
5.a No seleccionó un usuario o grupo a eliminar	
	5.a.1 Muestra el mensaje: “Debe seleccionar algún usuario o grupo”.
	5.a.2 Finaliza el caso de uso.
Sección 3: “Listar permisos de un aspecto”	
Flujo básico <Listar permisos de un aspecto>	
Actor	**Sistema**
1. Especifica un aspecto a buscar.	2. Muestra el aspecto con un listado de sus metadatos.
3. Selecciona el aspecto mostrado.	4. Muestra una tabla con los usuarios y grupos a los que se le han asignado permisos sobre ese aspecto, así como los permisos que fueron asignados y el estado.
	5. Termina el caso de uso.
Flujos alternos	

Tabla A.4: Descripción textual del CU: Gestionar permiso sobre aspectos.

Caso de uso	**Asignar permiso a un aspecto.**
Actor	Administrador: (Inicia) Asigna permisos a un usuario o grupo de usuarios.
Resumen	El caso de uso inicia cuando el administrador desea asignar permisos a usuarios y grupos sobre un aspecto.
Prioridad	Alta
Complejidad	Alta
Referencias	RF 1.1, RF 5
Precondiciones	El usuario se ha autenticado en el sistema. El usuario tiene los permisos de administrador para poder realizar esta acción.
Poscondiciones	Se asignan los permisos a los aspectos.
Flujo de eventos	
Flujo básico <Asignar permiso a un aspecto>	
Actor	**Sistema**
1. Especifica un aspecto a buscar.	2. Muestra el aspecto con un listado de sus metadatos.
Continúa en la próxima página	

3. Selecciona el aspecto mostrado y presiona el botón Nuevo permiso.	3. Muestra un menú desplegable con diversas opciones para realizar la asignación de permisos. ▪ Asignar permiso. Ver Sección 1: “Asignar permiso”. ▪ Asignar múltiples permisos. Ver Sección 2: “Asignar múltiples permisos”. ▪ Asignar permiso a múltiples usuarios/grupos. Ver Sección 3: “Asignar permiso a múltiples usuarios/grupos”. ▪ Asignar permiso avanzado. Ver Sección 4: “Asignar permiso avanzado”.
Flujos alternos	
1.a No se especificó ningún aspecto	
	1.a.1 Muestra el mensaje de error: “Debe especificar algún elemento”.
	1.a.2 Finaliza el caso de uso.
Sección 1: “Asignar permiso”	
Flujo básico <Asignar permiso>	
Actor	**Sistema**
Continúa en la próxima página	

1. Selecciona la opción Asignar permiso.	2. Muestra una interfaz con un campo para seleccionar el usuario o grupo al que se le va asignar el permiso, y un campo para especificar el permiso.
3. Inserta el usuario o grupo, el permiso y presiona el botón Aceptar.	4. Termina el caso de uso.
Flujos alternos	
Sección 2: "Asignar múltiples permisos"	
Flujo básico <Asignar múltiples permisos>	
Actor	**Sistema**
1. Selecciona la opción Asignar múltiples permisos.	2. Muestra una interfaz con un campo para seleccionar el usuario o grupo al que se le va asignar el permiso, un campo para especificar el permiso y otro donde se van agregando los permisos especificados.
3. Inserta el usuario o grupo y los permisos.	4. Va agregando los permisos especificados en una lista con una cruz roja al lado por si el usuario desea eliminar el permiso de la lista.
5. Presiona el botón Aceptar.	6. Termina el caso de uso.
Flujos alternos	
5.a Selecciona la cruz roja al lado de un permiso	
	5.a.1 Elimina el permiso de la lista.
Sección 3: "Asignar permiso a múltiples usuarios/grupos"	
Flujo básico <Asignar permiso a múltiples usuarios/grupos>	
Continúa en la próxima página	

Actor	**Sistema**
1. Selecciona la opción Asignar permiso a múltiples usuarios/grupos.	2. Muestra una interfaz con un campo para seleccionar el permiso, un campo para especificar el usuario/grupo, otro donde se van agregando los usuarios/grupos especificados.
3. Inserta el permiso, los usuarios o grupos.	4. Va agregando los usuarios o grupos especificados en una lista con una cruz roja al lado por si el usuario desea eliminar el usuario o grupo de la lista.
5. Presiona el botón Aceptar	6. Termina el caso de uso.
Flujos alternos	
5.a Selecciona la cruz roja al lado de un permiso	
	5.a.1 Elimina el permiso de la lista.
Sección 4: “Asignar permiso avanzado”	
Flujo básico <Asignar permiso avanzado>	
Actor	**Sistema**
1. Selecciona la opción Asignar permiso avanzado.	2. Muestra una interfaz con un campo para especificar un usuario o grupo y otro donde se van añadiendo los usuario o grupos a los que se les va a añadir los permisos, de igual forma se estructura para los permisos.

Continúa en la próxima página

3. Inserta los usuarios o grupos y los permisos.	4. Va agregando los usuarios o grupos y los permisos especificados en sus respectivas listas con una cruz roja al lado por si el usuario desea eliminar el usuario, grupo o el permiso de la lista.
5. Presiona el botón Aceptar.	6. Termina el caso de uso.
Flujos alternos	
5.a Selecciona la cruz roja al lado de un usuario, grupo o permiso.	
	5.a.1 Elimina el usuario, grupo o permiso de la lista.

Tabla A.5: Descripción textual del CU: Asignar permiso a un aspecto.

Caso de uso	**Denegar permiso a un aspecto.**
Actor	Administrador: (Inicia) Deniega permisos a un usuario o grupo de usuarios.
Resumen	El caso de uso inicia cuando el administrador desea denegar permisos a usuarios y grupos sobre un aspecto.
Prioridad	Alta
Complejidad	Alta
Referencias	RF 1.2, RF 5
Precondiciones	El usuario se ha autenticado en el sistema. El usuario tiene los permisos de administrador para poder realizar esta acción.
Continúa en la próxima página	

Poscondiciones	Se deniegan los permisos a los aspectos.
Flujo de eventos	
Flujo básico <Denegar permiso a un aspecto.>	
Actor	**Sistema**
1. Especifica un aspecto a buscar.	2. Muestra el aspecto con un listado de sus metadatos.
3. Selecciona el aspecto mostrado y presiona el botón Nuevo permiso.	3. Muestra un menú desplegable con diversas opciones para realizar la denegación de permisos. ▪ Denegar permiso. Ver Sección 1: "Denegar permiso". ▪ Denegar múltiples permisos. Ver Sección 2: "Denegar múltiples permisos". ▪ Denegar permiso a múltiples usuarios/grupos. Ver Sección 3: "Denegar permiso a múltiples usuarios/grupos". ▪ Denegar permiso avanzado. Ver Sección 4: "Denegar permiso avanzado".
Flujos alternos	
1.a No se especificó ningún aspecto	
	1.a.1 Muestra el mensaje de error: "Debe especificar algún elemento".

Continúa en la próxima página

	1.a.2 Termina el caso de uso.
Sección 1: “Denegar permiso”	
Flujo básico <Denegar permiso>	
Actor	**Sistema**
1. Selecciona la opción Denegar permiso.	2. Muestra una interfaz con un campo para seleccionar el usuario o grupo al que se le va denegar el permiso, y un campo para especificar el permiso.
3. Inserta el usuario o grupo, el permiso y presiona el botón Aceptar.	4. Termina el caso de uso.
Flujos alternos	
Sección 2: “Denegar múltiples permisos”	
Flujo básico <Denegar múltiples permisos>	
Actor	**Sistema**
1. Selecciona la opción Denegar múltiples permisos.	2. Muestra una interfaz con un campo para seleccionar el usuario o grupo al que se le va denegar el permiso, un campo para especificar el permiso y otro donde se van agregando los permisos especificados.
3. Inserta el usuario o grupo y los permisos.	4. Va agregando los permisos especificados en una lista con una cruz roja al lado por si el usuario desea eliminar el permiso de la lista.
5. Presiona el botón Aceptar.	6. Termina el caso de uso.
Flujos alternos	
Continúa en la próxima página	

5.a Selecciona la cruz roja al lado de un permiso	
	5.a.1 Elimina el permiso de la lista.
Sección 3: "Denegar permiso a múltiples usuarios/grupos"	
Flujo básico <Denegar permiso a múltiples usuarios/grupos>	
Actor	**Sistema**
1. Selecciona la opción Denegar permiso a múltiples usuarios/grupos.	2. Muestra una interfaz con un campo para seleccionar el permiso, un campo para especificar el usuario/grupo, otro donde se van agregando los usuarios/grupos especificados.
3. Inserta el permiso y los usuarios o grupos.	4. Va agregando los usuarios o grupos especificados en una lista con una cruz roja al lado por si el usuario desea eliminar el usuario o grupo de la lista.
5. Presiona el botón Aceptar.	6. Termina el caso de uso.
Flujos alternos	
5.a Selecciona la cruz roja al lado de un permiso	
	5.a.1 Selecciona la cruz roja al lado de un permiso.
Sección 4: "Denegar permiso avanzado"	
Flujo básico <Denegar permiso avanzado>	
Actor	**Sistema**
Continúa en la próxima página	

1. Selecciona la opción Denegar permiso avanzado.	2. Muestra una interfaz con un campo para especificar un usuario o grupo y otro donde se van añadiendo los usuarios o grupos a los que se les va a denegar los permisos, de igual forma se estructura para los permisos.
3. Inserta los usuarios o grupos y los permisos.	4. Va agregando los usuarios o grupos y los permisos especificados en sus respectivas listas con una cruz roja al lado por si el usuario desea eliminar el usuario, grupo o el permiso de la lista.
5. Presiona el botón Aceptar.	6. Termina el caso de uso.
Flujos alternos	
5.a Selecciona la cruz roja al lado de un usuario, grupo o permiso.	
	5.a.1 Elimina el usuario, grupo o permiso de la lista.

Tabla A.6: Descripción textual del CU: Denegar permiso a un aspecto.

Anexo B

Segundo apéndice

B.1. Diagramas de clases del diseño

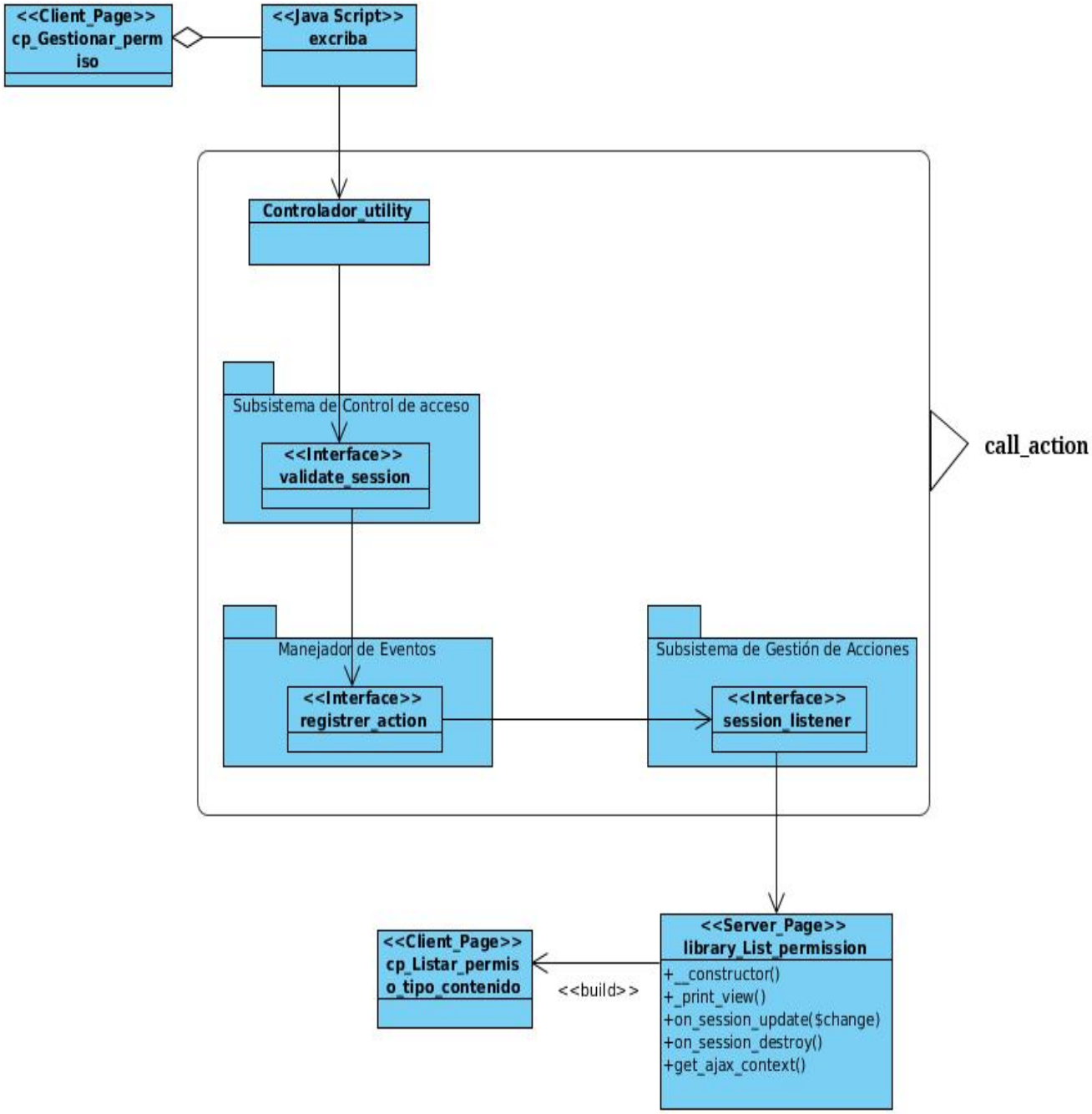


Figura B.1: Diagrama de clases del CU Listar permisos de un tipo de contenido.

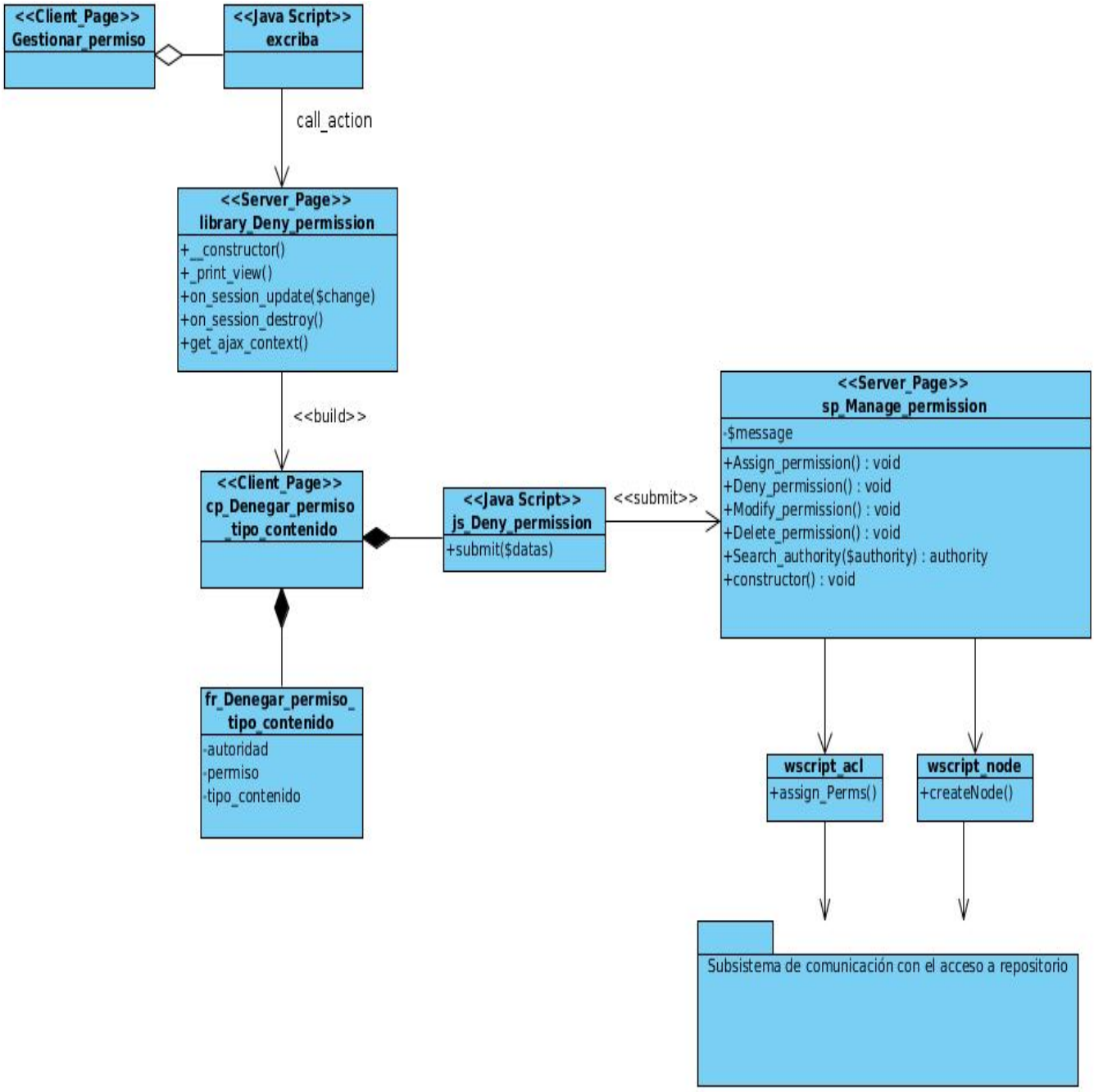


Figura B.2: Diagrama de clases del CU Denegar permiso a un tipo de contenido.

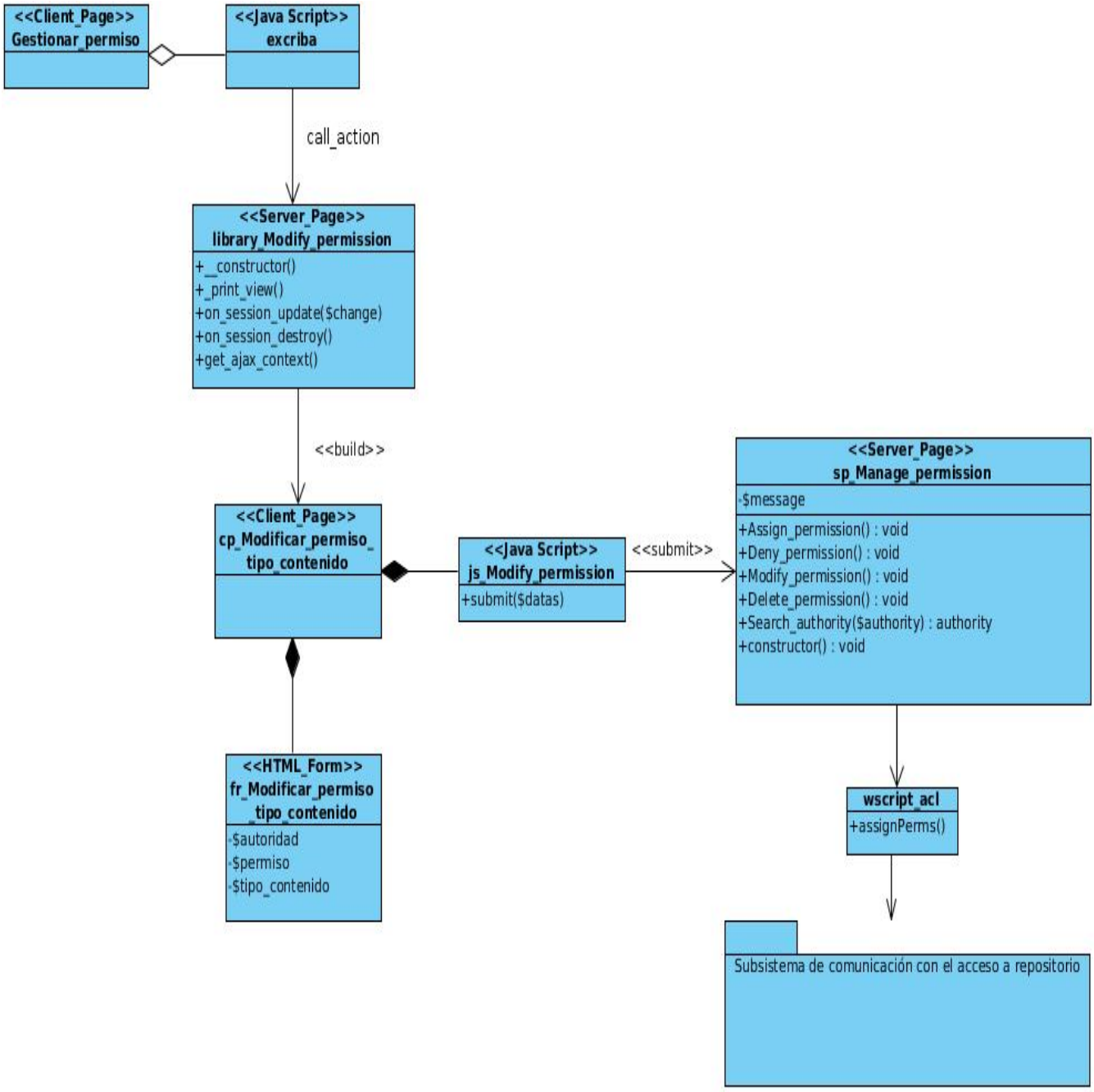


Figura B.3: Diagrama de clases del CU Modificar permiso a un tipo de contenido.

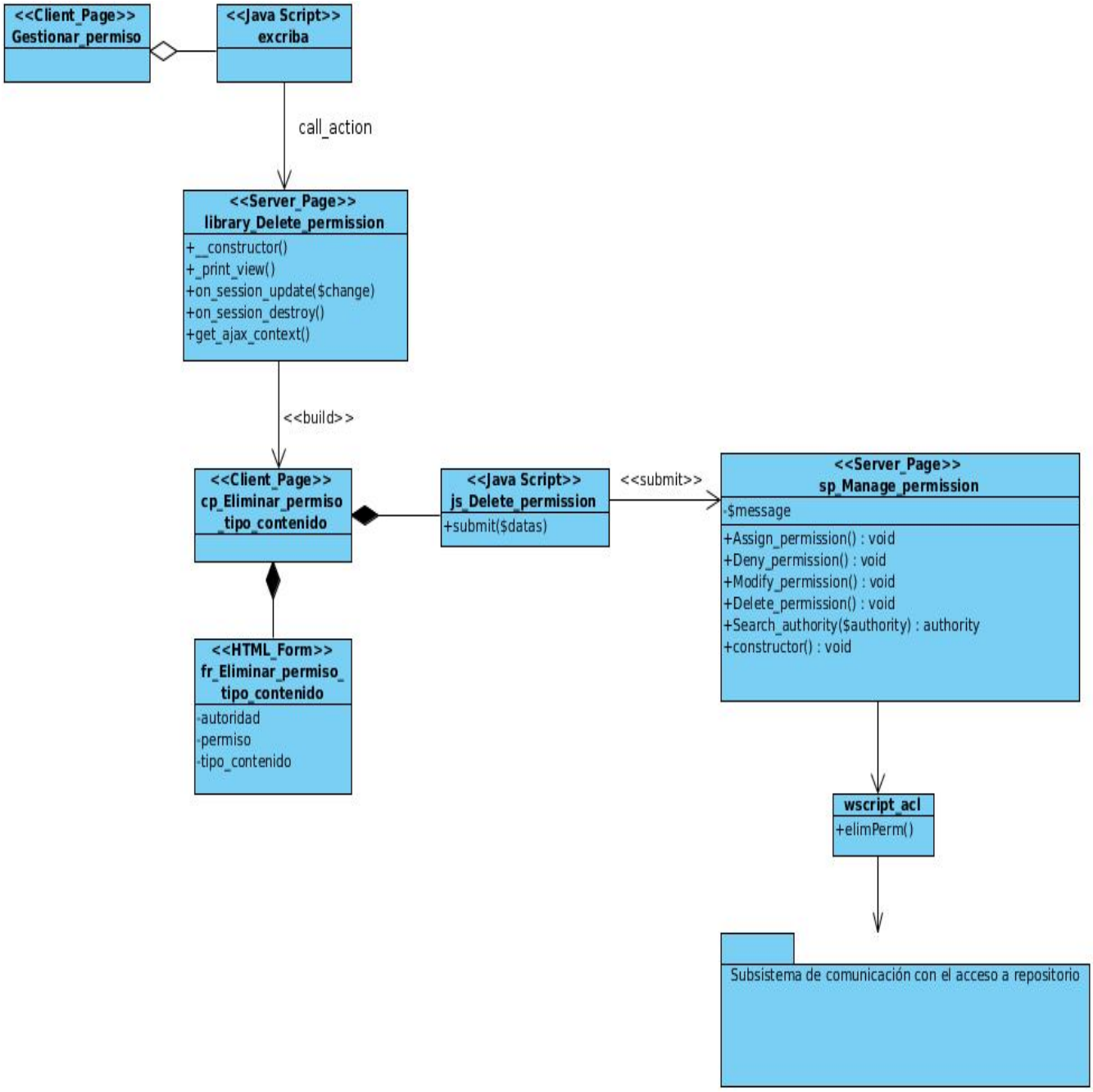


Figura B.4: Diagrama de clases del CU Eliminar permiso a un tipo de contenido.

Anexo C

Tercer apéndice

C.1. Descripción de las clases

Descripción de las clases del CU Denegar permiso a un tipo de contenido

Nombre: Manage_permission	
Tipo de clase: Controladora	
Atributo	**Tipo**
$message	
Para cada responsabilidad:	
Nombre:	Assign_permission()
Descripción:	Asigna los permisos a un tipo de contenido o aspecto
Nombre:	Deny_permission()
Descripción:	Deniega los permisos a un tipo de contenido o aspecto
Nombre:	Modify_permission()
Descripción:	Modifica los permisos asignados a un tipo de contenido o aspecto
Nombre:	Delete_permission()
Descripción:	Elimina los permisos asignados a un tipo de contenido o aspecto
Nombre:	Search_autority()
Descripción:	Busca una autoridad en el sistema

Nombre: library_Deny_permission	
Tipo de clase: Controladora	
Atributo	**Tipo**

&	
Para cada responsabilidad:	
Nombre:	_print_view()
Descripción:	Muestra la vista para denegar los permisos a un tipo de contenido o aspecto
Nombre:	on_session_update($change)
Descripción:	Ejecuta la acción de denegar permiso
Nombre:	constructor()
Descripción:	Constructor

Nombre: Denegar permiso tipo contenido	
Tipo de clase: Interfaz	
Atributo	**Tipo**
$autoridad	
$permiso	
$tipo_contenido	
Para cada responsabilidad:	
Nombre:	
Descripción:	

Descripción de las clases del CU Modificar permiso a un tipo de contenido

Nombre: Manage_permission	
Tipo de clase: Controladora	
Atributo	**Tipo**
$message	
Para cada responsabilidad:	

Nombre:	Assign_permission()
Descripción:	Asigna los permisos a un tipo de contenido o aspecto
Nombre:	Deny_permission()
Descripción:	Deniega los permisos a un tipo de contenido o aspecto
Nombre:	Modify_permission()
Descripción:	Modifica los permisos asignados a un tipo de contenido o aspecto
Nombre:	Delete_permission()
Descripción:	Elimina los permisos asignados a un tipo de contenido o aspecto
Nombre:	Search_autority()
Descripción:	Busca una autoridad en el sistema

Nombre: library_Modify_permission	
Tipo de clase: Controladora	
Atributo	**Tipo**
&	
Para cada responsabilidad:	
Nombre:	_print_view()
Descripción:	Muestra la vista para modificar los permisos a un tipo de contenido o aspecto
Nombre:	on_session_update($change)
Descripción:	Ejecuta la acción de modificar permiso
Nombre:	constructor()
Descripción:	Constructor

Nombre: Modificar permiso tipo contenido
Tipo de clase: Interfaz

Atributo	**Tipo**
$autoridad	
$permiso	
$tipo_contenido	
Para cada responsabilidad:	
Nombre:	
Descripción:	

Anexo D

Cuarto apéndice

D.1. Diagramas de secuencia

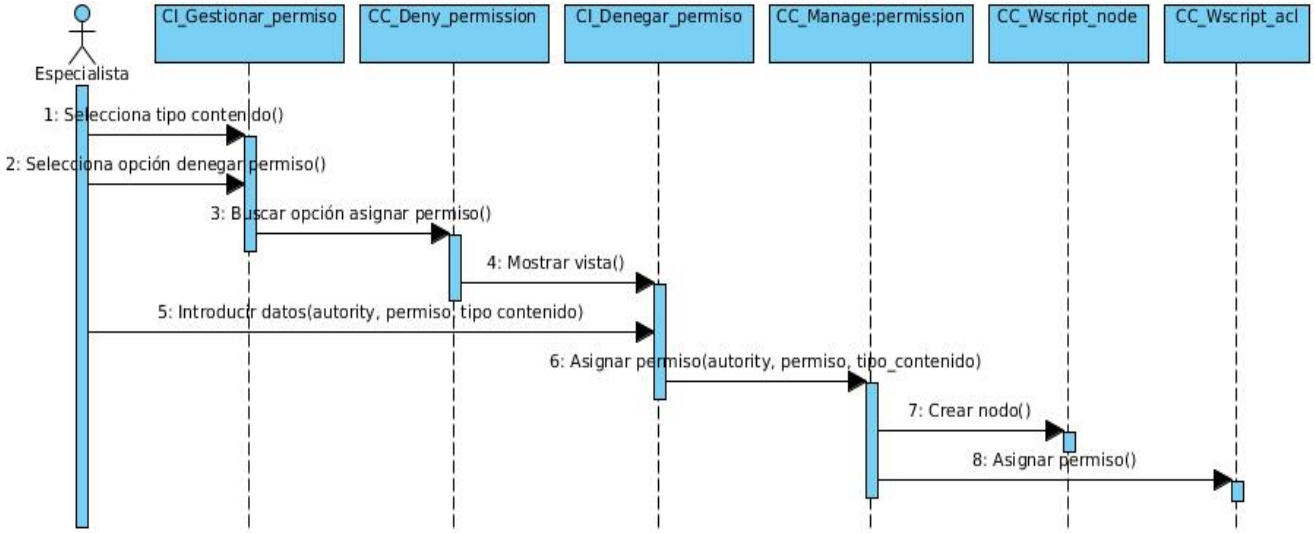


Figura D.1: Diagrama de secuencia del CU Denegar permiso.

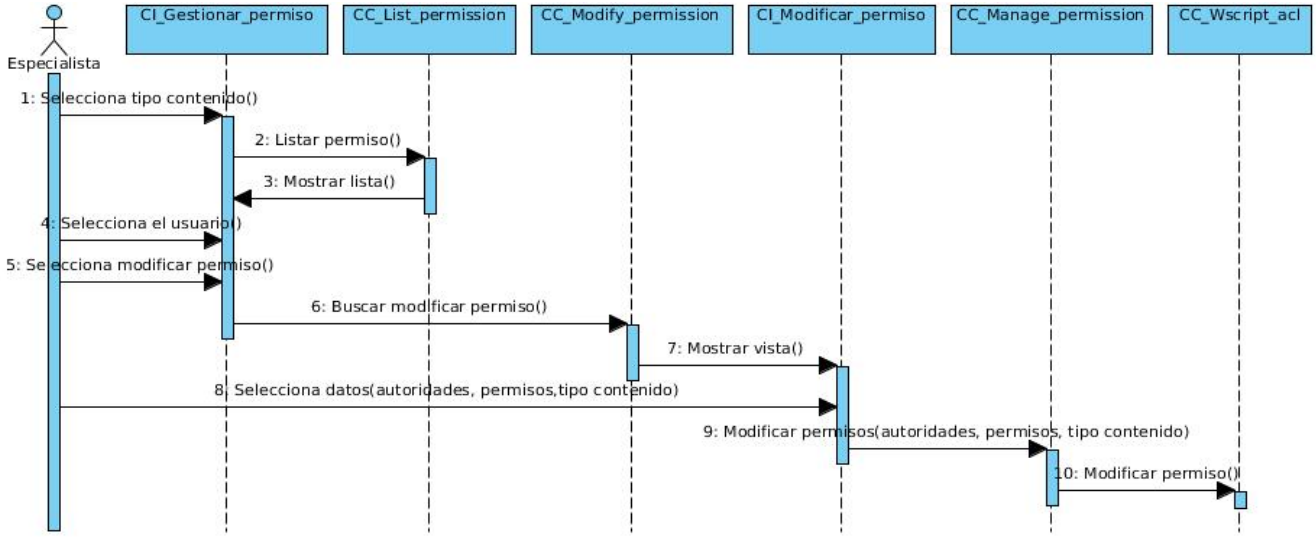


Figura D.2: Diagrama de secuencia del CU Modificar permiso.

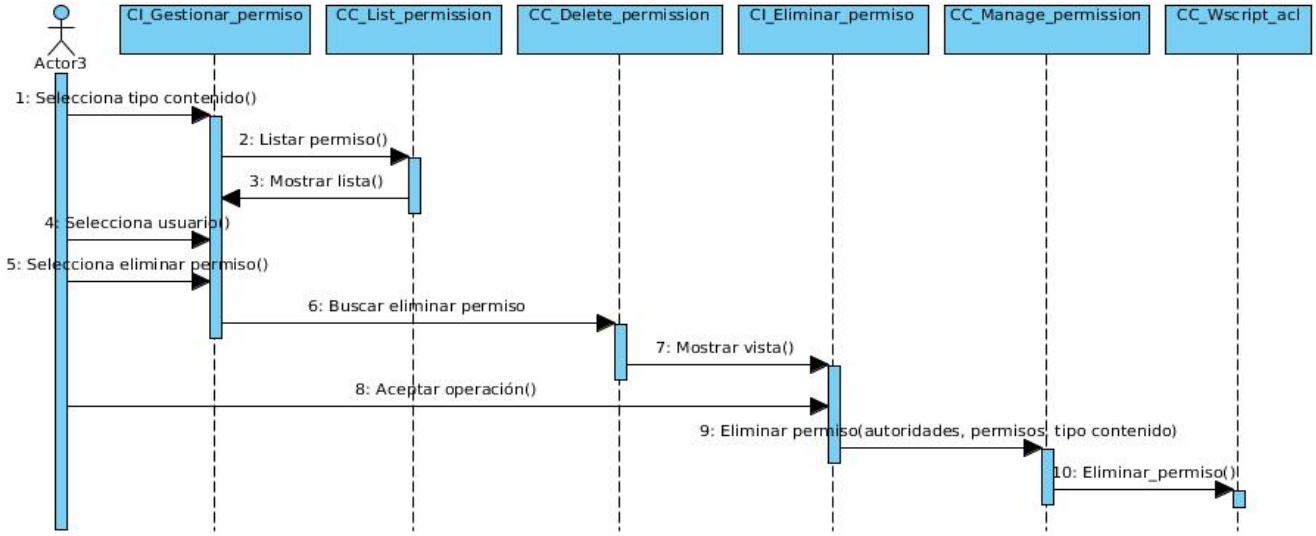


Figura D.3: Diagrama de secuencia del CU Eliminar permiso.